(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 649 578 B1

(12) EUROPEAN PATENT SPECIFICATION

(45) Date of publication and mention of the grant of the patent:
14.05.2003 Bulletin 2003/20

(21) Application number: **93914555.3**

(22) Date of filing: **05.07.1993**

(51) Int Cl.[7]: **H03H 17/06**, H03H 17/02, G06F 15/00

(86) International application number:
PCT/AU93/00330

(87) International publication number:
WO 94/001933 (20.01.1994 Gazette 1994/03)

(54) DIGITAL FILTER HAVING HIGH ACCURACY AND EFFICIENCY

DIGITALES FILTER MIT HOHER GENAUIGKEIT UND EFFIZIENZ

FILTRE NUMERIQUE A HAUTE PRECISION ET A HAUT RENDEMENT

(84) Designated Contracting States:
BE DE DK FR GB IT NL SE

(30) Priority: **07.07.1992 AU PL338692**
07.07.1992 AU PL338792
20.04.1993 AU PL835793

(43) Date of publication of application:
26.04.1995 Bulletin 1995/17

(60) Divisional application:
03001549.9 / 1 304 797

(73) Proprietor: **LAKE TECHNOLOGY LIMITED**
Maroubra, NSW 2035 (AU)

(72) Inventor: **McGRATH, David, Stanley**
Randwick, NSW 2031 (AU)

(74) Representative: **Nicholls, Michael John**
J.A. KEMP & CO.
14, South Square
Gray's Inn
London WC1R 5JJ (GB)

(56) References cited:
EP-A- 0 250 048
EP-A- 0 448 758
WO-A-88/03341
US-A- 4 623 980
US-A- 4 992 967

- **Proceedings of the IEEE, Vol. 75, No. 9, issued September 1987 (New York), R.C. AGARWAL, "Vectorized Mixed Radix Discrete Fourier Transform Algorithms", pages 1283-1292, whole document.**

EP 0 649 578 B1

Description

FIELD OF THE INVENTION

[0001] The present invention relates to the art of electronic signal processing, and more particularly but not exclusively, to an electronic filtering environment wherein relatively high accuracy and efficiency is desired and a relatively short flow-through delay (termed "latency") is desired.

DESCRIPTION OF THE PRIOR ART

[0002] With reference to Fig. 1, electronic filters are utilised to modify the characteristics of an incoming electronic signal so *as* to provide an output signal which is modified in some defined fashion. In the case of Fig. 1 a "notch" filter is illustrated wherein, in the frequency domain, frequencies in the spectrum of the incoming signal S1 are attenuated in the F_1 to F_2 band so as to produce output signal S2.

[0003] Such filters can be implemented from entirely analog components although, in more recent times, there is a preference, in many circumstances, to implement the filter in a digital fashion. Digital implementation can be by means of dedicated digital circuitry or by means of computers (micro processors) programmed to operate as a filter.

[0004] Filters have many applications in the field of electronic modelling of real world conditions. For example, filters can be used to provide a model of the acoustic characteristics of rooms or halls. Filters are also used to model deficiencies in systems so as to apply appropriate correction factors for the purpose of removing (cancelling) imperfections in signals caused by the deficiencies.

[0005] Frequently it is desirable that such processing take place in "real time". Also, it is desirable that there is effectively no delay in filtering of a signal generated in a real/live environment so that the modelling/correcting steps performed by the filter are, to all intents and purposes, without any delay being perceptible to the end user.

[0006] To achieve this the delay introduced by the filter F while it performs its filtering function must be reduced to a negligible figure. That is, the time when signal S1 first presents to filter F and the time when the results of the filtering by filter F of the first incoming portion of signal S1 become available at the output of the filter S2 must be almost the same. The delay between these two events is hereinafter referred to as the "latency" of the filter system.

[0007] Where the filter F is implemented in a digital manner it may first be necessary to sample the incoming signal S1 (via an analog to digital converter) then perform the filtering function and then convert the digital signal back to an analog signal (by means of a digital to analog converter). The sampling process takes samples of the incoming signal at discrete time intervals t_i. The time between each sample is usually the same.

[0008] The sampling processing itself introduces finite delays into the system. Additionally, where the filter is implemented by one of the popular fast convolution techniques there is a delay introduced which, in very broad terms, is proportional to the accuracy (or length) of the filter.

[0009] Mathematically, the filtering operation (that is, the step of imposing the filter characteristic upon the incoming signal S1 so as to produce outgoing signal S2) is known as "convolution" in the time domain. The step of convolution in the time domain becomes a multiply operation in the frequency domain. That is, if the incoming signal S1 is first sampled, then Fourier transformed into the frequency domain, the frequency response of the filter F is vector multiplied with the Fourier transform of the signal S1. The signal is then inverse Fourier transformed to produce a sampled (convolved) output (which can be converted back to analog form if required).

[0010] Figure 2 shows the way a convolver (also known as a Finite Impulse Response (FIR) filter) has its impulse response $\{a_k\}$ measured (for a convolver operating on a treatment of sampled data). For a physical filter, a_k is zero for all k<0. For a general input sequence $\{x_k\}$, the filter's output $\{Y_k\}$ is defined as:

$$Y_k=\sum_{i=0}^{\infty} a_i \ x_{k-i} \qquad (1)$$

[0011] A linear filter such as this has a measurable latency, d, defined as:

$$a_d \neq 0, \text{ and}$$

$$a_k = 0 \text{ for all } k < d \tag{2}$$

[0012] In other words, a_d is the first non-zero value in the output sequence. The latency d is never negative in a physical system. In a similar fashion, for a Finite Impulse Response Filter, we can determine which is the LAST non-zero value in the output sequence. This will give us the length of the impulse response. If we call the length l_1 then this means that a_{d+1-1} is the last non-zero value in the output sequence (see Figure 3).

[0013] Typical schemes for implementing FIR filters fall into two categories:

1. Time domain filters that compute each output sample as each new input sample arrives, thus allowing latencies as low as d=0 or d=1. Typical filter lengths (l) are short.
2. Fast convolution filters that compute a number of output samples in a block. Typical filter lengths (l) can be very long. The lowest achievable latency is
   usually related to the filter length, $d \approx l \div K$ or

$$K \approx l \div d \tag{3}$$

where K is a measure of the efficiency of the particular algorithm used. A typical value of K, for the commonly used fast convolution algorithms such as illustrated in Figs. 4 and 5, is 0.5.

[0014] WO-88 03341 discloses an echo-canceller in which an input signal is divided into blocks and, on the decreased number of samples in each block, a fast Fourier transform and FIR type digital filtering are effected to decrease the processing delay while reducing the number of calculations. In particular, the echo-cancellor of WO-88/03341 implements a method of infinite impulse response (FIR) filtering an input signal to produce a filtered output signal which is an echo-cancelling signal. The method uses the error between an echo-signal derived from the input signal passed through an echo path and the filtered output signal to adaptively control a filter characteristic (specified in terms of time-domain impulse response values) which is an estimate of the impulse response of an echo path. The method comprises the steps of:

(a) dividing the input signal into overlapping successive input blocks of 2N samples, with each successive input block being delayed by N samples relative to the previous input block;
(b) updating each of k frequency-domain coefficient blocks using the error by:

   (i) taking N values of the error;
   (ii) adding N zeros to said N values of the error to form a block of 2N zero-padded values; and
   (iii) computing a frequency-domain transformation of said block of 2N zero-padded values to form 2N updating error coefficients; and
   (iv) using the updating error coefficients to update a corresponding one of said k frequency-domain coefficient blocks; and

(c) for each of said input blocks:

   (i) computing a frequency-domain transform of said input block to form a corresponding frequency-domain input block;
   (ii) combining together the most recent k successive said frequency-domain input blocks with k frequency-domain coefficient blocks, to produce k frequency-domain filtered blocks;
   (iii) summing together said k frequency-domain filtered blocks to form a single frequency-domain output block;
   (iv) computing an inverse transform, which is the inverse of said frequency-domain transform, of said frequency-domain output block to form a time-domain output block;
   (v) discarding the first N samples of said time-domain output block, to produce a new set of N output samples; and
   (vi) outputting said N output samples as a portion of said filtered signal.

[0015] It is an object of at least a preferred embodiment of the present invention to provide a method and apparatus for performing relatively long convolutions on digital sampled data so as to provide relatively higher efficiency for a given length than is ordinarily produced with other methods.

[0016] In this specification it is assumed that the filter characteristics can be modelled as approximately linear so that the principles of superposition can be applied.

[0017] Accordingly, according to the present invention there is provided a method of finite impulse response (FIR) filtering an input signal using a predetermined portion of a desired time domain impulse response representing a filter characteristic and specified in terms of time-domain impulse response values, so as to produce a filtered output signal, the method comprising the steps of:

(a) dividing said input signal into overlapping successive input blocks of P samples, with each successive input block being delayed by N samples relative to the previous input block, where P≥2N-1;
(b) creating each of substantially M frequency-domain coefficient blocks by:

(i) dividing said predetermined portion of the desired time domain impulse into a series of segments;
(ii) computing a frequency-domain transformation of each segment to form a corresponding one of said M frequency-domain coefficient blocks; and

(c) for each of said input blocks:

(i) computing a frequency-domain transform of said input block to form a corresponding frequency-domain input block;
(ii) combining together the most recent M successive said frequency-domain input blocks with M frequency-domain coefficient blocks, to produce M frequency-domain filtered blocks;
(iii) summing together said M frequency-domain filtered blocks to form a single frequency-domain output block;
(iv) computing an inverse transform, which is the inverse of said frequency-domain transform, of said frequency-domain output block to form a time-domain output block;
(v) discarding predetermined portions of said time-domain output block, to produce a new set of N output samples; and
(vi) outputting said N output samples as a portion of said output signal.

[0018] Embodiments of-the invention will now be described with reference to the accompanying drawings wherein:

Fig. 1	is a generalised block diagram of a filter operation in the frequency domain,
Fig. 2	defines the basic terminology used for a convolution filter,
Fig. 3	defines the latency and length of the filter of Fig. 2,
Fig. 4	illustrates in a diagrammatic flow chart form, a prior art method of processing sampled data by a Fast Fourier Transform approach,
Fig. 5	further illustrates the approach of Fig. 4,
Fig. 6,7,8	develop a method of filtering according to a generalised first embodiment of the invention whereby a relatively high efficiency factor K can be achieved as compared with the approach of Fig. 4,
Fig. 9	is a diagram of an embodiment of the invention implementing the method of Fig. 8 where the number of sub-filters is 6,
Fig. 10	illustrates in block diagram form a summed filter a part of which can be implemented advantageously with the filter of Fig. 8,
Fig. 11	is a block diagram of the summed filter of Fig. 10 incorporating sub-filters some of which implement the method of Fig. 8,
Fig. 12	illustrates the manner of processing of an input signal by an example of the filter arrangement of Fig. 10 which utilised five filter portions,
Fig. 13	illustrates the manner of selection of the filter characteristics of the filter of Fig. 12,
Fig. 14	is a bloc diagram of an alternative implementation of the summed filter cf Fig. 8,
Fig. 15	illustrates a typical flow of a (prior art) fast convolution algorithm implementation suitable for filters F_2 - F_5 of Fig. 12,
Fig. 16	illustrates a DFT engine which forms the basis for an explanation of a fourier transform algorithm optimised to process real number strings, and
Fig. 17	is a block diagram of a further embodiment of the invention wherein the summed filter of Fig. 10 is implemented utilizing a Modified Discrete Fourier Transform.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS OF THE INVENTION

1. Righ Efficiency Filter

[0019] Figure 4 illustrates the time-flow of a typical fast-convolution algorithm. This is an overlap-discard algorithm implemented using the Fast Fourier Transform (FFT). 2M words of input data that arrives during time segments a and b is processed fully during time segment c with a forward FFT, a vector multiply, and an inverse FFT. The resulting M words of output data are buffered, to be presented at the output during time segment d. The FFT and inverse FFT (IFFT) are *only used* to transform the data between the time-domain and frequency domain. The actual filter operation is executed in the vector multiply operation, which actually takes only a small fraction of the total compute time. So, the relevant parameters of the filter of Fig. 4 are:- Length = M,

Latency = 2M,

and, therefore K = 0.5

[0020] With reference to Figs. 6, 7 and 8, the rationale behind the method and apparatus according to at least one embodiment of the present invention is derived.

[0021] Fig. 6 illustrates a filter of length ML where the filter characteristics of each of the component filters F1, F2. .... F_n are separate, discrete, component portions of the desired filter characteristic for the entire filter assembly. The delays L, 2L.... (M-1) L are imposed as illustrated so as to recreate, following addition, an output y_k equivalent to that achieved by passing input samples x_k through a filter having the filter characteristic from which the filter characteristic portions for filters F1, F2..... were derived. Fig. 7 is derived by implementing the filters F1, F2.... of Fig. 6 using the Fast Fourier Transform algorithm of Fig. 5.

