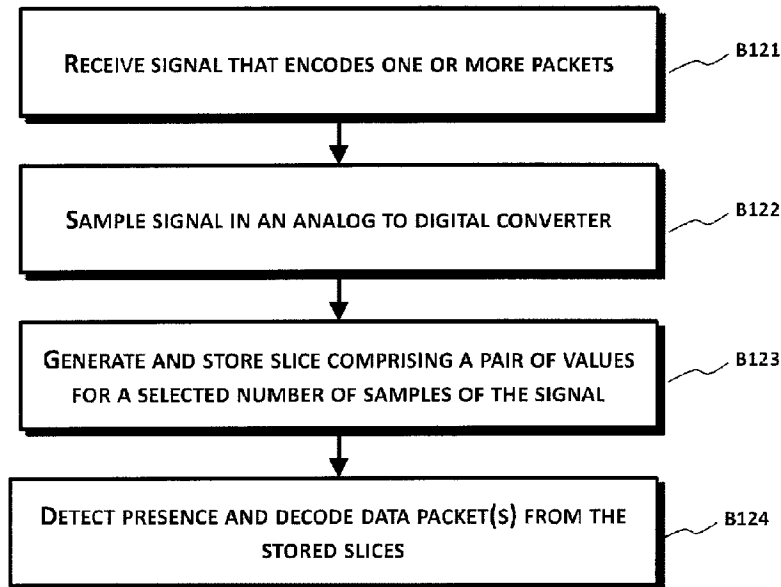




(22) Date de dépôt/Filing Date: 2014/09/19
(41) Mise à la disp. pub./Open to Public Insp.: 2015/03/26
(45) Date de délivrance/Issue Date: 2020/01/28
(62) Demande originale/Original Application: 2 924 815
(30) Priorité/Priority: 2013/09/20 (US61/880,786)

(51) Cl.Int./Int.Cl. *H04L 12/951* (2013.01),
H03D 7/00 (2006.01), *H04L 27/00* (2006.01),
H04L 27/144 (2006.01)
(72) Inventeurs/Inventors:
KUSHNER, CHERIE, US;
FLEMING, ROBERT, US;
MCALLISTER, WILLIAM H., US;
ZDEBLICK, MARK, US
(73) Propriétaire/Owner:
PROTEUS DIGITAL HEALTH, INC., US
(74) Agent: RIDOUT & MAYBEE LLP

(54) Titre : PROCÉDES, DISPOSITIFS ET SYSTÈMES DE RÉCEPTION ET DE DÉCODAGE DE SIGNAL EN PRÉSENCE DE BRUIT À L'AIDE DE TRANCHES ET D'UNE DISTORSION
(54) Title: METHODS, DEVICES AND SYSTEMS FOR RECEIVING AND DECODING A SIGNAL IN THE PRESENCE OF NOISE USING SLICES AND WARPING



(57) **Abrégé/Abstract:**

A method may comprise receiving and sampling a signal. The signal may encode a data packet. A slice may be generated and stored comprising a pair of values for each of a selected number of samples of the signal representing a correlation of the signal to reference functions in the receiver. The presence of the data packet may then be detected and the detected packet decoded from the stored slices. The generating and storing slices may be carried out as the received signal is sampled. The sampled values of the signal may be discarded as the slices are generated and stored. The slice representation of the signal can be manipulated to generate filters with flexible bandwidth and center frequency.

METHODS, DEVICES AND SYSTEMS FOR RECEIVING AND DECODING A SIGNAL IN THE PRESENCE OF NOISE USING SLICES AND WARPING

ABSTRACT OF THE DISCLOSURE

A method may comprise receiving and sampling a signal. The signal may encode a data packet. A slice may be generated and stored comprising a pair of values for each of a selected number of samples of the signal representing a correlation of the signal to reference functions in the receiver. The presence of the data packet may then be detected and the detected packet decoded from the stored slices. The generating and storing slices may be carried out as the received signal is sampled. The sampled values of the signal may be discarded as the slices are generated and stored. The slice representation of the signal can be manipulated to generate filters with flexible bandwidth and center frequency.

METHODS, DEVICES AND SYSTEMS FOR RECEIVING AND DECODING A SIGNAL IN THE PRESENCE OF NOISE USING SLICES AND WARPING

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application claims the benefit of US Provisional Application No. 61/880,786 titled "Methods, Devices and Systems for Receiving and Decoding a Signal in the Presence of Noise Using Slices and Warping," filed September 20, 2013.

INTRODUCTION

[0002] Ingestible sensors may comprise a low power communicator whose transmissions are received by a receiver that may be worn outside of the body. Conventional 'body communication systems' should be capable of processing high-speed raw data in a predetermined amount of time, with considerations to available power consumption and memory size. In a conventional receiver, the incoming signal passes through an 'analog front-end' circuit comprising analog filters and analog electronic amplifiers. The analog filter typically has a wide bandwidth, to allow for the detection of all possible transmitted frequencies, as determined by the manufacturing tolerance of the transmitter carrier frequency. The filtering provided in the analog front-end is modest, and allows a significant amount of noise to get through along with the desired signal. After analog amplification and filtering, the signal is digitized by an analog-to-digital converter (ADC). The remainder of the processing of the received signal may be carried out in digital hardware, such as an embedded microprocessor, state machine, logic gate array, among others. The now-digitized signal may pass through one or more narrow-band digital filters to remove as much noise as possible before decoding is attempted.

[0003] In cases in which the receiver's estimate of the carrier frequency has a significant amount of uncertainty, the receiver is required to start with a wider-bandwidth digital filter and to, therefore, admit a greater amount of noise. The greater amount of noise means that a weak signal may be missed entirely. To reject the most noise, however, the receiver may apply a digital filter with a narrow bandwidth. But, if the narrow filter is centered on the incorrect carrier frequency, the incoming signal may be missed entirely. For efficient detection and decoding of the incoming signal, therefore, a balance must be achieved between narrow-bandwidth filters to remove as much

noise as possible and filters having a greater bandwidth to increase the likelihood that the signal's carrier frequency will be captured when the receiver's knowledge of the incoming carrier frequency is imprecise. The receiver, therefore, may be configured to iteratively adjust the center frequency of the narrow filter, move it to a new center and to thereafter again attempt detection. This process of searching for the carrier with a narrow bandwidth filter is both time consuming and power intensive. Significantly, to re-filter at the new center frequency, the receiver either must retain a copy of the original data record in memory, or, if the original data is not available, capture an entirely new data record. This process not only requires significant memory resources (especially using high resolution ADCs) but also expends a significant amount of device battery life merely to identify the carrier frequency of the incoming signal.

SUMMARY OF THE INVENTION

[0004] The present invention in a further aspect provides a program. Such a program can be provided by itself or carried by a carrier medium. The carrier medium may be a recording or other storage medium. The transmission medium may be a signal.

[0005] According to one embodiment, a method may comprise receiving and sampling a signal. The signal may encode a data packet. A slice may be generated and stored comprising a pair of values for each of a selected number of samples of the signal. The presence of the data packet may then be detected and the detected packet decoded from the stored slices. The samples of the signal may represent a correlation of the signal to reference functions in the receiver. The generating and storing slices may be carried out as the received signal is sampled. The sampled values of the signal may be discarded as the slices are generated and stored. The slice representation of the signal can be manipulated to generate filters with flexible bandwidth and center frequency.

[0006] According to one embodiment, a method of detecting and decoding a signal arriving at a receiver may begin with the receiver receiving an incoming signal, optionally carrying out some analog pre-processing (e.g. amplifying and filtering) at an analog front-end, after which the pre-processed data may be sampled in an ADC. The sampled raw data, according to one embodiment, then may be compared against internal reference templates stored in memory, using, for example, a correlation algorithm. One exemplary technique comprises correlating the sampled incoming signal with predetermined reference templates over a time period.

[0007] Embodiments address the problems inherent in capturing and storing a great many high-speed samples, which strains both computational capability and memory size. Embodiments solve both problems by capturing “slices”. The slice data representation, according to one embodiment, contains sufficient information to efficiently and compactly represent the incoming signal and to implement filters of most any bandwidth. According to one embodiment, slices may be subject to a warping operation, by which sets of slices are transformed in useful ways to complete the detection process. Indeed, slices may be combined, according to one embodiment, to create filters having selectably wide or narrow pass-bands. According to embodiments, the warping operation may be configured to transform slices captured at one frequency to slices at another nearby frequency. This warping operation may be carried out by an algorithm configured to find an incoming carrier frequency and to find evidence of data packets in a noisy environment. The slice representation of signal data, coupled with the warping function, according to embodiments, represent a novel and highly efficient way to perform sophisticated detection algorithms with modest hardware and memory resources.

[0008] Further features of the present invention will become apparent from the following description of exemplary embodiments with reference to the attached drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

[0009] Fig. 1 shows various waveforms and an example slice, according to one embodiment. Fig. 1 also shows a system comprising a transmitter and a receiver configured according to one embodiment.

[0010] Fig. 2A illustrates correlation of two sampled waveforms.

[0011] Fig. 2B illustrates shows the manner in which one term (the sine term, in this case) is calculated, according to one embodiment.

[0012] Fig. 3 illustrates aspects of a method of calculating a combined slice term (the cosine term, in this case), according to one embodiment.

[0013] Fig. 4 shows aspects of a method of combining sine and cosine slice terms to form a longer correlation, according to one embodiment.

[0014] Fig. 5 shows the phase of a signal depicted as a rotating vector in a polar coordinate system.

[0015] Fig. 6A shows a rotating vector at a reference frequency in a polar coordinate system.

[0016] Fig. 6B shows a rotating vector at a reference frequency and a rotating vector of a signal at a frequency that is greater than the reference frequency, in a polar coordinate system.

[0017] Fig. 6C shows a rotating vector at a reference frequency and a rotating vector of a signal at a frequency that is lower than the reference frequency, in a polar coordinate system.

[0018] Fig. 7 shows aspects of warping, according to one embodiment.

