Introduction: The direct sequence spread spectrum communication system has many attractive properties compared with other communication techniques. The most well known properties are its anti-jamming capability, multipath rejection, low probability of intercept, etc. [1]. In those systems the despreading of random code is an important issue [1] Among many despreading algorithms, the use of a matched filter is supposed to be a fast way to acquire the random code [2]. However, the major disadvantage of conventional digital matched filters is that as the number of stages increases the amount of multiplication and accumulation (M and A) will be greatly increased. This constrains the chip rate and the hardware implementations. The number of stages of commercial digital matched filters is mostly in the range 8–64, and the chip rate is limited to <30 M chip/s [3].

We propose a digital differential matched filter (DDMF), which employs novel schemes to reduce the amount of M and A. For those matched filters with long stages, this new architecture saves half the number of M and A in comparison with the conventional filter, while maintaining an identical processing gain (PG). By cutting down the number of M and A, not only the hardware implementation but also the power consumption is reduced. This makes digital matched filters more suitable for low power personal communication system (PCS) implementation.

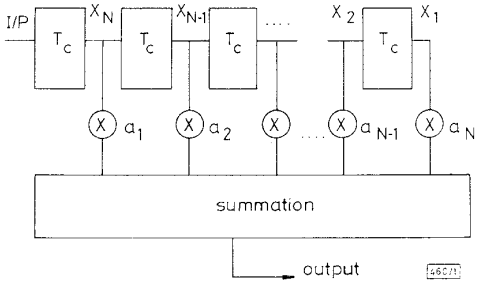

Fig. 1 *Conventional matched filter structure*

Architecture: A block diagram of the conventional digital matched filter (CDMF) with N stages is shown in Fig. 1 [1], where a block with Tc is a tap with one chip time delay. The output of the matched filter at time n–1 and n can thus be expressed as follows:

$$f_{C,n-1} = a_N x_0 + a_{N-1}x_1 + \ldots + a_2 x_{N-2} + a_1 x_{N-1}$$
$$f_{C,n} = a_N x_1 + a_{N-1}x_2 + \ldots + a_2 x_{N-1} + a_1 x_N \quad (1)$$

where N is the length of the PN sequence, $a_{i,i} = 1{\sim}N$ are the PN sequence coefficients, and $x_{i,i} = 0{\sim}N$ is the received digital signal. By subtracting two consecutive outputs of this matched filter, a new sequence $f_{D,i} = f_{C,i} - f_{C,i-1}$, $i = 1, 2, \ldots$ can be generated, where

$$f_{D,n} = b_{N+1}x_0 + b_N x_1 + b_{N-1}x_2 + \ldots + b_2 x_{N-1} + b_1 x_N$$
$$= -a_N x_0 + (a_N - a_{N-1})x_1 + \ldots + (a_2 - a_1)x_{N-1} + a_1 x_n \quad (2)$$

Apparently, by accumulating the $f_{D,i}$, $i = 1, 2, \ldots, n$, $f_{C,n}$ can be obtained. Therefore, a DDMF structure, as shown in Fig. 2, can be constructed. Since a_i, $i = 1, 2, \ldots, N$ are either 1 or –1, the b_i, $i = 1, 2, \ldots, N$ in DDMF are 2, –2, or 0 except b_1 and b_{N+1}. For a coefficient of zero, there is no need for multiplication. Thus the number of multiplication is reduced. Table 1 is a comparison of the number of multiplications in CDMF and in DDMF. In this comparison, a well known maximum length random code (m-sequence) [1] is used.

Table 1: Comparison of M&A number between conventional and proposed matched filter

	CDMF	DDMF	Ratio of M&A number between CDMF and DDMF
Number of M&A	$2^r - 1$	$2^{r-1} + 1$	$2^{r-1} + 1 / 2^r - 1 \simeq 0.5$

r: order of generator polynomial of m-sequence [1]

It is obvious from the comparison that the number of M and A is reduced by one half. For a system with M sample/chip, the DDMF furthermore reduces the number of M and A to $1/2M$ times. Thus the power and the hardware are reduced accordingly in a real implementation, while the original property of CDMF, such as processing gain, is still retained.

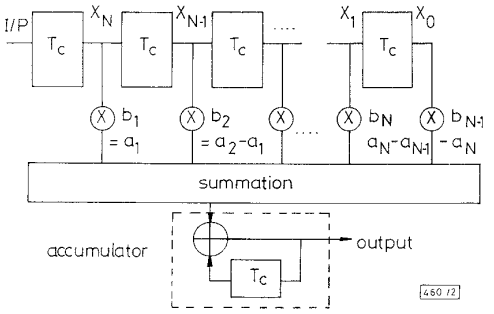

Fig. 2 *Proposed differential matched filter structure with half the coefficients equal to zero*

Conclusion: A differential digital matched filter for DSSS is proposed. This filter reduces the number of M and A by half for one sample/chip system, and to $1/2M$ for M sample/chip system, while maintaining all the properties of the conventional matched filter. These advantages will be more apparent for matched filters with long stages, thus leading to a more suitable implementation of low power VLSI in long operation time handset application.

