

IPOS COPY
9602069-3

Transmission system comprising at least a coder.

The invention relates to a transmission system comprising at least a coder for coding a signal, at least a transmitter for transmitting the coded signal to at least a receiver, at least a decoder for decoding the coded signal and at least an adaptive prediction filter.

- 5 The invention likewise relates to a terminal unit, a coder, a decoder and an adaptive prediction filter.

Such a transmission system can be obtained, for example, without difficulty from the document "Draft of G.728 Fixed Point Specification" Doc. AH. 93-D.3, CCITT Study Group XV, London, March 29-30-1993, and from CCITT
10 Recommendation G.728, "Coding of speech signals at 16 kbit/s using low-delay code excited linear prediction". The two documents will henceforth be referenced [I] and [II]. Such transmission systems can always be used when multiple use is to be made of a given transmission capacity. The reduction of the bit rate of a signal during coding makes it possible, for example, to hold four telephone conversations simultaneously
15 over one 64 kbit/s transmission channel.

The multiple use of a given transmission capacity has very much importance in radio channels, as will be readily understood if one thinks of the rising number of subscribers of mobile radio systems. Also the storage capacity of a memory with an arbitrary storage medium can be used to considerably more advantage when bit
20 rate reducing coders and appropriate decoders are used, because less storage space is necessary for storing an information signal.

It is known to utilize linear prediction for bit rate reduction. In [I] and [II] prediction coefficients are computed by an adaptive prediction filter. The computation is performed segment by segment with the sample values of an auxiliary signal. The
25 auxiliary signal may be an (electric) audio signal or a speech signal, for example, coming from a person. Alternatively, it is possible for the auxiliary signal to be a synthetic audio or speech signal as produced in a coder which operates according to the

This Sheet Has Been Replaced



THIS SHEET HAS BEEN
REPLACED

principle of "analysis by synthesis". A linear correlation between a predicted sampling value (prediction value) of the auxiliary signal and previous sampling values of this signal is realised with the prediction coefficients. The prediction coefficients are determined so that the sum of the squares of all the errors computed for a segment of sampling values assumes a minimum value. An error is here meant to be understood as the difference between a sampling value and its predicted value. More accurate descriptions will be given hereinbelow.

In [I] and [II] an excited signal is converted into a synthetic audio signal by a synthesis filter. This synthetic audio signal is subtracted from an audio signal to be coded and the difference is used for optimizing the selection of the excited signal.

The computation of the prediction coefficients calls for correlation coefficients which are derived from the sampling values of the synthetic audio signal. The computation of the prediction coefficients on the basis of the correlation coefficients calls for much circuitry and cost. To reduce this circuitry and cost the prediction coefficients are computed recursively. To begin with, a first reflection coefficient is computed from the two first correlation coefficients. From the first reflection coefficient the first prediction coefficient is derived. Furthermore, also a prediction error is computed, which is an indication of the quality of the prediction. Then, a next (second) reflection coefficient is determined with the next (second) correlation coefficient, the previously computed prediction coefficients (at this point only the first prediction coefficient is concerned) and with the current prediction error. With this (second) reflection coefficient and the previously computed prediction coefficients the previously computed prediction coefficients are recomputed. Then, a next (second) prediction coefficient and a new prediction error are computed. The recomputation of the previously computed prediction coefficients and the computation of a further prediction coefficient is repeated so many times that all the necessary prediction coefficients are known.

In [II] an implementation in floating point arithmetic is proposed for obtaining sufficient accuracy. The drawback of an implementation in floating point arithmetic is the large circuitry. For this purpose, in [I] there is proposed to implement 16-bit fixed point arithmetic in which results and intermediate results of the computation are retained in the so-called block floating point format. This is meant to be understood





as the representation of a block of numbers in floating point format, with the restriction, however, that all the numbers of the block have the same exponent (with base 2). The common exponent is selected such that all the mantissas are situated within prescribed limits. In [I] there is provided that all the mantissas are situated between -1 and +1. In 5 the block floating point format, for example, the prediction coefficients are shown which, from previous experience, assume values that may be on the order of 10. If, for example, overflow occurs when the prediction coefficients are recomputed *i.e.* a recomputed prediction coefficient will be 1 or exceed 1, the mantissas of all the previously computed prediction coefficients will be divided by 2 and the common 10 exponent (with base 2) will be increased by 1. Then the recomputation is started anew. The change of the block floating point format due to overflow will here be called adaptation of the block floating point format. For the repeated recomputation only the mantissas of the previously computed prediction coefficients are used, and only with 16-bit resolution. The reduction of the resolution from at least 17 bits to 16 bits which is 15 sometimes necessary for the computation can be effected in different manners. The manner of reduction according to [I] will be outlined with the aid of an example. If, for example, the third reflection coefficient has just been computed with a resolution of 17 bits - between the third reflection coefficient and the third prediction coefficient to be derived therefrom there is only a difference in the numerical representation - it is 20 shortened by a rounding to 16 bits, so that the two first prediction coefficients can be recomputed therewith in 16-bit fixed point arithmetic. If there is an overflow during this recomputation, also the rounded third reflection coefficient is, for example, divided by two when recomputation has ended, and the result which has a resolution of 16 bits is stored as a mantissa of the third prediction coefficient. The division is broken off after 25 16 bits (including the sign bit). Due to the binary system this operation corresponds to a shift of the decimal point by one position while the least-significant bit is lost. This procedure causes the inaccuracies - especially with a large number of prediction coefficients - to become unacceptably large.

