DIGITAL AUDIO: recursive digital filtering for high quality audio signals

G.W. McNally, B.Sc., C.Eng., M.I.E.E.
DIGITAL AUDIO: RECURSIVE DIGITAL FILTERING
FOR HIGH QUALITY AUDIO SIGNALS
G.W. McNally, B.Sc., C.Eng., M.I.E.E.

Summary

A high speed, low cost digital signal processor has been designed and
built suitable for real-time operations on high quality sound signals. The purpose
of the work described in this Report is to examine its application to audio filters
of a kind currently found in equipment such as mixing desks. The methods by
which digital audio filters may be designed are reviewed and a detailed study is
presented of the many ways in which the design may be realised and
implemented. The Report describes two structures in detail which were identified
as providing high performance with ease of implementation. The effects of
coefficient quantisation and round-off noise are quantified by computer
simulation of the hardware structure, and by these means, a specification for the
required word-lengths is determined. An example is given of a four filter cascade
for audio equalisation with a suitable implementation in programmable
hardware. The filters are controlled by flexible commands issued on an
internationally standardised communications bus (IEEE-488). The total
processing involved in these techniques is conveniently handled by a machine
such as COPAS-2D – the subject of a companion Report.

Issued under the authority of

[Signature]

Research Department, Engineering Division,
BRITISH BROADCASTING CORPORATION

December, 1981
PH-227

Head of Research Department
# DIGITAL AUDIO: RECURSIVE DIGITAL FILTERING
## FOR HIGH QUALITY AUDIO SIGNALS

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Summary</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Introduction</strong></td>
<td>1</td>
</tr>
<tr>
<td>1.</td>
<td>The advantages of digital filtering of audio signals</td>
<td>1</td>
</tr>
<tr>
<td>2.</td>
<td>Review of digital filter design techniques</td>
<td>3</td>
</tr>
<tr>
<td>2.1.</td>
<td>Calculation of filter coefficients</td>
<td>3</td>
</tr>
<tr>
<td>3.</td>
<td>Realisation of filters</td>
<td>5</td>
</tr>
<tr>
<td>3.1.</td>
<td>Direct forms</td>
<td>5</td>
</tr>
<tr>
<td>3.2.</td>
<td>Cascade form</td>
<td>8</td>
</tr>
<tr>
<td>3.3.</td>
<td>Parallel form</td>
<td>8</td>
</tr>
<tr>
<td>3.4.</td>
<td>Continued fraction expansion structure</td>
<td>8</td>
</tr>
<tr>
<td>3.5.</td>
<td>Ladder and lattice structures</td>
<td>9</td>
</tr>
<tr>
<td>3.6.</td>
<td>Wave structures</td>
<td>9</td>
</tr>
<tr>
<td>3.7.</td>
<td>Discussion on the choice of structure</td>
<td>9</td>
</tr>
<tr>
<td>4.</td>
<td>Realisation of the biquadratic section</td>
<td>9</td>
</tr>
<tr>
<td>4.1.</td>
<td>Transpose configurations</td>
<td>10</td>
</tr>
<tr>
<td>4.2.</td>
<td>Coupled forms</td>
<td>11</td>
</tr>
<tr>
<td>4.3.</td>
<td>Agarwal-Burrus structures</td>
<td>12</td>
</tr>
<tr>
<td>4.3.1.</td>
<td>Structure 1AB</td>
<td>12</td>
</tr>
<tr>
<td>4.3.2.</td>
<td>Structure 2AB</td>
<td>12</td>
</tr>
<tr>
<td>4.4.</td>
<td>Error feedback</td>
<td>13</td>
</tr>
<tr>
<td>4.5.</td>
<td>Floating-point arithmetic</td>
<td>13</td>
</tr>
<tr>
<td>4.6.</td>
<td>Scaling</td>
<td>15</td>
</tr>
<tr>
<td>5.</td>
<td>The effects of finite wordlength in digital filters</td>
<td>15</td>
</tr>
<tr>
<td>5.1.</td>
<td>Coefficient quantisation</td>
<td>15</td>
</tr>
<tr>
<td>5.2.</td>
<td>Roundoff noise</td>
<td>17</td>
</tr>
<tr>
<td>5.2.1.</td>
<td>Transposed DF2 structure</td>
<td>17</td>
</tr>
<tr>
<td>5.2.2.</td>
<td>2AB structure</td>
<td>19</td>
</tr>
<tr>
<td>5.2.3.</td>
<td>Computer simulations of roundoff noise in filter</td>
<td>19</td>
</tr>
<tr>
<td>5.3.</td>
<td>Limit cycles</td>
<td>21</td>
</tr>
<tr>
<td>6.</td>
<td>A four filter cascade for audio equalisation</td>
<td>22</td>
</tr>
<tr>
<td>7.</td>
<td>Hardware implementation of the filters</td>
<td>25</td>
</tr>
<tr>
<td>8.</td>
<td>The control program</td>
<td>26</td>
</tr>
<tr>
<td>9.</td>
<td>Conclusions</td>
<td>27</td>
</tr>
<tr>
<td>10.</td>
<td>Acknowledgements</td>
<td>28</td>
</tr>
<tr>
<td>11.</td>
<td>Appendix — software summary</td>
<td>28</td>
</tr>
<tr>
<td>12.</td>
<td>References</td>
<td>28</td>
</tr>
</tbody>
</table>
DIGITAL AUDIO: RECURSIVE DIGITAL FILTERING FOR HIGH QUALITY AUDIO SIGNALS

Introduction

This Report examines the problems of realising high performance digital filters for use in an audio programme signal chain. Previous work\(^1\) introduced design theory and procedure for digital filters of a type suitable for use in a proposed digital mixing desk. Equations were developed by which the digital filter coefficients could be calculated directly from a simple filter specification. A hardware structure was suggested for implementing the filter in real-time and the methods were validated using equipment based on a high speed commercial microcomputer. A second Report\(^2\) described a high speed processor known as COPAS, designed specifically for real-time audio processing, and which by virtue of its programmability and increased processing power, permitted further study of digital filtering methods. In this Report, degradation of the filter performance by finite word-length arithmetic is examined in detail and new filter structures are investigated. The objective was to realise four "equalising" filters in cascade, maintaining the 16 bit performance of the digital signal chain for a wide range of individual filter combinations. The properties of several different structures are examined and the reasons behind the final choices are discussed. The performance of the resulting structures is analysed in some detail and observations are made on the practicalities of implementing the structures in programmable hardware.

1. The advantages of digital filtering of audio signals

Digital audio techniques are playing a growing part in broadcasting and sound recording. In the BBC, the first applications were in transmission and more recently in recording. But at the moment, digital recordings must be converted to analogue form before they can be processed in any way, whether it be balancing a multitrack recording or simply replaying a recording into the network. To maintain the advantages of digital working, it is clearly desirable to perform all these intermediate functions digitally as well. One of the most often used, and perhaps more complex, of these functions is equalisation. In even a simple mixing desk, it has become standard practice to provide a large number of equalisation options. Typically, in a single processing channel of a mixing desk, (Fig. 1), there may be some or all of the following controls:

- Low and high pass filters – each with choice of cut off frequencies and attenuation at up to 12 dB/octave outside the passband.
- Low and high frequency shelving filters – these provide a lift or cut to a range of frequencies. The shelving frequency and lift/cut can be specified.
- Presence/absence filters – these provide a peak or notch at a selected frequency and boost/cut. Two or more independently controlled presence/absence filters may be supplied with limited control of the \(Q\) of each.
- 50 Hz and 100 Hz notch filters.

It is often required to use more than one of these filters simultaneously and analogue processing
channels can usually provide many of the above facilities at the same time. In Figs 2 to 7, the frequency characteristics of a number of these equalisers are shown. It has been found that the best subjective results are obtained when the Q of the filter is modified slightly as the gain or frequency is changed. This is shown clearly in the characteristics of Figs 2, 3. In Figs 4, 7 the result of large variations in Q is shown for presence and shelving filters respectively; while these particular characteristics are not commonly found, they can in principle be realised using sufficiently stable equalisers. Figs 5 and 6 show shelving filters with 12 dB/octave slopes though 6 dB/octave is perhaps more common.