[0022] With reference to Fig. 8, reorganisation of the filter of Fig. 7 allows the use of only one Fast Fourier Transform module 11 and one inverse Past Fourier Transform module 12. It is implicit that the Fast Fourier Transform module is adapted to process a block of samples from input x_k equal to twice the length of each of the filters Filter 1, Filter 2. Filter 3..... illustrated in Fig. 8.

[0023] As previously stated the filter characteristic (impulse response) of each consecutive filter F1, F2.... is taken from and corresponds to consecutive corresponding portions from the impulse response desired of the entire filter module.

[0024] The time delay L before each Fast Fourier transformed block of data is passed through the next filter is equal to half the sample length originally processed by the Fast Fourier Transform module.

[0025] Figure 9 shows the computation of one block of output data. in a similar style to Figure 4, but using the improved length/latency efficiency method derived in Figs. 6, 7 and 8. The method of Fig. 8 as used by Fig. 9 is summarised below.

[0026] During time segment h. the input data that arrived during time segments f and g is FFT'd and the resulting block of Frequency Domain Input Data is stored for future use. We then compute the next block of Frequency Output Data, which is inverse FFT'd and presented as output during time segment i. The old way of computing fast-convolution simply took the latest block of Frequency Domain Input Data, and multiplied it by a vector that represents the desired filter response, to get the new Frequency Domain Output Data. The improved length/latency efficiency method uses a number of previous Frequency Domain Input Data blocks to compute the new Frequency Domain Output Data block, as shown in Figure 9. In this example, the blocks of filter data are called Filter A, Filter B, ..., Filter F. In this example, the filter implemented is 6 times as long as the filter implemented in Figure 4, but with no greater latency. By comparison with Fig. 4, the relevant parameters of the filter of Fig. 9 are:- Length = 6M,

Latency = 2M,

and, therefore K = 3

[0027] Fig. 8 summarises the logic behind the implementation of the embodiment of Fig. 9.

[0028] Particularly, it will be noted that the progressive delays L, 2L, 3L, ...(M-1)L of Fig. S are achieved in Fig. 9 by the taking of delayed, overlapped groupings of consecutive samples a, b, c, d, ...

[0029] The above described filter arrangement can be used advantageously in a low-latency FIR filter arrangement such as illustrated in Fig. 10.

[0030] Figure 10 shows an architecture for implementing an FIR filter by adding together N filters. If each filter is characterised as: Filter F_i, latency d_i, length l_i, then generally the N filters are chosen so that their latencies are ordered in ascending order, and furthermore $d_{i+1}=d_i+l_i$. This means that the first non-zero value in the impulse response of filter F_{i+1}, comes immediately following the last non-zero value in the impulse response of filter F_i. Hence this summation of filters results in a single, longer filter with its impulse response being the sum of the impulse responses of the N component filters.

[0031] The important property of this filter is the length/latency efficiency, K, which is higher than any of the component filter efficiencies.

[0032] That is, the filter of Fig. 10 uses the technique of adding together several filters to form a new filter which is as long as the sum of the component filter lengths, and whose latency is as short as the latency of the lowest-latency component filter.

[0033] Fig. 11 shows an implementation of the low-latency filter 10 of Fig. 10 wherein there are three filter modules F1, F2, F3. The first module F1 is a low-latency (d=O) time domain filter whilst filters F2 and F3 are implemented according to the embodiment described in respect of Figs. 8 and 3.

2. <u>A low-latency FIR filter</u>

[0034] As previously described. Fig. 10 shows an architecture for implementing an FIR filter by adding together N filters. If each filter is characterised as: Filter F_i, latency d_i, length l_i, then generally the N filters are chosen so that their latencies are ordered in ascending order, and furthermore $d_{i+1}=d_i+l_i$. This means that the first non-zero value in the impulse response of filter F_{i+1}, comes immediately following the last non-zero value in the impulse response of filter F_i. Hence this summation of filters results in a single, longer filter with its impulse response being the sum of the impulse responses of the N component filters.

[0035] The important property of this filter is the length/latency efficiency, K, which is higher than any of the component filter efficiencies.

[0036] That is, the filter of Figs. 10 and 12 uses the technique of adding together several filters to form a new filter which is as long as the sum of the component filter lengths, and whose latency is as short as the latency of the lowest-latency component filter.

[0037] Particularly, the composite filter assembly of Fig. 12 utilises the technique of combining a first time-domain (low latency) filter with additional fast-convolution (longer latency) filters to maximise filter length while minimising latency. This technique is implemented by adding together N filters, F_1, F_2, ...F_N where F_1 is a filter with very low latency, implemented with time-domain techniques, and the other filters. F_i. are each implemented with fast-convolution techniques. More specifically, the assembly adopts the technique whereby the N-1 fast-convolution filters, Fi, are composed of a sequence of filters, each with longer filter length than its predecessor, and hence each with longer latency, but still preserving the property that $d_{i+1}=d_i+1_i$. This ensures that the filter, F, which is made by summing together the N component filters, has an impulse response without any "holes" in it.

[0038] With particular reference to Fig. 12 the composite filter F comprises five filter portions F1, F2, F3, F4 and F5. The impulse response a_k of the composite filter F is illustrated at the top of Fig. 12 and has a total sample length of 1024 samples.

[0039] Filter F1 has an impulse response comprising the first 64 samples of the impulse response a_k. That is, the filter has a length of 64 samples. The filter as implemented has a low latency filter (such as is referenced in Motorola document APR 7/D in respect of the DSP 56000 series of Integrated Circuits). This filter has an effective latency of 0.

[0040] The subsequent filters F2, F3, F4, F5 are implemented using fast convolution digital techniques. Fig. 15 illustrates the basic algorithm for such techniques which comprises taking the fast Fourier transform of the incoming sampled data, multiplying the transformed data samples by the impulse response of the filter, converting the fast Fourier transformed data samples back to the time domain by use of an inverse fast Fourier transform and then outputting the data. An overlap/discard method is used whereby only a portion of the output data is utilised.

[0041] The length and latency of the additional filters F2, F3. F4, F5 is selected according to the rule illustrated diagrammatically in Fig. 13, whereby each filter portion has a latency equal to the sum of the length and the latency of the immediately preceding filter portion.

[0042] In this case the end result is a filter having a total length of 1024 samples and a latency of 0.

[0043] Fig. 14 illustrates a variation of the filter of Fig. 8 wherein delay is introduced after the filter algorithm is applied in the frequency domain.

3. <u>Optimized Real String Handling</u>

[0044] With reference to Fig. 16, a common method of frequency analysis is via the Discrete Fourier Transform (hereafter referred to as the DFT), which can be implemented efficiently in electronic apparatus using the Fast Fourier Transform algorithm (hereafter referred to as the FFT).

[0045] The DFT is formulated to operate efficiently when its input data and output data are both complex (having a real and imaginary component). When the data input to the DFT is real, the output data from the operation will contain some redundancy, indicating that some of the processing that led to this output data was unnecessary.

[0046] In this embodiment what is described is a new transform for operating on real numbers in the digital environment, that has many of the same applications as the DFT. but without the inefficiencies of the DFT for operation on real numbers. For the purposes of this document, the algorithm described herein will be named the Modified Discrete Fourier Transform (MDFT).

[0047] The DFT is computed according to the equation below:

$$X(n)=\sum_{k=0}^{N-1} x(k)e^{\frac{-2\pi jnk}{N}} \qquad (1)$$

[0048] If the input data x(k) is real (ie. it has no imaginary component), the output data X(n) has the following properties:

$$\begin{aligned} &X(0) \in \Re \\ &X(\tfrac{N}{2}) \in \Re \\ &X(n) = [X(N-n)]^{*} \; for \; 0<n<\tfrac{N}{2} \end{aligned} \qquad (2)$$

[0049] Where the * operator is used to signify complex conjugation. This means that the imaginary part of X(O), the imaginary part of X(N/2) and all {X(n):H/2<n<N} are redundant. The process of extracting only the necessary information out of the DFT output is therefore not trivial.

[0050] An alternative transform is shown below:

$$Y(n) = \sum_{k=0}^{N-1} x(k)e^{\frac{-2\pi jk(n+\frac{1}{2})}{N}} \qquad (3)$$

[0051] This is like a standard DFT except that the output vector Y(n) represents the signal's frequency components at different frequencies to the DFT. The output vector Y(n) has redundancies (just as the DFT output X(n) has), except that the redundant part of the data is more clearly extracted from Y(n) that from X(n). The redundancy in Y(n) that results when x(k) is real can be expressed as follows:

$$Y(n) = [Y(N\text{-}1\text{-}n)]^{*} \qquad (4)$$

[0052] This implies that the second half of this vector Y(n) is simply the complex conjugate of the first half, so that only the first half of the output vector is required to contain all of the information, when x(k) is real.

[0053] An alternative view of the above equation is that all of the odd elements of the vector are simply the complex conjugate of the even elements:

$$Y(1) = [Y(N\text{-}2)]^{*} \;\; Y(3) = [Y(N\text{-}4)]^{*} \;\; Y(N\text{-}3) = [Y(2)]^{*}$$

$$Y(N\text{-}1) = [Y(0)]^{*} \qquad (5)$$

[0054] This means that we only need to compute the even elements of Y(n) to obtain all of the required information from the modified DFT of the real signal x(k). We can name the array Z(p) the array that contains the even elements from Y(n), as follows:

$$Z(p) = Y(2p) \; for \; 0 \le p < \frac{N}{2} \qquad (6)$$

Based on our previous equation for Y(n), we get:

$$Z(p) = Y(2p) = \sum_{k=0}^{N-1} x(k) e^{\frac{-2\pi kj(2p+\frac{1}{2})}{N}} \quad (7)$$

which after some manipulation becomes:

$$Z(p) = \sum_{k=0}^{\frac{N}{2}-1} [x(k) - jx(k+\frac{N}{2})] e^{\frac{-\pi jk}{N}} e^{\frac{-2\pi jpk}{(N/2)}} \quad (8)$$

if we create an N/2 length complex vector from the N length real vector:

$$x'(l) = [x(l) - jx(l+\frac{N}{2})]e^{\frac{-\pi jl}{N}} \ \text{for}\ 0 \le l < N/2 \quad (9)$$

then we can see that:

$$Z(p) = DFT_{(N/2)}[x'(l)] \quad (10)$$

[0055] This means that we have computed the vector Z(p) by using a DFT of length N/2.

[0056] We say that Z(p)=MDFT[x(k)] (where MDFT indicates the Modified Discrete Fourier Transform operator). The procedure to follow for computing Z(p) is then as follows:

1. Take the input vector x(k) of length N, where each element of x(k) is real.
2. Create the vector x'(l), a complex vector of length N/2 by the method of equation (9) above.
3. Compute the N/2 point DFT of x'(l) to give the N/2 complex result vector Z(p).

[0057] The MDFT has many properties that make it useful in similar applications to the DFT. Firstly, it can be used to perform linear convolution in the same way as the DFT. Secondly, it has an inverse transform that looks very similar to the forward transform:

$$x'(k) = IDFT_{(N/2)}[Z(p)] \quad (11)$$

where IDFT indicates the N/2 point Inverse Discrete Fourier Transform.

[0058] The algorithm can be implemented in an electronic apparatus supplied with a set of N real numbers and producing N/2 complex output numbers, representing the MDFT of the input data. This apparatus uses digital arithmetic elements to perform each of the arithmetit operations as described in the preceding text.

[0059] Another embodiment of the present invention is a pair of apparatus, the first of which computes an MDFT as described in the previous paragraph, and the second of which computes an inverse MDFT, using the arithmetic procedures described previously in this document. 3y passing overlapped blocks of data from a continuous stream of input data through the MDFT computer, then multiplying the Z(p) coefficients by appropriate filter coefficients, then passing the resulting data through the Inverse MDFT computer, and recombining segments of output data appropriately, a modified Fast Convolution processor can be built.

[0060] The above described a modification to the DFT that makes it more useful in a number of applications particularly but not limited to the real time filter applications previously described. All of these extensions to the DFT can also be applied to the FFT algorithm and other fast implementations of the DFT.

<u>Example 1</u>

[0061] Fig. 17 illustrates an implementation of the summed filter of Fig. 11 wherein the Modified Discrete Fourier

Transform (MDFT) described immediately above is applied for the purposes of transforming the data stream into the frequency domain and the corresponding Inverse Modified Discrete Fourier Transform (IMDFT) is applied following application of the filter algorithm and prior to discard for conversion from the frequency domain.

