Two Unit Selection Singing Synthesisers

Fiona Skilling, B.A.

A thesis submitted in fulfilment of requirements for the degree of
Master of Science in Speech and Language Processing

to

Theoretical and Applied Linguistics
School of Philosophy, Psychology and Language Sciences
University of Edinburgh

August 2005

Declaration

I hereby declare that this thesis is of my own composition, and that it contains no material previously submitted for the award of any other degree. The work reported in this thesis has been executed by myself, except where due acknowledgement is made in the text.

Fiona Skilling

Abstract

Two speech synthesisers were adapted for singing synthesis using unit selection techniques provided by the Festival speech synthesis system. A limited domain approach was used by focussing on the pitch, duration and word of each note. The first synthesiser used the cluster unit technique on a database of an octave range, where each note had a specific word assigned to it. Some of the automatic techniques used (e.g. for segmentation) were designed for speech and should ideally be adapted to take account of the differences between singing and speaking.

Better quality was achieved with a multisyn engine and improved database design. This database used a smaller pitch range and only three syllables, 'la' 'ti' and 'so', but each syllable could be synthesised on any available note, and in any combination of notes and syllables. This was achieved by weighting the target cost of selecting units from the database in favour of choosing units with the correct pitch and duration. Finally, prosodic modification was applied to units in the multisyn engine, but this degraded quality as a result of how the units were modified.

Although the quality of synthesis was appropriate for the intended applications, the database was small and linguistic structure simple. To build a larger scale singing synthesiser, either some aspect of the database should be kept simple, such as vocabulary, or prosodic modification of units should be improved through further analysis of the characteristics of singing.

Acknowledgements

Very grateful thanks to:

Rob Clark for agreeing to supervise this project, solving so many technical problems in setting everything up and for being so helpful in dealing with all my questions;

All of MSc SLP '04-'05 for the community spirit that kept us going through this year of transitions - from the first deadlines to the last long hours in the lab;

Evia Kainada and Joanna Keating for their helpful feedback on the final draft;

Everyone who made it possible for me to move house in June and July;

My friends and family for all their support, understanding, and for keeping me sane - special thanks to Catherine Burt;

The Sacramento Micro Camp group, for everything that happened that weekend;

And especially, to my parents, for their never-ending support and encouragement.

Contents

List of Tables

List of Figures

CHAPTER 1

Introduction

1.1 Introduction to Singing Synthesis

Singing voice synthesis deals with many of the same issues as, and is able to draw on the techniques used in speech synthesis, but there are fundamental differences between singing and speaking that cannot be handled in a speech synthesiser. On top of the arbitrary sequences of words in speech is now an equally arbitrary sequence of notes. This brings advantages and disadvantages: on the one hand, intonation and duration of segments is much more predictable in singing than in speech, specified in advance as melody and note length, which removes a difficult area of speech synthesis. When considering the motivation behind speaking and singing though, a major difficulty arises: the intention in speech and its synthesis is usually to communicate a message, but in singing it is more often to convey the emotion of the lyrics, and expressing emotion in synthesis is an ongoing research topic in itself. Where understanding a message is most important, a degree of unnaturalness in voice quality may be acceptable if the message is still clear, but this would not be suitable for singing, particularly if synthesising a performance.

The goals behind speaking and singing are different, so it follows that the approaches taken in their synthesis will also be different. This section describes the main differences between speaking and singing and how these pose difficulties when building a singing voice synthesiser.

1.1.1 Pitch/Fundamental Frequency

The most obvious difference between singing and speech is that intonation is controlled by a melody which bears no relation to the linguistic message. Pitch movements in singing are also wider and more rapid than in speech, although less unpredictable and more consistent, constrained by the key the music is in and always aiming for a specific pitch (Uneson, 2002a). It can be assumed that two sung recordings of the same note, if in tune, will be very close in fundamental frequency, which is an advantage that can be exploited in synthesis techniques where prerecorded segments are concatenated, as these recordings would hopefully have a minimum fundamental frequency mismatch when joined [1].

As well as the melody, there are several ways in which the pitch of a sustained vowel does not remain constant in singing. Firstly, there are natural fluctuations in frequency present in any real voice, and a singing-specific phenomenon of initially overshooting the intended pitch when changing notes (Uneson, 2002b; Saitou et al., 2002). A more noticeable variation in fundamental frequency is vibrato, which can be described as:-

> ”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” (Uneson, 2002a).

If synthesising a style of singing which involves this or ’performance’ level output, vibrato must be added appropriately. The ear recognises when a sound is invariant, which is why synthesised notes which do not deviate in frequency at all sound lifeless or rigid (Ternström, 2004). To avoid this, the natural frequency fluctuation of a human voice would need to be included if a realistic voice were required.

Therefore, a singing voice synthesiser must be able to produce a wide range of notes, with some variation of fundamental frequency necessary to make a voice sound natural, and larger variations necessary to synthesise a realistic performance.

[1] The terms ’pitch’ and ’fundamental frequency’ are both used in this paper, though fundamental frequency could be used in all cases. ’Pitch’ is used to refer to a note in its musical context, and ’fundamental frequency’ where a more precise description is required in relation to Hertz.

1.1.2 Vocabulary

The vocabulary used in singing can be as varied as that found in speech synthesis or as simple as the 'ah's and 'oo's of backing vocals. Where only a subset of a language needs to be by synthesised, a lot of the difficulty and time involved in building a voice is saved, for example if only one song were being synthesised (Yoram, 1999). In contrast to speech, the identity of a word is sometimes less important than its intonation or timbre, so some of the time spent in working on clarity of articulation may also be saved (Uneson, 2002a).

Some speech phenomena such as vowel reduction could potentially also be ignored (or would need to be ignored if using a system designed for speech), since in general, vowels are not reduced in singing (Uneson, 2002a). Similarly, hyper-articulation which would sound out of place in speech (synthesis) is acceptable and even normal in singing, and can go unnoticed in singing synthesis. Again depending on genre, there may be specific requirements according to pronunciation, for example to sing 'ih' vowels in a more 'ii' manner when singing in a choir, which would have implications for text preprocessing. Overall, it must be remembered that though the vocabulary used may be the same as that found in speech, it is realised differently, so singing synthesis should not be approached from a purely speech perspective.

1.1.3 Duration/Rhythm

The second aspect of prosody controlled by a composer is the rate of singing, which is generally slower and has greater variance than in speech (Yoram, 1999). Notes and words can be very short or very long, and this variation is again arbitrary, apart from phrase final words which are usually sustained. As a result, there is a predominance of vowels in singing when compared to speech, since these are the units which carry the melody and can be sustained (Yoram, 1999). Singing also follows a predefined tempo, so synthesised singing must ensure that the timing of consonants and vowels is such that the perceived 'downbeat' falls on the onset of the vowel (Akagi, 2005). Therefore, in addition to the arbitrary sequence of notes that a synthesiser must be able to produce, all of these notes need to be available in all possible duration contexts, with these adhering to the specified tempo.

1.1.4 Vocal Quality/Timbre

There are as many considerations of voice quality in singing synthesis as there are in speech synthesis, though in singing this is often controlled and manipulated for artistic effect rather than as a result of emotion being expressed through voice quality in speech. To create a realistic, interesting and artistic voice, it should be possible to synthesise different voice qualities or timbres, for example breathy, pressed, nasal, strident, and different emotions. Again, this is an unsolved problem in speech synthesis and formal definitions of different voice qualities are elusive and lacking, so there are no perfected techniques available for use in singing synthesis (Rodet, 2002; Uneson, 2002a).

1.1.5 Singer's Formant

A characteristic of operatic and professional singers, the singer's formant is a peak of energy at 3-5kHz range caused by clustering of the third, fourth and fifth formants (Sundberg, 1999). Most singers trained to 'project' their voice will have a singer's formant, and without taking it into account, singing synthesis can sound weak or flattened (Yoram, 1999). Again, if seeking a true performance, this would need to be taken into account in singing voice synthesis.

1.1.6 Musical Interpretation

A musical score just provides the framework upon which a performance is built, and there are an inexhaustible number of ways to interpret this. One of the simplest is to vary dynamics, and information about dynamics is often included in the score (Uneson, 2002a). To create something approaching a musical interpretation, a synthesiser should be able to create dynamic contrasts, realised as gradual and controlled changes in amplitude across a phrase . The more authentic a performance is desired, the more specific the changes in dynamic would be required, for example so that each note had a slight crescendo and decrescendo, as highly trained singers are able to do.

To create a completely natural voice, it would also be necessary to account for the perception that amplitude increases as pitch does (Uneson, 2002b). Some other aspects of musical interpretation would involve varying the smoothness of the singing (staccato/legato), diction, place on the beat, and all of the ways in which a natural voice can add nuances through tone and timbre. This would

be the most challenging aspect to consider in synthesising a real-time singing voice, since it would involve understanding of the lyrics and what emotion is to be expressed.

From the above it can be seen that although singing provides some simplifications to the synthesis process, there are many additional features that must be taken into account, becoming increasingly challenging for computational approaches the more natural and high quality a performance is desired.

1.2 Applications and approaches to singing synthesis

The many different styles of singing place various levels of importance on the characteristics of a performance; an operatic singer will always sing with vibrato and a singer's formant, a choral singer much less so. It follows that a singing synthesiser's design depends on how it is to be used and so will not necessarily need to consider all the characteristics listed above. For example, using copy synthesis to provide stimuli in a perceptual study may not require much expression or even a large vocabulary range (Uneson, 2002a). Similarly, in the more obvious application of synthesising backing vocals, it may not be necessary to consider different vocal qualities, where this would not be noticed over accompanying instrumentation.

Another application is where synthesis is used to recreate performances of singers or styles which have died, so only those characteristics would need to be researched. Similarly, if the synthesis were to preserve or correct errors in recordings, or to mimic a given performer (Yoram, 1999). Even a trained singer is usually a specialist in a particular style, so though it would certainly be an achievement to create a synthesiser able to recreate many styles, in practical terms it is normal to focus on one area.

An application of interest to the current project is synthesis as an aid for composers and arrangers (Uneson, 2002a). In this context, a synthesiser would act as the tool to provide drafts, so therefore it might again not be necessary to include the higher level performance charactersistics of singing. This would also be useful as a teaching tool if it were not possible to record a competent singer, or where this was difficult to organise. In both of these applications, clarity and precision of notes is more important than vibrato or varying timbre, as singers will add this themselves.

The ultimate singing synthesiser would be one able to synthesise a performance, where any lyric could be synthesised on any musical phrase. The closer the intention is to create or recreate a real performance, the more characteristics of singing must be taken into account.

Singing voice synthesisers have been created for many of the above applications, with approaches including formant-based, source-filter, vocal modelling techniques (hereafter referred to as rule-based methods) and concatenative approaches where recordings of real singers are used (Ternström, 2004). For rule-based synthesisers, where pitch and duration can be precisely controlled, the difficulty is in how to make the voice produce all the characteristics of a real voice, and controlling where and when they are present. Where duration is controlled by repeating an existing recording of singing or lengthening the characteristics of one segment, this can give the impression of 'buzziness', and is a sign of a synthesiser which has not correctly modelled all the characteristics of a natural voice (Ternström, 2004).

