Extended Remez algorithms for filter amplitude and group delay approximation

R.K. Henderson, BSc, PhD
Li Ping, MS, PhD
Prof. J.I. Sewell, BSc, PhD, CEng, FIEE

Indexing terms: Mathematical techniques, Filter amplitude, Group delay functions

Abstract: Various Remez-type algorithms for the computation of rational filter functions of a general form are proposed. They allow filter amplitude and group delay functions to be arbitrarily shaped and tapered. Such functions can be applied to predistort for undesirable effects in communication systems such as sinc (x), transmission line weighting or distortion due to nonideal components. High-order touch points are introduced as a generalisation of the concept of maximum flatness. They are used to trade off between the amplitude, group delay and passive sensitivity properties of a filter. The high-order touch points can be used directly to design a passive ladder prototype by an iterative algorithm.

1 Introduction

The paper is concerned with computational methods for the design of rational filter functions. Classical functions which approximate ideal filter amplitude specifications are commonly known. These functions have special properties of symmetry and constant equiripple attenuation in passband or stopband. Their coefficients can be conveniently calculated by explicit formulas or simple iterations [1, 2]. However, classical approximations are not suitable for highly asymmetric specifications or amplitude equalisation tasks which are often encountered in modern communications systems [3]. Approximation methods for such specifications are not well developed. They are usually highly specific to a given filtering task and are not flexible enough for a general, easy-to-use software package [3–7]. Although general multiple-criterion optimisation techniques can certainly be applied, they tend to involve a large amount of computation and do not always guarantee convergence [8–10]. This is frequently because they do not make enough use of the special properties of filter functions, which are particular cases of rational functions with well-confined root locations. Moreover, the approximation problem is often not expressed conveniently for a filter designer without detailed knowledge of optimisation theory.

Paper 7762G (E10), first received 1st March and in revised form 17th August 1990

R.K. Henderson is with CSEM, Case Postal 41, CH-2007, Neuchatel, Switzerland

Li Ping is with the Department of Electrical and Electronic Engineering, University of Melbourne, Parkville, Victoria 3052, Australia

Prof. J.I. Sewell is with the Department of Electronics and Electrical Engineering, University of Glasgow, Glasgow G12 8QQ, Scotland, United Kingdom

Methods for the approximation of polynomial filter functions within arbitrarily weighted amplitude specifications are considered. Several new algorithms are proposed, which bear interesting relationships to the Remez minimax approximation technique [11, 12]. The concept of maximum flatness is generalised to allow compromises between equiripple and flatband properties. The computational aspects of the algorithms are discussed. As accuracy is particularly critical in the approximation of filter functions in finite word length computer arithmetic, methods to preserve accuracy without recourse to the complications of transformed variables are given [1, 2].

The design of minimum phase rational functions with arbitrary passband and stopband tolerance schemes is investigated. These functions are of particular importance because they can be efficiently realised as the transfer functions of linear analogue networks, including switched-capacitor filters (SCFs). Numerator and denominator polynomials have special properties which are best designed by a combination of two different methods.

The approximation discussion so far has considered only filter amplitude functions and has ignored their associated group delay. Modern digital communications and signal-processing systems often require filters which satisfy simultaneous specifications on amplitude *and* group delay. A common practical design approach is to separate the two approximation problems by employing an all-pass function to equalise the group delay of a minimum phase amplitude function. The latter function should first be optimised to reduce the peaking of the delay towards the passband edges, either by smoothing the passband amplitude function (e.g. Butterworth) or reducing the rolloff into the stopband. The general amplitude approximation methods offer various ways to trade off between the amplitude and group delay characteristics. High-order touch points can be introduced into the passband and notches can be placed to tailor the stopband rolloff. Although the demands on the group delay correction can be reduced in these ways, it is still costly to use allpass functions. They are known to offer a noncanonic solution to the combined amplitude and group delay approximation problem. Greater efficiency can be achieved by employing a general nonminimum phase function [13, 14]. However, these functions cannot be simulated by low-sensitivity SCFs at present. The argument for all-pass equalisation is strengthened by the recent development of low-sensitivity all-pass SC ladder structures [15].

A method is presented whereby the techniques developed to approximate the amplitude of a transfer function can also be applied to the group delay of an allpass function. Unfortunately, when this group delay function is interpolated, a system of ill-conditioned nonlinear equations arises which becomes very difficult to solve with increasing order [16]. Since Newton- and Remez-type approximation methods depend on an efficient interpolation step, they are difficult to apply with efficiency or reliability [17]. Alternative methods based on optimisation techniques have therefore been studied [18, 19].

A new algorithm is proposed which permits direct application of Remez-type approximation methods to the problem [20]. By observing the similarity between the group delay function and a filter amplitude function, the techniques and theorems for amplitude approximation are still valid. A stable, accurate algorithm is then developed for arbitrary group delay correction.

2 Filter amplitude approximation

A transmission function is designed by working with a magnitude squared function. It has the following property:

$$T(s)T(-s)\Big|_{s=j\omega} = T(x)\Big|_{x=\omega^2} \tag{1}$$

The phase information has been removed and the function has been linearised in terms of a single real variable, avoiding the use of complex arithmetic in later computations.

2.1 Curve-fitting problem

Consider the problem of fitting a polynomial $p(x) = a_n x^n + \cdots + a_0$ in a minimax sense to some prescribed function $m(x)$ on the interval $[a, b]$ such that the maximum error $\max|p(x) - m(x)|$ is minimised. A variant of this problem is of interest to filter designers (Fig. 1). Two

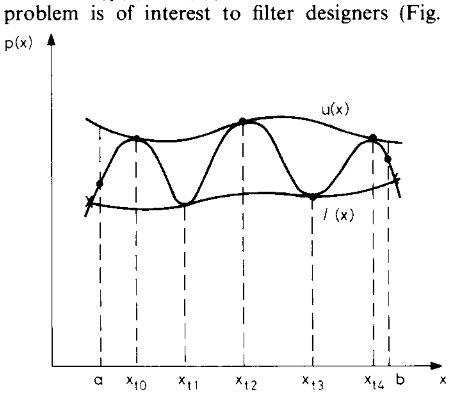

Fig. 1 *Solution of minimum curve-fitting problem by polynomial p(x)*

curves, $u(x) = m(x) + \delta$ and $l(x) = m(x) - \delta$, can be seen as boundary functions and $p(x)$ is sought to fit between them. At a series of points, the so-called *touch points*, $p(x)$ will touch $u(x)$ and $l(x)$ alternately, which implies that $p(x)$ will have the same zero- and first-order derivative values of $u(x)$ or $l(x)$. In a general sense, $u(x)$ and $l(x)$ can be any functions satisfying $u(x) > l(x)$ on $[a, b]$ and the order of tangency at the touch points can be greater than one. At M points (the touch points) on the upper and the lower function, $\{x_{ti}: a < x_{ti} < x_{ti+1} < b\}$

$$p^{(r)}(x_{ti}) = u^{(r)}(x_{ti}) \quad \text{or} \quad l^{(r)}(x_{ti}) \quad r = 0, 1, 2, \ldots, \mu_i \tag{2}$$

where $i = 1, 2, 3, \ldots, M$ and the superscript (r) signifies the rth derivative with respect to x. The exact locations of $\{x_{ti}\}$ are unknown but the sequence $\{\mu_i\}$ is specified (Fig. 2). For convenience we fix the two endpoints by

$$p(a) = \mathrm{A} \quad l(a) \leqslant A \leqslant u(a)$$

$$p(b) = \mathrm{B} \quad l(b) \leqslant B \leqslant u(b) \tag{3}$$

where A and B are usually fixed to the upper and lower boundaries, but could be assigned to some arbitrary values in between. In total there are N_c specifications on the values and the derivatives of $p(x)$ where

$$N_c = 2 + \sum_{i=1}^{M} (\mu_i + 1) \tag{4}$$

The aim of the curve-fitting problem is to find the lowest order approximating polynomial which fits the specifications in eqns. 2 and 3.

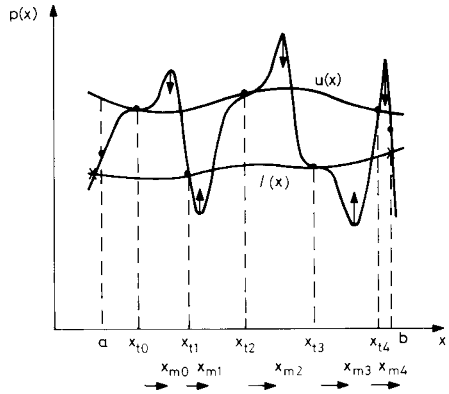

Fig. 2 *Bilateral approximation*

2.2 Interpolation

The unknown positions of $\{x_{ti}\}$ provide M degrees of freedom, which can be used to reduce the order of the polynomial from the nominal problem order N_c. Thus $N_c - M$ of the specifications can be chosen as constraints to form a polynomial of order N, where

$$N + 1 = N_c - M \tag{5}$$

The remaining M specifications must be met by adjusting the M positions of the touch points.