[0019] Fig. 8 shows slices warped, aligned and ready for combination, according to one embodiment.

[0020] Fig. 9 shows aspects of a method for searching for a carrier frequency using warping of slices, according to one embodiment.

[0021] Fig. 10 shows aspects of Frequency Shift Keying (FSK) carrier detection, according to one embodiment.

[0022] Fig. 11 shows aspects of fine tuning FSK carrier detection, according to one embodiment.

[0023] Fig. 12 is a logic flow of a method of detecting a signal, according to one embodiment.

[0024] Fig. 13 is a logic flow of a method according to one embodiment.

DETAILED DESCRIPTION

[0025] Fig. 1 shows a system comprising a low-power oscillating transmitter 102 and a receiver 104, according to one embodiment. As shown therein, the oscillating transmitter 102 may be separated from the receiver 104 by a communication channel 103. For example, the oscillating transmitter 102 may be disposed within an ingestible sensor whose transmissions 105 are received by a receiver patch comprising the receiver 104 that may be worn outside of the body, such as on the skin 106. In this case, the communication channel 103 may comprise the aqueous environment of the body. The receiver 104 may comprise an analog front-end in which the received signal may be pre-processed, before being input to an ADC 110, which may generate a time-series of raw digital samples. The samples may be represented as binary numbers, from 1 to 24 bits in size, for

example. The receiver 104 also may comprise a controller 112, which may be coupled to a memory 114. The memory 114 may be configured to store, as detailed below, slice data, reference templates and other temporary values as needed by controller 112. The receiver may also comprise a communication interface (not shown), to enable decoded payload of packets encoded in the received signal to be communicated to the outside world.

[0026] According to one embodiment, a computer-implemented method of detecting and decoding a signal arriving at a receiver 104 may begin with the receiver 104 receiving an incoming signal 105, carrying out some analog pre-processing (e.g. amplifying and filtering) at analog front-end 108, after which the pre-processed data may be sampled in ADC 110. The sampled raw data, according to one embodiment, then may be compared by the controller 112 against internal reference templates stored in memory 114, using a correlation algorithm. One technique comprises correlating the sampled incoming signal with predetermined reference templates over a time period.

[0027] Embodiments address the problems inherent in capturing and storing a great many high-speed samples, which strains both computational capability and memory size. Embodiments solve both problems by capturing “slices”. The slice data representation, according to one embodiment, contains sufficient information to efficiently and compactly represent the incoming signal and to implement filters of most any bandwidth. According to one embodiment, slices may be subject to a warping operation, by which sets of slices are transformed in useful ways to complete the detection process. Indeed, slices may be combined, according to one embodiment, to create filters having selectably wide or narrow pass-bands. According to embodiments, the warping operation may be configured to transform slices captured at one frequency to slices at another nearby frequency. This warping operation may be carried out by an algorithm configured to find an incoming carrier frequency and to find evidence of data packets in a noisy environment. The slice representation of signal data, coupled with the warping function, according to embodiments, represent a novel and highly efficient way to perform sophisticated detection algorithms with modest hardware and memory resources. For example, one or more microcontrollers, one or more Field Programmable Gate Arrays (FPGAs) or Application Specific Integrated Circuits (ASICs) may be used to carry out the processing disclosed herein. A Digital Signal Processor (DSP) may also be used to good advantage.

[0028] SLICE: According to one embodiment, a slice construct is introduced. Short correlations, achieved through correlating a relatively short portion of the incoming signal (e.g. approximately 4 - 8 cycles), are denoted as slices herein. A slice interval, according to one embodiment, may be defined as a predetermined period of time. Fig. 1 shows various segments of a 20,000 Hz signal. As shown, reference 102 shows a single cycle of such a 20,000 Hz signal, whose period T is 1/20,000 Hz or 50µsec. Reference 104 shows a single slice interval, defined as a time equal to 4 cycles of the 20,000 Hz signal, or 200 µsec. Herein, a slice interval is arbitrarily defined as 4 cycles of the incoming signal. A slice interval, however, may comprise a different amount of time or number of cycles. For example, a slice interval may comprise the time equal to 8 cycles. Below, unless specifically noted, a slice interval is defined as comprising 4 cycles of the incoming signal, it being understood that other slice intervals may readily be implemented. For example, the slice definition may be expressed in cycles, but is not required to be a multiple of full cycles of any signal or template. A slice may be any defined amount of time. The slice time may be changed in the receiver as needed. For example, the receiver could implement two slice routines to capture two slice streams simultaneously, one at 20 kHz and another at 12.5 kHz, for example. The two slice computations could use different slice times suitable for each channel. As shown at 106 in Fig. 1, four slice intervals may comprise 16 cycles and have a period of 800 µsec. Lastly, 64 cycles of the reference frequency may be divided into 16 slice intervals as shown at 108. The number of samples of the incoming signal included in one slice is governed by the definition of the slice interval and the sampling rate of the ADC:

$$\text{samples per slice} = \text{ADC sample rate} \cdot \text{slice interval.}$$

[0029] The ADC sampling rate may be at least as often as the Nyquist theorem call for; namely, at least twice the frequency of interest. According to one embodiment, the ADC sample rate may be chosen to be higher, such as five or more times the frequency of interest of the incoming signal. Other sampling rates may be utilized. In one embodiment, the ADC in the receiver (adhered to a patient's abdomen, for example) may be configured to carry out forty or more samples per second. The starting times of consecutive slices may advantageously be selected to be periodic according to some fixed, for example, interval. However, acceptable results may also be obtained even when there are brief periods of time when no sampling is being carried out.

[0030] To determine the similarity between digitized samples of an incoming signal and a

reference template, a dot product (the sum of the products of corresponding samples) or correlation operation may be carried out. Fig. 2A shows such a correlation operation of a digitized incoming signal with a cosine template. Here, A may represent the digitized incoming signal and B may represent a template of a first reference function such as, for example, a cosine template at the reference frequency (e.g. 20,000 Hz). In other words, the cosine template B, according to one embodiment, is a representation of what the receiver 104 expects the cosine component of the received signal to look like and the correlation operation determines the degree of similarity between signal A and cosine template B. As shown, samples of signal A are multiplied with the corresponding samples of the cosine template B, and the results of these additions summed over the number of samples N. Stated more formally, C is the scalar product of A and B and may be expressed as:

$$C = A_1 \times B_1 + A_2 \times B_2 + A_3 \times B_3 + \dots + A_N \times B_N$$

$$C = \sum_{n=1}^N A_n \times B_n$$

[0031] Similarly, Fig. 2B shows correlation with a sine template. Here, A may represent the digitized incoming signal and D may represent a template of a second reference function in quadrature with the first reference function. For example, the template of the second reference function may be, for example, a sine template at the reference frequency (e.g. 20,000 Hz). As shown, samples of signal A are multiplied with the corresponding samples of the sine template D, and the results of these additions summed over the number of samples N. Stated more formally, S is the scalar product of A and D and may be expressed as:

$$S = A_1 \times D_1 + A_2 \times D_2 + \dots + A_N \times D_N$$

$$S = \sum_{n=1}^N A_n \times D_n$$

The orthogonal cosine and sine templates are in a quadrature phase relationship. The two

correlation results, C and S, when taken together, represent a slice. In complex polar notation, $C + j \cdot S$ is a vector with an angle indicating the phase between the incoming signal and the receiver's reference templates. In practice, the slice may be thought of as a $1/(\text{slice interval})$ filter.

[0032] According to one embodiment, the scalars C and S may be scaled by a scaling factor. For example, C and S may be scaled such that they may assume a range of values between, for example, 0 and 1. Other scaling factors and ranges may be accommodated.

[0033] As shown and discussed herein, the reference templates are sine templates and cosine templates. Other periodic shapes, however, may be used as the reference templates such as, for example, sawtooth, triangle or square signals. Selecting non-sinusoidal waveforms for the reference templates may result in some information being discarded, but the signal of interest may still be extracted from the received signal. Moreover, even though having the reference templates 90 degrees out of phase with one another (in quadrature), reference templates having other phase relationships with one another may be used. For example, the two reference templates could be 89 degrees or 91 degrees out of phase with one another, without substantial ill effect.

[0034] According to one embodiment, slice correlations (or, simply, slices) may be calculated from the raw digitized samples generated by the receiver's ADC 110. These raw digitized samples may be correlated against samples of both cosine and sine reference templates at the reference frequency (freqRef) stored in the receiver 104. The cosine term and sine term of a slice, according to one embodiment, may be defined as:

$$SliceCosTerm = \sum_{n=1}^N signal_n \times referenceCos_n$$

$$SliceSinTerm = \sum_{n=1}^N signal_n \times referenceSin_n$$

where N is the number of samples in one slice.

The vector magnitude of a slice may be computed in Root Mean Square (RMS) fashion:

$$Slice\ Magnitude = \sqrt{SliceCosTerm^2 + SliceSinTerm^2}$$

The Slice Magnitude quantity is a scalar indicative of the magnitude of the combined slices.

The vector angle of a slice thereof (Slice Angle), is given by

$$\text{Slice Angle} = \arctan\left(\frac{\text{SliceSinTerm}}{\text{SliceCosTerm}}\right)$$

[0035] COMBINING SLICES: Fig. 3 is a diagram showing the scalar dot product of signal A and template B over two slice intervals (where, in this figure, the slice interval encompasses one cycle of the cosine template), and shows the additive nature of the correlation. According to one embodiment, for slices to be combinable, each of the reference signals of each reference template should be coherent, meaning in phase with one another. As shown, the correlation or dot product of A and B over two slice intervals (2N samples, in this case) corresponds to the simple scalar sum (accumulation) of the correlation over the first N cycles with the correlation over the second N cycles of A and B. Or,

$$C_1 = \sum_{n=1}^N A_n \times B_n$$

$$C_2 = \sum_{n=N+1}^{2N} A_n \times B_n$$

$$C_{12} = \sum_{n=1}^{2N} A_n \times B_n \quad C_{12} = C_1 + C_2$$

[0036] Moreover, to compute the correlation for a time interval corresponding to 3 slice intervals of A and B, it is not necessary to re-compute C1 and C2. Simply, compute the correlation C3, and add the result to C12 to generate the correlation (dot product of vectors A and B over a signal length of 3 slice intervals) C13. As a slice is equivalent to a 1 / (slice interval) filter, as slices are combined into longer correlations, the filter bandwidth is correspondingly reduced, as further detailed below.