Like speech synthesis, singing synthesis has been criticised for sounding unnatural because it is neutral, with no range of expression (Yoram, 1999). Singing synthesis will also sound unnatural when a large pitch range is used by the same voice. The ear is aware of what a normal production range is, and although singers are able to sing very high and very low with quality, their normal range does not usually include both extremes. Therefore, to hear a single voice sing with the same voice quality at extremes which are beyond any human singer is unnatural (Ternström, 2004). Some examples can be heard on the website of the 2nd Conference on the Physiology and Acoustics of Singing in a song commissioned for several singing synthesisers to sing together, with some (deliberate) comic effects (Ternström, 2004).

This again returns to the issue of how natural a synthesised voice is - there are unavoidable changes in voice quality throughout any singer's register, so to hear a synthesised voice where these are lacking is also noticeable. Where the intention is to mimic a human singing voice therefore, not only the capabilities but also the limitations of a human voice should be considered for the synthesis to sound natural.

Large advances have been made in achieving natural sounding speech synthesis with unit selection synthesis, because recordings of real speakers are used. This concatenative method of synthesis is particularly effective when used in

a limited domain (e.g. talking clocks in speech synthesis), as the database of recordings is specifically designed for the given domain, ensuring high quality synthesis (Black and Lenzo, 2000). Concatenative techniques were applied to singing relatively recently, but have the same potential for improving naturalness of the synthesis, since characteristics of singing are already present in the recordings and do not have to be artificially created. Several singing voice synthesisers exist which use recordings of singers, including Lyricos, developed at the Georgia Institute of Technology; Flinger, a customised version of the Festival speech synthesis system; Vocaloid, developed at Yamaha; and various research systems (Uneson, 2002a; Yoram, 1999).

Though a high level of naturalness can potentially be achieved using existing recordings, other problems are raised in doing this, and the concatenative approach does not solve all the problems of rule-based singing synthesisers. In all the systems listed above, units require prosodic modification to match the target song, which can cause the problems associated with rule-based systems (Yoram, 1999; Ternström, 2004; Uneson, 2002a)[2]. Some other problems of rule-based synthesisers are discussed in the next section.

All singing voice synthesisers aim to mimic a real voice, but the level of naturalness required depends on the application. Even where a full performance is desired, a level of unnaturalness may go unnoticed when the singing is accompanied by instrumentation or is in the background. The next section describes the singing synthesiser this project aims to create, and the methods that will be used to achieve this.

1.3 The present project

My interest in singing synthesis is based on using synthesised four part harmony singing to learn songs. The synthesis had the tinny quality of a formant or rule-based synthesiser, with no expression or variation in the musical line. Because the same voice was used for all four parts this made it difficult to distinguish which voice I needed to listen to, as there were no individual voice characteristics to tune in to. Musical intervals did not sound natural since they were very precisely timed, which is not expected in terms of how singers can overshoot

[2] An interesting point to note however, is that a natural voice can also sound 'buzzy', particularly when listened to out of context, so if a unit were chosen which had this quality, this would still sound unnatural (Ternström, 2004).

notes and was at worst very unnatural, since a real singer would be physically unable to move their larynx to produce such abrupt changes in fundamental frequency. The tinny quality also made it hard to understand the words, which did not help in learning the songs.

Expressiveness was unnecessary, but improved naturalness would certainly help, particularly given that the singers learning from these recordings were trained in blending and matching to the quality of the voice they are listening to, and for harmony parts, in tuning to the voice carrying the melody (Kalin, 2005). From experience of learning from both synthesised and natural recordings, this is intuitively easier with a natural voice.

The second application I am interested in using singing synthesis for is in arranging material. Although this again is related to working in harmony, a synthesiser would be a very useful tool if it could easily synthesise and store any number of melodies and play them back immediately, since this is a time consuming task when using just a piano and simple hand held tape recorder. Again, expression is not essential, neither are vibrato and other musical embellishments.

The singing style of interest here is barbershop singing, which presents specific difficulties to do with acapella harmony singing that are beyond the scope of this project, but will be mentioned briefly here since they relate to both applications. Barbershop singing involves several specific characteristics - tuning according to Just Intonation, formant frequency adjustments not found in other singing styles, and audible overtones in high level performances (Kalin, 2005; Sundberg and Hagerman, 1980; Szabo, 1976). These would need to be considered for natural sounding synthesis in this style. For present purposes however, a synthesiser that would be a tool for producing drafts even by synthesising just one melodic line would still be quicker than recording one's voice and playing it back.

A large problem with the synthesised tapes I was learning from was that they sounded unnatural. To improve this using a rule-based approach, the characteristics of singing would need to be further studied and understood, to modify the acoustic signal more accurately. Alternatively, the task can be approached with unit selection synthesis, where the benefits of using a real singer include (potentially) a high level of naturalness, and for the two applications of interest here, would be easier for those wishing to tune to a real voice. The rest of this section describes unit selection synthesis in more detail and why it is the approach used in this project.

1.3.1 Unit selection synthesis

The main issues in unit selection synthesis, both for singing and speech, are database design, how units are selected from it, and how they are modified. These issues are now described in relation to the present task.

Database design

The size and design of databases for unit selection synthesis is an ongoing research topic in speech synthesis, further complicated when adding all of the characteristics of singing. A database for English already requires 1000-2000 diphone contexts, and each diphone would need to be available on any note, resulting in an impractically large database (Clark, 2005). Add to this that each note may be of an arbitrary duration and the numbers increase again. This is why the existing unit selection or concatenative singing synthesisers listed earlier involve some prosodic modification of units in their database. Modification of units should be kept to a minimum to avoid introducing artefacts from the required signal processing (Yoram, 1999).

Since it is out of the scope of this project to build a free text synthesiser, it is clear that a limited domain style approach may be useful. Unit selection synthesis already limits the amount of material that must be recorded by using units such as diphones or triphones in enough contexts to cover a given language, but in this case, the question is how to simplify the language itself in terms of vocabulary. Some existing limited domain approaches to singing synthesis include recording only a specific song or set of lyrics, just the relevant possible vowel or consonant-vowel combinations for a language, which would be simpler for languages which only have consonant-vowel syllables such as Japanese (Yoram, 1999; Uneson, 2002a; Akagi, 2005). Another option is to only use vowels, since voiceless consonants in particular cause difficulty in singing synthesis (Yoram, 1999).

In terms of musical range, pitch could be restricted to a certain range (and would necessarily be restricted by the range of the person making the recordings). To reduce recording time, some fundamental frequency modification could be carried out on existing recordings to extend the range (with increasing impact on quality). Maintaining similar acoustic properties across all recordings will be vital to ensure that the units can be joined together as smoothly as possible, so using just one voice quality would be preferable. The safest place in which to

limit the domain is vocabulary, as a singing voice synthesiser would be of little use to a composer or arranger if missing notes.

Selecting units for synthesis

Although I have described unit selection synthesis as being separate from rule-based methods of synthesis, this is not entirely true, since the method of selecting units for synthesis can follow as many conditions and be as hand-written as a synthesiser based on rules written to recreate the characteristics of singing. In this approach, a sequence of units needs to be selected from the database which best matches a given target song. Firstly, the units which best match the specification are selected by way of a target cost for each unit, and this is traded off with a concatenation cost for joining these units. The global cost, calculated using a Viterbi search, is a weighted sum of the individual costs, balanced according to the relative importance of the target and join costs (Rodet, 2002).

Two unit selection distance measures will be used during this project. One of these methods uses a true target cost; the sum of hand-written weighted functions, where a penalty cost is added to a candidate unit which does not match the target, and the best units are those with the lowest costs (Clark et al., 2004). For speech, this includes for example costs for phones being the correct stress and fundamental frequency, and what position they are in a phrase (Clark et al., 2004). These weightings will need to be adjusted for singing, for example to ensure that a unit with the correct pitch and duration is very strongly weighted, as a vocabulary error would be less noticeable than an error in pitch (Rodet, 2002).

The other method for selecting units from a database used in this project is clunit synthesis, where a target cost is approximated by clustering units according to phonetic and prosodic characteristics (Black et al., 2002). Units are selected for synthesis according to which is closest to the centre of the cluster, using LPC coefficients to provide the acoustic distance measure (Black et al., 2002). The advantage in this is that units recorded next to each other have a zero join cost and so are more likely to be selected where they match the target utterance/song. This is why unit selection can produce such high quality speech synthesis, particularly when using a limited domain, because using units larger than diphones/triphones reduces the number of concatenation points which could cause artefacts.

Practically speaking, however, unit selection synthesisers require much more storage space than rule-based methods, and though rule-based synthesisers may sound synthetic, they do at least produce a consistent quality. This cannot be guaranteed from unit selection synthesis, for example through artefacts caused by mismatches at concatenation points or the result of bad digital signal processing. Since the rest of the quality of the singing will be completely natural, these artefacts would be more noticeable and unacceptable than listening to something which is obviously synthetic, but at least consistently so.

Approaches to be used

In the present project, the Festival speech synthesis system, developed at the Centre for Speech Technology Research at the University of Edinburgh, will be used to build two singing synthesisers, using the target cost and clunit distance measures to select units. The cluster unit approach will follow a procedure I am familiar with, having previously used it to build a speech synthesiser. This will be used to test issues regarding database design and time needed for recordings, as well as the feasibility of using a speech synthesiser for singing synthesis. A second design will use target costs as applied in the multisyn engine of Festival Version 2, which offers the opportunity to hand write target costs, so that the units are chosen according to the singing specification, rather than rules based on speech.

An issue which has not yet been discussed is how to encode musical structure into the database. This is necessary so that a word can be requested on a specific note, a feature speech synthesisers do not require. An additional advantage in using Festival is that it already contains software designed to handle this situation. A singing mode was developed by Dominic Mazzoni which reads musical structure from XML files (Mazzoni, 2001; Black et al., 2002). This can also be used to allow prosodic modification of existing units so that they match the pitch and duration specified in the XML file, producing musically accurate, though not very natural sounding synthesis, because it is based on speech recordings. The task here will be to investigate using this singing mode with a new (and specifically singing) voice which has been built from scratch.

CHAPTER 2

Clunit Synthesiser

2.1 Design and preprocessing

The first synthesiser made in this project used the building procedure of a clunit speech synthesiser provided on the Festival website (Black et al., 2002). A very limited domain database was designed which simplified away many problems associated with singing synthesisers. As described in the introduction, many features would need to be considered to create high quality synthesis, but for a synthesiser to be used as a teaching or arranging tool, only the following three issues need be considered:-

- Pitch
- Word
- Duration

The simplest way to account for pitch and words together is to have a vocabulary where each word is only ever sung on one note. For teaching purposes it is also very helpful to learn just a sequence of, for example, 'doh ray mi fa so la ti do' knowing that each note will always be the same word. Therefore, although this approach may be considered simplistic, it might be the most practical one for a teaching aid. Using this design made it possible to use the same architecture as for a speech synthesiser, since no musical information needed to be encoded into the database, and units would be selected as words, not notes.