The relation between the behaviour and the order of a polynomial is a complicated issue. To decide the minimum order is a difficult problem and some theoretical discussion can be found in Reference 21; however, in most cases the order determined by eqn. 5 is satisfactory. A Nth order polynomial can always be interpolated by $N + 1$ constraints. Osculatory Newton interpolation [22] can be used to interpolate a number of derivatives with certain computational advantages over other interpolation methods [23].

2.3 Bilateral method

Assume that, for the specifications

(i) all μ_i are odd

(ii) the touch points are assigned alternately to $u(x)$ and $l(x)$, i.e. $\{x_{ti} | i = 1, 3, \ldots, M_u\}$ and $\{x_{ti} | i = 2, 4, \ldots, M_l\}$ are the set of touch points on $u(x)$ and $l(x)$, respectively (where $M_u = M$ and $M_l = M - 1$ if M is odd and $M_u = M - 1$ and $M_l = M$ if M is even).

These assumptions are true for a Chebyshev function where all μ_i are equal to 1 (osculatory points) and are valid for most filter functions. The Weierstrass polynomial approximation theorem guarantees that, if the filter order is high enough, a solution lying between the two boundary functions will exist [21].

Interpolate $p(x)$ such that

$$p^{(r)}(x_{ti}) = u^{(r)}(x_{ti}) \quad r = 0, 1, \ldots, \mu_i - 1 \quad i = 1, 3, 5, \ldots, M_u \tag{6a}$$

$$p^{(r)}(x_{ti}) = l^{(r)}(x_{ti}) \quad r = 0, 1, \ldots, \mu_i - 1 \quad i = 2, 4, 6, \ldots, M_l \tag{6b}$$

and

$$p(a) = \mathrm{A} \quad p(b) = \mathrm{B} \tag{6c}$$

Thus, exactly $N_c - M$ specifications are met by interpolation. It now remains to adjust $\{x_{ti}\}$ to make $\{p^{(\mu i)}(x_{ti})\}$ satisfy the other M specifications.

Definitions

upper error function	$e_u(x) = p(x) - u(x)$
lower error function	$e_l(x) = l(x) - p(x)$
mid-function	$m(x) = (l(x) + u(x))/2$
search function	$s(x) = \max\,[e_u(x), e_l(x)]$
combined error function	$e(x) = \begin{cases} -e_u(x) & \text{if } p(x) > m(x) \\ e_l(x) & \text{if } p(x) \leqslant m(x) \end{cases}$

From assumption (i) above, $\{\mu_i - 1\}$ are restricted to be even, so, in general, the touch points are now points of inflection (Fig. 2) and $s(x)$ will change sign in the neighbourhood of each touch point $\{x_{ti}\}$. If the polynomial is manipulated such that $p(x)$ does not cross $u(x)$ or $l(x)$ at these points, $p(x)$ must possess an extra order of tangency to $u(x)$ or $l(x)$, having then up to the μ_ith order tangency at $\{x_{ti}\}$ required for $p(x)$ to be a solution. At this stage, $\max\,[s(x)] = 0$ in the neighbourhood of $\{x_{ti}\}$. Notice that from assumption (ii) there must be at least one minimum of $e(x)$, denoted as x_{mi}, on $[x_{ti-1}, x_{ti+1}]$. Therefore, if

$$x_{mi} = x_{ti} \quad i = 1, 2, 3, \ldots, M \tag{7}$$

is achieved, $p(x)$ is a solution. An approximation scheme can be adopted to generate an adjustment $\{\Delta x_{ti}\}$

$$\{x_{ti}\} \leftarrow \{x_{ti} + \Delta x_{ti}\} \quad i = 1, 2, 3, \ldots, M \tag{8}$$

to reduce $\{e(x_{mi})\}$.

2.4 Newton's method

Obviously, $\{\Delta x_{ti}\}$ can be generated by a technique based on Newton's method which is found by solving a Jacobian system [22]

$$\begin{bmatrix} g_1(x_{m1}) & g_2(x_{m1}) & \cdots & g_M(x_{m1}) \\ g_1(x_{m2}) & \vdots & \cdots & \vdots \\ \vdots & \vdots & & \vdots \\ g_1(x_{mM}) & g_2(x_{mM}) & \cdots & g_M(x_{mM}) \end{bmatrix} \begin{bmatrix} \Delta x_{t1} \\ \Delta x_{t2} \\ \vdots \\ \Delta x_{tM} \end{bmatrix} = \begin{bmatrix} e(x_{m1}) \\ e(x_{m2}) \\ \vdots \\ e(x_{mM}) \end{bmatrix} \tag{9}$$

where

$$g_i(x) = \frac{\partial p(x)}{\partial x_{ti}} \tag{10}$$

The computational cost of setting up the Jacobian matrix $\boldsymbol{J}$ and solving the Newton system is usually large ($O(n^3)$). Efficient methods to obtain the derivatives $g_i(x_{mj})$ and to solve for the touch point increments $\{\Delta x_{ti}\}$ are now presented.

Theorem 1: Define a Nth order polynomial $q(x)$ subject to the following $N + 1$ interpolation conditions

$$q(x_{mi}) = e(x_{mi}) \quad i = 1, 2, \ldots, M \tag{11a}$$

$$q^{(r)}(x_{ti}) = 0 \quad r = 0, 1, \ldots, \mu_i - 2, i = 1, 2, \ldots, M \tag{11b}$$

and

$$\delta_i(x) = \frac{(x - x_{ti})q(x)}{\mu_i e(x)} \quad i = 1, 2, \ldots, M \tag{11c}$$

then the Newton system (eqn. 9) can be solved for the touch point increments $\{\Delta x_{ti}\}$ by

$$\Delta x_{ti} = \lim_{x \to x_{ti}} \delta_i(x) \tag{12}$$

Proof of theorem 1: Consider the solution of the Newton system for touch point movements Δx_{ti} by interpolation of $q(x)$. Two remarks are necessary for the proof.

Remark 1: Suppose $u(x)$ or $l(x)$ (and so $e(x)$) are differentiable up to μ_ith order at all the touch points. The function $g_i(x)$, formed by differentiating the interpolated polynomial $p(x)$ with respect to a touch point x_{ti}, is therefore itself an Nth order polynomial of x. It can be calculated by interpolation subject to the following constraints:

$$g_i^{(r)}(x_{tj}) = 0 \quad r = 0, 1, \ldots, \mu_j - 1 \quad j = 1, 2, \ldots, M \quad j \neq i \tag{13a}$$

$$g_i^{(\mu_i - 1)}(x_{ti}) = e^{(\mu_i)}(x_{ti}) \quad r = 0, 1, \ldots, \mu_i - 1 \tag{13b}$$

Proof: Eqn. 13*a* is evident as the μ_j interpolated derivatives of $p(x_{tj})$ are fixed with respect to another touch point x_{ti}, $i \neq j$. The proof of eqn. 13*b* follows. Suppose that one of the touch points on $u(x)$, x_{ti}, changes to $x'_{ti} = x_{ti} + h$ and the ordinate from $u(x_{ti})$ to $u(x_{ti} + h)$. Define the new polynomial interpolated from $\{x_{t1}, x_{t2}, \ldots, x'_{ti}, \ldots, x_{tM}\}$ by $p_h(x)$. As the polynomial is interpolated up to $\mu_i - 1$th tangency to $u(x)$ at this touch point,

$$p_h^{(r)}(x_{ti} + h) = u^{(r)}(x_{ti} + h) \quad r = 0, 1, 2, \ldots, \mu_i - 1 \tag{14}$$

Expand $p_h(x)$ at x'_{ti} by a Taylor series and evaluate $p_h^{(r)}(x)$ at $x = x_{ti}$ and notice (eqn. 14)

$$p_h^{(r)}(x_{ti}) = \left\{ \sum_{k=0}^{\infty} \frac{p_h^{(k)}(x_{ti} + h)}{k!}(-h)^k \right\}^{(r)} = u^{(r)}(x_{ti} + h) - p_h^{(r+1)}(x_{ti} + h)h + O(h^2) \tag{15}$$

for $r = 0, 1, 2, \ldots, \mu_i - 1$.

$$g_i^{(r)}(x_{ti}) = \left. \frac{\partial p^{(r)}(x)}{\partial x_{ti}} \right|_{x = x_{ti}} = \lim_{h \to 0} \frac{p_h^{(r)}(x_{ti}) - p^{(r)}(x_{ti})}{h} \tag{16}$$

$$= \lim_{h \to 0} \frac{[u^{(r)}(x_{ti} + h) - p_h^{(r+1)}(x_{ti} + h)h] - u^{(r)}(x_{ti}) + O(h^2)}{h} \tag{17}$$

$$= u^{(r+1)}(x_{ti}) - p_h^{(r+1)}(x_{ti}) \tag{18}$$

Eqn. 13*b* follows by noting that $\lim p_h^{(r+1)}(x_{ti} + h) = p^{(r+1)}(x_{ti})$ as $h \to 0$ and the definition of $e(x)$ for $r = 0, 1, 2, \ldots, \mu_i - 1$. In general, the above proof can be applied to all $\{x_{ti}\}$, which may be touch points on either $u(x)$ or $l(x)$.