[0062] In filter F2 of Fig. 17. the MDFT takes 64 real words of input and produces 32 complex words of output. The IMDFT takes 32 complex words of input and produces 64 real words of output.

[0063] In filter F3 of Fig. 17 the MDFT takes 256 real words of input and produces 128 complex words of output. The IMDFT takes 128 complex words of input and produces 256 real words of output.

[0064] The filter of Fig. 17 is implemented using a Motorola DSP 56001 processor incoporating software (bootable from ROM or from another host computer) to implement the algorithm. The delay elements are implemented using a bank of external memory chips comprising three MCM 6206 memory chips.

[0065] Data input and output between the analog and digital domains is effected by an ADC and DAC chip, the Crystal CS 4216, communicating via the syncrhonous serial communication port of the DSP 56001.

INDUSTRIAL APPLICABILITY

[0066] Embodiments of the invention may be applied to digital filters implemented in software, hardware or a combination of both for applications such as audio filtering or electronic modelling of acoustic system characteristics. The method is broadly applicable in the field of signal processing and can be used to advantage, for example, in: adaptive filtering; audio reverberation processing; adaptive echo cancellation; spatial processing; virtual reality audio; correlation. radar; radar pulse compression; deconvolution; seismic analysis: telecommunications; pattern recognition; robotics; 3D acoustic modelling; audio post production (including oralisation, auto reverberant matching); audio equalisation: compression; sonar; ultrasonics; secure communication systems: digital audio broadcast, acoustic analysis: surveillance; noise cancellation; echo cancellation.

Claims

1. A method of finite impulse response (FIR) filtering an input signal using a predetermined portion of a desired time domain impulse response representing a filter characteristic and specified in terms of time-domain impulse response values, so as to produce a filtered output signal, the method comprising the steps of:

(a) dividing said input signal into overlapping successive input blocks of P samples, with each successive input block being delayed by N samples relative to the previous input block, where $P \geq 2N-1$;

(b) creating each of substantially M frequency-domain coefficient blocks by:

(i) dividing said predetermined portion of the desired time domain impulse into a series of segments;

(ii) computing a frequency-domain transformation of each segment to form a corresponding one of said M frequency-domain coefficient blocks; and

(c) for each of said input blocks:

(i) computing a frequency-domain transform of said input block to form a corresponding frequency-domain input block;

(ii) combining together the most recent M successive said frequency-domain input blocks with M frequency-domain coefficient blocks, to produce M frequency-domain filtered blocks;

(iii) summing together said M frequency-domain filtered blocks to form a single frequency-domain output block;

(iv) computing an inverse transform, which is the inverse of said frequency-domain transform, of said frequency-domain output block to form a time-domain output block;

(v) discarding predetermined portions of said time-domain output block, to produce a new set of N output samples; and

(vi) outputting said N output samples as a portion of said output signal.

2. A method as claimed in claim 1, wherein P is equal to 2N.

3. A method as claimed in any one of the preceding claims wherein at least one of said frequency-domain transformation said frequency-domain transform comprises a 2N-point Real fast Fourier transform producing and N complex values in said frequency-domain coefficient block.

4. A method as claimed in claim 3, wherein said inverse transform comprises an N complex value fast Fourier inverse transform producing 2N-point Real values in said output block.

5. A method as claimed in any previous claim, wherein said step of combining together frequency-domain input blocks comprises element by element multiplication of said frequency-domain input blocks.

6. A method as claimed in any previous claim, wherein said summing together of frequency-domain filtered blocks comprises element by element addition of the blocks.

7. A method as claimed in any previous claim, wherein said combining and said summing are carried out in a single operation with the successive results of the combining operations being accumulated into the frequency-domain output block.

8. A method as claimed in any previous claim, wherein said time-domain output block is of length 0.5 P.

9. A method as claimed in any previous claim, wherein said discarding step (v) comprises discarding P-N samples.

10. A method as claimed in any previous claim, wherein said predetermined portion of a desired impulse response comprises M times N time-domain values h(k) where $0 \le k < NM$ and the m-th block ($0 \le m < M$) of values is made up of the N sample points h(mN) to h(mN+N-1).

11. A method as claimed in claim 1 wherein said method is applied in parallel to a series of predetermined portions of an overall desired impulse response representing a filter characteristic and specified in terms of time-domain response values with the outputs of each parallel application of the method being combined so as to form an overall output which comprises a filtering of the input signal with the overall desired impulse response.

12. A method as claimed in claim 11 wherein said series of predetermined portions are of different lengths.

13. A method as claimed in claim 12 wherein initial members of said series of predetermined portions are shorter then subsequent members of said series.

14. A method as claimed in claim 11 wherein the latency of application of said method is varied so that each parallel application of the method produces an output for combination substantially simultaneously.

Patentansprüche

1. Verfahren zum Filtern eines Eingangssignals mit finiter Impulsantwort (FIR) mittels eines vorbestimmten Teils einer gewünschten Zeitdomänen-Impulsantwort, die eine Filtercharakteristik darstellt und hinsichtlich Zeitdomänen-Impulsantwortwerten spezifiziert ist, um ein gefiltertes Ausgangssignal zu erzeugen, wobei das Verfahren folgende Schritte umfasst:

(a) Aufteilen des Eingangssignals in überlappende aufeinanderfolgende Eingangsblöcke von P Abtastungen, wobei jeder aufeinanderfolgende Eingangsblock um N Abtastungen bezüglich des vorhergehenden Eingangsblocks verzögert wird, wobei $P \ge 2N-1$ ist;
(b) Erzeugen von im wesentlichen M Frequenzdomänen-Koeffizientenblöcken durch:

(i) Aufteilen des vorbestimmten Teils des gewünschten Zeitdomänen-Impulses in eine Reihe von Segmenten;
(ii) Berechnen einer Frequenzdomänen-Transformation von jedem Segment, um einen entsprechenden

Block der M Frequenzdomänen-Koeffizientenblöcke zu bilden; und

(c) für jeden Eingangsblock:

(i) Berechnen einer Frequenzdomänen-Transformierten des Eingangsblocks, um einen entsprechenden Frequenzdomänen-Eingangsblock zu bilden;
(ii) Zusammenfassen des letzten M aufeinanderfolgenden Frequenzdomänen-Eingangsblöcke mit M Frequenzdomänen-Koeffizientenblöcken, um M Frequenzdomänengefilterte Blöcke zu erzeugen;
(iii) Aufsummieren der M Frequenzdomänen-gefilterten Blöcke, um einen einzigen Frequenzdomänen-Ausgangsblock zu bilden;
(iv) Berechnen einer Rücktransformierten, die die Umkehrung der Frequenzdomänen-Transformierten des Frequenzdomänen-Ausgangsblocks ist, um einen Zeitdomänen-Ausgangsblock zu bilden;
(v) Verwerfen vorbestimmter Teile des Zeitdomänen-Ausgangsblocks, um einen neuen Satz von N Ausgangsabtastungen zu bilden; und
(vi) Ausgeben der N Ausgangsabtastungen als einen Teil des Ausgangssignals.

2. Verfahren gemäß Anspruch 1, bei dem P gleich 2N ist.

3. Verfahren gemäß einem der vorhergehenden Ansprüche, bei dem mindestens eine der Frequenzdomänen-Transformation und der Frequenzdomänen-Transformierten eine Fast-Fourier-Transformierte mit N reellen Punkten umfasst, die N komplexe Werte in dem Frequenzdomänen-Koeffizientenblock erzeugt.

4. Verfahren gemäß Anspruch 3, bei dem die Rücktransformierte eine Fast-Fourier-Rücktransformierte mit N komplexen Werte umfasst, die 2N Punkte reeller Werte in dem Ausgangsblock erzeugt.

5. Verfahren gemäß einem der vorhergehenden Ansprüche, bei dem der Schritt des Zusammenfassens der Frequenzdomänen-Eingangsblöcke eine elementweise Multiplikation der Frequenzdomänen-Eingangsblöcke umfasst.

6. Verfahren gemäß einem der vorhergehenden Ansprüche, bei dem das Aufsummieren der Frequenzdomänen-gefilterten Blöcke eine elementweise Addition der Blöcke umfasst.

7. Verfahren gemäß einem der vorhergehenden Ansprüche, bei dem das Zusammenfassen und das Aufsummieren in einer einzigen Operation durchgeführt werden, wobei die aufeinander folgenden Ergebnisse der Zusammenfassungsoperationen in dem Frequenzdomänen-Ausgangsblock akkumuliert werden.

8. Verfahren gemäß einem der vorhergehenden Ansprüche, bei dem der Zeitdomänen-Ausgangsblock eine Länge von 0,5 P aufweist.

9. Verfahren gemäß einem der vorhergehenden Ansprüche, bei dem der Verwerfungsschritt (v) ein Verwerfen von P-N Abtastungen umfasst.

10. Verfahren gemäß einem der vorhergehenden Ansprüche, bei dem der vorbestimmte Teil einer gewünschten Impulsantwort M mal N Zeitdomänenwerte h(k) umfasst, wobei $0<k<NM$ und der m-te Block ($0<m<M$) von Werten aus den N Abtastpunkten h(mN) bis h(mN+N-1) zusammengesetzt ist.

11. Verfahren gemäß Anspruch 1, bei dem das Verfahren parallel auf eine Reihe vorbestimmter Teile einer gewünschten Gesamtimpulsantwort angewendet wird, die eine Filtercharakteristik darstellt und hinsichtlich Zeitdomänen-Antwortwerten spezifiziert ist, wobei die Ausgaben jeder parallelen Anwendung des Verfahrens zusammengefasst werden, um eine Gesamtausgabe zu bilden, die ein Filtern des Eingangssignals mit der gewünschten Gesamtimpulsantwort umfasst.

12. Verfahren gemäß Anspruch 11, bei dem die Reihe von vorbestimmten Teilen unterschiedliche Längen haben.

13. Verfahren gemäß Anspruch 12, bei dem die Anfangselemente der Reihe von vorbestimmten Teilen kürzer als nachfolgende Elemente der Reihe sind.

14. Verfahren gemäß Anspruch 11, bei dem die Latenzzeit der Anwendung des Verfahrens geändert wird, so dass jede parallele Anwendung des Verfahrens eine Ausgabe zur im wesentlichen gleichzeitigen Zusammenfassung erzeugt.

Revendications

1. Procédé de réponse impulsionnelle finie (RIF) filtrant un signal d'entrée utilisant une partie prédéterminée d'une réponse impulsionnelle temporelle désirée représentant une caractéristique de filtrage et spécifiée en fonction de valeurs de réponse impulsionnelle temporelle, afin de produire un signal de sortie filtré, le procédé comprenant les étapes consistant à :

(a) diviser ledit signal d'entrée en blocs d'entrée successifs superposés de P échantillons, avec chaque bloc d'entrée successif retardé par N échantillons par rapport au bloc d'entrée précédent, où $P \geq 2N-1$;
(b) créer chacun des blocs de coefficient substantiellement dans le domaine de fréquence M :

(i) en divisant ladite partie prédéterminée de l'impulsion temporelle désirée en une série de segments ;
(ii) en calculant une transformation dans le domaine de fréquence de chaque segment pour former un bloc correspondant desdits M blocs de coefficient dans le domaine de fréquence ; et

c) pour chacun desdits blocs d'entrée :

(i) en calculant une transformée dans le domaine de fréquence dudit bloc d'entrée pour former un bloc d'entrée dans le domaine de fréquence correspondant ;
(ii) en combinant ensemble les M blocs d'entrée successifs dans ledit domaine de fréquence avec M blocs de coefficient dans le domaine de fréquence, pour produire M blocs filtrés dans le domaine de fréquence ;
(iii) additionner ensemble lesdits M blocs filtrés dans le domaine de fréquence pour former un seul bloc de sortie dans le domaine de fréquence ;
(iv) en calculant une transformée inverse, qui est l'inverse de ladite transformée dans le domaine de fréquence, dudit bloc de sortie dans le domaine de fréquence pour former un bloc de sortie dans le domaine temporel ;
(v) en rejetant des parties prédéterminées dudit bloc de sortie dans le domaine temporel, pour produire un nouvel ensemble de N échantillons de sortie ; et
(vi) en délivrant lesdits N échantillons de sortie comme une partie dudit signal de sortie.

2. Procédé selon la revendication 1, dans lequel P est égal à 2N.

3. Procédé selon l'une quelconque des revendications précédentes, dans lequel au moins une de ladite transformation dans le domaine de fréquence et de ladite transformée dans le domaine de fréquence comprend une transformée de Fourier rapide réelle de 2N points produisant N valeurs complexes dans ledit bloc de coefficients dans le domaine de fréquence.