The vocabulary consisted of the numbers one to eight, and a one octave range was chosen, initially from F3 to F4, but this was changed to be from G3 to G4 when recording prompts, as I found my change in vocal register between the

Note 1	Note 2	Note 1	Note 1	Note 2	Note 1
one G3	one G3	one G3	two A3	one G3	two A3
one G3	two A3	one G3	two A3	two A3	two A3
one G3	three B3	one G3	two A3	three B3	two A3
one G3	four C4	one G3	two A3	four C4	two A3
one G3	five D4	one G3	two A3	five D4	two A3
one G3	six E4	one G3	two A3	six E4	two A3
one G3	seven F#4	one G3	two A3	seven F#4	two A3
one G3	eight G4	one G3	two A3	eight G4	two A3

Table 2.1: The recordings required to produce all necessary contexts for two of the notes in the octave used (eight per note)

higher and lower notes less noticeable in this range, and it was important to keep as consistent a quality over recordings as possible[1]. The final factor, duration, I controlled by making each note the same length. It was suggested at an early stage that prosodic modification could be carried out to produce durations other than those recorded, so one-second notes were used, to modify these to two-second and half-second notes if needed.

The principle behind using a limited domain is that every possible context is recorded in advance, which, due to the arbitrary nature of a musical phrase, means that every note/word should be recorded next to every other note/word (and also next to silence). Certain intervals (i.e. contexts) are likely to be used less often than others, but it would not be appropriate to restrict a tool designed for an arranger or composer, so all contexts were considered equally important. I felt that with suitable prompts it would be possible to sing all the required intervals. After experimenting with a variety of approaches, I found the simplest way to ensure coverage of all contexts was as illustrated in table 2.1, where the word in question begins and ends a phrase, with all other words recorded between it.

With an octave range, this requires 64 recordings of three notes each, which provides two samples of each context plus eight samples of the note before and after silence. I decided to keep the phrases just three notes each so there was a good chance the voice quality would remain consistent, not being affected by a dip in energy or change of tone caused by running out of breath. In planning ahead for how to extend this design, I felt that once I had sung an interval once, it would be easy to repeat this same interval several times to produce a larger number of

[1]Musical notes are referred to according to the convention that numbering starts on C, where C4 is middle C on a piano. Table 2.1 illustrates the notes used in the key of G.

	Note 1	Note 2	Note 3	Note 4
1	one G3	two A3	three B3	four C4
2	four C4	three B3	two A3	one G3
3	five D4	six E4	seven F#4	eight G4
4	eight G3	seven F#4	six E4	five D4

Table 2.2: Additional prompts used in the clunit synthesiser database

samples, given that two is the minimum suggested for a unit selection synthesiser (Black and Lenzo, 2000).

Alternative approaches include writing a programme to create random phrases of notes where all contexts are covered and taking the appropriate contexts from these, providing the melodies could be easily learnt. Using songs or familiar melodies may be easier for the singer making the recordings, but might encourage artistic interpretation, affecting voice quality. It may also take longer to record all contexts. For present purposes I was satisfied to use the design as in table 2.1. To provide some examples of longer phrases, I included four sequences of sequential phrases as shown in table 2.2.

To build this synthesiser, a file was required which listed all phrases to be recorded (hereafter referred to as the .data file). This was exactly the same design as that required in building a speech synthesiser (Black et al., 2002).

2.1.1 Recordings

Recordings were carried out in quiet but not soundproofed conditions (the only sounds coming from nearby computers), using an AKG D 90 S microphone. Although Black and Lenzo (2000) emphasise the need for high quality recordings, they do also suggest this approach as a simple way of achieving suitable quality recordings. Duration of each note/word was controlled using a stopwatch as a simple alternative to a metronome, with each second treated as a downbeat. Though this approach took some practising, when checking the phrases after recording they were perceptually the right tempo. All sequences were recorded individually, rather than using the available setup to automatically prompt, record and name files, since this made re-recording sequences easier.

I controlled for pitch using piano prompts of each sequence and checking each recording for accuracy. In making the prompts and practising them I noticed that

some intervals were harder to sing than others, particularly those beginning on 'seven'. No intervals were impossible, and if present in a melody would simply be learnt through practice, but singing them in isolation was unusual. Listening back to the recordings, an occasional but obvious problem was a difference in amplitude between different notes. This was related to singing large intervals, where the higher note was much quieter than the lower due to the different singing register I was using. More practice or warm-up should reduce this effect.

As described in the introduction, one of the difficulties of synthesising singing is that there are so many additional factors to take into account regarding voice quality. For all the recordings I carried out however, the only major differences to speech were in duration and pitch. It happens that I do not usually sing with vibrato, since my training is choral, so this was not an issue. I aimed to sing in a relaxed mode and to be as precise as possible, similar to how I have sung when making recordings for others to learn from. Accuracy is necessary for a teaching tool, and I felt that singing in a relaxed manner would be easier to maintain over a long recording time than singing loudly. A result of this was that I did not project in the manner that would produce a strong singer's formant, which would hopefully mean that energy in the higher frequencies would also be relatively consistent throughout the database.

Overall, it was easy to maintain this approach to recording the database, but between each recording I had to save files, occasionally erase and re-record, then listen to or find the right prompt and practice the next interval. These were not ideal conditions for remembering and maintaining consistent vocal quality between recordings, and also not for recording always at the same distance from the microphone. From experience of building a speech synthesiser I was aware that any changes in voice quality as a result of these problems might affect synthesis quality. A power normalisation script was run on the database at a later stage to account for possible differences in amplitude.

When listening back to the intervals I had found hard (including most recordings involving 'seven'/F#4), they did not sound musically correct, but careful checking against the prompts and using a pitch pipe confirmed that all intervals had been sung correctly. This illustrates that the recordings could not have been made without prompts, and that the simplest looking design was not the easiest to record.

The reason these intervals were difficult to sing relates to how easily the key note or tonal centre of the scale could be kept in mind. Musically trained singers can be expected to know certain intervals, such as a third, fourth, fifth and octave, and this is easily done from the key note of a scale. The problem here is that the intervals required between, for example, 'seven'/F#4 and every other note in the key of G, do not include many of these familiar intervals, and the temptation is to treat F# as the key note since it is most often sung. Though an interesting exercise in musical skill, this was a difficult design to follow and would need to be adjusted to avoid the risk of errors.

Two errors went unnoticed in the recordings. Unsurprisingly, these were two of those added at the last moment and which did not have prompts (those sequences involving 'seven' from table 2.2). They resulted from confusion over the key note caused by the difficult intervals. On reflection, I considered that it may not be easy to record semitones using this design, since these would cause many more such unusual intervals. I removed the two incorrect files from the database, since corrections would probably have a noticeably different voice quality to the original recordings. I also removed the remaining two sequences in table 2.2, to allow a test of the synthesis given just the simple database design of table 2.1.

After recording this database design, I listed the following areas for improvement:

- Change the design of the database to avoid isolated unusual intervals;
- Include more samples of each context;
- Test using a shorter note length;
- Record in sound-proofed conditions;
- Record everything at once and segment it afterwards, to make it easier to repeat phrases without affecting voice quality;
- Investigate different methods of keeping to tempo.

A person with perfect pitch (the ability to recognise what note they are singing without reference to a tuned instrument) may be useful for recordings, since hopefully they would not make simple errors, but this was not possible. Total recording time was around one and a half hours.

2.1.2 Labelling

Following the instructions and prompts to build a speech synthesiser provided on the Festvox website, scripts were run to automatically label the recordings. These used Dynamic Time Warping to align between the mel-scale spectral coefficients of the synthesised utterances of the .data file (phone labels known) and the recorded waveforms (phone labels unknown) (Black and Lenzo, 2000). Unsurprisingly, since the synthesised utterances were made with reference to the charactersistics of speech, the automatic alignment was rarely successful. It often began labelling during the silence before the singing had started, which naturally caused all of the following labels in that file to be incorrect. I had previously noticed this when using these scripts to label speech, so this was not purely as a result of giving it singing to recognise.

From experience, and as generally acknowledged, the quality of the synthesis would depend on the quality of the alignment, particularly given that phones were the units of selection, so all labels were checked and edited (Black and Lenzo, 2000; Macon, 1996). As discussed by Yoram (1999), recognising sung words is already harder for humans in comparison to recognising spoken words, so either the synthesis of the .data file or the recogniser would need to be adapted for singing to improve labelling. The best option would be to incorporate knowledge of the musical structure into the process, since durations are known, but this was not possible given the design of the speech synthesiser.

While listening to the sound files more closely than I had been able to during recording, I noticed that pitch was not always consistent throughout a word, particularly for 'seven', where pitch was only accurate on the second syllable. This is another result of the difficulty in singing unusual intervals, which had extended the natural overshoot in singing the interval (Akagi, 2005).

2.2 Building

2.2.1 Pitch/Fundamental Frequency Marking

An important stage in building synthesisers is extracting pitch periods, and good pitchmarking is essential for seamless concatenation of units. The step following pitch-marking (finding the Mel Frequency Cepstrum Coefficients (MFCCs) for use in clustering and join measurements) is done pitch-synchronously, so also

	Speech	Singing	Values used
Upper limit	295	392.Hz (G4)	410
Lower limit	120	195.99 (G3)	180

Table 2.3: Fundamental frequency range used

relies on pitchmarks (Black et al., 2002). An autocorrelation technique was used to extract pitchmarks from the recordings here, but it should it is accepted that the most effective way is to use an electro-glottograph (EGG) signal (Black et al., 2002). The fundamental frequency parameters provided were based on speech, so were changed to fit the octave range that would be found in singing, adding a little over and below the expected values for this octave, as shown in table 2.3. Unvoiced segments are difficult for automatic techniques to mark correctly, so there is an advantage in working with singing where there are fewer unvoiced sections overall (Yoram, 1999).

An alternative solution to this octave range is to use the fact that the data is labelled. An upper and lower limit could be set for each segment, matching what the pitch of that segment is known to be, since it can hopefully be assumed that the correct note was sung. This would avoid potential errors as a result of providing a very wide range. Of course, as described above, 'seven' was sometimes only correct on its second syllable, so allowances would have to be made before causing errors through giving too restricted a range.

On analysing the pitchmarking with xwaves, it appeared that most voiced sounds had been pitchmarked appropriately, but there were general problems, including that the background humming of the nearby computers was periodic, so may have interfered with the acoustic signal of the singing. The automatic procedure had difficulty in marking the edges of vowels and the nasal in 'seven'. The pitch of the note seemed to affect pitchmarking; sections of nearly all samples of notes from D4 and higher were missing pitch periods, particularly for 'seven' and 'eight', on F#4 and G4. The question was therefore if this would result in poor quality synthesis on these notes. If I had been projecting my voice it may have been necessary to address how the energy in the voice was dealt with, since such energy is not found in conversational speech, but since I was using a relaxed or 'neutral' singing voice, this was not an issue (Yoram, 1999).

2.2.2 Cluster Unit Synthesis

In the synthesis method used here, units are clustered together according to decision trees based on phonetic and prosodic contexts, and are chosen for synthesis in relation to their position in a cluster. Units are assigned a cost, determined by the acoustic distance (in LPC coefficients) of that unit to the centre of its cluster; the further from the centre, the higher the cost of selecting that unit (Black and Lenzo, 2000). To this cost of selecting a unit from a cluster is added the join cost of concatenating that unit to its neighbour. Units selected for synthesis are chosen according to the best acoustic distance measure of their joins, where the sequence with the lowest overall cost is used. A unit may have a good target cost in terms of position in a cluster, but a different unit from that cluster actually chosen at synthesis because it has a better join cost to its neighbours.