Remark 2: The Newton system can be solved for the touch point increments $\{\Delta x_{ti}\}$ by

$$\Delta x_{ti} = \frac{q^{(\mu_i - 1)}(x_{ti})}{g_i^{(\mu_i - 1)}(x_{ti})} = \frac{q^{(\mu_i - 1)}(x_{ti})}{e_i^{(\mu_i)}(x_{ti})} \quad i = 1, 2, \ldots, M \tag{19}$$

Proof: A single row of the Jacobian system (eqn. 9) is

$$\sum_{i=1}^{M} g_i(x_{mj})\,\Delta x_{ti} = e(x_{mj}) \quad j = 1, 2, \ldots, M \tag{20}$$

and define

$$q(x) = \sum_{i=1}^{M} g_i(x)\Delta x_{ti} \tag{21}$$

From theorem 1, $q(x)$ is also a polynomial and meets the constraints of eqn. 11. Substitute $x = x_{tj}$ into eqn. 21 and

$$q_j^{(\mu_j - 1)}(x_{tj}) = \sum_{i=1}^{M} g_i^{(\mu_j - 1)}(x_{tj})\,\Delta x_{ti} = g_j^{(\mu_j)}(x_{tj})\,\Delta x_{tj} \tag{22}$$

Eqn. 19 follows.

From eqns. 11*b* and 6*a*, x_{ti} is a $(\mu_i - 2)$th order zero of $q(x)$ and $(\mu_i - 1)$th order zero of $e(x)$. They can be expanded by a Taylor expansion at x_{ti} as

$$q(x) = A_q(x - x_{ti})^{\mu_i - 1} + O[(x - x_{ti})^{\mu_i}] \tag{23}$$

$$e(x) = A_e(x - x_{ti})^{\mu_i} + O[(x - x_{ti})^{\mu_i + 1}] \tag{24}$$

From eqns. 19, 23 and 24,

$$\Delta x_{ti} = \frac{q^{(\mu_i - 1)}(x_{ti})}{e^{(\mu_i)}(x_{ti})} \cong \frac{(\mu_i - 1)!\,A_q}{\mu_i!\,A_e} = \frac{A_q}{\mu_i A_e} \tag{25}$$

From eqns. 23, 24 and 25, it is easily seen that eqn. 12 is true.

As both numerator and denominator of eqn. 19 tend to zero at x_{ti}, each touch point increment Δx_{ti} can only be calculated from the limiting values of the increment polynomial $q(x)$ and error function $e(x)$ in the proximity of the touch point, $x_{ti} + h$. The distance h must be chosen suitably according to word length and order of touch point. A suggested rule is $\mu_i \times 10^{-6}/N$ for double precision arithmetic.

The computational cost of the whole procedure is very small as it involves only repeated interpolation. The $O(n^3)$ step of solving the Newton system has been reduced to an $O(n^2)$ interpolation. Each evaluation of the interpolated polynomial costs $O(n)$ multiplications.

2.5 Generalised Remez methods

As has been shown, Δx_{ti} can be approximately evaluated by $\delta_i(x)$ at a point close to x_{ti}. If this point is selected as $x = x_{mi}$, a very simple adjustment to the touch point positions is revealed (notice eqn. 11*a*)

$$\Delta x_{ti} = \delta_i(x_{mi}) = \frac{(x_{mi} - x_{ti})q(x_{mi})}{\mu_i e(x_{mi})} = \frac{x_{mi} - x_{ti}}{\mu_i} \tag{26}$$

In the special case of the curve-fitting problem with all $\mu_i = 1$, eqn. 26 results in the well-known Remez method which updates the variable vector by

$$\{x_{ti}\} \leftarrow \{x_{mi}\} \quad i = 1, 2, \ldots, M \tag{27}$$

This indicates that the interpolation ordinates are simply exchanged with the locations of the extrema and reinterpolated (Fig. 2). It may be expected that ordinates separated by an excessively large ripple will be brought together and those separated by an insufficiently large ripple will be moved apart. When the $\{x_{ti}\}$ are close to the solution, the $\{x_{mi}\}$ are also close to $\{x_{ti}\}$, and the adjustment given by eqn. 26 becomes similar to that given by a Newton method. This confirms that the Remez method achieves the good convergency of Newton iteration on approach to the solution. Convergency of this algorithm is guaranteed [21] for sufficiently large N, and it has been widely adopted in FIR and IIR digital filter design [24–26]. For the case of $\mu_i > 1$, the simple exchange process of eqn. 27 is unsuitable. Instead, the adjustment given by eqn. 26 is applicable

$$\{x_{ti}\} \leftarrow \{x_{ti} + (x_{mi} - x_{ti})/\mu_i\} \quad i = 1, 2, \ldots, M \tag{28}$$

This can be seen as a generalisation of the Remez method of eqn. 27 in which, instead of moving the abscissa all the way to the extremum, it is moved by a fraction of the distance dependent on the order of the touch point.

2.6 Unilateral method

In most filter applications, emphasis is given to one of the bounding functions. For example, in the passband region of a filter, $u(x)$ is most important as it determines the points of maximum transmission. All the high-order touch points (with $\mu_i > 1$) could be assigned to $u(x)$ for greatest effect. In a unilateral method the $N_c - M$ specifications can be met by directly interpolating $p(x)$ to μ_ith order tangency at all the touch points on $u(x)$. The lower curve $l(x)$ is used only as a bound for the ripple, so that all touch points on $l(x)$ should be adjusted to $\mu_i = 1$ (Fig. 3). The difference between $p(x)$ and $l(x)$ is used as the objective function. Only half of the touch points are kept as variables compared with the bilateral method.

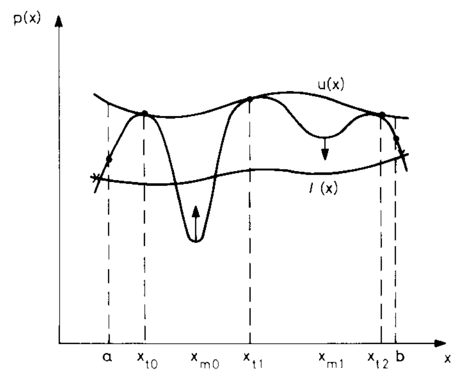

Fig. 3 *Unilateral approximation*

2.7 Computer algorithm

The approximation methods proposed in Section 2.6 can be implemented on a computer by the following algorithm:

Step 1: Read the boundary functions $u(x)$ and $l(x)$ as piecewise linear functions on range $[a, b]$. Read the number of touch points and specified orders μ_i.

Step 2: Distribute $\{x_{ti}\}$ uniformly over $[a, b]$ assigning $x_{t0} = a$ and $x_{tn} = b$.

Step 3: Interpolate $\{x_{ti}\}$ alternately to boundaries $u(x)$ and $l(x)$.

Step 4: Set $x_{m0} = a$

$i = 0, 2, 4, \ldots, n$; choose x_{mi} to maximise

$$\{p(x) - u(x),\ x_{mi-1} \leqslant x \leqslant x_{ti+1}\}$$

$i = 1, 3, 5, \ldots, n - 1$; choose x_{mi} to maximise

$$\{l(x) - p(x),\ x_{mi-1} \leqslant x \leqslant x_{ti+1}\}$$

Step 5: Compute improved touch point estimates by one of the methods in Section 2.3. The extended Remez method indicates that

$$x_{ti} = x_{ti} + ((x_{mi} - x_{ti})/\mu_i)$$

Step 6: Compute convergence estimate for kth iteration as

$$\varepsilon_k = \frac{\sum_{i=1}^{n} e(x_{mi})}{\sum_{i=1}^{n} |u(x_{mi}) - l(x_{mi})|}$$

Terminate if $\varepsilon_k <$ tolerance or $\varepsilon_k > \varepsilon_{k+1}$ (divergent)

2.8 Software considerations

2.8.1 Interpolation: The interpolation of the polynomial at Step 3 can be done by *osculatory* Newton interpolation. This is an extended form of the well known Newton interpolation, whereby a number of derivatives can be matched to the specified function.