4. Procédé selon la revendication 3, dans lequel ladite transformée inverse comprend une transformée inverse de Fourier rapide de N valeurs complexes produisant des valeurs réelles de 2N points dans ledit bloc de sortie.

5. Procédé selon l'une quelconque des revendications précédentes, dans lequel ladite étape consistant à combiner ensemble les blocs d'entrée dans le domaine de fréquence comprend une multiplication élément par élément desdits blocs d'entrée dans le domaine de fréquence.

6. Procédé selon l'une quelconque des revendications précédentes, dans lequel ladite sommation des blocs filtrés dans le domaine de fréquence comprend l'addition élément par élément des blocs.

7. Procédé selon l'une quelconque des revendications précédentes, dans lequel ladite combinaison et ladite sommation sont exécutées dans une simple opération avec les résultats successifs des opérations de combinaison étant cumulées dans le bloc de sortie dans le domaine de fréquence.

8. Procédé selon l'une quelconque des revendications précédentes, dans lequel ledit bloc de sortie dans le domaine temporel a une longueur de 0,5 P.

9. Procédé selon l'une quelconque des revendications précédentes, dans lequel ladite étape de suppression (v) comprend la suppression de P-N échantillons.

10. Procédé selon l'une quelconque des revendications précédentes, dans lequel ladite partie prédéterminée d'une réponse impulsionnelle désirée comprend M fois N valeurs dans le domaine temporel h(k) où $0 \leq k < NM$ et le $m^{\text{iène}}$ bloc ($0 \leq m < M$) de valeurs est constitué de N points d'échantillonnage h(mN) à h(mN + N - 1).

11. Procédé selon la revendication 1, dans lequel ledit procédé est appliqué en parallèle à une série de parties prédéterminées d'une réponse impulsionnelle globale désirée représentant une caractéristique de filtre et spécifiée en fonction de valeurs de réponse dans le domaine temporel avec les sorties de chaque application parallèle du procédé étant combinées afin de former une sortie globale qui comprend un filtrage du signal d'entrée avec la réponse impulsionnelle globale désirée.

12. Procédé selon la revendication 11, dans lequel lesdites séries de parties prédéterminées sont de différentes longueurs.

13. Procédé selon la revendication 12, dans lequel les éléments initiaux desdites séries de parties prédéterminées sont plus courts que les éléments suivants desdites séries.

14. Procédé selon la revendication 11, dans lequel le temps d'attente d'application dudit procédé est modifié afin que chaque application parallèle du procédé produise une sortie pour combinaison substantiellement simultanée.

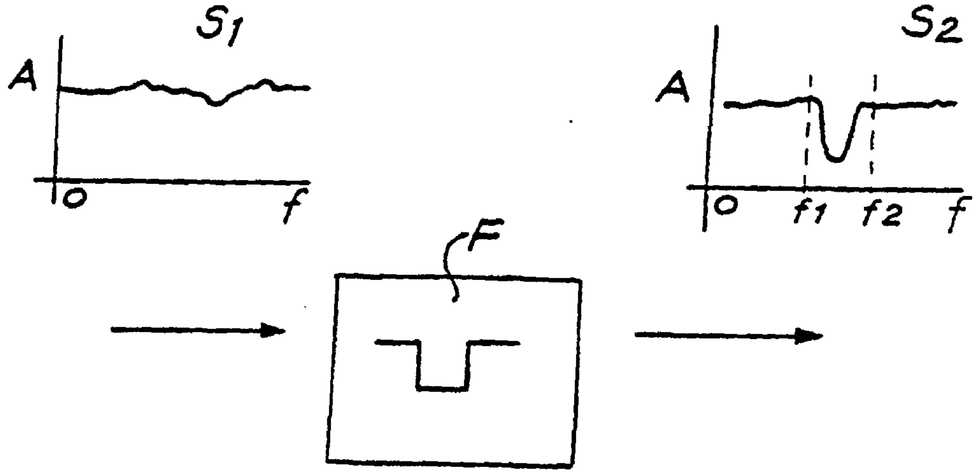


FIG. 1

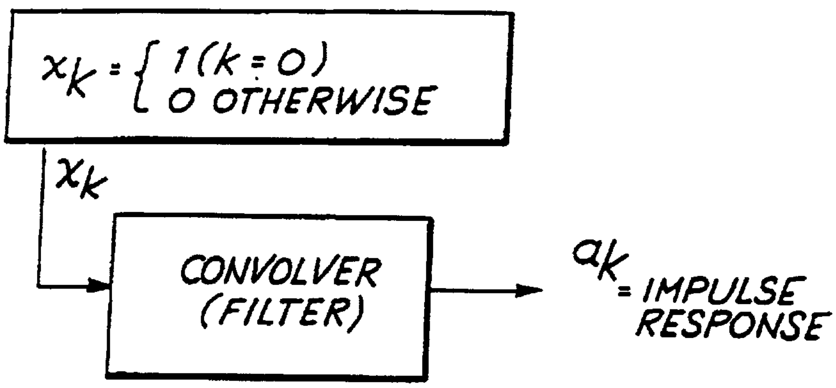

$x_k = \begin{cases} 1 (k = 0) \\ 0 \text{ OTHERWISE} \end{cases}$
x_k
CONVOLVER
(FILTER)
a_k = IMPULSE
RESPONSE

FIG. 2

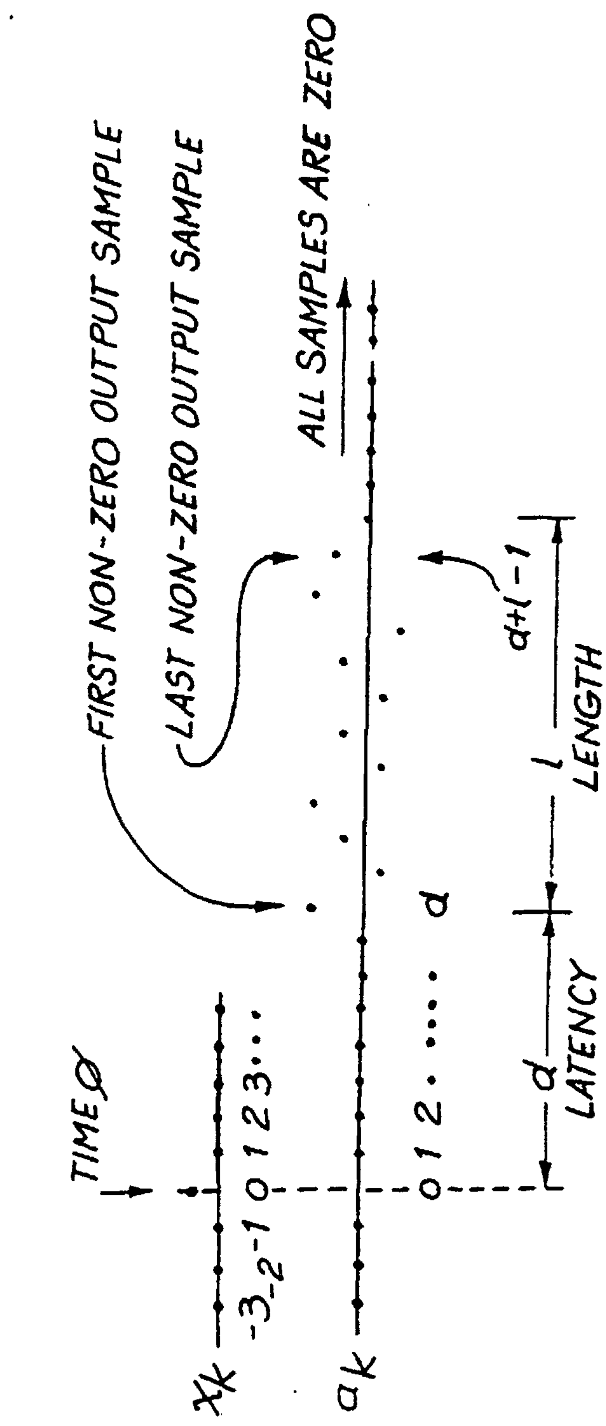

FIRST NON-ZERO OUTPUT SAMPLE
LAST NON-ZERO OUTPUT SAMPLE
ALL SAMPLES ARE ZERO
TIME Ø
x_k
a_k
-3 -2 -1 0 1 2 3 ...
0 1 2 .... d
d+l-1
d
LATENCY
l
LENGTH

FIG.3

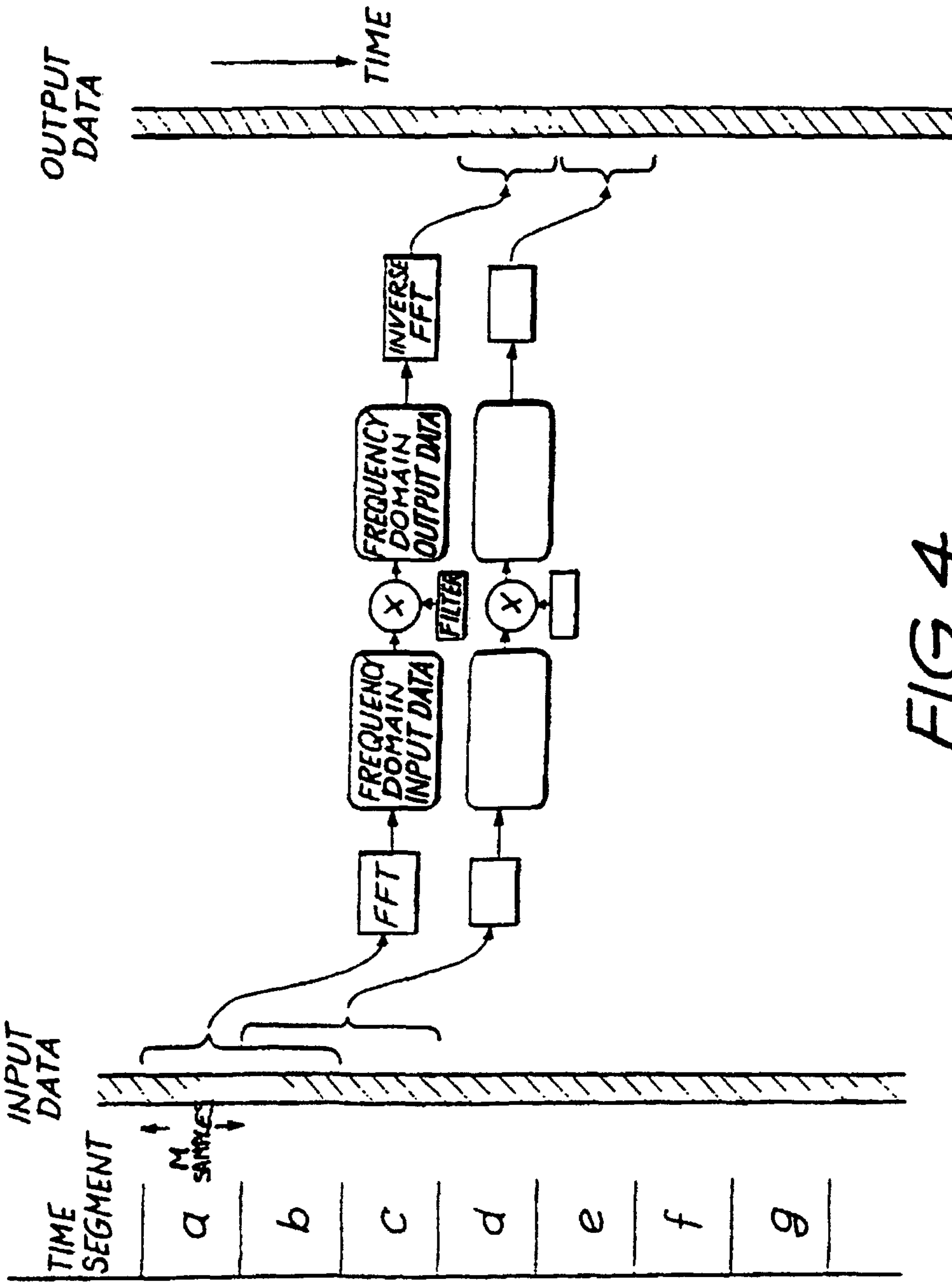

OUTPUT DATA
TIME
INVERSE FFT
FREQUENCY DOMAIN OUTPUT DATA
FILTER
X
X
FREQUENCY DOMAIN INPUT DATA
FFT
INPUT DATA
M SAMPLES
TIME SEGMENT
a
b
c
d
e
f
g