As described in the introduction, in unit selection synthesis there is also the possibility that larger units are chosen, effectively as if they are diphones, reducing the overall number of concatenation points (Black and Lenzo, 2000). LPC coefficients were calculated for the recordings, and Mel Cepstrum parameter files generated at pitchmarks. After these steps, scripts to build the clunit (cluster-unit) synthesiser were run which built the decision tree according to the distances calculated as described above (Black et al., 2002).

2.2.3 Evaluation

One of the intentions in using this speech synthesiser was to test issues regarding database design, and since so many improvements had already suggested themselves, the evaluation of this synthesiser was basic. Based on the observations regarding the number 'seven' I expected some poor quality when using this note, and changes in voice quality across large intervals. Some general observations follow here, based on synthesising phrases which I knew the database contained in complete form, such as 'one one one', phrases it did not have, and some longer phrases that were specific melodies, with which I could judge how easy it would be to learn from this voice.

From synthesising phrases stored in the database it was clear that even where there was a complete recording of a sequence available, this was not necessarily chosen. Of the 64 phrases synthesised (the number of sequences in the database) 15 had some small artefact in them, and eight had an obviously noticeable artefact. The other 41 had clearly been taken directly from the recordings and so

were completely natural in quality, although the difficulties I had had in producing some of the intervals showed in an apparent lack of confidence.

Interestingly, there were errors at some point for every note, so the artefacts could not all be attributed to poor pitchmarking, since this had mainly been the case for numbers from 'five' upwards. However, the difficulties in pitchmarking nasals was reflected in some poor quality synthesis of these. The pre-recorded sequences might not have been simply selected as a whole because where there were pitchmarking errors, this resulted in a given unit not having a good enough join cost to be selected. Individual missing pitchmarks should not by themselves create artefacts, so another possibility is that artefacts were caused by bad signal processing implementation, and as concatenation points were at phone boundaries, errors in labelling may also have caused the artefacts (Black and Lenzo, 2000; Black et al., 2002).

These possibilities also hold as explanations of why artefacts occurred in synthesising truly arbitrary sequences of notes. Overall, quality ranged from relatively natural to obviously synthesised. As well as artefacts there were also differences in voice quality, already explained as a result of the database design, and which show that the power normalisation step had not been able to account for this aspect of differences in amplitude. Another noticeable error was that duration of segments was not consistent. Of course, no facilities had been put in place to ensure that each note would be a second long, apart from aiming to sing them as such, so this problem was not surprising. It was also difficult to judge duration in the synthesis due to the slow tempo when each note was a second long. Two examples of synthesis using this voice are available on http://www.ling.ed.ac.uk/ s0454934/, as can the later examples of synthesis in this project.

The evaluation of this design was brief, given the planned improvements which would hopefully solve issues of voice quality. In design, this was still a speech synthesiser, so the issue of how to incorporate musical knowledge in selecting units had not been addressed. Although the technique had successfully produced a synthesiser able to sing any combination of the available words, a more interesting challenge would be to create a true singing synthesiser which could sing any word on any note. The original list of areas for improvement is reviewed below to include the observations made during building and testing this voice.

- Design
  - Remove unusual intervals to help voice quality consistency.
  - Be able to synthesise a word on any note.
  - Include more samples of each context.
  - Use shorter notes.
  - Use a simpler linguistic structure.
  - Use more than one duration.
- Recordings
  - Use a sound-proofed room to help pitchmarking and overall quality.
  - Investigate different methods of keeping to tempo.
  - Record everything at once to improve voice quality.
- Labelling
  - Use musical knowledge when labelling.

I was also interested in checking whether the observations made regarding pitchmarking would hold in a second design, and if overall quality could be improved. The clunit design, since it was familiar to me, took less than a week to build once the design was complete, so does stand as an example of what can be achieved by oversimplifying the requirements of a singing synthesiser. At this stage however, the resulting quality does not warrant using unit selection as a tool for arrangers or to teach others. The next chapter discusses how these issues were addressed by using different software provided by Festival, and a different approach to unit selection synthesis.

CHAPTER 3

Multisyn Engine Synthesiser

3.1 Differences from the first design

This second design combined two facilities available in Festival: the singing mode and multisyn engine. The main difference in the synthesis approach is that instead of using a cluster tree to select units, a multisyn engine voice uses hand written target costs, allowing for more control over unit selection.

The voice was built according to the documentation provided for a multisyn engine, and has similarities with the clunit synthesiser in terms of software and procedure. As before, building involved recording, labelling and pitchmarking units and calculating information that would be needed at synthesis, but several speech-specific steps were skipped, and the .data file referred to XML files containing the musical structure of the sequences, instead of just containing words. The singing mode was configured to accept a voice built using the multisyn engine, and the multisyn engine configured so that it could select units according to the singing mode, and from the correct directory. (Full credit and thanks to my supervisor for understanding and solving these problems when they arose.)

3.2 Design and preprocessing

3.2.1 Modifying the database design

Although the cluster unit synthesiser was intended to be a simple design, I have already described how it was difficult to follow, and using full words resulted in more concatenation points than could be described as simple. The database

in this design uses single syllables, and the 'first note, second note, first note' format was modified in several ways. Firstly, to provide more samples, notes were shortened to 500ms, meaning that longer phrases could be recorded. The easiest in terms of breath support was a nine note sequence as below:

G3-la B3-la, G3-la B3-la, G3-la B3-la, G3-la B3-la, G3-la

Using this pattern, not only are all contexts of G3-la to B3-la recorded at once, but so are all those of B3-la to G3-la, so it becomes unnecessary to also record this sequence beginning on B3-la. This solves the problem of (some of the) unusual intervals, since a sequence can be started on either note in an interval, so the easiest starting note can be chosen. For example, in singing the interval of a seventh ('one' to 'seven' in the old design), it would no longer need to be recorded starting on 'seven', the note for which it was harder to keep the key in mind. Reducing the duration of each note would also hopefully make it easier to keep to tempo and evaluate this in the synthesis.

The cluster unit synthesiser bypassed the problem of pitch in singing by having each note be one specific word, but in this design the intention is to test how well a less restricted vocabulary can be built. Since a text free vocabulary is beyond the scale of this project, the words/syllables used here will just be unrestricted in the sense that the available words will be singable on any available note. The original intention was to use the 'doh ray mi' scale in an octave, though this was reduced to be the syllables 'la', 'ti' and 'so' (themselves part of the 'doh ray mi' scale). These syllables were chosen as they include a plosive, liquid, fricative and diphthong, providing a range of different sound types for the synthesiser to deal with.

First half	Second half
Silence	la
la	la
la	ti
ti	la
la	so
so	la
la	silence

Table 3.1: Contexts required for 'la'

As before, each syllable needed to be recorded in every context, so taking for example 'la', the 'l' and 'a' must be recorded in the contexts listed in table 3.1, so that it can be synthesised going to every other word. These can be covered by recording 'la' followed by each of 'la' 'ti' and 'so', and ending on 'la'. Using the nine note sequence described above this provides four samples of each context (first note/word to second note/word and vice versa) and one sample of the first note/word preceded and followed by silence in one recording. For the three words, the phrases in table 3.2 cover all necessary word contexts.

	Word 1	Word 2	Pair repeated, ending on Word 1						
1	la	la	la	la	la	la	la	la	la
2	la	so	la	so	la	so	la	so	la
3	la	ti	la	ti	la	ti	la	ti	la
4	ti	ti	ti	ti	ti	ti	ti	ti	ti
5	ti	la	ti	la	ti	la	ti	la	ti
6	ti	so	ti	so	ti	so	ti	so	ti
7	so	so	so	so	so	so	so	so	so
8	so	la	so	la	so	la	so	la	so
9	so	ti	so	ti	so	ti	so	ti	so

Table 3.2: All word to word contexts

These nine phrases form the word context recordings needed for a single interval. Table 3.3 illustrates the interval context recordings that would be needed if using the old design of recording each interval starting on each note (576), and those needed with this new design (324). When beginning, no interval contexts have been recorded, so G3 requires all contexts to be recorded. However, when looking at the interval contexts required for A3, using table 3.3 it can be seen that the interval A3-G3 has already been covered in the context G3-A3, so the nine recordings needed for this interval can be skipped. Using this design, every note needs to be recorded in the interval context of 'zero' interval (i.e. G3-G3, A3-A3) and to all other notes above it. This approach saves 252 recordings (9 word contexts x 28 interval contexts).

One other saving in recording time can be made through necessary contexts being recorded elsewhere. When the interval is to the same note, there is no need to record both 'la ti' etc. and 'ti la' etc., since just one recording will provide all samples. For these notes therefore, six sequences suffice. These are: 'la la', 'la so' and 'la ti'; 'ti ti' and 'ti so'; and 'so so' (1, 2, 3, 4, 6 and 8 in table 3.2). The one exception to the nine or six sequence formats is the very last interval of G4-G4,

	G3	A3	B3	C4	D4	E4	F#4	G4
G3	N	N	N	N	N	N	N	N
A3	O	N	N	N	N	N	N	N
B3	O	O	N	N	N	N	N	N
C4	O	O	O	N	N	N	N	N
D4	O	O	O	O	N	N	N	N
E4	O	O	O	O	O	N	N	N
F#4	O	O	O	O	O	O	N	N
G4	O	O	O	O	O	O	O	N

Table 3.3: Interval contexts in the old (O) and new (N) designs

since using only the six sequences provides less than four samples of each word next to silence. To provide a minimum of four samples of each note in this context, this interval was extended to 12 sequences; the original nine plus one extra sample of each word going to 'la' (1, 5 and 8 in table 3.2).

Overall, the 576 recordings (64 x 9) can be reduced to 306 (7 x 6, 1 x 12, 28 x 9) for recording all contexts required in an octave range with a vocabulary of three words. As well as saving recording time, this design removes some of the more unusual intervals, for example in that only two intervals would need to begin on the note 'seven'. To test the building process the first four notes of this octave were used: G3, A3, B3 and C4, requiring 84 recordings (3 x 6, 1 x 12, 6 x 9) as illustrated in table 3.4. (Ultimately, there was not time to record the whole octave, so testing the design with the harder intervals of notes above C4 was not possible.)

	G3	A3	B3	C4
G3	6	9	9	9
A3	-	6	9	9
B3	-	-	6	9
C4	-	-	-	12

Table 3.4: Intervals and number of recordings required in the subset design

3.2.2 Input to the multisyn engine

The multisyn engine required the following as input:

Lexicon This was required even though the vocabulary was very small. The standard lexicon was used as it already contained 'la' 'ti' and 'so'. (Had it not these would have been added.)

Phoneset This listed the phonemes in the database.

List of back-off phones The standard list was used which, though speech-specific, had to be included for the system to compile.

.data file This referenced the XML files which made the database.

XML files Described below.