30 May 1996
Electronics Letters Online No: 19961072

W.-C. Lin (*Department of Communication Engineering, National Chiao Tung University, Hsinchu, Taiwan, Republic of China*)

K.-C. Liu and C.-K. Wang (*Department of Communication Engineering, Central Tung University, Chung-Li, Taiwan, Republic of China*)

References

1 SIMON, M.K., OMURA, J.K., SCHOLTZ, R.A., and LEVITT, B.K.: 'Spread spectrum communications' (Computer Science Press, Rockville, Maryland, 1985)
2 SU, Y.T.: 'Rapid code acquisition algorithms employing matched filters', *IEEE Trans.*, 1988, **COM-36**, (6)
3 POVEY, G.J.R., and GRANT, P.M.: 'Simplified matched filter receiver design for spread spectrum communication applications', *Electron. Commun. Eng. J.*, 1993, pp. 59–64

Improving the performance of cell-loss recovery in ATM networks

Hyo Taek Lim, DaeHun Nyang and JooSeok Song

Indexing terms: Forward error correction, Asynchronous transfer mode

A new method for improving the performance of cell-loss recovery using FEC (forward error correction) in ATM networks is proposed. This method provides more correcting coverage than existing methods.

Introduction: The major source of errors in high-speed networks such as B-ISDN is buffer overflow during congested conditions which results in cell loss. Conventional cell loss recovery methods using FEC in ATM networks recover up to 16 consecutive cell losses because lost cells are recognised from a 4 bit sequence number (SN). A cell loss recovery method which can recover up to 18 consecutive cell losses in ATM networks is presented in [1]. This means that the method cannot extend the row length of the coding matrix to more than 18. We review the method briefly and then employ a new algorithm for recovering more cell losses. The performance estimation is based on the two-state Markov model

called the Gilbert channel. It is shown that the method using the new algorithm reduces the cell loss rate more than that of the previous method.

Coding algorithm of [1]: The FEC decoder at the receiving side buffers the cells from the network. The decoder finds lost cells by observing the 4 bit SN and 2 bit segment type (ST) values in the stream of received cells, and recovers the lost cells using parity cells. The SN and ST values in a cell are used to consider a new cell SN as shown in Table 1. The encoding tasks at the sending side are the generation of the parity cell and transfer of the coding matrix (M*N) to the network without modifying any value of the field. There is no processing delay in encoding because of the multiplicative and additive natures of the encoding operation [2].

Table 1: SN and ST values for new SN

SN	ST	new SN
0	BOM	0
1	COM	1
0	COM	16
1	EOM	17

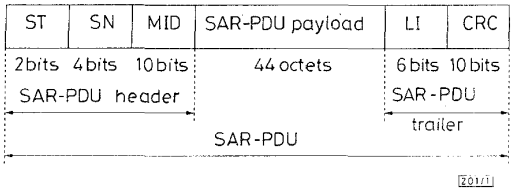

Fig. 1 *SAR-PDU format for AAL 3/4*

CRC: cyclic redundancy check
LI: length indicator
MID: multiplexing identifier
PDU: protocol data unit
SAR: segmentation and reassembly
SN: sequence number
ST: segment type

Table 2: New cell SN(SN*)

SN	ST	LI	SN*	LI value modified? (y/n)
0	BOM	44	0	no
1–15	COM	44	0–15	no
0	COM	44	16	no
X	COM	0–43	17–60	yes
X	COM	45–63	61–79	yes
X	EOM	X	80 (last cell no.)	no

(X: don't care)

New cell-loss recovery method: Here we describe a new coding algorithm which focuses on AAL (ATM adaptation layer) 3/4 sublayer. Fig. 1 depicts the SAR-PDU for AAL 3/4, which is used for service classes C and D [3]. The 6 bit length indicator (LI) specifies the number of octets from the convergence sublayer PDU (CS-PDU) which are included in the SAR-PDU payload field (with a maximum of 44 octets). Note that when the ST field of a cell indicates COM (continuation of message), the LI value of the cell indicates 44, which is the maximum value of the SAR-PDU payload field, because the cell is at the middle of the message. Also, the LI value of the cell whose ST field indicates BOM (beginning of message) is always 44. That is, the LI field of the cell whose ST field indicates COM or BOM is meaningless. We use the LI field together with SN and ST fields to consider a new cell SN(SN*) up to the maximum of 80 (from 0) as shown in Table 2. The encoding tasks that the FEC encoder carries out are the same as in [1], except for modifying the LI value to consider a new SN*. The decoding procedure is as follows:

(i) buffer the cells from the networks
(ii) find the lost cells by observing the SN, ST and LI values in the received cells as in Table 2
(iii) recover the lost cells using parity cells
(iv) adjust the modifed LI values of the cell whose ST value is COM to 44.