It is an object of the invention to reduce the computation circuitry and 30 cost in a transmission system of the type defined in the opening paragraph, without reducing the computational accuracy.

A transmission system having this property comprises

THIS SHEET HAS BEEN
REPLACED



- at least a coder for coding a signal,
- at least a transmitter for transmitting the coded signal to at least a receiver,
- at least a decoder for decoding the coded signal and
- at least an adaptive prediction filter provided at least for a coder, the adaptive prediction filter including:
 - 5 - means for computing correlation coefficients from segments of sampling values of an auxiliary signal that depends on the signal or on the decoded signal, as well as
 - means for determining reflection and prediction coefficients from the correlation coefficients in N-bit fixed point arithmetic,
 - 10 and in that the following means are provided for recursively determining the prediction coefficients to be represented in the block floating point format:
 - a) means for computing the k^{th} reflection coefficient,
 - b) means for recomputing the $k-1$ previously computed prediction coefficients while
 - 15 utilizing the k^{th} reflection coefficient obtained under a),
 - c) means for adapting the block floating point format of the $k-1$ prediction coefficients already recomputed and not yet recomputed if overflow occurs during the recomputation of the $k-1$ previously computed prediction coefficients obtained under b), so that N bits are no longer sufficient to represent a recomputed
 - 20 prediction coefficient,
 - d) means for continuing the recomputation of the $k-1$ prediction coefficients for the prediction coefficient at which the overflow as under c) has occurred.
 - e) means for converting the k^{th} prediction coefficient from the k^{th} reflection coefficient computed under a) in a floating point format, whose exponent is
 - 25 represented by the exponent determined under c) of the block floating point format, and
 - f) means (X79) for storing the number obtained under e) as the k^{th} prediction coefficient.

30 If overflow occurs during the recomputation of the prediction coefficients, according to [I] the floating point format of all the $k-1$ previously computed prediction coefficients is adapted and then the recomputation is started again from the beginning. In contrast, the embodiment according to the invention avoids a not inconsiderable part

THIS SHEET HAS BEEN
REPLACED



of the computation circuitry, especially in the event of a large number of coefficients, because the prediction coefficient already recomputed need not be computed once again. The invention is based on the recognition that the prediction coefficients already recomputed, except for the block-floating point format, correspond to the values which
5 would appear if the recomputation were repeated.

The order of the means indicated above and also to be indicated below is not to be understood in such a way that they are also to be activated in this order in time.

A further embodiment of the invention comprises

- 10 2a) means for computing the k^{th} reflection coefficient with a minimum resolution of $N+1$ bits,
- 2b) means for buffering the k^{th} reflection coefficient with a minimum resolution of $N+1$ bits,
- 2c) means for converting the k^{th} reflection coefficient buffered under 2b) to a
15 floating point format defined under e), whose mantissa has a resolution of at least $N+1$ bits,
- 2d) means for rounding to N bits the mantissa obtained under 2c), and
- 2e) means for storing the mantissa rounded under 2d) as a mantissa of the k^{th} prediction coefficient.

The transmission system according to the invention is distinguished from a
20 transmission system described in [I] in that the k^{th} reflection coefficient after its first computation is buffered with a minimum accuracy of $N+1$ bits. For the recomputation of the $k-1$ previously computed prediction coefficients a value is used which distinguishes itself from the buffered value by a limitation to N bits. If overflow occurs during the recomputation of the $k-1$ previously computed prediction coefficients, the
25 buffered value is divided, for example, by 2 after the recomputation has finished, and then rounded to N bits. This way of incorporating the k^{th} prediction coefficient in the (adapted) block floating point format produces an error which is at most as large as the corresponding error in [I]. However, equally important is it to find that the sign of the error according to the invention appears to turn out as often positive as negative, which
30 is different from [I].

These assertions will now be confirmed with the aid of two simple examples.

THIS SHEET HAS BEEN
REPLACED



Let us assume that three ($N = 3$) bits are available for representing the results and at least four bits for intermediate results. Furthermore, let us assume that the adaptation due to overflow requires a division by 4. Numerical examples for k^{th} reflection coefficients having a 4-bits resolution in fixed point representation are started from.

In a first example a k^{th} reflection coefficient has a value of $+5/8$. In the binary system this reflection coefficient is represented as 0.101, the first zero denoting the sign. According to [I] the procedure is as follows:

- a) 0.101 (k^{th} reflection coefficient: $+5/8$ with 4-bits resolution)
- 10 b) 0.11 (value of b) rounded to 3 positions for the recomputation of the $k-1$ previously computed prediction coefficients)
- c) 0.00 (value of b) divided by 4, also with 3-bits resolution. This value is the mantissa of the k^{th} reflection coefficient in the block floating point format with the exponent 2 and with base 2)

15 For making an assertion about the error if the approximation c) is used *in lieu of a*), c) is first to be multiplied again by 4. The difference between the value resulting therefrom and the output value is the error F. In the present case (representation as a fraction) $F = (-5/8) = -5/8$.

In contrast, according to the invention there is provided:

- 20 a) 0.101 (k^{th} reflection coefficient: $+5/8$ with 4-bits resolution),
- b) 0.001 (value of a) divided by 4, also with 4-bits resolution),
- c) 0.01 (value of b) rounded to 3 bits).

Analogous with the preceding observations we now have as an error: $F = (1 - 5/8) = 3/8$.

