Marcus Uneson
Fil kand (musikvetenskap)

Burcas

A Simple Concatenation-based MIDI-to-Singing Voice Synthesis System for Swedish

D-uppsats i fonetik, ht 2002
Handledning: Joost van de Weijer

Abstract

After a brief outlook on the field of concatenative synthesis of singing, with emphasis on the differences in comparison to synthesis of speech, the present paper gives an overview of a simple system for singing synthesis in Swedish based on concatenation of diphones. The system, called Burcas, accepts as input a text file for lyrics, from which it extracts a target phoneme sequence using basic letter-to-sound conversion, and a MIDI file—possibly holding multiple parts—, from which it extracts melodical information, i.e. note duration and frequency. After associating a syllable (or a part of a syllable) to each note, a simple model of segment durations is used to calculate the duration of each segment of the syllable. Finally, segment data are then used as control parameters (allophone, duration, frequency) for the MBROLA speech generator. The speech generator outputs sound files in standard format, given a suitable diphone database.

In a concluding section, the far more sophisticated corpus-based approach to concatenative synthesis of singing is considered.

Contents

1. Introduction

Using the voice is arguably the most human of human behaviours. The voice is, of course, the obvious tool when we employ the most human of human abilities—the faculty of language. However, the voice may express quite a few things outside a linguistic message—supplementing, suppressing, perhaps even contradicting it.

Occasionally, in fact, the expressive task of the voice dominates over the linguistic. It is only human to now and then use the voice as a built-in, will-controlled, incredibly versatile sound-producing tool—for music, certainly, but likewise for, say, imitation of sounds, including other voices. As Cook (1998) puts it:

> The human voice is the most ubiquitous, flexible, and general of acoustic instruments. (...) Most functions of this instrument we take for granted, but huge regions of our brains are dedicated to controlling and perceiving the sounds made by it. (...) The voice can exact independent control across a broad range of pitch, amplitude, brightness, harmonicity, noise amount, and spectral shape.

An area of particular interest, then, is singing, where, as it were, the expressive and the linguistic functions of the voice meet. Modelling singing is modelling something very human (quite apart from the fact that the modelling itself is another particularly human behaviour).

The present paper at least enters this fascinating area (although admittedly a bit short on the modelling part). It describes the most important phonetic aspects of Burcas[i], a system for simple singing synthesis in Swedish. Burcas takes as input a MIDI file and a text file containing arbitrary lyrics in Swedish, syllabified (as is common practice for song texts) but otherwise in ordinary orthography. Melodical information—durations and frequencies of notes—is extracted from the MIDI file, and a rough letter-to-sound (LTS) conversion of the text file yields a target phoneme sequence, retaining the syllable boundaries. For each syllable, durations of the component phonemes (or rather allophones) are then calculated according to a simple model. The allophone, frequency, and associated time-points determined in this manner are used as control parameters for the MBROLA (MBROLA www) speech generator, which outputs audio files in standard format, one per voice in the MIDI file. The system thus relies on a prerecorded diphone database and the

MBROLA speech generator for all digital signal processing (DSP). In fact, it could quite accurately be described as an alternative front-end to a basic text-to-speech synthesis system (TTS), replacing the intonation model with MIDI data.

However, although the purpose of the work described here has been to construct a reasonably working concatenative-based singing synthesis system for Swedish, the paper aims at a more general perspective (as far as the scope of the term paper permits, that is). The field is new and there are really no surveys. Thus, some principal opportunities and challenges of the method are discussed as well along the way—challenges associated with concatenative synthesis per se, or with concatenative synthesis of singing, or with singing synthesis for Swedish. In particular, the promising corpus-based synthesis scheme, by far more ambitious than the one employed in Burcas, is discussed in a concluding section.

This is a paper in phonetics, rather than, say, computer science, electrical engineering, or language technology. Requirement specifications, especially the underlying considerations and trade-offs from a phonetic point of view, are paid more attention than implementational details. Algorithms are generally not commented at all.

Furthermore, as stated above and as will be repeated several times in the following, the scope is limited. The field, by contrast, is practically boundless. Producing a working system involves some crude simplifications to keep the task tractable, and sticking to the general, surveying ambition means a cut in empirical data. For instance, no listener tests have been made. For related reasons, the recording of a sung database has not been considered part of the work. Of course, the perceived naturalness of the system depends largely on the database used, and a system aiming at synthesis of singing should ultimately employ a database constructed from sung language (rather than spoken). However, most of the work when constructing such a database consists of tedious routine chores, of little research interest in itself.

The paper is organized as follows. Next section (2) takes a general view on synthesis of singing, with emphasis on the differences between synthesizing singing and synthesizing speech. Some possible applications are also mentioned, and properties peculiar to the concatenative approach are touched upon. Section 3 in turn describes the specification and premises of Burcas: the restrictions of the system, its interfaces and building-blocks (the MIDI standard, the MBROLA generator, and the currently used database). In section 4, focus rests on the current implementation of Burcas; its anatomy is

accounted for from a high-level point of view, with emphasis on the phonetically relevant parts: the letter-to-sound conversion, and the modelling of segment frequencies and durations. The following section (5) attempts at a preliminary evaluation of the system, pointing out weaknesses. (The section is short, however—this fact should be ascribed to sparsity of testing occasions rather than to sparsity of flaws.)

Finally, as hinted at above, section 6 describes some aspects of an interesting area for future research: singing synthesis based on unit selection from a large corpus, following the predominant trend in speech synthesis during the latest years.

An earlier version of the system was presented at the conference "Fonetik 2002" at the Royal Institute of Technology, Stockholm (Uneson 2002).

2. Singing synthesis

This section describes some general characteristics of synthesis of singing as opposed to synthesis of speech, regardless of the technique chosen. Thereafter, some possible applications are hinted at, and the strengths and weaknesses of the concatenation-based approach are briefly reviewed.

2.1 Synthesis of singing versus synthesis of speech

When synthesizing the human voice a great number of challenges meet. Many of them are, of course, common to singing synthesis and the more explored field of speech synthesis. For good overviews of the latter, see Klatt 1987, Carlson & Granström 1997, Dutoit 1997, Lemetty 1999.

The present paper rather concentrates on the differences between synthesis of singing and synthesis of speech. There are indeed several important aspects in which the two differ considerably (Sundberg 1986; review in Macon 1996; Meron 1999). In the following, these differences have been divided into those that make synthesis of singing more difficult than synthesis of speech, and those that make it less so. It should be noted that the perspective taken is the concatenative (see further section 2.3), where large manipulations inherently are more awkward than small; for other synthesis schemes, the differences are of course the same but their classification possibly another.

Synthesis of singing: more difficult...

Thus, synthesis of singing is more difficult than synthesis of speech in the following aspects:

1) In singing, syllables may occasionally be extremely short, but more often, vowels and (sometimes) sonorant consonants are much longer than in speech, with durations up to several seconds (figure 1 illustrates this).

2) In singing, the pitch movements are wider and more rapid than in speech. Furthermore, as can be seen in figure 2, also the range of the pitch is wider. The long-term average pitch is generally higher.

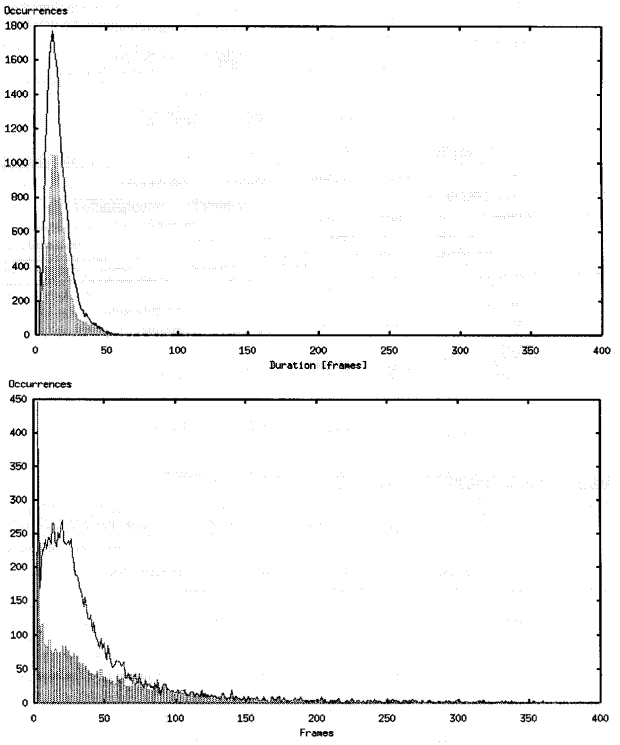

Figure 1. Distribution of segment durations, extracted from automatic segmentation of Japanese speech (top) and German singing (bottom; the music is Schubert, Winterreise) databases. The dark line represents all phonemes; the grey bars indicate vowels only. (Meron 1999, figure 1.3)

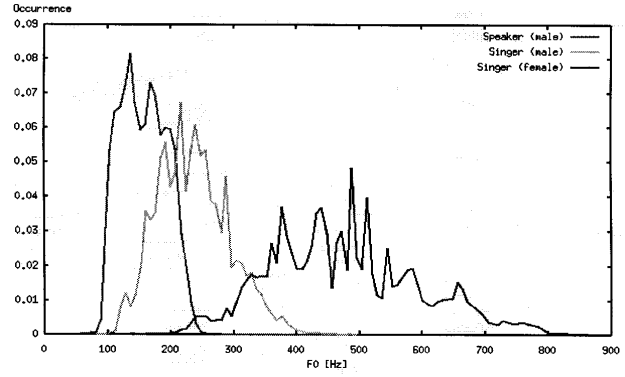

Figure 2. Distribution of fundamental frequency for male and female singer (Winterreise), and male speaker (Japanese). (Meron 1999, figure 1.1)

3) In singing, various means of musical expression add static or dynamic acoustic properties to the sounding output, the most important of which are

a) *Timbre*: a generic term referring to overall voice quality (of e.g. a certain singer, a certain interpretation, or a certain phrase of an interpretation). Possible ways of varying timbre include adding nasality or phonation types such as creaky or breathy voice; the acoustic correlates are generally elusive. An interesting instance is the so-called *singer's formant* in male singers (for some of them, and for some genres only, e.g. the operatic), a fixed resonance which appears at 2.5 to 3 kHz independent of F0 and vocal tract shape. A related phenomenon among female singers in operatic style is *formant shift*, the tendency to shift formant locations for a given vowel in response to pitch changes, accomplished by a learned method of varying articulator positions when singing; by so doing, a gain in timbre (according to the genre-specific norm) and overall intensity is accomplished.

b) *Dynamics*: controlled variations over time of loudness and vocal effort, perceptual qualities whose most obvious acoustic correlates are a change in overall amplitude and in spectral tilt. Dynamics may be regarded as a special case of controlling timbre.[ii]

c) *Vibrato*: a (more or less) controlled quasi-periodic low-frequency modulation of frequency (and, as a consequence, of amplitude and spectrum as well) on sustained vowels, typically evolving dynamically with time.

d) *Portamento*: perceivable pitch movement within a given note in the direction of (the next note of) the melody. To some extent, such a movement is physiologically conditioned and unavoidable. Portamento refers only to the conscious use, as a means of musical expression.

e) *Voice onset* (attack): in particular, rapid or slow evolution to full tone.

Generally, such traits have no communicatively relevant counterpart in speech. Their presence and manifestation is dependent on individual taste and on performance practice (genre conventions), the latter presenting a

spectrum ranging from melodic speech ("Sprachgesang") to operatic performance. For instance, the actual delivery of vibrato is highly dependent on genre, somewhat less on individual style. In some genres (e.g. some styles of Scandinavican folk music) it is hardly used at all. Similarly, in percussion-oriented musical cultures, as in Western Africa, attack is generally fast or 'hard'. The balance between individual and genre-specific considerations varies widely.

... and less

On the other hand, with regard to the following, synthesis of singing is easier than synthesis of speech (from text):

1) In singing, the otherwise enormous problem of modelling intonation, crucial for a naturally sounding output in any language, is already solved—or rather bypassed—by the composer (or by the singer, if a specific performance is modelled).

2) In singing, the problem of modelling segment durations, especially important for languages with quantity oppositions, is less crucial.

3) In singing, overall voice quality is more important than in speech; however, the quality of *individual* vowels is less critical—for one thing, the harmonics are more sparsely distributed due to the generally higher F0 (in fact, the fundamental may occasionally lie above F1 for female singers) and therefore the formant locations are less precisely determinable for the listener.