FIG. 4

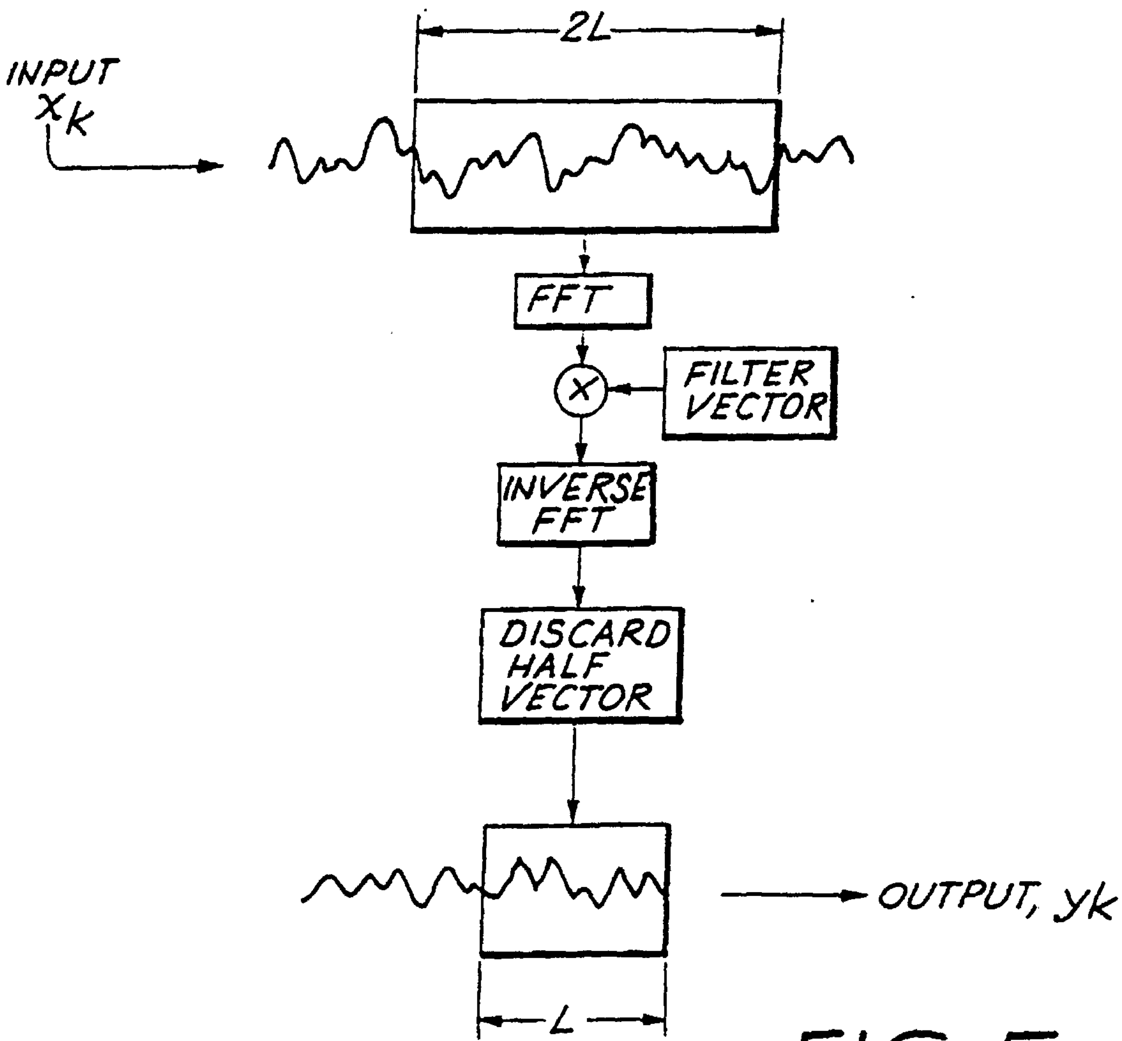

2L
INPUT
x_k
FFT
FILTER VECTOR
INVERSE FFT
DISCARD HALF VECTOR
OUTPUT, yk
L

FIG. 5

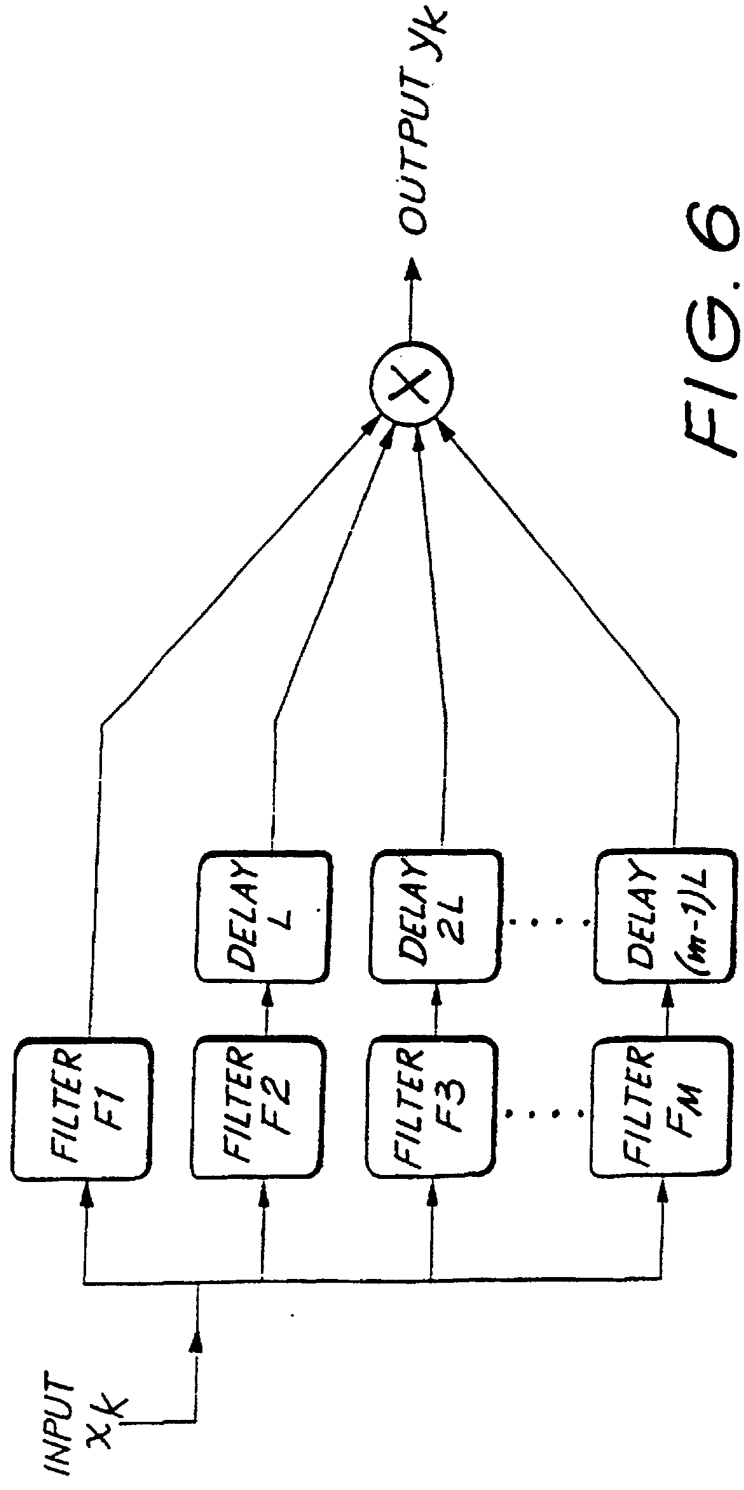

INPUT x_k
FILTER F1
FILTER F2
FILTER F3
FILTER F_M
DELAY L
DELAY 2L
DELAY (m-1)L
OUTPUT y_k

FIG. 6

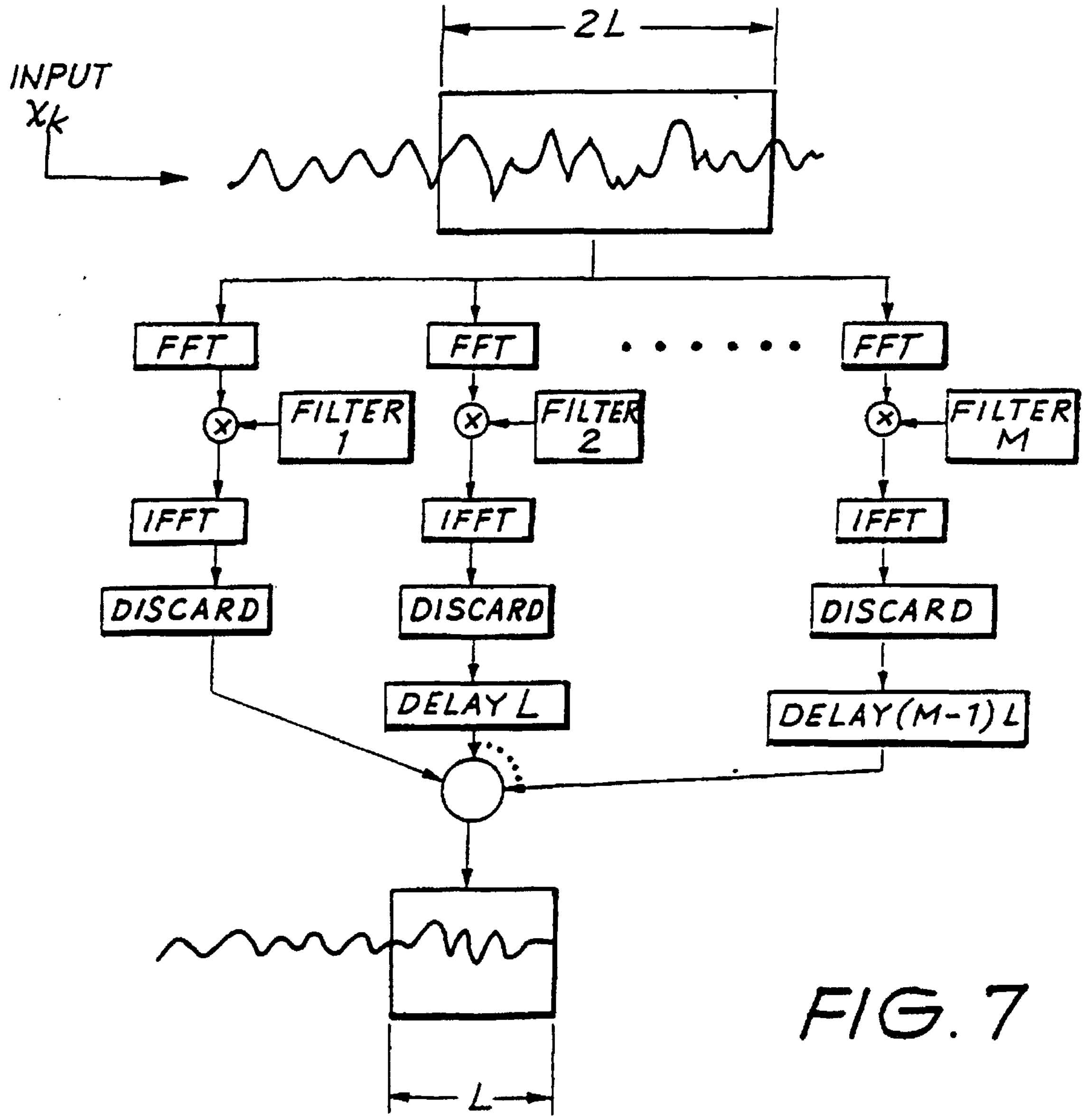

2L
INPUT
x_k
FFT
FFT
FFT
FILTER 1
FILTER 2
FILTER M
IFFT
IFFT
IFFT
DISCARD
DISCARD