25 If 0.011 ($+3/8$) is selected as the output value, there is a value of $F = (0 - 3/8) = -3/8$ for the error according to the same scheme of [I]. According to the invention, we now have: $F = (0 - 3/8) = -3/8$.

The invention with its embodiments will be explained with reference to the drawing Figures, in which:

30 Fig. 1 shows a specific transmission system in which the invention can be used,

Fig. 2 shows the basic circuit diagram of an adaptive prediction filter,

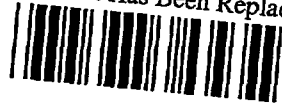


Fig. 3 shows the basic circuit diagram of a speech coder and

Fig. 4 shows diagrams for clarifying a difference between the invention and the state of the art.

Fig. 1 gives a diagrammatic representation of a transmission system
5 comprising a coder 101 and a transmitter 102 on a transmitting side and a receiver 103 and a decoder 104 on a receiving side. The combination of the coder 101 with the transmitter 102 and/or the combination of the receiver 103 with the decoder 104 forms a terminal unit LE.

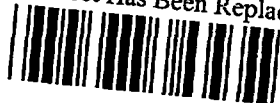
The coder 101 codes a speech signal 10a by means of an adaptive
10 prediction filter 10A. This filter has characteristic features according to the invention, which will become apparent in the following. The speech signal 10a has a bit rate of 64 kbit/s. The coder 101 reduces this bit rate to the bit rate of the coded signal 10b. The bit rate of the coded signal 10b is 16 kbit/s. The signal 10b consists of parameters of the speech signal 10a which parameters are determined segment by segment.

15 The transmitter 102 transmits the coded signal 10b - transcoded into a transmission code, as required - via a transmission channel shown in the drawing to the receiver 103 which transcodes the transmitted signal into a signal 10c and which signal corresponds to the signal 10b except for coding and transmission errors.

The decoder 104 decodes the signal 10c also with the aid of an adaptive
20 prediction filter 10B. The result is a signal 10d. Except for coding, decoding and transmission errors the signal 10d corresponds to the speech signal 10a. The adaptive prediction filters 10A and 10B determine signal-dependent filter coefficients (prediction coefficients) according to the Levinson-Durbin Recursion. The Levinson-Durbin Recursion is an algorithm for determining in a step-by-step fashion the solutions of a
25 linear set of equations of which the coefficient matrix is a Töplitz matrix, thus a symmetrical matrix having the same elements along a given diagonal.

The NCOF prediction coefficients (NCOF stands for the number of these coefficients) for an adaptive prediction filter are unequivocally a consequence of the requirement that the sum of the prediction error squares be minimized on average over
30 a signal section of a given length.

For example, if sampling values $s(k)$ are applied to the input of the filter, the following linear combination will be the output value of the filter at the instant n :



$$y(n) = \sum_{i=1}^{NCOF} a(i) * s(n-i) \quad (1)$$

The value $y(n)$ is considered the predicted sampling value $s(n)$. If the sum of the error
5 squares is formed *i.e.*

$$\sum_{i=0}^{LSEG-1} [y(n+i) + s(n+i)]^2 \quad (2)$$

10 of a segment comprising LSEG sampling values, if (1) is put in (2) while the
coefficients $a(i)$ are minimized in (2), above set of equations with the Töplitz matrix
will arise if the sampling values outside this segment are set to zero. The elements of
the Töplitz matrix are in this case the values of the autocorrelation function of the input
signal at the sampling instants of the input signal, which function will be referenced
15 autocorrelation coefficient $r(\text{minc})$ hereinafter. A detailed representation of the
Levinson-Durbin Recursion is found in: L.R. Rabiner/R.W. Schafer, "Digital Processing
of Speech Signals", Prentice-Hall, 1978.

In the following the algorithm of the Levinson-Durbin Recursion
expressed in Rabiner/Schafer by the equations (8.67) to (8.72), is represented as a
20 pseudo code because this code permits a better detection of the program steps an
arithmetic unit is to carry out for the computation of the prediction coefficients.

LEVINSON-DURBIN Recursion (Pseudo code):

25 Input values: NCOF = Predictor Coefficient
 $r(0) \dots r(NCOF)$ = Autocorrelation coefficients
Output values: $a(0) \dots a(NCOF)$ = Prediction coefficients

$a(0) = 1.0$ O1
30 $a(1) = -r(1) / r(0)$ O2
 $\alpha = r(0) + r(1) * a(1)$ O3
For $\text{minc} = 2, 3, 4, \dots, NCOF$ O4

```

      { sum = 0                                O5
        for ip=0,1,2,3,...,minc-1              O6
          sum = sum + r(minc-ip) * a(ip)        O7
                                                O8
5      rc  = - sum / alpha                      O9
        alpha = alpha + sum * rc                O10
                                                O11
        if(alpha <= 0.0) break                  O12
                                                O13
10     For ip=1,2,3,...,minc/2                  O14
        { R1 = a( ip) + rc * a(minc-ip))       O15
          R2 = a(minc-ip) + rc * a( ip))       O16
          a(ip ) = R1                          O17
          a(minc-ip) = R2                      O18
15
                                                O19
          a(minc) = rc                          O20
                                                O21

```

This Sheet Has Been Replaced



To render the program steps given in the pseudo code even more
 20 transparent, several rows of the algorithm shown above will be further explained.

