Emotionally expressive song synthesis using formants and syllables

DD143X

Degree Project in Computer Science, First Cycle
Group 19
Dexter Gramfors, 911012-3735, dexter@kth.se
Andreas Johansson, 920702-3897, andjoh3@kth.se
Supervisor: Roberto Bresin
CSC, KTH

March 2014

Abstract

Speech synthesis is an area of computer science with many practical uses, such as enabling people with visual impairments to take part of text and to provide more human-like feedback from information systems. A similar area of research is text-to-song, where systems comparable to those used in text-to-speech provide mappings from text to melodic units of song. This paper discusses how a text-to-song algorithm can be developed and what parameters affect what emotion is communicated. Fifty participants listened to music generated with our algorithm. Results show that tempo and mode both heavily account for what emotion is communicated; a melody performed with a tempo of 250 bpm was perceived as significantly more happy than a performance with a tempo of 120 bpm, and a melody in major tonality was perceived as significantly more happy than a melody in minor tonality. Combined, these parameters gave even more significant results. A fast tempo combined with major tonality produced a performance that was perceived as even more happy. The opposite was observed when a slow tempo was combined with minor tonality. When a fast tempo was combined with a minor tonality the average answer was neutral with answers distributed over the whole spectrum from sad to happy. A slow tempo combined with a major tonality gave almost identical results. We concluded that generating emotionally expressive song with the use of an algorithm is definitely possible, but that the methodology can be improved in order to convey emotions even more clearly.

Contents

1 Introduction **3**

2 Problem statement **3**

3 Background **4**
3.1 Music and emotions 4
3.2 Phonetics 7
3.2.1 Formants 7
3.2.2 Using formant-pitch matching 7
3.2.3 Syllables 8
3.3 Text-to-speech synthesis 8
3.4 Contemporary text-to-song systems 11
3.4.1 Vocaloid 12
3.4.2 Oddcast Text to Sing 12
3.4.3 Let them sing it for you 13
3.4.4 Melobytes 14

4 Methodology **15**
4.1 Mary TTS 15
4.2 Algorithms and program structure 16
4.2.1 Syllable number and pattern length matching 16
4.2.2 Formant matching 17
4.2.3 Setting phoneme attributes 18
4.3 Survey 19

5 Results **20**

6 Discussion **23**
6.1 Perceived emotions 23
6.1.1 Text 23
6.1.2 Tempo 23
6.1.3 Mode 23
6.1.4 Tempo and mode 23
6.2 General discussion of results 24
6.3 Other approaches to the problem 25

7 Conclusion **25**

8 Literature **27**

9 Links **27**

1 Introduction

Speech synthesis is an area of computer science with many practical uses, such as enabling people with visual impairments to take part of text and to provide more human-like feedback from information systems. A similar area of research is text-to-song, where systems comparable to those used in text-to-speech provide mappings from text to melodic units of song. An example of a popular system of this type is Vocaloid[1], which lets users manually compose melodies using computer-generated singing.

When composing music a multitude of different factors can be utilized to invoke certain emotions, which can lead to a richer experience for the listener. By applying these methods in order to create singing from an arbitrary string of text, its contents could possibly be enhanced and perceived in new ways.

As the words of a text can be broken down into attributes relevant to music composing, such as formants, syllables and phonemes, it is possible to analyze how it would best fit in a melody. Combining the previously mentioned music theory of emotions with computer science and phonetic analysis of text, it is conceivable that enjoyable and emotionally expressive music can be algorithmically generated.

For this project we will limit the scope to include research about how tempo and mode affects communicated emotion using our algorithm and the possibility of generating music based on formant analysis.

2 Problem statement

Is it possible to algorithmically generate emotionally expressive music based on syllable length and formants of a string of text?

[1] http://www.vocaloid.com/en/about/, 24-03-2014

3 Background

Before creating a text-to-song system with focus on emotion, it is necessary to study music theory, phonetics, and how text-to-speech systems work in general. It is also useful to analyze similar contemporary systems in order to find what methods are suitable, and how much they differ from the aim of this report.

3.1 Music and emotions

In order to accurately portray the message of a given text in a song, it is important that the musical elements match the lyrical meaning. A happy text combined with a sad melody will most likely lead to a confusing and undesired result. The opposite situation, a sad text and a happy melody, could also lead to inappropriate results. It is therefore crucial that the generated music can portray different emotions accurately.

The types of emotions people associate with a piece music is based on many different aspects of the music. Attributes such as tempo, articulation and pitch are all part of what makes listeners think of happy or sad things when listening. In an article published in the Psychological bulletin (Juslin and Laukka, 2003) some of these attributes are shown to correlate with certain emotions. The study is a review of 145 different studies conducted by playing pieces of music with different attributes to test subjects and letting them determine which emotions were communicated. The emotions Juslin and Laukka compiled the studies into were anger, sadness, happiness, fear and tenderness. By conducting this study a table of emotions and corresponding acoustic cues was created as shown in Table 1[2] [3].

Another article, published in the Handbook of Music and Emotion (Gabrielsson and Lindström, 2012), is based on similar experiments. The attributes compared in this study were focused on structure of the written music, rather than the performance of the music. A selection of relevant attributes and corresponding emotions perceived can be seen in Table 2-7[4].

[2] Juslin, Patrik N. and Petri Laukka (2003). Communication of Emotions in Vocal Expression and Music Performance: Different Channels, Same Code?. *Psychological Bulletin* 129.5: 770-814

[3] Technical terms that are presented in this table but are not used later will not be further explained. Please see the original publication for more thorough explanation.