It is particularly important in a filter problem to take care with the accuracy of construction and representation of the polynomials within finite word length arithmetic. Cancellation errors are particularly severe as the touch points of a filter function become closely spaced near a band edge. A typical calculation would be

$$f[x_{t0}, x_{t1}] = \frac{f(x_{t0}) - f(x_{t1})}{x_{t0} - x_{t1}} \tag{29}$$

where cancellation errors occur in numerator and denominator as x_{t0} approaches x_{t1}.

The effects of these cancellation errors can be minimised by calculating interpolated values by the zig-zag path method [22]. The principle is that a path is taken through the Newton table, such that the coefficients with the largest errors are multiplied by abscissae with the smallest differences. By using this accurate interpolation and representation of polynomials, high-order functions can be obtained (up to length 110 FIR designs). This avoids the complications of transformed variable methods [1, 2].

2.8.2 Searching: At Step 4 a search must be made for the touch point extrema. For reliability, the best method is found to be a single linear search over a uniform grid of points. Normally only 10–20 points are required per touch point for a terminating accuracy better than 1%. The linear search requires a fairly large number of function evaluations for higher order approximation. Faster searching is available by applying cubic, quadratic or golden section search methods which require only five or six steps per touch point for very high accuracy (10^{-8}%). However, when the attenuation boundaries are specified in piecewise linear form and not by a smooth function with continuous derivatives, these search methods can mislead and may determine the extrema erroneously. A combination of the reliability of the linear search with the speed of gradient search methods is discussed by Antoniou [27].

2.8.3 Cluster method: In most cases, the boundary functions are only given by values and the derivatives are not available. Although the derivatives can be calculated by numerical differentiation, this is notoriously inaccurate for high orders. The polynomial obtained by a Newton interpolation may become totally unreliable in the neighbourhood of a high-order touch point. A better conditioned method is to interpolate the polynomial at a *cluster* of points with first-order tangency to the boundary function. A μ_ith-order touch point with μ_i odd requires $(\mu_i + 1)/2$ first-order touch points distributed in the neighbourhood of x_{ti}. In practice, it is found that a spacing of 10^{-6} (with normalised passband width of 1) can be chosen. The error caused by this approximate method can be controlled and made much smaller than the allowed ripple [the separation of $u(x)$ and $l(x)$].

2.8.4 Damping: The term damping is used for the process whereby the step sizes determined by Newton's method may be reduced to avoid divergence. A form of damping can be useful where Newton's method is used to predict adjustments Δx_{ti}, and the touch points would cross one another or move entirely outside the approximating region $[a, b]$. In these cases (usually far from solution), it is found useful to limit the movement of the touch points to half the distance in the direction of their closest neighbour. In this way, no touch point may cross or escape the region $[a, b]$, and yet the direction required to reduce the extrema is observed.

2.8.5 Convergency, accuracy and storage: Computation costs are $O(n^2)$ for passband approximation. Stopband approximation requires solution of a matrix system with $O(n^3)$ efficiency. Convergence is quadratic near solution, a property of algorithms based on Newton's method. Divergence occurs only in those cases where the boundary functions are too severe for the selected order of the function. The accuracy of the algorithm is limited solely by the fineness of the search grid used to determine the positions of the extrema. Storage is dominated by the matrix system and Newton interpolation tables and is of $O(n^2)$ size.

3 Rational approximation

3.1 Approximation method for rational functions

In this section, a design technique for rational filter transfer functions will be considered. The filter amplitude specifications need not be ideal and can have arbitrary weightings in both passband and stopband. The approximating function can be designed in a minimax fashion, with high-order touch points assigned to certain positions in each band. Classical functions result as special cases from a general algorithm.

For simplicity, a lowpass filter specification will be considered first. The filter specification is defined as a piecewise template of attenuation in dB against frequency in Hz. The following parameters must be specified by a designer:

f_{plo}, f_{phi}: passband edge frequencies (Hz)
f_{slo}, f_{shi}: stopband edge frequencies (Hz)
NN, ND: numerator and denominator orders

The transfer function to be designed is

$$T(x) = \frac{N(x)}{D(x)} \quad x = \omega^2 \tag{30}$$

The zeros of $T(x)$ are contained in the numerator polynomial. In a filter transfer function they are most effectively assigned to the imaginary axis of the s-plane (real x-plane locations) and placed in the stopband region for maximum attenuation. By making this restriction, the rational function becomes a minimum phase function. The denominator polynomial contains the complex pole locations which must be positioned to control the passband transmission characteristics.

The following procedure is used to design a rational filter approximation.

Step 1: Read ND, $NN, f_{plo}, f_{phi}, f_{slo}, f_{shi}$, touch point orders $\{\mu_{tsi}\}$, $\{\mu_{tpi}\}$ and piecewise linear descriptions of $L(\omega)$ and $U(\omega)$.

Step 2: Initialise $\{x_{tpi}\}$ in the passband region $[x_{plo}, x_{phi}]$ and $\{x_{tsi}\}$ in the stopband region $[x_{slo}, x_{shi}]$, equidistantly spaced. Set $N(x)$ to 1.

Step 3: Solve the passband approximation problem on $[x_{plo}, x_{phi}]$ using $D(x)$ such that

$$u(x) = N(x)/L(x)$$

$$l(x) = N(x)/U(x)$$

and compute initial convergence estimate ε_p.

Step 4: Solve stopband approximation problem on $[x_{slo}, x_{shi}]$ using $N(x)$ such that

$$u(x) = \beta U(x)/D(x)$$

$$l(x) = \beta L(x)/D(x)$$

and compute initial convergence estimate ε_s.

Step 5: Terminate if ε_p and $\varepsilon_s <$ tolerance *or* $k >$ maxiter.

Owing to the special properties of the numerator and denominator polynomials, two different approaches are appropriate for solving steps 3 and 4. Note that a multiplying factor β is introduced in step 4 so that the stopband attenuation will only be met to a constant dB error. In general, it is not possible to meet the stopband and passband specifications exactly and some error margin must be allowed in either passband or stopband, or both. In this approach, the passband specifications will be met as closely as possible, and the stopband attenuation characteristics will have some error above or below the specified attenuation. This expression of the filtering problem is useful in practice, as good control of the passband characteristics is usually of greater importance than the stopband. Note that if the factor β is less than 1, then the specifications have been exceeded and it may be possible to reduce the order of the function or the number of zeros. Conversely, if $\beta > 1$ the order must be increased or more zeros should be introduced.

3.1.1 Passband design

Any of the methods of Sections 2.3–4 are suitable for the design of the passband function $D(x)$. It is found that the bilateral method has very good global convergence. The unilateral method is then useful to ensure that the function does not exceed maximum transmission (i.e. $0 < T(x) < 1$) for passive filter realisation. The touch points are all fixed to the upper boundary.

3.1.2 Stopband design

The numerator polynomial is of the following particular form:

$$N(x) = Kx^{n0} \prod_{i=1}^{nf} (x - x_{tsi})^{\mu_{tsi}} \tag{31}$$

This corresponds to a special case of the unilateral method where all touch points are tangential to the lower boundary which is zero. It remains to compute the attenuation margin β (Fig. 4). Two methods can be

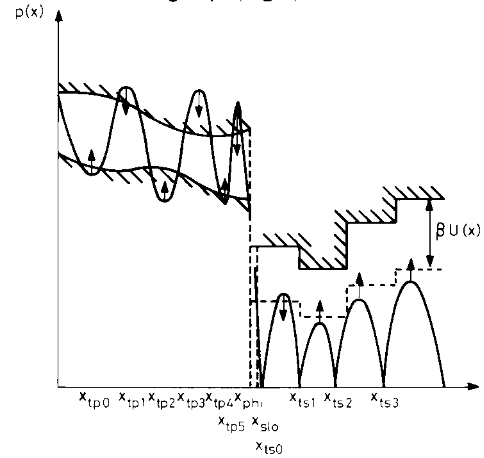

Fig. 4 *Scheme for approximation of an arbitrary lowpass filter specification by a rational function*

applied: the heuristic method [26] permits the approximation methods of this section to be used. However, a variant of the method of Temes and Smith [1] has proved to be more stable and more easily extended to multiband approximations. A Jacobian matrix of the peak positions with respect to the touch points must be set up and solved.

