combined with correlatively encoded CPFSK signals. The multi-T realisation of CCE/2-h signals discussed in Reference 3, that alternate the modulation index between h_1 and h_2 at the end of every n intervals (or equivalently, signals that periodically select slots of nT for the two modulation indices) is considered here. As in Reference 3, correlative encoding that converts the message sequence $\boldsymbol{b} = (\ldots, b_{i-1}, b_i, b_{i+1}, \ldots), b_i \in \{+1, -1\}$ into a new sequence $\boldsymbol{a}$ according to

$$a_i = (b_i + b_{i-1} + \ldots + b_{i-L+1})/L \tag{2}$$

is considered here, where L is the depth of correlation. The sequence $\boldsymbol{a}$ is then modulated by a multi-T CPFSK modulator to generate the corresponding CCE/2-T signal. Similar to CTC/2-T signals, CCE/2-T signals use a constant modulation index at $(h_1 + h_2)/2$, and change the symbol duration to $\lambda_i T$ if the corresponding modulation index of the 2-h scheme during that interval is h_i. Effectively, CCE/2-T signals change the symbol duration after every n symbols cyclically between $\lambda_1 T$ and $\lambda_2 T$. Clearly, the case $n = 1$ corresponds to regular 2-h signalling.

Table 2 PROPERTIES OF COMBINED CORRELATIVELY ENCODED/MULTI-T SIGNALS

L	h_1	h_2	n	d^2_{min}	d^2_{min} of CTC/2-h	Gain
						dB
2	0·5	0·4	2	2·100	2·264	−0·327
2	7/9	13/18	2	4·638	4·607	+0·030
2	0·75	9/16	2	3·607	3·909	−0·349
3	0·5	0·7	1	2·368	2·385	−0·031
3	0·5	0·8	1	2·732	2·763	−0·049
3	19/24	5/4	1	4·980	5·004	−0·021
3	0·5	0·85	1	2·886	2·846	+0·061
3	0·5	0·35	1	1·254	1·277	−0·079
3	19/24	7/6	3	5·385	5·125	+0·215
3	3/4	7/8	3	3·968	3·931	+0·041
3	3/4	13/16	3	3·702	3·692	+0·012

Table 2 lists the minimum distance of CCE/2-T signals along with those of the corresponding CCE/2-h signals discussed in Reference 3. As with trellis coding, it is seen that CCE/M-T signals can achieve distances greater than, and are always close to, those of the corresponding CCE/M-h signals.

Conclusions: A multi-T realisation of combined trellis coded/multi-h and combined correlatively encoded/multi-h signals has been considered. Numerical results indicate that the multi-T realisation can achieve a minimum distance higher than the corresponding multi-h realisation.

20th May 1993

J. P. Fonseka (*School of Engineering and Computer Science, The University of Texas at Dallas, EC 33, 2601, N. Floyd Rd., Richardson, TX 75080, USA*)

References

1 ANDERSON, J. B., AULIN, T., and SUNDBERG, C. E.: 'Digital phase modulation' (Plenum Press, 1986)
2 FONSEKA, J. P., and DAVIS, G. R.: 'Combined coded/multi-h CPFSK signaling', *IEEE Trans.*, 1991, **COM-38**, pp. 1708–1715
3 FONSEKA, J. P., and DAVIS, G. R.: 'Combined correlatively encoded/multi-h CPFSK signaling'. GLOBECOM '91, pp. 23.7.1–23.7.5
4 HOLUBOWICZ, W., and SZULAKIEWICA, P.: 'Multi-T realisation of multi-h phase codes', *IEEE Trans.*, 1985, **IT-31**, pp. 528–529
5 HOLUBOWICZ, W., and CASSARA, F.: 'Simulation study of Multi-T phase codes', *IEEE Trans.*, 1990, **COM-38**, pp. 1664–1666

NOVEL METHOD FOR DESIGNING DIGITAL ALLPASS FILTERS BASED ON EIGENVALUE PROBLEM

X. Zhang and H. Iwakura

Indexing terms: Digital filters, Allpass filters

A novel method is presented for designing digital allpass filters with equiripple phase responses based on the eigenvalue problem, and shows that the optimal filter coefficients can be easily obtained by computing the maximum eigenvector and applying an iteration procedure.