[4] Gabrielsson, Aalf and Erik Lindström (2012). The Role of Structure in the Musical Expression of Emotions. *Handbook of Music and Emotion*; Oxford University Press

Emotion	*Acoustic cues (vocal expression/music performance)*
Anger	Fast speech rate/tempo, high voice intensity/sound level, much voice intensity/sound level variability, much high-frequency energy, high F0/pitch level, much F0/pitch variability, rising F0/pitch contour, fast voice onsets/tone attacks, and microstructural irregularity
Fear	Fast speech rate/tempo, low voice intensity/sound level (except in panic fear), much voice intensity/sound level variability, little high-frequency energy, high F0/pitch level, little F0/pitch variability, rising F0/pitch contour, and a lot of microstructural irregularity
Happiness	Fast speech rate/tempo, medium–high voice intensity/sound level, medium high-frequency energy, high F0/pitch level, much F0/pitch variability, rising F0/pitch contour, fast voice onsets/tone attacks, and very little microstructural regularity
Sadness	Slow speech rate/tempo, low voice intensity/sound level, little voice intensity/sound level variability, little high-frequency energy, low F0/pitch level, little F0/pitch variability, falling F0/pitch contour, slow voice onsets/tone attacks, and microstructural irregularity
Tenderness	Slow speech rate/tempo, low voice intensity/sound level, little voice intensity/sound level variability, little high-frequency energy, low F0/pitch level, little F0/pitch variability, falling F0/pitch contours, slow voice onsets/tone attacks, and microstructural regularity

Table 1: Acoustic cues and corresponding emotions. Review study by Juslin and Laukka (2003) based off of 145 different studies.

Interval type	*Emotional expression*
Consonant	Pleasant, 'non-active'
Dissonant	Displeasing, unpleasant, 'active', strong
High-pitched	Happy, powerful, 'activity', potency
Low-pitched	Sad, less powerful
Large	Powerful
Minor 2nd	Melancholy
Perfect 4th, perfect 5th, major 6th, minor 7th, octave	Carefree
Perfect 5th	Activity
Octave	Positive/strong

Table 2: Intervals and corresponding emotions. Based on study by Gabrielsson and Lindström (2012).

Tempo	*Emotional expression*
Fast	Exciting, uneasy, agitation, triumph, happy, glad, gaiety, joy, graceful, mischievous, whimsical, flippant, vigorous, happiness, pleasantness, activity, surprise, potency, fear, anger, excitement, energy arousal, tension arousal, activity
Slow	Serene, tranquil, dreamy, longing, sentimental, dignified, serious, solemn, sad, lamentation, excited, sadness, boredom, disgust, tenderness, peace

Table 3: Tempos and corresponding emotions. Based on study by Gabrielsson and Lindström (2012).

Mode	*Emotional expression*
Major	Happy, joy, graceful, serene, solemn, happiness, tenderness
Minor	Sad, lamentation, dreamy, dignified, agitation, sadness, tension, disgust, anger

Table 4: Modes and corresponding emotions. Based on study by Gabrielsson and Lindström (2012).

Loudness	*Emotional expression*
Loud	Excitement, triumphant, joy, gaiety, intensity, strength/power, tension, anger, energy arousal and tension arousal
Soft	Melancholy, delicate, peaceful, softness, tenderness, solemnity, fear, sadness, lower intensity, increased valence

Table 5: Loudness levels and corresponding emotions. Based on study by Gabrielsson and Lindström (2012).

Pitch level	*Emotional expression*
High	Graceful, serene, happy, joy, dreamy, sentimental, pleading, triumph, exciting, gaiety, surprise, potency, anger, fear, activity, hapiness, increased tension arousal
Low	Sad, melancholy, lamentation, vigorous, dignified, serious, solemn, exciting, agitation, tranquil

Table 6: Pitch levels and corresponding emotions. Based on study by Gabrielsson and Lindström (2012).

Melodic direction	*Emotional expression*
Ascending	Dignified, serene, tension, happiness
Descending	Exciting, graceful, vigorous, sadness

Table 7: Melodic directions and corresponding emotions. Based on study by Gabrielsson and Lindström (2012).

3.2 Phonetics

3.2.1 Formants

Formants are the frequencies that shape a sound that we hear. They are most notable when discussing vowels, in spoken and sung words, and in text-to-speech synthesis. The formants are numbered from 1 and up: F1, F2 and so on. The lowest frequency of a sound is called F0 and is what determines the pitch of the sound, which is what we perceive as the tone. When singing, the F0 frequency corresponds to the melody of the song. Additional higher frequencies are what we call formants, which are the reason that we can hear the difference between different vowels and the difference between different speakers.

Formants F1 and F2 are the formants that has the highest impact on what type of vocal sound is produced. It is the relation between F0, F1 and F2 that determines if the vowel we hear sounds like an “a”, “o” or “i”, for example. The F1 frequency of a female voice is often in the 300-800 hz interval. A higher F1 frequency, creates the same result as a lower tongue does when speaking; a so-called low vowel. An example of this is the Swedish pronunciation of the vowel “a”. A low F1 frequency creates a high vowel such as the Swedish “i” and “u”. The F2 frequency of a female voice is in the range of 800-2000 hz. It correlates to the front/back position of the tongue when producing sound. A high F2 frequency creates a front vowel such as the Swedish “i” and a low F2 creates a back vowel such as “u”.[5]

3.2.2 Using formant-pitch matching

In general, the formant of the the vowel is not considered when writing lyrics for a melody. Aspects such as the meaning of the lyrics and beauty of the melody is often what is focused on instead. However, a study made by John Smith and Joe Wolfe (2009) show that some vowels does sound better at certain pitches than others; at least in some types of music.

The study was done on soprano solos in eight classic operas written by Strauss, Wagner, Rossini and Mozart. The number of low and high vowels at certain notes was measured in each opera to determine the correlation between low vowels (open jaw) and a higher pitch, and between high vowels (closed jaw) and a lower pitch. The conclusion was that operas written by Strauss, Mozart and Rossini had no noticeable correlation between vowels and pitch. The four operas written by Wagner however, all had clear correlations between vowels and pitch, especially low vowels and high pitch.

[5]Engstrand, Olle (2004). *Fonetikens grunder*. Studentlitteratur, 97

This shows that Wagner clearly had a preference for combining low vowels (open jaw) with high pitch, although no records suggest that this was a conscious decision. What is more likely is that he considered the acoustics of the soprano voice when writing the music to the lyrics. The reason why other composers lack such correlations might depend on the order text and melody was written or how important it is for the audience to hear all the lyrics. At high pitch it is easier to make out a low vowel than a high vowel and vice versa at low pitch. Even though formant-pitch matching is not necessary to create beautiful music, it might provide a helpful tool in algorithmic generation of songs.[6]

3.2.3 Syllables

Syllables are important units in speech since they contain a large amount of information on how words should be pronounced. All words can be divided into syllables and they therefore become building blocks that build words. A syllable is made up of three different parts, the onset, the nucleus and the coda. The onset is a consonant and determines how the syllable starts. The nucleus is a vowel that is the main part of the syllable. Finally the coda is the end of the syllable. The nucleus and coda is what is called the rhyme of the syllable. Neither the onset nor the coda is necessary, some syllables have neither of them, but the nucleus of the syllable always exists. A syllable without a nucleus becomes a extra-metrical phone and is attached to the previous syllable (such as the s-sound in "it's").[7]

One interesting aspect of syllables is their role in singing. Often when singing, each syllable is sung with one note. Deviations from this principle can sound awkward if syllables get stretched over multiple notes or several syllables are cramped into a single note.