Of this input, the XML files are the only additional information required for singing. They described each sequence in terms of pitch, duration and words/lyrics. A sample of the information included in these files is shown below.

```
SINGING BPM="60"
DURATION BEATS="0.5" PITCH NOTE="C4" la
DURATION BEATS="0.5" PITCH NOTE="C4" ti
DURATION BEATS="0.5" PITCH NOTE="C4" la
```

I used a Python script to create the XML and .data files, which was useful when inevitable changes had to be made and errors corrected.

3.2.3 Recordings

Recordings were carried out in sound-proofed conditions using an AKG CK98 Hypercardoid microphone and sampled at 48kHz. Sequences were recorded onto one file and segmented by hand using Cool Edit, with roughly 500ms of silence before and after the sequence. The documentation suggested a method of automatically splitting a database into individual files, but since I usually sang phrases more than once before I was satisfied with the quality, it was easier to do this by hand. As I intended to record a full octave at a later stage, I again used a stopwatch to control for duration.

When listening to the recordings I found that the last note was usually a whole second long, caused by my choral experience, where final notes are always sustained. This was only recognised as an error during testing, which will be discussed later. As a result of this, continued difficulties involved in using a stopwatch as a metronome, and in anticipation of later recording a full octave, I created visual prompts in ePrime to act as a metronome for any future recordings, but this was never used.

In general, recordings were more consistent than in the first design, never being of a large enough interval to cause obvious differences in voice quality, although intensity decreased towards the end of the phrases. However, no power normalisation was run on this design. Overall recording time was around two hours.

3.2.4 Labelling

Segmentation was carried out by forced alignment using the HTK HMM toolkit (rather than recognition, since the phone sequence is known in advance), where each sequence was linguistically analysed, synthesised, and this synthesis used to align labels with the recordings (Clark et al., 2004). Scripts provided for building a multisyn engine speech synthesiser were used to create a master label file of labels established by the linguistic analysis and make MFCCs of the recordings, used for initial training of the HMMs, after which the forced alignment was carried out, with the final alignment producing the labels for each recording (Clark et al., 2004). Two errors were noticed and corrected after checking the labelling: the contents of one XML file were incorrect (I could find no cause for this) and the original XML files had only eight notes in them instead of nine.

Although still using a technique designed for speech, when labelling 'la la la' etc. sequences the automatic segmentation performed better than expected, and little hand editing was required for these files. An example is illustrated in figure 3.1, showing the similarity between the original and actual segments for this file. In most examples, all the singing had been segmented, which the same process had not achieved in the clunit synthesiser segmentation. A possible explanation is that it was easier for the recogniser to cope with the simple linguistic structure of syllables, particularly when these were fully voiced.

Although segments for 'l' often overlapped into neighbouring vowels, this was better than the automatic segmentation of the other consonants as illustrated in figure 3.2, where both the 't' and 's' are longer than they would be even in

speech. The 't_cl' segment was for closure of the stop and is also incorrectly labelled in figure 3.2, but there were also occasions when there was no completely silent portion of the stop, so some sound would be present when selecting these units. The 'sp' marker is a segment added by the multisyn engine, based on the possibility there are gaps between words, but at synthesis these very short segments would be assimilated or ignored as there are no pauses here, which had been determined during the alignment (Clark et al., 2004). In some samples including 'so', an additional 'silence' segment had been inserted, though I could not find a reason for this.

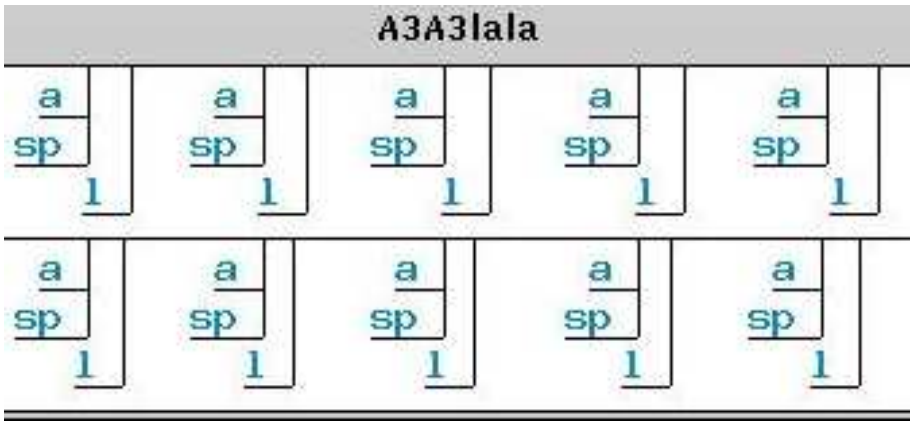

Figure 3.1: Original (top) and actual (bottom) labels for a sequence

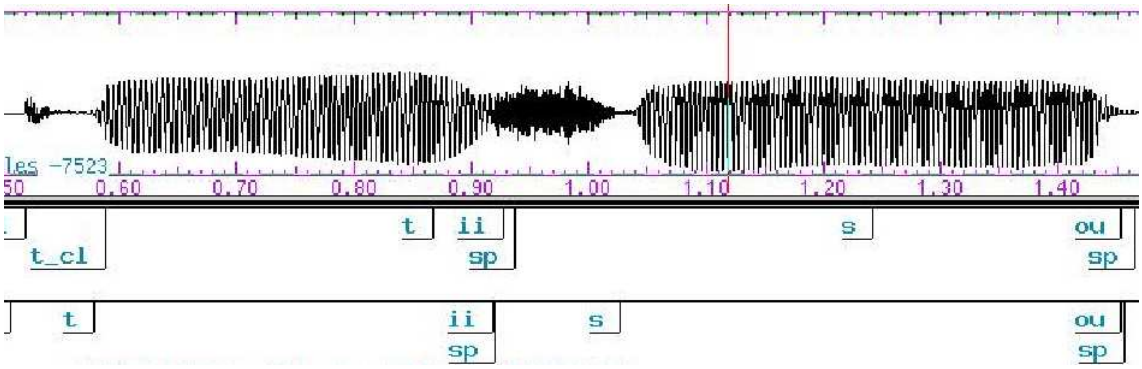

Figure 3.2: Original (top) and actual (bottom) labels for a sequence, aligned with the waveform

Labels were hand edited, but some room for further improvement was left to test how this would affect quality. It would be worth knowing that hand-editing need not be meticulous. Though phones were being labelled, the multisyn engine concatenates diphones, so the exact position of some of the labels might not be as influential as in the phone-concatenating clunit synthesiser. As described in chapter 2, ideally, the fact that duration is known in advance should be used

	Speech	Singing	Values used
Upper limit	295	261.63Hz (C4)	270
Lower limit	120	195.99(G3)	180

Table 3.5: Fundamental frequency range used in the subset design

to help with alignment, to acknowledge that singing and not speech is being recognised (Yoram, 1999).

An alternative to using techniques provided by Festival, if one were very confident of the duration of individual segments, would be to make a simple script to create label files without any reference to the recordings themselves. This is probably a slightly optimistic scenario, and is one based on the time spent editing incorrect label files, but would be more appropriate than using a speech-based alignment technique if labelling a large database. Another approach would be to hand label a selection of the database and use this to train a recogniser.

3.3 Building

3.3.1 Pitch/Fundamental Frequency

The same procedure was used for pitchmarking as for the clunit synthesiser, the only difference being in the range, which now fell within that of normal speech as illustrated in table 3.5. Overall, pitchmarking was better than in the first design, since there was much less linguistic information to take into account (only syllables, one of which was fully voiced). I examined each file and found that all words on A3 and almost all on B3 were accurately pitchmarked, with one peak per period and no missed segments. In contrast, G3 and C4 tended to have segments which had not been pitchmarked. In the clunit design, bad pitchmarking could not be marked as the only cause of artefacts, but if there was noticeably poorer quality synthesis for G3 and C4, this could be traced to pitchmarking here. However, the documentation did state that any un-pitchmarked section would be discarded, which would avoid problems at synthesis (Black et al., 2002).

3.3.2 Further building

Utterances were generated with reference to the .data file and lexicon, after which Festival was recompiled. For each utterance in the prompt file, the aligned phones

of the corrected labels were merged with the predicted segments, and any insertions (none) and deletions (here, of pauses) were also dealt with (Clark et al., 2004). Utterances were rebuilt and Festival recompiled each time anything from earlier was changed. Two steps designed to fix duration errors and one to add pauses to the database were not used, since their reference to speech properties would corrupt the editing carried out for singing.

Ideally, all join costs would have been calculated and stored during the building process, but there were too many to store effectively so these were calculated at run-time, equally weighting the Euclidean distances between MFCCs, fundamental frequency and spectral energy of joins (Clark et al., 2004). For this, a script was run to store fundamental frequency values, for which the default values for female speakers were also changed to the values above in table 3.5, and MFCCs were normalised and combined with these tracks. Finally, LPC coefficients were generated for use at synthesis.

After starting Festival and setting it to access the correct voice, text/lyrics were presented to be synthesised through XML files with the command (tts 'filename.xml' 'singing) on the Festival command line. (Using '(SayText 'la ti so')' accessed notes according to speech settings.) A problem solved at this point was that Festival expected sound files sampled at 16kHz, which was changed to accept the 48kHz sampling rate used here. For the musical information in the XML files to be synthesised (other than just a sequence of the correct words), the default target costs were changed to make units with incorrect pitch or duration to receive the highest costs, after which Festival was recompiled. Table 3.6 lists these changes and shows how all other factor weights were reduced. There was little linguistic variation in the database which would make all the speech related target costs relevant, so weights were kept at these levels.

A further feature was added by Dr. Rob Clark so that target costs could be given a greater overall weighting than the join costs, given the possibility that a good join cost (e.g. a sequence of words on the same note) would have a greater influence than a target where words were on different notes. However, the voice was already able to synthesise required pitches and words just using the target cost values in table 3.6, so this feature was not ultimately required. At synthesis time, the best path was chosen by Viterbi search where the sequence with the lowest overall target and join cost was selected and concatenated for synthesis. In a speech synthesiser the join costs would usually be stored in advance, but in this case they were calculated during the search, which took a few seconds.

Target Cost	Speech	Singing
Pitch (from XML)	-	50
Duration (from XML)	-	50
Amplitude accent	20	5
Stress	10	5
Punctuation	10	5
Part of speech	6	5
Position in syllable	5	5
Position in word	5	5
Position in phrase	4	5
Left context	4	4
Right context	3	3
Bad duration	-	2

Table 3.6: Target costs for speech and singing

The singing mode specified the pitch and duration of each word in the XML file and created a pitch track based on this. In later tests, the feature of prosodic modification was added, where this track would be used to modify selected units so that they precisely matched the specification of the XML files. In initial tests however, units were selected simply with reference to their labels in the database. These first tests were already of very acceptable quality, demonstrating that the recordings had consistent acoustic qualities, as they resulted in smooth transitions at concatenation points. That this was also achieving perceptually good durations illustrates that for those samples chosen, the duration was also consistent. Where duration was noticeably incorrect, the original unit had been taken from a phrase final position.