Introduction: In many applications of digital signal processing, digital allpass filters are often used as phase equalisers. Several design procedures for digital allpass filters have been proposed based on the minimum p-error criterion approximation [1], a generalised exchange method [2], and linear programming algorithms [3, 4]. However, the disadvantages of these approaches include the need for initial solutions and heavy computational burden.

The purpose of this Letter is to develop a new method for designing digital allpass filters with equiripple phase responses. The design procedure is based on the formulation of an eigenvalue problem by using the Remez exchange algorithm. The optimal filter coefficients are obtained by computing an eigenvector corresponding to the maximum eigenvalue and applying an iteration procedure.

Transfer function of digital allpass filters: The transfer function of an Nth-order digital allpass filter $A_N(z)$ is defined as follows:

$$A_N(z) = z^{-N} \frac{\sum_{n=0}^{N} a_n z^n}{\sum_{n=0}^{N} a_n z^{-n}} \tag{1}$$

where the filter coefficients a_n are real, and $a_0 = 1$. The phase of a stable allpass filter is 0 when $\omega = 0$, $-N\pi$ when $\omega = \pi$, and is required to decrease monotonically with increasing frequency.

Assuming that $\theta(\omega)$ and $\theta_d(\omega)$ are the phase response of $A_N(z)$ and the desired phase response, respectively, the difference $\theta_e(\omega)$ between $\theta(\omega)$ and $\theta_d(\omega)$ is

$$\exp[j\theta_e(\omega)] = \exp[j(\theta(\omega) - \theta_d(\omega))] = \frac{\sum_{n=0}^{N} a_n \exp\left[j\left(n\omega - \frac{N\omega + \theta_d(\omega)}{2}\right)\right]}{\sum_{n=0}^{N} a_n \exp\left[-j\left(n\omega - \frac{N\omega + \theta_d(\omega)}{2}\right)\right]} \tag{2}$$

and

$$\theta_e(\omega) = 2\tan^{-1}\frac{\sum_{n=0}^{N} a_n \sin\left(n\omega - \frac{N\omega + \theta_d(\omega)}{2}\right)}{\sum_{n=0}^{N} a_n \cos\left(n\omega - \frac{N\omega + \theta_d(\omega)}{2}\right)} = 2\tan^{-1}\Phi(\omega) \tag{3}$$

Therefore, the phase approximation problem of digital allpass filters is to minimise the phase error $\theta_e(\omega)$ of eqn. 3.

Formulation based on eigenvalue problem: To solve the phase Chebyshev approximation problem of digital allpass filters, we use the Remez exchange algorithm and formulate the condition for the phase error $\theta_e(\omega)$ in the form of an eigenvalue problem. By selecting extremal frequencies ω_i ($i = 0, 1, \ldots, N$) in the specified frequency range, we want to find a set of filter coefficients a_n that satisfy the following conditions:

$$\Phi(\omega_i) = \frac{\sum_{n=0}^{N} a_n \sin\left(n\omega_i - \frac{N\omega_i + \theta_d(\omega_i)}{2}\right)}{\sum_{n=0}^{N} a_n \cos\left(n\omega_i - \frac{N\omega_i + \theta_d(\omega_i)}{2}\right)} = \tan\left[(-1)^i \frac{\theta_e}{2}\right] = (-1)^i \delta \tag{4}$$

In general, the phase error θ_e will be very small, so that

$$\theta_e = 2 \tan^{-1} \delta \simeq 2\delta \tag{5}$$

We can rewrite eqn. 4 in matrix form as

$$\boldsymbol{PA} = \delta \boldsymbol{QA} \tag{6}$$

where

$$\boldsymbol{A} = [a_0, a_1, \ldots, a_N]^T \tag{7}$$

The elements of matrix $\boldsymbol{P}, \boldsymbol{Q}$ are given by

$$\begin{cases} P_{ij} = \sin \Theta_{ij} \\ Q_{ij} = (-1)^i \cos \Theta_{ij} \end{cases} \tag{8}$$

where

$$\Theta_{in} = \left(j - \frac{N}{2}\right)\omega_i - \frac{\theta_d(\omega_i)}{2} \tag{9}$$

