INVESTIGATION OF THE APPLICATION OF CONCURRENCY TO A DIGITAL SIMULATION OF THE HUMAN BASILAR MEMBRANE

K ADAMSON, G DONNAN, N D BLACK

ABSTRACT
A concurrent model of the Basilar Membrane (BM) using the Occam programming language is presented. The model is a modification of the 1-D transmission line model developed by Schroeder (ref 2). The BM is represented as a basic electrical transmission line consisting of 128 elemental sections connected in cascade, with filter centre frequencies spaced on a modified logarithmic scale, and incorporating the physiological properties of cochlear mechanics. The structure of the concurrent model is largely derived from the electrical model, indicating potential design benefits for more complex models. Using the model it is possible to simulate at each point along the membrane the waveform of the sound pressure in the cochlear fluid, as well as the deflection of the membrane itself.

FILTERBANK DESIGN
The model presented in this paper is as described by BLACK et al (ref 1) and is a modification of the 1-D transmission line model developed by Schroeder (ref 2). The simplified electrical equivalent of one elemental section of length $\Delta x$ is shown in Fig 1. The different passive components represent different physical processes performed in the cochlea. For instance, $M(x)$ is an inductance representative of the mass of fluid which moves longitudinally or parallel to the basilar membrane. $L(x)$ is an inductance representing the basilar membrane mass and associated structures and laterally moving fluid. $R(x)$ is a resistance representing losses of the membrane and $C(x)$ is the capacitance which represents the compliance of the basilar membrane (BM) and it is the charge developed across the capacitor which is analogous to the displacement of the BM.

The dynamic behaviour of the BM is modelled by the serial connection of $N$ elemental sections, with the value of the passive parameters determining the position $X$ along the membrane. As $N \to \infty$ the behaviour of the cascade connection will approach that of the true distributed system. In order to make the computational task more tractable, $N$ is restricted to 128. Each elemental section is considered to work in isolation from its neighbours by the inclusion of a buffer in series with $M(x)$ and an extra impedance, $Z_t$, appearing across $R(x)$, $L(x)$, and $C(x)$ to represent the input impedance of the remainder of the membrane. This simplification results in the voltage transfer function of an isolated section being given as:

$$\frac{V_o(s)}{V_i(s)} = K \frac{a}{s+a} \frac{W_p^2}{s^2+b_z s+W_z^2}$$

where $s$ is the complex frequency variable; $K$ is the attenuation factor; $a$ is the low pass pole; $W_p$ and $W_z$ are the resonant pole and zero frequencies respectively; $b$ is the pole and zero bandwidths respectively.

Department of, *Computing Science,** Electrical & Electronic Engineering
University of Ulster at Jordanstown
This transfer function results in a low pass filter response. The displacement of the BM is analogous to the charge across the capacitor however, and its transfer function, given in equation (2), does not exhibit the resonant zero term.

\[
\frac{V_m(s)}{V_i(s)} = \frac{K}{s + a} \cdot \frac{w^2}{s^2 + b p s + w^2}
\]  

The digital filter model of the BM is obtained by transforming the analogue filter equations (1) and (2) into their equivalent digital filter form. Since it is important to maintain the time structure of equations (1) and (2) in the digital implementation the analogue to digital transformation is achieved through the impulse invariant transformation. This gives the digital pressure and displacement transfer functions as in equations (3) and (4) respectively.

\[
\frac{V_p(z)}{V_i(z)} = \frac{K \cdot (1-a_0)}{(1-a_0 \cdot z^{-1})} \cdot \frac{(1-b_1 + b_2)}{(1-b_1 \cdot z^{-1} + b_2 \cdot z^{-2})} \cdot \frac{(1-a_1 \cdot z^{-1} + a_2 \cdot z^{-2})}{(1-a_1 + a_2)}
\]  

\[
\frac{V_m(z)}{V_i(z)} = \frac{K}{(1-a_0 \cdot z^{-1})} \cdot \frac{(1-b_1 + b_2) \cdot z^{-1}}{(1-b_1 \cdot z^{-1} + b_2 \cdot z^{-2})}
\]

where \( z \) is a complex variable and \( a_0, a_1, b_1 \) and \( b_2 \) are the digital filter coefficients. From equations (3) and (4) it can be seen that the displacement transmission function is contained in the pressure transfer function, thus a simple cascade arrangement is possible. The digital realisation of one elemental section of the filterbank is shown in Fig 2.