DISCARD
DELAY L
DELAY (M-1) L
L

FIG. 7

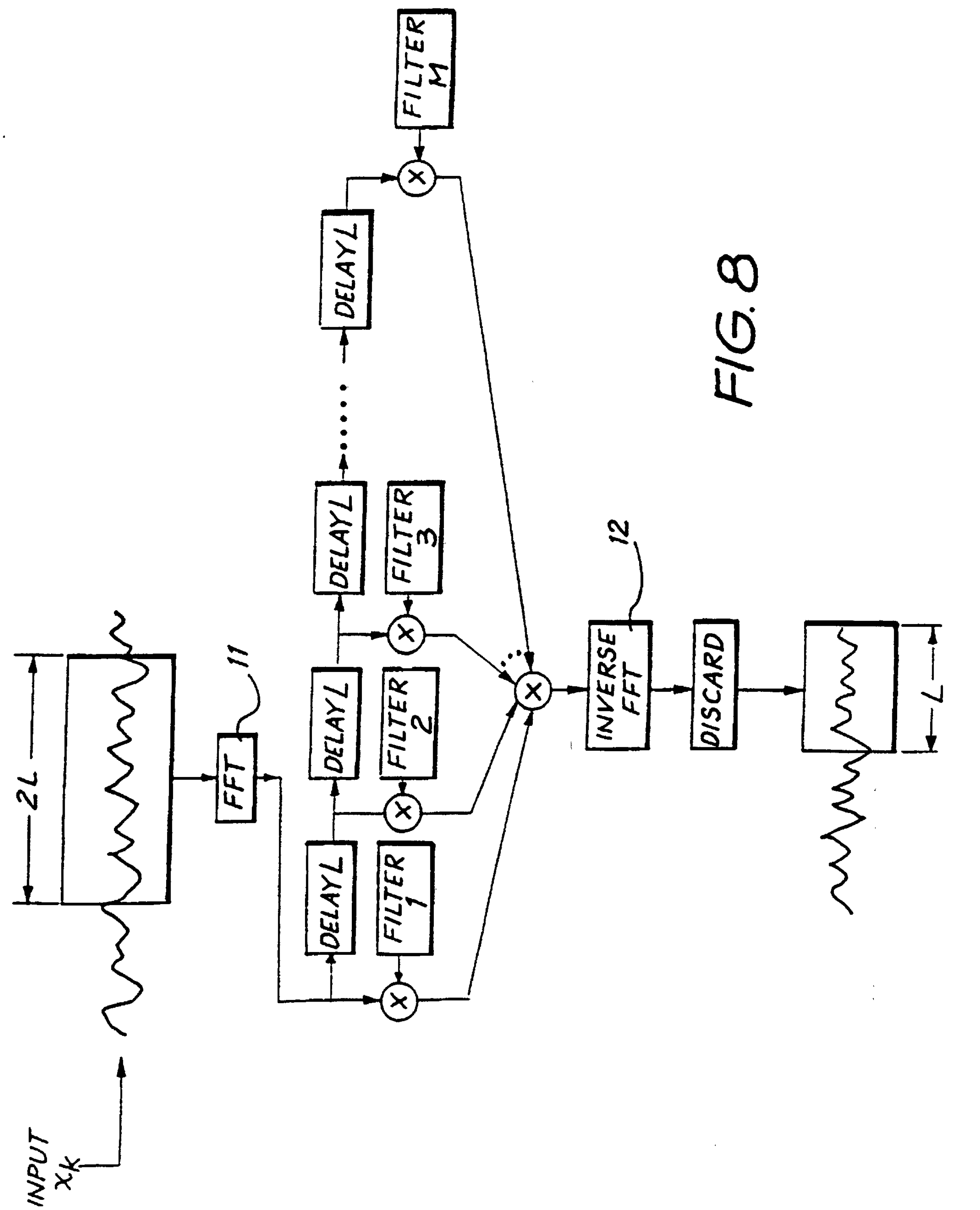

INPUT
x_k
2L
FFT
11
DELAY L
DELAY L
DELAY L
DELAY L
FILTER 1
FILTER 2
FILTER 3
FILTER M
INVERSE FFT
12
DISCARD
L

FIG. 8

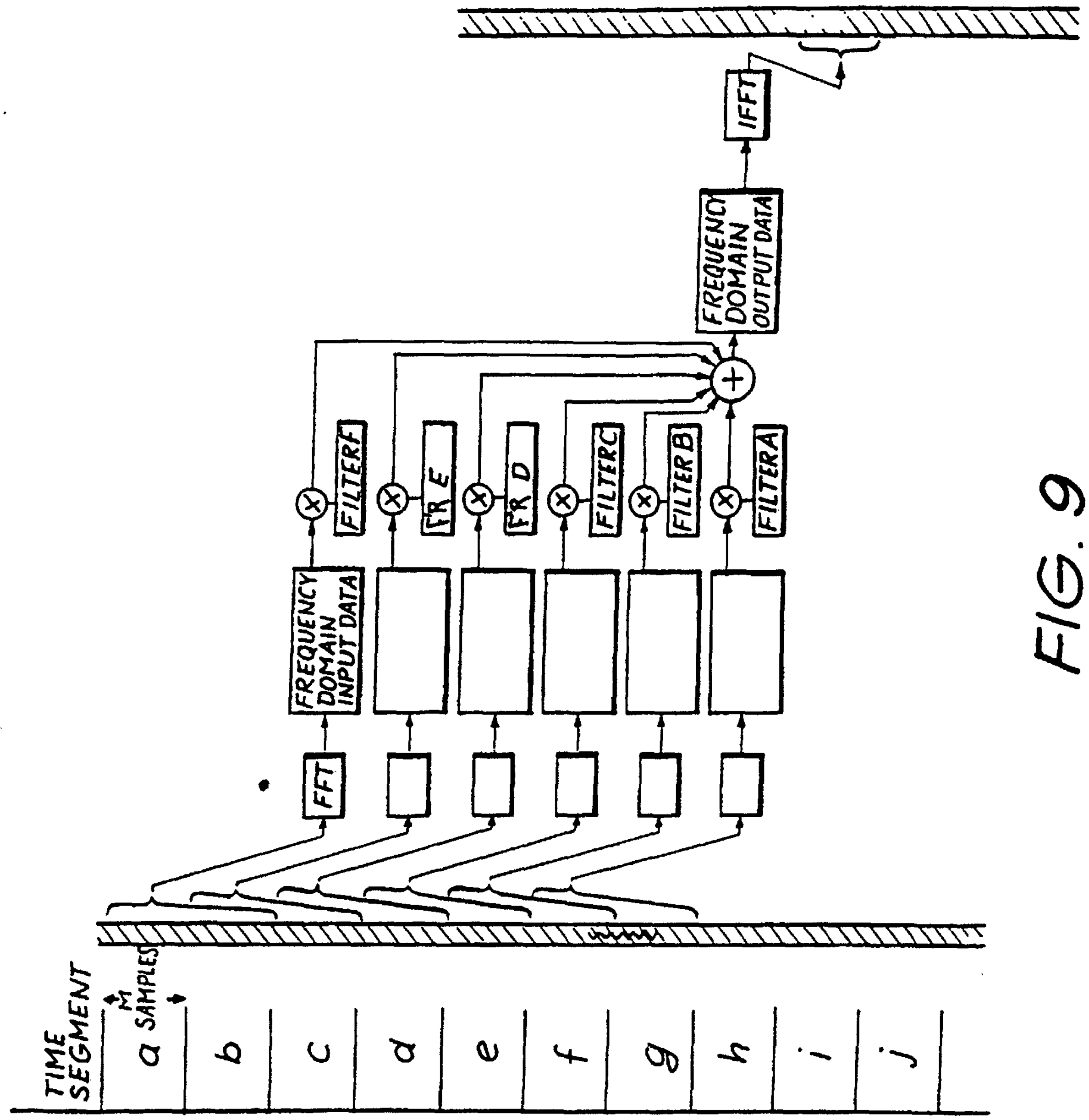

TIME SEGMENT
M SAMPLES
a
b
c
d
e
f
g
h
i
j
FFT
FREQUENCY DOMAIN INPUT DATA
FILTER F
FR E
FR D
FILTER C
FILTER B
FILTER A
FREQUENCY DOMAIN OUTPUT DATA
IFFT