3.3 Text-to-speech synthesis

There are several different models for designing text-to-speech systems, which define the methods of interpreting written signals (text). The *common-form model* eliminates many of the problems that arises when using the other models, as it uses an integrated approach.[8] This means that the model does the necessary operations in a single step, in contrast to other models which have a large number of independent modules arranged in a pipeline. This makes it easier to analyze the model and improve performance. The common-form model defines two basic components of a text-to-speech system, a text-

[6]Smith, John and Joe Wolfe (2009). Vowel-pitch matching in Wagner's operas: Implications for intelligibility and ease of singing. *Acoustical Society of America* 125.5: 196-201

[7]Taylor, Paul (2009). *Text-to-speech Synthesis.* Cambridge, UK: Cambridge UP, 184-186

[8]Ibid, 41-42

analysis system, and a speech-synthesis system. The text-analysis system decodes the text into input for the speech-synthesis system, which then produces speech.[9]

The process of synthesizing speech begins with the input, which is a sequence of ASCII characters.[10] As sentences can largely be processed independently, the input text is subjected to sentence splitting. This is done by checking for end-of-sentence characters, and being sure to check that they are in fact used to end a sentence and not, for example, in an abbreviation. Then, the sentence is divided into different tokens, in a process called tokenization. The tokens are often written words, but may also be numbers, punctuation and other sequences of characters. This process is executed by using a tokenization algorithm which separates parts of the sentences from each other when it encounters certain characters, such as whitespace or punctuation.[11]

The next step in the text-to-speech system is to analyze the tokens using various algorithms in order to find their semiotic class. The semiotic classes are different ways of conveying a meaning, such as numbers, dates, mathematical symbols, or natural language.[12] For the tokens which are not in the semiotic class of natural language, a conversion to natural language words is made. This process is called verbalization. For example, the token "1" is converted to "one" if its semiotic class is determined as cardinal, and to "first" if it is determined as ordinal.[13] For the natural language tokens, ambiguity may arise because of *homographs*, words that share the same written form but differ in meaning and sometimes pronunciation. The context that these tokens occur in is used to determine which of the possible homographic words the token is meant to be interpreted as, through different probability algorithms.[14]

The final step for the text-analysis part of the text-to-speech system is prosodic analysis. It concerns the *phrasing*, *prominence*, and *intonation* of the words in a sentence. Phrasing describes the grouping of words in a spoken utterance[15], and prominence is related to the strength or stress of a word, syllable, or phrase when spoken.[16] Intonation in the context of prosodic analysis is defined as the pitch of different parts of speech, such as

[9]Taylor, Paul (2009). *Text-to-speech Synthesis.* Cambridge, UK: Cambridge UP, 38
[10]Ibid, 41
[11]Ibid, 64-68
[12]Ibid, 33-34 & 78-79
[13]Ibid, 92-97
[14]Ibid, 78-85
[15]Ibid, 112
[16]Ibid, 115-116

the pitch rise common in the end of a question.[17] All of these have different prediction approaches, and advanced algorithms are used to compensate the lack of prosodic information in written text.[18]

At this point, the speech synthesis component takes over, by first encoding the words found by the first component as phonemes.[19] These are heavily influenced by the the syllables of the words they are part of.[20] The phonemes, together with all the different information extracted earlier in the process, is used as input to the module which performs the actual synthesis. In modern text-to-speech systems, the most common type of speech synthesis module is *unit-selection synthesis*.[21] It uses a database of sets of pre-recorded speech units, which correspond to a basic linguistic type. For each type, the algorithm selects the unit that best matches the specified input, in order to produce speech that is as natural and intelligible as possible.[22] Finally, the units are joined together into continuous speech in the shape of a waveform.

Older systems used modeling of the vocal-tract of humans, in order to create the waveforms directly from a specification, instead of storing pre-recorded units.[23] These techniques use formants for the synthesis, and require certain parameters derived from the given specification. As these parameters where either calculated by hand, or derived from an explicit model, the quality of the output was limited.[24] This was later improved on by measuring values from waveforms of real speech, and defining units of speech. These units only consisted of one linguistic type, in contrast to modern unit-selection systems which consist of mappings of one type to several different-sounding units.[25] Another approach to speech synthesis is to use a *hidden Markov model*. This approach uses statistical and probabilistic methods to train the system into describing a model of speech. By doing this, the parameters used for the speech synthesis are generated automatically, and the produced output is of high quality. Voices generated by hidden Markov models are also easier to alter, as the model can be modified.[26]

[17] Taylor, Paul (2009) *Text-to-speech Synthesis*. Cambridge, UK: Cambridge UP, 121-122
[18] Ibid, 41
[19] Ibid, 41
[20] Ibid, 206
[21] Ibid, 41 & 474
[22] Ibid, 474-475
[23] Ibid, 387
[24] Ibid, 412
[25] Ibid, 435
[26] Ibid, 436 & 472

3.4 Contemporary text-to-song systems

Four different text-to-song systems have been chosen for a state-of-the-art demonstration: *Vocaloid*, *Oddcast Text to Sing*, *Let them sing it for you*, and *Melobytes*.