4) In singing (at least for Swedish), the difficulties of modelling coarticulation and reduction phenomena according to speech rate and/or speech style are less annoying; what may sound stiff and hyperarticulated when spoken usually passes without notice when sung[iii]. More generally stated: knowing a lexeme means knowing its pronunciation in singing; phrase-level modelling (other than the implicit one made by the composer) is not crucial.

5) In singing, unlike in speech, it is quite normal to prepare an interpretation of a piece (and then to rehearse it) for some time; for synthesis, then, some preparations and/or finishing might be acceptable (e.g. the letter-to-sound conversion need probably not be perfect—texts are generally short and the

time needed to correct occasional LTS errors is anyway little compared to that of preparing the musical input).

The first three items may be regarded as different facets of a general property of singing: some of the phonological information present in speech (much of which is redundant anyway) is sacrificed on the altar of musical expression (Macon 1996). That is, in singing, phonological contrasts tend to diminish or dissolve, or, with yet another phrasing, musical performance is more important than intelligibility of lyrics.

Although the matter has not been systematically studied from a cross-linguistic perspective, there are several examples from unrelated languages, concerning very different aspects of phonology. Thus, in Mandarin (but not in Cantonese), lexical tones are generally ignored when lyrics are put to a melody (Chan 1987). According to a study concerning lament singing in Estonian, the acoustic correlates of the three-way quantity opposition of disyllabic words, crucial to speech, is largely lost in singing (Ross & Lehiste 1994).

The list could probably be much extended by researchers so inclined.[iv] Here are some of the author's informal observations on singing in Swedish (mostly gathered from choirs):

* word accent type, 'grave' or 'acute', a distinction typical of most Scandinavian dialects, generally does not influence a composer when writing a melody;

* two words forming a minimal pair with these two accents are generally indistinguishable when sung to a given melody (although they may arguably in some cases exhibit subtle perceptual differences: there may be secondary correlates, like aspiration time, or vowel quality of unstressed syllable);

* it is not uncommon for (some) choir leaders to give impressionistic instructions along the lines "make that i-vowel more e-like for reasons of sonority";

* the two-way quantity opposition of (stressed) syllables, which in contemporary standard Swedish distinguishes syllables with long vowel + short consonant from syllables with short vowel + long consonant, tends to dissolve for syllables over a certain duration (not very long)—the

consonants will have standard values, whether they pertain to an unstressed syllable, or a stressed syllable with short consonant, or a stressed syllable with long consonant (the vowels then will, as it were, take what is left).

2.2 Applications

The possible applications for synthesis of singing are perhaps not as abundant as those for synthesis of speech, but nevertheless there are a few interesting options. Thus, a system for singing synthesis may function as a musicological or phonetic research tool for producing controlled perceptual stimuli of singing, say, for studying temporal aspects of phrasing in different musical genres (say, jazz, ethnic music, lullaby, rap). It might serve as a didactic aid in music education. Simple entertainment applications (singing web pages, or singing e-mail) are also conceivable (although potentially rather annoying).

Of more practical benefit is the possibility for composers and arrangers of vocal music to have synthetic voices as stand-ins for human singers, perhaps not at a performance, but rather at draft stage. In particular, such a system might be useful as a plug-in in high-end notation software. Even a wordless (and thereby language-independent) voice may be more attractive than today's workaround (often a sampled wind instrument for the voice part/s). Such a 'vocalese' voice need not be very sophisticated to be helpful; the capacity of producing, say, perhaps the 18 possible V or CV syllables using /a/, /u/, /o/, /m/, /t/, /d/, /l/, /h/ may be enough for some applications. A system capable of producing any sung text in a given language—and perhaps also of some rendering of the most important musical expressions, such as dynamics and vibrato—is of course even more useful. However, requirements of DSP and language-specific linguistic processing of input text increase accordingly.

For any of the applications mentioned above, timbre is not crucial. Indeed, in some cases, such as for perceptual stimuli, a neutral, non-specialized and genre-independent voice quality might even be preferable to a more bel-canto style of singing. It is possible, however, to raise the ambition even further, in regarding the synthesized sound not as a working tool, but rather as a musical piece of art. Most things certainly remain to be done, but such ‘high quality singing synthesis’ no longer seems inacheivable. See section 6 for a few examples.

In addition, research in singing synthesis may conceivably provide ideas for making speech synthesis more expressive and less machine-like in the future, if and where this is desirable.

2.3 On the concatenative approach to singing synthesis

Concatenative synthesis, and others

In comparison to speech synthesis, little work has been done on computer synthesis of the singing voice, and most of it has explored other paths than the concatenative. Although not further commented in this paper, mention may be made of the SPASM system by Perry Cook et al (Cook www), which builds on a graphically interfaced articulatory vocal tract model; the general CHANT formant waveform synthesis by Xavier Rodet et al (Rodet www), and formant synthesis methods explored by Johan Sundberg (as in MUSSE, Berndtsson and Sundberg 1993).

Attempts at concatenative synthesis of singing are few in numbers. Of course, a trivial reason is that the entire technique is fairly new. It is computationally expensive, especially in terms of memory, and it is only quite recently that hardware that fills the demand has become generally available.

Another reason, however, may be rather ideological. At least historically, researchers in singing (as well as in speech) seem to have been more attracted by the "purer" rule-based synthesis, which aims at a more general modelling of relevant phenomena—of vocal articulation and acoustics, of course, but also of general acoustic properties of sound, not necessarily only those produced by the human vocal tract. The CHANT system, for instance, has also been used for synthesis of musical instruments.

It is certainly true that the explanatory value from an acoustic or articulatory point of view is little in concatenative synthesis. Whereas speech, or singing, in other synthesis schemes is described as the evolution over time of a number of parameters—acoustic, or articulatory, or both—concatenative synthesis *encodes* rather than *models.* Thus, a lot of low-level difficulties are bypassed (as in any system which builds on sampling of existing sounds) by handling rather opaque building-blocks. Then again, this strategy could be regarded as a practical way of concentrating resources; some questions may be left for later, if it facilitates the study of other

questions right now. For singing, for instance, more effort may be put into models of timbre if consonantal transitions are cared for elsewhere.

Overview. Sound model

In concatenative synthesis, put in one sentence, units to concatenate are chosen from a prerecorded database, glued together, and manipulated prosodically. Obviously, three critical issues for the performance of such a system, be it for singing or speech, are the quality of the database, of the concatenation method, and of the prosody modification. For the first two, most problems are common to speech and singing; concatenation will here be commented only briefly, and the database itself not at all. However, while the modification of prosody (including spectral properties) is important to speech, it is absolutely decisive for singing, and thus requires more attention.

In a concatenative system, the signal is modelled in one way or another; the analysis and synthesis algorithms refer to that model (Dutoit 1997). The choice of model is not trivial. For instance, the sinusoidal model (see below) handles periodic sounds well and makes related spectral modifications rather easy (such as, e.g., vibrato or changes in spectral tilt); but it is computationally rather demanding and less apt for representing speech sounds involving stochastic components, such as voiceless sounds.

Concatenation

In speech, as in singing, junctures between concatenation units are crucial. Discontinuities between segments may introduce artifacts. Concatenation points should ideally be placed in steady-states only, with matching F0, intensity, spectral energy distribution, and phase of the segments to be concatenated. However, while employing smoothing methods certainly helps, occasional mismatches are difficult to avoid. For instance, for some segments, such as glides (in Swedish, mainly /j/), no clear steady-state is to be identified; they are thus notoriously difficult to concatenate (Dutoit 1997).

Modification of prosody

In most concatenative synthesis systems (even those with very large databases), some means of modifying prosody is indispensable, allowing the manipulation of prosodic parameters with as little deterioration of sound quality as possible. For speech, the predominant technique for the latest years has been the PSOLA (pitch-synchronous overlap-add) algorithm, developed by CNET, the research organization of France Telecom. The various versions of PSOLA all perform resynthesis of the waveform in

basically the same manner. First, in an analysis stage, the speech waveform is divided into short-term windowed (usually Hanning) signals, moving through the signal by glottal pulses ('pitch-synchronously') for voiced segments (and by some fixed interval for unvoiced speech). Then, in the synthesis stage, these short-term signals are again multiplied by windows and recombined ('overlap and add'). By repeating or deleting short-term segments in synthesis, duration may be manipulated; by altering the time spacing between them, pitch may be scaled (Lemmetty 1999, Dutoit 1997).

PSOLA has proved very useful for the phonetic research community, offering the possibility to manipulate prosodically interesting parameters while retaining voice quality. As mentioned, several versions have been proposed. The simplest and perhaps most widely used is the time-domain approach, TD-PSOLA, which handles small modifications of time and pitch very well. In front of all, it is computationally very efficient and may even be used for real-time high-quality synthesis. The chief drawback is inflexibility. The structure of TD-PSOLA is non-parametric—it does not model the speech signal in any explicit way (and is therefore known as a 'null' model, Stylianou 2001). Spectral manipulations are impossible, and for large time- or pitch manipulations, the naturalness decreases considerably. Repetition of unvoiced short-term signals may result in local periodicity, perceived as tonal noise. Spectral smoothing at concatenation boundaries is not possible.

Some of these problems have been addressed in later versions, especially those relating to concatenation point artifacts. For instance, in MBR-PSOLA (Dutoit et al 1996), which lies behind the MBROLA speech generator used in Burcas, the segment inventory is resynthesized with the expensive multi-band-excited resynthesis (MBR) algorithm. Pitch and phase are made uniform once and for all when building the database, which makes the crucial glottal pulse identification easier (and also allows for simple spectral interpolation at concatenation points).

The impressing performance/cost ratio of TD-PSOLA aside, for some applications, flexibility may be the first and foremost concern. This is certainly true for high quality singing synthesis: prosodic and spectral modification within wide limits and little distortion are more important desiderata than computational efficiency.

Convenient spectral manipulation in the frequency domain requires another representation of the signal than the null model. One important approach is sinusoidal modelling (SM), in which a steady-state of the speech signal is represented as the sum of a small number (typically 20-80) of

sinusoids with constant amplitude and constant frequency. By introducing time-varying amplitudes and frequencies and allowing the introduction and deletion ('births and deaths') of sinusioids, the entire signal may be approximated. There are several techniques within this framework. For singing synthesis, ABS/OLA (analysis by synthesis/overlap-add) has been tried by Macon (Macon 1997), and a hybrid between sinusoidal modelling and PSOLA known as SM-PSOLA by Meron (1999) (see section 6). A rather new model which has shown promising results for speech synthesis is HNM (harmonics plus noise model), which represents speech signals as a time-varying harmonic component plus a modulated noise component (Stylianou 2001).

An issue specific to singing (at least for the basic diphone approach) is the handling of segments with long durations: synthesis of extended segments inevitably involve looping (that is, a set of analysis frames extracted from the steady-state of the vowel is repeated). However, sampled sounds, when looped, are often perceived as lifeless and rigid by human listeners, and this is particularly true for sampled voices. From a human listener's point of view, they lack several important cues of naturalness: the natural pitch fluctuations that are typical of the human voice, and (in singing) important expressive means such as dynamic changes in vocal effort, timbre, or vibrato (cf section 2.2). Such expressions must be catered for separately, by DSP or by being included in an appropriately tagged database. All of them are much more conveniently implemented in the frequency domain.

Corpus-based synthesis

An alternative—or rather, supplementary—way of mitigating problems related to DSP is to simply record more tokens to choose from in the database, and to have them annotated in some appropriate way. In this way, one may minimize the necessary pitch/time-scalings and the number of concatenation points, or even conceivably offer different voice qualities—a wider spectrum of spectra, as it were.

More sophisticated concatenation-based systems for speech nowadays often employ large corpora rather than fixed unit inventories (Sagisaka 1988, Black and Campbell 1995; general review in Möbius 2000). This technique is attractive for singing, as well. Section 6 presents it briefly and offers a few considerations on its use for this purpose.

3. Burcas: premises

This section outlines the premises of Burcas: the restrictions of the system; the existing building-blocks (the MIDI standard and the MBROLA speech generator), and their interfaces. For details about questions specific to the implementation, see next section.