[0037] According to one embodiment, slices are treated as complex pairs, comprising both a cosine term and a sine term. The cosine term of a slice, according to one embodiment, represents the correlation between the sampled incoming signal and a cosine template stored in the receiver 104 at the reference frequency (freqRef). Similarly, the sine term of a slice, according to one

embodiment, represents the correlation between the sampled incoming signal and a sine template stored in the receiver 104 at freqRef. FreqRef can be set to the expected or nominal frequency at which the transmitter is specified to transmit, but which may vary due to manufacturing variations (which may occur in both the transmitter and the receiver), ambient conditions such as the temperature of the transmitter and receiver, distortion through the communication channel (e.g. the aqueous and physiologic environments of the human body such as the salinity of the stomach and surrounding tissues). Other factors may include, for example, variations in the frequency calibration process used on the transmitter and receiver, which may not be very accurate, or might have large frequency steps in their adjustment method.

[0038] According to embodiments, once the slice calculations have been carried out and the slice terms stored in memory 114, the original raw samples generated by the ADC (and from which the slices were generated) now may be discarded, as all subsequent packet detecting, frequency determination and payload decoding steps may be based on the stored slice data, without the need to ever consult or re-generate the digitized samples generated by the ADC. According to embodiments, the slice calculation and the storage of the slice data in memory 114 may be carried out 'on-the-fly' in real time by a suitable controller provided within the receiver 104. According to one embodiment, the slice correlation data may be calculated and stored in memory 114 by the receiver's controller 112 in the controller command execution cycles available between ADC sample times. Accordingly, there may be no need to store the raw digitized sample stream from the ADC 110 in memory 114, which represents a significant efficiency.

[0039] According to embodiments, significant reductions in the amount of data stored by the receiver 104 may be achieved. For example, the reference frequency of the carrier may be 20,000 Hz and the sample rate of the ADC may be 3.2 million samples per second (SPS), which corresponds to 160 ADC samples per cycle of the carrier. The sample rate of the ADC, however, may be freely chosen. For example, the sample rate of the ADC may be selected to be in the thousands of samples per second. For example, the sample rate of the ADC may be chosen to be about 200 kSPS, which corresponds to 10 ADC samples per cycle of the carrier. A controller 112 may be configured to execute, for example, 16 million instructions per second. If a slice interval were to be defined as 4 cycles of the reference frequency, at a sample rate of 200 kSPS, there are $10 \cdot 4$ or 40 ADC samples in each slice. There are $16,000,000 / 20,000$ or 80 processor cycles available between each ADC sample, which is generally sufficient to generate and store the slice

record. According to one embodiment, each individual new sample may be incorporated into the accumulating slice cosine and sine dot products and stored within these available processor cycles, thereby enabling the controller 112 to generate the slice data while keeping pace with the samples as they are generated by the ADC. The result of the slice correlation calculation is two numbers (a cosine term and a sine term), which represents a compression, per slice (e.g. 4 cycles of the incoming signal) of 40:2 or a compression factor of 20 relative to the raw sample stream. In this particular example, this represents over an order of magnitude reduction in memory requirements. Increasing the slice time or increasing the sampling rate linearly increases this compression rate. In one embodiment, a sampling rate of 760 kSPS allows for 21 processor cycles between samples, which is sufficient computational power to generate slice data while keeping pace with the samples as they arrive. Each cycle is represented by $760 / 20$ or 38 samples, so each slice represents $4 \cdot 38$ or 152 samples of the incoming signal. The resulting compression factor is 152:2 or a compression factor of 71.

[0040] ANALOG SLICE PROCESSING – According to one embodiment, the incoming signal may be multiplied by two analog multipliers (e.g. quadrature mixers) with two reference signals. Each of the product signals may then be summed (e.g. by analog integration using a capacitor or an active circuit based on stored capacitor charge) for a period of time and then sampled at a much lower frequency. Each such sample pair represents a slice pair. Such an analog embodiment may enable power consumption advantages to be realized.

[0041] COMBINING SLICES, FILTERING – Effectively, the slice correlation calculation represents a filter with a bandwidth of $1/(\text{slice interval})$ which, in the example case of a reference frequency of 20,000 Hz and 4 cycles per slice, works out to $1/200\mu\text{sec}$ or 5,000Hz, which is a filter having a relatively broad bandwidth. According to one embodiment, the constituent cosine components of the slice pair may be combined and the constituent sine components of the slice pair may be combined, thereby increasing the slice time and creating a filter having a narrower bandwidth. Due to the inverse relationship between slice interval and filter bandwidth, according to one embodiment, a narrower bandwidth filter may be achieved through combining slice terms. Indeed, slice correlations computed over short periods of time may be extended to longer correlations by combining such short periods of time; that is, by combining slices. Combining slice terms, according to one embodiment, may be carried out by summing a number of sequential cosine slice terms, summing the same number of sequential sine slice terms. The resulting two

new terms, when paired together form a combined slice representing a longer correlation.

[0042] According to embodiments, such a slice combination calculation may be performed at every slice index (i.e., without skipping to every N^{th} slice index). Fig. 4 is a graphical representation of combining previously-computed and stored slice pairs of cosine and sine components. As shown, the original cosine components of the stored slice data are labeled as “original slice cosine terms” and the original sine components of the slice data are labeled “original slice sine terms”. To combine four slices, the first four cosine terms ($i=1, 2, 3, 4$) are summed into a “combined slice cosine term” with slice index 1. Likewise, the first four sine components of the slice data are summed into a “combined slice sine term”, starting with the current index 1. Therefore, on the first iteration, $i = 1$ and the previously computed cosine terms indexed at $i=1, i=2, i=3$ and $i=4$ are summed to form SliceCosTerm_1 , and the previously computed sine terms indexed at $i=1, i=2, i=3$ and $i=4$ are combined to form SliceSinTerm_1 , whereupon i is incremented to 2. SliceCosTerm_2 may then be formed by the four consecutive slice cosine terms, starting with the current $i=2$ slice index; namely, $i=2, i=3, i=4$ and $i=5$. Likewise, SliceSinTerm_2 may then be formed by a similar computation. This operation may be carried out for the entire slice record. By varying the number of slices over which the combining is carried out, the bandwidth of the resultant filter may be selected at will. This ability to rapidly and simply generate different filters is a generally useful capability in a receiver. By way of a simple example, when the receiver 104 is searching for the carrier frequency of the received signal, a small number of slice cosine and sine terms may be combined to generate what is, in effect, a filter having a relatively wide bandwidth, thereby increasing the probability that the carrier will be present somewhere within the frequency range encompassed by the wide bandwidth filter. However, such a wide bandwidth filter also admits a correspondingly large amount of noise, which may render detection of especially weak signals difficult. Alternatively, a larger number of slice terms may be combined to generate what is, in effect, a filter having a correspondingly narrow bandwidth. Such a narrow bandwidth filter, however, does not admit a large amount of noise, which may facilitate the detection of the carrier frequency.

[0043] According to embodiments, one result of combining slices is a digital filter having reduced bandwidth, while maintaining the time resolution of the original slices. It is to be noted that such filters may be constructed using only the slice data stored in memory 114, as the original raw ADC data may have already been discarded and may be, therefore unavailable. According to

embodiments, slice combinations over a greater number of slices may be implemented. Moreover, slice combinations may be repeatedly performed over different numbers of slices (hence implementing filters of different bandwidths) using the original slice data or using previously combined slice records, without re-referencing the original raw ADC samples (which may have been previously discarded anyway) and without re-acquiring the incoming signal and re-generating new raw ADC samples. Because of the high level of compression represented by slice data (i.e., over an order of magnitude in the example being developed herewith), long recordings of slice data may be stored in, for example, controller memory, even in the face of strict memory size constraints. The memory 114 shown in Fig. 1 may be external to the controller 112 or internal thereto.

[0044] According to one embodiment, one need not combine slices if the original slice interval is defined to be as long a period of time as a combined slice would be, had the slices been combined. For example, the slice interval may be defined to be longer than 4 cycles, which is the exemplary implementation discussed herein. This may be desirable in systems in which there is good crystal control of the transmitter and the receiver. In such cases, warping (as discussed herein below) need be carried out over only a narrow frequency range to find the carrier frequency and/or to detect the presence of a packet in a noisy environment. Therefore, according to one embodiment, the originally-captured set of slices may be used to form a filter, without the need to combine slices as described herein.

[0045] As the slice combining calculations described and shown herein are largely composed of additions, such combining calculations may be carried out efficiently. Also, as the slice combining operation may operate only on the indexed slice cosine and sine terms stored in memory 114, the combining operation need not be carried out in real time, as the raw samples arrive, as it may be carried out after all slice pairs have been generated from the raw ADC samples of the incoming signal and stored in memory 114. Moreover, as the combining operations do not, according to one embodiment, alter the stored indexed slice pairs, the slice combining operations may be repeated any number of times, depending on the needs of the overall detection and decoding algorithms. That is, the original slice data may be reused many times at will. Alternatively, the slice combining operation may be performed on slices that themselves are the result of a combining operation. For example, a combination of four slices (a '4-slice' slice record) may be achieved either by 1) Combining four original slices to generate a 4-slice slice record, or

2) Combining two original slices into a 2-slice slice record, and then combine two slices from the 2-slice slice record to generate the desired 4-slice record. Such flexibility can be exploited to, for example, conserve memory in the processor.