Supposing that $\boldsymbol{P}$ is a singular matrix, there must exist an $\boldsymbol{A}$ ($\boldsymbol{A} \neq \mathbf{o}$, $\mathbf{o} = [0, 0, \ldots]^T$) that satisfies $\boldsymbol{PA} = \mathbf{o}$; we can then obtain $\delta = 0$ or $\boldsymbol{QA} = \mathbf{o}$ from eqn. 6. Assuming that $\boldsymbol{QA} = \mathbf{o}$, the error δ is of no significance in eqn. 6, hence we can obtain $\delta = 0$. In other words, we can obtain the desired phase response by the filter coefficients a_n; however, this is generally impossible in the practical design problem. Therefore, we conclude that $\boldsymbol{P}$ is a nonsingular matrix, and eqn. 6 can be rewritten as

$$\frac{1}{\delta}\boldsymbol{A} = \boldsymbol{P}^{-1}\boldsymbol{QA} \tag{10}$$

It should be noted that eqn. 10 corresponds to an eigenvalue problem, i.e. $1/\delta$ is an eigenvalue of matrix $\boldsymbol{P}^{-1}\boldsymbol{Q}$, and $\boldsymbol{A}$ is an eigenvector. Hence, by solving the eigenvalue problem of eqn. 10, a set of solutions a_n can be obtained as an eigenvector. In the design problem, the aim is to minimise the phase error δ. Therefore, to minimise the error δ, we must maximise the eigenvalue $1/\delta$, and then the corresponding maximum eigenvector gives a set of filter coefficients a_n. To obtain an equiripple phase response, we find the peak frequencies of $\theta_e(\omega)$ and substitute them into the extremal frequencies, then solve the eigenvalue problem of eqn. 10. The above-mentioned process is repeated over and over again until the equiripple response is obtained. The design algorithm is shown as follows:

Design algorithm:

procedure { design algorithm of digital allpass filters }
begin
1 Read N, and the desired phase response $\theta_d(\omega)$.
2 Select initial extremal frequencies Ω_i (for $i = 0, 1, \ldots, N$) equally spaced in the specified frequency range.
repeat
3 Set $\omega_i = \Omega_i$ (for $i = 0, 1, \ldots, N$).
4 Compute $\boldsymbol{P}$ and $\boldsymbol{Q}$ by using eqn. 8, and find the maximum eigenvector of $\boldsymbol{P}^{-1}\boldsymbol{Q}$ to obtain the filter coefficients a_n.
5 Search the peak frequencies of the phase error $\theta_e(\omega)$, and store them in the corresponding Ω_i.
until the following conditions are satisfied for a prescribed small constant ε:

$$|\Omega_i - \omega_i| \leq \varepsilon \qquad (\text{for } i = 0, 1, \ldots, N)$$

end.

Convergence of design algorithm: The phase design problem of digital allpass filters is nonlinear. Here, we discuss convergence of the preceding design algorithm. In the proposed design algorithm, the key point to guarantee its convergence is that the number of obtained peak frequencies is at least equal to that of the extremal frequencies at each iteration [2]. We rewrite eqn. 3 as

$$\Phi(\omega) = \frac{N(\omega)}{D(\omega)} \tag{11}$$

where $N(\omega)$ and $D(\omega)$ are linear polynomials. Linear polynomials always change their signs through 0. We can see from eqn. 11 that the sign change of $\Phi(\omega)$ is caused by the sign change of $N(\omega)$ or $D(\omega)$. When $N(\omega)$ or $D(\omega)$ changes its sign, $\Phi(\omega)$ crosses 0 or ∞ to change its sign. From eqn. 4, we have $\Phi(\omega_i) = -\Phi(\omega_{i+1})$ and $|\Phi(\omega_i)| = \delta$. Hence, depending on the sign change of $\Phi(\omega)$ through 0 or ∞, there must exist more than one solution for the interpolation problem of eqn. 4. In the design problem, the aim is to minimise the maximum phase error; then, we want to obtain the phase characteristic where $\Phi(\omega)$ crosses 0 to change its sign in the specified frequency range. If $\Phi(\omega)$ crosses 0 to change its sign in the specified frequency range, there must exist a peak frequency in the neighbourhood of each extremal frequency. Hence, sufficient peak frequencies whose number is not fewer than that of the extremal frequencies can be obtained, and convergence of the design algorithm be guaranteed. When $\Phi(\omega)$ crosses 0 to change its sign in the specified frequency range, the resulting error δ will become smallest. Therefore, to force $\Phi(\omega)$ to change its sign through 0, we must minimise the error δ with the aid of computation of the maximum eigenvalue, then convergence of the design algorithm can be guaranteed.