$$\left[\begin{array}{ccc|c} \dfrac{\partial T(x_{ms0})}{\partial x_{ts0}} & \cdots & \dfrac{\partial T(x_{ms0})}{\partial x_{tsn}} & -U(x_{ms0}) \\ \vdots & & \vdots & \vdots \\ \dfrac{\partial T(x_{msn})}{\partial x_{ts0}} & \cdots & \dfrac{\partial T(x_{msn})}{\partial x_{tsn}} & -U(x_{msn}) \end{array}\right] \begin{bmatrix} \Delta x_{ts0} \\ \vdots \\ \Delta x_{tsn} \\ \beta \end{bmatrix} = \begin{bmatrix} -T(x_{ms0}) \\ \vdots \\ -T(x_{msn}) \end{bmatrix} \tag{32}$$

and because

$$\frac{\partial T(x)}{\partial x_{tsi}} = \frac{-\mu_i}{(x - x_{tsi})} T(x) \tag{33}$$

Eqn. 32 can be rearranged and evaluated as

$$\left[\begin{array}{ccc|c} \dfrac{\mu_0}{(x_{ms0} - x_{ts0})} & \cdots & \dfrac{\mu_n}{(x_{ms0} - x_{tsn})} & \dfrac{U(x_{ms0})}{T(x_{ms0})} \\ \vdots & & \vdots & \vdots \\ \dfrac{\mu_0}{(x_{msn} - x_{ts0})} & \cdots & \dfrac{\mu_n}{(x_{msn} - x_{tsn})} & \dfrac{U(x_{msn})}{T(x_{msn})} \end{array}\right] \times \begin{bmatrix} \Delta x_{ts0} \\ \vdots \\ \Delta x_{tsn} \\ \beta \end{bmatrix} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \tag{34}$$

The touch points are updated as

$$x_{tsi} = x_{tsi} + \Delta x_{tsi} \tag{35}$$

and some damping may be necessary. Note that the constant K must also be determined. A good method of assigning a value to K is to fix the passband edge position between passband and stopband iterations.

$$K = L(x_{phi})/T(x_{phi}) \tag{36}$$

3.2 Passive ladder design

Often the transfer function designed by the methods of previous sections will be decomposed into a passive ladder prototype for later simulation by active circuits. This is an error-prone numerical procedure. In this section, it will be demonstrated that the ladder can be designed directly from the touch points to provide high accuracy at the ripples that characterise the filter response.

Orchard has proposed a very simple but efficient algorithm to design an RLC ladder from a given reflection function $\rho(\omega)$ [28]. The structure of the ladder is prescribed and only the component values remain to be determined. A set of real and imaginary parts $\{\mathrm{Re}\,[\rho(\omega_{ti})], \mathrm{Im}\,[\rho(\omega_{ti})]\}$ are used to set up the objective function vector $\boldsymbol{F}$ for Newton-type iteration, and component values $\{y_k\}$ form a vector of variables $\boldsymbol{Y}$ [22]. The core of Orchard's algorithm is an elegant, well conditioned method to compute $\boldsymbol{F}$ and the Jacobian matrix of derivatives

$$\mathrm{Re}\left[\frac{\partial\rho(j\omega_{tk})}{\partial y_i}\right] \quad \text{and} \quad \mathrm{Im}\left[\frac{\partial\rho(j\omega_{tk})}{\partial y_i}\right]$$

by chain matrix calculations.

In the case of certain classical approximations, where the points of maximum or minimum transmission ($\rho(j\omega) = 0$ or $\rho(j\omega) = 1$) are known, the explicit calculation of $\rho(j\omega)$ is not necessary for Orchard's algorithm. However, in general, Orchard's method requires the formation of $\rho(j\omega)$ by Hurwitz factorisation of $|\rho|^2$ as in classical synthesis, which is an ill-conditioned procedure [1]. An extension of Orchard's method is described below which works with more general forms of $|\rho|^2$ but eliminates any root-finding requirement.

The value of $|\rho|^2$ and its deriatives at the touch points $\{x_{ti}\}$ can be chosen as the objective function for the Newton scheme. The derivatives of $|\rho|^2$ with respect to the element values $\{y_k\}$ are required for the construction of the Jacobian matrix and are given by (let $x_{ti} = \omega_{ti}^2$)

$$\frac{\partial}{\partial y_k}\left(\frac{d^r|\rho(\omega_{ti})|^2}{dx^r}\right) = 2\frac{d^r}{d\omega^r}\left\{\mathrm{Re}\left[\bar{\rho}(j\omega_{ti})\frac{\partial\rho(j\omega_{ti})}{\partial y_k}\right]\right\} \tag{37}$$

for $r = 0, \ldots, \mu_i$ and $i = 1, 2, \ldots, M$.

Notice that here $\bar{\rho}$ (the conjugate of ρ) and $\partial\rho/\partial y_k$ are obtained from the approximate network with guessed component values, which can be generated by Orchard's algorithm and then the remaining part of eqn. 37 can be calculated by a numerical differentiation. Here it is also found to be efficient to use the 'cluster' method mentioned in Section 2.8.3. The objective function $|\rho(\omega_{ti})|^2$ and derivatives are obtainable from a unilateral passband approximation of Section 2.6. This provides a direct link between approximation and ladder design procedures, bypassing the traditional Hurwitz factorisation step.

3.3 Computed examples

A series of computed approximation examples is now given to illustrate the power and flexibility of the above methods. Fig. 5 shows a polynomial (FIR) approximation to arbitrarily shaped boundaries. A touch point of fifth-order tangency is seen at the centre of the function. FIR approximations up to length 110 ($N = 55$) have been obtained using double precision arithmetic.

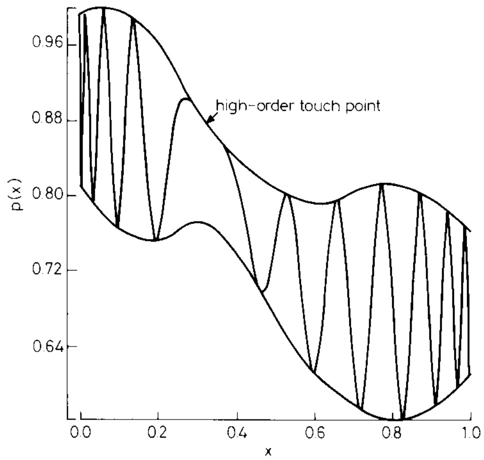

Fig. 5 *Length 51 FIR approximation to arbitrary boundaries*

Arbitrary rational function (IIR) filter responses are shown in Figs. 6–7. An 11th-order function is fitted touching the passband boundaries and meeting the stopband boundary to within some constant dB error (Fig. 6). A bandpass approximation with asymmetric stopband and sagging passband is shown in Fig. 7. High-order touch points can be placed in the passband to trade off between group delay, stopband attenuation and passive sensitivity properties. A series of such passbands is shown in Fig. 8, including inverse Chebyshev and elliptic forms as special cases. The effects on stopband attenuation and group delay is shown in Table 1. As high-order touch points are introduced into the response, there is a significant reduction in the peaks of the group delay, which follows from the progressive smoothing of the amplitude characteristic. There is also an accompanying improvement in sensitivity behaviour of the resulting passive filter realisation, as would be expected from Orchard's criterion [29], since the higher order touch points ensure adherence to maximum transmission over a wider interval. This property is inherited by active simulations, where sensitivity considerations are of major concern. There is some attendant cost, with a corresponding loss in stopband attenuation. Trade-off between these factors must be considered in design. It is obviously not possible to dictate the location, sequence and order of the touch points in a general manner, but it is recommended that they are deployed in regions where the filter characteristic is most critical.

