The CyberWhistle - An Instrument For Live Performance

Dylan Menzies, David Howard

Department of Electronics, University of York UK

Abstract. The CyberWhistle is an integrated approach to a new electronic instrument, with emphasis given to aesthetic and practical considerations. It consists of a *penny whistle*[1] fitted with sensors and electronics connected by cable to a desktop computer, a Silicon Graphics Indy. In contrast to many windcontrollers, the continuous position of the fingers is sensed. Audio is produced on multiple channels using various software synthesis methods, including waveguide modeling. The design of software and hardware have proceeded in parallel, which has helped in the musical unification of the instrument. The word *cyber* is chosen to reflect the welding of a ‘natural’ form, the whistle, with modern technology, and also, hopefully, the fusing of the player with the instrument.

1 Introduction

Aesthetic Background The development of the CyberWhistle was strongly influenced by the traditional whistle music of Ireland. The effect, on the whistle, of gradually withdrawing the lowest covering finger is to raise the pitch smoothly to the note above. Irish music exploits this technique to the full by combining such shifting *shadings* in rapid succession. The resulting intricate patterns of sound help to compensate for the simplicity of tuning and tone. In the contemporary electroacoustic style, tuning, or at least the ability to create complex hierarchies with pitch, is frequently not very important. More important is the ability to shape the sound in detail. The whistle thus suggests itself as a possible means of performance in electroacoustic music: The movements of the fingers simultaneously provide a rich method for continuous control of musical parameters. Complementary to this is the technique of *tonguing*, in which the player rapidly blocks and unblocks the mouthpiece opening with the tongue tip, affording a way to generate discontinuities.

Note Transitions Even for standard, pitch based, windcontrollers there are good reasons for tracking the position of the fingers, rather than just using switches. The depression of a switch can only effect the sound when it becomes closed, and is unaffected by the speed of closure. In real instruments the inter-note transitions depend on the closure time profile. This is not only of interest in emulating real instruments, but of general use in creating instruments with rich response characteristics.

The Justified Whistle Many interfaces are possible for measuring continuous finger movements, for instance there exist several *dataglove* types. It is therefore important to identify reasons why the whistle should be retained and not developed into some other form unaffected by the restrictions of acoustic design. The whistle has several excellent ergonomic qualities. The finger holes are wide and can be accurately sensed at the rims, giving feedback on the position of the fingers [2]. Instrument controllers which track movements of free body parts limit force-feedback to that which occurs inside the players body. Conversely, force feedback which requires sustained application of pressure by the player is tiring. The fingers are in view, which helps visual feedback. The mouthpiece is easily used. The blowing feel, or force feedback, of a real whistle is substantially unaltered in the CyberWhistle: Many windcontrollers attempt to emulate reed instruments for which it is not possible to retain the more complex mouth and throat interactions. The visual appearance is uncluttered and pleasing. A more abstract aesthetic comes from the whistle, one of the earliest instruments, being combined with modern technology. The form of the penny whistle is familiar to a great many penny whistle players, many of whom began playing the whistle before moving to more elaborate wind instruments. In the author’s experience such players are attracted by the CyberWhistle.

Practical Considerations The whistle provides a very convenient outer shell. The electronics can be mounted on cylindrical former which slides into the whistle bore.(In fact the whistle can be used as normal by withdrawing the the former.) Penny whistles are inexpensive. Six channels of continuous finger movement turns out to be technically reasonable. Certainly a more elaborate finger scheme would need a more elaborate approach than described in the following section.

2 The Electronic Development

2.1 Finger Sensing

The penny whistle player bends a note by rocking the finger to the side of the hole. Once the finger rises more than a few mm it ceases to have any effect. For the CyberWhistle we should like to have good resolution in this lower region, possibly with some resolution at wider separations, to extend the technique. Electric field sensing has a long history of use in electronic instruments dating back to the Theremin [2]. Of the various operation modes

---

[1] Also known as a *tin whistle*. The metal bore is cylindrical and contains six finger holes on the up side.