3.4.1 Vocaloid

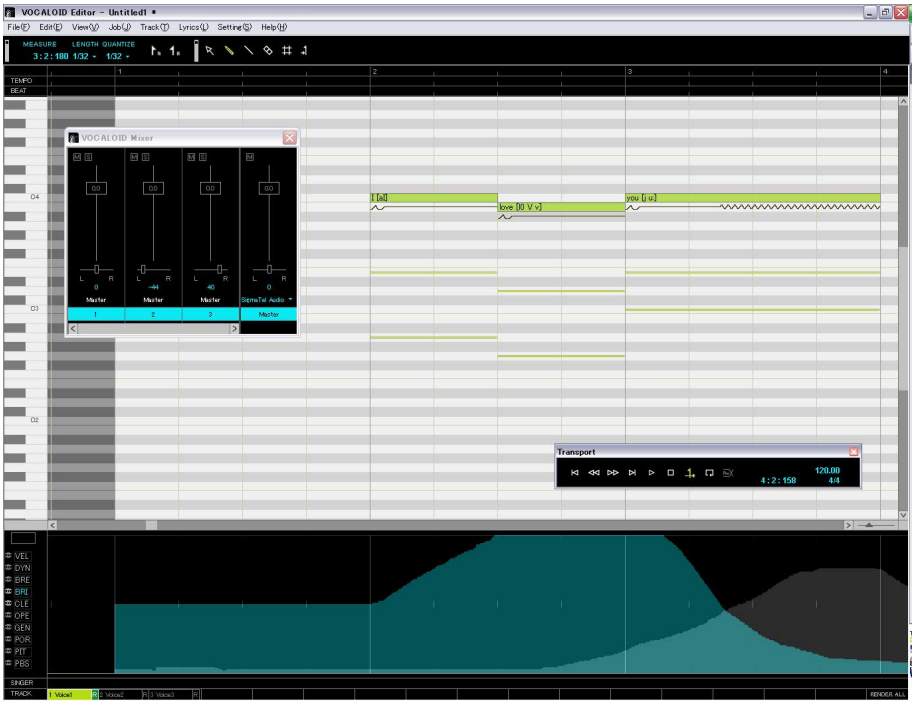

Figure 1: Vocaloid

Vocaloid is a software text-to-speech synthesizer developed by YAMAHA.[27] The software allows users to enter melodies and lyrics in order to synthesize singing. The software is used in the same way as music arrangement software and is utilized by professional producers. Vocaloid and songs created with Vocaloid are very popular today. This is especially true in Japan where the fictional character Hatsune Miku has been created to become an avatar of the voice.[28]

The software is very powerful and is updated and supported by YAMAHA. However, the software is limited to manually created melodies, and has no feature that automatically generates singing.

3.4.2 Oddcast Text to Sing

Oddcast Text to Sing works by letting the user choose from a predefined list of songs and voices, and then alter the lyrics. When the "sing" button is pressed, the altered lyrics are then sung in the melody of the specified song.

[27] http://www.vocaloid.com/en/about/, 24-03-2014
[28] http://www.crypton.co.jp/miku_eng/, 24-03-2014

Figure 2: Oddcast Text to Sing interface

The system produces high-quality output, but is limited by only providing pre-recorded songs, which require manual configuring.

3.4.3 Let them sing it for you

Figure 3: Let them sing it for you interface

Let them sing it for you lets the user enter text freely into a field. When the play button is pressed, the system selects audio samples from popular songs, word by word, and plays them in a sequence. The result consists of a mix of many different songs, scales, arrangements and singers. This differs from the type of music generation attempted in this report, as it uses pre-recorded music units for each word instead of using generation through

programmatic methods.

3.4.4 Melobytes

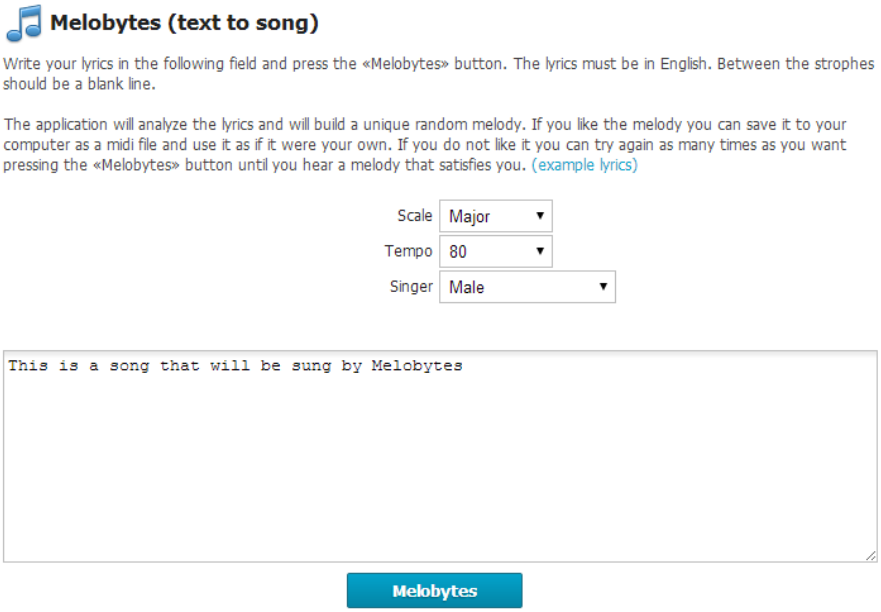

Figure 4: Melobytes interface

Of the text-to-sing systems used in this state-of-the-art demonstration, Melobytes is the one most similar to the system handled in this report. Given a tonality, major or minor, a tempo, and a singer, the system can generate an audio clip in which the text is sung by a synthesized voice. The song is backed up by a musical arrangement, which helps to further enhance the output. However, the selected notes sometimes seem to be selected at random from the scale, which might impair the perceived enjoyability of the music.

4 Methodology

To be able to develop the software used in the research, existing tools were utilized. The tool that is most relevant to the project is the text-to-speech platform Mary TTS. It was used in the creation of a text-to-song program that produces an audio file with computer song given a string of text. The program uses a combination of a database of pitch patterns and rhythm patterns, syllable number matching, and formant matching to produce the output. Mode and tempo can be tuned to affect the result, and the markup language MaryXML was used to modify the necessary parameters of the text. 12 audio clips were generated and used in a survey, where the communicated emotions were evaluated.

4.1 Mary TTS

Mary TTS[29] is a text-to-speech platform developed as a collaboration between DFKI (German Research center for Artificial Intelligence) and the Computational Linguistic and Phonetics department at Saarland University. The software is extensive and contains several languages and provides both unit selection- and hidden Markov model synthesis. It is distributed as a web application, a desktop application and as a programming framework. This gives users access to the features of Mary TTS when developing Java applications.