3.1 Restrictions

Most of the applications outlined in section 2.2 require (among other things) a reasonably user-friendly interface, with sensible error handling. It should be noted that Burcas currently does not offer any of those. Burcas is a modest first attempt at concatenation-based singing synthesis in Swedish; at present, it only possesses a crude command-line user interface and rudimentary error handling. While it certainly is highly desirable to remedy such deficiencies in the long run, it is not considered top priority for the purposes of the small, experimental, phonetically oriented zero-budget project described in this paper.

However, it is important to state the limitations of the scope also from a phonetic point of view. In particular, it should be stressed that timbre and dynamics are not issues. Burcas does not and is not meant to imitate the performance of a trained singer (not even that of an untrained one). As pointed out above, the system currently depends entirely on the MBROLA speech generator—in fact, the speech generator is (almost; the exception is a few command-line switches) consistently treated as a black box with three control parameters (allophone, duration, frequency). Any other data (for instance, any MIDI data not directly relating to pitch or duration) will be silently ignored. Section 2.1 mentions three important aspects in which singing synthesis is more difficult than synthesis of speech (duration, pitch range, and richness in musical expressions). The current, simplistic approach of Burcas delegates the first two to the speech generator and ignores the third entirely.

3.2 Building-blocks

The MIDI standard

As input for melody, Burcas takes a MIDI file, which may contain multiple parts. The Musical Instrument Digital Interface (MIDI) standard is an industry-standard protocol for controlling electronic music instruments.[v] It is performance-oriented and rather crude—for one thing, structural information is difficult to include—but it is simple and free and supported by practically all applications for music (notation software, synthesizers, sequencers, etc). MIDI also supports important musical expressions such as dynamics, vibrato, and portamento (although the interpretation of such parameters vary with instrument and application). Any practically oriented system which deals with music in editable form should at least accept MIDI as input.

MIDI is a binary format, not readable to humans; however, there are free and efficient MIDI-to-text converters available. The one used for Burcas is written by Günther Nagler (Nagler www)[vi]. A MIDI file containing the simple melody of figure 5 (section 4.6) may yield something like listing 1 when run through one of those.

Listing 1.

```
// durtest.mid
mthd
  version 1 // several tracks with separated channels to play
all at once
  // 2 tracks
  unit 1024 // is 1/4
end mthd

mtrk  // track 1
 /* 0ms */   tact 4 / 4 24 8
 /* 0ms */   key "1b maj"
 /* 0ms */   beats 96.00000 /* 625000 microsec/beat */
  12289; /* 7500ms */
end mtrk

mtrk(1)  // track 2
 /* 0ms */   trackname "Grand Piano"
 /* 0ms */   program GrandPno

        /* 0ms */      +c4    64;
  512;  /* 312ms */    -c4    0;
        /* 312ms */    +a3    64;
  512;  /* 625ms */    -a3    0;
        /* 625ms */    +a#3   64;
  512;  /* 937ms */    -a#3   0;
        /* 937ms */    +g3    64;
```

```
    512;   /* 1250ms*/    -g3   0;
           /* 1250ms */   +a3   64;

    (...)

  end mtrk
```

The excerpt shows (a text version of the) MIDI file 'durtest.mid', with whitespace added for clarity. The header (mthd) contains static data on the file: MIDI version (1), number of tracks (2), and the number of temporal units ('ticks') for a given note value (1024 to the crotchet). The first track (mtrk) usually holds data valid for entire sections (only one in this case), such as key ("1b maj", that is, F major), meter (4/4), tempo (96 beats/minute). The remaining tracks (only one in this case, mtrk(1)) contain instrument specifications (GrandPno) and data for individual notes. Thus, at timepoint 0, MIDI note c4 (which happens to correspond to middle C) is set to 'on' (indicated by the '+' sign) with a velocity of 64 (roughly corresponding to dynamics; 0-128 scale). At a timepoint 512 ticks after the last instruction (in this case, at 312 ms), the very same note is set to 'off' (indicated by the '-' sign) and a new note is set to on (+a3), etc.

Mbrola

The aim of the MBROLA project (MBROLA www, Dutoit et al 1996) is to boost academic research on speech synthesis by gathering diphone databases recorded on a voluntary basis for various languages, and providing them freely to the research community (for non-commercial and non-military use). Currently the project offers about 50 databases in some 25 languages. The databases must be used with the likewise named MBROLA speech synthesizer, which employs a version of the PSOLA algorithm (section 2.3) for time- and pitch-scaling.

As input (in addition to a specified database), the MBROLA speech generator takes as input a list of phonemes, with their respective durations and frequencies. A short extract from such a list is given in listing 2. The first column identifies the target allophone in SAMPA (SAMPA www), with `_' meaning "silence"; the second gives its duration. Then follows, optionally, one or more pair of values; in each of these, the first number specifies a timepoint (measured from the starting point of the segment, as a percentage of its entire duration) and the second gives the frequency value at that timepoint.

Listing 2.

```
_ 249
u:333 0 196 100 196
t 100
r 70
u:249 0 220 100 220
u:313 0 220 100 220
l 70
```

The limitation for time scaling depends on the frequency (typically 2-5 seconds for a female singing voice, although performance suffers long before that). For more on time-scaling in Burcas, see section 4.6.

The output is synthetic, concatenated speech in standard audio format (*.wav, *.au, *.aiff).

The MBROLA project has recently added support for non-uniform concatenative synthesis (Bozkurt et al 2001), more on which in section 6.

Ofelia

Ofelia is one of the above-mentioned diphone databases offered by the MBROLA project (in which it is known as ‘sw2’). It was recorded by Adina Svensson as part of her master's thesis (Svensson 2001).

Ofelia provides a female voice with a South Swedish dialect. In relation to standard Swedish, the most obvious dialectal hallmarks are uvular /r/ and no supradental allophones for /r/ followed by dental. The database is thus somewhat smaller than a corresponding one for standard Swedish (Ofelia is made up of 1658 diphones, whereas the older MBROLA sw1 database consists of 2059). For considerations on Ofelia specific to Burcas, see section 4.7.

4. Burcas: implementation

This section describes the current implementation of Burcas with emphasis on phonetic considerations, especially the LTS conversion and the model of segment durations.

4.1 Overview

As pointed out above, Burcas is little more than an alternative front-end to a simple concatenated text-to-speech synthesis system (TTS), a front-end that replaces intonation synthesis with a predefined melodical pattern and predicts segment durations for singing rather than for speech. In this way, many of the difficult tasks of the natural language processing (NLP) parts of a typical (spoken) TTS are bypassed. Figures 3 and 4 attempt to illustrate this.

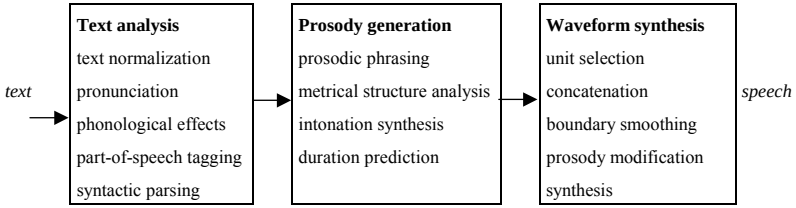

Figure 3. Block diagram of a concatenation-based TTS system (Macon fig 2.4).

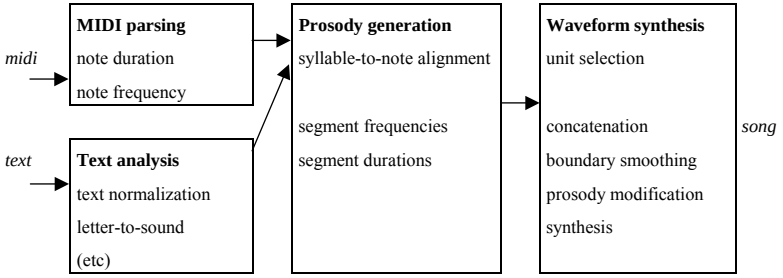

Figure 4. Block diagram of a simple concatenation-based MIDI-to-singing voice synthesis system.

The NLP part of the TTS system (the two left-most boxes of figure 3) has been simplified in the singing synthesis system; the text analysis is less sophisticated, and the prosody (if the term is taken to include suprasegmental phenomena in singing as well as in speech, such as aspects of timbre, pitch, length, or intensity) is largely extracted from melodical data. Not shown but conceivable is prosodic generation of musical expressions controlled by MIDI data.

Figure 4 also illustrates the basic working scheme of Burcas: from a MIDI file, note durations and frequencies are extracted and assigned to syllables of the LTS-processed text file in (syllabified) orthography. The syllables are then segmentized and frequencies and durations calculated for their component segments. These are used as control parameters for waveform synthesis, the inner workings of which are left completely to the MBROLA module. The synthesized waveform is output in standard audio format (*.wav, *.au, *aiff). The procedure is repeated for each part of the MIDI file (and, thus, for each part there should be corresponding words in the text file).

In the following, the choice of programming language is somewhat commented (4.2). Thereafter, the letter-to-sound conversion (4.3) is commented, as is the singing synthesis system's counterpart to prosody generation: syllable-to-note alignment (4.4), segment frequency assignment (4.5), and segment duration calculation (4.6). Finally, the database used is considered (4.7)..

4.2 Programming language

Burcas is currently written in Perl, in the function-oriented paradigm. This is fine just to quickly create a prototype (to leave time for other questions, in this case more phonetically oriented). However, for a large enough application, an object-oriented approach would clearly be preferable. Additionally, a graphical user interface would facilitate in producing more helpful error reports, especially for incorrect note-to-syllable assignment. In these respects, Perl is not optimal. While these questions may not be pressing for the purposes of the present paper, any predecessor to Burcas is likely to be written in some other language (probably Java).

4.3 From text to phoneme sequence

Letter-to-sound conversion for speech

The daunting task of transducing a given input text in the normal orthography of a language into a corresponding phoneme sequence is well-known from TTS systems, and, for most languages, nowhere near to be solved at a more general level. Roughly, the currently used strategies may be classified as either dictionary-based or rule-based. The former aims at storing a maximum of phonological knowledge into a lexicon, reserving rules only for entries not found. The latter rather summarizes as much as possible of the phonological information of dictionaries in a set of letter-to-sound (LTS; the term "grapheme-to-phoneme" is also widely used, although a more adequate term would be "grapheme-to-allophone") rules, using a small dictionary of exceptions only for words that cannot reasonably be described that way. The balance between lexicon and rules is set by application- and language-specific considerations (Carlson and Granström 1997).

Each orthographic system provides its own set of language-specific difficulties for LTS conversion. For Swedish, they include, but are by no means restricted to (hyphens, stress, word accent—´ for acute, ` for grave—added for disambiguation):

a) the absence of orthographical markings of morpheme boundaries in compounds: *bil-drulle/bild-rulle* ’road hog’/’film roll’, *häng-er/hän-ger* ’hang’/’devote’

b) the absence of orthographical markings of lexical stress: *pla´net/´planet* ‘planet’/’the plane’, *ba´nan/`banan* ‘banana’/’the path’

c) the absence of orthographical markings of lexical word accent: *´iden/`iden* ’the ide’/'the hibernating-dens’, *´gripen/`gripen* 'the gryphon'/'detained'

d) loan words fully or partly retaining their original spelling: *bourgogne, chianti, rave, aficionado, nachspiel*

e) various pronunciations (loan words aside), not easily described by rules, of some graphemes, most notably <o>: *kort* ’card’, *hov* ’hoof’, *hosta* ‘cough’ *harmonisk* 'harmonic' (all with [u:] or [u]); *kort* ‘short’, *hov* ‘court’, *kosta* ‘cost’, *elektronisk* 'electronic' (all with [o:] or [o])

f) occasional exceptions from the main orthographical markings of quantity (something like "vowel is long if followed by zero or one consonant graphemes morpheme-internally; otherwise short"): *vän* ‘friendly’, *lam* ‘lame’, *blåst* ‘wind’ (long); *vän* ‘friend’, *kam* ‘comb’ (short).

Of course, similar problems recur in LTS converters of many languages, as do those of correctly handling numbers, dates, abbreviations, special characters, etc. Even so, it should be noted that converting standard orthography to a phoneme sequence is rather straightforward compared to the delicate task of modelling phrasal intonation from unrestricted text. Such modelling usually presupposes full POS-tagging, often syntactic parsing, sometimes also semantic analysis including reference tracking and discourse modelling (all of which of course may help in resolving LTS problems as well).