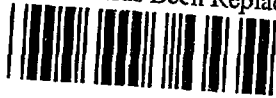
The predictor coefficient carrying the name of NCOF and the NCOF+1
 autocorrelation coefficients are required as input data. The predictor coefficient NCOF
 is identical with the number of filter coefficients $a(i)$ in equation (1). The
 autocorrelation coefficients $r(0)$, $r(1)$ and so on, up to $r(\text{NCOF})$ are determined by other
 25 functional units not disclosed here, which, however, are known to a person skilled in
 the art.

The prediction coefficients $a(0)$, $a(1)$ and so on, up to $a(\text{NCOF})$ are output
 data.

Rows 01 and 02 comprise assignments to the two first prediction
 30 coefficients. In row 03 there is an assignment to an auxiliary variable alpha (prediction
 error for $\text{NCOF}=1$) whose augend is the zeroth autocorrelation coefficient and whose
 addend is the product of the first correlation coefficient and the first prediction

THIS SHEET HAS BEEN
 REPLACED

This Sheet Has Been Replaced



10

coefficient.

If the predictor coefficient NCOF is equal to 1, the computation is over, otherwise a loop having a running variable minc is passed through, which variable assumes all integers from 2 to NCOF. In the loop for the running variable minc all the
5 previously computed prediction coefficients $a(1)$, $a(2)$ and so on, up to $a(\text{minc}-1)$ are changed (recomputed), as is shown in rows 017 and 018. At the end of all the instructions which are to be carried out for the current value k of the running variable minc, a value is assigned to the prediction coefficient having the index minc, as is shown in row 020. This prediction coefficient is to be referenced the k^{th} prediction
10 coefficient; the variables sum and rc (reflection coefficient) which are computed for the first time for the current value k of the running variable minc are to obtain the same addition.

In the loop having the running variable minc two further loops are passed through which are not interleaved and have the running variable ip, the upper value of
15 which variable depends on the current value of the running variable minc, as is shown in rows 06 and 014.

Auxiliary variables are those carrying the names of alpha, sum, rc, R1 and R2. The auxiliary variable alpha is the sum of the error squares according to equation (2), and also has a descriptive meaning; alpha is the energy content of the
20 error signal (prediction error). This content is always to be positive. The computation is therefore broken off if there is a value for alpha smaller than or equal to zero, as is shown in row 012. Also the variable rc has a descriptive meaning; it is the so-termed reflection coefficient which plays a part in the so-called tube model of the vocal tract. A reflection coefficient which occurs as a quotient of the auxiliary variables sum and
25 alpha, as is shown in row 09, belongs to each prediction coefficient.

Most real-time implementations of speech coding methods according to which also coefficients for adaptive filters are to be computed, are based on 16-bit fixed-point arithmetic units, while intermediate results are available partly with enhanced accuracy (for example, 32 bits). The accuracy of the result of the Levinson
30 Recursion decisively depends on the computational accuracy in the recursion. As regards computational accuracy the most sensitive positions in the pseudo code are found where intermediate results of multiplication or division are available with higher

THIS SHEET HAS BEEN
REPLACED



resolution and are then to be reduced to the simple resolution. They are especially the rows 02, 03, 07, 09, 010, 015, 016 and 020.

A module for realising the Levinson-Durbin Recursion in 16-bit fixed-point arithmetic, will be described with reference to the following pseudo code. Several definitions are to be introduced first:

> > Right shift operation;

< < Left shift operation;

10

round (reg) Rounding operation performed on a 32-bit-wide register "reg". Let us assume that the bits in "reg" are numbered 0, 1, 2, ..., 31, while the sign bit is bit 31 and the significance of the individual bits decreases from bit 30 towards bit 0. Rounding with an accuracy of 16 bits then means that bit 15 is added to bit 16 after which the bits 16 to 31 are used as the result.

15

2 Fixed-point multiplication;

20 / Fixed-point division;

Since the division is very costly to carry out, the result of the division is computed accurate to only as many positions as are absolutely necessary. Results of the divisions are flush right in the register containing the results.

25 x_h means the 16 most significant bits of a 32-bits word x .

x_l means the 16 least significant bits of a 32-bits word x .

16-Bits fixed-point Levinson Recursion (Pseudo code):

	EXPa = 0;	X1
30	den = r[0];	X2
	num = r[1];	X3
	if (num < 0) num = -num;	X4

THIS SHEET HAS BEEN
REPLACED



```

a2 = num / den; X5
a2 = a2 << 15; X6
a1 = round(a2); X7
if (r[1] > 0) a1 = -a1; X8
5   rc1 = a1h; X9
    a[1] = a1h; X10
    a0l = 0; X11
    a0h = r[0]; X12
    a0 = a0 + rc1 * 2 * r[1]; X13
10  a0 = round(a0); X14
    alpha = a0h; X15
    for (minc=2; minc <= NCOF; minc++) X16
    { a0 = 0; X17
      for (ip=1; ip < minc; ip++) X18
15    a0 = a0 + r[minc-ip] * 2 * a[ip] X19
                                         X20
    a0 = a0 << EXPa; X21
    a1l = 0; X22
    a1 = r[minc]; X23
20  a0 = a0 + a1; X24
    a0 = round(a0); X25
    sign = a0h; X26
    num = a0h; X27
    if (num < 0) num = -num; X28
25  if (num >= alpha) X29
        goto LEDUMO_failed; X30
    a2 = num / alpha; X31
    a2 = a2 << 15; X32
    a1 = round(a2); X33
30  if (a0 > 0) a1 = -a1; X34
    rc = a1h; X35
                                         X36