FIG. 9

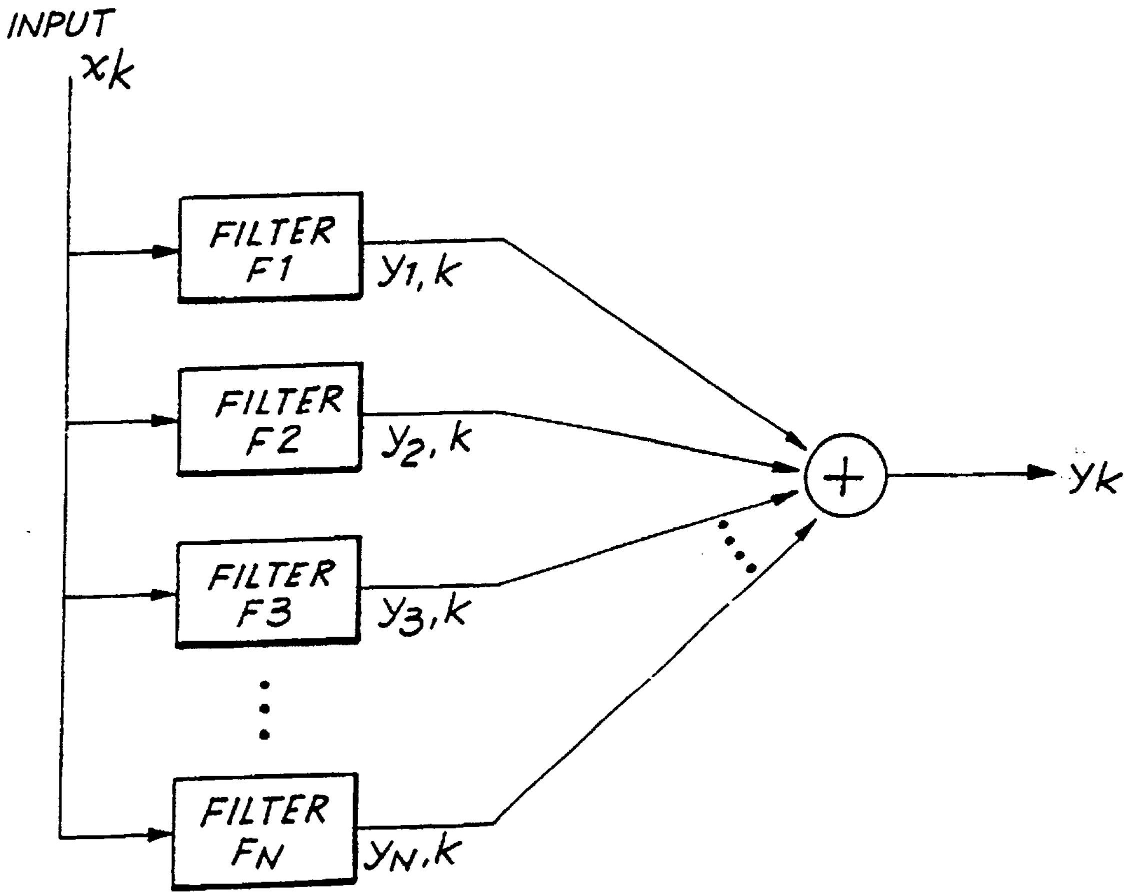

INPUT
xk
FILTER F1
y1, k
FILTER F2
y2, k
FILTER F3
y3, k
FILTER FN
yN, k
+
yk

FIG. 10

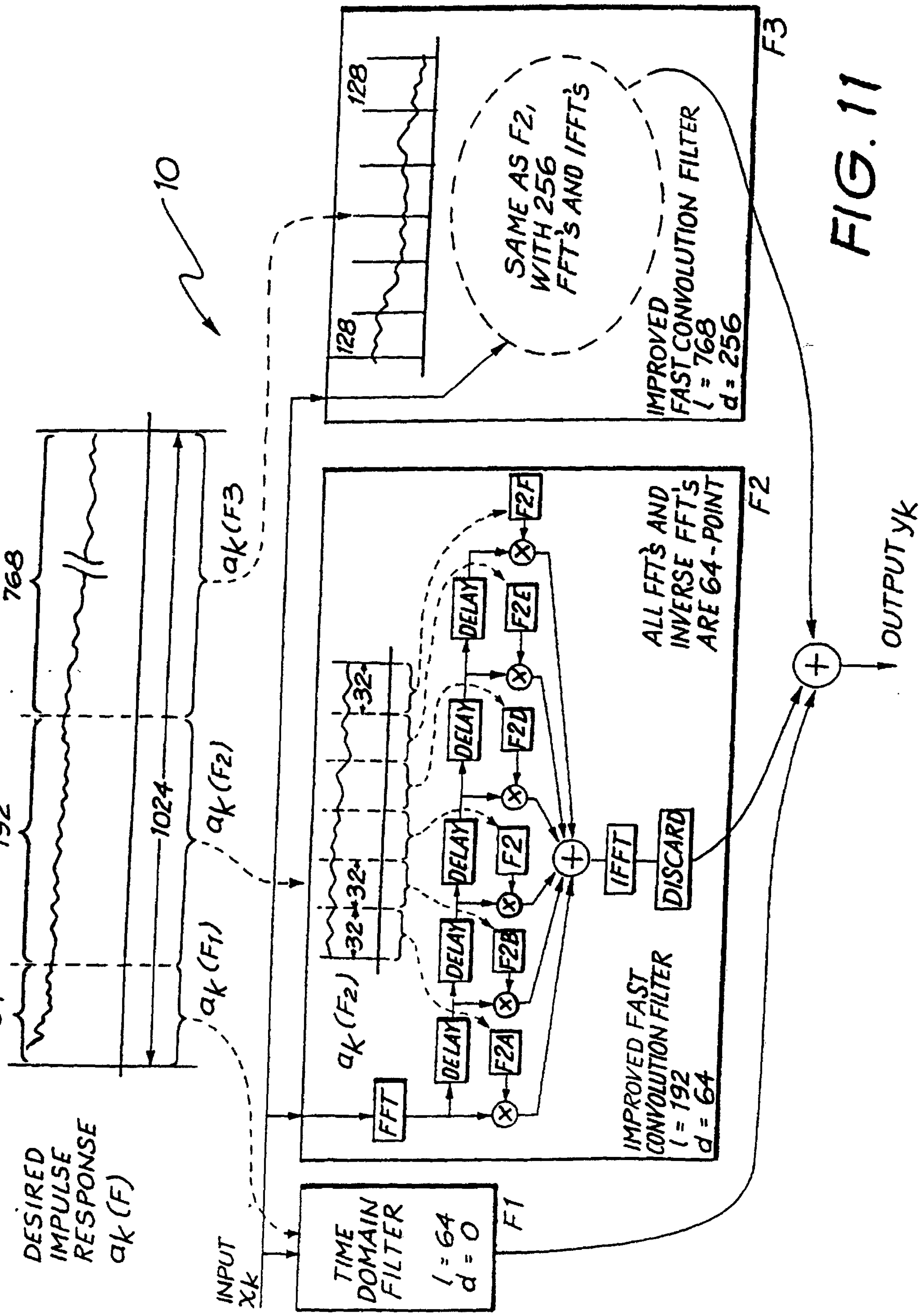

10
DESIRED IMPULSE RESPONSE ak (F)
64
192
768
1024
ak (F1)
ak (F2)
ak (F3
INPUT xk
TIME DOMAIN FILTER
l = 64
d = 0
F1
FFT
DELAY
F2A
F2B
F2
F2D
F2E
F2F
ak (F2)
32
IFFT
DISCARD
ALL FFT'S AND INVERSE FFT'S ARE 64-POINT
IMPROVED FAST CONVOLUTION FILTER
l = 192
d = 64
F2
128
SAME AS F2, WITH 256 FFT'S AND IFFT'S
IMPROVED FAST CONVOLUTION FILTER
l = 768
d = 256
F3
OUTPUT yk

FIG. 11

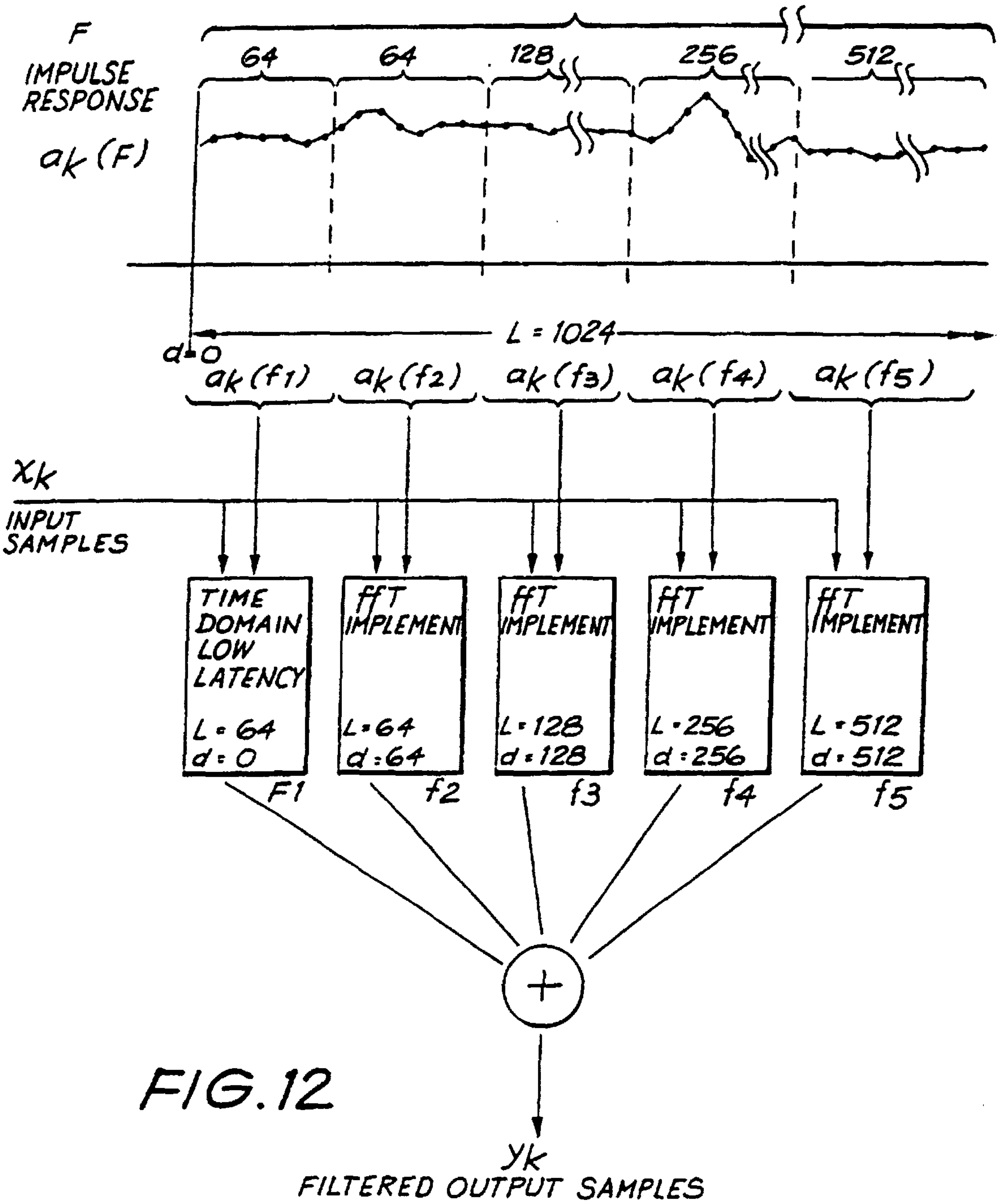
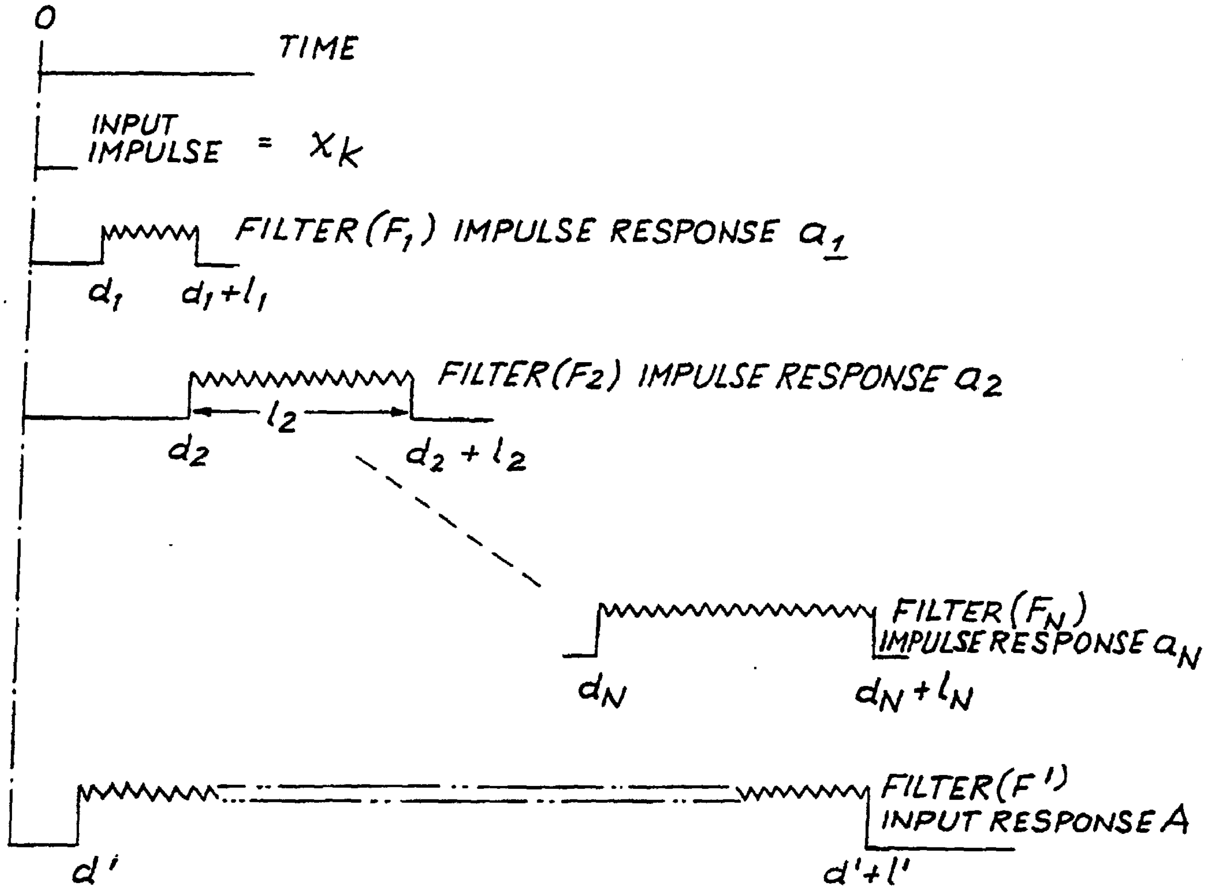

F
IMPULSE RESPONSE
ak (F)
64
64
128
256
512
L = 1024
d=0
ak (f1)
ak (f2)
ak (f3)
ak (f4)
ak (f5)
xk
INPUT SAMPLES
TIME DOMAIN LOW LATENCY
L = 64
d = 0
F1
FFT IMPLEMENT
L = 64
d = 64
f2
FFT IMPLEMENT
L = 128
d = 128
f3
FFT IMPLEMENT
L = 256
d = 256
f4
FFT IMPLEMENT
L = 512
d = 512
f5
+
yk
FILTERED OUTPUT SAMPLES

FIG.12



THE FILTER F' IS MADE BY SUMMING TOGETHER THE OUTPUTS OF FILTERS $F_1, F_2, \ldots\ldots, F_N$.