Mary TTS provides four separate main features:

1. the preprocessing, or text normalization;
2. the natural language processing, doing linguistic analysis and annotation;
3. the calculation of acoustic parameters, which translates the linguistically annotated symbolic structure into a table containing only physically relevant parameters;
4. the synthesis, transforming the parameter table into an audio file.

These allow users to generate acoustic parameters from a string of text, alter them and then continue the synthesis, resulting in a customized output. This is done through utilization of MaryXML, a markup language which defines all acoustic parameters of the synthesis, and allows the input to be divided into tokens and syllables.

The acoustic parameters that are relevant in singing, such as pitch and duration, can be altered using MaryXML, and this process can be conducted

[29] `http://mary.dfki.de`, 04-03-2014

using the provided programming framework[30]. This, in combination with the ability to alter individual syllables, allows song to be generated algorithmically. Mary TTS is therefore a fitting tool in answering the research question.

4.2 Algorithms and program structure

The song-generating program that was used in the research was implemented in the Java programming language, in combination with the Mary TTS interface. The program takes a string of text and generates a sound clip containing a song with the input text as lyrics. The clip is generated according to a specified tempo, pitch and emotion (which mainly decides the mode). The program's main components are called SongSyntheziser, SongUnitGenerator, and FormantPitchMatcher. These components contain various algorithms, and handle different important parts of the song generation process. They are described in more detail in the following sections. In addition to these components, a smaller database of static data was assembled. The frequencies of all necessary notes were stored in a class in order to make them easily accessible by the main program components. In order to make it easier to change key, a function which returns the frequency of the note a specified number of steps away from a certain other note. This also makes it possible to store sequences of pitches using indices relative to each other, rather than storing absolute notes. A number of these types of sequences, or pitch patterns, were created and stored in XML format. The patterns were created in accordance with the research on mode, intervals, and their corresponding communicated emotions that is described in the background section of this report. All pitch patterns were assigned the label "happy" or "sad" according to their mode. They were also divided into sections by their respective length. Some patterns were given an 'end' label, and progressions ending in the tonic as a very short study indicated this would lead to natural endings to the songs. Patterns very similar to these, but representing rhythms, were also created and stored in an almost identical manner. Before execution, a variable specifying the intended emotion is specified as either 'happy' or 'sad'.

4.2.1 Syllable number and pattern length matching

The program uses a system where the words are divided into different syllable groups, which can later be matched with pitch patterns. This process is handled by the SongUnitGenerator. To create the syllable groups the number of groups to use have to be determined. The number of groups used in the experiment were 1, 2, 4 or 8. The max number of notes per group is determined by a constant field. The value 5 was chosen in this case. The

[30] `http://github.com/marytts/marytts`, 04-03-2014

number of syllables in the given text is counted and the appropriate number of groups is decided by finding the least number of groups possible. For example: If the text contains 5 or less syllables, 1 group is used. If the text contains more than 5 but less than 11 syllables two is used and so on. The maximum number of syllables that can be used is 40, but this number can easily be larger if more than 8 groups are allowed.

After determining the number of groups, the algorithm matches up each word to the different groups. The groups have a length which is the number of syllables divided by the number of groups. To ensure that each word is in a single group the algorithm starts adding words to the groups until the number of syllables in a group is greater than the length. The word is added to the group which require the least movement, e.g. if a group has the length 3.5 and already have one word with 2 syllables in it , it has room for 1.5 syllables more. If the next word has 2 syllables, 1.5 of the syllables fit in the current group and 0.5 syllable fit in the next. It is therefore a smaller movement to add the word to the current group instead of the next group. If the new word instead has 4 syllables, 2.5 syllables would fit in the next group. In that case it would be a smaller movement to add the word to the next group. By doing this the words are evenly distributed while still avoiding being split up in to different groups.

4.2.2 Formant matching

After the syllable groups have been created, a pitch pattern for each group is selected. The number of notes in a pattern needs to be the same as the number of syllables in the corresponding group. The mode of the pattern also needs to correspond to the emotion specified earlier in the process. The group of patterns that fit these criteria is extracted through the use of an XPath expression. After the patterns have been found, the process of finding the pattern that best fits the vowel formants in the syllable group is started. This is the main responsibility for the FormantPitchMatcher component, which is a part of the SongUnitGenerator. As indicated in the background section, certain formants interoperate better with certain pitches. For example, it is better to sing closed vowel formants at high pitches. Groups of vowels where assembled in the following way: closed, near-closed, closed-mid, mid, open-mid, near-open, and open. In order to find the pattern with the best formant-pitch correlation, a point system was implemented. For every vowel group, an optimal pitch value was assigned (according to the research). If a pitch matches the formant perfectly, the highest amount of points should be rewarded. If the pitch is slightly lower or higher than the perfect match, a slightly lower amount of points should be awarded. The amount of points awarded decline in a similar fashion as the pitches move further away from the perfect match. This closely resembles the density function of the normal

distribution in probability theory, which was also used as a model for the point system in the program. A normal distribution curve was created for each vowel group, with the corresponding optimal pitch value as the mean (which becomes the highest point on the y-scale) in the function that defines the curve. In summary, the score for a given pitch is the value of the density function of that pitch.

For a single syllable group, the best pattern is found by finding the score of each vowel-pitch match in every pattern of the correct length. The score for the pattern is the sum of the scores of each individual matching. The pattern that receives the highest score is designated as the best pitch pattern for the given syllable group. This process is repeated for each syllable group.

After the pitch pattern has been found, a rhythm pattern with the correct length and emotion is selected. This process is simpler than the pitch pattern selection, and selects one pattern out of the ones found through an XPath expression by using a random number generator. When both a pitch pattern and a rhytm pattern have been selected, a number of *SongUnits* are created. A single SongUnit describes the musical properties of a single syllable: the pitch and the duration, and is used by the SongSyntheziser later in the process. Consequently, the number of created SongUnits is the same as the number of syllables in the group. This process is repeated for every syllable group, until a SongUnit has been created for every syllable in the input text.

4.2.3 Setting phoneme attributes

The Mary TTS system is able to output an XML format called ACOUSTPARAMS, given a string of text. This format divides the text into tokens, which in most cases consist of the words in the text. The tokens have syllables as children in the XML hierarchy. These, in turn, have phonemes denoted with the 'ph' tag as their child nodes. The phoneme elements have attributes that affect the generated speech, 'f0' and 'd'. The 'f0' attribute specifies the pitch and the 'd' attribute specifies the duration.