Example: When a message is less than or equal to 18 cells, all random permutations of lost cells at the receiver can be identified and recovered by the SN and ST without LI. We consider the case when a message size consists of 34 cells. The sequence of 34-cell message at the sending side is shown below.

(B:BOM, C:COM, E:EOM, PC:ParityCell, —:message-size specific)

SN :	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	0
ST :	B	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C
LI :	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44
PC :	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
		*															

SN :	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	0	1
ST :	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	E
LI :	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	—
PC :	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
	*																*

Consider the case when the cells for which the SN value is 1(*) are lost. We cannot identify which of either the second or 18th cell is lost but can identify the last cell by SN and ST. Then, the decoder uses the LI field to determine which cell (the second or 18th) is lost.

Thus, the row length of the coding matrix can be extended to N = 81. This means that the proposed method substantially improves the recovery rate of lost cells.

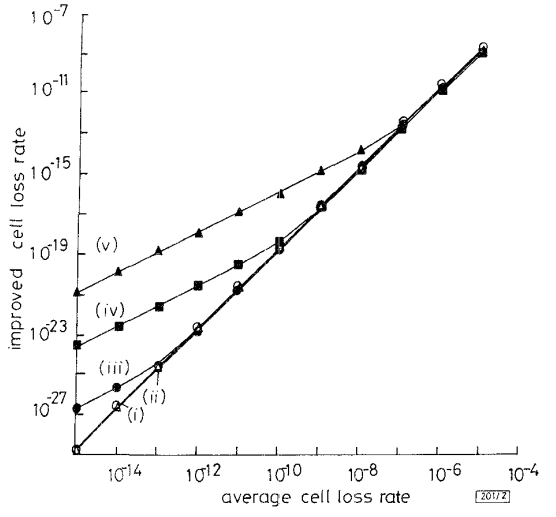

Fig. 2 *Improved cell loss rate*

α: parameter of Gilbert model
Matrix size: M = 30, N = 40
(i) α = 0.3
(ii) α = 0.4
(iii) α = 0.5
(iv) α = 0.6
(v) α = 0.7

Analysis and results: The performance analysis was based on the cell discard process method, which is the two-state Markov model as shown in [1]. The average cell loss rate produced by the model is

$$P_{loss} = \frac{1-\beta}{1-\alpha+1-\beta}$$

The probability that only one cell is lost in a column, denoted by L_1, is given as [1]

$$L_1 = P_{loss}\bar{\alpha}_N\beta_N^{M-2} + (1-P_{loss})\bar{\beta}_N\bar{\alpha}_N\beta^{M-3} * (M-2) + (1-P_{loss})\bar{\beta}_N\beta_N^{M-2}$$

where M, N denote the number of rows in the cell matrix and the row length of the cell matrix, respectively, and α, β the transition probability of the state. Thus, the improved cell loss rate using the proposed cell loss recovery method, denoted by P, is

$$P = P_{loss} - L_1/M$$

Fig. 2 shows the improved cell loss rate with the proposed method when the coding matrix (M = 30, N = 40) is employed.

29 April 1996
Electronics Letters Online No: 19961029

Hyo Taek Lim (*Department of Computer Science and Engineering, Dongseo University, Pusan, 617-716, Korea*)

DaeHun Nyang and JooSeok Song (*Department of Computer Science, Yonsei University, Seoul, 120-749, Korea*)

References

1 LIM, , and SONG, : 'Cell loss recovery method in B-ISDN/ATM networks', *Electron. Lett.*, 1995, **31**, (11), pp. 849–851

2 AYANOGLU, E., GITLIN, R.D., and OGUZ, N.C.: 'Performance improvement in broadband networks using forward error correction for lost packet recovery', *J. High Speed Networks 2*, 1993, pp. 287–303

3 ITU-T: 'Recommendation 1.363, B-ISDN ATM adaptation layer (AAL) specification'. 1993

Mandarin speech recognition using segment-based cepstral comparison in noisy conditions

Shin-Lun Tung and Yau-Tarng Juang

Indexing terms: Speech recognition, Cepstral analysis

A new scheme is proposed that compensates for the effects of noise in speech recognition systems. The new scheme was applied to Mandarin speech recognition. Another scheme, based on interpolation of the compensation vectors of several environments for a particular environment that is not obtained during the training phase, called interpolated SSDCN (ISSDCN), is also presented. Experimental results show that the scheme performs well under different SNR conditions.

