Proceedings of the Institute of Acoustics

A VOICE CONTROLLED MUSIC SYNTHESIS SYSTEM

Ian S Gibson and David M Howard

Department of Electronics, University of York, Heslington, York YO1O 5DD. UK.
(isg100@york.ac.uk)

1. INTRODUCTION

A Voice Controlled Sound Synthesis (VCSS) interactive software environment has been developed which allows mapping of parameters from a performer's voice onto parameters for a music synthesizer. The system is written in C and runs on a Silicon Graphics Indigo (SGI) workstation in real-time. The musician constructs instruments in a modular format using a graphical user interface and functions are available for analysis of audio control data, input of MIDI data, mapping of instrument data and output of sound control data in MIDI format.

Voice analysis functions are implemented in software and include formant estimation, Linear Predictive Coding co-efficients and spectral information. The resulting data is made available to the instrument designer and updated at intervals set by the user.

The system provides graphical displays of Linear Predictive Coding parameters, a time domain waveform, fundamental frequency estimation and spectral data from a Fast Fourier Transform. A number of dynamic graphical sliders are available for monitoring parameters.

Musical instruments are programmed using a graphical user interface. They map voice input to both sound synthesis and studio effects parameters. The paper presents a description of the system and results.

2. BACKGROUND

With digital sound synthesis techniques many parameters can be specified, including those relating to rhythm, pitch and timbre. The composer controls and refines the resulting music through a process of experimentation [Dannenberg, 1993].

The Musical Instrument Digital Interface (MIDI) [IMA, 1988] has been adopted by most manufacturers of computer-based music instrument hardware as the standard protocol for communication between devices. A typical MIDI keyboard provides only a limited number of control parameters in real-time; typically, these are pitch, velocity, after-touch, pitch bend, modulation and changes of sound *patches*. Music Computer Languages (MCLs) allow absolute control of synthesized sounds when the computer is usually used to process a script and generate the resulting sound.

The voice is an ideal source of control for synthesised sound as it is able to change rapidly in pitch, amplitude and timbre to communicate ideas and express a wide variety of emotions. Everyone who can speak has some control over these parameters, and professional singers have a great deal of control over them. Using the voice is a skill which most people practice every day, therefore a system using voice input would be instantly accessible to many without the need to learn new skills, provided the user interface is suitably intuitive. Previous research [Gibson and Howard, 1994] has been principally concerned with the control of synthesized sound by voice in a non real-time environment. This paper presents an