Letter-to-sound conversion for singing

For a singing synthesis system, the task is somewhat different and generally easier. Most important, in singing (at least for Swedish), a lexical pronunciation (as different from pronunciation in phrasal context) of each word is acceptable—indeed, in most genres even expected. Phrase level prosody, involving prominence, grouping, stress clash modification, final lengthening etc, are set by melody and thus may be left to the composer (whose solution may be disputable or not, but in any case should be respected by the synthesis system). For Swedish, this means among other things a) that the model needs to deal with two stress levels only—stressed and unstressed syllable (the common model for speech on a phrasal level (Elert 1970, Bruce 1998) has four levels of prominence); and b) that the quantity distinction with co-occurring vowel quality difference will occur in stressed syllables only. For spoken Swedish, by contrast, compare Elert's minimal pair *(han) talar om (händelsen)*, '(he) speaks about/tells (the event)' where phrasal accent lies either on *-tal-* or *om*—that is, *talar* may or may not bear one of two highest stress levels of the four-level model; but in both cases it has the vowel quality marking long quantity.

There are other alleviations of the LTS burden in singing. Abbreviations, numbers, etc seldom occur in song texts; if they do, they can always be spelled out. The common hyphenation between syllables in fact may bypass the compound boundary marking problem (and largely does so for Swedish).

Furthermore (at least for the current application), the texts are usually short and the time needed for preparation of the lyrics little compared to the time invested in the music itself. A somewhat interactive way of arriving at an acceptable allophone sequence may therefore be acceptable, and consequently, it is of little importance whether or not the LTS converter produces a perfect phoneme transcription at the first attempt. In addition, occasional errors on a segmental level are probably less critical than in speech synthesis, as long as melody is retained.

There are two desiderata, though: any corrections should be easy to make; and any graphotactically acceptable input string, although containing unknown words, should as far as possible produce a valid phoneme sequence with the correct number of syllables.

Letter-to-sound conversion in Burcas

A useful overview of existing letter-to-sound and sound-to-letter relationships in Swedish is given in Olsson (1998). In short, the consistency of Swedish orthography falls somewhere half-way between extremes as Finnish, Turkish, or Italian (on one hand) and English, French, or Danish (on the other). Most graphophonematic (letter-to-sound, that is) relations may reasonably well be encoded in a tractable number of string-matching rules, resembling well-known reading rules, and a small dictionary of frequent exceptions. (The inverse problem, that of finding a spelling given a phoneme sequence, is much harder; an extreme case is the /ʃ/ phoneme for which Olsson lists 22 possible spellings[vii]). Given the specifications above, this is a suitable LTS converter of Burcas.

The main problem of the string-matching approach is that of correctly placing lexical stress—the LTS relationships differ in unstressed and stressed syllables. While a number of rules of thumb may be helpful, some of which are hinted at below, reliably identifying lexical stress is difficult without morphological analysis, which in turn may presuppose other analyses as well. For instance, *kanon, syntes, fasan, sedan, legat, predikat* (examples from Bruce 1998) are homographic, minimal stress pairs (or as close as one can get, given the complex acoustic correlates). An ambiguity such as *'kanon* ‘canon’ vs *ka'non* ’cannon’; ‘gun’ is challenging for any system, both candidates being uninflected nouns. Furthermore, for Burcas, the rules should make sense not only out of existing words, but also out of graphotactically legal but non-existing words—in such cases, the aim can be no more than an educated guess. The approach taken in Burcas currently

makes use of the following predefined data (see appendix 1 for the sets defined at the time of writing, by no means exhaustive):

* a set of LTS rules for stressed syllables (LTS+ below)
* a set of LTS rules for unstressed syllables (LTS- below)
* a set of inclinational suffixes (IS-; never stressed), including transcription (e.g. *-or, or-na, er-na-s, a-de*)
* a set of derivational affixes which are never stressed (DA-), including transcription (e.g. *-ande, -ing, -isk, -ning, be-*)
* a set of derivational affixes which are always stressed (DA+), including transcription (e.g. *-abel, -tet, -tion,-ssion, -sion, -nom*)
* a set of vowel graphemes likely to imply stress when in last syllable (V+) ({åiyuäö})
* a set of exceptions (E) (e.g. *och*)

Each of the pre-syllabified words (or each morpheme, if marked as a prosodic compound—see under “Text input” below) is then transcribed by string matching. The longest possible match is searched, starting with the entire syllable and gradually shortening the search window until one of the LTS-rules match. The choice between LTS+ and LTS- uses some expandable heuristics; at the time of writing, the ones used were the following (shown in pseudocode rather than a flow diagram; the C ternary <test> ? <if_true> : <if_false> operator indicates conditional flow control).

```
w already transcribed?
  output transcription, stop
: continue

w in E ?
  output transcription, stop
: continue

w-ending in IS-, leaving one or more sylls?
  save partial transcription,
  strip suffix, leaving wrest
  continue
: continue

wrest-suffix in DA- , leaving one or more sylls?
  save partial transcription,
  strip suffix, leaving wrest
  continue
: continue

wrest-prefix in DA- , leaving one or more sylls?
  save partial transcription,
```

```
  strip prefix, leaving wrest
  continue
: continue

wrest one syllable?
  transcribe with LTS+,
  sum transcriptions, output sum, stop
: continue

wrest-affix in DA+ ?
  save partial transcription,
  strip affix, leaving wrest
  transcribe remainder with LTS-,
  sum transcriptions, output sum, stop
: continue

last syllable of wrest containing any of V+ ?
  transcribe last syllable with LTS+,
  strip last syllable,
  transcribe remainder with LTS-,
  sum transcriptions, output sum, stop
: transcribe first syllable with LTS+,
  strip first syllable, leaving wrest,
  transcribe wrest with LTS-,
  sum transcriptions, output sum, stop
```

The method is admittedly simplistic (and might be somewhat elaborated in the future). Although a statistical evaluation falls outside the scope of this paper, it is clear that it will fail to correctly identify lexical stress in many cases (*prelat* 'prelate', *kanot* 'canoe', stress shift in *rektor/rektor-er* 'headmaster/s', as a few examples among many).

However, it should be noted that incorrectly placed stress need not be crucial. The three most important acoustic correlates of syllable stress in Swedish are F0, duration, and spectral energy distribution, perceptually corresponding to intonation, quantity, and vowel quality. Now, modelling intonation is not an issue in singing, and quantity is not included in the current duration model anyway (section 4.6). The most important cue of stress thus lies in spectral properties—those of short allophone instead of those of long, or vice versa. For the vowel pairs iː/ɪ, yː/ʏ, ɛː/ɛ, øː/ø, æː/æ, œː/œ, uː/ʊ the mix-up may not be critical in sung standard Swedish—the quality difference between the allophones is not very obvious compared to other phonetic peculiarities of singing (section 2.3). For the pairs oː/ɔ, eː/ɛ and (in particular) ɑː/a, ʉː/ɵ occasional corrections may be needed.

Rule formalism

The letter-to-sound rules are described in a simple formalism borrowed (and slightly modified) from the Festival system (Black et al 1999). Two rules are shown below. The rule sets used (except exceptions) are given in appendix 1.

```
< g > Vs = [ j ]
# < s t j > V  = [ S ]
```

This tells us that the grapheme <g> followed by a letter pertaining to class Vs ('soft' vowels, defined elsewhere) should be transcribed as [j]; and that the grapheme sequence <stj> after a word boundary and before a letter pertaining to class V (vowels, defined elsewhere) should be transcribed as [S].

Text input

The MIDI standard does permit lyrics (or arbitrary text) to be syllable-wise assigned to notes of a MIDI file. Exploiting this feature would eliminate the need of the error-prone external syllable-to-note assignment. However, not all MIDI applications support it at present, corrections of failures in LTS-conversion are not as easily handled, and anyway there are millions of wordless existing MIDI files which may be interesting to use as input without relying on MIDI processing software. A practical system for singing synthesis therefore should accept lyrics in a separate text file (and perhaps text included in MIDI files as well), including individual lyrics for each of the parts of the MIDI file.

In Burcas, lyrics are given as an ordinary text file, with (manually) inserted hyphens for separating syllables and the individual notes of melismatic vowels (as is standard practice for sheets of vocal music). Any amount of whitespace between syllables is accepted. The lyrics of one part is separated from those of another by a special character (`+'). For morpheme boundaries in compounds, `=' may be used (instead of `-') to indicate (prosodic) compounds, thus forcing assignment of one stressed syllable on each side.

To facilitate corrections, a mixture of normal orthography and phonetic symbols (SAMPA) within square brackets is allowed as input. Similarly, for phonetically naïve users, ad hoc spellings (known to reading researchers as 'invented spelling') within curly brackets are allowed; they will also usually be correctly handled by the rule set.

Listing 3 gives an example of a text file containing lyrics of two parts, both of which read *Vi drack bourgogne när vi ankom till Paris* 'We drank bourgogne when we arrived in Paris'. In the first part, a difficult word is spelled out in SAMPA and a prosodic compound segmented with '=' in the morpheme boundary. By contrast (for demonstrational purposes), in the second, the strategy of invented spelling is used for both words. (Note that modern Swedish for historical reasons spells /u/ with <o>—usually, see above—and /o/ with <å>, unlike most languages using the Latin alphabet.)

Listing 3.

```
vi drack [bu:r-gOnj] när vi an=[kOm] till Pa-ris
+
vi drack {bor-gånj} när vi {ann-kåmm} till Pa-ris
```

4.4 Syllable-to-note alignment

The melodical information (duration and pitch, separate for each part) is extracted from the MIDI file (converted to text, and using regular expressions), as a series of note-on/note-off instructions with associated timepoints. This is generally straightforward for MIDI-files originating in notation software, where (for a given part on a monophonic instrument) the note-off instruction for the first of two notes in sequence will occur at the same timepoint as the note-on instruction for the next.

In MIDI files recorded from actual performances, less rigid alignment is expected (or desired). Due to deviations (deliberate, normally) from a mechanical rendering of the score, one note may begin somewhat after the previous ends, or (for polyphonic instruments, such as the piano) somewhat before. When transferred to singing, a new note should terminate a previous, and very short pauses should be filled. Burcas filters away any note and any gap under a certain duration (by prolonging the first note, if necessary).

Each note of the melody thus extracted is assigned a syllable from the syllabified, phonetically transcribed text file, in the order and number they appear in the file. The method is admittedly a bit error-prone. Some graphical feedback would be welcome but is currently not implemented.

4.5 Segment frequencies

The approach taken for frequency assignment in Burcas is rather crude and may be replaced in some later version. The system assigns frequency values to vowels only, and lets the MBROLA speech generator interpolate values for voiced consonants. For very large intervals on long voiced clusters (e.g. *stängd bro* ‘closed bridge’), the resulting, rather slow, F0 movement may create a slight impression of glissando.

However, the F0 movement must not be too fast, either, if a smooth, naturally sounding transition from one note to the next is to be brought about. Common abstract representations of a musical piece—most typically a music score—consider notes as independent, discrete units, quantized in time and pitch; so do MIDI files extracted from notation software (and, although less obviously quantized in time, MIDI files recorded from actual performances). On some instruments, such as the piano, a rather close rendering of such a representation is possible (although perhaps not musically advisable)—notes at any interval may follow each other with no constraints on transition time. For singing, by contrast, there are physiological limitations, given among other things by the inertia of the articulators. For instance, no singer in the world may perform an instantaneous octave jump (and neither will TD-PSOLA without introducing artifacts).

As for Burcas, no problems arise when there are consonants in between the syllable nuclei of two notes in succession—the interpolated F0 movement is distributed among them. However, when two vowels meet, the first of them must allow for some pitch movement towards the end. To accomplish this, the last F0 value corresponding to a note of the melody is set not at the end of such a vowel, but some time before that (the value is passed as a configuration parameter; informal experiments have indicated 25 ms as reasonable).

4.6 Segment durations

Assumptions on segment duration in sung Swedish

The tentative segment duration model of Burcas is based on the following simplistic assumptions about sung Swedish. They are perhaps not equally applicable to all genres (the additional simplification "it is possible to use

one model for all singing styles" could arguably have been listed along with the others) but given the scope, they will constitute a starting point.

1) Reductions need not be bothered about; as pointed out, what may sound hyperarticulated in speech (and the output of many speech synthesis systems does) is rather the expected in singing.

2) In sung Swedish (or in any language) syllable durations are most often longer than in speech; over a certain syllable duration, consonants will have fixed durations and any changes will be reflected in the vowel only. These maximum durations for consonants will be referred to as 'standard durations' in the following.