This illustrates the effect of altered target cost weightings; if using speech weightings, a phrase final segment would receive a high target cost if suggested for a phrase-central position, and would also have a high join cost to the next segment, since it was recorded in the context of note/silence. Using the singing weightings however, a phrase-final note could be selected for synthesis in the middle of a phrase, which I had not considered on first noticing the length of the last notes. The overall quality of the synthesis was still acceptable, but this did illustrate that an error in duration was just as noticeable as an error in pitch.

The next section describes the improvements and evaluations carried out on the voice. This includes correcting the XML files so that the one-second long final

notes were marked as such, investigating improving the labelling and pitch-marking, and applying prosodic modification to units rather than selecting them according to their labels.

3.4 Evaluation and improvements

I designed a set of test sequences to evaluate the voice with and use as comparison against any improvements, firstly with Festival selecting units just according to the labels in the database, to see what quality could be achieved before any prosodic modification was carried out on the units. I did not arrange formal tests but some considerations of what might need to go into such an evaluation are discussed in chapter 4. The test sequences used are listed below and were designed to judge how well the engine could synthesise sequences it already had, and how well it could synthesise a truly arbitrary sequence of notes. Areas of particular interest included which units had been selected, if the synthesis sounded natural, and if there was noticeably worse quality on specific words/notes.

Test 1 Sequences which the synthesiser has recorded:
- G3 lalalala, A3 lalalala, B3 sosososo, C4 titititi

Test 2 Unrecorded sequences of more than two words:
- G3 latisotila, A3 latisotila, B3 solatilaso, C4 tilasolati

Test 3 Unrecorded sequences of more than two notes:
- la GABCG, ti GABCG, so GABCG

Test 4 An unrecorded sequence, more than two notes and two words:
- G3 la, A3 ti, B3 la, C4 so, B3 ti, A3 so, G3 la

Test 5 The melody of 'Merrily we role along', which fitted the available notes.

Of the tests run, the synthesis of phrases which the synthesiser already had recorded were, unsurprisingly, the best quality, even though there was still an inevitable join cost happening, since the sequences were four notes instead of the recorded nine. For comparison, I ran some nine note sequences which exactly matched originals. These were replicas of the recordings, with a join cost of zero throughout.

In test 1, there were voice quality changes at concatenation points for sequences on A3 and G3 which shows that even when aiming to sing notes the same way all the time, the waveform of one sample will never exactly match that of another.

Encouragingly however, the sequences on B3 and C4 were particularly natural, and I only found their concatenation points after examining which units had been selected and spectrograms of the synthesis. As in the clunit synthesiser, there did not appear to be an influence of the 'badly' pitchmarked originals when synthesising words on G3 and C4.

In test 3, synthesis of 'ti' on different notes had the most artefacts, showing that it is not a simple case that one note/word is better than another overall (in test 1, 'ti' was the most natural). Overall, synthesis ranged from being very natural to having several artefacts, though none as noticeable as those found with the clunit synthesiser. Perceptually, artefacts were what might be expected if listening to a bad recording, or a singer who was singing at the 'break' between their lower and higher singing registers, which can cause a change in voice quality. No incorrect note was selected, showing that the target cost weightings for duration and fundamental frequency had been enough to make sure that musical structure was more important than other costs.

The samples which used multiple words (tests 2, 4 and 5) had the greatest number of artefacts, but in these sequences the artefacts were noticeable for being the exception to otherwise natural quality, rather than a moment of natural quality being the exception to the rule. A disadvantage of using so few notes was that not many melodies could be synthesised, but a familiar tune 'Merrily we role along' was within the database's range, and although there were enough artefacts to identify it as obviously synthesised, it would be acceptable for teaching someone this melody. This example is also available online at http://www.ling.ed.ac.uk/ s0454934/.

In general, though the output would probably still be recognised as synthesised, it was clear that the recordings had at least been made by a real singer, in contrast to the existing example recordings available through the Festival website where a speech diphone database had been modified to produce singing. The only genuine error was when duration was inaccurate because a phrase final note had been selected for the middle of a phrase. Before altering this in the database however, I decided to investigate improving the labelling and pitchmarking so that I would have direct comparisons with the original tests.

3.4.1 Editing labelling and pitchmarking

High quality annotation of the database is vital for good quality unit selection synthesis, but in this case, since joins were mid-vowel/phone, artefacts occur-

ring here might not be improved through editing the phone labels. This is different to speech, where vowels are much shorter, so incorrect label boundaries can potentially cause very poor quality synthesis. Because I had intentionally left some room for improvement in the labels, and to conduct a second pass check of annotation is a good principle, I re-checked and edited the labels in more detail and re-synthesised the test sentences.

However, in these (limited) tests, there were no perceptual differences to the original tests and in some cases the same artefacts occurred. This shows that the original labelling was sufficiently accurate (and there were no large errors which when corrected would produce large gains in quality), or, could indicate that the files chosen still had the best overall costs. Editing labelling improves quality with diminishing returns, and the result here shows that a second pass check could be skipped if necessary (Black and Lenzo, 2000).

I had expected to spend time editing pitchmarking, but on reflection, there did not seem to be the need to. The 'worst' pitchmarking was where pitch periods were missed and this only happened in recordings on G3 and C4, but synthesis of words on these notes was not worse quality than words on A3 and B3, even when specifically testing these intervals. The majority of words on G3 and C4 had some missed pitchmarks, so it is not possible that the synthesised tests were just using better samples of these notes. Even well pitchmarked sections had artefacts, so I did not hand edit pitchmarks, since artefacts were not just being caused by bad pitchmarking.

Although the database was small, overall quality in the tests was still better than expected, and the only real errors involved duration, which was corrected as described in the next section.

3.4.2 Editing Duration

Although correct notes and words could be synthesised, on closer inspection there were several errors in how the engine processed the XML files, since in the utterance files, each unit was listed as 0.25 seconds instead of the 0.5 second label given in the database. I considered that, if the engine found a recording that was not exactly 0.5 seconds, it may instead label it as the next lowest value. This theory did not hold, since even the longer, phrase final notes were being processed as 0.25 seconds, and at this stage the multisyn engine was just selecting units according to their labels and not analysing them. At this point, I corrected

the database so that phrase final notes were labelled as a full second, but this just resulted in 1.0 second notes being processed as 0.5 seconds.

On returning to the code for the singing mode and checking more of the example XML files provided in the Festival documentation, one final correction was made. A beats per minute (BPM) variable (not present in all the example files provided) was added to all the XML files, i.e. those in the database *and* files given as tests for synthesis, which corrected the processing error. Unfortunately, it was not possible to change the speed of the synthesis simply by changing the BPM value, possibly because all files in the database were listed as 60BPM, though this would be a useful way to change tempo.

The database was not designed to produce different durations, but because of the low target cost weighting for position in a phrase, notes could be synthesised at either length. There were also no more duration errors when synthesising a sequence of 0.5 second notes. However, it could not be guaranteed that requesting a 1.0 second note in the middle of a sequence would result in a note perceptually twice the value of 0.5 second notes around it, because join positions were just made with reference to central points in the labels and not to actual durations of the units. This is demonstrated in another synthesis example of 'Merrily we roll along', where second notes are included at the appropriate points but the effect is not of the correct rhythm (see website for example). Some other examples of using different durations were perceptually better though, which was more than expected for a database not designed to produce different durations.

3.4.3 Prosodic Modification

The tests run so far involved selecting units without reference to their actual pitch and duration, but units could also be prosodically modified to exactly fit pitch track of the target XML file. This would be applied to the sequence selected to be synthesised. First tests using this feature had very poor quality, which was surprising, given the good quality of the synthesis using just labels and that the same units were being selected. After much investigation, the error was traced to a confusing description of how the singing-mode handled fundamental frequency. There are several formats available to encode fundamental frequency in music, as listed in table 3.7. (The numbers missing in the MIDI notation are, of course, those notes not found in the key of G.)

Musical Notation	Value in Hertz	MIDI Notation
G3	195.99	55
A3	220	57
B3	246.94	59
C4	261.63	60
D4	293.66	62
E4	329.63	64
F#4	369.99	66
G4	391.99	67
A4	440	69

Table 3.7: Methods of describing fundamental frequency

For non-musicians, the concept that numbering restarts on C and not A can be confusing, so having reference to a note's value in Hertz or MIDI notation is useful, and was included in the documentation for the singing mode (Richmond, 2002). The error/confusion in the script was a reference to pitches being calculated in relation to MIDI note 69, using the convention that "A5 = 440". This was followed by a comment that "some people call this A3". As can be seen from table 3.7, 440Hz is more usually called A4. Given that all notes were being calculated with reference to A5 equalling 440Hz, it became unsurprising that the synthesis quality was so poor, as each note was being altered by an octave. This was shown in the utterance relation files, where any A3 note was referred to as being 110Hz, the equivalent of A2.

The value of A5 was corrected in the script to be 880Hz, which resulted in each note being assigned its correct Hz value. This was sufficient for my needs, but this part of the script should ideally be rewritten so that correlation between values is correct - the notes are calculated with reference to MIDI note 69/440Hz, so changing this to believe that note 69 is actually 880Hz may cause future problems if a larger scale is used. An initial suspicion that I had actually recorded everything an octave lower than I thought was quashed on checking the fundamental frequency in Praat.

The intention of using the prosodic modification feature was to synthesise additional notes and durations without needing to record them. However, it soon became clear that the error described above was not the sole cause of the low quality output when using prosodic modification. Although any note and duration could be synthesised, they all had the 'buzzy' quality usually associated with rule-based synthesis at some point in each note, so the synthesis would

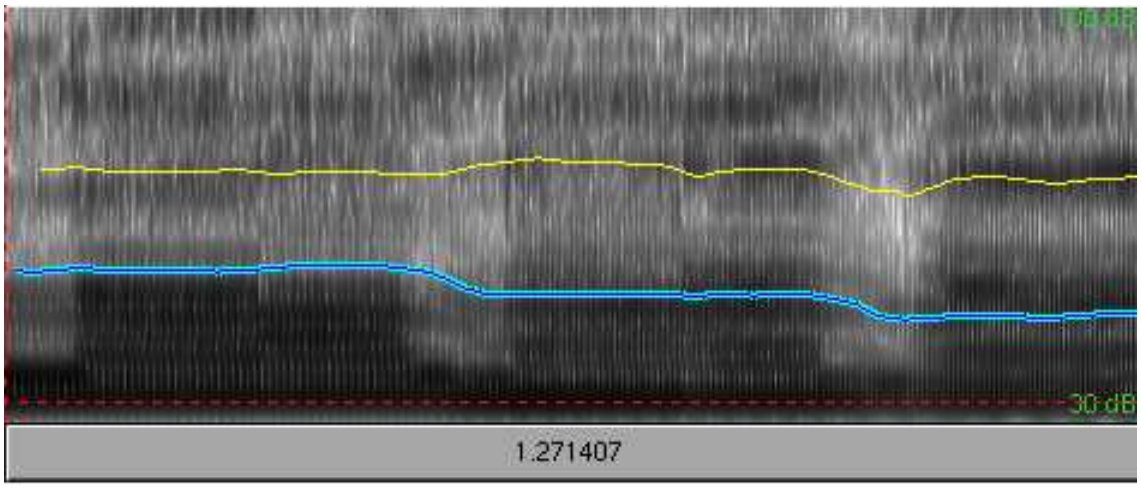

Figure 3.3: First three notes from 'Merrily we roll along', sung as 'la la la'. Intensity is indicated in higher line, fundamental frequency in the lower.

not be mistaken for a real singer. This was the case even when synthesising sequences the database contained. A possible explanation is that the prosodic modification was removing the natural 'flutter' present in a real voice, adjusting it so that all portions of the note were on the exact pitch note. As a result, even though most of the original note was on pitch, some sections which may have wavered slightly (though acceptable perceptually) were adjusted.