Fig. 2—Family of presence filters. \( Q_{0.25} \) varies 4.5 to 9.0

Fig. 3—Family of presence filters. \( Q_{0.25} \) varies 5.0 to 0.25

Fig. 4—Family of presence filters. \( Q_{3.0} \) varies 0.75 to 10.0

Fig. 5—Low frequency shelving filters. \( Q_{3.0} = 0.71 \)

Fig. 6—High frequency shelving filters. \( Q_{3.0} = 0.71 \)

Fig. 7—Low frequency shelving filters. \( Q_{0.25} \) varies
The method by which the $Q$ factor of these characteristics is defined is explained in Section 2.1.

In analogue circuitry, great care has to be taken to realise all these characteristics while maintaining high signal quality. The result is circuitry of relatively high cost for which the task of remote programming of the equaliser settings is extremely difficult. This explains, in part, the historical necessity for the circuitry in mixing desks to be very close to the control knobs and the resulting large size of the desks. If the filters are instrumented digitally, then some new advantages accrue. Firstly, the characteristic is determined solely by a small number of coefficients – thus, new equalisers can be installed simply by changing the coefficients. The same digital filter can be used for low pass, shelving and presence characteristics. Secondly, the filter is by its nature, remotely programmable – a highly desirable feature operationally since it means that assignable control systems can be used and the hardware and ergonomic aspects of studio equipment design can be separated. Finally, it is practical where required to produce linear phase characteristics using transversal filtering techniques – if programmable hardware is used then even the same circuitry can be used.

Added to this, of course, are the well known reasons for choosing a digital system, namely reliability, precision and reducing costs.

2. Review of digital filter design techniques

It will be shown* here that the specifications described in the previous Section can be substantially met by a single digital filter structure in which the chosen characteristic is determined by an appropriate set of coefficients. The transfer function for an analogue filter producing these characteristics is a biquadratic function having two poles and two zeros and can be expressed in terms of the Laplace transform as described in Ref. 4.

$$H(s) = \frac{s^2 + 2\varepsilon_1 s + \omega^2_0}{s^2 + 2\varepsilon_1 s + \omega^2_0}$$

The corresponding digital filter can be derived by means of the $z$ transform and the general form of this new transfer function is

$$H(z) = \frac{1 - 2r_1 \cos \phi_1 z^{-1} + r_1^2 z^{-2}}{1 - 2r_2 \cos \phi_2 z^{-1} + r_2^2 z^{-2}}$$

where $r_1$, $\phi_1$, and $r_2$, $\phi_2$ represent the positions of the poles and zeros in the $z$ plane using polar coordinates and $z^{-1}$ is the transform operator which corresponds to a unit delay in the discrete-time sequence. This transfer function can be directly implemented using multipliers, adders and delay elements by various means discussed in Sections 3 and 4.

Rewriting

$$H(z) = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2}}{1 + b_1 z^{-1} + b_2 z^{-2}}$$

it can be seen that the coefficients

$$a_1 = -2r_1 \cos \phi_1$$
$$a_2 = r_1^2$$
$$b_1 = -2r_2 \cos \phi_2$$
$$b_2 = r_2^2$$

The coefficient $a_0$ controls the overall gain through the filter when $a_1$ and $a_2$ are scaled accordingly. A further scaling, arranged to be a factor of two, may be performed at the filter output to make best possible use of the available dynamic range. The five coefficients and scaling factor can be chosen to provide all the characteristics described in the previous Section.

2.1 Calculation of filter coefficients

Consider a specification for a presence filter (see Fig. 8a) in which the amplitude characteristic is described by a centre frequency, $\omega_0$; the boost/cut required, $A$; and the sharpness of the peak/notch by a factor, $Q$. So that characteristics which, for example, peak by only 1 dB can be accommodated, this $Q$ factor will be defined as $\omega_0/\Delta \omega$, where $\Delta \omega$ is the width of the characteristic at a point where the gain has changed by $\Delta f$ dB from its extreme value.

These characteristics are realised by confining the root locations in the $s$ plane (Fig. 9a) to

zeros at: $\omega_0 \exp(\pm j\theta_1)$
poles at: $\omega_0 \exp(\pm j\theta_2)$

The bold lines indicate the locus of the root positions as $\theta_1$, $\theta_2$ vary. The transfer function thus simplifies to

$$H(s) = \frac{s^2 + 2\omega_0 \sin \theta_1 s + \omega^2_0}{s^2 + 2\omega_0 \sin \theta_2 s + \omega^2_0}$$

* See Ref. 3 for an introduction to the basic theory.
It has been shown\(^1\) that the boost/cut at \(\omega_0\) is
\[
A = \sin \theta_1 \sin \theta_2
\]
and that if \(F\) is the gain corresponding to a \(\Delta f\) dB change from the extreme value, then
\[
Q = [(1 - F)/4(F \sin^2 \theta_2 - \sin^2 \theta_1)]^{1/2}
\]
It can be seen that \(\theta_1, \theta_2\) control the depth of the peak/notch and its \(Q\) while \(\omega_0\) sets the frequency scale. Now the positions of the poles/zeros in the \(s\) plane can be calculated from
\[
\begin{align*}
\sin \theta_1 &= \frac{A}{2Q} \left[ \frac{1 - F^2}{F^2 - A^2} \right]^{1/2} \\
\sin \theta_2 &= \frac{1}{2Q} \left[ \frac{1 - F^2}{F^2 - A^2} \right]^{1/2}
\end{align*}
\]
For low values of \(Q\) with \(A^2\) approaching \(F^2\); \(\sin \theta_1, \sin \theta_2\) can exceed unity. For this condition the poles/zeros lie on the negative real axis of the \(s\)
plane and if \( \sin \theta = x, (x > 1) \) then the singularity positions are at \(-[x \pm (x^2 - 1)^{1/2}]\omega_0\)

These singularities are then mapped into the \(z\) plane using the transform

\[ z = \exp(-sT) \]

where \(T\) = the sample period and thus the coefficients \(a_0, a_1, a_2, b_1, b_2\) are calculated.

A similar procedure is followed for the design of shelving filters. In this case, the specification (Fig. 8b) is described by the boost/cut, \(A\); the shelving frequency, \(\omega_b\), defined at a point \(\Delta f\) dB from its extreme value and a \(Q\) factor to determine the steepness of the characteristic. These characteristics are obtained by confining the root locations in the \(s\) plane (Fig. 9b) to:

zeros at: \(w_q \exp(\pm j\theta)\)

poles at: \(\omega_p \exp(\pm j\theta)\) where \(\sin \theta = 1/2Q\)

The amplitude characteristic is solved at \(\omega = \omega_b\) to give the required gain and using the relationship that at zero frequency

\[ A = \frac{\omega_b^2}{\omega_p^2} \]

the values of \(\omega_q\) and \(\omega_p\) are calculated. For a given value of \(Q\), the pole/zero locations are identified and the digital filter coefficients calculated as before.

Low pass and high pass characteristics can be generated using Butterworth filters. A wealth of literature on filter design can be used to determine the pole locations from a given specification. However, before transforming into the equivalent digital filter it is desirable to introduce zeros in the stopband the same in number as the poles to avoid producing extremely large attenuations in the stopband. This avoids the generation of very large/small numbers in the internal arithmetic of the digital filter.

With the parameters of the digital filter identified, it is now possible to consider the best methods of realising the filter in hardware.

3. Realisation of filters

Realisation is the term used for the process of converting a transfer function into a filter network or structure. In principle, there are many distinct structures which can be used to implement a given filter specification, though the properties of these structures may vary widely. The choice of structure will be influenced by such factors as sensitivity to coefficient quantisation, level of output noise due to arithmetic rounding or truncation, computational efficiency, number system used, and type of filter. It is not possible to generalise about which structure is most suitable for given problem types and so in the following Sections, a brief review is given of the different realisations that have been considered. In this Section the realisation of a general \(N\)th order filter is considered. Throughout the Section, \(x(n)\) and \(y(n)\) denote the \(n\)th sample at the input and output of a filter.

3.1 Direct forms

These structures obtain their name from the direct relationship between the structure and the difference equation or transfer function when expressed as a ratio of polynomials in \(z\)

\[ H(z) = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_{N-1} z^{-(N-1)} + a_N z^{-N}}{b_0 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_{N-1} z^{-(N-1)} + b_N z^{-N}} \]

Each coefficient in the transfer function \(H(z)\) appears directly as a multiplier coefficient and the corresponding structure, known as Direct Form 1 (DF1) is shown in Fig. 10. It uses separate delays for the inputs and outputs and it has been shown\(^6\) that when poles of the filter transfer function lie close to each other or the unit circle in the \(z\) plane, there is a severe coefficient sensitivity problem with this structure. Because the input is processed first by the \(a_0, a_1, a_2\) multipliers, this is sometimes called a zeros-before-poles realisation.