Practical examples of the use of amplitude-shaping capabilities are given in Figs. 9–10. In Fig. 9 a length 69 differentiator is designed with a high-order touch point at 0 Hz for increased DC linearity [4]. A 14th-order wideband SC filter with 200 kHz clock frequency shows a 1 dB passband droop due to sinc (x) weighting. The original 0.1 dB flat equiripple passband behaviour can be restored by prewarping the filter response upwards. Attenuation line-weighting and LDI ladder termination distortion can be treated in a similar manner. This technique is used in a more general sense (Fig. 11) to compensate for distortion caused by switch and amplifier

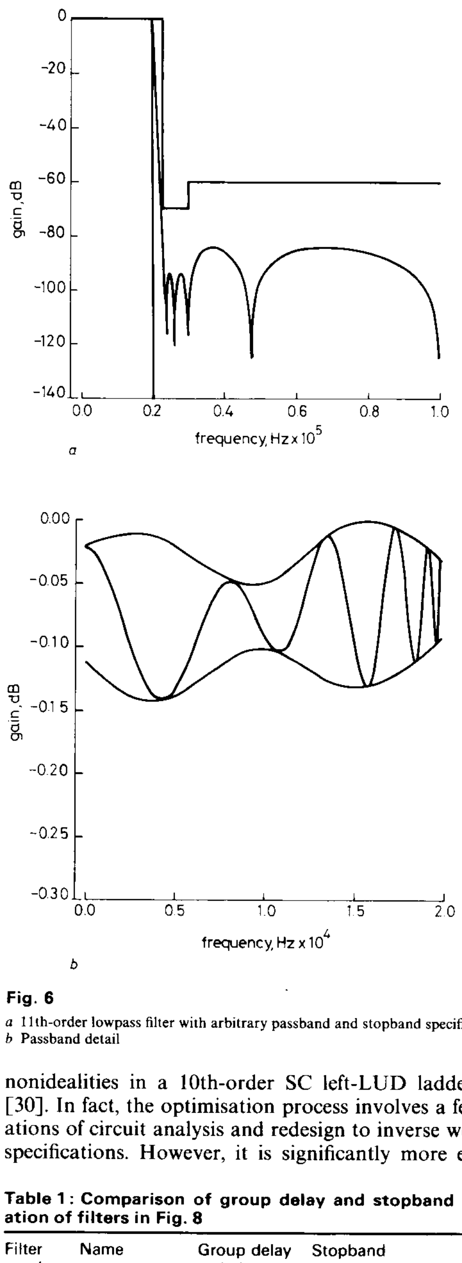

Fig. 6

a 11th-order lowpass filter with arbitrary passband and stopband specifications
b Passband detail

nonidealities in a 10th-order SC left-LUD ladder filter [30]. In fact, the optimisation process involves a few iterations of circuit analysis and redesign to inverse weighted specifications. However, it is significantly more efficient

Table 1: Comparison of group delay and stopband attenuation of filters in Fig. 8

Filter number	Name	Group delay variation, ms	Stopband rejection, dB
1	inverse Chebyshev	0.7	22
2	2-10-2	1.35	43
3	4-6-4	1.4	47
4	2-2-6-2-2	1.9	53
5	6-2-6	1.3	48
6	4-2-2-2-4	1.6	55
7	elliptic	2.2	58

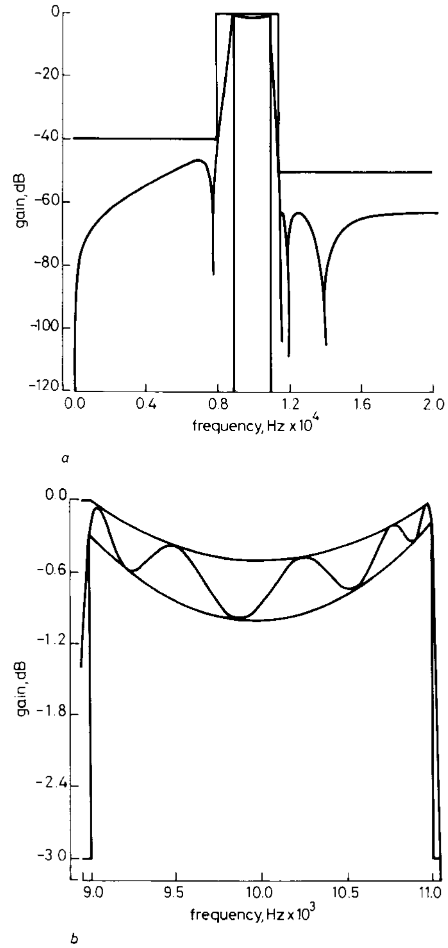

Fig. 7

a 10th-order asymmetric bandpass filter with sagging passband
b Passband detail

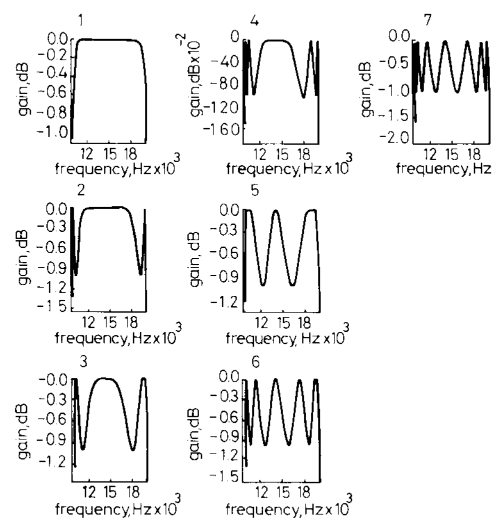

Fig. 8 *Sequence of 14th-order filter passbands employing high-order touch points*

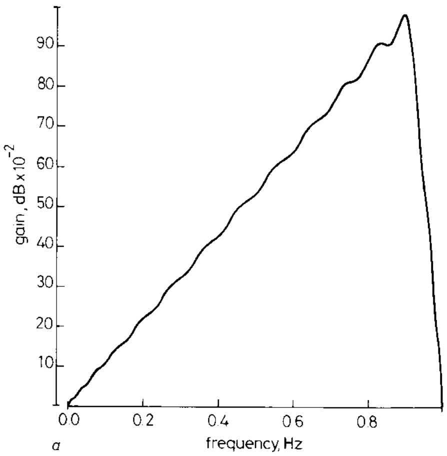

Fig. 9A *Length 69 FIR differentiator*

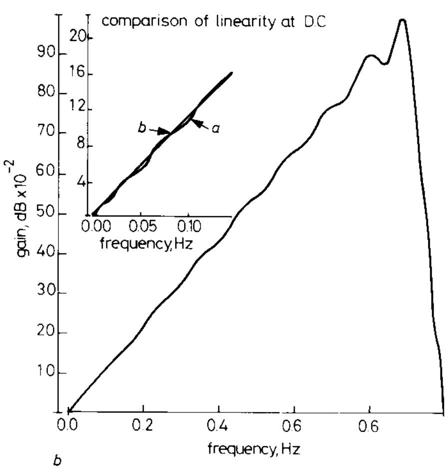

Fig. 9B *Length 69 FIR differentiator with 13th-order touch point at 0 Hz (linearity comparison shown as inset)*

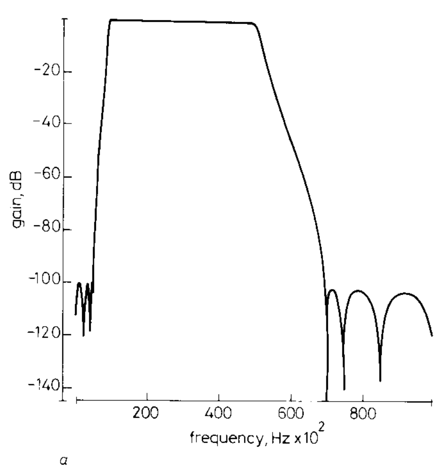

Fig. 10A *Wideband bandpass SC filter with sinc (x) distortion*

than a full circuit optimisation as the computation is dependent on the order of the filter rather than the number of components in the circuit (e.g. a 10th-order filter will be realised as a circuit with 40 components).

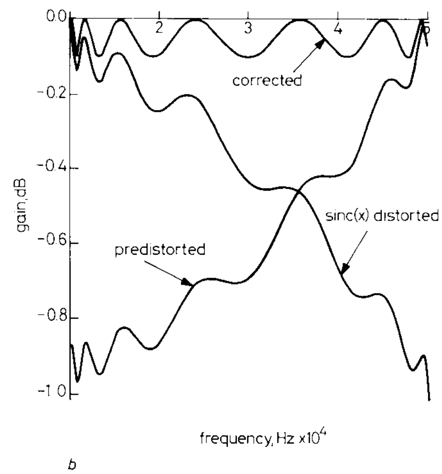

Fig. 10B *Sinc (x) correction*

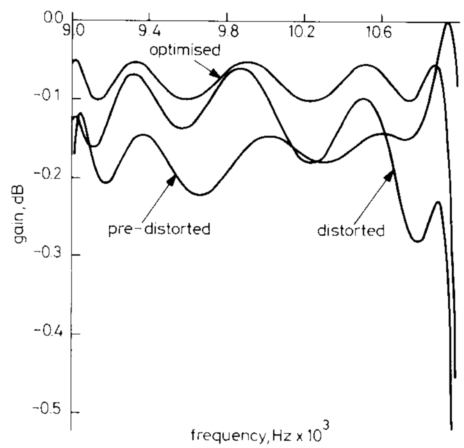

Fig. 11 *Optimisation of distortion in left-LUD SC filter due to op-amp and switch nonidealities*

4 Group delay approximation by allpass functions

The allpass function in the continuous time s-domain is

$$T(s) = \frac{X(-s)}{X(s)} \tag{38}$$

and the group delay function can be expressed as

$$\tau(\omega) = \left. \frac{2 \operatorname{Re}\left\{X(s)\dfrac{dX(-s)}{ds}\right\}}{X(s)X(-s)} \right|_{s=j\omega} = \frac{N(\omega)}{D(\omega)} \tag{39}$$

As the denominator of eqn. 39 is a magnitude-squared function, it can be designed by standard Remez-type methods. The numerator function is an even function of s which can be formed by Hurwitz factorisation of the denominator polynomial.