After the SongUnitGenerator and FormantPitchMatcher components have performed their tasks, SongSynthezier uses the SongUnits that were generated to set phoneme attributes. This is done by iterating through every syllable in the input text, finding the phonemes, and modifying the relevant attributes. The Java class Document provides the functionality for doing this programatically. Each syllable in the text has a corresponding SongUnit, from which the pitch and duration to be set are retrieved. The pitch is set for each phoneme in the syllable· Regarding the duration, the initial approach was to convert each phoneme's duration to a percentage of the total

syllable duration, and then multiplying this with the duration specified by the SongUnit to determine the duration to be set. However, this had a clear impact on the intelligibility of the output, as some consonants were overly shortened. Instead, only the durations of the vowel phonemes are modified in the current approach.

After every syllable in the text has been modified, the new XML is used as input for the speech generating system and the resulting sound clip is played.

4.3 Survey

To be able to determine if the generated songs successfully communicate the desired emotions and to determine what parameters affect the output the most, a survey consisting of 50 participants was conducted. The survey consisted of 12 audio clips generated with different parameters. The parameters that were altered were mode, tempo and lyrics.

The lyrics were based on three Facebook statuses and were chosen to somewhat represent a happy statement, a neutral statement and a sad statement. The different emotions of the lyrics were chosen to be able to determine if musical attributes can outweigh the literal meaning of the lyrics i.e. a happy text with sad music or a sad text with happy music. The happy lyric is *"I just got my first own apartment in central Gothenburg"*. The neutral lyric is *"Does anyone know where I can buy literature for the physics course"*. Finally, the sad lyric is *"I am tired because i did not get any sleep last night"*. The lyrics were purposely chosen to not be very happy or very sad since it's possible that the lyrical meaning of the text would outweigh any emotions communicated through the music. Mode and tempo were chosen as simply happy or sad. Combining these parameters in all possible permutations gave us 12 different audio clips to use in the survey.

The survey was conducted individually by playing each clip once and letting the test subject choose what emotion they believed was communicated in the clip. The alternatives were: Sad, Somewhat sad, Neutral, Somewhat happy and Happy. Instructions were given that the communicated emotion, and not the conveyed emotion, should be evaluated. The order of the audio clips were the same for all surveys and were as follows:

1. Neutral text; Fast tempo; Major tonality
2. Neutral text; Fast tempo; Minor tonality
3. Neutral text; Slow tempo; Major tonality
4. Neutral text; Slow tempo; Minor tonality

5. Happy text; Slow tempo; Major tonality
6. Happy text; Fast tempo; Minor tonality
7. Happy text; Fast tempo; Major tonality
8. Happy text; Slow tempo; Minor tonality
9. Sad text; Fast tempo; Major tonality
10. Sad text; Fast tempo; Minor tonality
11. Sad text; Slow tempo; Major tonality
12. Sad text; Slow tempo; Minor tonality

5 Results

All raw data compiled from the survey can be seen in appendix A.

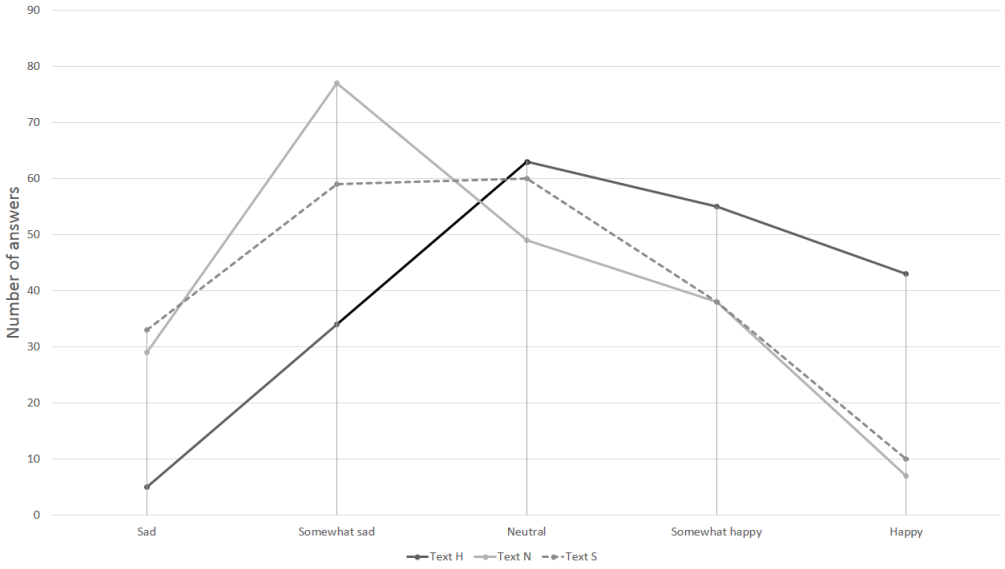

Figure 5: This graph visualizes the distribution of perceived emotions for the three different texts, with no regard to the tempo or mode. 'Text H' (black) is the happy text, 'Text N' (grey) is the neutral text, and 'Text S' (dashed) is the sad text.