[0046] SUMMARY: SLICE AND SLICE COMBINING – To review the slice representation up to this point in the discussion, an incoming signal can be captured by a sequence of short correlations against reference templates. The templates may comprise a first reference function and a second reference function. According to one embodiment, the first and second reference functions are in quadrature. For example, the first reference function may be or comprise a cosine function and the second reference function may be or comprise a sine function. The length of the correlation may be conveniently selected to be a few periods (or more) of the template functions. The result of a correlation is two scalar terms that can be thought of as representing a complex number: $\text{costerm} + j \cdot \text{sinterm}$. Each correlation result is herein referred to as a slice, and a number of slices are captured in memory in a slice record. One operation that may be applied to a slice record is slice combination as described above. Combining slices is performed with simple additions of the individual slice terms. Combining slices results in a new slice record representing a filter of narrower bandwidth than the original slice record. This capability is highly useful in receiving and filtering a signal embedded in noise.

[0047] To this point in the discussion, the center frequency of the combined-slice narrow-band filter is the frequency of the reference template functions. This choice of only a single center frequency is a significant limitation to the slice capture and slice combining operations described to this point. The following sections describe a method, according to one embodiment, to move the slice record to any nearby frequency, thereby significantly increasing the utility of the slice representation.

[0048] WARP – An important function in any signal processing device is the ability to respond to variations in the transmitted signal frequency. For systems capturing a signal in the slice representation described above, the same need applies. After capturing a signal in slice form using correlation to reference templates, it may be desirable to create filters at a frequency other than the reference frequency, (e. g. at a frequency freqRef plus a frequency delta (freqDelta). The frequency delta may be either a positive or negative offset from freqRef . According to one embodiment, such a new narrow-band filter centered at $\text{freqRef} + \text{freqDelta}$ may be created by a)

capturing slice records at a reference frequency (freqRef), b) transforming (also denoted as “warping” herein) the original slice record into a new warped slice record using a complex vector rotation operation in which the rotation angle is governed or determined by a so-called warping function (WF), and c) combining the warped slices to generate a narrow-band filter now centered at frequency $\text{freqRef} + \text{freqDelta}$.

[0049] One embodiment, therefore, enables slice data taken at one frequency (e.g. freqRef) to be warped to slice data at another frequency, say $\text{freqRef} + \text{freqDelta}$. This may be carried out, according to one embodiment, without acquiring new data and without the need to re-use the original samples generated from the ADC 110 at the analog front end of the receiver 104, as such original data stream may be discarded – or may simply never be stored. According to one embodiment, therefore, a warping method may be configured to shift the center frequency of a digital filter without re-acquiring new data and without re-using the original samples generated by the ADC 110 to which the (processed) incoming signal is input.

[0050] POLAR NOTATION – Fig. 5 shows a vector 504 of length 1 in a polar coordinate system 502. As shown, any point in the polar coordinate system 502 may be represented as a complex pair, namely (x, y) . Equivalently, any point in the polar coordinate system 502 can be represented by a magnitude 504 and angle, (r, θ) where θ (505) is the angle of the vector 504 relative to the positive x-axis. Points z in the complex plane may be defined as those points satisfying the equation $z = r \cos \theta + j \cdot r \sin \theta$. The coordinates of any point comprises both a cosine term: $r \cos \theta$ (508) and a sine term: $r \sin \theta$ (506).

[0051] As shown in Fig. 6A, a reference frequency freqRef , such as the frequency of a reference template used in a correlation operation, may be represented as a rotating vector in a polar coordinate system. Ideally, the frequency of a signal received by a receiver would be exactly the same frequency that was transmitted, the reference frequency. Practically, however, such is not often the case. The frequency of the received incoming signal may be higher than that of the reference frequency freqRef . In that case, using the rotating vector representation of Figs. 5, the vector representing the incoming signal would lead (rotate faster than) the vector representing the reference frequency freqRef , as shown in Fig. 6B. Similarly, the frequency of the received incoming signal may be lower than that of the reference frequency freqRef . In that case, the vector representing the incoming signal would lag (rotate slower than) the vector representing the

reference frequency $freqRef$, as shown in Fig. 6C.

[0052] In the example of Fig. 7, the incoming signal is shown as a higher frequency than the reference frequency. With reference to Fig. 7, a polar coordinate system is illustrated, with the x-axis corresponding to the cosine term and the y-axis corresponding to the sine term. A reference signal ($freqRef$, solid line) is shown, by convention, as a vector pointing along the positive x-axis (cosine axis). Slice data generated from an incoming signal are shown as dashed vectors representing slices 1, 2, 3, 4, etc. In this static representation, it can be seen that the vector representing the first slice establishes an arbitrary (0 to 2π radian) phase angle α with respect to the reference frequency vector. In this example, the subsequent slice vectors having slice indices 2, 3, 4, etc., lead (i.e., rotate faster than) the reference vector by an ever-increasing angle. The observation central to the warping concept is that the angle for each successive slice increases by a constant angle for all slices, Φ . That is, the second slice vector is at an angle Φ relative to the first slice vector, the third slice vector is at an angle of Φ relative the second slice vector, or equivalently 2Φ relative to the first slice vector, and the fourth slice is located at an angle Φ relative to the third slice, or equivalently 3Φ relative to the first slice vector. The angle Φ , and multiples thereof, therefore, may be thought of as the amount of lead or lag from slice to slice, and multiples thereof represent the amount of lead or lag with respect to the reference vector. Fig. 7 demonstrates that for an incoming signal frequency that does not perfectly match the reference frequency, the slice data becomes more and more out of phase (leads or lags) with the reference vector as the slice number increases. Even a very small initial angle Φ tends to grow such that the slices become significantly out of phase over time. The angle Φ is proportional to the ratio of $freqDelta$ (the frequency difference between the incoming signal and the reference templates in the receiver) to the frequency of the reference templates, $freqRef$. The angle Φ is also proportional to the slice interval. According to one embodiment, the angle Φ in radians may be defined as

$$\Phi = 2\pi \left(\frac{freqDelta}{freqRef} \right) \cdot \text{cycles per slice}$$

where $freqDelta$ is the difference between the frequency of the incoming signal ($freqSignal$) and the frequency of the reference signal ($freqRef$),

$$freqDelta = freqSignal - freqRef$$

[0053] For a signal with a constant frequency, the angular shift between slices is consistent across slices. As graphically seen in Fig. 7, the amount of rotation for successive slices is not a constant angle with respect to the reference. Rather, the angle by which each successive slice is shifted, relative to the first slice is, in this illustrative example, an integer multiple of the angle Φ .

[0054] VECTOR ROTATION – The general form of a complex vector rotation by an angle θ can be represented in matrix form as:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Where x and y are the original vector coordinates and θ is the rotation angle, with positive rotation in the counterclockwise direction. The resulting rotated vector coordinates are x' and y' . In algebraic form, the rotation operation can be expressed by two equations:

$$x' = x \cos \theta - y \sin \theta$$

$$y' = x \sin \theta + y \cos \theta$$

The operation may be represented informally as

$$\text{rotated vector} = \text{VectorRotate}(\text{input vector}, \text{angle})$$

In slice notation, costerm plays the role of the x value, and sinterm plays the role of the y value.

[0055] WARP FUNCTION – A complex representation allows slices to be displayed as vectors on a complex polar plane. Complex vector notation is a convenient way to illustrate warping operations in the following description of the so-called warp function (WF). Slices may be represented as complex pairs; namely, $\text{costerm} + j \cdot \text{sinterm}$. According to one embodiment, the manner in which slice data are operated upon may be characterized as vector rotation where the rotation angle is determined by a Warp Function (WF). Warping of a slice record may be the result of a complex vector rotation operation (say, VectorRotate), which takes two arguments: the input slice data record (denoted Input Slice below) and a rotation angle (determined by the output of a Warp Function) to which each slice in the slice data record is to be rotated. Stated more succinctly, the generalized warping operation may be described as:

$$\text{Warped Slice}(i) = \text{VectorRotate}(\text{Input Slice}(i), \text{WF}(\theta, i, \text{other arguments}))$$

Where i runs from 1 to the number of slices in the slice record. The rotation angle is derived from

a warp function,

$$\text{angle}(i) = \text{WF}(\theta, i, \text{other arguments})$$

[0056] In various embodiments, the selection of the warp function WF and the angle θ in the equation determines the properties of the resulting warped slice record.

[0057] WARP FUNCTION EXAMPLES - This section describes a number of warp functions, from a simple case to a more complex case from which several useful definitions may be derived.

[0058] Beginning with a relatively simple example, the warp function may be defined as $\text{WF}() = 1 \cdot \theta$. Applying this warp function to the slice record results in the entire slice record being shifted by a constant phase angle θ . In the polar coordinate diagram of Fig. 5, this warp function corresponds to rotating all slice vectors by the same amount, θ . In the time domain, the constant phase shift advances or delays the incoming signal with respect to the receiver's reference templates, without otherwise altering the properties of the signal.