Another explanation for the unnatural quality could be that there were no facilities for ensuring smooth transitions between notes, as seen when comparing figures 3.3 (synthesis without prosodic modification) and 3.4 (with prosodic modification). In figure 3.3, where units were selected just according to their labels, joins are smooth and continuous while perceptually still distinct notes (this is the earlier example of 'Merrily we roll along'). By contrast, figure 3.4 shows unnaturally abrupt changes in fundamental frequency at each note, and the obvious artefact where fundamental frequency drops below the intended pitch, possibly due to lengthening a section of the note so it would fit the required duration. Intensity is also not consistent in figure 3.4. Duration, however, was now perceptually longer than heard in the synthesis without using prosodic modification, showing that the recorded notes were not 100% accurate, and that every unit will have been modified for duration, which may also have contributed to the unnaturalness. These two examples can also be listened to on the website mentioned earlier.

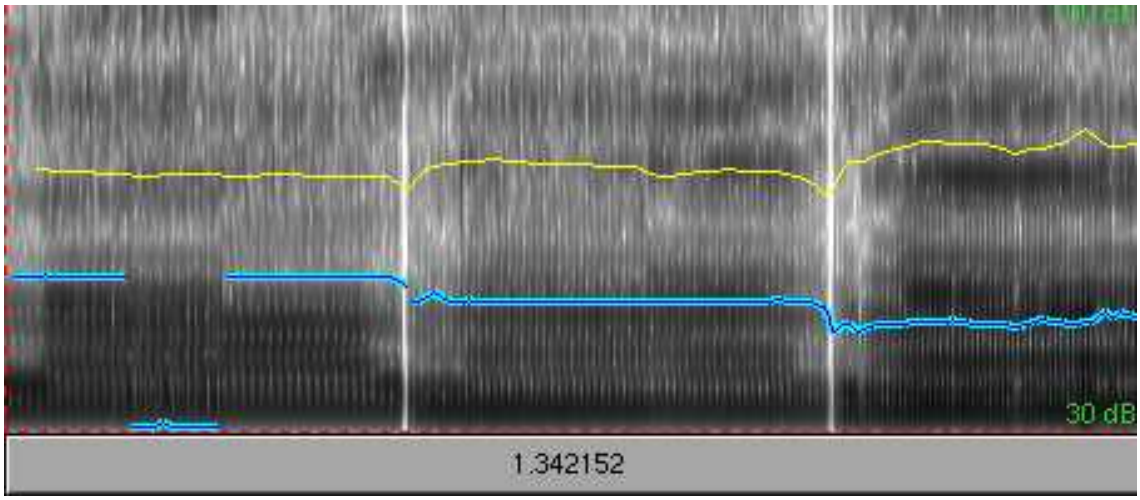

Figure 3.4: First three notes from 'Merrily we roll along', sung as 'la la la', where prosodic modification had been applied. Intensity and fundamental frequency are indicated as previously described.

Although it was now possible to accurately synthesise any pitch at any length, the natural quality of the synthesis was lost when prosodic modification was applied.

CHAPTER 4

Discussion and Conclusions

The achievements and shortcomings of the two singing synthesisers made in this project are discussed below in the context of whether unit selection was an appropriate technique to use. I also suggest how the present designs could be extended, and discuss how larger scale singing synthesisers could be built based on unit selection synthesis.

4.1 Evaluation of the synthesisers

The voices used different techniques and designs, as summarised in table 4.1. The intended application was a tool for arrangers and for learning music from, so the accuracy and naturalness were the most important qualities required. The multisyn engine provided the most acceptable quality when just selecting units according to their labels, and the least when these units were prosodically modified.

	Clunit Synthesiser	Multisyn Engine	Prosodic Modification
Synthesis method	Cluster units	Target costs	Target costs
Pitch range	one octave	one octave	unlimited
Vocabulary	'one' - 'eight'	'la' 'ti' 'so'	'la' 'ti' 'so'
Duration	1.0 second notes	0.5 and 1.0 second notes	Any duration

Table 4.1: Differences between the synthesisers

Though comparisons between the clunit and multisyn engine voices may be fairer if both used the same (simple) database, the clunit technique used phones as the unit of selection and so was more susceptible to artefacts at concatenation points and errors in labelling, compared to the diphone-selecting multisyn engine voice. This was demonstrated when over 30% of the sequences synthesised with the clunit voice which were replicas of material in its database had some artefact in them. Even if quality could be improved by using the (linguistically) simpler vocabulary in the clunit voice, more time would need to be spent in ensuring accurate labelling than potentially required for the multisyn engine voice. The clunit synthesiser remained a speech synthesiser in essence, so would need its architecture modified before it could produce the musical range afforded by the multisyn engine voice.

The second voice had an improved database design; its pitch range was easier to sing with consistent voice quality, the consonant-vowel vocabulary involved fewer concatenation points, and units of selection were diphones. These differences resulted in less spectral mismatch at joins, fewer necessary joins overall and acoustically stable join positions. The architecture of the system allowed for words to be synthesised on specific notes (even of different durations), by exploiting that target cost weightings could be set to make duration and fundamental frequency the most important factors in selecting units. In design, the multisyn voice was a singing rather than speech synthesiser, since lyrics could be synthesised according to a musical format, so the overall approach was more appropriate to the task.

The second design, because a limited musical range was used, did not address issues of voice quality when singing large intervals and how to simplify singing unusual intervals. There would still have been changes in voice quality for large intervals, but the question was whether these would be less noticeable with shorter note length and recording sequences multiple times in the same conditions. When prosodic modification was applied to the multisyn engine the resulting synthesis occasionally sounded as if it had not come from a real singer, due to how fundamental frequency and duration were manipulated. The only advantages in using this feature were that duration and to a lesser extent, pitch, were accurate in a way that neither the clunit nor original multisyn engine voices could have achieved.

4.1.1 Extending the current database design

None of the voices would be suitable for synthesising a performance of singing, but the intended applications were to assist arrangers and composers or a teaching tool, for which characterstics such as a singer's formant and vibrato are not required. The obvious restriction of the current design is its limited range of notes and durations. Vocabulary size is less significant; only a completely unrestricted vocabulary would be truly useful for a teaching tool, since any word which could not be synthesised would have to be learnt later. Given this, the vocabulary could contain just one syllable such as 'la' (consonant-vowel so that note/word boundaries could be kept distinct), which though simple, saves much recording time and building time.

Alternatively, as per the clunit database design, each note could be synthesised as a specific word or syllable. Although again simple, this is a useful approach for a learning tool, since each note is distinct. Arbitrary syllables could be used, a known sequence such as 'do ray me' etc., or even the 'do ray me' scale on each key note (repeating the sequences required for one note on all eight notes) so appropriate note names could always be used.

The most relevant limitations of the existing design are in pitch range and note lengths. The first extension of the design would be to ensure that a larger pitch range will produce good enough quality, and to investigate methods of improving this if it did not (of which an option is to improve how prosodic modification of units is applied). The overriding issue in extending any aspect of the design is the number of recordings these extensions require, as illustrated in table 4.2, which is based on the nine note sequence described in chapter 3 and also shows the effect of increased vocabulary size. (12 notes are required per octave range, to include all semitones.) The number of recordings is calculated in this way, where number of notes = n :

$12 + (n\text{-}1 \times 6) + (((n^2 - n) / 2) \times 9)$
12 = sequences required for the final note
6 = sequences where there is 'zero' interval
9 = sequences required for all other notes

To avoid unwieldy recording requirements, some modification of existing units is required, for example by using praat or a similar tool to modify the pitch of

Notes	8	9	10	11	12	13	14	15	16	24
Three words	306	384	471	567	660	786	909	1041	1182	2850
One word	36	45	55	66	78	91	105	120	136	300

Table 4.2: Number of recordings required with the current design (three word vocabulary), as pitch range increases

original recordings, if this could be achieved without degrading quality. This was one of the first tests I planned to carry out had there been time, along with labelling a section of silence as a 'rest' beat, so that pauses could be included. If global changes in pitch sounded natural and could be implemented at run time, this may be a method of applying automatic key changes to synthesis.

An alternative way to reduce the number of separate sequences required (though recording time would still be long) is to redesign the database. The design used in chapter 3 did ensure all contexts were recorded, but the same is possible by designing appropriate melodies, or extracting them from existing recordings (Yoram, 1999; Uneson, 2002a). In terms of vocabulary, many of the same issues apply as in speech synthesis regarding how to obtain sufficient coverage of all diphones, and the trade-off between using a smaller sized unit (which would make recording all contexts easier) and keeping the number of concatenation points low (Clark et al., 2004). Using the experience of database design in speech synthesis and applying it to music, an option is to analyse musical structure itself, determine which intervals are common and rare (the unusual intervals) and record sample numbers appropriately.

Regarding duration, the issue is how to include any note length and synthesise it accurately. The multisyn engine was not designed to have two durations which is why these were not perceptually correct. Some possibilities are discussed below; in all cases it must be considered how many extra recordings would be required. A first problem is that the nine-note sequence design cannot accommodate very long notes because it would eventually become impossible to maintain breath support, but shortening the phrase would reduce sample numbers.

The concatenative approach could be exploited here to make a required length by joining notes, though this would risk a buzzy quality from any perceptual (large) periodic signal created as a result. To avoid this, only very long notes could be made in this way, so that perhaps just one concatenation point between two long notes would ever be required, instead of multiple repetitions of very short notes.

To use this approach, the synthesiser would need to understand how long a unit is originally and as a diphone. Phone labels could be used to judge where notes start, taking the beginning of the vowel as a downbeat/zero point and analysing durations of consonants so that these would be correct in relation to the beat (Akagi, 2005). If still assuming a vocabulary of just one syllable, it may be useful to use a voiceless initial consonant so that the downbeat can be easily judged in labelling.

A potentially simpler approach to the above extensions to pitch and duration is to improve prosodic modification so the natural quality of the recordings is preserved. To ensure no unit is modified to the extent that quality is affected, a large proportion of original recordings would still be required, or a limit on how much a unit can be modified (Yoram, 1999). Of course, this would be limited by the range of the singer in question - beyond this, all notes would need to be modified. Applying global prosodic modification would also be a much more practical method of producing different rates of singing than recording them. Given the improvement in quality after correcting how fundamental frequency was encoded, it is possible that there were other errors I did not find which were causing poor quality. It follows that large improvements could be made in synthesis quality, capabilities, and required storage space by experimenting with how units were modified, which could apply the knowledge of rule-based synthesis.

4.1.2 Further improvements

Improvements discussed in this section relate to the building procedure (recording, labelling, pitchmarking and selecting units for synthesis) and, like the section above, also apply to creating a larger scale synthesiser.