4.1 New Remez-type algorithm

Consider now the problem of fitting the delay function to lower and upper boundaries $L(\omega)$ and $U(\omega)$ in a minimax sense over a frequency interval ω_{lo} to ω_{hi} (Fig. 12). It is required that

$$L(\omega) + C \leqslant \frac{N(\omega)}{X(j\omega)X(-j\omega)} \leqslant U(\omega) + C \tag{40}$$

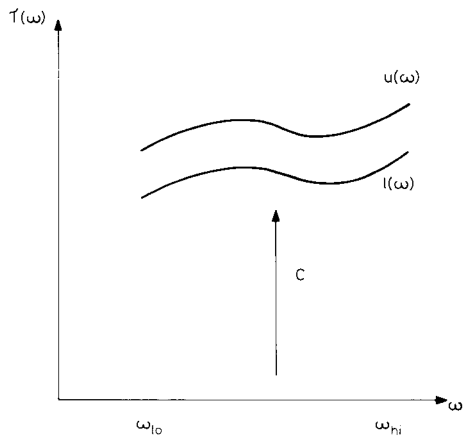

Fig. 12 *Approximation scheme for allpass group delay*

where the unknown constant delay offset C is necessary to ensure that

$$\int_0^{\infty} \tau(\omega)\, d\omega = n\pi \tag{41}$$

The constant C can be added without affecting the relative delay variation over the approximating region.

The allpass approximation method may be summarised by the following steps:

Step 1: Read lower and upper delay boundaries as piecewise boundaries $L(\omega)$ and $U(\omega)$ and equaliser order n.

Step 2: Set numerator function $N(\omega) = 1$ and guess constant $C = n\pi/(\omega_{hi} - \omega_{lo})$

Step 3: Apply Remez approximation techniques to solve

$$\begin{aligned} u(\omega) &= N(\omega)/(L(\omega) + C) \\ l(\omega) &= N(\omega)/(U(\omega) + C) \end{aligned} \tag{42}$$

over the range ω_{lo}–ω_{hi} using $D(x)$.

Step 4: Recalculate C as average delay constant between specified and approximated $\tau(\omega)$ over $\omega_{lo} to\ \omega_{hi}$.

Step 5: Form numerator function by Hurwitz factorisation of $D(\omega)$. Let the roots be $s_i = -a_i + jb_i$, then the delay function is

$$\tau(\omega) = 2\sum_{i=1}^{n/2} \left\{ \frac{a_i}{a_i^2 + (\omega + b_i)^2} + \frac{a_i}{a_i^2 + (\omega - b_i)^2} \right\} \tag{43}$$

The numerator function can be calculated from eqns. 43 and 44

$$N(\omega) = D(\omega) \times \tau(\omega) \tag{44}$$

Step 6: Repeat from step 2 until converged.

4.2 Computer implementation

The factorisation at step 5 can be made very efficient by utilising root positions from the previous factorisation as good initial guesses of roots for the present factorisation. Using Muller's quadratic interpolation method, this typically only requires two to three iterations per root [31].

Accuracy is preserved in the algorithm by avoiding representation of polynomials in coefficient form. Instead, Newton interpolated form is used at step 3 and factored form at step 4. Both forms are well conditioned on the approximation region, allowing high-order functions and narrow band allpass functions to be designed.

No theoretical proof has been obtained of convergence. However, experience has shown that convergence is good and that five or six cycles will generally suffice. The mechanism of the algorithm is dependent on the similarity between the group delay function $\tau(\omega)$ and the denominator magnitude function $D(\omega)$ in eqn. 39. The numerator is observed to be a smooth function over the approximating region, which warps the delay function of the denominator. Further theoretical investigation is being undertaken.

Digital allpass functions can be obtained by bilinear transformation. The delay specifications must be prewarped by a factor of $\cos(\omega T/2)^2$.

Group delay equalisation can be performed by combining $L(\omega)$ and $U(\omega)$ with the additive inverse of the group delay function of the amplitude filter. In this case, the total group delay of the allpass and amplitude filter stages will meet the desired specifications.

4.3 Computed examples

High-order touch points can be introduced into the delay function by the approximation methods of Section 2. Fig. 13 shows a 12th-order maximally flat group delay response (11th-order touch point). Fig. 14 shows a stepped form of group delay. Fig. 15 shows a 28th-order stepped group delay response with a 5th-order touch point at the lower band edge. Finally, a 28th-order allpass function is employed to equalise the delay of a 10th-order elliptic amplitude filter to a variation of $\pm 50\ \mu s$ over the passband (Fig. 16).

4.4 Advantages

An iterative design procedure which works on the group delay function and employs Remez-type approximation has been proposed. Advantages of this algorithm are as follows:

(1) Good initial guesses of parameters are not required for convergence. As with the Remez approximation the

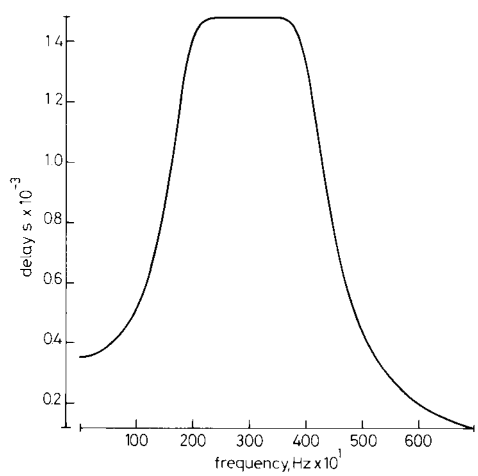

Fig. 13 *12th-order maximally flat allpass group delay approximation*

interpolation abscissae can be arbitrarily spaced on the approximation region (normally equidistantly).

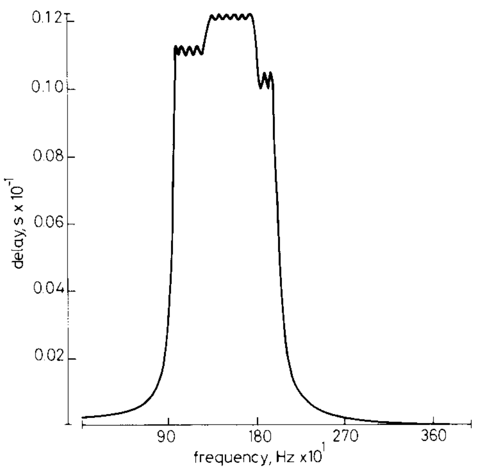

Fig. 14 *28th-order stepped allpass group delay approximation*

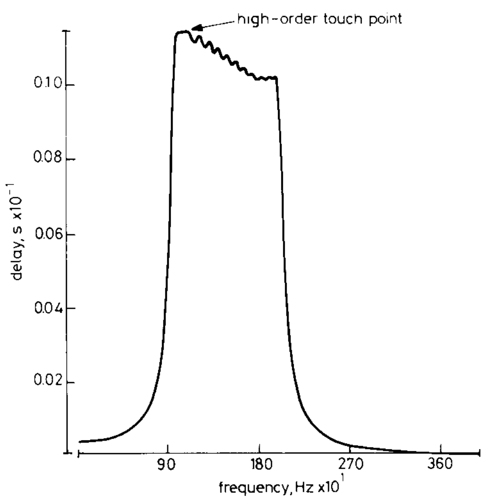

Fig. 15 *28th-order allpass group delay approximation with high-order touch point*

(2) The algorithm is well conditioned. Accuracy is maintained by representing the design polynomials in either Newton or factored form, instead of the ill conditioned coefficient form. High orders (>40) and narrow band design can be obtained. High-order designs are of some interest for digital filters where a very selective linear phase filter is required, which would be too expensive in nonrecursive form.

(3) Stability of the solution is guaranteed at all stages of the algorithm. The roots of the denominator of the allpass function must lie in the left-half plane because of the Hurwitz factorisation step.