[0059] WARPING TO TUNE SLICES TO A NEW CENTER FREQUENCY – In one embodiment, the warp function may be defined as

$$\text{WF}() = -i \cdot \Phi$$

where the canonical index i is the slice index number (not the complex root “ i ”) and Φ is the angle between successive slices. Then

$$\text{Warped Slice}(i) = \text{VectorRotate}(\text{Input Slice}(i), -i \cdot \Phi)$$

[0060] The warp operation may be carried out on the original slice terms (costerm, sinterm) to generate the warped slice record comprising warped slice terms (warped costerm, warped sinterm):

$$\text{warped costerm}(i) = \text{costerm}(i) \cdot \cos(-i \cdot \Phi) - \text{sinterm}(i) \cdot \sin(-i \cdot \Phi)$$

$$\text{warped sinterm}(i) = \text{costerm}(i) \cdot \sin(-i \cdot \Phi) + \text{sinterm}(i) \cdot \cos(-i \cdot \Phi)$$

[0061] The warp operation immediately above effectively re-tunes the receiver 104, using the stored slices, to a new frequency ($\text{freqRef} + \text{freqDelta}$). According to embodiments, this re-tuning is achieved from the stored slice data and not from a re-acquisition of slice data at some

other frequency (such as the new frequency) or a re-processing of the original ADC samples – which were may have been discarded or never even stored upon acquisition thereof. Moreover, such an operation is not a straightforward vector rotation, but rather a warping operation on slices, which has the resulting effect of tuning a slice record from one frequency (freqRef) to another ($\text{freqRef} + \text{freqDelta}$). As shown in Fig. 8, slices 1, 2, 3 and 4, ... N become aligned with each other. Performing a slice combining operation, as described earlier, on a set of warped slices produces a peak response at the warped frequency, $\text{freqRef} + \text{freqDelta}$. This corresponds to a filter tuned with this center frequency. Fig 8 illustrates how slice combination (a vector addition), according to one embodiment, combines the aligned slice vectors resulting from the warp operation. If the incoming signal is a frequency equal to $\text{freqRef} + \text{freqDelta}$, slices in the warped slice record will be aligned with each other or substantially aligned with each other, and will combine to give the maximum possible filter response.

[0062] FINDING CARRIER BY WARPING AND SLICE COMBINATION – According to one embodiment, the warping and slice combining functions shown and described herein may be used to identify the incoming carrier during the initial phase of the detection process by searching for the transmitted carrier over a range of frequencies. As shown in Fig. 9, freqRef is a reference frequency such as, for example, the frequency at which the transmitter was nominally designed to transmit. The actual carrier 904 may be unknown *a priori* to the receiver 104, which may then search for the actual carrier, armed only with the knowledge of the reference frequency and perhaps some knowledge of the transmitter (for example, that the actual frequency at the receiver is unlikely to deviate from the reference frequency by more than some number of Hertz). According to one embodiment, to find the actual carrier 904 of the incoming signal, the incoming signal may be sampled and converted to digital form (optionally after some analog pre-processing) and converted to slice data (complex cosine, sine pairs). The received incoming data is, therefore, converted to slice data, indexed and stored (the sequential storing of the slice data starting from a known memory location may inherently operate to index the slice data) as the ADC 110 generates samples from the incoming pre-processed (e.g. filtered, amplified and/or normalized among other possible operations) analog data. The sampled incoming data (e.g. samples generated by the ADC 110) need not be stored and if stored, may be discarded after the generation and storage of the slice data. The stored slices may then be combined over a selectable number of slices to achieve a filter 905 having a correspondingly selectable bandwidth. The bandwidth of the filter may be selected

by combining fewer (resulting in a broader filter) or a greater number of slices (resulting in a narrower filter). A peak in the filtered slice data may be indicative of the actual carrier. If no peak is detected indicative of the presence of the actual carrier 904 within the pass-band of the filter, the warping function shown and described above may be used to warp the original slices (in exemplary Fig. 9) to a next candidate frequency 906, a shift of freqDelta Hz in Fig. 9. The warped slices may again be combined to form a selectably narrow or broad filter at a new center frequency 907 and the presence of a peak 908 that is indicative of the actual carrier may be checked. This process may be repeated rapidly until the frequency of the actual carrier 904 is encompassed within the pass-band of the filter 909. Increasingly good estimates of the frequency of the actual carrier 904 may then be made by constructing one or more filters having a narrower band-width (by combining a greater number of slices) and checking for the presence of the actual carrier 904. Such narrower filters may aid the detection process, as a great deal of the noise may be attenuated, such that much of the energy within the pass-band of the filter originates from the carrier 904. The carrier hunt strategy described above is one simple strategy for locating the actual carrier. Other strategies may be envisioned that use warping and slice combination functions to achieve the same end.

[0063] USING A SINGLE SLICE RECORD TO DETECT FSK – According to one embodiment, the warping function shown and described herein may be used for efficient detection of Frequency Shift Keying (FSK) modulation. It is to be noted that FSK detection may also be carried out by performing two parallel slice computations, one at freq0 and one at freq1 . Referring now to Fig. 10, the incoming data may be converted to slice data at one reference frequency (freqRef) 1001 that may be selected, according to one embodiment to be, for example, about midway between the known or nominal upper (freq1) 1002 and nominal lower (freq0) 1003 FSK frequencies. If not already, the slice data may then be indexed and stored as the ADC 110 generates digital samples from the incoming pre-processed analog data. The incoming data (e.g. samples from the ADC 110) need not be stored and if stored, may be discarded after the acquisition and storage of the slice data. The stored slices may then be selectably warped over a selectable number of frequencies and combined to achieve a first relatively wide-band filter having a center frequency that is centered on one of the two nominal FSK frequencies, say freq0 1004. Effectively, this retunes the receiver 104 from a first frequency (freqRef in this example) to a second frequency freq0 away from the first frequency by an amount (in Hz) equal to the difference between freqRef and

freq0. Similarly, the original stored slices may then be selectably warped over a selectable number of frequencies and combined to achieve a second relatively wide-band filter having a center frequency that is centered on the second of the nominal FSK frequencies, freq1 1005 in this example. As was the case with the re-tuning of the receiver 104 to freq0, this effectively retunes the receiver 104 from the first frequency (freqRef in this example) to a second frequency freq1 away from the first frequency by an amount equal to the difference between freq1 and freqRef. When re-tuning the receiver 104 to freq0 and freq1, the pass-band of the first 1004 and second 1005 filters may be configured to be relatively wide (by combining relatively few slices) so as to increase the likelihood that, in each instance, the actual FSK frequencies (presumably in the vicinities of freq0 and freq1) will be located within the pass-band of the respective first and second filters. The warping function may be applied as needed to hunt or fine-tune for the actual FSK frequencies. The detection may be refined by constructing relatively narrower filters (by combining a relatively greater number of slices), which would increase the S/N of the output by attenuating a greater amount of noise.

[0064] Indeed, according to one embodiment and with reference to Fig. 11, supposing that an indication of the actual first and second FSK frequencies (actualfreq0 at reference numeral 1104 and actualfreq1 at reference numeral 1110) is detected within the pass-band of the wide-bandwidth filters generated from the slice data, the warping function may be used again for a precise identification of the two actual FSK frequencies actualfreq0 1104 and actualfreq1 1110. As shown, freq0 1102 and actualfreq0 1104 differ by freqDelta0 Hz, as shown at reference numeral 1106. Similarly, freq1 1108 and actualfreq1 1110 differ by freqDelta1 Hz, as shown at reference numeral 1112. The two deltas, namely freqDelta0 1106 and freqDelta1 1112, represent the amount of deviation of the two FSK frequencies away from the nominal FSK frequencies freq0 1102 and freq1 1108 at which the transmitter was designed to transmit. Such deviation may be caused by, for example, a calibration error caused by imperfect tuning of the transmitter at the factory, temperature effects, or other environmental effects such as local conductivity around the transmitter that influence the transmitted frequency. As such, freq0 1102 and freq1 1108 may be thought of as a first order approximation of the location of actualfreq0 1104 and actualfreq1 1110, respectively. To fine tune the receiver 104 to the two actual FSK frequencies actualfreq0 1104 and actualfreq1 1110 and to reject unwanted signal(s) (if any), the warping function may be again applied to the already-warped slice data to iteratively (if required) create suitably narrow

bandwidth filters at different center frequencies until strong peaks indicative of the presence of the actual frequencies at 1104 and 1110 appear in the pass-bands of the filters. This process may be iteratively carried out until the actual frequencies at 1104 and 1110 are sufficiently isolated and the frequencies (noise, generally) on either side of the thus-created narrow-band filters are rejected to enable reliable detection and decoding.

[0065] Referring again to Fig. 11, after having detected the actual FSK signals around nominal frequencies freq0 1102 and freq1 1108, the warping function may be applied to re-tune the receiver 104 (if not already re-tuned as a result of searching for the two actual FSK frequencies) from freq0 1102 to actualfreq0 1104, by warping the filter by a few Hz, shown in Fig. 11 at freqDelta0 1106. Similarly, the warping function also may be applied to re-tune the receiver 104 from freq1 1108 to actualfreq1 1110, by again warping the filter by a few Hz, shown in Fig. 11 at freqDelta1 1112. The result of this fine tuning, therefore, is a receiver 104 that utilizes slice data acquired at freqRef and that has been re-tuned to the first and second actual FSK frequencies; namely, freqwarp0 1114 (equal to $\text{freq0} - \text{freqRef} + \text{freqDelta0}$) and freqwarp1 1116 (equal to $\text{freq1} - \text{freqRef} + \text{freqDelta1}$). As the relationship between the two FSK frequencies is known *a priori* to the receiver (such as a known ratio relationship), such relationship may be exploited by the receiver as it tunes the two separate FSK frequencies.

[0066] According to one embodiment, therefore, an FSK receiver 104 may be configured to be tuned at a frequency freqRef that is neither the first FSK frequency freq0 nor the second FSK frequency freq1 . The receiver 104 may then be re-tuned, using warp and slice combining functions, to each of the first and second FSK frequencies freq0 and freq1 and, thereafter, to the actual FSK frequencies through fine-tuning without, however, re-acquiring data at either of these frequencies; that is, without re-acquiring new raw ADC data at the re-tuned frequency or without reading previously stored sampled raw data from memory 114. Moreover, such re-tuning according to embodiments may be carried out by processing vastly less data (by, e.g. orders of magnitude or more) than would otherwise be required had new ADC data been acquired or had the original data been maintained in memory 114 and re-processed to detect the freq0 and freq1 FSK frequencies. That is, according to one embodiment, the re-tuning of the receiver 104 may be effected solely by carrying out what are, for the most part, addition operations with some multiplication operations on a limited store of previously-acquired slice data.