Re-examining the transfer function above, a second structure can be derived known as Direct Form 2 (DF2) and is shown in Fig. 11. It is sometimes called the canonic form because it has the minimum number of multiplier, adder and delay elements though as will be seen other structures may also possess this feature. The properties of this structure are similar to those of DF1, i.e. high sensitivity to coefficient quantisation and the generation of a high level of arithmetic roundoff\(^7\) noise.

The direct forms are popular hardware realisations because of the high degree of parallelism\(^17\) that these structures offer. Referring to Figs 10 and 11 it can be seen that the inputs to each multiplier in the structure do not depend on a previous calculation. Thus, if the hardware is so designed, all multiplications may take place simultaneously and the

\(^*\)The term roundoff noise will be used to describe rounding or truncation unless otherwise stated.
Fig. 10—Direct form 1 filter structure

Fig. 11—Direct form 2 filter structure

Fig. 12—Cascade form filter structure

Fig. 13—Parallel form filter structure
Fig. 14—Continued fraction expansion filter structure (type 1B)

Fig. 15—One-multiplier lattice filter structure
highest possible throughput is obtained. Filters operating at sampling rates in excess of 2 MHz have been reported using this technique. Alternatively, each multiplication can be associated with an addition and the calculation performed as a series of multiply-accumulate operations. This is well suited to existing special purpose LSI circuitry.

Of the two structures DF1 is to be preferred since it is resistant to overflow problems. For a filter with an input and output of magnitude less than unity, the inputs to all multipliers can be expressed in fractional arithmetic. This cannot be guaranteed in the DF2 structure. If modulo arithmetic such as two's complement is used, the partial sums of products may be allowed to overflow since it is known that the final output will be within the range of the number system used.

3.2 Cascade form

This is the first method by which the problems of coefficient sensitivity and truncation noise may be reduced. The transfer function must be arranged so that it may be factorised and represented by a cascade of first and second order sections, thus

$$H(z) = \prod_{i=1}^{K} H_i(z)$$

where $H_i(z)$ is either a first or second order section of the form

$$H_i(z) = \frac{a_{0i} + a_{1i}z^{-1} + a_{2i}z^{-2}}{1 + b_{1i}z^{-1} + b_{2i}z^{-2}}$$

The corresponding structure is shown in Fig. 12 and each of the first or second order networks can be realised using second order canonical direct forms or an equivalent structure. It suffers less from coefficient sensitivity problems than the direct forms but requires close attention to scaling between sections to optimise accuracy in fixed point applications. It may also be necessary to control the ordering of sections and the pairing of the poles and zeros for best results.

3.3 Parallel form

A fourth structure can be obtained from the partial fraction expansion of the $z$ transform of the filter, $H(z)$

$$H(z) = \sum_{i=1}^{K} H_i(z)$$

where $H_i(z)$ is a first or second order section of the form,

$$H_i(z) = \frac{a_{0i} + a_{1i}z^{-1}}{1 + b_{1i}z^{-1} + b_{2i}z^{-2}}$$

This parallel structure is shown in Fig. 13 and each of the sections can be realised using the direct form or an equivalent. This structure produces a lower level of roundoff noise than cascade structures.

3.4 Continued fraction expansion structure

A further method of realising a transfer function is by means of the continued fraction expansion. The original transfer function in the form of a ratio of polynomials, can under certain conditions be expanded\(^{11}\) as:

$$H(z) = \frac{1}{c_1 + \frac{1}{c_2 z^{-1} + \frac{1}{\cdots + \frac{1}{c_N z^{-1}}}}}$$

By expressing the transfer function in this form it can be seen that a new structure can be generated using building blocks of the form,

$$G_1(z) = \frac{1}{B z^{-1} + T_1(z)}$$

$$G_2(z) = \frac{1}{A + T_2(z)}$$

These simple first order sections can then be connected as shown in Fig. 14. The resulting filter is canonical in delays and multipliers, i.e. for an $N$th order system, the structure uses $N$ delays and no more than $(2N + 1)$ multipliers. One advantage of the technique is that it tends to produce coefficient values with a small range of values, but it has been shown\(^{7}\) that these structures do tend to generate a high level of product-quantisation noise.

3.5 Ladder and lattice structures

This class of structures is dissimilar to those discussed so far in that they are not generated by means of manipulation of the transfer function. Synthesis procedures have been developed which are closely related to the conventional two-port network theory used in the design of analog filters, and methods have been described for synthesising
ladder or lattice structures from any stable direct form,\textsuperscript{12} (the terms ladder and lattice refer to the topological properties of the resulting network). Several different structures have been suggested including one in which each two-port section incorporates only one multiplier and delay. (Fig. 15). In general, the well known attributes of ladder networks in analogue filters are retained in the digital realisation giving low coefficient sensitivity and superior roundoff or truncation noise characteristics. The structures have been widely applied where high accuracy with limited wordlength is required and specifically, in linear prediction speech analysis where the reflection coefficients most suited to "transmission" are precisely those used in implementing the filter.\textsuperscript{13}

3.6 Wave structures

These structures have been developed to take advantage of the inherently low sensitivity of terminated LC analogue filters\textsuperscript{14}. In this approach, an analogue LC filter meeting the required specifications is first designed and then transformed to a digital filter by means of replacing the analogue elements by appropriate digital realisations. Difficulties in transforming signal flow diagrams into a digital network are resolved by using wave flow diagrams, in the same way as in classical scattering parameter theory\textsuperscript{14}. Just as there are many ways to synthesise analogue filters, there are also many ways to synthesise the corresponding digital filters. However, it has been shown\textsuperscript{15} that these filters will in general require a larger number of multipliers and a much larger number of adders and delays. This feature is offset by the reduced sensitivity of wave digital filters and the corresponding smaller wordlengths. The great complexity of these filters makes them impractical for high speed real-time work and for this reason an example structure is not given here.

3.7 Discussion on the choice of structure

It has been shown that many distinct structures are possible for a given filter specification of order \( N \). It is now necessary to impose some practical constraints on the design in order to narrow down this choice.

As a result of previous work it was decided to implement the structure using a second generation digital audio signal processor – COPAS-2D. The design of this processor is described in detail in a companion Report\textsuperscript{16} and was a parallel development with this investigation of filter structures. Nevertheless, certain features were identified at the start: for example, that it would use fixed point 2's complement notation and by nature of its programmability would implement filter structures using an orderly sequence of instructions. As a result the degree of parallelism offered by each structure is not of immediate interest.

A restriction in the choice of structures is imposed by the nature of the filtering required. In Section 1 current analogue practices were described and the need was shown for a number of separate filters which are controlled independently. In Section 2 it was shown that each of the filter types, e.g. low pass, high pass, presence, absence etc. can be realised by a single biquadratic section. In total, four to six such sections are needed to provide versatile control of equalisation. Clearly, it is highly desirable that a change in just one of the sections should not involve recalculation of all the coefficients of the whole filter structure. This would be the case in all but the cascade form. For a fixed equalisation characteristic, this structure may be inferior to the ladder or wave structures in terms of coefficient sensitivity and roundoff noise performance. Nevertheless, the most practical approach was seen to be the cascade form and it remains to determine the most efficient means of realising each biquadratic section. The choice is made with an awareness of problems covered only superficially in this Report (see Sections 5.2 and 5.3) such as limit cycle effects and dynamic range handling.

A brief summary of these structures is included here for completeness.