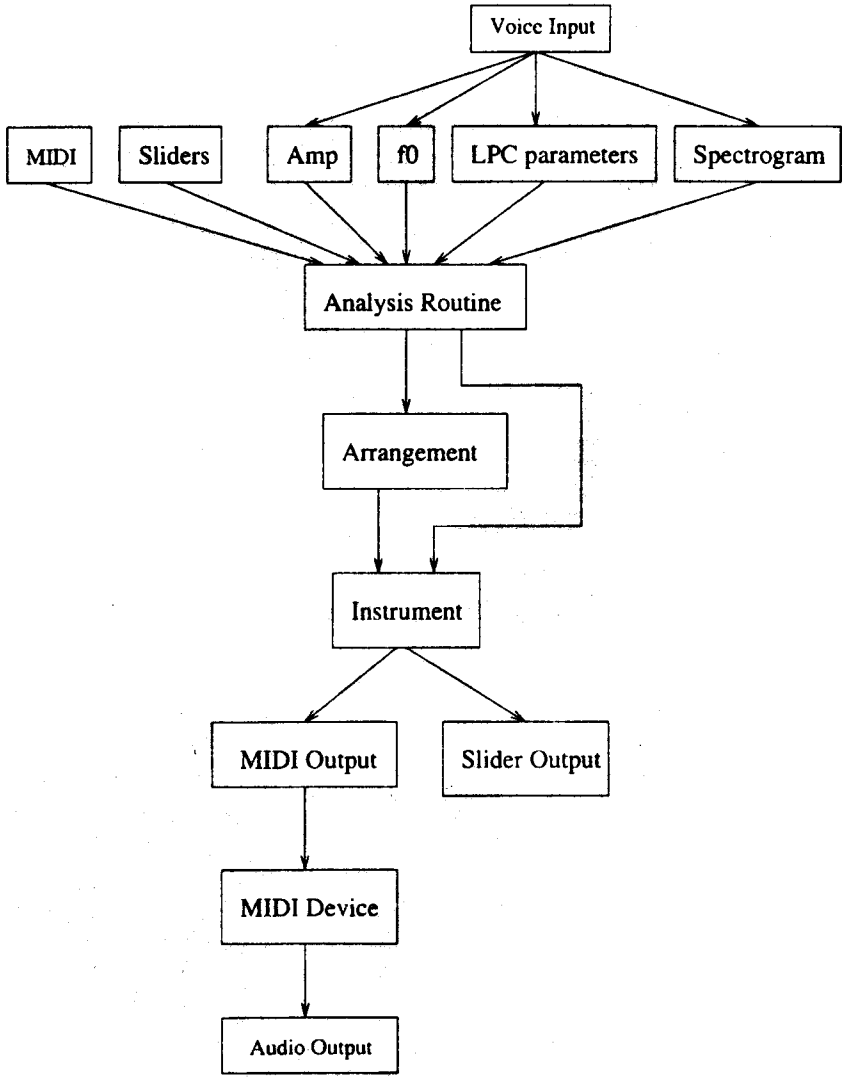

Figure 1: Overview of the VCSS System

updated version of the system which allows the composer to control synthesized sounds in real-time using the voice.

3. THE MUSIC SYNTHESIS SYSTEM

The *Voice Controlled Sound Synthesis* (VCSS) system (figure 1) allows creation of *virtual* instruments. Voice input is taken through a microphone, and all analysis and data processing is undertaken by software.

The VCSS system comprises of the following sections:

- input (voice, MIDI and GUI),
- processing (instrument definition, piece arrangement, performance feedback)
- output (MIDI and graphics), and
- configuration (to optimise real-time system performance).

Input sources of the system are audio, MIDI and the graphical user interface. These are processed by the analysis routines. This data is then made available to both the *arrangement* and *instrument* functions of the program. The *arrangement* section processes and outputs parameters which control global aspects of a piece. The *instrument* section processes and outputs parameters which control individual instruments.

3.1. THE VOICE ANALYSIS FUNCTIONS

This section describes the VCSS voice analysis routines.

3.1.1. INPUT AMPLITUDE ANALYSIS

The Sound Pressure Level (*SPL*) is measured by the system. Each frame of data input into the audio buffer is analysed and its peak absolute value ascertained. The hardware allows a maximum positive value of 32767 for a sample, and a minimum value of -32768. Given that an absolute maximum peak sample value of 32768 produces a 0dB result, the following equation can be used to establish the change in SPL from that point:

$$SPL_tdB = 20log_{10}\frac{A}{32768}$$

where A is the peak sample value.

3.1.2. FUNDAMENTAL FREQUENCY ANALYSIS

The fundamental frequency is estimated using a threshold analysis basic extractor algorithm [Hess, 1983] (figure 2). A low pass filter is applied to the current input data frame (or window). The higher harmonics are filtered out enabling the fundamental frequency to be estimated accurately

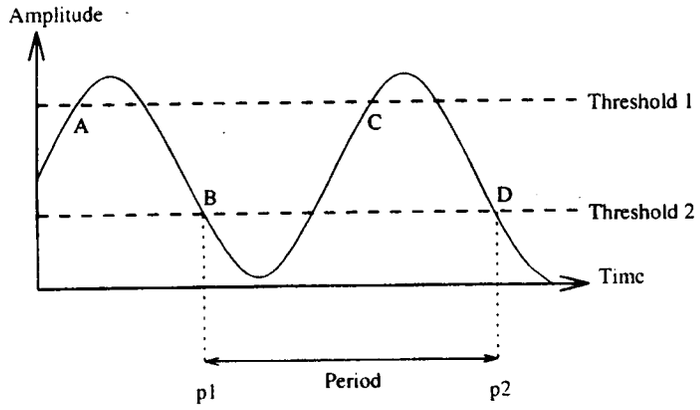

Figure 2: The Threshold Analysis Basic Extractor

using the threshold analysis basic extractor algorithm. After the low pass filter is applied, the waveform is normalised (i.e. the waveform is scaled so that the highest sample has a value of 32767).

The first method of fundamental frequency estimation involves a single threshold level. The threshold must be crossed by an increasing voltage for the position to be noted (in units of samples) by the system. When this event has occurred twice (Points A and C in figure 2), then the period of the waveform may be calculated.

The second method of calculation involves two threshold levels. In order to establish the fundamental frequency, the waveform must first cross the first threshold with increasing voltage (Points A and C in figure 2), and then cross the second threshold with decreasing voltage (Points B and D in figure 2). The second set of points (B and D) are used to indicate one period of the waveform.

With either the first or second method, two points are established (A and C for method 1, B and D for method 2). The fundamental frequency (f0) can be calculated with two points (p1 and p2) using the following equation:

$$f0 = \frac{samplerate}{p2 - p1}$$

where p1 is the position of the first crossing of the waveform (in units of integer samples) and p2 is the position of the second crossing of the waveform (in units of integer samples). All calculations are quantised to the nearest sample.