[0067] WARPING TO REDUCE NOISE, ALIGN SLICES TO AN AXIS – Referring to Fig. 8, the aligned slice vectors have a non-zero cosine component along the x-axis and a non-zero sine component along the y-axis. Each of these components may include some signal component and some noise. According to one embodiment, if the aligned slice vectors of Fig. 8 were forced to align with, for example, the x-axis (thereby driving the sine component thereof to zero), the sine components thereof would include zero signal and only noise. This noise may be safely ignored, as all of the energy of the slice (and thus of the signal) is now aligned with the x-axis. Accordingly, one embodiment changes the warping function WF in the detection to put all the slice energy into one of the two dimensions. For instance, if all slices were to be pointed along the real axis (cosine, x-axis), then no signal would be left in the imaginary (sine, y-axis) axis, leaving only noise therein. According to one embodiment, therefore, aligning the warped slices to either the x, or y axis may be carried out by adding a constant angle (Θ) to the warped slices:

$$WF(\Phi) = (i \cdot \Phi) + \Theta;$$

[0068] Accordingly, this implementation of the warping function adds a constant angle after scaling Φ by i , the slice index number. The addition of the constant angle, Θ (which may be positive or negative in sign) causes the output slices to be aligned in a selected (and preferred) direction, for example, aligned with the real axis (cosine component or x-axis) or the imaginary axis (sine component or y-axis). The warped slices, however, may be aligned by warping to any angle though judicious selection of the constant angle.

[0069] WARPING TO CORRECT FREQUENCY DEFECT – According to further embodiments, warping functions may be devised based on more sophisticated patterns or sequences of the slice index number. For example, the scaling factor need not be an integer. For example, if a transmitter transmits packets whose frequency falls (or rises) towards the end of a packet, the warping function may be adapted to track that falling frequency toward the end of the packet. For example, assuming the receiver has identified the starting slice index of a packet, the following warp function could be applied to the slice record for the purpose of aligning all slices in a packet.

$$WF() = (\text{Scaling Factor} \cdot i \cdot \Phi)$$

where, for example, Scaling Factor = [1 1 1 1 1 1 1 1 .9 .9 .8 .8 .7 .6 .5 .3 etc.] The scaling factor

may be an algebraic expression or may be read from a table stored in memory 114 with suitable values stored therein. The warping function, in this manner, may be configured to track any quantifiable change in the frequency profile of the received packets, thereby allowing for, for example, non-constant and/or non-integer sequential adjustments of warping angle Φ from slice to slice.

[0070] WARPING TO DETECT CHIRP – Warping, according to one embodiment, also may be applied to any incoming signal having a non-constant frequency, such an intentional chirp-type signal, or a transmitter with poor frequency control where the frequency of the transmitted signal increases or decreases as the transmitter battery depletes.

[0071] For example, if the incoming signal is a rising chirp, the slice data may be warped by an angle that increases faster than the integer pattern shown and described relative to Figs. 7 and 8. For example, the first slice may be warped by $1 \cdot \Phi$, the second slice may be warped by $2.2 \cdot \Phi$, the third slice may be warped by $3.3 \cdot \Phi$, and so on. According to embodiments, therefore, the computation of the warping angle may comprise any function that reflects the frequency structure of the expected incoming signal. The use of slices, according to embodiments, enables efficient use of resources, in that a high degree of data compression may be achieved by converting the raw sample stream from the ADC 110 to slice data and discarding (or failing to store) the raw sample data. This is significant not only in terms of the size of the memory 114 required, but also in terms of the amount of calculations to be carried out later in the detection and decoding processes. The use of slices, warping functions, and slice combining functions, according to embodiments, also affords the receiver 104 a high degree of flexibility at multiple places in the detection algorithm. Because the original slices can be designed to have a relatively wide bandwidth, they can be re-tuned/warped over great number of Hz in either direction. For example, a slice with a 5000 Hz bandwidth, according to embodiments, may be warped 1000-2000 Hertz or more, up or down, without significant loss of signal strength. **SLICE CORRELATION: FINDING A KNOWN PATTERN** – According to one embodiment, a detection procedure may be carried out, to determine the presence of one or more data packets in the slice record. According to one embodiment, it is not the original raw ADC sampled data stream that is analyzed (which may have been previously discarded anyway), but the indexed and stored slice data. According to one embodiment, a function (for example, a real or complex correlation function) may be applied to

the slice data, to compare the slice data with one or more pre-stored slice patterns corresponding to a known slice pattern in the signal. According to one embodiment, the data packets sought to be detected (and framed, to determine the boundaries thereof) may comprise a preamble of known length and configuration, followed by a payload of known length from which useful information may be extracted by a decoding process. For example, each data packet sought to be detected may comprise a preamble comprising 11 bits. For example, the preamble may comprise a known sequence such as, for example, a sequence of 7 zeros, followed by 1010 (00000001010). To determine the presence of a packet, therefore, a real or complex correlation function may be applied, according to one embodiment, to cross-correlate slice data to a slice pattern corresponding to the known preamble. To the extent that the slice data encodes data corresponding to one or more preambles of one or more data packets, the correlation function will return higher results when the preamble(s) of the input slice data and that of the template are aligned with one another, correspondingly lower results as the preambles in the input slice data and the template are only partially aligned with one another and lowest results when the preambles in the input slice data and the template are not aligned with one another or the input slices do not comprise any packets. This cross-correlation operation represents a very narrow-band filter, with bandwidth proportional to the reciprocal of the number of slices in the known preamble.

[0072] In one embodiment, slice correlation and warping may be used together to provide a fair estimate of the actual carrier frequency of the received signal, as the receiver 104 is iteratively re-tuned through warping and the resultant warped slices correlated with, for example, the expected slice pattern used to determine the presence and boundaries of the preamble. In this manner, a high correlation value may be associated with the actual carrier frequency of the received signal.

[0073] SLICE CORRELATION: FINDING EVIDENCE OF A PACKET –According to one embodiment, a detection procedure may be carried out to determine the presence of one or more data packets in the slice record prior to determining the frequency(ies) of the carrier(s). As in the discussion of cross-correlation with a pre-stored template above, only the indexed and stored slice data need be analyzed. According to one embodiment, a function (for example, a real or complex correlation function) may be applied to the slice data, to compare the slice data with itself (auto-correlation). Often, it is useful to perform correlation calculations at just a few different lags.. For example, is the energy for an entire slice record, A , can be estimated by slice correlation

with lag = 0:

$$Corr(0) = \sum_{n=1}^N A_n \times A_n$$

[0074] AutoCorr(0) represents the baseline energy level for the slice record, against which other autocorrelations can be compared.

[0075] For a slice record containing no packets, slice auto-correlation with lag = 1:

$$Corr(1) = \sum_{n=1}^{N-1} A_n \times A_{n+1}$$

[0076] According to one embodiment, prior to determining the frequency(ies) of the carrier(s), an autocorrelation may be performed on the slice record A to determine if a packet is present therein. For a case where the slice record contains one or more packets, $Corr(1)$ will have a higher value relative to $Corr(0)$. This is an indication that a packet exists somewhere in the slice record. For a slice record containing no packets, slice auto-correlation with lag = 1 will have a very low value relative to $AutoCorr(0)$ if the slice record contains only uncorrelated noise. According to one embodiment, a packet may be considered to have been detected when the autocorrelation term $Corr(1) / Corr(0)$ is determined to be above a predetermined threshold.

[0077] Confirmatory evidence for the presence of a packet can be developed if multiple packets exist in the slice record at a known packet separation m (measured in slices). Correlating the slice record with a lag = m (lag equal to the packet spacing) produces a high correlation result if packets are present at the anticipated spacing:

$$Corr(m) = \sum_{n=1}^{N-m} A_n \times A_{n+m}$$

[0078] According to one embodiment, a packet may be considered to have been detected when the correlation terms computed multiple times over a range of anticipated packet separations, $Corr(m \pm \text{range})$, are determined to be above a predetermined threshold relative to $Corr(0)$. The expected range of packet separations arises due to variations in the as-yet to be determined packet frequency. In this manner, using slice data, packet detection may be carried out by correlating a delayed version of the slice record A with the slice record A and monitoring the magnitude of the

resulting correlation terms.

[0079] OVERLAPPING PACKETS – According to one embodiment, the greater the number of packets in a slice data record, the better the auto-correlation results may be. Slices representing multiple suspected packets may be added to one another, to increase the likelihood of correct packet detection. Moreover, the packet boundaries may be determined by adding two or more suspected packets with one another. The result of the addition will be highest when the respective packets are perfectly aligned. The suspected packets may be shifted by one or more slices (according to one embodiment, by the number of slices between packets) and the addition operation may be applied to the shifted packets in this manner to determine the boundaries of the packets. It is to be understood, however, that there is more than one method of packet detection and framing. All such methods are understood to be encompassed by the present embodiments. It is also to be understood that, having identified the boundaries of packets, the signal to noise ratio is increased when only the packet is observed, as the only noise present is that within the packet and as all noise outside of the packet boundaries may be excluded or greatly attenuated.

[0080] MODULATION SCHEME: BPSK – The packet need not be encoded and decoded using FSK modulation. According to one embodiment, another forms of digital modulation may be used such as, for example, binary phase shift keying (BPSK). In such an encoding scheme, the symbol 0 may be encoded using a sine waveform of a certain number of cycles and the symbol 1 may be encoded using a –sine waveform out of phase by π radians of the same number of cycles. For example, a packet encoded using BPSK may comprise a preamble and a payload. The preamble may comprise, for example, seven 0s, followed by 1, 0, 1 and 0, in the form (00000001010). Real or complex correlation methods may be utilized, as described above, to determine the presence of one or more packets by comparing the slice record to a predetermined slice pattern representing the preamble. This operation serves to identify the presence of a packet and to synchronize the receiver 104 with the starting bit of the preamble. As noted above, the correlation function may additionally provide an estimate of the actual carrier frequency of the signal.