[2] Other wind instruments with open holes such as the clarinet, have much smaller holes.

described clearly in [5] the *shunt mode* proves to be workable. The finger is earthed to the player, and effectively reduces the conduction across the capacitor-sensor. The metal bore helps to isolate the finger sensors. Circuitry for accurately measuring the change in conduction is described in [5]. For six channels the total circuitry becomes a little awkward.

Light Sensing with LEDs One possibility is to measure the strength of infra-red light reflected from the finger back into the fingerhole. The main drawback is dependence on skin reflectance. Another approach is to measure the occluded ambient light intensity relative to the ambient level detected by a free sensor, **Figure 1**.

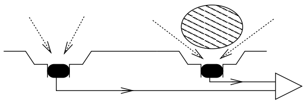

Fig. 1. Compensated ambient light sensor.

The ratio of these quantities is constant to ambient level change, and hence a measure of finger position. Combining a photo diode or transistor with a transconductance amp gives a voltage nearly proportional to light intensity. So, dividing the occluded voltage by the free voltage gives finger position. Unfortunately dividing voltages is not easy. The best option would be to divide with a suitable ADC by supplying the ambient level voltage as the reference voltage, but even this would add significantly to the total circuit complexity.

Light Sensing with LDRs LDRs are can be used as switching devices, for a musical example see [3], but can also be used for measuring light intensity. Miniature LDRs are available from several manufacturers which suit the dimensions of finger holes very well. They are approximated by a pure resistance which under steady lighting conditions satisfies

$$R = AI^B$$

for constants A and B, intensity I, resistance R. If the ambient intensity is I_a and the shaded intensity on the finger hole is I_f, then

$$\frac{R_f}{R_a} = \left(\frac{I_f}{I_a}\right)^B$$

The intensity ratio is constant to ambient light level change, as for the LEDs. So the resistance ratio will also be constant, and so can be used to measure finger position. Producing a voltage related to the resistance ratio is simple using a voltage divider. However, we should like to compensate six finger hole LDRs using one free LDR. The circuit used for achieving this is shown in **Figure 2**.
Preset 1 is used to adjust the working distance range. Preset 2 controls the voltage output range. Some variation is encountered in the behaviour of a batch of LDRs. Usually the variations are small, and it is more convenient to correct these in software than in hardware, using a simple calibration test.

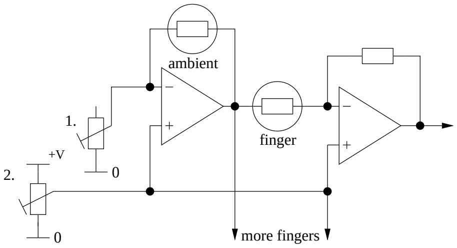

Fig. 2. LDR compensation circuit.

Testing An LDR Fingerhole A simple test rig for the LDR fingerhole reveals more strengths and weaknesses. The old term for covering a finger hole, *shading*, is particularly apt because the player can see the shadows cast by the fingers directly. For small source lights the shadows are harder, but the finger control is still continuous because the shadows move across the receptive area of the LDRs. Photo semiconductors have much smaller receptive areas. Most lighting conditions give a suitable variation profile. The greatest voltage variation is near the finger down position, ensuring better resolution in this region when sampled. The circuit can be made to work over a wide ambient light range, but in the lower region the response rapidly deteriorates due to a *cooling* effect in the LDRs. The response to a rapid finger depression is a slowing up near saturation. To work around this, the soft ware can generate a finger down state before the voltage saturates. The finger release doesn't suffer the same delay because the light level is immediately increased.

2.2 Breath Sensing

Breath Noise Detection Two schemes have been implemented for breath control. The first measures the noise generated by the breath passing over a rough plastic surface, using an inexpensive sub-miniature electret microphone of the kind found on computer desktops (See **Figure 3**).

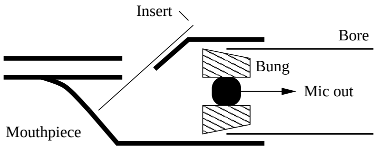

Fig. 3. Using an electret mic in the mouthpiece.