```

THIS SHEET HAS BEEN
REPLACED

	a1l = 0;	X37
	a1h = alpha;	X38
	a1 = a1 + rc *2* a0h;	X39
	a1 = round(a1);	X40
5	if (alpha <= 0)	X41
	goto LEDUMO_failed;	X42
	alpha = a1h;	X43
		X44
	a0 = 0;	X45
10	a1 = 0;	X46
	for (ip=1; ip <= (minc>>1); ip++)	X47
	{ a0h = a[ip];	X48
	a0 = a0 + rc *2* a[minc-ip];	X49
	a0 = round(a0);	X50
15	If (overflow in a0)	X51
	{ EXPa = EXPa + 1;	X52
	for (lp=1; lp<minc; lp++)	X53
	a[lp] = a[lp]>>1;	X54
	a0h = a[ip];	X55
20	a0 = a0 + rc *2* a[minc-ip];	X56
	a0 = round(a0);	X57
	}	X58
	a1h = a[minc-ip];	X59
	a1 = a1 + rc *2* a[ip];	X60
25	a1 = round(a1);	X61
	If (overflow in a1)	X62
	{ EXPa = EXPa + 1;	X63
	for (lp=1; lp<minc; lp++)	X64
	a[lp] = a[lp]>>1;	X65
30	a0h = a[ip];	X66
	a0 = a0 + rc *2* a[minc-ip];	X67
	a0 = round(a0);	X68

This Sheet Has Been Replaced

THIS SHEET HAS BEEN
REPLACED

```

        alh = a[minc-ip];          X69
        al = al + rc *2* a[ip];    X70
        al = round(al);           X71
        }                          X72
5      a[ip  ] = a0h;              X73
        a[minc-ip] = alh;          X74
        }                          X75
        a2 = a2 >> EXPa;           X76
        a2 = round(a2);           X77
10     if (sign > 0) a2 = -a2;      X78
        a[minc] = a2h;             X79
        }                          X80
        a[0] = EXPa;               X81
        return;                   X82
15                                     X83
                                     X83
        LEDUMO_failed;             X84
        a[0] = -1;                 X85
        return;                   X86
20

```

This Sheet Has Been Replaced



Explanations of the pseudo code:

In fixed-point representation the prediction coefficients $a[i]$ can only assume values between -1.0 and +1.0. Since, however, the coefficients $a[i]$ can exceed 1.0 (rows X47 ... X75) during the recursion, all the coefficients are represented in the block floating-point format. This is to say, all the coefficients are normalized to a common basis 2^{EXP_a} , so that the mantissa of all the normalized prediction coefficients lies again between -1.0 and +1.0. At the start of the Levinson Recursion no coefficients have been computed yet, therefore the exponent EXP_a is set to 0 (row X1).

Row 01 is not carried out here, because $a(0) = 1.0$ need not additionally be computed.

Rows 02 and 03 are converted similarly in the loop having the running variable minc. Therefore, an explanation of the fixed-point resolution will be given

below.

There will be a description of the fixed-point implementation of rows 05 to 07: in rows X17, X18 a0 denotes a 32-bit-wide accumulation register. A multiplication input register is loaded with the autocorrelation coefficient $r[\text{minc-ip}]$. A further multiplication input register is loaded with the mantissa of the previously computed normalized predictor coefficient $a[ip]$; the associated exponent EXP_a is the same for all the coefficients. It should especially be emphasized that the coefficient $a[0]$ is not included in the normalization, because a multiplication by $a[0] = 1$ need not additionally be carried out.

The final result of the accumulation, first without the component $a[0]*2*r[\text{minc}]$, is reduced in row X21 from the floating-point representation to the fixed-point representation by a left shift arrangement for denormalizing the 32-bit result in accumulator a0.

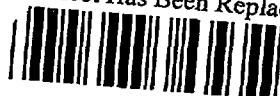
In rows X22 to X24 the component $a[0]*2*r[\text{minc}]$ is added up while a further 32-bit-wide register a1 is used. There should be noted that the lower part of the register a1 is to be erased, because, owing to its use, the auxiliary register a1 could still contain previous values at other locations.

In rows X25 to X28 the counter term for the division is prepared to compute the current (k^{th}) reflection coefficient rc . The division demands positive input data having a 16 bits data width which is achieved in row X25 by a rounding operation of the 32-bit-wide result in a0.

Registers at the input of a divider unit are directly loaded to carry out the division for the computation of the current (k^{th}) reflection coefficient rc ; an exponent is not necessary. There should be stated here that in [I] the division by floating-point values of α and α_{sum} , recovered from a0 of row X24, is carried out. This corresponds to a considerable increase of the necessary computing speed.

The result of the division in row X31 is computed with an accuracy of 17 bits, the 17th bit being necessary for the rounding operation. This rounding is performed in row X33 to compute the current (k^{th}) reflection coefficient and in row X77 to compute the current (k^{th}) prediction coefficient. In row X32 the result of the division, of which the 17 bits are stored right-justified in the 32-bit-wide register a2, is appropriately adapted to the fixedly set decimal point before the rounding operation.

This sheet Has Been Replaced



THIS SHEET HAS BEEN
REPLACED

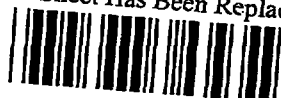
After the correct sign has been used in row X34, finally the current reflection coefficient rc is derived from the upper part of the 32-bit-wide register $a1$ by reading the upper 16-bit-wide word alh .