[0081] ITERATIVE DECODING – According to one embodiment, the bits of the packet payload may be decoded in the receiver one at a time in succession. To determine whether a bit is a logic zero or a logic one, successive correlations against a “zero template” and a “one

template” may be used, with the larger of the two correlation results indicating the value of the bit. Such a method may, according to one embodiment, be used to decode the payload of a packet that appears after the preamble thereof, as the bit sequence in the payload is most often unknown a-priori by the receiver.

[0082] ARCTANGENT – According to one embodiment, in cases in which the signal to noise ratio is reasonable (e.g. around 0 dB or above), taking the arctangent of slices containing suspected packets may be revealing, and may identify the presence or absence of a packet.

[0083] CARRIER HUNT STRATEGY – According to one embodiment, once the presence of one or more packets in the slice data is determined, to determine the frequency(ies) with which the packets were modulated, whether encoded using FSK or PSK (for example) or however encoded, if a rough estimate of the actual frequency of the signal is known (say within 20 Hz, for example for an exemplary 20kHz signal), the magnitudes of the correlations of the preamble, for example, may be determined at each of 20 different frequencies, at 1 Hz (or less) increments. According to one embodiment, the rough estimate of the carrier frequency(ies) may be the nominal frequency(ies) with which the transmitter is designed to transmit. Some knowledge of the communication channel may enable such an educated guess as to the frequency range within which the actual signal is likely to be found. In such a case, after having computed the correlation for each frequency within the frequency range, the frequency associated with the largest correlation magnitude may be safely assumed to be the (or one of the) carrier frequencies.

[0084] FIX DETECTION BY FLATTENING PHASE – It is to be understood that other methods of determining the frequency of a detected packet may be employed, without departing from the scope of the embodiments described in the present disclosure. For example, for each bit of a packet, the phase angles of the bit’s constituent slices may be determined. The phase angles may, according to one embodiment, be determined by taking the arctangent (the ratio of the sine component of the slice to the cosine component) of each slice. Such a method may be best implemented when the signal to noise ratio is above a predetermined threshold such as, for example, about 0 dB. For BPSK modulation, such a phase angle, may swing between 0 and 2π , in a saw-tooth like fashion. The presence of such a saw-tooth pattern is suggestive that the constituent slices making up the bits being examined are, using the polar representation of Fig. 7,

misaligned, as are slices 1, 2, 3 and 4 in that figure. With reference to Fig. 8, when the frequency being tested results in warping angles that form more or less a straight line (as opposed to a saw-tooth pattern), that frequency may be the or close to the actual frequency of the signal of interest. For PSK, for example, the warping angles will shift from one warping angle to another warping angle that is indicative of the PSK frequency at which the data was encoded. The resultant pattern may then resemble a square wave, from which the data may be readily apparent.

[0085] MODULATION SCHEME: MSK – Using methods similar to those previously described, data encoded using other modulation formats may be detected and decoded using only the stored slices and the warping function described herein. For example, the data in the slices may have been encoded using, for example, Multiple Shift Keying (MSK) using, for example, 4 frequencies or, for example, 16 frequencies to represent different symbols. In this case, each symbol may comprise information bits encoded with one or more frequencies (e.g. one or two) out of a plurality (e.g. 16) of frequencies, with each symbol potentially representing more than one bit. Other modulation formats that encode data may be decoded using only the slice information (and not the original data from the ADC 110, which has since been discarded) and the warping and slice combining functions described herein. Moreover, data encoded using combinations of modulation formats also may be detected and decoded, again using only slice information, warping, and slice combination. For example, data encoded with a combination of MSK and PSK may be decoded from the retained slice data.

[0086] In each case, the computational load on the controller 112 portion of the receiver 104 is lighter than it otherwise would be if the controller 112 were obliged to re-process the original raw data stream. For similar reasons, the memory requirements of the receiver's controller 112 are orders of magnitude less than would be the case had it been necessary to store the original raw incoming data in order to operate thereon later, during detection and decoding.

[0087] ONE-BIT ADC – For situations exhibiting an especially low signal to noise ratio, it may be advantageous for the receiver 104 to use an analog comparator or a 1-bit ADC to quantize the signal as being above or below a predetermined threshold (encoded as two values: +1 and -1). In this manner, the amount of data that is stored in the slice construct, according to embodiments, greatly decreases compared to storing multi-bit representations of the signal. A comparator or a 1-bit ADC may be used to good advantage in situations exhibiting low signal to

noise ratio, as it enables samples to be gathered at a very high sample rate while still computing slices in a fast real-time loop on an ordinary processor. Inside the real-time loop, multiply operations are greatly simplified because one of the operands is either +1 or -1.

[0088] Fig. 12 is a logic flow of a method according to one embodiment. As shown therein, at B121 a signal that encodes one or more data packets is received. At B122, the received signal then may be sampled in an ADC to generate sampled values. At B123, a slice then may be generated and stored in memory, where each slice comprises a pair of values representing a selected slice interval of time. At B124, data packets are detected and decoded from the stored slices using various combinations of warp and slice combination operations.

[0089] Fig. 13 is a logic flow of a method according to one embodiment. As shown therein, at B131, a signal may be received that encodes a data packet at a first frequency. The received signal, as shown at B132, may then be sampled in an ADC to generate sampled values. The sampled values, as called for at B133, may then be correlated with first and second templates of values obtained at a second frequency that may be different from the first frequency to generate slices at the second frequency. According to one embodiment and as described and shown herein, the first template may be generated using a first reference function and the second template may be generated using a second reference function that is in quadrature with the first reference function. Some or all of the slices at the second frequency may be transformed (also denoted as “warped” herein) to slices at the second frequency (also denoted as “freqRef” herein), plus or minus an offset (denoted as “freqDelta” herein), as shown at B134. As shown at B135, a filter having a center frequency at the second frequency plus or minus the offset may be generated by combining the transformed (warped) slices.

[0090] According to one embodiment, a determination may then be made, as suggested at B136, whether the first frequency (the frequency of interest at which the data packet(s) is/are encoded) is within the pass-band of the generated filter. If the first frequency is indeed within the pass-band of the thus-generated filter, further steps may be carried out such as, for example, detection and decoding steps, as detailed herein. If the first frequency is not present within the pass-band of the generated filter, the slice transforming (warping) and filter generating (slice combining) steps may be iteratively repeated using respectively different offsets until the first frequency is indeed within the pass-band of the filter, as indicated by the NO branch of B136.

[0091] While certain embodiments of the disclosure have been described, these embodiments have been presented by way of example only, and are not intended to limit the scope of the disclosure. Indeed, the novel methods, devices and systems described herein may be embodied in a variety of other forms. Furthermore, various omissions, substitutions and changes in the form of the methods and systems described herein may be made without departing from the scope of the disclosure. The accompanying claims and their equivalents are intended to cover such forms or modifications as would fall within the scope of the disclosure. For example, those skilled in the art will appreciate that in various embodiments, the actual physical and logical structures may differ from those shown in the figures. Depending on the embodiment, certain steps described in the example above may be removed, others may be added. Also, the features and attributes of the specific embodiments disclosed above may be combined in different ways to form additional embodiments, all of which fall within the scope of the present disclosure. Although the present disclosure provides certain preferred embodiments and applications, other embodiments that are apparent to those of ordinary skill in the art, including embodiments which do not provide all of the features and advantages set forth herein, are also within the scope of this disclosure.

[0092] Embodiments of the present invention have been described above. Further embodiments of the present invention can also be realized by systems or apparatuses that read out and execute programs recorded on a memory device to perform the functions of the above-described embodiment(s), and by a method, the steps of which are performed by, for example, reading out and executing a program recorded on a memory device to perform the functions of the above-described embodiment(s). For this purpose, the program may be provided to the system or apparatus (e.g. receiver), for example via a network or from a recording medium of various types serving as the memory device (e.g. computer-readable medium).

[0093] The present invention may be defined by way of the following clauses. It will be understood that the features recited are interchangeable defined by the following clauses and their dependencies. That is, the features of the clauses may be combined to define the present invention.

CLAUSES

1. A method, comprising:
receiving a signal, the signal encoding a data packet;
sampling the received signal;

generating and storing a plurality of slices comprising pairs of values for each of a selected number of samples of the signal; and

detecting a presence of and decoding the data packet from the stored slices.

2. The method of clause 1, wherein generating each of the plurality of slices comprises:

correlating samples of the signal with a first reference template;

generating a first value of the pair of values;

correlating the selected number of samples of the signal with a second reference template;

and

generating a second value of the pair of values.

3. The method of clause 2, wherein the first reference template comprises a cosine function at a reference frequency and the second reference template comprises a sine function at the reference frequency.

4. The method of any of clauses 1 to 3, further comprising forming a filter by combining a number of the plurality of slices.

5. The method of any of clauses 1 to 4, wherein detecting the presence of the packet comprises detecting a carrier frequency within a pass-band of a filter formed by the plurality of slices.

6. The method of any of clauses 1 to 4, wherein detecting the presence of the packet comprises detecting a carrier frequency within a pass-band of a filter formed by combining slices.

7. The method of any of clauses 4 to 6, wherein detecting further comprises re-tuning a center frequency of the filter from a first center frequency to a second center frequency that is different from the first center frequency using the stored slices.

8. The method of clause 7, wherein re-tuning the center frequency of the filter comprises warping the slices from which the filter was formed by rotating the respective pairs of values thereof by a quantity.

9. The method of clause 8, wherein the quantity comprises a rotation angle, a scaling factor and indices associated with the slices from which the filter was formed.

10. The method of clause 8, wherein the quantity comprises a sum of a phase angle from a reference frequency and a product of a rotation angle and a slice index.

11. A signal receiver, comprising:

analog-to-digital converter means (ADC) configured to sample a received signal;
memory means;

controller means coupled to the memory means and configured to:

generate and store, in the memory means, a slice comprising a pair of values for each of a selected number of samples of the signal; and

detect a presence of and decode the data packet from the stored slices.

12. The signal receiver of clause 11, wherein the memory means is configured to store at least a first reference template and a second reference template and wherein the controller means is further configured to correlate the selected number of cycles of the sampled signal with the first reference template to generate a first value of the pair of values and to correlate the selected number of samples of the signal with the second reference template to generate a second value of the pair of values.