(4) Computational requirements are light. The process only involves the fast Remez exchange and a factorisation step.

(5) The convergence of the algorithm is good.

In computer terms, this algorithm can be conveniently combined with amplitude approximation software, since it draws on the same numerical methods of interpolation and factorisation.

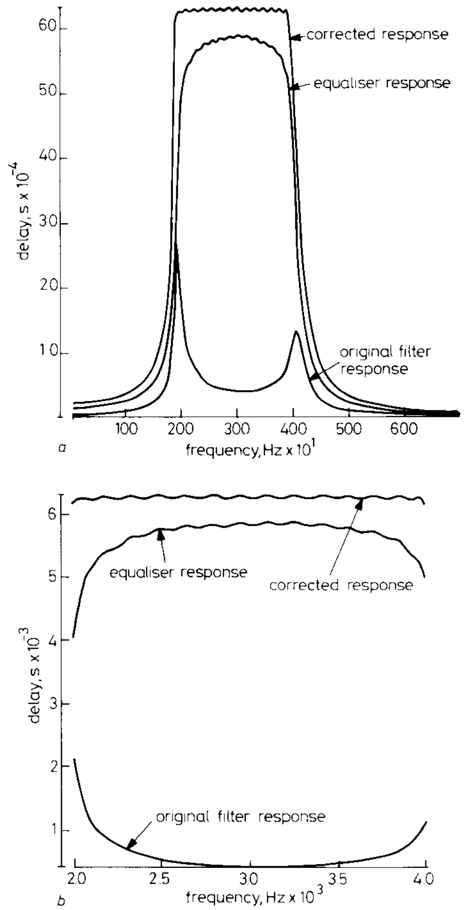

Fig. 16

a Equalisation of 10th-order elliptic filter delay response by 28th-order allpass function
b Passband detail

5 Conclusions

The amplitude and group delay response of filters can be designed with great flexibility by a series of extended Remez exchange algorithms. The amplitude response can be weighted arbitrarily in passband and stopband. High-order touch points have been introduced as a generalisation of the idea of maximum flatness. They are shown to offer the designer freedom to influence the filter characteristic in critical regions, when trade-off in the properties of group delay, passive sensitivity and stopband attenuation can be effected. Similar manipulations are possible in the group delay of an allpass function. The new algorithms are fast and efficient and have found successful application in the PANDDA filter compiler [30].

6 Acknowledgments

The authors would like to acknowledge the financial support of GEC Hirst Research Centre and the UK Science and Engineering Research Council.

7 References

1 SEDRA, A.S., and BRACKETT, P.O.: 'Filter theory and design: active and passive' (Pitman, London, 1979)
2 DANIELS, R.W.: 'Approximation methods for electronic filter design' (McGraw-Hill, New York, 1974)
3 NAKHLA, M.S.: 'Approximation of lowpass filters with frequency dependent input gain characteristic', *IEEE Trans.*, 1979, **CAS-26**, (3), pp. 198–202
4 MEDLIN, G.W.: 'A new design technique for maximally linear differentiators'. Proceedings of the IEEE ICASSP, Glasgow, June 1989, pp. 825–828
5 FILHO, S.N., SEARA, R., and BERMUDEZ, J.C.M.: 'A new method for the compensation of the sin $(x)/x$ distortion in discrete time to continuous time signal conversions'. Proceedings of the IEEE ISCAS, Portland, Oregon, May 1989, pp. 1668–1671
6 DE QUEIROZ, A.C.M., and CALOBA, L.P.: 'Physically symmetrical and antimetrical ladder filters with finite transmission zeros'. Proceedings of the 30th Midwest Symposium on Circuits and Systems, Syracuse, NY, 1987, pp. 639–643
7 VAIDYANATHAN, P.P.: 'Optical design of linear-phase FIR digital filters with very flat passbands and equiripple stopbands', *IEEE Trans.*, 1985, **CAS-32**, (9), pp. 904–917
8 CHARALAMBOUS, C., and KHREISHI, H.: 'Minimax design of one-dimensional recursive digital filters of a given approximation type: a closed form approach', *IEE Proc. G, Circuits, Devices and Systems*, 1987, **134**, (1), pp. 38–44
9 STEIGLITZ, K.: 'Computer-aided design of recursive digital filters', *IEEE Trans.*, 1970, **AU-18**, (2), pp. 123–129
10 TEMES, G.C., and CALAHAN, D.A.: 'Computer-aided network optimisation: the state-of-the-art', *Proc. IEEE*, 1967, **53**, pp. 1832–1863
11 REMEZ, E. YA.: 'General computation methods of Tchebysheff approximation'. Atomic Energy Translation 4491, Kiev USSR, 1957, pp. 1–85
12 LI PING, HENDERSON, R.K., and SEWELL, J.I.: 'A new filter approximation and design algorithm', Proceedings of the IEEE ISCAS, Portland, Oregon, May 1989, pp. 1063–1066
13 CORTELAZZO, G., and LIGHTNER, M.R.: 'Simultaneous design in both magnitude and group delay of IIR and FIR filters based on multiple criterion optimisation', *IEEE Trans.*, 1984, **ASSP-32**, (5), pp. 949–967
14 RHODES, J.D., and ZALABAWI, I.H.: 'Design of selective linear-phase filters with equiripple amplitude characteristics', *IEEE Trans.*, 1978, **CAS-25**, (12), pp. 989–1000
15 LI PING, and SEWELL, J.I.: 'Active and digital ladder-based allpass filters', *IEE Proc. G, Circuits, Devices and Systems*, 1990, **137**, (6), pp. 439–445
16 GREGORIAN, R., and TEMES, G.C.: 'Design techniques for digital and analog all-pass circuits', *IEEE Trans.*, 1978, **CAS-25**, (12), pp. 981–988
17 DECZKY, A.G.: 'Equiripple and minimax (Chebyshev) approximations for recursive digital filters', *IEEE Trans.*, 1974, **ASSP-22**, (2), pp. 98–111
18 CHARALAMBOUS, C., and ANTONIOU, A.: 'Equalisation of recursive digital filters', *IEE Proc. G, Circuits, Devices and Systems*, 1980, **127**, (5), pp. 219–225
19 JING, ZHONGQI: 'A new method for digital all-pass filter design', *IEEE Trans.*, 1987, **ASSP-35**, (11), pp. 1557–1564
20 HENDERSON, R.K., and SEWELL, J.I.: 'A design algorithm for all-pass delay equalisers'. Digest of IEE Saraga Colloquium on Electronic Filters, London, 1989, pp. 11/1–11/8
21 McCALLIG, M.T., and LEON, B.J.: 'Constrained ripple design of FIR digital filters', *IEEE Trans.*, 1978, **CAS-25**, (11), pp. 893–901
22 CONTE, S.D., and DE BOOR, C.: 'Elementary numerical analysis' (McGraw-Hill, New York, 1980)
23 HILDEBRAND, F.B.: 'Introduction to numerical analysis' (McGraw-Hill, New York, 1974, 2nd edn.)
24 RABINER, L.R., McCLELLAN, J.H., and PARKS, T.W.: 'FIR digital filter design techniques using weighted Chebyshev approximation', *Proc. IEEE*, 1975, **63**, (4), pp. 595–609
25 LIANG, J.-K., and DE FIGUEIREDO, R.J.P.: 'An efficient iterative algorithm for designing optimal recursive digital filters', *IEEE Trans.*, 1983, **ASSP-31**, (5), pp. 1110–1120
26 MARTINEZ, H.G., and PARKS, T.W.: 'Design of recursive digital filters with optimum magnitude and attenuation poles on the unit circle', *IEEE Trans.*, 1978, **ASSP-26**, (2), pp. 150–156
27 ANTONIOU, A.: 'New improved method for the design of weighted-chebyshev, nonrecursive digital filters', *IEEE Trans.*, 1983, **CAS-30**, (10), pp. 740–748
28 ORCHARD, H.J.: 'Filter design by iterated analysis', *IEEE Trans.*, 1985, **CAS-32**, (11), pp. 1089–1096
29 ORCHARD, H.J.: 'Inductorless filters', *Electron. Lett.*, 1966, **2**, pp. 224–225
30 HENDERSON, R.K., LI PING, and SEWELL, J.I.: 'A design program for digital and analogue filters: PANDDA'. Proceedings of ECCTD 89, Brighton, UK, Sept. 1989, pp. 289–293
31 MULLER, D.E.: 'A method for solving algebraic equations using an automatic computer', *Math. Tables and other Aids for Computation*, 1956, **10**, pp. 208–215