3.1.2. ESTIMATION OF LPC COEFFICIENTS

Linear Predictive Coding coefficients are calculated with each data frame. The system applies an autocorrelation algorithm to the data. Autocorrelation is suited to longer frame sizes than the Covariance method requiring a window to be applied to the frame. This does require additional computation time, but it ensures that instabilities which can arise from the Covariance method are avoided.

3.1.3. INPUT SPECTRUM ANALYSIS

An FFT algorithm is applied to the input waveform. The FFT has a resolution of 128 points. The frequency range is dependent on the input window frame size (which is user-defined) and the sample rate (typically 8kHz to 16kHz). A frequency range of 0 - 4kHz is available at a sampling rate of 8 kHz. This will enable monitoring for example of the singer's formant which is generally located in the range 2kHz to 4kHz.

3.2 INSTRUMENT DESIGN

The *Instrument definition* section allows the user to create instruments which map input parameters (graphic sliders, voice parameters and MIDI input) to output parameters (MIDI and graphics). Each instrument comprises of one or more *routines*. Each *routine* performs a high-level function. An example of a high-level function might be to gather all of the input data for an instrument. A *routine* comprises of a number of lower-level primitives called *modules*.

When an instrument is activated each *routine* is executed sequentially. Typically, when a *routine* is *active*, each *module* within a *routine* is also executed sequentially. After the last *module* in the last *routine* has been executed, control returns to the first *module* in the first *routine*.

3.3 PERFORMING WITH THE SYSTEM

The *Arrangement* section is used to structure a piece. The user specifies times when each instrument shall become active within a piece. A set of inputs are provided to enable control of parameters at the *arrangement* level (i.e. parameters which effect the whole piece, e.g. tempo). Variables are updated periodically from the input sources. The rate of update is variable, for example once a second.

3.3.1. MONITORING AND FEEDBACK

Several features are provided for monitoring and feedback of instrument performance parameters. The *On screen* form displays:

- the current voice f0,
- the current MIDI note on value,
- the current MIDI pitchbend value,
- the current amplitude,
- the audio input waveform,
- a spectrogram for the current input,
- LPC parameters for the current input and
- instrument parameter displays.

4. RESULTS

Instrument one (figure 3) allows the voice fundamental frequency to control MIDI pitchbend. Pitchbend allows finer control of output pitch than can be achieved with the conventional 12 semitone octave. A MIDI *note-on* value allows output to the nearest semitone and this is then followed by a pitchbend controller. The instrument *locks* onto a MIDI note-on value when voice input exceeds an amplitude threshold. This is achieved by using conditional execution of a VCSS routine based upon input amplitude values. The *locked* value corresponds to the closest semitone. Output pitch is then modified by pitchbend (allowing continuous tracking of the voice fundamental) until the input amplitude drops below the threshold level.

Instrument 2 changes the output oscillator waveform according to the input vowel. It is identical in structure to instrument 1 (figure 3) except that the *play pitchbend* module is replaced by the subroutine shown in figure 4 which outputs several controllers for pitchbend, DWSS wavetable selection and filter modulation amplitude. Dynamic Spectral Wavetable Synthesis (DWSS) is used and involves selecting one wavetable from a set of 64. Each wavetable in the set bears some relation to its neighbour in the set. Therefore gradual changes in timbre are possible by moving from one wavetable to its neighbour.

An /a:/ vowel sound input results in one DWSS waveform being selected and an /u:/ vowel sound selects another waveform. A spectrogram of system output is shown in figure 5. The current sound input is established by examining LPC coefficients provided by the VCSS voice analysis routines. By gradually changing input from one vowel to the next, a *morphing* effect is achieved. In addition, increasing input volume causes filter modulation amplitude to increase.

5. CONCLUSIONS

A system has been implemented which maps voice input parameters to sound synthesis controllers. A variety of functions have been implemented thereby demonstrating that sound synthesis can be controlled by voice.