In rows X37 to X40 the new value of α is computed in the manner known henceforth, while all the input and output variables are represented in the 16-bit fixed-point format. Here too special mention is made of the fact that at this position in [I] computations are made with floating-point representation for α and $a0h$ ($a0h$ corresponds to sum in row 09), which corresponds to a considerable increase of the required computing speed.

In rows X45 to X74 the previously computed predictor coefficients are adapted (recomputation and adaptation of the block floating-point format). This corresponds to the operations from rows 015 to 018. The special thing about this is that in the case of an overflow during the recomputation of the prediction coefficients $a[ip]$ and $a[ming-ip]$ respectively, all the (i.e. the coefficients already recomputed and not yet recomputed, thus those previously computed) coefficients $a[1]$, $a[2]$, ... $a[ming-1]$ are reduced by the factor of $1/2$ (= right shift by 1 bit in row X54, X65, respectively). No overflow can then take place any longer when there is a recomputation in rows X55, X56, X57 and X66 to X71, respectively.

After the adjustment of the old coefficients (computed previously) has been terminated, the new (k^{th}) prediction coefficient $a[ming]$ is computed from rc (see row 020). In the fixed-point format there should be noted that also the new (k^{th}) coefficient is to be stored in the block floating-point format. Therefore, the reflection coefficient rc computed above, which is available in the non-normalized fixed-point format, is stored in $a[ming]$ after a right shift by EXP_a bits. With the solution given here, first the reflection coefficient, which is still stored in register $a2$ and has a 17-bit resolution, is shifted to the right by EXP_a bits and subsequently rounded with an accuracy of 16 bits. This is the decisive difference from the solution published in [I] in which the reflection coefficient already rounded directly after the division is shifted to the right by EXP_a bits and the new prediction coefficient is derived by clipping to 16 bits. This very difference at this position leads to the 16-bit fixed-point implementation of the Levinson Recursion described here which produces better results with attendant considerably less cost of computation than the solution shown in [I].

This Sheet Has Been Replaced



THIS SHEET HAS BEEN
REPLACED



Fig. 2 shows an adaptive prediction filter. Such prediction filters comprise a filter in a narrower sense 2 and a control unit 1. The filter in the narrower sense 2 produces output values $y(n)$ according to equation (1) and sends them over a line a2. Input values $s(n)$ are applied both to the filter in the narrower sense 2 and to the control unit 1 over a line b1.

The control unit 1 may again coarsely be subdivided into two functional units 1A and 1B. The functional unit 1A computes correlation coefficients from a section of the input signal, conveys these values to the functional unit 1B which determines therefrom, by reducing the expression (2), new prediction coefficients with the means described above. The control unit 1 then substitutes over a line b1 the whole block of coefficients computed by unit 1B for the prediction coefficients used so far. The described filter comprising the functional unit 1B can be used several times in modern speech codecs.

Fig. 3 shows a predictive speech coder by which the coded speech signal is also decoded to improve the coding. Therefore, a special description of a decoder is omitted.

The speech signal to be coded is applied in the form of sampling values to the coder via a line 6a. A subtracter 6 subtracts therefrom the appropriate values of the speech signal again decoded by the coder, which speech signal is applied to the subtracter 6 over a line 4a. Over a line 6b the difference is sent to a unit 9 which makes a spectral weighting of the difference signal (for an extensive representation compare, for example, CHEN, J.H.: HIGH-QUALITY 16 KB/s SPEECH CODING WITH A ONE-WAY DELAY LESS THAN 2MS. 1990, IEEE S9.1, pp. 453 to 456), after which for a speech signal segment the sum of the error squares is formed and stored. For a segment of the speech signal on line 6a, 128 different coding options are tested and, finally, the option bringing about the fewest errors is sent to the receiver as a code signal.

These 128 options arise from the fact that a code book referenced 1 stores 128 so-called vectors. Each vector is a sequence of sampling values, which sequence contains a simulation of the signal, by which the vocal tract of a human being can be excited, for example, for resonance. The vectors are read consecutively from code book 1 and scaled by a controllable scaler 2 (this corresponds to the volume control when

human speech is generated). The scaler 2 is controlled *via* a unit 3. Subsequently, the scaled vectors are fed to a predictor 4, 5 and 8 having backward prediction. The predictor forms a copy of the operation of the human vocal tract. It comprises a filter 5, 8 as shown in Fig. 2. Such filters are furthermore included in the units 3 and 9. *Via* a line 1a is transmitted, for example, the number of the vector that has led to the fewest errors in the current speech signal segment.

The diagrams of Fig. 4 graphically show what has been demonstrated hereinbefore with respect to two examples, that is to say, that with the conversion of prediction coefficients in a block floating point format with mantissas to 3 bits, the invention on average produces better results than according to the state of the art (I). Output data are all the positive binary numbers with 4-bits resolution. To facilitate the verification of the two lower diagrams, the upper diagram shows an assignment of the output data in binary representation (the sign bit is omitted) as against their representation in fractions.

15 The middle diagram shows the error F which is made if the output data are first rounded according to [I] and then the mantissas are computed with 3-bits resolution. The lower diagram shows the error F which is made if the mantissas are first computed with 4-bits resolution and then rounded to 3 bits. The scale for the error F is the same in either case and corresponds to that of the upper diagram.