Introduction: In recent years, speech recognition systeins that are robust with respect to adverse environments have attracted an increasing amount of interest. A variety of approaches have been considered in the development of noisy-speech recognition systems including techniques based on spectral subtraction [1], the use of a comb filter [2], a family of distortion measures [3], and the use of cepstral normalisation [4, 5], etc.. Among these many approaches, a series of normalisation algorithms have been developed that reduce the effects of environmental variations on recognition accuracy [4, 5]. The normalisation algorithms based on cepstral comparison assume that differences between the training and testing environments can be characterised by an additive correction to the cepstral vectors that represent the speech.

In this Letter, we applied the cepstral normalisation algorithm to Mandarin speech and called the new scheme segment-based SNR-dependent cepstral normalisation (SSDCN). However, the testing environment does not closely resemble any single environment in the training set in some conditions. In that case, the interpolation of the compensation vectors of several environments may be more useful. Based on the above, an interpolated SSDCN (ISSDCN) algorithm is also proposed in this Letter.

Pitch-based Mandarin speech recognition: The scheme focuses on the pitch-based segmental model for Mandarin speech recognition. The main process is to obtain the representative frames that are important and necessary for speech recognition. First, we designed a pulse-based pitch detector to extract the pitch period. The detector predicts the location of successive pitch pulses based on the position and amplitude of the preceding pitch pulse. The used algorithm attempts to reduce the number of computations and promotes the accuracy of pitch detection. The discrimination criterion is given by

$$r_a * Ampf + r_p * PPf > \rho \qquad (1)$$

where r_a and r_p are the similarity ratios of amplitude and position, respectively, *Ampf* is the pitch amplitude, *PPf* is the pitch period, and ρ is a threshold value. Then according to the acoustic phonetics and pitch information of Mandarin speech, the isolated Mandarin syllable is divided into three segments: consonant-segment, vowel-segment, and residual-segment. In every segment, only one representative frame was selected for speech recognition. The principle of selecting the representative frames is as follows:

(i) In a consonant-segment, based on the experimental observation, the first representative frame is decided by selecting M sample points of the whole consonant part before the first pitch peak.

(ii) In a vowel-segment, we select N representative pitch peaks as the second frame in place of the whole section.

(iii) In a residual-segment, to maintain the residual part of speech signal, the final pitch peak to the end of the speech is chosen to be the third frame.

The feature vectors of the LPC cepstrum are then obtained from the three frames.

Segment-based SNR-dependent cepsiral normalisation: In this Section, we describe the proposed algorithms, related to as SNR-dependent normalisation procedures, which compensate for environmental variation based on the different SNR and segment compensation vectors.

SSDCN: A segment-based SNR-dependent cepstral normalisation algorithm applies additive correction in the cepstral domain that depends on the instantaneous SNR of the segmental frame for pitch-based Mandarin speech recognition. When a Mandarin syllable from some unknown environment is input to the recognition system, the system first determines which of the testing environments in the training data is most similar to the current testing environment. The compensation vectors from the chosen testing environment are applied to normalise the utterance according to the expression

$$\hat{x}_{sf} = z_{sf} + r_{sf}[SNR] \qquad (2)$$

where *sf* is the segmental frame of the syllable index, *SNR* is the signal to noise ratio of environment, $\hat{x}$, z, and r are the compensated cepstral, original cepstral and compensation vectors, respectively. The compensation vectors in the SSDCN are described as follows

$$r_{sf}[SNR] = \frac{\sum_{i=1}^{SET} (x_{sf}^i - z_{sf}^i)\delta(s_{sf} - SNR)}{\sum_{i=1}^{SET} \delta(s_{sf} - SNR)} \qquad (3)$$

where SET is the training set number, s_{sf} is the SNR value for the segmental frame of the syllable s, and x_{sf}^i, z_{sf}^i are the cepstral vectors of training and testing syllables, respectively.

ISSDCN: In cases where the testing environment does not resemble the training environments used to develop the compensation vectors for SSDCN, interpolation of the compensation vectors of several environments can be more beneficial than using a single compensation vector. The interpolated compensation vectors are obtained by interpolating several of the closer compensation vectors:

$$\hat{r}_{sf}[SNR] = \sum_{n=1}^{E} w_n \cdot r_{sf}[SNR_n] \qquad (4)$$

where w_n is the weighting factor for the nth environment, $\hat{r}[SNR]$ is the estimated compensation vector, and $r[SNR_n]$ is the compensation vector for the nth environment. The weighting factors for each closer compensation environment are described as follows:

$$w_n = \frac{SNR}{SNR_n} \frac{\sum_{i=1, i\neq n}^{E} |SNR - SNR_i|}{(E-1)\sum_{i=1}^{E} |SNR - SNR_i|} \qquad (5)$$

In eqn. 5, the first item is the compensation ratio and the second item is the weighting ratio.