EACH FILTER F_i DEFINES OUTPUTS $y_{i,k}$ THE OUTPUT OF FILTER F_i AT TIME SAMPLE k, AS BEING

$$y_{i,k} = \sum_{n=d_i}^{d_i+l_i-1} a_{i,n}\, x_{k-n}$$

IF WE SET $y'_k = \sum_{n=1}^{N} y_{i,k}$

THEN WE HAVE $y'_k = \sum_{n=d'}^{d'+l'-1} A_n\, x_{k-n}$

FIG. 13

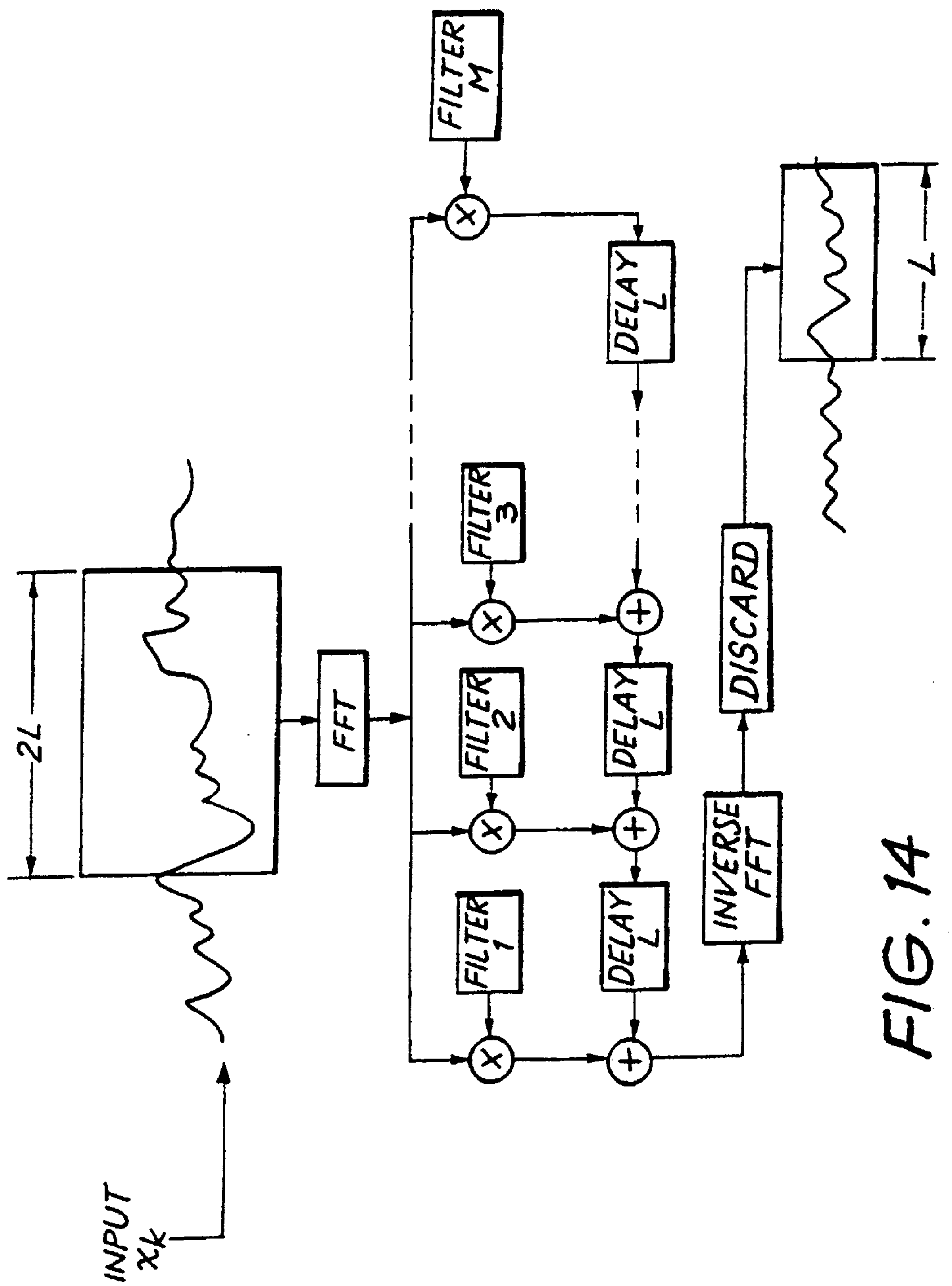

INPUT x_k
2L
FFT
FILTER 1
FILTER 2
FILTER 3
FILTER M
DELAY L
DELAY L
DELAY L
INVERSE FFT
DISCARD
L

FIG. 14

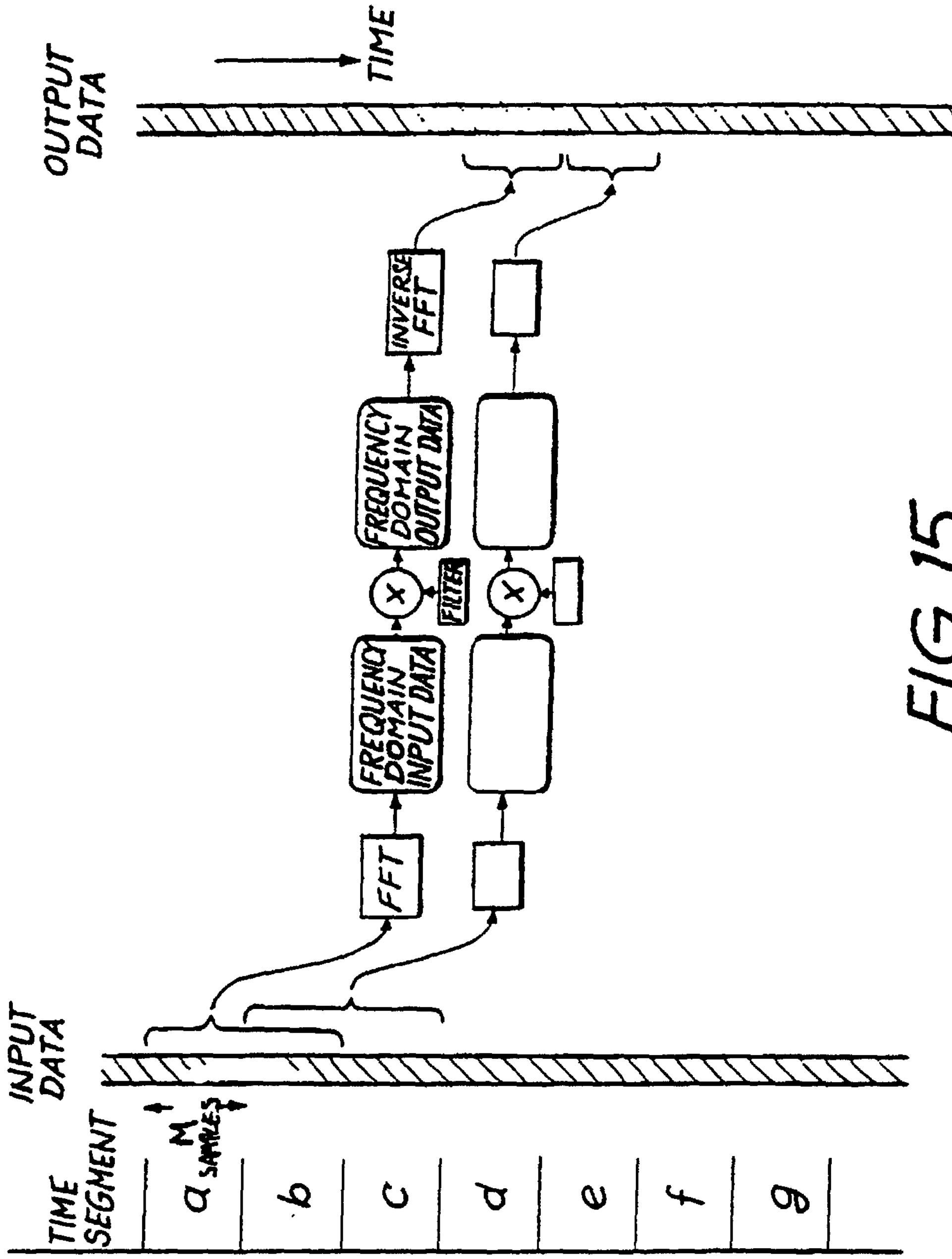

TIME SEGMENT
a
b
c
d
e
f
g
M SAMPLES
INPUT DATA
FFT
FREQUENCY DOMAIN INPUT DATA
x
FILTER
x
FREQUENCY DOMAIN OUTPUT DATA
INVERSE FFT
OUTPUT DATA
TIME

FIG. 15

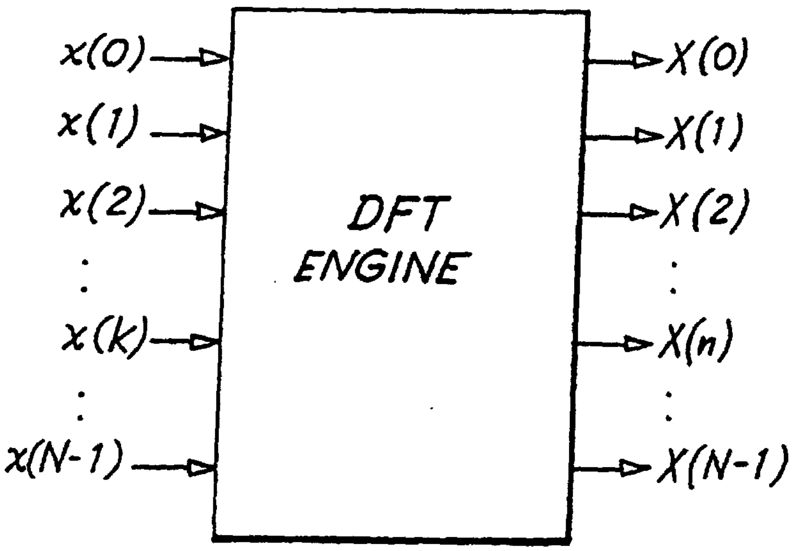


FIG. 16

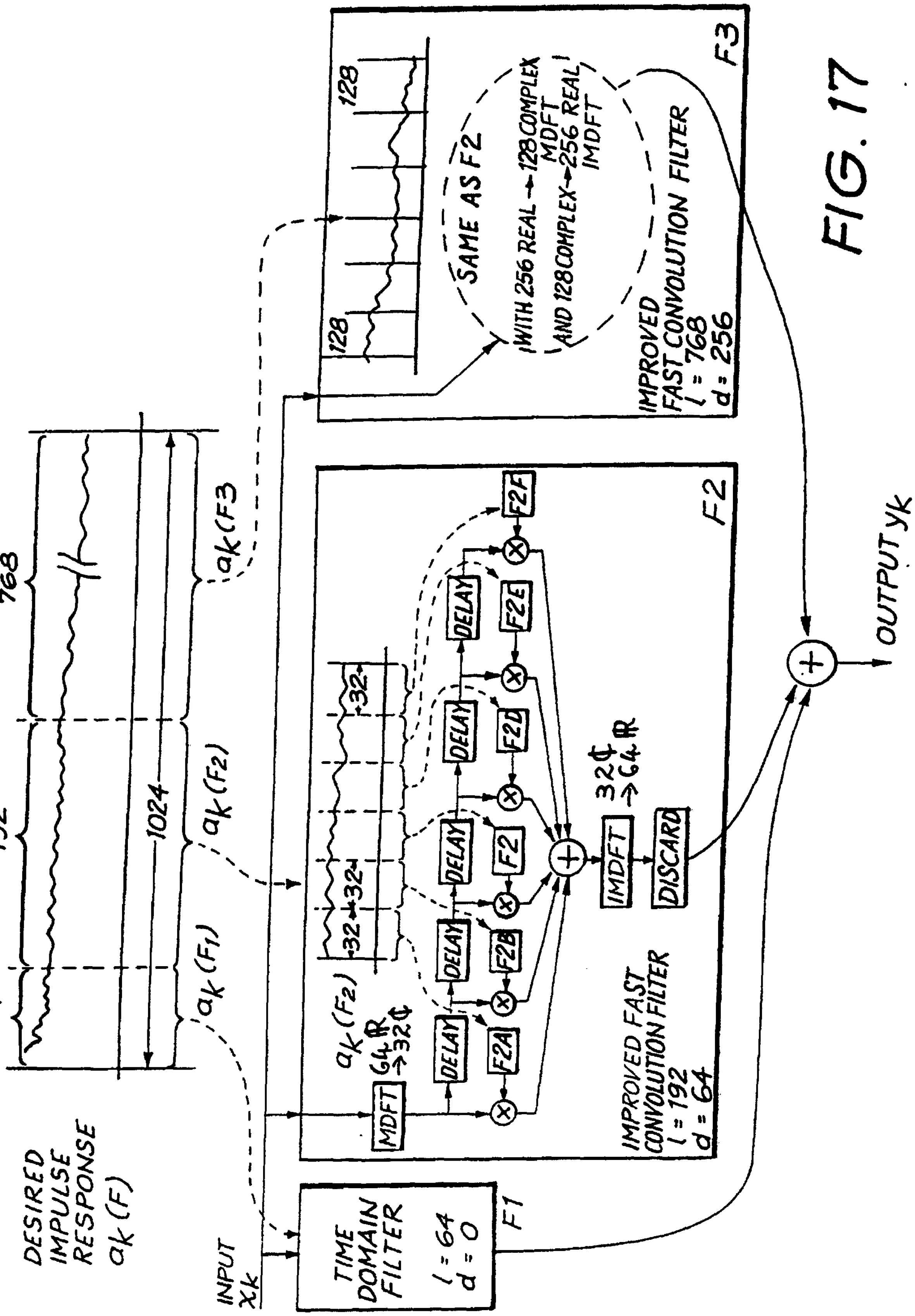

DESIRED IMPULSE RESPONSE ak (F)
64
192
768
1024
ak (F1)
ak (F2)
ak (F3)
INPUT Xk
TIME DOMAIN FILTER
l = 64
d = 0
F1
MDFT
ak (F2)
64 R → 32 ℂ
DELAY
F2A
F2B
F2
F2D
F2E
F2F
32
32 ℂ → 64 R
IMDFT
DISCARD
IMPROVED FAST CONVOLUTION FILTER
l = 192
d = 64
F2
128
SAME AS F2
WITH 256 REAL → 128 COMPLEX MDFT
AND 128 COMPLEX → 256 REAL IMDFT
IMPROVED FAST CONVOLUTION FILTER
l = 768
d = 256
F3
OUTPUT yk

FIG. 17