3) It is possible to model segment duration in sung Swedish without considering the distinction between short and long consonant. This assumption (which greatly facilitates LTS conversion) is a generalization of the fact that syllables over a certain duration cannot hold the standard Swedish complementary V:C versus VC: distinction. (In contrast to many languages, quantity is not segmental in standard Swedish. Its domain is rather the syllable nucleus and the following coda: a long vowel occurs only immediately followed by a short consonant and vice versa. See Bruce 1998.) Thus, in a long syllable, of which there are plenty in singing, the distinction between short and long consonant may be entirely lost, leaving disambiguation to vowel quality or context (cf section 4.3).

   Although more data would be desirable, it seems that this may happen in more speech-like tempos as well. Compare the already mentioned lost quantity distinction in lament singing in Estonian (Ross and Lehiste 1994), for a language where quantity is indeed segmental and more important by far than in Swedish.

4) In sung Swedish, as in speech, there is some room for individual and occasional variation in segment durations; deviations within this span will not be phonologically contrasting but perceived as acceptable articulations (perhaps reflecting different degrees of, say, emotional commitment).

5) In sung Swedish, it is possible to disregard the influence of phonetic context. That is, consonant duration does not depend on prominence level, position (onset or coda), or possible cluster membership.

The last simplification (in particular) is admittedly very crude, based on practical reasoning in view of the large number of possible onsets and codas. A more ambitious approach to segment duration modelling, beyond the scope of Burcas, would indeed be dependent on context, allowing durations to vary according to phonetic surroundings (and perhaps to syllable duration in some more sophisticated way than the one described below). The value of such a method compared to the one chosen must eventually be evaluated by listener tests (and the gain balanced with the effort behind it).

Anchoring the syllable

The syllable is, in the words of Macon et al (1997), a natural 'quantal unit' of rhythm, in speech as well as in vocal music. In singing, each syllable of lyric is associated with a number of notes of the melody—one, in syllabic singing, or several, in melismatic. In the following, such a set of notes will be referred to as a ‘note group’; but one should keep in mind that such a group may have one member only.

The rhythmic information of a MIDI file is no more than a set of ”note-on/note-off”- instructions, with associated timepoints. For syllable-timepoint alignment, any given timepoint must be anchored to a specific location in the syllable. An appropriate anchor is the CV-border between onset and nucleus—perceptual experiments have shown that listeners and performers reliably place the beat of a syllable, its ‘perceptual center’, at that point, both for speech and for music (Macon et al 1997, reviewing research by Sundberg).

Calculating segment durations

The segment duration model of Burcas is a somewhat modified version of the one used in the Lyricos system (Macon et al 1997; see section 6.5).

The Lyricos model is rather simple. Lyricos is corpus-based; among other things, this means that often there will be no concatenation points in segments and that segments thus have an inherent duration, as recorded in the corpus. For a given syllable, the durations D of all extracted segments are summed, yielding $\sum D_{segm}$; so are those of the note/s assigned to that syllable, yielding $\sum D_{notes}$. A scaling factor $\rho = \sum D_{segm}/\sum D_{notes}$ is calculated. If $\rho > 1$,

then the steady state of the syllable nucleus is looped until $\rho \approx 1$; if $\rho < 1$, all durations are scaled uniformly.

For diphone synthesis, as in Burcas, it is less natural to think in terms of inherent durations. Furthermore, it seems that the uniform scaling may not reflect the fact that when singing in high tempi, some consonants are easier to articulate than others, and therefore sound more natural when compressed. In fast singing (or speech, for that matter), [na-na-na] or [la-la-la] is easier to produce than, say, [ça-ça-ça] or [fa-fa-fa]. The uniform scaling may also cause problems when a very short note is followed by a longer (i.e., non-compressed) starting with a complex consonant cluster—if the cluster is not compressed, it might not leave enough time for the syllable nucleus-coda associated to the previous, short note.

The version of the model employed in Burcas is thus slightly altered to accommodate fast tempi. A given syllable is associated to a note group, as described above. However, what is supposed to be articulated in the course of this note group is not onset-nucleus-coda of the given syllable ('ONC'), but rather nucleus-coda of that syllable and onset of the subsequent ('NCO'). For each note group, a compression rate ρ is therefore calculated on NCO rather than ONC:

$$\rho = \frac{\sum_{i=1}^{N_n} Li - \sum_{j=1}^{N_{ph}} D_{\min} j}{\sum_{k=1}^{N_{ph}} (D_{\text{std}} k - D_{\min} k)}$$

where N_{ph} is the number of phonemes in NCO, D_{min} and D_{std} their minimum and standard duration, respectively, and N_n the number of notes with duration L in the note group. The calculation uses tabulated values for standard and minimum duration (for consonants) and a generic minimum duration value (for vowels). These are measured from singing in moderate and fast tempi (see below). In this way, one allows for different compression rates of, say, nasals (whose durations ordinarily are 100 ms, but in fast singing may be 45-50 ms) versus plosives (which typically are 110-130 ms, but seldom shorter than 80-90 ms).

When a given note group is not preceded by a pause (as in legato singing), the model is used as is. When a note group is indeed preceded by a pause, the compression rate of the note group itself is calculated as before, and then the onset of the associated syllable is added with standard durations.

A negative value of ρ raises an error (interpreted as 'note duration less than sum of minimum durations of associated segments'). If $0 <= \rho <= 1$, the duration D of each segment is calculated as $D_{min} + \rho(D_{ord} - D_{min})$. Finally, if $\rho > 1$ (that is, if the syllable is sustained), all consonants get standard durations and the nucleus is prolonged. Diphthongs (a peripheral phenomenon in standard Swedish, occurring—arguably—in loan words as *aula* 'assembly hall', *paus* 'pause') may for purposes of segmental duration be defined as 'two vowels in phonetically transcribed syllable nucleus to one note in melody'; the calculated duration of the nucleus is simply divided equally between the vowels of the diphthong.

As for vowel prolongation, the TD-PSOLA algorithm implemented the MBROLA speech generator cannot handle extremely long segments, sometimes occuring in singing (the actual limits depend on pitch; see section 3.2). This deficiency must be worked around by concatenating several tokens of a long vowel without worrying too much about the resulting concatenation points. Burcas tries to make a virtue out of necessity and emulate a crude LFO, by letting the durations of the compounding vowel tokens vary quasi-periodically around 200-300 ms. Informally, this slightly randomized splitting was reported to yield a somewhat less static timbre than splitting into equal durations. It may be used (as a configuration option) also for vowel durations that MBROLA actually handles acceptably.

Standard and minimum segment durations in sung Swedish

The model described presupposes tabulated values for segment durations in sung Swedish: standard and minimum duration for consonants, and a generic minimum vowel duration. For speech at ordinary speech rate, duration tables are given for instance in Elert 1964, but, apparently, there are no corresponding investigations for singing. The relevance of consonant duration values obtained from speech for singing in slower tempo might not be immediately clear and called for at least some verification; additionally, for the model chosen, allophone-specific minimum durations for singing needed to be determined.

A tentative acoustic investigation was thus carried through. Phrase A of figure 5 were sung five times each for each consonant investigated and

recorded onto hard disk (as sung by the author, male speaker, standard Swedish dialect, experienced choir member but otherwise not a trained singer). The tempo was "moderate" (operationally defined as 72 to the crotchet, corresponding to 420 ms per syllable, kept steady by metronome). A carrier phrase was used (*satt i Xi:-na som jag sa* 'sat in Xi:na like I said', for onset, and *det var ni:-X-i som jag sa* 'it was ni-Xi that I said', for coda; X denotes the variable consonant), in order to imitate (one of) the phonetic contexts investigated by Elert in speech. Phrase B was recorded in the same way. It was sung twice and all occurrences but the first were measured. The tempo was as fast as possible, or close to it.

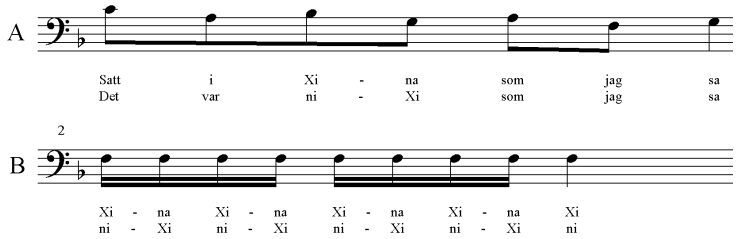

Figure 5.

The sound files were segmented in Praat (Praat www), the segmental durations were measured, and their mean was calculated. 'Standard duration' was simplistically defined as the mean of the duration of one-consonant onsets and one-consonant codas before and after /i:/, respectively, measured from singing in moderate tempo. 'Minimum duration' was defined in the same way from the recordings in fast tempo (but without the onset-coda division, which tends to dissolve in fast tempi).The results are given in table 1. The values given as reference are from Elert 1964 (table 6.3, consonants embedded in sentence context).

The generic minimum duration of vowels also sets the lowest possible note duration; being musically relevant, this value is best passed as a parameter (currently set in the configuration). The default is set to 60 ms, as the overall mean for fast singing in the measurements above.

Table 1.
Durations of one-consonant onsets and codas in singing. Means of five tokens.

Consonant X (SAMPA)	mod tempo onset (I'Xi:)	mod tempo coda ('i:XI)	Elert 1964 (C after /i:/)	mean (std dur)	fast tempo (min dur)
m	105	80	76	93	58
n	88	81	50	85	45
N	-	95	58	95	
f	149	135	146	142	100
v	64	73	52	66	48
j	90	84	-	87	71
s	113	111	111	112	90
S	134	136	113	135	108
C	135	130	-	133	103
h	90	-	-	90	-
l	85	70	68	78	45
r	50	52	50	51	38
ph	134	-	123	134	84
th	130	-	117	130	88
kh	144	-	124	144	94
b	84	78	88	81	65
d	80	71	72	76	59
g	83	70	78	77	57
(s)p*	106	120	-	113	86
(s)t*	94	122	-	108	80
(s)k*	91	118	-	105	81

* unaspirated stops in onsets were preceded by /s/ in the recordings

As can be seen in table 1, the duration values of singing in moderate tempo are slightly higher for most allophones than those reported by Elert for speech (notable exceptions are the voiced stops). The relevance of these differences may be perceptually negligible or not (cf assumption 4); however, the point here was rather to investigate whether it is reasonable at all to use consonantal duration data for speech also for singing in

considerably lower tempo. By and large, this seems indeed to be the case, although more data would be desirable.

4.7 Diphone database

The diphone database currently used (section 3.2) was produced from and primarily for spoken language. Although it has proved valuable as a testing tool, in particular for the duration model, it does have some drawbacks. Most obviously, the timbre is of course not very much like singing. More critical is that the [æ] and [œ] allophones of /ɛ/ and /ø/ have no transitions but /r/; they thus cannot be looped for sustained vowels.

There are, of course, workarounds for the allophone problem. The one chosen is the obvious: to replace extended tokens of [æ] and [œ] with [ɛ] and [ø] respectively (some varieties of Swedish in fact lack this distinction anyway). Nevertheless is the glitch described only the surface manifestation of a more basic, structural problem: the entire approach of using a spoken database is not entirely satisfying. A system aimed at singing synthesis should certainly employ a database produced from singing. Indeed, when first outlining the project, recording and construction of a sung diphone database especially tailored for Burcas was planned.

However, there are inherent limits for the system (section 2.3). The time-domain PSOLA algorithm for prosodic modification cannot really handle the very demanding DSP, including inevitable pitch scaling up to and exceeding one octave. The diphone approach inevitably requires a high density of concatenation points. Given these facts, and the considerable, tedious segmentation work (generally of little research interest) involved, the construction of a sung diphone database has been left for the future. Although the idea has not been entirely abandoned, there may be more efficient ways of spending the effort (cf section 6).

If a diphone database for sung Swedish is to be recorded, it should specifically include a glottal stop, or else something like a short /h/, between all vowels—in many genres, they are used between each note in melismas, especially in fast tempi, and they are likely to facilitate the difficult transition in such cases (cf section 4.5).

An ideal system for singing synthesis would allow an instant change of voice, by letting the user switch freely between prerecorded databases. This presupposes a modular design, keeping a uniform interface to the database, and all database-specific configurations separate from the rest. In particular,

any references to the actual diphone inventory are critical, to its set of allowed and disallowed phoneme sequences. As pointed out above, such "concatenative phonotactic restrictions" may differ somewhat between databases, mainly due to dialectal differences. A database interface that easily accommodates such restrictions is on Burcas' future wish list (where it certainly may remain indefinitely).