**CONCURRENT DESIGN**

The sequential programming model (ref 1) of the filterbank reflects the view that a specific sample is processed to completion along the entire length of the BM, before the succeeding sample can be interpreted. At any point in time therefore, only one filter in the filterbank is actively transforming a sample. This is in contradistinction to the digital filter design (Fig 2), where clearly any filter can engage in activity as soon as \( V_i \) is available. Although a particular sample \( S \) is processed serially across the filterbank, at any instant 127 filters are free to process samples preceding and succeeding \( S \). Achievement of this degree of concurrency in software and hardware, could raise computational capacity by a factor of 128 for the maximum number of samples-2, against an equivalent sequential implementation. Our design strategy as attempted elsewhere (ref 3), is to determine an initial maximal process structure and through subsequent stages of process collapsing, realise an implementation of optimum performance. We describe the process structure using a notation (ref 4) developed for languages supporting the principles of CSP (ref 5).

The digital filter design (Fig 2) serves as an unusual software specification. As shown in Fig 3, the filterbank is structured as a FILTER process, replicated as appropriate, each FILTER communicating via an input and output channel called sample. Each FILTER (Fig 4) is structured as two processes, DIGITALFILTER representing the actual digital filter (Fig 2) and DATACOLLECTOR whose purpose is to collect the displacement. A FILTER at this level of operation accepts \( V_i \) and produces \( V_p \) on sample. The calculated displacement of the membrane for the respective elemental section is communicated to DATACOLLECTOR on a channel called displacement. The DIGITALFILTER is structured as 3 processes, GAINER, LOWPASS, and BANDPASS, related one to another as
illustrated in Fig 5. Their correspondence with the digital filter design and the functionality attributed to each process is indicated by the dashed lines in Fig 2.

Development of the GAINER is straightforward, requiring the transformation of the real number input on sample, through multiplication by the coefficient K, and its output on the channel gainitolow. All gains are similarly implemented throughout the model.

The development of LOWPASS necessitates the implementation of an adder and a delay. Gains Go and Ao are processes, both communicating a transformed sample to the ADDER process, as expressed in Fig 6. The delay is implemented in the ADDER by utilising the synchronous communication properties of Occam. To effect a one sample delay, the ADDER inputs on gainGotoadd, and outputs on the channels lowtohigh and addtogainAo. For subsequent samples, when the ADDER inputs a sample S on gainGotoadd, it inputs sample S-1 on gainAotoadd and adds them. Process GAINAo thus acts as both a gain and a synchronised sample buffer. Note that it is the ADDER which determines the delay, by not inputting on gainAotoadd until a sample is available on gainGotoadd. Synchronisation is thereby implemented in software, allowing the time required to produce a sample in some FILTER F and in GAINAo within FILTER F+1 to be different, without any effect on the logic of a FILTER. This is essential when targetting to multiple hardware processors which are not synchronised by a common clock. BANDPASS is a more complex organisation of the design of LOWPASS and is similarly constructed from gains and adders. Because of both forward and backward, single and double sample delays and a displacement output, a DISTRIBUTOR process is introduced to disperse the same sample to GAINBl, BAINB2, BAINA2, GAINAl and on displacement.

Besides the concurrency of FILTERs, there is a high degree of concurrency internal to each FILTER. Indeed it would be possible, though unlikely, for BANDPASS to be processing sample S+2 in GAINGz, adding sample S+1 in ADDER2, adding sample S in ADDER1 and gaining sample S-1 in GAINOp. This is logically feasible as a consequence of synchronous communication, which also ensures synchronisation across the entire filterbank, irrespective of the number of transputers used and the process distribution. The scope for performance adjustment is wide and is a question of software efficiency, transputer volume and the process/transputer topology.

RESULTS AND FURTHER WORK

The filter frequency response can be made to closely resemble that measured on the BM by correct choice of the filter coefficients. Fig 7 shows the frequency response of the bank at 7kHz with the filter coefficients chosen so as to match the data measured by Rhode (ref 6). Fig 8 shows the membrane displacement due to a 2 kHz sine wave 20ms after onset. A single FILTER implemented on a T414, processes 2000 samples in approximately 1.4 seconds. Due to software implemented floating point arithmetic on the T414, a T800 is essential, which is calculated to execute in approximately 0.9 seconds. A considerably more efficient model is currently being derived by process collapsing and transputer topology experiments, and it is expected that a time of around 0.1 seconds could be achieved.

REFERENCES

1. N D Black, E Ambikairajah, R Linggard, K Adamson, To be published
Fig (1) TOP, An Electrical Section of the BM

(2) ABOVE, Digital Filter Realisation

Fig (4) TOP, (3) ABOVE
FILTER, FILTERBANK process structure

Fig (5) DIGITALFILTER process structure

Fig (7) TOP, Frequency Response at 7kHz against measured data by Rhode
(8) ABOVE, Displacement due to a 2kHz tone at the Stapes after 20ms
(6) RIGHT, LOWPASS process structure