The audio signal is fed directly to the computer audio input and processed digitally as outlined in **Figure 4**.
Tonguing can be followed quickly by detecting the changes to and from low level signals, when the noise generated

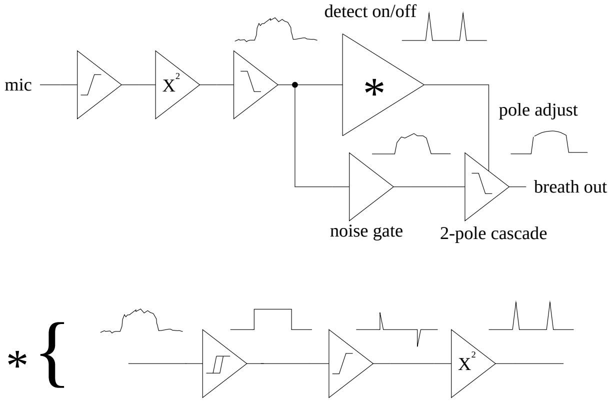

Fig. 4. Mic processing overview.

by air turbulence ceases. In-note signals are filtered to reject noise, while still offering enough response so that vibrato technique can be used. The computational load is moderate but not inappropriate in relation to the high load demanded by the synthesis software. The system suits a low pressure, relaxed blowing style. Generally, wind-controllers operate at higher pressures. The plastic insert helps keep moisture from the mic, and also shields the mic from audio feedback. Breath moisture is carried directly into the air and doesn't build up in the bore. The mouthpiece is easily cleaned by removing.

Breath Pressure Detection The second scheme uses a miniature breathsensor of the micro-silicon bridge kind that have recently become available. This is much more expensive than the condenser mic, but offers wider pressure ranges and direct pressure measurement. The sensor is conveniently mounted in a bung like the mic. The back pressure can be changed by adjusting a block fitted to the mouthpiece exhaust. By blocking the air exhaust completely the player can play without expending breath. The breath pressure is sampled and conveyed by serial link to computer. While it is convenient to combine finger and breath together on one serial line, it is found that lower latency can be achieved for breath control by using audio generated according to the noise detection scheme. A DC block and a 2-pole low pass filter with a cutoff of 400 Hz applied to the pressure sensor signal are effective in isolating a reasonably clean audio signal which can be used, for example, to implement *growling* style effects by modulating the synthesis process with the player's own voice.

2.3 Sensor Sampling

The sensor voltages are sampled and converted to midi controller messages using a low cost microcontroller microcontroller. Additional analog inputs are read via a multiplexer. MIDI provides ample bandwidth for the six finger holes. A more serious problem lies in the desktop software. The processor disruption when receiving the serial information is very high, despite the low bandwidth, and reduces the potential for software synthesis. Various compromises can be made. The microcontroller has dip switches for setting the control message blanking time and the resolution of the sample: The average bandwidth can be reduced, maintaining fast response to sudden changes at the expense of lower resolution. This reduces the ability for subtle expression, for example vibrato. Reducing the message rate to 1 per 4 ms with a resolution of 64 (6 bits) made a useful compromise for most of the software used. The interface electronics and microcontroller have been integrated into a single circuit board which fits inside a Bb penny whistle. An earlier prototype of the CyberWhistle has an external control box, which requires many more connecting wires, and introduces parasitic feedback problems in the high impedance lines.

3 Software Synthesis

3.1 The Programming Environment

Initially, real time Csound, [7], was used to prototype ideas rapidly. However the need for greater flexibility and efficiency prompted a move to C++.[3] The object programming style is particularly appropriate for physical models which reflect the hierarchical nature of real objects. It is also useful for the more general abstract structures frequently found in music, possibly because the human mind is biased towards natural structures. The inlining of small audio functions proves effective in improving speed, especially when fully optimized. The core audio code is necessarily compact, so inlining does not produce an excessively large executable; around 100 Kbyte.