(1) Direct and continued fraction structures, particularly of high order, tend to be very sensitive to coefficient quantisation.\textsuperscript{17}

(2) Cascade, parallel and wave structures have similar coefficient sensitivity properties for fixed-point arithmetic.\textsuperscript{7}

(3) Direct and continued fraction structures tend to generate a high level of roundoff noise.\textsuperscript{7}

(4) Parallel structures generate a lower level of roundoff noise than cascade structures.\textsuperscript{10}

(5) Certain ladder and lattice structures have good coefficient sensitivity and roundoff noise performance.\textsuperscript{12}

(6) Wave structures are less sensitive to coefficient quantisation and have good roundoff noise properties but entail an excessive number of additions.\textsuperscript{14}

4. Realisation of the biquadratic section

This Section is concerned with the realisation of
the biquadratic transfer function,

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

The conditions for stability are conveniently summarised in Fig. 16 - the stability triangle, i.e.,

\[ (b_2 + 1) > \pm b_1 \]
\[ |b_2| < 1 \]

 Canonical realisations of such a transfer function will use four, two-input adders, five multipliers and two delays. There are many different realisations that could be used (15 are described in Ref. 18) but the structure most commonly associated with this transfer function is shown in Fig. 17a. Although these structures are equivalent for a given linear discrete filter, they are no longer equivalent when the effects of coefficient quantisation and roundoff noise are considered. The selection of the appropriate structure is the principle factor determining the character of the output roundoff noise and hence determines the word-length required to meet a given performance specification.

4.1 Transpose configurations

Transpose configurations are a useful family of digital filter structures that are simply generated from any structure. It has been shown,\(^\text{15}\) that the roundoff noise and dynamic range limitations of these new structures are quite different from the original structures and can be used to good effect in filter design. Fig. 17b shows the transpose digital filter generated from the Direct Form 2 structure of 17a. The signal flow graph representations in Fig. 18 shows the operation of the following rules for generating the transpose.

1. Reverse all the directions of the branches in the network.

Fig. 17—Equivalent representations of a digital biquadratic section
(2) Change directions of all the delays and multipliers – keeping their values constant.
(3) Replace branch nodes by summation nodes and vice-versa.

The resulting structure of 17b has significant advantages over the DF2 structure and combines the best features of a DF1 structure with canonic forms. The nodes at which overflow must be prevented (because the result of an addition is presented to the input of a multiplier) are indicated by * in Fig. 17. When two’s complement arithmetic is used in the transposed DF2 structure and the gain from filter input to output is always less than unity, then the partial sums written into each filter store may be allowed to overflow since the final summation at the output produces a value of magnitude less than unity. Additionally, it is easy to arrange that roundoff errors occur predominantly at just one node at the output of the filter, a desirable property for the minimisation of limit cycles. The different structures that transposition allow give an additional degree of freedom in hardware design, particularly in choosing register lengths and bus structures.

4.2 Coupled forms

This is one example of a non-canonic structure which can yield certain advantages. As will be seen in Section 5.1, the canonic structures lead to a highly non-uniform density of pole positions in the z plane due to coefficient quantisation. This can lead to gross inaccuracies or in the limit, instability in the filter responses produced, particularly when the poles/zeros are near the unit circle of the z plane. In the coupled form proposed in Ref. 20, the quantised pole positions lie on an even rectangular grid in the z plane and so the maximum error in pole position due to finite precision is the same for all regions of the z plane (see Fig. 27). The second order structure proposed contained just one zero and this has been developed to the general biquadratic case. An example of this structure is shown in Fig. 19 and the extra computation involved is clearly evident. In this case the transfer function realised is

\[ H(z) = \frac{c_1^2 + (c_3c_2^2 + c_4c_1)z^{-1} - c_4c_1z^{-2}}{1 - (2 - c_1c_2 - c_1^2)z^{-1} + (1 - c_1c_2)z^{-2}} \]

Fig. 18—Signal flow graph representation of Fig. 17

Fig. 19—Coupled form
4.3 Agarwal-Burrus structures

These structures have been proposed\textsuperscript{22} as an efficient solution to the realisation of digital filters for which poles lie close to the unit circle and near \(z = (1, 0)\). This is an important condition since it corresponds to a filter with moderately high \(Q\) and a pole frequency which is small compared with sampling frequency. For example, this would correspond to a low pass filter at 100 Hz with a sampling frequency of 32 kHz. The method can therefore be viewed as an attempt to optimise the filter structure to the desired transfer function.

4.3.1 Structure 1AB

Consider the denominator of the biquadratic transfer function.

\[
H(z) = \frac{1}{1 + b_1 z^{-1} + b_2 z^{-2}}
\]

where

\[
b_1 = -2r \cos \theta
\]
\[
b_2 = r^2
\]

For the conditions described above, \(r\) is very close to 1 and so \(b_2\) will also be very near 1. When \(\theta\) is also very small then \(b_1\) will be very near 2. For a given wordlength it is more efficient to realise \(b_2\) as \((1 - b_2')\) and \(b_1\) as \((2 - b'_1)\) where \(b'_1, b'_2\) are small positive quantities. The transfer function is then realised as

\[
H(z) = \frac{1}{1 + (2 - b'_1)z^{-1} + (1 - b'_2)z^{-2}}
\]

The resulting structure is abbreviated 1AB and is shown in Fig. 20. Note that the multiplication by 2 would be achieved with a shift operation. The coefficients \(b'_1, b'_2\) can be expressed accurately by, for example, selecting \(n_1\) so that

\[
b'_1 \cdot 2^{n_1} < 1
\]

and following the multiplier with a bit shifter to accomplish \(2^{-n_1}\). See Fig. 21. Such an arrangement results in multiplier input wordlengths which are shorter than would otherwise be required.

4.3.2 Structure 2AB

A second approach to improving filter performance as the poles near \(z = (1, 0)\) involves a transformation. The frequency response and stability of the filter is determined by the distance of the poles from \(z = (1, 0)\) for this condition. When the filter is realised with \(z = (0, 0)\) as the origin, a small relative change in the distance of the poles from \((0, 0)\) appears as a large relative change when viewed from \((1, 0)\). If we define a \(\hat{z}\) plane in which \(z = (1, 0)\) is used as the origin, this problem does not occur. The transformation is given by

\[z = \hat{z} + 1\]

To realise this filter, it is necessary to consider first the realisation of \(\hat{z}^{-1}\). The equation above can be written

\[
\hat{z}^{-1} = \frac{z^{-1}}{1 - z^{-1}}
\]

Fig. 20—Agarwal-Burrus structure (type 1AB)

Fig. 21—Maintaining accuracy of coefficient multipliers
and this can be easily realised as in Fig. 22. From the block diagram it can be seen that the \( z^{-1} \) block acts like an integrator and using this as a building block it is possible to realise the required \( z \) filter.

![Fig. 22—Realisation of \( z \)-1 block](image)

For the biquadratic section of interest

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

and substituting for \( z^{-1} \) gives

\[
\hat{H}(\hat{z}) = \frac{\hat{z}^{-1}}{1 - \hat{z}^{-1}}
\]

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

\[
= \frac{\hat{a}_0 + \hat{a}_1 \hat{z}^{-1} + \hat{a}_2 \hat{z}^{-2}}{1 + \hat{b}_1 \hat{z}^{-1} + \hat{b}_2 \hat{z}^{-2}}
\]

and so the coefficients for the \( \hat{z} \) realisation are related to the direct form by

\[
\begin{align*}
\hat{a}_0 &= a_0 \\
\hat{a}_1 &= 2a_0 + a_1 \\
\hat{a}_2 &= a_0 + a_1 + a_2 \\
\hat{b}_1 &= 2 + b_1 \\
\hat{b}_2 &= 1 + b_1 + b_2
\end{align*}
\]

It can be easily shown that the coefficients \( \hat{a}_1, \hat{a}_2, \hat{b}_1, \hat{b}_2 \) are small quantities for poles and zeros near \( (1, 0) \) and thus computational accuracy can be maintained by using the techniques described for the IAB structure. The realisation of the 2AB structure is then identical to Figs. 17a and b with \( z^{-1} \) replaced by \( \hat{z}^{-1} \) and the coefficients and multipliers changed for the “hatted” coefficients above with the multiply/shift arrangement of Fig. 21. The filter structures that result are shown in Figs 23a and b where the lower complexity of the zeros-before-poles structure obtained by transposition can be seen clearly. Using the direct form, shift controls \( S_0, S_1, S_2 \) are required to control internal gains with the result that shift indices at the multiplier outputs are not directly calculated from the scaled coefficients as in the transpose configuration.

A detailed analysis of this family of structures\(^{24}\) has shown that they offer high accuracy, good round-off noise, and small limit cycles.