Design example: The aim is to design a digital allpass filter with the desired phase response as follows:

$$\theta_d(\omega) = -30\omega - 0{\cdot}5\pi \qquad (0{\cdot}04\pi \leq \omega \leq 0{\cdot}94\pi)$$

A 31st-order digital allpass filter is designed by using the proposed design algorithm. The phase response is shown in Fig. 1. We can see from Fig. 1 that the phase response is equiripple.

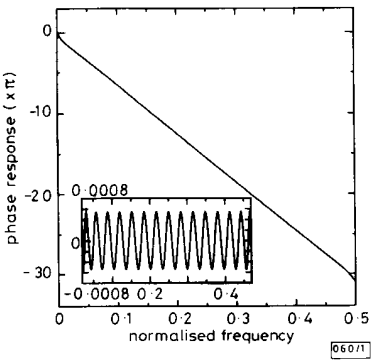

Fig. 1 *Phase response of digital allpass filter*

Conclusion: A new design method for digital allpass filters with equiripple phase responses has been proposed based on the eigenvalue problem. The design problem has been reduced to the computation of the maximum eigenvector, hence the design algorithm is guaranteed to be convergent and the optimal solution can be easily obtained.

17th May 1993

X. Zhang and H. Iwakura (*Department of Communication and System Engineering, University of Electro-Communications, 1-5-1 Chofugaoka, Chofu-shi, Tokyo 182, Japan*)

References

1 DECZKY, A. G.: 'Synthesis of recursive digital filters using the minimum p-error criterion', *IEEE Trans.*, 1972, **AU-20**, pp. 257–263
2 DECZKY, A. G.: 'Equiripple and minimax (Chebyshev) approximations for recursive digital filters', *IEEE Trans.*, 1974, **ASSP-22**, pp. 98–111

3 SUGAHARA, K.: 'Linear programming design of IIR digital phase network', *Trans. IEICE Japan*, 1985, **J68-A**, pp. 444–450 (in Japanese)
4 JING, Z.: 'A new method for digital all-pass filter design', *IEEE Trans.*, 1987, **ASSP-35**, pp. 1557–1564

ALGORITHMIC MEASURES FOR PREVENTING MIDDLEPERSON ATTACK IN IDENTIFICATION SCHEMES

C. H. Lim and P. J. Lee

Indexing terms: Information theory, Cryptography, Codes

Several algorithmic methods are presented for preventing the middleperson attack in identification schemes based on the zero-knowledge technique.

Introduction: Based on the zero-knowledge technique, many efficient identification schemes have been proposed [1–3], but it is also well known that these schemes are vulnerable to the middleperson attack (also known as mafia fraud) [4, 5]. In this attack a verifier can always masquerade as a genuine prover just by relaying all messages between the prover and the victim to whom he is trying to falsely prove himself as the prover. An active attacker may also perform the middleperson attack by manipulating the identification interaction between the prover he is trying to impersonate and a verifier. This is a serious problem especially when an identification scheme is used for granting access to crucial resources in a computer network, because an attacker can victimise any authorised users. Therefore these schemes cannot be used without proper countermeasures, especially in network environments. Some countermeasures against the middleperson attack have been proposed [5, 6], but they are not algorithmic and not suitable for network environments. We present simple and efficient countermeasures suitable for network environments. The following system setup and notation will be used throughout this letter.

System parameters and notation: A trusted authority publishes two primes p and q, a base element $g \in Z_p^*$, a one-way hash function h and a security parameter t such that $q|p-1$, $p \geq 2^{512}$, $q \geq 2^{140}$, $g^q \equiv 1 \bmod p$ $(g \neq 1)$, $h: \{0,1\}^* \rightarrow [0, 2^t)$, and $t \geq 20$. Each user i possesses a secret key $S_i \in_R Z_q^*$ and the corresponding public key $P_i \equiv g^{-S_i} \bmod p$, where $x \in_R Z_q^*$ means that x is chosen at random over Z_q^*. Any logarithm R of $X \equiv g^R \bmod p$ should be interpreted to be chosen at random over Z_q^*. For each verified user i, the trusted authority prepares the identity information ID_i of user i, computes the public key certificate C_i as a digital signature on the pair (ID_i, P_i) and gives them to user i. The symbols $\oplus$ and $\|$ are used to denote mod-2 addition and concatenation, respectively.