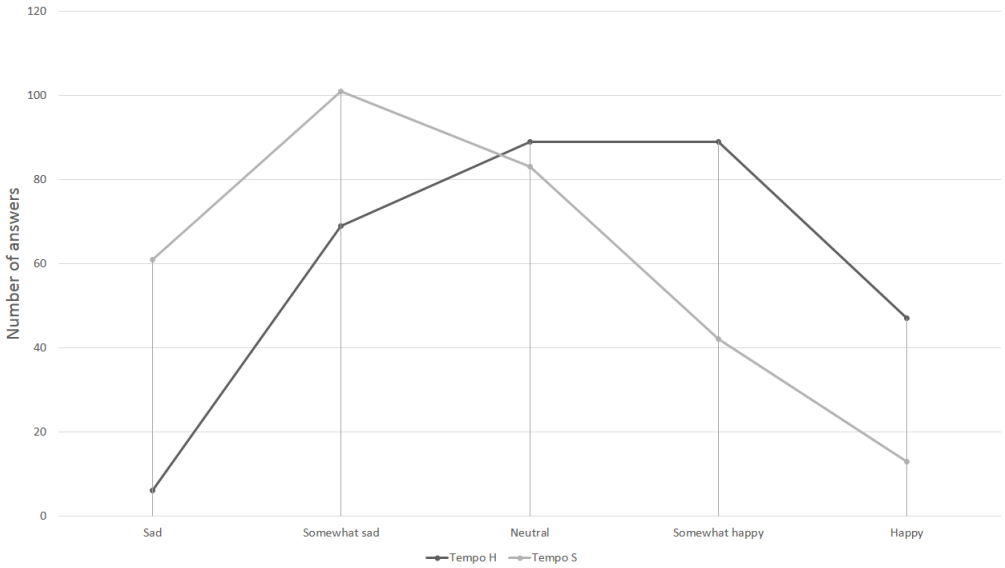

Figure 6: This graph visualizes the distribution of perceived emotions for the two different tempos, with no regard to the mode or text. 'Tempo H' (black) is the happy, fast tempo, and 'Tempo S' (grey) is the sad, slow tempo.

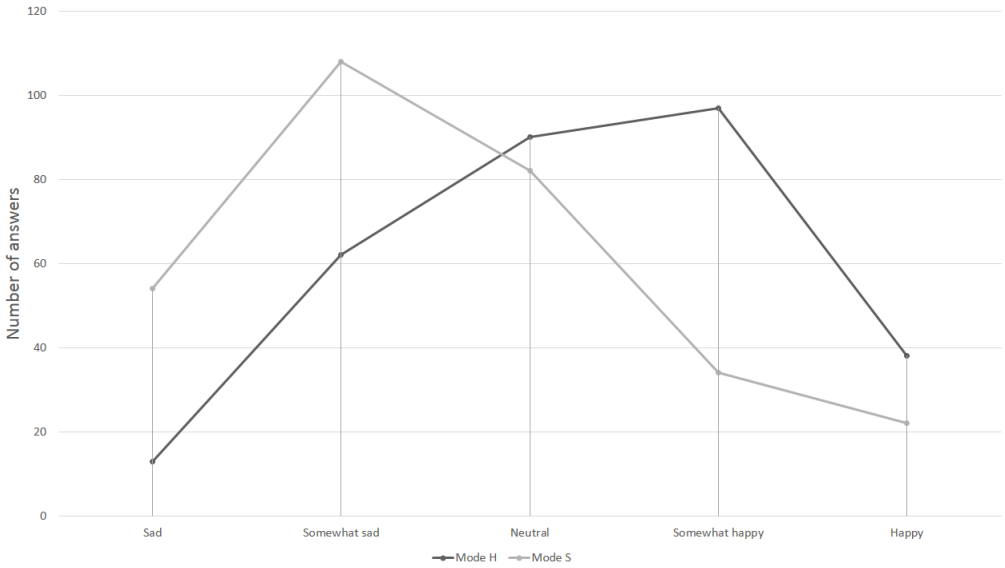

Figure 7: This graph visualizes the distribution of perceived emotions for the two different modes, with no regard to the tempo or text. 'Mode H' (black) is the happy, major mode, and 'Mode S' (grey) is the sad, minor mode.

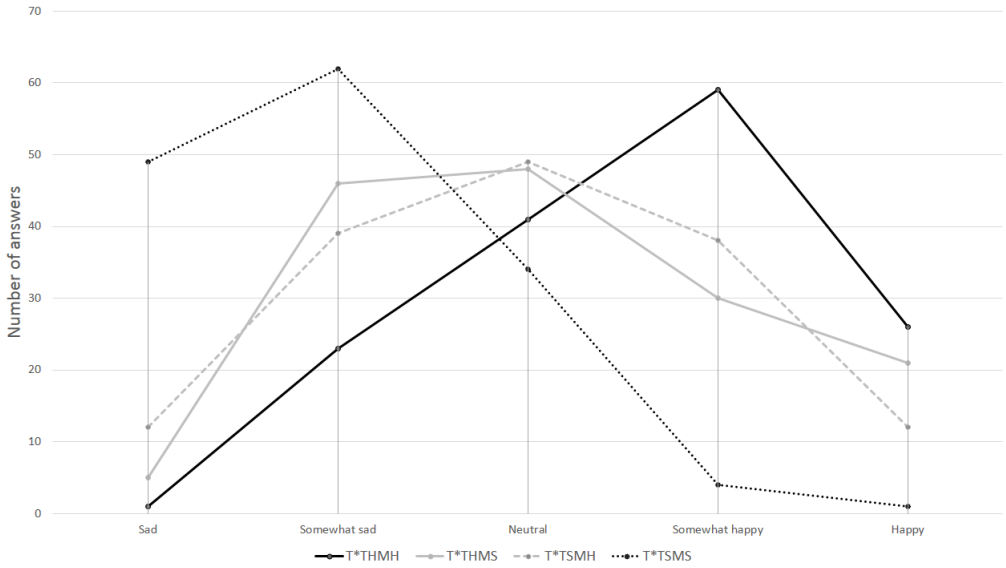

Figure 8: This graph visualizes the distribution of perceived emotions with respect to both the tempo and the mode, but with no regard to the text. 'T*' denotes that the text is ignored. 'T*THMH' (black) denotes happy tempo in combination with happy mode. 'T*THMS' (grey) denotes happy tempo in combination with sad mode. 'T*TSMH' (dashed) denotes sad tempo in combination with happy mode. 'T*TSMS' (dotted) denotes sad tempo in combination with sad mode.

6 Discussion

6.1 Perceived emotions

6.1.1 Text

Figure 5 visualizes the distribution of perceived emotions for the three different texts, with no regard to the tempo or mode. The data shows that the text itself has at least some effect on the communicated emotions of song. However, the neutral text received more 'sad' responses than the sad text overall. This could point to the the possibility that the fact that the generated melodies are different contribute more to the conveyed emotion than the content of the text. The neutral text could have generated melodies that, in general, are perceived as more sad than those generated by the sad text.

6.1.2 Tempo

Figure 6 visualizes the distribution of perceived emotions for the two different tempos, with no regard to the mode or text. The data indicates that the tempo of the song clearly influences the perceived communicated emotions, in accordance with the research conducted in the background section. This is evident throughout the whole spectrum between sad and happy. Note that a low tempo inspired sad responses slightly more often than high tempo inspired happy ones. This is discussed later in section 6.1.2.