4.4 Error feedback

In the next section the effect of finite wordlength within a filter structure is considered in more detail. Although there is some latitude for placing roundoff nodes, truncation must occur at least one node within the structure. It has been shown\(^{25}\) that certain structures having only one roundoff node can be virtually free of limit cycles. In other structures,\(^{26}\) it is possible to save and feed back the roundoff errors so that the roundoff noise at the filter output is reduced in some sense. The recursive section of a second order section incorporating error feedback is shown in Fig. 24. The block marked “Q" is a quantiser with two outputs. One output is a rounded or truncated version of the input, the other passes on the error or roundoff bits for further processing. The technique is particularly effective in filter structures since it is possible to choose coefficients, \( K_{b1}, K_{b2} \) such that a zero is positioned in the \( z \) plane to cancel effectively the noise produced by each pole of the filter. In common with the previous structures in this section, the improvement is most marked as the filter poles approach \( z = (1, 0) \). The penalty for this improvement in performance is, of course, the additional processing associated with the roundoff bits.

4.5 Floating-point arithmetic

For each structure so far considered it has been necessary to truncate or round to 16 bits at least one node of a structure at the input to a multiplier. This imposes an immediate limit to the performance attainable since even with the use of error feedback, dynamic range between filter sections is limited by the necessary roundoff at the input to the following section. It is therefore essential to increase the dynamic range at these points to avoid signal degradation. Providing extra bits would involve a larger, more expensive parallel multiplier or a different multiplication technique. For this reason a floating-point technique was investigated to increase processing accuracy within and between filter sections. Multiplications can be carried out in floating-point arithmetic with extra bits allocated to the exponent. Floating-point addition is more complex, and for this reason it may be expedient to retain this operation in fixed-point arithmetic. Hardware for these operations is relatively simple to incorporate in a signal processor and its application is discussed further in Section 5.
Fig. 23—Equivalent 2AB structures
form $F(j\omega)$ is written $\| F(j\omega) \|_p$ and defined by

$$\| F \|_p = \left[ \frac{1}{2\pi} \int_{-\pi}^{\pi} |F(j\omega)|^p d\omega \right]^{1/p}$$

It can be seen that when the expression is evaluated for $p = 1$, the mean absolute value of $|F(j\omega)|$ is obtained. For $p = 2$ the energy of $|F(j\omega)|$ is the result and for $p = \infty$, the maximum value of $|F(j\omega)|$ is obtained.

Returning to the evaluation of the magnitude of the signals at a filter node, then from a knowledge of the nature of the input signals a suitable norm, $L_q$, can be selected; likewise from a knowledge of the transfer function from the input to the node, an $L_p$ norm can be chosen. Using spectral quantities, it has been shown that

$$\| V \|_1 \leq \| F \|_p \cdot \| X \|_q \text{ for } \frac{1}{p} + \frac{1}{q} = 1$$

where $\| V \|_1$ is the mean absolute value of $V(j\omega)$, the spectrum of the signal at that filter node. This in turn provides a bound on the data sequence $|x(n)|$.

5. The effects of finite wordlength in digital filters

At each stage of the digital audio signal chain, decisions must be made about the appropriate wordlength to be used. The first decision is in the analogue to digital converter itself, for this sets the performance by which all subsequent processing must be matched. In this digital signal processing and in particular, filtering, the three major impairments will be from

(1) inaccuracies in filter response caused by quantisation of the coefficients
(2) roundoff noise
(3) limit cycles.

As has been pointed out, the choice of structure has great effect on these impairments as will be shown using some examples in this section.

5.1 Coefficient quantisation

If, for the moment, the effects of arithmetic roundoff in a particular filter structure are ignored, then the accuracy with which a required filter is achieved is determined by the accuracy of the filter coefficients. It has already been pointed out that the structure is an important factor in determining the
sensitivity of the filter characteristics to coefficient quantisation. An insight into the effect of structure choice on coefficient sensitivity can be gained by observing the density of allowable root positions in the $z$ plane. This can then be compared with the root positions for the filters to be implemented. For example, in Fig. 25, the root positions for a family of filters with a peaking gain of 18 dB at centre frequencies of 7 kHz down to 75 Hz is shown. This reveals a requirement for accurate root location near the unit circle and $z = (1, 0)$ point. For the direct form the quantisation of coefficients results in a highly non-uniform distribution of allowable root positions (Fig. 26). This result is most noticeable near the unit circle and at the point $z = (1, 0)$, important regions from the point of view of filter stability.

By design, the coupled form leads to a uniform distribution – the roots lie on a rectangular grid in the $z$ plane (Fig. 27) and so the maximum error in root position is the same for all regions in the $z$ plane. As previously mentioned, the price for this convenience is a non-canonic realisation. There are many other non-canonic realisations which permit a variety of root distributions in the $z$ plane.

The AB structures can fulfil this requirement by taking advantage of the feature that near the unit circle, $z = (1, 0)$ region the coefficients become small. Thus, the coefficients may be expressed in a binary floating point format with a small number of bits to the multiplier, accompanied by binary shifting. For a fixed number of bits assigned to the multiplier/shift control combinations, the effect is

---

Fig. 25—Root locations for a family of presence filters

Fig. 26—Grid of root locations (5 bits) – direct form

Fig. 27—Grid of root locations (5 bits) – coupled form
to produce a greater density of roots towards the unit circle and $z = (1, 0)$ region (Fig. 28).

5.2 Roundoff noise

Addition and multiplication are the two arithmetic processes used in digital filter structures. The addition of two or more numbers in fixed point arithmetic cannot lead to inaccuracy unless overflow occurs. If two’s complement arithmetic is used, partial sums may be permitted to overflow and as long as the final sum remains within the number range, then no error will occur. By scaling these outputs as outlined in Section 4.5, overflow is avoided and the adders will not contribute roundoff errors in the filter output.

In the case of multiplication, overflow cannot occur. But inaccuracy is introduced by the necessary roundoff of products to maintain manageable wordlengths. The effect of these roundoff operations on the noise performance of a digital filter can be analysed by modelling each roundoff node by a perfect multiplier and a noise source. For the purposes of this analysis, it is necessary to assume that these noise sources are uncorrelated with each other or the input, and thus can be regarded as spectrally flat.

5.2.1 Transposed DF2 structure

The method is applied to the transposed DF2 structure in Fig. 30, which shows the location of noise sources at the output of each multiplier and an additional noise source at the filter output because the multipliers ($b_1, b_2$) can be presented with only a limited number of signal bits. This reduced number of bits is also handed on to the next filter stage with its own set of ($a_0, a_1, a_2$) multipliers. If $E_{03}(z), E_{14}(z), E_{25}(z)$ are the $z$ transforms of the combined noise sources [$e_0(n) + e_1(n)$], [$e_1(n) + e_2(n)$] and [$e_2(n) + e_3(n)$] respectively, then for a zero signal input, the $z$ transform of the output is given by

$$Y(z) = \frac{E_{03}(z) \cdot z^2 + E_{14}(z) \cdot z + E_{25}(z)}{z^2 + b_1z + b_2}$$

Thus the spectral properties of the quantizing noise contained within the output depends only on the denominator (the poles) of the transfer function. From the formula, it can be seen that since $z$ is spectrally flat, all the noise sources undergo the same spectral shaping. Considering the noise performance at low frequencies, i.e. $z \rightarrow 1$, each noise source will contribute equally to the output. However, in a practical implementation of this structure an increased wordlength can be used at the output of multipliers, while at the filter output the wordlength must be reduced to suit the multiplier inputs.
Thus the noise source $E_{o,i}(z)$ dominates the filter performance. As has been shown, the effect of this can be reduced by error feedback, conversion to floating point arithmetic, and rounding rather than truncation.

The spectral shaping of the noise is given by the transfer function

$$G(z) = \frac{1}{z^2 + b_1 z + b_2} = \frac{1}{(z - p_1)(z - p_2)}$$

where $p_1$, $p_2$ are the pole locations in the $z$ plane as shown in Fig. 31. The maximum of this function will indicate the worst case of the effect of roundoff noise in the filter output and in this example, the poles are assumed close to the unit circle, thus

$$p_1, p_2 = r \exp (\pm j\theta) \quad \text{where} \quad r = 1 - \varepsilon, \text{small} \ \varepsilon$$

We also assume that the poles have a low natural frequency, so $\theta$ is small.