5. Burcas: current state

5.1 Performance

Given the restrictions of the TD-PSOLA algorithm of MBROLA and the diphone approach (2.3), the current implementation (3.1), and the database used (4.7), the following predictions may be made on the performance of the system:

* that it generally handles consonant transitions acceptably, but extended vowels much less so;

* that it does syllabic singing sounds (far) better than melismatic; or, more generally stated,

* that it does better on music with speech-like traits (i.e. rather small intervals, no very long durations and perhaps not so strong genre-related timbral requirements; compare for instance rap to opera).

The practical results with Burcas basically fulfil these expectations. Most consonant transitions sound fine, even those spanning rather large intervals. Extended vowels, however, range from mechanic but acceptable to hardly identifiable—especially front vowels seem to be difficult. The shortcomings of the simplistic duration model, which disregarded the quantity distinction between long and short consonant, is not apparent in the few samples analysed.

Informally, first-time listeners reported occasional difficulties in distinguishing lyrics, particularly in multi-part arrangements. This is certainly to be ascribed to Burcas to some extent (due to incomplete models, a generally unnatural voice quality, artifacts introduced by DSP etc). However, language perception in singing is normally far more difficult than in speech—for instance, few people perceive all the words of an unknown text when sung by a choir. A relevant comparison should be made with a human singer or vocal ensemble.

Adding parts otherwise does a lot for the perceived naturalness—perhaps just by drawing the listener's attention away from the static spectral characteristics. As listeners, we also have different expectations on group

and solo performances. Where a solo singer usually aims at a personal and powerful voice, a good choir singer acts more like an organ pipe, an anonymous part of an entirety. In fact, many trained singers use one voice type when performing as soloists and another, quite different, when performing in a group or a choir (Sundberg 1986). The uninteresting timbre of Burcas is more suited for the latter.

A disappointing discovery with added parts, though, was that the intonation of Burcas is not always perceived as immaculate. A partial explanation may be that the input frequencies to Burcas are well-tempered (like an idealized piano, with equal frequency ratio—$2^{1/12}$—between any two keys). By contrast, the main intonation strategy for an unaccompanied vocal ensemble is generally built on striving for pure intervals and minimization of beats, which may involve quite substantial deviations from well-tempered tuning. For instance, Sundberg 1986 have measured 24 cents for a skilled barbershop ensemble[viii].

However, well-temperedness versus beatlessness aside, it seems that the TD-PSOLA algorithm, which relies on precisely defining each glottal pulse in time, may miss the target slightly in large frequency manipulations. While this may pass unnoticed in some musical contexts, an ever so slight deviation may be very inconvenient in others. Of course, while manipulations of frequency require high precision in synthesis of singing, this is largely irrelevant in synthesis of speech.

A few samples of synthesized songs can be found on Burcas www; more examples are under construction. At the time of writing, the web page is rather outdated, but an update is planned for February 2003.

5.2 Future (re)directions

The purpose of the work described here was, to quote the introduction, “to construct a reasonably working concatenative-based singing synthesis system for Swedish”. Quite a few suboptimal choices had to be accepted, also outside using MBROLA and a database built from spoken rather than sung diphones. Thus, the segment duration model is built on rather little data, does not handle quantity, does not consider tempo under a certain limit, and does not care about phonetic context.

Furthermore, the LTS conversion is not entirely reliable. This may be addressed by expanding the rules, but—for the current uses of the system—

also by the simple “method of elimination”; that is, one could perhaps require lyrics to be phonetically transcribed instead. The loss is not too great; texts which mix SAMPA, invented spelling and standard orthography do not look too natural anyway. One may note that the far more ambitious Lyricos project (section 6.5) does not provide LTS conversion. Phonetically transcribed input would much facilitate the handling of quantity in the segment duration model.

Any part could be improved, of course. However, as pointed out, the entire technique has its limitations, and a possible successor is more likely to be constructed with the corpus-based approach (next section).

6. Beyond Burcas: corpus-based synthesis of singing

The approach to synthesis of singing taken in Burcas does have drawbacks. While some of them are inherent to the concatenative technique, most are associated with the current particular (and simplistic) implementation of it, in particular its use of a fixed and predefined inventory of diphones.

For concatenative synthesis of speech, an alternative approach has emerged in the latest years: concatenation of segments chosen at run-time ('unit selection') from large speech databases (as in the CHATR speech synthesis system, Black and Campbell 1995). This approach potentially offers a substantially lower density of concatenation points and a reduced need for signal processing, both of which yield higher quality synthesis. The field is quite vivid with several publications per year. For a good overview, see Möbius (2000).

This section considers a few issues when such corpus-based techniques are adapted to singing, and two existing systems are very briefly introduced. The corpus-based scheme is very little exploited (or at least very little accounted for in English); the tentative and incomplete nature of the account given here perhaps need not be stressed.

6.1 Corpus-based synthesis: overview

The key idea of concatenative synthesis, generally spoken, is that many coarticulatory effects and difficult-to-model transitions are included for free, as it were, in the acoustic inventory; that is, within the segments to be concatenated. The use of fixed inventories of concatenation units is straightforward and relatively cheap, computationally-wise. It does put limitations on performance, though; in particular:

a) the number of concatenation points will be high; for instance, for a diphone database, it will be equal to the number of segments in the synthesized utterance in speech, and much more than that in singing, given the need of concatenation in prolonged vowels;

b) the DSP, especially the pitch scaling, will be demanding (for speech) to very demanding (for singing), making high quality synthesis difficult to achieve;

c) the method is inherently rigid: in a diphone database, for instance, any effects of coarticulation not relating to the closest neighbouring phonemes will be ignored.

By using a larger database, several of these problems may be bypassed, taking the idea of including-for-free one step further. As for synthesis from a fixed inventory, a basic assumption is that a finite set of sounds is sufficient to approximate a given language. However, instead of having only one candidate for a given target allophone sequence, the database in corpus-based[ix] synthesis may contain hundreds or thousands of candidates. Each of the candidates has certain associated features, such as phonetic context, duration, F0 etc.

In most corpus-based schemes for speech synthesis, the run-time choice between the candidates is dictated by two things: how well those feature values ('feature vectors') correspond to those of the target, and how well the candidates concatenate. Particularly, sequences of target allophones which already occur in the database will concatenate without any distortion at all—in this way stretches substantially longer than the units of the previous concatenative techniques (often diphones or demisyllables) may be selected for concatenation.

The main advantages of the corpus-based technique has already been pointed out: the potentially lower density of concatenation points and the reduced need for signal processing, both of which yield higher quality synthesis.

On the other hand, large corpora are needed (for speech, anything between 30 and 120 minutes or more has been tried) and the computation involved in selecting the most appropriate candidate is expensive (much more so than for fixed unit inventories; it is, in fact, only quite recently that the continuing development of computer processing capability and memory has made the method a viable alternative). In the following, the corpus and the selection of concatenation units from it are touched upon, with applications of singing in mind.

6.2 Corpus

Corpus design for speech synthesis is not an easy matter. Large corpora are difficult to annotate consistently and to maintain (this is true whether it is

done automatically, semi-automatically or manually). Annoyingly enough, a large part of the units of an annotated speech database are never or very seldom selected for synthesis. Removing such redundant segments would reduce the size of the database, which makes maintenance and quality control easier. However, finding out which units to remove without first making the actual annotation is difficult (although, for speech, the process may be automated). Another way to produce a slimmer database is to statistically analyse the segmental and prosodic coverage directly from candidate texts to be recorded. A subset of the texts with the same coverage may then be extracted with a greedy algorithm (e.g. Black & Lenzo 2001). This method can substantially reduce the material that actually needs recording.

Devising a training corpus for a singing synthesis system is certainly no easier. If constructed from scratch, it will probably be even larger than a corpus for speech synthesis of comparable quality. Above all, it will be differently balanced. The number of allophones can conceivably be reduced (cf section 2.1); in Swedish, for instance, in some cases perhaps short and long vowel allophones may both serve as candidates for a given vowel (see short discussion in section 4.3). On the other hand, there must probably be a higher number of tokens of each allophone (or, possibly, of each segment), offering more candidates for the wider range of pitch and duration to minimize frequency and time modifications by DSP. Furthermore, the corpus must include candidates for those categories of musical expression which cannot be acceptably modelled with pure DSP.

For the latter categories, some musically relevant tagging system must be devised. The actual annotation for those categories may have to be made manually. Such annotations—in particular dynamics—generally refer to phrase level rather than segment level, which makes the task less overwhelming.

However, also the annotation at segment level is rather different in a sung corpus, compared to that of a spoken. There are quite reliable aligners (that is, systems that automatically time align speech signals with corresponding text) for speech, useful for corpus annotation (see for instance Sjölander 2001 for Swedish); however, they must be modified to handle the prosodic peculiarities in singing. Extracting information from musical scores greatly facilitates the task (Loscos et al 1999, Meron 1999).

An interesting and speculative alternative to recording a corpus from scratch is to use an existing one, such as the collected recordings of artists

perhaps no longer alive, for instance from record companies' archives. Given a large enough corpus and good enough methods, new songs could conceivably be synthesized.

6.3 Unit selection

For the computation involved in unit selection in speech synthesis, most existing systems employ two central components: a distance measure, and a selection algorithm. The distance measure is an evaluation (and possibly a quantification) of the distance—in one sense or another—between a target sequence and a candidate sequence. Various different distance measures have been tried, most of them based on cepstral coefficients. Unfortunately none of these acoustic measures correlates reliably with human perception—the correlation is moderate at best (Wouters and Macon 1998). Human perception is very sensitive in some respects, but rather insusceptible in others (for instance, in a complex sound, phase of harmonics is of little perceptual relevance). Finding an easily calculated distance measure which takes that into account is not an easy task.

The task of the selection algorithm is to make best possible use of available data by choosing what segments to concatenate, out of possibly very many candidates. Given a certain database, the segments selected from it should be as close to the target sequence as possible according to the distance measure used, with as few concatenation points as possible. Additionally, the algorithm should be able to generalize its selection criteria to unseen cases, picking the best compromise choice if an exact match cannot be found. Most currently used selection algorithms are computationally complex and expensive. They are usually based on finite state networks, on so called "context-oriented clustering" (building on classification-and-regression trees, Breiman et al 1984), or on blends of those.

A rather different approach is tried in British Telecom's Laureate TTS system (Breen and Jackson 1998). This phoneme-based approach exploits more than its competitors the fact that speech is a structured phenomenon: most acoustic variability between realizations of a given phoneme is in fact predictable from linguistic context. The variation should be possible to model in purely phonological terms, given sufficiently good models (yet to be completed in some respects, as the authors admit).

The Laureate system thus completely ignores the computationally expensive acoustic distance measurements. Its counterpart of acoustic feature vectors is a phonologically motivated set of mostly binary abstract features (all of which should be computable from text). Typical such features are 'phoneme resides in syllable nucleus', 'vowel is articulated with lip rounding' etc. Candidate selection is based on phonetic context; selection between candidates is made for cheapest path through the candidates, with the various discriminating features assigned relative importance by a tree structure.

For singing, the general problem is of course similar: how to quantify the distance between candidate and target, and how to find the optimal combination of units? Given a corpus, there must be a distance measure which in some way takes into account costs of pitch and frequency modifications (just like for speech synthesis, but generally higher) and of the added musical categories; and a selection algorithm which makes the task of choosing units for concatenation tractable. The selection algorithm may be based on existing solutions; however, devising some kind of phonological modelling which also caters for musical categories is an interesting thought.

6.4 DSP

When trying to expand the expressional potential of a corpus-based synthesis system, the basic choices are either to record a larger database (and annotate it appropriately), or to resort to DSP—to find a new or improved way of modifying the concatenated waveform. The balance between expanding the corpus and covering the new categories by DSP is a tradeoff. On the corpus side, there is potentially high performance, even and in particular for phenomena not yet satisfyingly modelled, at the cost of inherent rigidity, expensive construction and difficult maintenance. On the DSP side, there is potential flexibility and generality at the cost of computational complexity and the work involved in finding reliable acoustic and perceptual models.