I/O, Latency The best response time for incoming midi data is achieved by making a shared process sleep with a Unix *select* command until woken by a midi event. As noted earlier, it is important to limit the midi rate so as not to disrupt the main audio process too much. Using the interrupt method the best latency from sensor voltage to audio output is 10 ms. The audio buffers can be set to a minimum of 256 samples[4]. In normal free running operation with the output buffer full, the audio latency from input to output is then 6 ms.

3.2 Software Synthesis

Physical Models A range of simple physical models were constructed consisting of woodwind and string elements, [6], [1], [4], combined with some greatly simplified models for the finger holes (**Figure 5**). Even with such a minimal arrangement very interesting results can be produced by careful control of the finger holes.

---

[3] For example, the audio block size in Csound, *ksmps*, must be set to the order of 100 samples to achieve good efficiency, but this limits the minimum global feedback time and minimum audio latency. Correcting the feedback time is awkward.

[4] Only in 4 channel mode on the Silicon Graphics Indy

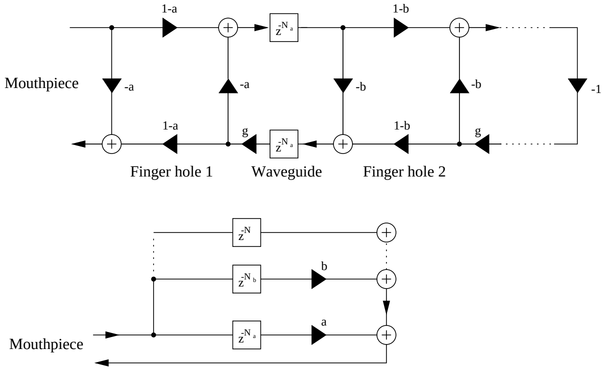

Fig. 5. Simplified bore models.

Relaxing the bore filter and extending the instruments to very low registers is effective in generating a rich spectrum. Different bore tunings lead to contrasting multiphonic effects.

Control Filter Models *Control Filtering* is a broad technique for trying to bridge the gap between traditional live synthesis systems and physical modeling. In the older systems the controls typically map directly into the audio rate synthesis engine. For example, a key triggers an envelope, or breath pressure maps onto depth of frequency modulation. The result is that the output at a given time depends only on the very recent input, and possibly some isolated previous events such as a pedal sustain. In a physical modeling instrument the combination of long delays, feedback and non-linearity ensure that the instantaneous output is dependent on a much broader *history*. The idea of control filtering is to enrich the dynamics of the response to the controls without creating an audio rate physical model. The control signals are filtered, possibly nonlinearly and nonindependently, then upsampled to drive the audio rate synthesis section. For example, take a simple linear filter consisting of DC pass with some resonance at 15 Hz. Apply a finger control through this filter to the index of an FM oscillator. The output ripples when the input is displaced.

4 Summary

The CyberWhistle demonstrates that additional effort expended in controller design may be well rewarded, even when using simple synthesis processes and inexpensive hardware. Likewise, care taken in combining the synthesis process with the continuous control parameters is worthwhile. Frequently too much emphasis is given to the complexity of the synthesis and not the overall instrument. The serial communication method seriously compromises performance, and this is a general feature of pre-emptive operating systems. Using the audio bus as an indirect communication channel is more efficient and offers lower latency.

References

1. Perry R. Cook. A meta wind instrument physical model and a meta controller for real time performance. In *ICMC proceedings*, pages 273–276, 1992.
2. Robert L Doerschuk. The life and legacy of leon theremin. *Keyboard*, pages 49–68, February 1994.
3. Stuart Favilla. The ldr controller. In *ICMC Proceedings*, pages 177–180, 1994.
4. M Karjalainen, UK Laine, T Laasko, and V Valimaki. Transmission-line modeling and real-time synthesis of string and wind instruments. In *ICMC Proceedings*, pages 293–296, 1991.
5. JA Paradiso and N Gershenfeld. Musical applications of electric field sensing. *Computer Music Journal*, 21(2):69–89, 1997.
6. Julius O. Smith. Efficient simulation of the reed bore and bow string mechanisms. In *ICMC proceedings*, pages 275–280, 1986.
7. Barry Vercoe. *CSOUND: a manual for the audio processing system*, 1992.