The maximum noise amplification is then the maximum value of $G(z)$,

$$|G(z)|_{\text{max}} = \frac{1}{|z - p_1| \cdot |z - p_2|} = \frac{1}{d_1 d_2}$$

where $d_1, d_2$ are the distances from the poles to the unit circle at the frequency at which noise amplification is greatest. From simple geometrical observations it can be seen that this approximates

$$|G(z)|_{\text{max}} \approx \frac{1}{2\theta}$$
For example, a presence filter peaking +18 dB at 75 Hz has a denominator \( D(z) \)
\[
D(z) = z^2 - 1.986542z + 0.98676
\]
giving \( \varepsilon = 1 - \sqrt{b_2} = 0.0067 \)
\( \theta = (1 - \frac{b_2^2}{4b_1})^{1/2} = 0.0132 \)

Thus the maximum noise amplification is 5679, corresponding to 75 dB, or expressed differently – with a 16 bit output signal, the quantising accuracy in the region of 75 Hz is approximately 3½ bits.

5.2.2 2AB structure

Consider now a similar noise analysis for the 2AB structure. The structural similarities of the 2AB and transpose DF2 structures (compare Figs 17b and 23b) result in identical locations of the noise sources and the output noise is obtained by substituting \( \hat{z} = (z - 1) \) in the zero input transfer function
\[
Y(z) = \frac{E_{03}(z) - (z - 1)^2 + E_{14}(z) - (z - 1) + E_{25}(z)}{\hat{z}^2 + \hat{b}_1 \hat{z} + \hat{b}_2}
\]

For the same conditions, i.e. \( z \to 1 \), it can be seen that the noise term \( E_{03}(z) \) is heavily attenuated. From Fig. 31,
\[
(z - 1) \approx \theta, \text{ small } \theta
\]
and for the values of the previous example
\[
(z - 1)^2 \approx \theta^2 = 0.00017 = -75 \text{ dB}
\]

The \( E_{14}(z) \) term is also attenuated with the result that the \( E_{25}(z) \) term now dominates. In the practical implementation, extra bits are allocated at this node, a much easier thing to do than in the previous case where extra bits are needed at multiplier inputs.

Clearly, as the poles move away from \( z = (1, 0) \) the benefits of this structure are reduced. A point is reached when the magnitude of the coefficients \( a_1, a_2, b_1, b_2 \) exceed unity and thus cannot be expressed in fractional arithmetic. This occurs for
\[
|1 - z_s| > 1
\]
where \( z_s \) is the singularity position in the \( z \) plane. (Fig. 28).

5.2.3 Computer simulations of roundoff noise in filter structures

In order to validate the simplified noise analysis above it was decided to carry out a computer simulation and noise analysis of the two structures. The operation of the filters was simulated exactly by using fixed point arithmetic on a PDP-11/34 computer – a 16 bit machine. Extensive use was made of integer double precision routines to extend the accuracy to 32 bits where required.

A two bit exponent was used in the floating-point representation at the filter output giving the equivalent of 19 bit dynamic range in fixed-point arithmetic. The resulting structures are shown in Fig. 33. The filter output is converted to floating-point in the block labelled "FP", and the exponent, \( L_N \) is used to scale the outputs of the \( b_1, b_2 \) multipliers and also the \( a_0, a_1, a_2 \) multipliers of the next section. The scheme is illustrated in Fig. 32. The roundoff errors were measured by subtracting the output of the filter under test from that of a reference filter. The reference filter used 32 bit accuracy processing, for these purposes this can be considered "perfect". Both reference and test filters only required 16 bit accuracy for the filter coefficients.

The program allows up to 9 biquadric sections to be cascaded and the structure is defined by simple Fortran statements. The excitation was a synthesised sine-wave of arbitrary level and frequency, though subsequent developments permit the input signal to be "real" audio. Other inputs to the program are the run-length (number of samples processed), a run-up time (to allow adequate settling time for the filter), and the processing accuracy (the wordlength for all arithmetic other than multiplier inputs).

The spectral properties of the roundoff noise were calculated from its autocorrelation function measured as the noise was generated. The spectrum

![Fig. 32—Simulation method for noise analysis](image)

* The work on computer simulation of filter structures was carried out by T.A. Moore.
was then evaluated using the maximum entropy technique which gives good results with a small number of autocorrelations.\textsuperscript{32}

This technique was applied to a large number of filter characteristics and two representative examples are given in Fig. 33. The performance of the transposed DF2 and 2AB structures is compared in Fig. 34 for two filters; 18 dB presence filters at 100 Hz and 2 kHz.

\textbf{Fig. 33—Structures used in the computer simulation}
(a) TDF2 structure, (b) 2AB structure

\textbf{Fig. 34—Noise spectra for TDF2 and 2AB structures}
The rms noise is expressed in dB's relative to peak-to-peak level, (clipping level in the ADC) and the results are shown for 16, 20, 24 and 28 bit accumulator accuracies. In each case a 2 bit exponent corresponding to 0, 1, 2 or 3 shifts after the multipliers, is used in the floating-point representation of the filter output.

From the results, the following conclusions were drawn:

(1) 24 bit accumulator accuracy combined with a 2 bit exponent in the floating-point technique provide adequate performance.

(2) For filter characteristics operating in the region of 2 kHz, (corresponding to \( \frac{1}{f_n} \) of sampling frequency), the performance of the two structures is comparable. For filters active below this, the 2AB structure is preferred; and above it, the TDF2 structure.

5.3 Limit cycles

In the previous Sections the effects of finite precision arithmetic were discussed and the errors so produced were analysed. In certain circumstances, e.g. when signal levels become very low during a pause in speech or music, the quantisation errors can be highly correlated and because of the inherent feedback in recursive filters, the filter can be placed in a mode of self-sustained oscillation known as a limit cycle. In the preceding analysis of roundoff noise, the assumptions made would probably result in an optimistic expectation of the real filter's performance. When limit cycles are considered, a pessimistic expectation may result – it is therefore of interest to consider both.

Consider a simple recursive filter transfer function

\[
H(z) = \frac{1}{1 + b_1 z^{-1} + b_2 z^{-2}}
\]

The presence of zeros in the numerator does not affect the generation of limit cycles, only the form that they take at the filter output and so their action will be ignored here. For the purposes of the example, let \( b_1 = -0.9, b_2 = 0.8 \). The output of the filter is given by the difference equation:

\[
y(n) = 0.9 \, y(n - 1) - 0.8 \, y(n - 2) + x(n)
\]

The first kind of limit cycle can be found by examining the response to a zero-input, \( x(n) = 0 \) assuming pre-defined initial conditions for \( y(n - 1) \) and \( y(n - 2) \). Starting with initial conditions \( y(n - 1), y(n - 2) = 4 \) and with each product rounded to the nearest integer, the output sequence is

\[
y(n) = 4, 4, 1, -2, -3, -1, 1, 2, 1, -1, -2, -1, 1, 2, \ldots
\]

If truncation is used instead of rounding

\[
y(n) = 4, 4, 0, -3, -2, 1, 1, 0, 0, 0, 0, \ldots
\]

These two cases can be conveniently represented by plotting \( y(n - 1) \) vs. \( y(n) \) for successive values of \( n \). The result is shown in Fig. 35 in a diagram known as the successive value plane. The limit cycle produced as a result of rounding can be

Fig. 35—Example of rounding and truncation in the successive value plane
clearly seen while, surprisingly, truncation did not produce a limit cycle. This simple example does not tell the whole story however. If the initial conditions are changed, different limit cycles can occur. In fact it has been shown \(^\text{13}\) that in the case of rounding, limit cycles will always occur if \(|b_2| > 0.5\) and that for truncation, limit cycles will exist for \(|b_1| \geq 1\). In the case of truncation, the limit cycles take one of two forms. The first is a discrepancy error, i.e., a steady state error between the finite precision filter and its ideal equivalent which occurs when \(b_2 < 0\). If \(b_1 > 0\) a limit cycle at half sampling frequency occurs. Both these phenomena are termed deadband limit cycles.

A second kind of limit cycle may result if overflow occurs during addition, often causing large amplitude oscillations. In the TDF2 structure of Fig. 17b, the difference equation can be written

\[
y(n) = b_1 y(n-1) + b_2 y(n-2) + x(n)
\]

To find the conditions that prevent sustained oscillations \(x(n)\) can be set to zero, and \(y(n)\) required to be \(< 1\), (assuming fractional arithmetic is used). This holds if

\[
|b_1| + |b_2| < 1.
\]

This also applies to the 2AB structure with \(b_1, b_2\) replaced by \(b_1, b_2\). This is a severe restriction on coefficient values and thus indicates the desirability of prevention of overflow.