The system has several areas of further research. One area is related to improvements within the system. This would include research into:

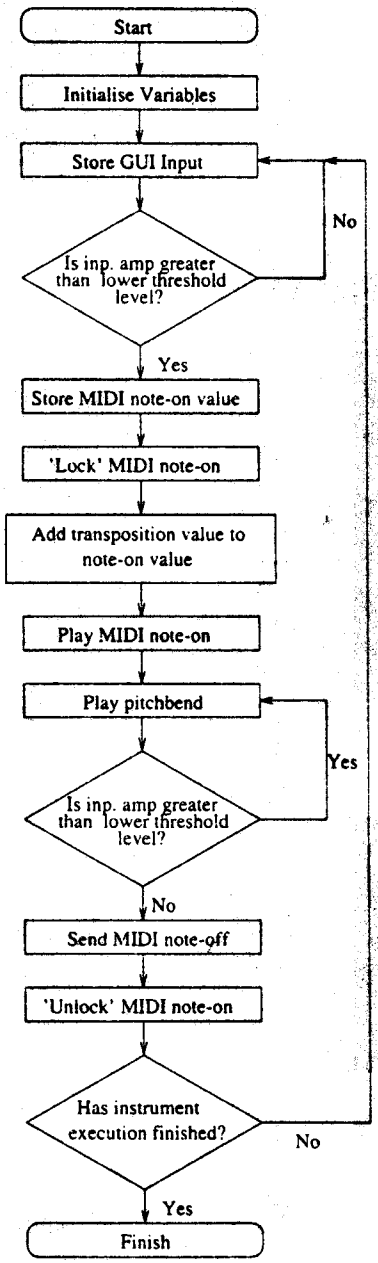

Figure 3: Instrument 1. Voice fundamental frequency controlling pitchbend

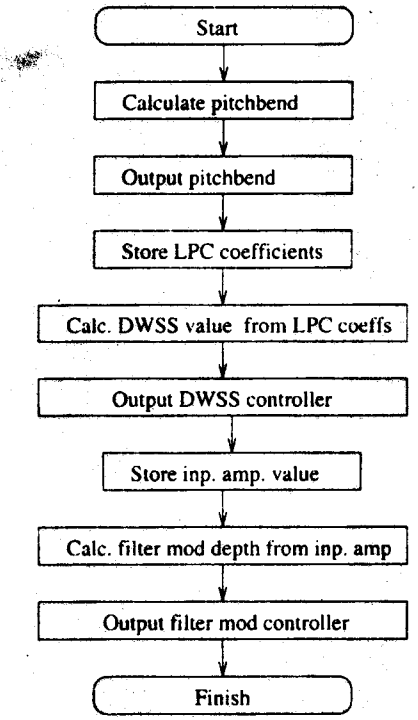

Figure 4: Instrument 2 subroutine to output MIDI controllers

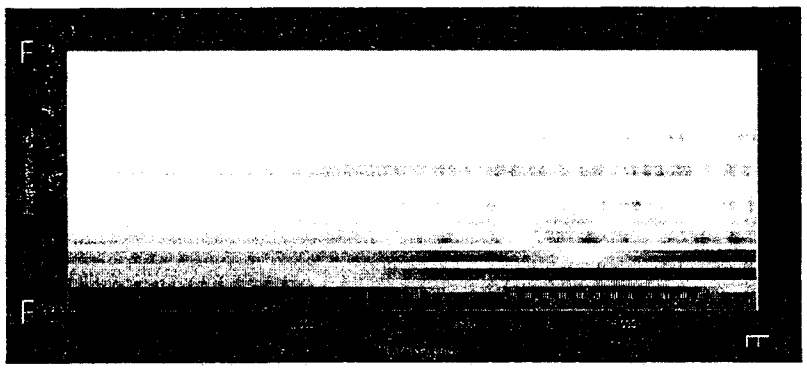

Figure 5: A spectrogram of output from Instrument 2. The performer gradually changes the input vowel from /u:/ to /a:/, resulting in a change of the output sound timbre.

- running the system on a more powerful (perhaps parallel) system and overcoming problems associated with delays caused by one data input window for all analysis routines,
- new functions to easily detect vocal features such as jitter and shimmer,
- routines to prevent feedback of output sound back into the analysis routines,
- which library routines are most useful to performers, and the best performance environment for the system.

The second area of further research relates to applications of the system such as music therapy and singing training. Investigation may be carried out into factors governing voice controlled virtual instruments which appear most intuitive to control by the performer. Also consideration may be given to factors of instrument design for performance of different musical styles.

References

[Dannenberg, 1993] DANNENBERG R.B. (1993). Music Representation Issues, Techniques, and Systems. *Computer Music Journal*, **17**, (3), 20-30.

[Gibson and Howard, 1994] GIBSON I. & HOWARD D. (1994). Sound Processing Control Using Voice and Graphical Information. *Eurographics UK Chapter 12 Eurographics UK Conference Proceedings*. 19-28.

[Hess, 1983] HESS W. (1983). *Pitch determination of speech signals: algorithms and devices*. Berlin: Springer.

[IMA, 1988] IMA (1988). MIDI 1.0 Detailed Specification, version 4.0. International MIDI Association.