In either case, some DSP is indispensable. Even with a very large corpus and optimal unit selection, and even for genres where timbre does not play a decisive role (for instance, for two extremes, compare rap to opera), some pitch- and time-scaling is inevitable. Ideally, this process should not introduce any artifacts, nor any perceptible degradation of voice quality.

In singing, the quality of the sounding output is even more important than in speech—in fact, the output's being musically pleasing is often more important than its being intelligible—and the desired palette of expressions is larger by far. This is a strong argument for searching DSP solutions wherever possible—the corpus may grow unwieldy anyway. For some phenomena (such as natural pitch fluctuations, portamento, and vibrato), there are good or at least acceptable DSP algorithms.

For others, such as dynamics and other timbral adjustments, the acoustic correlates are more numerous and sometimes more elusive; the modelling is therefore more difficult. For instance, singing more softly (i.e. with decreased vocal effort) is associated with an increase in the downward spectral tilt, which is rather easily modelled. However, the breathy voice quality typically associated with soft singing (in some genres) is less straightforward to implement (Macon 1996), and it may be more convenient to include in the corpus a musical category to cover at least some variation in dynamics (e.g. annotated "forte" or "piano").

6.5 Existing systems

As pointed out, the corpus-based approach to singing synthesis is very little exploited. To date, there seems to be only two publications in English on serious attempts: Lyricos under the late Michael Macon (Macon 1996, PhD dissertation; related is Macon et al 1997), and the system accounted for by Yoram Meron (Meron 1999, PhD dissertation; related are Meron & Hirose 1998, 1999, 2000). Macon's thesis (in electrical engineering) deals in first place with sinusoidal modelling from a practical and theoretical point of view; naturally, it concentrates on rather low-level issues. Meron's thesis (in electronic and information engineering) takes a slightly more general perspective on singing, but the focus still lies on engineering (rather than say, phonological or musicological) issues. The very short descriptions below do not make justice to the systems, but at least give an idea about the methods used.

Lyricos

Lyricos (Macon 1996, Macon et al 1997, Lyricos www) was developed at Georgia Institute of Technology. Its database consists of specially designed nonsense words, recorded from a professional singer and annotated manually. To minimize pitch-scaling, each unit has been recorded at several

pitches. The concatenation units are sinusoidal waveform model parameters, and run-time unit selection is performed from a phonologically motivated, partially precalculated binary decision tree. The output candidates of the tree are ranked according to their closeness to desired pitch, using the ABS/OLA algorithm for further adjustments.

Lyricos takes MIDI and phonetically-spelled lyrics as input. The ABS/OLA sinusoidal model permits convenient modification in the frequency domain of spectral properties, such as tilt; thereby, MIDI-controllable parameters such as vibrato and velocity (vocal effort) may be interpreted by DSP manipulation. Rough vocal tract scaling is also fairly easily performed, e.g., in order to change a baritone to a bass.

Lyricos is closed, but it has a descendant, called Flinger, at Center for Spoken Language Understanding, Oregon Graduate Institute. Flinger is basically a customized version of the Festival text-to-speech system (Black et al 1999, Festival www), developed at Centre for Speech Technology Research, University of Edinburgh. However, the project does not seem to be active any longer and has not yielded any publications outside the web page (Flinger www).

Meron

A similar but contrasting approach is taken by Yoram Meron at the university of Tokyo (Meron 1999). Meron's system is designed to be practical and is especially strong on training. The original database was recorded with a vocalist (a Japanese student of singing) singing along to MIDI-created playback (on authentic music, Schubert's Winterreise) and annotated automatically (using musical information). However, the system permits the automatized learning of new voices from existing recordings and provides useful tools to that end: for aligning an audio recording of a singer to a MIDI file, and for dividing a mixed recording into separate singer and piano accompaniment audio files given the musical score (neither of which, however, yet handle realistic recording conditions flawlessly).

The unit selection employs an acoustic distance measure supplemented with a few musical parameters and a Viterbi search for cheapest path through the network of candidates. For prosodic manipulation, a hybrid synthesis scheme called SM-PSOLA is introduced, with sinusoidal modelling of the periodic parts of the signal, aperiodic parts stored as residual waveforms.

The representation permits spectral manipulations much as for Lyricos. Important singing characteristics are interpreted, such as vibrato (by DSP) and singer's formant (in unit selection).

6.6 Conclusion

High quality singing synthesis based on unit selection from corpora is little explored, and many problems are certainly unsolved. However, the results exhibited by systems like Lyricos and the one of Meron should be encouraging.

The continuous development of new parametric signal models for the databases (such as the sinusoidal model or the harmonic-plus-noise model) and new associated models and algorithms for important general or singing-specific phenomena (such as natural pitch fluctuation, vibrato, portamento, or—to some degree—vocal effort) challenges the thought of concatenative synthesis as purely reassembling predefined, pitch-and-time-scaled building-blocks. Still, when modelling is difficult or incomplete (such as for singer's formant, or for different phonation types), the responsibility is passed to the corpus.

Among the challenges for the scheme, DSP aside, are the design and annotation of the corpus. Ideally, the entire learning process should be automatized and training possible from non-specialized recordings and music scores (as is attempted at in Meron's system). Furthermore, the tagging should be devised with singing in mind, and the actual unit selection will have to consider partly other categories than in speech. Acoustic distance measures are unreliable for speech and probably even more so for singing; a phonological approach corresponding to that of BT's Laureate II TTS system seems to be an interesting alternative.

Acknowledgements

The author wishes to thank Günther Nagler for helpful MIDI-to-text converters, Johan Frid for useful comments in general, and Joost van de Weijer for supervision.

Appendix 1

```
;==========================
; LTS-RULES,
; STRESSED SYLL
;==========================

; classes

class: Vs = e i y ä ö
class: Vh = a o u å
class: V = a o u å e i y ä ö
class: C = b c d f g h j k l
m n p q r s t v w x z


; context rules

< a > r r = [ a ]
< a > r C = [ A: ]
< a > C C = [ a ]
< a > C = [ A: ]
< a > = [ a ]

< b b > = [ b ]
< b > = [ b ]

< c > Vh = [ k ]
< c > Vs = [ s ]
< c h > V = [ C ]
< c k >  = [ k ]
< c c >  = [ k s ]
< c > = [  ]

# < d j > V = [ j ]
< d d > = [ d ]
< d > = [ d ]

< e > C C = [ e ]
< e > C = [ e: ]
< e >  = [ e ]

< é > = [ e: ]

< f f > = [ f ]
< f > = [ f ]

< g j > V = [ j ]
< g n >  = [ N n ]
< g g> = [ g ]
# < g > Vs = [ j ]
< g > = [ g ]

# < h j > V = [ j ]
< h > = [ h ]

< i > C C = [ I ]
< i > C = [ i: ]
< i > = [ i: ]

< j > = [ j ]

< k j >  = [ C ]
# < k > Vs = [ C ]
< k > = [ k ]


< l g > = [ l j ]
< l j > V = [ j ]
< l l > = [ l ]
< l > = [ l ]

< m m > = [ m ]
< m > = [ m ]

< n k >  = [ N k ]
< n g >  = [ N ]
< n n> = [ n ]
< n > = [ n ]

< o > C C = [ U ]
< o > C = [ u: ]
< o > = [ u: ]


< p p> = [ p ]
< p > = [ p ]

< q > = [ q ]

< r g > = [ r j ]
< r r> = [ r ]
< r > = [ r ]

< s t j > V  = [ S ]
< s k j > V = [ S ]
< s c h > V = [ S ]
< s h > V = [ S ]
< s k  > Vs = [ S ]
< s j > V = [ S ]
< s s> = [ s ]
< s > = [ s ]

< t j >  = [ C ]
< t t> = [ t ]
< t > = [ t ]

< u > C C = [ u0 ]
< u > C  = [ }: ]
< u >  = [ }: ]


< v v> = [ v ]
< v > = [ v ]

< w > = [ v ]

< x > = [ k s ]

< y > C C = [ Y ]
< y > C = [ y: ]
< y > = [ y: ]


< z > = [ s ]

< å > C C = [ O ]
< å > C  = [ o: ]
< å > = [ o: ]

< ä > r C = [ { ]
< ä > C C = [ e ]
< ä > r = [ {: ]
< ä > C = [ E: ]
< ä > = [ E: ]

< ö > C C = [ 2 ]
< ö > r C = [ 9 ]
< ö > r  = [ 9: ]
< ö > C  = [ 2: ]
< ö > = [ 2: ]



; ==========================
; LTS-RULES,
; NON-STRESSED SYLL
;==========================

; classes

class: Vs = e i y ä ö
class: Vh = a o u å
class: V = a o u å e i y ä ö
class: C = b c d f g h j k l
m n p q r s t v w x z


; context rules

< a > = [ a ]

< b b > = [ b ]
< b > = [ b ]

< c > Vh = [ k ]
< c > Vs = [ s ]
< c h > V  = [ C ]
< c k >  = [ k ]
< c c >  = [ k s ]
< c C > = [ k ]
< c > = [ k ]
# < d j >  = [ j ]
< d d > = [ d ]
< d > = [ d ]

< e >  = [ e ]

< f f > = [ f ]
< f > = [ f ]

# < g j >  = [ j ]
# < g > Vs = [ j ]
< g n >  = [ N n ]
< g g> = [ g ]
< g > = [ g ]

# < h j > V  = [ j ]
< h > = [ h ]

< i > = [ I ]

< j > = [ j ]

< k j >  = [ C ]
# < k > Vs = [ C ]
< k > = [ k ]

# < l j >  = [ j ]
< l g > e = [ l j ]
< l g > # = [ l j ]
< l l > = [ l ]
< l > = [ l ]

< m m > = [ m ]
< m > = [ m ]

< n k >  = [ N k ]
< n g >  = [ N ]
< n n> = [ n ]
< n > = [ n ]

< o > C C = [ U ]
< o >  = [ o: ]

< p p> = [ p ]
< p > = [ p ]

< q > = [ q ]

< r g > e = [ r j ]
< r g > # = [ r j ]
< r r> = [ r ]
< r > = [ r ]

< s t j >  = [ S ]
< s k j >  = [ S ]
< s c h >  = [ S ]
< s h >  = [ S ]
< s k  > Vs = [ S ]
```

```
< s j >  = [ S ]
< s s j > = [ S ]
< s s> = [ s ]
< s > = [ s ]

< t j >  = [ C ]
< t t> = [ t ]
< t > = [ t ]

< u > C C = [ u0 ]
< u >  = [ u0 ]


< v v> = [ v ]
< v > = [ v ]

< w > = [ v ]

< x > = [ k s ]

< y > C C = [ Y ]

< z > = [ s ]

< å > C C = [ O ]
< å >  = [ O ]

< ä > r C = [ { ]
< ä > C C = [ e ]
< ä >  = [ e ]

< ö > C C = [ 2 ]
< ö > r C = [ 9 ]
< ö > = [ 2 ]


;==========================
; UNSTRESSED SUFFIXES
; (INFL & DERIV)
;==========================

en = @n
el = @l
er = @r
ens = @ns
els = @ls
ers = @rs
or =  Ur
ar =  ar
ors = Urs
ars = ars
orna = Ur-na
ornas = Ur-nas
arna =  ar-na
arnas = ar-nas
erna = @r-na
ernas = @r-nas
ade = a-d@
at =  at
ades = a-d@s
ats =  ats
ad = ad
ande = an-d@
ende = @n-d@
re = r@
are = a-r@
ast = ast
aste =ast-@
ig = Ig
igt = Ikt
ige = Ig-e
iga = Ig-a
lig = lIg
ligt = lIkt
lige = lIg-e
liga = lIg-a
isk = Isk
iskt = Iskt
iske = Isk-e
iska = Isk-a
ning = nIN
ningen = nIN-en
ningar = nIN-ar
ningarna = nIN-ar-na
nings = nINs
ningens = nIN-ens
ningars = nIN-ars
ningarnas = nIN-ar-nas
ing = IN
ingen = IN-en
ingar = IN-ar
ingarna = IN-ar-na
ings = INs
ingens = IN-ens
ingars = IN-ars
ingarnas = IN-ar-nas
ium = I-u0m

;==========================
; STRESSED DERIV SUFFIXES
;==========================

fon = fo:n
for = fo:r
ob = o:b
sof = so:f
log = lo:g
gog = go:g
nom = no:m
skop = sko:p
os = o:s
ot = o:t
od = U:d
tion = SU:n
sion = SU:n
```



```
ssion = SU:n
abel = A:bel
ist = Ist
ism = Ism
tet = te:t
ment = maN
mang = maN
ang = aN
ens = ens
ent = ent
er = e:r
ant = ant
eri = er-i:
ess = es
tris = tri:s
ur = }:r
ör = 9:r
ös = 2:s
ett = et
inna = In-a
issa = Is-a
i = i:
ell = el
al = al
erium = e:r-I-u0m
arium = A:r-I-u0m
ette = et
ett = et
il = i:l
al = A:l
gram = gram
age = A:S
oge = o:S
ige = i:S



;==========================
;UNSTRESSED DERIV PREFIXES
;==========================

be = be
för = f9r
```

References

(URL:s given valid as for December 2002)

Literature

Berndtsson, G & J. Sundberg. 1993. The MUSSE DIG Singing Synthesis, KTH, Stockholm (Baritone and Bass). In: SMAC93, Proceedings of the SMAC93 (Stockholm Music Acoustics Conference 1993), pp. 279-281. Stockholm: Royal Swedish Academy of Music, 1994.