This Sheet Has Been Replaced



THIS SHEET HAS BEEN
REPLACED

CLAIMS:

This Sheet Has Been Replaced



1. Transmission system comprising:
 - at least a coder (101) for coding a signal (10a),
 - at least a transmitter (102) for transmitting the coded signal (10b) to at least a receiver (103),
 - at least a decoder (104) for decoding the coded signal, and
 - 5 - at least an adaptive prediction filter (10A, 10B) provided at least for a coder, the adaptive prediction filter including:
 - means for computing correlation coefficients from segments of sampling values of an auxiliary signal that depends on the signal or on the decoded signal, as well as
 - 10 - means for determining reflection and prediction coefficients from the correlation coefficients in N-bit fixed point arithmetic, and in that the following means are provided for recursively determining the prediction coefficients to be represented in the block floating point format:
 - a) means (X31) for computing the k^{th} reflection coefficient
 - 15 b) means (X74, X75) for recomputing the $k-1$ previously computed prediction coefficients while utilizing the k^{th} reflection coefficient obtained under a),
 - c) means (X51-X58; X62-X72) for adapting the block floating point format of the $k-1$ prediction coefficients already recomputed and not yet recomputed if during the recomputation of the $k-1$ previously computed prediction coefficients overflow
 - 20 occurs, so that N bits are no longer sufficient to represent a recomputed prediction coefficient,
 - d) means (X62-X72) for continuing the recomputation of the $k-1$ prediction coefficients for the prediction coefficient at which the overflow as under c) has occurred,
 - 25 e) means (X76) for converting the k^{th} reflection coefficient to a floating point format number whose exponent is represented by the exponent determined under c) of the

THIS SHEET HAS BEEN
REPLACED



block floating point format, and

- f) means (X79) for storing the number obtained under e) as the k^{th} prediction coefficient.

2. Transmission system as claimed in Claim 1, characterized by:

- 5 2a) means (X31) for computing the k^{th} reflection coefficient with a minimum resolution of $N+1$ bits,
- 2b) means (a2) for buffering the k^{th} reflection coefficient with a minimum resolution of $N+1$ bits,
- 10 2c) means (X76) for converting the k^{th} reflection coefficient buffered under 2b) to a floating point format according to e) (Claim 1) whose mantissa has a resolution of at least $N+1$ bits,
- 2d) means (X77) for rounding to N bits the mantissa obtained under 2c) and
- 2e) means (X79) for storing the mantissa rounded under 2d) as the mantissa of the k^{th} prediction coefficient.

15 3. Terminal unit (LE) comprising:

- at least a coder (101) and/or at least a decoder (104),
- at least a transmitter (102) and/or at least a receiver (103), and
- at least an adaptive prediction filter (10A, 10B), the adaptive prediction filter including:

- 20 - means for computing correlation coefficients from segments of sampling values of a signal, as well as
- means for determining reflection and prediction coefficients from the correlation coefficients in N -bit fixed point arithmetic,

and in that the following means are provided for recursively determining the prediction coefficients to be represented in the block floating point format:

- 25 a) means (X31) for computing the k^{th} reflection coefficient
- b) means (X74, X75) for recomputing the $k-1$ previously computed prediction coefficients while utilizing the k^{th} reflection coefficient obtained under a),
- c) means (X51-X58; X62-X72) for adapting the block floating point format of the $k-1$ prediction coefficients already recomputed and not yet recomputed if during the
- 30 recomputation of the $k-1$ previously computed prediction coefficients overflow occurs, so that N bits are no longer sufficient to represent a recomputed prediction

coefficient,

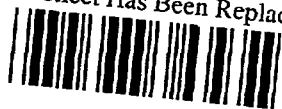
- d) means (X62-X72) for continuing the recomputation of the k-1 prediction coefficients for the prediction coefficient at which overflow as under c) has occurred,
 - 5 e) means (X76) for converting the k^{th} reflection coefficient to a floating point format number whose exponent is represented by the exponent determined under c) of the block floating point format, and
 - f) means (X79) for storing the number obtained under e) as the k^{th} prediction coefficient.
- 10 4. Terminal unit as claimed in Claim 3, characterized by:
- 2a) means (X31) for computing the k^{th} reflection coefficient with a minimum resolution of N+1 bits,
 - 2b) means (a2) for buffering the k^{th} reflection coefficient with a minimum resolution of N+1 bits,
 - 15 2c) means (X76) for converting the k^{th} reflection coefficient buffered under 2b) to a floating point format according to e) (Claim 3) whose mantissa has a resolution of at least N+1 bits,
 - 2d) means (X77) for rounding to N bits the mantissa obtained under 2c) and
 - 2e) means (X79) for storing the mantissa rounded under 2d) as the mantissa
 - 20 of the k^{th} prediction coefficient.
5. Coder (101) for coding a signal (10a),
- comprising at least an adaptive prediction filter (10A, 10B), the adaptive prediction filter including:
 - means for computing correlation coefficients from segments of
 - 25 sampling values of an auxiliary signal that depends on the signal or on the decoded signal, as well as
 - means for determining reflection and prediction coefficients from the correlation coefficients in N-bit fixed point arithmetic,
- 30 and in that the following means are provided for recursively determining the prediction coefficients to be represented in the block floating point format:
- a) means (X31) for computing the k^{th} reflection coefficient,