It can be seen from the above difference equation that for a constant output \(y(n) = y(n-1) = y(n-2)\), and with \(x(n) = 0\),

\[
(b_1 + b_2)y(n) = y(n) - 2
\]

due to the overflow characteristics of a two's complement adder. Solving this expression for \(y(n)\) indicates a deadband limit cycle of amplitude \(Y_{ss}\),

\[
Y_{ss} = 2/(1 - b_1 - b_2)
\]

and by similar reasoning a limit cycle at half sampling frequency can occur when \(y(n) = -y(n-1) = y(n-2)\) with an amplitude,

\[
Y_{ss} = 2/(1 + b_1 - b_2)
\]

These limit cycles can be avoided however by using an adder with a saturating characteristic.

It is impractical in a Report of this nature to undertake a detailed analysis of limit cycles. In fact, there are many aspects of this subject that are not fully understood. Nevertheless, it is of great importance to the digital filter designer to have some idea of the magnitude of limit cycles if the noise so produced is to be kept within defined limits. For the particular structures under study, the susceptibility to limit cycles was therefore investigated experimentally.

Three methods were used to induce limit cycles.

1. Zero input limit cycles — with the filter set up in a variety of different initial conditions and zero input, \(x(n) = 0\), the filter output was monitored. A logic analyser was used to monitor the 24 bit filter output before truncation. For the zeros before poles direct form, there were no limit cycles but discrepancy errors of about 512\(q\) were found. \((q = \text{the quantisation interval} = 2^{-23})\). Fortunately, discrepancy errors are not of great importance for audio filters since for real signals, the internal arithmetic never settles for long enough for these errors to be apparent, whereas limit cycles can occur because of a disconnected input. Similarly, the 2AB structure exhibited no limit cycles, but a maximum discrepancy error of 16\(q\).

2. Step input limit cycles — the initial stored values in the filter were set to zero with zero input and then a step input applied. The step amplitude was varied to discover worst case conditions at the filter output. For the TDF2 structure, no limit cycles occurred with discrepancy errors similar to the zero input test, but for the 2AB structure limit cycles did occur with an amplitude of between 19\(q\) and 27\(q\) equivalent to about 4 to 5 least significant bits.

3. Overflow oscillations — a data sequence was supplied to a filter with coefficients such that overflow was possible. The TDF2 structure produced large amplitude oscillations at half sampling frequency, some of which did not cease even when a normal input was applied. The 2AB structure appeared always to recover from an overflow. This indicates the need to prevent overflow and suggests that saturating arithmetic should be seriously contemplated. Nevertheless, for all normal inputs, which are scaled correctly, no problems should occur.

6. A four filter cascade for audio equalisation

The kinds of equalising filters which are used in a mixing desk channel were described in Section 1. With the results of the work described in Sections 3, 4, and 5 it is now possible to select a high performance digital filter structure which will be suit-
able. Firstly, however, it is necessary to review the dynamic range and headroom requirements for such a filter set.

Headroom is defined (usually in dB) as the range above the peak signal level in which short term transients may be handled without distortion. In a digital system, the headroom corresponds to the distance between peak signal level and the clipping level of the ADC and current BBC practice for digital transmission is +1.5 dB. This small amount of headroom can be tolerated because the ADC is preceded by a protective limiter. In sound recording studios and broadcast studios where no limiters are presently used, it is not uncommon to find headroom specifications of up to 30 dB. The consequence for digital signal processing such as equalisation is that sufficient bits must be reserved at all stages of processing to accommodate the headroom specification.

Dynamic range is defined as the ratio (in dB) of the strongest to the weakest signal which can be handled by a system. In digital systems, this corresponds to the ratio of the signal at clipping level to a signal which just excurses two quantum levels. Thus the dynamic range of a signal coded with 16 bit accuracy is 96 dB.

Consider the action of a peaking filter with a gain of +18 dB. Immediately, it will be necessary to attenuate the input signal by 18 dB if headroom is to be conserved. To avoid the discarding of useful information bits, the floating point format described previously is adopted at this stage. This will act precisely as an instantaneous compandor in which a 2 bit exponent is generated in the compression from 19 bits to 16 bits and used again in expanding from 16 bits to 19 bits (19–16–19). If necessary this can be extended up to (23–16–23) with a 3 bit exponent. In each case the increased dynamic range will of course be accompanied with very low level pro-

gramme-modulated noise. In Fig. 36, headroom is protected and data converted to floating-point format in the block labelled ATT.

Filtering is then carried out using the techniques already described. Data passed between filter sections is always in floating-point (FP) format. Some of these filters will introduce gain at certain frequencies but, in practice, equaliser characteristics rarely overlap to a significant extent and a total maximum gain at any single frequency is unlikely to exceed 18 dB. Fig. 37 is an input-to-output level diagram which illustrates the action of a single filter stage. At the filter output, the noise level depends on the filter characteristic, e.g. a worst case figure would be $\approx -95$ dB for a +18 dB presence filter at 100 Hz (see Fig. 34 for the TDF2 structure with 24 bit accumulators) but typically, this noise level will be better than $-107$ dB – i.e. true 16 bit performance.

The final stage of processing in Figs. 36 and 37 is that of gain make-up (GMKUP). This combines the functions of restoring the signal to its expected level, i.e. compensating for ATT, with a gain adjustment derived from a manual control such as a fader. It is useful to be able to provide a small amount (e.g. 18 dB) of gain as well as the full range of attenuation at this point. Additionally, the output is of course required in fixed-point arithmetic. This is achieved most efficiently by providing extra bits at the output ports so that the genuinely increased dynamic range is passed on to subsequent processing, e.g. mixing. Instead of shifting data into a more significant position (inefficient in COPAS hardware) the weighting of the output bits is notionally changed by $2^{-4}$. Thus, if the filter output is not shifted, it is equivalent to +24 dB gain. This is a realistic upper limit and combines the requirements of compensating for headroom protection and manual gain make-up.

---

**Fig. 36**—Block diagram representation of a COPAS program for 4 filters in cascade

23
Alternatively, if it is required to produce a 16 bit output, compatible with 16 bit inputs of other COPAS processors, digital gain is applied to the filter output in the conventional sense, but with a saturating overflow characteristic (Fig. 38). With this arrangement, signal overload can occur at only the final stage of this processing chain and it can be corrected by manual control of the fader. The output is then forced into the same dynamic range as the original input.

An important feature of Fig. 36 is that the four filters in operation can be reconfigured to provide, for example, four mid-band presence filters or four notch filters at the same frequency. As long as the total processing power of the hardware is not exceeded, an enormous range of filter combinations can be implemented. Specific requirements such as a high Q notch filter at 50 Hz can be implemented with the 2AB structure executed at high accuracy using double precision arithmetic (48 bits), though naturally this involves extra processing time.

To provide equalisation over a wide range of frequencies, it is necessary to use both the 2AB and the TDF2 structures. It has been shown that because of limitations on coefficient values and performance criteria, the 2AB structure should be used for filters active below approximately 2 kHz.

(a) Saturation. (b) Zeroing. (c) Two's complement.

Fig. 38—Adder overflow characteristics
and the TDF'2 structure above this. This conveniently meets operating requirements, where it is common to split the mid-range equalisers at about 2 kHz. In the four filter cascade, two 2AB structures provide equalisation for low and midband ranges and two direct form structures are used for upper midband and high frequency ranges.

The computer program implementation of this block diagram includes automatic re-initialisation of registers when a filter is not in use. The bracketed numbers indicate the number of COPAS instructions required for each function. When a filter is not in use, only 3 instructions are required for the initialisation and bypass to be effected. This is a useful feature since the processing time so released can be used effectively for other signal processing.

The control program provides filter coefficients in an orderly manner and calculates the degree of input attenuation and gain make-up required. These are slowly changing parameters and are easily dealt with by the on-board microprocessor in each COPAS module. The details of this control program will be elaborated in Section 8. Care must be taken during these slow-speed operations, however, since the sudden change of a filter parameter or gain setting can result in a momentarily unstable filter or an audible click as the gain changes.

7. Hardware Implementation of the Filters

The detailed discussion of programmable hardware for implementing filters is given in a companion Report16. Nevertheless, a brief overview is presented here for convenience. The essential features of the processor are shown in Fig. 39 and it will be shown that with this bus configuration, the two filter structures of interest can be implemented with high efficiency while maintaining a flexible arrangement which can be programmed to handle other signal processing requirements.