Schnorr identification scheme: For illustration purposes; the following Schnorr identification scheme will be used throughout this Letter. We denote the prover as user i and the verifier as user j:

(1) user i first sends user j an initial witness $X_i \equiv g^{R_i} \bmod p$ $(R_i \in_R Z_q^*)$ together with ID_i, P_i and C_i

(2) user j then replies by sending a random challenge $E \in_R [0, 2^t)$

(3) user i reponds with $Y_i \equiv R_i + S_i E \bmod q$

(4) finally user j, after verifying C_i, computes a final witness $X_i \equiv g^{Y_i} P_i^E \bmod p$ and checks that it is equal to the initial witness X_i received in step (1).

Middleperson attack: In this Letter, by the middleperson attack we mean the simplified version of original mafia fraud, where only three people are involved: a prover i, a verifier j (attacker) and a third party k. When i initiates an identification protocol with j, j also initiates the same protocol with the third party k to whom he wants to falsely prove himself as i. Now j, positioned in the middle between i and k, relays an initial witness from i to k and a challenge from k to i, and so on. Therefore the protocol is actually carried out between i and k and thus j can successfully cheat k. This real-time attack for an identification protocol is always possible in computer communication environments. In the following, we present three countermeasures for preventing the middleperson attack.

Countermeasure 1: One obvious way for preventing the middleperson attack will be that a prover ties together, through a simple one-way function, identity information of the verifier (to whom he wants to prove) and an initial witness so that the verifier's identity information cannot be separated from the initial witness. This binds the prover and the verifier together, and thus transference of the initial witness to any third party will be of no effect.

(1) User i sends user $j X_i \equiv (T_i \oplus ID'_j) \bmod q$ along with ID_i, P_i, C_i and ID'_j, where $T_i \equiv g^{R_i} \bmod p$ and ID'_j may be either ID_j itself or an approximation to ID_j (for example, the name and/or address of his counterpart).

(2) User j checks ID'_j and replies by sending a random challenge $E \in_R [0, 2^t)$.

(3) User i sends user $j Y_i \equiv R_i + S_i E \bmod q$ as a response.

(4) After verifying C_i, user j checks that $X_i \equiv (T_i \oplus ID'_j) \bmod q$, where $T_i \equiv g^{Y_i} P_j^E \bmod p$.

We simply used a reduction mod q for binding the verifier's identity information and the prover's initial witness in step (1). This is enough for our purpose because the probability is negligible for user j (would-be attacker) to extract the exact T_i from X_i before receiving the response Y_i in step (3). A one-way hash function h can also be used for his purpose, in which case message authentication capability can be easily incorporated into the protocol. That is, to send a message M to user j, user i computes $X_i = h(T_i \| M)$ with $T_i \equiv g^{R_i} \bmod p$ and sends X_i and M to user j in step (1) together with other information. Then in step (4) user j can check that $X_i = h(T_i \| M)$ with $T_i \equiv g^{Y_i} P_i^E \bmod p$. A successful check will imply that the message is authentic and originated from user i. Message M should contain the identity information of the destined receiver so that the receiver cannot assert himself as the origin of the message to another user.

Countermeasure 2: Another method for preventing the middleperson attack is to compute a challenge value as a function of the verifier's identity information and the random number chosen by the verifier, which also makes the protocol valid only between the prover and the verifier.

(1) User i sends user $j X_i \equiv (g^{R_i} \bmod p) \bmod q$ together with ID_i, P_i and C_i.

(2) User j sends user $i R_j \in_R Z_q^*$ together with ID_j. He then computes $E_j = h((X_i \| R_j)^2 \bmod p \| ID_j) \in [0, 2^t)$.

(3) User i checks that the received ID_j corresponds to the identity information of his intended counterpart user j. If it is then he sends $Y_i \equiv R_i + S_i E_i \bmod q$ to user j, where $E_i = h((X_i \| R_j)^2 \bmod p \| ID_j)$.

(4) After verifying C_i, user j checks that $X_i \equiv (g^{Y_i} P_i^{E_j} \bmod p) \bmod q$.

Note that the middleperson attack is successful only when user j (attacker) can find in real time a number R_j for a given E_k such that $E_k = h((X_i \| R_j)^2 \bmod p \| ID_j)$. This is because in the middleperson attack scenario user j will relay the initial witness X_i from user i to another user k and then the challenge value E_k will be determined by the random number sent by user k and finally user j should receive from user i the response Y_i computed with this E_k in order to cheat user k successfully. The attacker's complexity is $\sim 2^t$ multiplications mod p and hashing operations which must be carried out in