- b) means (X74, X75) for recomputing the $k-1$ previously computed prediction coefficients while utilizing the k^{th} reflection coefficient obtained under a),
- c) means (X51-X58; X62-X72) for adapting the block floating point format of the $k-1$ prediction coefficients already recomputed and not yet recomputed if during the recomputation of the $k-1$ previously computed prediction coefficients overflow occurs, so that N bits are no longer sufficient to represent a recomputed prediction coefficient,
- d) means (X62-X72) for continuing the recomputation of the $k-1$ prediction coefficients for the prediction coefficient at which overflow as under c) has occurred,
- e) means (X76) for converting the k^{th} reflection coefficient to a floating point format number whose exponent is represented by the exponent determined under c) of the block floating point format, and
- f) means (X79) for storing the number obtained under e) as the k^{th} prediction coefficient.
6. Coder (101) as claimed in Claim 5, characterized by
- 2a) means (X31) for computing the k^{th} reflection coefficient with a minimum resolution of $N+1$ bits,
- 2b) means (a2) for buffering the k^{th} reflection coefficient with a minimum resolution of $N+1$ bits,
- 2c) means (X76) for converting the k^{th} reflection coefficient buffered under 2b) to a floating point format according to e) (Claim 5) whose mantissa has a resolution of at least $N+1$ bits,
- 2d) means (X77) for rounding to N bits the mantissa obtained under 2c) and
- 2e) means (X79) for storing the mantissa rounded under 2d) as the mantissa of the k^{th} prediction coefficient.
7. Decoder (104) for decoding a signal (10c),
- comprising at least an adaptive prediction filter (10B), the adaptive prediction filter including:
 - means for computing correlation coefficients from segments of sampling values of an auxiliary signal that depends on the signal or on the decoded signal, as well as

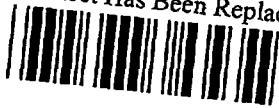




- means for determining reflection and prediction coefficients from the correlation coefficients in N-bit fixed point arithmetic, and in that the following means are provided for recursively determining the prediction coefficients to be represented in the block floating point format:
- 5 a) means (X31) for computing the k^{th} reflection coefficient
- b) means (X74, X75) for recomputing the $k-1$ previously computed prediction coefficients while utilizing the k^{th} reflection coefficient obtained under a),
- c) means (X51-X58; X62-X72) for adapting the block floating point format of the $k-1$ prediction coefficients already recomputed and not yet recomputed if during the
- 10 recomputation of the $k-1$ previously computed prediction coefficients overflow occurs, so that N bits are no longer sufficient to represent a recomputed prediction coefficient,
- d) means (X62-X72) for continuing the recomputation of the $k-1$ prediction coefficients for the prediction coefficient at which overflow as under c) has
- 15 occurred,
- e) means (X76) for converting the k^{th} reflection coefficient to a floating point format number whose exponent is represented by the exponent determined under c) of the block floating point format, and
- f) means (X79) for storing the number obtained under e) as the k^{th} prediction
- 20 coefficient.
8. Decoder (104) as claimed in Claim 7, characterized by:
- 2a) means (X31) for computing the k^{th} reflection coefficient with a minimum resolution of $N+1$ bits,
- 2b) means (a2) for buffering the k^{th} reflection coefficient with a minimum
- 25 resolution of $N+1$ bits,
- 2c) means (X76) for converting the k^{th} reflection coefficient buffered under 2b) to a floating point format according to e) (Claim 7) whose mantissa has a resolution of at least $N+1$ bits,
- 2d) means (X77) for rounding to N bits the mantissa obtained under 2c) and
- 30 2e) means (X79) for storing the mantissa rounded under 2d) as the mantissa of the k^{th} prediction coefficient.
9. Adaptive prediction filter (Fig. 2), comprising



This Sheet Has Been Replaced



means for computing correlation coefficients from segments of sampling values of a signal as well as

means for determining reflection and prediction coefficients from the correlation coefficients in N-bit fixed point arithmetic,

5

and in that the following means are provided for recursively determining the prediction coefficients to be represented in the block floating point format:

- a) means (X31) for computing the k^{th} reflection coefficient,
- b) means (X74, X75) for recomputing the $k-1$ previously computed prediction coefficients while utilizing the k^{th} reflection coefficient obtained under a),
- c) means (X51-X58; X62-X72) for adapting the block floating point format of the $k-1$ prediction coefficients already recomputed and not yet recomputed if during the recomputation of the $k-1$ previously computed prediction coefficients overflow occurs, so that N bits are no longer sufficient to represent a recomputed prediction coefficient,
- d) means (X62-X72) for continuing the recomputation of the $k-1$ prediction coefficients for the prediction coefficient at which overflow as under c) has occurred,
- e) means (X76) for converting the k^{th} reflection coefficient to a floating point format number whose exponent is represented by the exponent determined under c) of the block floating point format, and
- f) means (X79) for storing the number obtained under e) as the k^{th} prediction coefficient.

10

15

20

25 10.

Adaptive prediction filter (Fig. 2) as claimed in Claim 9, characterized by:

- 2a) means (X31) for computing the k^{th} reflection coefficient with a minimum resolution of $N+1$ bits,
- 2b) means (a2) for buffering the k^{th} reflection coefficient with a minimum resolution of $N+1$ bits,
- 2c) means (X76) for converting the k^{th} reflection coefficient buffered under 2b) to a floating point format according to e) (Claim 7) whose mantissa

30



THIS SHEET HAS BEEN REPLACED

has a resolution of at least $N+1$ bits,

- 2d) means (X77) for rounding to N bits the mantissa obtained under 2c) and
2e) means (X79) for storing the mantissa rounded under 2d) as the mantissa of the k^{th} prediction coefficient.



This Sheet Has Been Replaced



THIS SHEET HAS BEEN
REPLACED