Recordings

The demands of recording any interval in a large musical range for long stretches of time has implications for who can make these recordings. If using a professional singer, vibrato may need to be taken into account, and asking for a 'neutral' singing voice is a strange request to make of any singer. Even singing in a very relaxed manner is more tiring after a long period of time than speaking in a relaxed manner, so some way to handle changes in voice quality caused by taking breaks would be necessary. The capabilities of the singer determine the

database that can be recorded, in terms of their accuracy, pitch range, and voice quality across their range. A person with perfect pitch might find some aspects of the recordings easier than others, but the issue of range still applies.

Labelling

The forced alignment technique used in this project was designed for speech, and every file had to be checked and almost all corrected. As described in chapter 2, the most appropriate approach would be to use the duration information in the XML files to direct the alignment, or to automatically generate labels based on assumptions of where segments should be. Alternatively, a recogniser could be designed or trained to recognise singing, so that the differences between singing and speech are used to help and not hinder automatic labelling techniques. Checking the segmentation is a good principle to follow whichever technique is used.

Pitchmarking

The most accurate way to pitchmark the recordings would be using an EGG signal, if this were practical for very long recording sessions (Black and Lenzo, 2000). The effectiveness of the autocorrelation technique used here appeared to depend on the range being sung: there were missing pitchmarks for most of the notes above (and some below) D4 in the octave range, and the highest and lowest notes in the smaller design. However, these problems could not always be explained with reference to pitch or syllable used, and neither could artefacts in the synthesis. A more appropriate approach to ensure accuracy may be to assign a specific fundamental frequency range to each segment before running the autocorrelation function. Again, using a simple linguistic structure simplifies the problem somewhat, particularly if the initial consonant were voiced, as voiced segments are easier to pitchmark (Yoram, 1999). A voiced stop is therefore a compromise to my earlier statement that a voiceless consonant would be preferable for making labelling easier.

Synthesis

Both synthesisers should be tested with the same database design to compare the synthesis methods fairly, but there still were significant advantages in using the multisyn engine. Most importantly, fundamental frequency and duration (i.e. a

note) and any other factor could be specified as the most important weight in the target cost, instead of relying on position in a cluster. Even if using the clunit technique, for example if there were a specific word on each note, other factors such as position in phrase and amplitude might not be as reliably controlled as when these features are included in a target cost.

When the vocabulary is more linguistically complex, weights in the target cost could be experimented with to determine what factors are perceptually important to listeners. For example, in singing (and speaking), particularly in untrained singers, there is a tendency for intensity to drop as breath support runs out. Therefore, a segment from towards the end of a phrase may not be appropriate for synthesis at the start, which could be imposed by increasing the weighting for 'position in phrase'. Drops in intensity could also be handled by applying power normalisation to the recordings.

Although the current design processes musical structure correctly, it would be appropriate to test whether this could be handled elsewhere in the architecture of the synthesiser. This would leave the target cost free to determine such issues as position in phrase and voice quality, just with reference to units which were already musically correct, which might also save searching time in a larger database. A backing off principle (which neither synthesiser used in this project) would also be easier to implement in this case, as there would be no risk of an alternative note having the wrong pitch or duration.

Formal Evaluation

I did not carry out formal tests since it was clear that the multisyn engine voice had the best quality and capabilities. There are also no established evaluation procedures for singing synthesisers which could be referred to in comparing performance (Rodet, 2002). However, perceptual experiments could be used to test altering target cost weightings, for example through listeners judging which example is more natural, or which they prefer (Yoram, 1999; Ternström, 2004). If using a test which rated how much the synthesiser sounded like a singing voice, the current voices may not score well, as they lack the performance level characteristics of singing (Akagi, 2005).

Evaluation should relate to the intended application, so in this case subjects who might learn from singing synthesisers, for example choristers, should be chosen. Arrangers (in acapella singing) would be synthesising multiple voice parts, so

this would need to be possible before evaluations could be made. A method of testing the synthesis itself is to have a real singer sing a phrase and then compare this to a synthesised version. People could be used to compare quality, or cepstral differences between the original and synthesised versions, which could also be used as a method to ensure the best units are selected for synthesis (Rodet, 2002). Overall, the specific requirements of arrangers, composers and learners would need to be investigated further, preferably before extending the design, as the motivation for creating the current synthesisers came from my own and anecdotal experience.

Applications

The intended applications for this project were: a voice from which people could learn songs, and a tool for arrangers and composers in creating and storing melodies. For the first application the multisyn voice was able to synthesise with what I felt were the main requirements: natural overall quality and precision of notes, making it intuitively easier to tune to and learn from than 'buzzy' synthesis. Lyrics, however, would always need to be learnt separately - a problem not encountered by rule-based synthesisers. For arrangers and composers, once the musical range were extended, the multisyn voice would also be capable of synthesising drafts of melodies in (presently) real time, and could be used by those wishing to synthesise their own voice. If storage space and consistency of quality were the most important issues however, rule-based synthesisers would be more appropriate. For all three intended applications, a necessary improvement is implementing a user interface to communicate the musical structure to the synthesiser, for example in MIDI format, rather than using time consuming and error prone XML files.

As mentioned above, the current design would be inappropriate for an arranger since only one melody can be synthesised at a time, though there are some simple methods of playing several melodies at once which would remedy this providing duration (i.e. prosodic modification) was accurate (Black et al., 2002). More appropriately, this could be incorporated into the synthesiser itself. To teach a relevant harmony part, the amplitude of each voice could be altered, though it may still be difficult to distinguishing which part should be listened to if all are the same quality. Alternatively, databases could be recorded for separate singers so each part could be synthesised separately, which could also provide a much larger pitch range than that of a single singer.

If Just Intonation were required, for example in creating a synthesiser for the barbershop style, the whole design would need to be altered. This may be too high-level a characteristic to consider, equivalent of performance level synthesis, but some thoughts on this follow here based on my experience as an amateur arranger in this style. If using the principle that all contexts are required (and not considering vocabulary or duration), a quartet would need to sing every necessary chord in every context, so that tuning were appropriate throughout. A simpler (and feasible) alternative would be to record individual chords in isolation, designing a synthesiser where the harmony and not the melody is of interest. To take this idea further, the specific chords of the barbershop style could be recorded, or only those which are physically possible to sing, which would be useful tool for arrangers to learn from.

4.1.3 Final thoughts

Singing focuses on the areas that are particularly difficult and sometimes avoided in speech synthesis, such as expressing emotion. To an extent, as demonstrated in the database designs used here, these issues can also be quietly ignored in singing voice synthesis when performance level synthesis is not required. Two methods of unit selection synthesis were used in this project, and the multisyn engine synthesiser achieved the goal of synthesising arbitrary melodies which others could learn from. Its best features, and those I would keep if extending the design, are listed below.

- The original recordings had as consistent a voice quality as possible.
- Within the range tested, the design avoided unusual intervals.
- Musical information was encoded in the database.
- Using the target cost allowed the correct notes to be selected, and gives the potential for improving the quality of synthesis by further adapting these costs to reflect the characteristics of singing.
- Overall quality was natural, clearly from a recording of a real singer, and without the 'tinny' quality of an over-modified recording or rule-based synthesiser.

Particular difficulties and limitations are listed below, in relation to all three synthesis methods.

- The database required for unit selection singing synthesis is potentially much larger than that required for speech.
- The recording process places unusual demands on the singer, and the current design may not be appropriate for larger scale databases (i.e. intervals, range, time required).
- Either vocabulary or musical range must be kept simple to keep recording requirements feasible.
- Automatic alignment procedures must be adapted to singing.
- Applying prosodic modification currently degrades quality.

The biggest improvement to the current design would be to include a larger musical range, either through more recordings, or improving the prosodic modification. Without prosodic modification I would use just one syllable (e.g. 'da') so that recording time could be spent on as many musical contexts as possible. With it, synthesis would be consistent, the only restriction would be in vocabulary, and fewer recordings would be required.

Unit selection synthesis techniques can be applied to singing to produce a high level of naturalness, given that recordings of real singers are used. The quality of the synthesis in this project would be appropriate for the applications of interest, but the more aspects of singing are taken into account the harder it becomes to record all necessary contexts. Therefore, some form of prosodic modification must be applied to the units. To apply this appropriately the features of singing need to be further analysed and understood, which brings us back to rule-based approaches. Overall it should be remembered that people have a separate 'speaking voice' and 'singing voice', so the analysis and synthesis techniques of one will not always be appropriate for the other.

References

Akagi, M. (2005). Perception of 'singing-ness' and its application to singing voice synthesis. Presentation given at the Centre for Research in Speech Technology, University of Edinburgh.

Black, A. and Lenzo, K. (2000). Limited domain synthesis. In *ICSLP*.

Black, A., Taylor, P., and Caley, R. (2002). The Festival speech synthesis system, system documentation, edition 1.4 for Festival version 1.4.3. Online at http://festvox.org/docs/manual-1.4.3/festival_toc.html.

Clark, R. (2005). Speech processing 2. Course taught in the Department of Theoretical and Applied Linguistics, The University of Edinburgh.

Clark, R., Richmond, K., and King, S. (2004). Festival 2 – build your own general purpose unit selection speech synthesiser. In *Proc. 5th ISCA workshop on speech synthesis*.

Kalin, G. (2005). Formant frequency adjustment in barbershop quartet singing. Master's thesis, Department of Speech, Music and Hearing, KTH, Sweden.

Macon, M. (1996). *Speech Synthesis based on sinusoidal modeling*. PhD thesis, Electrical Engineering, Georgia Institute of Technology.

Mazzoni, D. (2001). Festival singing mode. *Speech: Phonetics, Prosody, Perception and Synthesis*, pages 11–752.

Richmond, K. (2002). Code documentation for the Multisyn Engine in Festival. Available via http://festvox.org.

Rodet, X. (2002). Synthesis and processing of the singing voice. In *1st IEEE Benelux Workshop on Model based Processing and Coding of Audio*, pages 99–108, Leuven, Belgium.

Saitou, T., Unoki, M., and Akagi, M. (2002). Extraction of F0 dynamic characteristics and development of F0 control model in singing voice. In *Proceedings of the 2002 International Conference on Auditory Display*, Kyoto, Japan.

Sundberg, J. (1999). Level and center frequency of the singer's formant. Speech, Music and Hearing Quarterly Progress and Status Report Volume 40, KTH, Sweden.

Sundberg, J. and Hagerman, B. (1980). Fundamental frequency adjustment in barbershop singing. *Journal of Research in Singing*, 4:3–17.

Szabo, B. (1976). Theory of barbershop harmony. SPEBSQSA, USA.

Ternström, S. (2004). Singing synthesis - what's the buzz? Presentation given at the 2nd International Conference on the Acoustics and Physiology of Singing. Online at http://www.ncvs.org/pas/2004/pres.htm.

Uneson, M. (2002a). Burcas - a simple concatenation-based midi-to-singing voice synthesis system for Swedish. Master's thesis, Lund University.

Uneson, M. (2002b). Outline of Burcas - a simple concatenation-based MIDI-to-singing voice synthesis system. *Fonetik*, 44:133–136.

Yoram, M. (1999). *High Quality Singing Synthesis using the Selection-based Synthesis Scheme*. PhD thesis, University of Tokyo.