6.1.3 Mode

Figure 7 visualizes the distribution of perceived emotions for the two different modes, with no regard to the tempo or text. It shows that mode also clearly affects the emotions communicated by the generated text. Similar to the results in Figure 6, the sad, minor mode resulted in more sad answers than the happy, major mode resulted in happy ones. It is possible that this phenomenon is caused by the fact that the clips were generally interpreted as negative slightly more often than as positive.

6.1.4 Tempo and mode

Figure 8 visualizes the distribution of perceived emotions with respect to both the tempo and the mode, but with no regard to the text. Tempo and mode in combination undoubtedly affect the result more than the individual parameters by themselves. The combination of happy tempo and happy mode produced the most happy responses on average, and similarly, sad mode in combination with sad tempo produced the most sad responses on average. Tempo and mode seems to affect perceived emotion equally since happy tempo and sad mode gives almost identical results as sad tempo with happy mode.

6.2 General discussion of results

One interesting result that the survey shows is that the magnitude of the emotion answers for both tempo and mode seems to peak at somewhat happy/somewhat sad for their respective happy and sad values. One possible reason for this, when observing tempo, is that the values chosen (120 bpm for sad and 250 bpm for happy) simply weren't expressive enough i.e. sad should be a lower value than 120 and happy should be a higher value than 250. However, both values were chosen because undesirable results could be observed when using values outside this range; the output from Mary TTS started to sound unnatural, which was deemed as something that could affect the user study negatively.

When observing mode, it is not possible to use "more major" or "more minor" scales than the ones used in the study (assuming one does not use other scales such as melodic or harmonic minor scales which were beyond the scope of this project). However, certain notes; thirds, sixths and sevenths, can be emphasized to make the mode more distinct. Since all melodies were generated using the formant-matching algorithm no care was given to ensure that the modes were emphasized. This may have resulted in texts generating, what would be perceived as, sadder or happier melodies despite their literal meaning and what mode was used to generate them. This could explain why the sad text and the neutral text had very similar results.

Another possible reason why people were hesitant to use happy and sad to describe the song and instead used somewhat happy and somewhat sad is the computer generated voice. The voice generated by Mary TTS sounds robotic and not very lifelike. It is possible that people have difficulty determining the emotion expressed when the voice expressing the emotion does not sound human. To determine if this is the case the experiment could be repeated using technologies such as Vocaloid, which generates a much more lifelike voice. One could also compare the computer generated voice with a recording of a real human voice singing the same melody and tempo to see if this gives the same result or if the human voice expresses more emotion by simply being human.

As mentioned previously, the clips were interpreted as negative slightly more often than as positive, on average. Apart from randomness and subjectivity, a possible reason for this is that all clips end with a downward note progression. This is a natural result of the choice to end all clips on the tonic. Descending note progressions can be interpreted as sad (as described in table 7), and the fact that the audio clips end with them may increase their impact on the communicated emotion.

While it is a difficult task to determine exactly how the formant matching component of the program affects emotions, it is conceivable that it enhances the result. Since the matchings provide more pleasant-sounding music, potential sources of disturbance are eliminated, and the emotions can be conveyed more clearly. A simple way to improve the formant matching algorithm is to provide more pitch patterns, as this increases the likelihood of finding a fitting pattern for every syllable group. Overall, the impact of formant matching on conveyed emotions is an interesting subject for further study.

6.3 Other approaches to the problem

When the research began, much focus was given to what properties the melodies should have, and not how it should be implemented. The structure that was used was based on patterns of melodies and rhythms. This meant that all melody patterns that could be selected had to be manually added. Not every single possible pattern could be used since the result could then end up sounding random and unnatural. The patterns had to be created with concern of the melodic rules set up in the literature study. Therefore, all patterns that were added were patterns that were deemed pleasant-sounding. In total more than 300 patterns were added which should be enough to give accurate results, but the results have undoubtedly been affected in some way by doing this. Another approach that would be very interesting to compare results with would be to create a Markov model of intervals in music. By analyzing a large number of songs a transition matrix could be created, which could be used to generate new songs which hopefully would sound like they were composed by humans. This would mean that controlling parameters, such as what intervals should be avoided or and what should be used more often, would be very easy since it would just mean adjusting numbers in the transition matrix.

7 Conclusion

The results show that the generated audio clips successfully conveyed emotions to the listeners, especially when combining several sad parameters or several happy parameters. However, the results only show this to some extent, and on some occasions clips with mainly happy parameters were perceived as somewhat sad. This leads to the conclusion that it is definitely possible to generate emotionally expressive song algorithmically, though it can likely be done more efficiently. By using even more parameters in the algorithms that generate the song, such as overall pitch and volume, the produced audio could possibly convey the intended emotions more clearly and to a greater extent. As mentioned earlier, methods such as Markov models could also be used to test if more satisfactory results can be produced.

The study indicated that using the technique of algorithmically generating songs, an arbitrary string of text could be enhanced to be more enjoyable, convey a clearer message, or change how it is interpreted. For example, the emotion conveyed by the neutral text was very different depending on the different parameters. By using and enhancing the techniques described in this report, emotionally expressive song could conceivably be used in exciting ways in the future.

8 Literature

1. Engstrand, Olle (2004). *Fonetikens grunder*. Studentlitteratur

2. Gabrielsson, Aalf and Erik Lindström (2012), *Handbook of Music and Emotion: The role of structure in musical expression*; Oxford University Press, p 378-38

3. Juslin, Patrik N. and Petri Laukka (2003). Communication of Emotions in Vocal Expression and Music Performance: Different Channels, Same Code?. *Psychological Bulletin* 129.5: 770-814

4. Smith, John and Joe Wolfe (2009). Vowel-pitch matching in Wagner's operas: Implications for intelligibility and ease of singing. *Acoustical Society of America* 125.5: 196-201

5. Taylor, Paul (2009). *Text-to-speech Synthesis*. Cambridge, UK: Cambridge UP

9 Links

The GitHub repo for the program used for research in this report:
`https://github.com/andreascmj/text-to-song-synthesis`

The audio clips used in the survey:
`https://dl.dropboxusercontent.com/u/40501118/Survey%20audio%20clips.zip`