Black. A., P. Taylor & R. Caley. 1999. *The Festival Synthesis System. System Documentation.* v 1.4. Available at: http://www.cstr.ed.ac.uk/projects/festival/manual/festival_toc.html

Black A. & N. Campbell. 1995. Optimising selection of units from speech databases for concatenative synthesis. In *Proceedings of the European Conference on Speech Communication and Technology* (Madrid, Spain), vol 1, 581-584.

Black A. & K. Lenzo. 2001. Optimal data selection for unit selection synthesis. In *Proceedings of the 4th ISCA Tutorial and Research Workshop on Speech Synthesis* (Blair Atholl, UK).

Bozkurt, B., M. Bagein, T. Dutoit. 2001. From MBROLA to NU-MBROLA. *Proc. 4th ISCA Tutorial and Research Workshop on Speech Synthesis*, pp. 127-130, Blair Atholl, Scotland.

Breen A. & P. Jackson. 1998. Non-uniform unit selection and the similarity metric within BT's laureate TTS system. In *Proceedings of the Third ESCA Workshop on Speech Synthesis* (Jenolan Caves, Australia), 373 - 376.

Breiman L., J. Friedman, R. Olshen, C. Stone. 1984. *Classification and Regression Trees.* Wadsworth & Brooks, Monterrey, CA, 1984.

Bruce, G. 1998. *Allmän och svensk prosodi*. Praktisk Lingvistik 16/1998. Institutionen för lingvistik, University of Lund.

Carlson R. & B. Granström. 1997. Speech Synthesis. In: W Hardcastle & J Laver (Eds.), *The Handbook of Phonetic Sciences*, Blackwell Publishers Ltd., Oxford, pp. 768-788

Chan, M. 1987. Tone and melody in Cantonese. In: *Berkeley Linguistic Society, Proceedings of the Thirteenth Annual Meeting*, pp. 26-37. Available at: http://deall.ohio-state.edu/chan.9/articles/bls13.htm

Cook, P. 1998. Toward the Perfect Audio Morph? Singing Voice Synthesis and Processing. First European COST Conference on Digital Audio Effects, Barcelona.

Dutoit, T. 1997. *An introduction to text-to-speech syntesis.* Dordrecht, Kluwer.

Dutoit, T. V. Pagel, N Pierret, F. Bataille & O. van der Wrecken. 1996. The MBROLA project: towards a set of high quality speech synthesizers free of use for non-commercial purposes. *Proceedings ICSLP '96* (Philadelphia) 3, 1393-96.

Elert, C-C. 1970. *Ljud och ord i svenskan.* Almqvist & Wiksell, Stockholm.

Elert, C-C. 1964. *Phonologic studies of quantity in Swedish*. Almqvist & Wiksell, Stockholm.

Klatt, D. H. 1987. Review of text-to-speech conversion for English. *Journal of the Acoustical Society of America*, 82 (3), September 1987.

Lemmetty, S. 1999. Review of speech synthesis technology. Master's thesis, Dept. of Electrical and Communications Engineering, Helsinki University of Technology. Availablet at: http://www.acoustics.hut.fi/~slemmett/dippa/

Loscos, A., P. Cano, J. Bonada. 1999. Low-delay singing voice alignment to text. *Proceedings of the ICMC99*.

Macon, M. 1996. *Speech Synthesis Based on Sinusoidal Modeling*. PhD thesis, Electrical Engineering, Georgia Institute of Technology.

Macon M., L. Jensen-Link , J. Oliverio, M. Clements & E. B. George. 1997. Concatenation-based MIDI-to-singing voice synthesis. In: *103rd Meeting of the Audio Engineering Society*, New York.

Meron Y. 1999. *High Quality Singing Synthesis Using the Selection-based Synthesis Scheme*. PhD thesis, University of Tokyo.

Meron Y & K. Hirose. 1998. Separation of singing and piano sounds. *Proceedings of the International Conference on Spoken Language Processing*, Sydney, Vol.3, pp. 1059-1062.

Meron Y & K. Hirose. 2000. Synthesis of vibrato singing. *Proceedings of the IEEE International Conference on Acoustics Speech & Signal Processing,* Istanbul Vol.2, pp.745-748.

Meron Y & K. Hirose. 1999. Efficient weight training for selection based synthesis. *Proceedings of the 6th European Conference on Speech Communication and Technology,* Budapest, Vol.5, pp.2319-2322.

Möbius B. 2000. Corpus-based speech synthesis: methods and challenges. *Arbeitspapiere des Instituts für Maschinelle Sprachverarbeitung* (Univ. Stuttgart), AIMS 6 (4), 87-116.

Olsson, L-J. 1998. Specification of phonemic repesentation, Swedish. DEL 4.1.3 of the EC project "SCARRIE Scandinavian proof-reading tools" (LE3-4239). Available at: http://www.ling.uu.se/wp/wp3b.pdf

Ross J. & I. Lehiste (1994). Lost prosodic oppositions: A study of contrastive duration in Estonian funeral laments. *Language and Speech* 37, 407-424. Available at: http://www.ling.ed.ac.uk/~lgsp/ross.html

Sagisaka Y. 1988. Speech synthesis by rule using an optimal selection of non-uniform synthesis units. In *Proc. of the Int'l Conf. on Acoustics, Speech, and Signal Processing*, page 679.

Selfridge-Field Eleanor. 1997. *Beyond MIDI: The Handbook of Musical Codes*. MIT Press.

Sjölander, Kåre. 2001. Automatic alignment of phonetic segments. *Working Papers* 49 (Fonetik 2001), 140–143. Lund University, Dept. of Linguistics.

Stylianou, Y. 2001. Applying the Harmonic Plus Noise Model in Concatenative Speech Synthesis. *IEEE transactions on speech and audio processing*, vol. 9:1, Jan 2001.

Sundberg, Johan. 1986. *Röstlära : fakta om rösten i tal och sång*. 2 ed. English translation: *The science of the singing voice*. Northern Illinois University Press, 1987.

Sundberg, Johan. 1989. Musikens ljudlära. Proprius, Stockholm. English translation: *The Science of Musical Sounds*, Academic Press, 1991.

Svensson, Adina. 2001. *Ofelia—en ny syntesröst. En studie om talsyntes i allmänhet och konkateneringssyntes i synnerhet*. D-uppsats i datalingvistik, inst f lingvistik, Lunds universitet. Available at: http://www.ling.lu.se/education/essays/AdinaSvensson_D.pdf

Uneson, M. 2002. Outlines of Burcas–a simple concatenation-based MIDI-to-singing voice synthesis system. *TMH-QPSR Vol. 43 – Fonetik 2002*

Wouters Johan & Macon Michael. 1998. A perceptual evaluation of distance measures for concatenative speech synthesis. *Proc. of International Conference on Spoken Language Processing*, November 1998.

WWW

Burcas www:
http://www.ling.lu.se/persons/Marcusu/music/ burcas/index.html

Cook www (many publications related to SPASM):
http://www.cs.princeton.edu/~prc/SingingSynth.html

Festival www:
http://www.cstr.ed.ac.uk/projects/festival/

Flinger www:
http://cslu.cse.ogi.edu/tts/flinger/

Hansper www (MIDI tutorial by George Hansper)
http://crystal.apana.org.au/ghansper/midi_introduction/

Lyricos www:
http://cslu.cse.ogi.edu/tts/research/sing/sing.html

MBROLA www:
http://tcts.fpms.ac.be/synthesis/mbrola.html

MIDI www (home page of MIDI manufacturers association):
http://www.midi.org/

Nagler www (home page of Günther Nagler):
http://www2.iicm.edu/Cpub

Praat www
http://www.fon.hum.uva.nl/praat/

Rodet www (many publications related to CHANT)
http://www.ircam.fr/equipes/analyse-synthese/rodet/

SAMPA www:
http://www.phon.ucl.ac.uk/home/sampa/home.htm

Notes

[i] The name might possibly be interpreted as an acronym for "Basic Universal Resynthesis- and Concatenation-based Artificial Singer". Other suggestions are welcome.

Supplementary note: contrary to the statement above, the anonymous suggestion “But Units R Complicated And Suck” was not welcome; if identified, its author will not receive any Christmas card from me next year.

[ii] It should be noted that the term 'dynamics' here is used in its musical sense, slightly different from its common use in, for instance, phonetics (that is, as opposed to 'static'). Vocal dynamics in the musical sense is related to loudness and vocal effort, not to articulatory movements. Most musically dynamic events in singing evolve on vowels; if spoken, they would generally have been considered steady-states by a phonetician.

[iii] For standard Swedish, one case in point is the supradentalization of /r/ + dental (as in hård, törne, pärla, fors, mört ’hard’, ’thorn’, ’pearl’, ’rapids’, ’roach’). In speech, supradentalization is compulsory if hyperarticulation is to be avoided; in singing, it is very rare.

[iv] An informal discussion on Linguist list summarized by Susan Fischer (8.567, Apr 22, 1997; available at http://www.linguistlist.org/issues/8/8-567.html) bears the heading "Whispering and singing in tone languages". It concerns in particular Chinese. Consensus is far away and the scholarly worth may be little, but here are Fischer's summaries (italicized) and some contributions from different participants (quoted):

> **In some classical Chinese songs as well as in Vietnamese songs, you have to have a match between tone changes in the lyrics and pitch changes in the music.*
> "I have had a reasonable exposure to the Cantonese pop music from Hong Kong and it's amazing how able they are to get the lexical tones to fit the desired melody. However, if there is ever a discrepancy, in my experience it seems that context disambiguates and secondary devices such as you mentioned are not implemented to do the job."
> "With singing it seems to me that if we are talking about singing traditional Chinese poetry, the issue would not arise because Chinese metre is based on tones so that the music and the words would not be fighting each other, as it were. "
>
> **In the case of singing, sometimes there is no way to disambiguate. However, the load carried by tone may not be that great anyway.*
> "For singing, words in Chinese songs are notoriously hard to understand, and people don't mind that much—they get the printed words when they want to sing karaoke."
>
> **In some cases, only context provides disambiguation.*
> "... the question of ambiguity seems to be part and parcel of tone languages not only in

such instances as whispering and singing, but also in cases where there is much noise in the background... "

[v] The complete MIDI 1.0 specification may be ordered from MIDI www. As for web tutorials and printed handbooks, there is a plethora. I have found Hansper www and Selfridge-Field 1997 useful.

[vi] In fact, there are two. There are currently three MIDI formats: 0, which contains a single track; 1, which contain several simultaneous tracks; and 2, which contain several independent tracks. While the latter is more of a way of packaging many small melodies in one file, format 0 and 1 are widely and interchangeably used. Nagler provides converters for both.

[vii] If exceptionally spelt person and place names, alternative or obsolete pronunciations etc are considered, the figure will be even higher. Answering a question from a reader of the Swedish encyclopaedia Nationalencyklopedin 2002-02-08, Claes-Christian Elert enumerates 43 different spellings of the /ʃ/ phoneme.

[viii] One cent corresponds to a well-tempered hundredth of a semi-tone, as it were, i.e., to a frequency ratio of $2^{1/1200}$. For more details on tuning and musical acoustics, see Sundberg 1989.

[ix] The technique is known by several other names; such as 'selection-based synthesis', 'data-driven synthesis', or 'unit selection'. In this paper, however, the latter term will be reserved for the process of selecting the most appropriate candidate, rather than for the entire approach.