To maximise throughput, the two inputs and output of the multiplier are connected to separate buses and a hardware shifter provides 0–7 shifts to fulfil the function outlined in Fig. 21. The requirement for shift operations before accumulation rules out the use of a multiplier accumulator. The Arithmetic Logic Unit (ALU) consists of six, type 2901B bit-slice microprocessors providing 24 bit data paths. The unit contains 16 registers which may be accessed two at a time. Thus, shifted products can be added to one register and the result stored in another within a single cycle. The $2^{-1}$ function (Fig. 23) is accomplished efficiently by adding two stored values and saving the result in a single machine cycle. Filter coefficients are stored in Random-Access Memory (RAM) and are loaded from the slow-speed microprocessor (MPU). The 32 bit product from the multiplier is truncated to 24 bits for connection to the bit-slices. The floating-point conversion between filter sections is conveniently handled using the shift functions of the ALU and the hardware shifter.

The 2AB and transpose direct forms are executed in 17 and 16 instructions respectively. This compares favourably with the minimum of 5 operations associated with necessary multiply functions in the filter. With an instruction cycle of 140 ns achieved with COPAS-2D, this represents a practical signal processor for digital equalisation.

![Diagram](image)

Fig. 39—Schematic hardware of COPAS-2D

25
8. The control program

It is a feature of digital systems in general that it is easy to separate the control functions from signal processing hardware. In COPAS, the IEEE-488 bus (GPIB) is used for transfer of all control information which is generated by a desktop computer. Each COPAS module has an interface with the bus and a resident microprocessor is programmed to respond to, and in some cases, generate data on the bus.

For experimental purposes, the control program in BASIC* requires a numeric filter specification to be entered from a keyboard. The program then determines the filter coefficients and structure from a library of stored data; the necessary headroom protection and corresponding gain make-up is calculated, and a data "package" is transmitted along the GPIB. This is done using a series of commands for which a convention has been defined (Table 1). When the data transfer is complete the computer display is updated to show the filter combination which is currently being executed. A typical display is shown in Fig. 40.

The use of defined commands on the GPIB is highly desirable for the following reason. It isolates the signal processing hardware from the function required, and if in the future the hardware were changed, alteration of the control program would not be required. It would be a great advantage, at this early stage in digital audio signal processing, if these commands could be standardised.

All commands obey the following rules:

1. They may be preceded by a comment, indicated by text contained in angle brackets (⟨⟩).
2. Each command starts with a period (.).
3. Commands consist of either one or two alpha-numeric characters.
4. The command is followed by a dash (−) if data follows.

Table 1—COPAS-2D Command Table

| Type 1 Commands – Contain multiple data words for loading in coeff. mem. |
|-----------------|------------------------------------------------------------------------------------------------|
| .F1− HHHH HHHH HHHH HHHH $ | filter 1 coefficients |
| .F2− HHHH HHHH HHHH HHHH $ | filter 2 coefficients |
| .F3− HHHH HHHH HHHH HHHH $ | filter 3 coefficients |
| .F4− HHHH HHHH HHHH HHHH $ | filter 4 coefficients |
| .CL− HHHH$ | compress/limit transition |
| .C− HHHH$ | compressor coefficients |
| .L− HHHH$ | limiter coefficients |
| .E− HHHH$ | expander coefficients |
| .TC− HHHH HHHH HHHH$ | time constants |
| .G− HHHH$ | main fader |
| .D− HHHH$ | delay (0–255 samples) |
| .P2− HHHH HHHH$ | stereo pan |
| .P4− HHHH HHHH HHHH HHHH$ | quad pan |

<table>
<thead>
<tr>
<th>Type 2 Commands – Contain no data</th>
</tr>
</thead>
<tbody>
<tr>
<td>.M1$</td>
</tr>
<tr>
<td>.M2$</td>
</tr>
<tr>
<td>.T$</td>
</tr>
<tr>
<td>.XG$</td>
</tr>
<tr>
<td>.XS$</td>
</tr>
<tr>
<td>.RO$</td>
</tr>
<tr>
<td>.RI$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Type 3 Commands – Special purpose with data</th>
</tr>
</thead>
<tbody>
<tr>
<td>.SS− H H H$</td>
</tr>
<tr>
<td>.SC− H H H$</td>
</tr>
<tr>
<td>.SI− H HHHH HHHH$</td>
</tr>
<tr>
<td>.Z−AA− HHHH HHHH$</td>
</tr>
</tbody>
</table>

* The BASIC program for controlling COPAS filters was written by D.J. Marshall.
FILTER PARAMETERS

| FILTER No1 | 185Hz  | 3dB    |
| FILTER No2 | 800Hz  | -4dB   |
| FILTER No3 | ------ | ------ |
| FILTER No4 | 3700Hz | 10dB   |

Fig. 40—Typical display for the digital EQ control program

5. Commands are terminated by the ASCII escape code, indicated by ($).

Consider the following example in which a midband presence filter is required, then the commands transmitted could be

```
.XG$
(800Hz : +10dB : Q = 4.75).F4-4914 19E0
017A ED9C FD9A 1$
(Patch in filter No. 4).SS-4$
```

The first command starts the COPAS program. The second supplies coefficients for the filter with a descriptive comment, and the third sets up context switches so that filter No. 4 is patched in. The ability to string together these commands permits complicated control functions to be achieved with comparative ease. For example, the sudden introduction of a filter in the middle of a performance might introduce an audible click. Using a number of commands, the filter can be gradually introduced with small changes in gain or frequency, i.e. the filter coefficients. The method can also be used to generate time varying filters. A more detailed account of the GPIB commands is given in the companion Report.

It is unlikely for the moment that a computer could calculate filter coefficients quickly enough to meet operational requirements and therefore it is of interest to know the size of the data "library". A very comprehensive range of equalisers is illustrated in Table 2.

Each characteristic requires storage of five coefficients and so a library of $31,040 \times 16$ bits is needed. This can be halved if only the coefficients for positive gains are stored – a simple calculation yielding the coefficients for negative gain values. The resulting memory could be implemented as $16K \times 16$, a moderate size which could be constructed with eight Erasable Programmable Read Only Memories (EPROMs). If the system sampling rate is liable to change, a read-write memory could be used and the appropriate coefficients loaded from tape.

9. Conclusions

This Report has detailed the complete process of design, realisation and implementation of digital filters having a performance suitable for high quality audio equalisers. The structures used are efficiently implemented on COPAS-2D and their performance is compatible with 16 bit quantising accuracy. Four of these filters are easily cascaded and the computations carried out sufficiently quickly that it is a practical matter to use a processor such as COPAS-2D to carry out the processing associated with a single channel of a mixing desk.

<table>
<thead>
<tr>
<th>No.</th>
<th>Type</th>
<th>specified by</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>frequencies</td>
<td>gains</td>
</tr>
<tr>
<td>2</td>
<td>Presence</td>
<td>32</td>
</tr>
<tr>
<td>2</td>
<td>Shelving</td>
<td>32</td>
</tr>
<tr>
<td>1</td>
<td>High Pass</td>
<td>32</td>
</tr>
<tr>
<td>1</td>
<td>Low Pass</td>
<td>32</td>
</tr>
</tbody>
</table>

Table 2—Characteristics required of one equaliser.

27
10. Acknowledgement

This work was carried out in collaboration with the Neve Group of Companies.

11. Appendix – software summary

In the course of the work described in this Report, many computer programs were written. It is not practical to include full details of these programs here and so a list is given with a brief description.

- PRES – calculates coefficients for presence filters from a simple specification.
- SHELF – calculates coefficients for shelving filters.
- BUTTER – calculates coefficients for Butterworth low pass and high pass filters.
- GRID – plots allowable root locations in z plane for various structures.
- TRANS – generates coefficients for 2AB structure from direct form coefficients.
- TRUNC – generates coefficients truncated to specified number of bits.
- IIRPLT – plots frequency, phase and group delay characteristics of Infinite Impulse Response (IIR) filters.
- FIRPLT – plots frequency characteristics of Finite Impulse Response (FIR) filters.
- OVRFLOW – searches for overflow conditions in recursive filters.
- BF19 – noise analysis of filters as described in Section 5.2.3.

12. References


9. TRW Multiplier Accumulators (type 1010j). TRW LSI Products sales literature, P.O. Box 1125, Redondo Bech, Ca. 90278.