13. The signal receiver of clause 11 or clause 12, wherein the controller means is further configured to combine a number of slices to form a filter.

14. The signal receiver of clause 13, wherein a bandwidth of the filter is related to the number of combined slices.

15. The signal receiver of any of clauses 11 to 14, wherein the controller means is further configured to detect the presence of the packet by detecting a carrier frequency within a pass-band of a filter formed by combining the slices.

16. The signal receiver of any of clauses 13 to 15, wherein the controller means is further configured to re-tune, using the stored slices, a center frequency of the filter from a first center frequency to a second center frequency that is different from the first center frequency.

17. The signal receiver of any of clauses 11 to 16, wherein the signal encodes data packets at a first frequency and wherein controller means is further configured to:

correlate the samples with first and second templates of values obtained at a second frequency that is different from the first frequency to generate a plurality of slices that each comprise a pair of values;

transform at least some of the plurality of slices at the second frequency to slices at the second frequency plus or minus an offset, and

generate a filter having a center frequency at the second frequency plus or minus the offset by combining the transformed slices.

18. A method, comprising:
receiving a signal, the signal encoding a data packet at a first frequency;
sampling the signal to generate sampled values;
correlating the sampled values with first and second templates of values obtained at a second frequency that is different from the first frequency to generate a plurality of slices at the second frequency, each of the slices comprising a pair of values;
transforming at least some of the plurality of slices at the second frequency to slices at the second frequency plus or minus an offset, and
generating a filter having a center frequency at the second frequency plus or minus the offset by combining the transformed slices.

19. The method of clause 18, further comprising determining whether the first frequency is within a pass-band of the generated filter

20. The method of clause 19, further comprising iteratively transforming, generating and determining using respectively different offsets until the first frequency is within the pass-band of the filter.

21. A method, comprising:
receiving a signal, the signal encoding a data packet;
sampling the signal to generate sampled values;
generating a slice record comprising a plurality of slices by correlating the sampled values with first and second reference templates, the first reference template comprising a first reference function and the second reference template comprising a second reference function in quadrature with the first reference function;

auto-correlating a portion of the slice record with a delayed version of the portion of the slice record to generate auto-correlation terms; and

determining when magnitudes of auto-correlation terms exceed a predetermined threshold for a predetermined number of auto-correlation terms.

22. The method of clause 21, further comprising determining a carrier frequency of the received signal.

23. The method of clause 22, wherein determining comprises:
warping at least some of the plurality of slices by a frequency offset, and
generating a filter from the warped slices and,

determining whether the carrier frequency is within a pass-band of the generated filter.

24. A method, comprising:

receiving a signal, the signal encoding a data packet;

sampling the signal to generate sampled values;

generating a slice record comprising a plurality of slices from the sampled values by correlating the sampled values with first and second reference templates, the first reference template comprising a first reference function and the second reference template comprising a second reference function in quadrature with the first reference function;

cross-correlating the slice record with a stored template to generate cross-correlation terms;

and

determining when a magnitude of the cross-correlation terms exceeds a predetermined threshold for a width of the stored template.

25. The method of clause 24, wherein the first reference template comprises a cosine function and wherein the second template function comprises a sine function.

26. The method of clause 24, further comprising determining a carrier frequency of the received signal.

27. A method, comprising:

receiving a signal;

sampling the signal to generate sampled values;

correlating the sampled values with predetermined first and second templates of values obtained at a first frequency to generate a plurality of slices at the first frequency;

transforming at least some of the generated plurality of slices at the first frequency to slices at a second frequency that is different from the first frequency;

generating a first filter having from the slices at the second frequency;

transforming at least some of the generated plurality of slices at the first frequency to slices at a third frequency that is different from the first and second frequencies, and

generating a second filter from the slices at the third frequency.

28. The method of clause 27, further comprising discarding the generated sampled values of the received signal after generating the plurality of slices at the first frequency.

29. The method of clause 27 or clause 28, further comprising detecting a first carrier frequency within a pass-band of the first filter and detecting a second carrier frequency within a

pass-band of the second filter.

30. A method, comprising:

receiving a signal, the signal encoding data packets;

sampling the signal to generate sampled values;

generating a slice record comprising a plurality of slices from the sampled values by correlating the sampled values with first and second reference templates, the first reference template comprising a first reference function and the second reference template comprising a second reference function in quadrature with the first reference function;

auto-correlating a portion of the slice record spanning at least two preambles of the encoded data packets with a delayed version thereof to generate auto-correlation terms; and

determining when magnitudes of auto-correlation terms exceed a predetermined threshold for a predetermined number of auto-correlation terms.

31. The method of clause 30, further comprising determining boundaries of the data packets from magnitudes of the auto-correlation terms.

32. The method of clause 30 or clause 31, wherein the carrier frequency of the signal is detected when successive phase angles, across bits of the data packet, least resemble a first predetermined pattern and most resemble a second predetermined pattern.

33. A program, which when executed by a computer, causes the computer to carry out the method of any of clauses 1 to 10 and 18 to 32.

34. A program which, when executed by a computer, causes the computer to function as the signal receiver of any of clauses 11 to 17.

35. A storage medium storing the program according to clause 33 or clause 34.

[0094] Accordingly, the preceding merely illustrates the principles of the invention. It will be appreciated that those skilled in the art will be able to devise various arrangements which, although not explicitly described or shown herein, embody the principles of the invention and are included within its scope. Furthermore, all examples and conditional language recited herein are principally intended to aid the reader in understanding the principles of the invention and the concepts contributed by the inventors to furthering the art, and are to be construed as being without limitation to such specifically recited examples and conditions. Moreover, all statements herein reciting principles, aspects, and aspects of the invention as well as specific examples thereof, are intended to encompass both structural and functional equivalents thereof. Additionally, it is

intended that such equivalents include both currently known equivalents and equivalents developed in the future, i.e., any elements developed that perform the same function, regardless of structure. The scope of the present invention, therefore, is not intended to be limited to the exemplary aspects shown and described herein.

CLAIMS:

1. A method, comprising:

receiving a signal, the signal encoding a data packet;
sampling the signal at a sample rate to generate a plurality of sampled values;
generating a slice record comprising a plurality of slices by correlating the sampled values with a first reference template and a second reference template, the first reference template comprising a first reference function and the second reference template comprising a second reference function in quadrature with the first reference function; and
auto-correlating a portion of the slice record with at least one delayed version of the portion of the slice record to generate at least one auto-correlation term.

2. The method of claim 1, further comprising:

determining that a data packet is present in the signal when one or more magnitudes of the at least one auto-correlation term exceed a predetermined threshold.

3. The method of claim 2, further comprising:

determining that a data packet is present in the signal when more than one magnitudes of the at least one auto-correlation term exceed a predetermined threshold.

4. The method of claim 2, wherein the predetermined threshold is a noise threshold.

5. The method of claim 4, wherein the noise threshold is determined based on auto-correlation of the portion of the slice record with a version of the portion of the slice record with zero delay.

6. The method of claim 1, wherein the at least one delayed version of the portion of the slice record comprises a version delayed by one slice.

7. The method of claim 1, wherein the at least one delayed version of the portion of the slice record comprises a version delayed by an expected packet separation.

8. The method of claim 1, wherein the at least one delayed version of the portion of the slice record comprises versions delayed by a range of expected packet separations.

9. The method of claim 1, wherein the first reference function comprises a cosine function and the second reference function comprises a sine function.

10. The method of claim 2, further comprising:

when a data packet is determined to be present in the signal, decoding the data packet from the slice record.

11. The method of claim 2, further comprising:

when a data packet is determined to be present in the signal, warping a portion of the slice record to determine a carrier frequency of the data packet.

12. The method of claim 11, wherein warping the portion of the slice record comprises:

warping the portion of the slice record so that a phase of a first slice of the data packet is substantially the same as a first slice of the first reference template or a first slice of the second reference template.

13. The method of claim 11, wherein warping the portion of the slice record comprises:

warping the portion of the slice record so that a phase of a first slice of the data packet is substantially equal to 0 or substantially equal to $\pi/2$.

14. The method of claim 11, wherein warping the portion of the slice record comprises:

warping the portion of the slice record to substantially match a frequency of the data packet and substantially align a phase of the data packet to axes of the plurality of slices.

15. The method of claim 11, further comprising using the warped slices to decode the data packet.

16. A receiver, comprising:

an analog-to-digital converter (ADC) configured to generate a plurality of sampled values at a sample rate from a received signal;

a memory configured to store a first reference template and a second reference template;

a controller coupled to the memory and the ADC and configured to:

generate a slice record comprising a plurality of slices by correlating the sampled values with the first reference template and the second reference template, the first reference template comprising a first reference function and the second reference template comprising a second reference function in quadrature with the first reference function; and

auto-correlate a portion of the slice record with at least one delayed version of the portion of the slice record to generate at least one auto-correlation term.

17. The receiver of claim 16, wherein the controller is further configured to:

Determine that a data packet is present in the signal when one or more magnitudes of the at least one auto-correlation term exceed a predetermined threshold.

18. The receiver of claim 17, wherein the predetermined threshold is determined based on auto-correlation of the portion of the slice record with a version of the portion of the slice record with zero delay.

19. The receiver of claim 16, wherein the at least one delayed version of the portion of the slice record comprises a version delayed by an expected packet separation.

20. A computer program product, comprising a computer readable memory storing computer executable instructions thereon that when executed by a computer perform the method step:

receive a signal, the signal encoding a data packet;

sample the signal at a sample rate to generate a plurality of sampled values;

generate a slice record comprising a plurality of slices by correlating the sampled values with a first reference template and a second reference template, the first reference template comprising a first reference function and the second reference template comprising a second reference function in quadrature with the first reference function; and

auto-correlate a portion of the slice record with at least one delayed version of the portion of the slice record to generate at least one auto-correlation term.

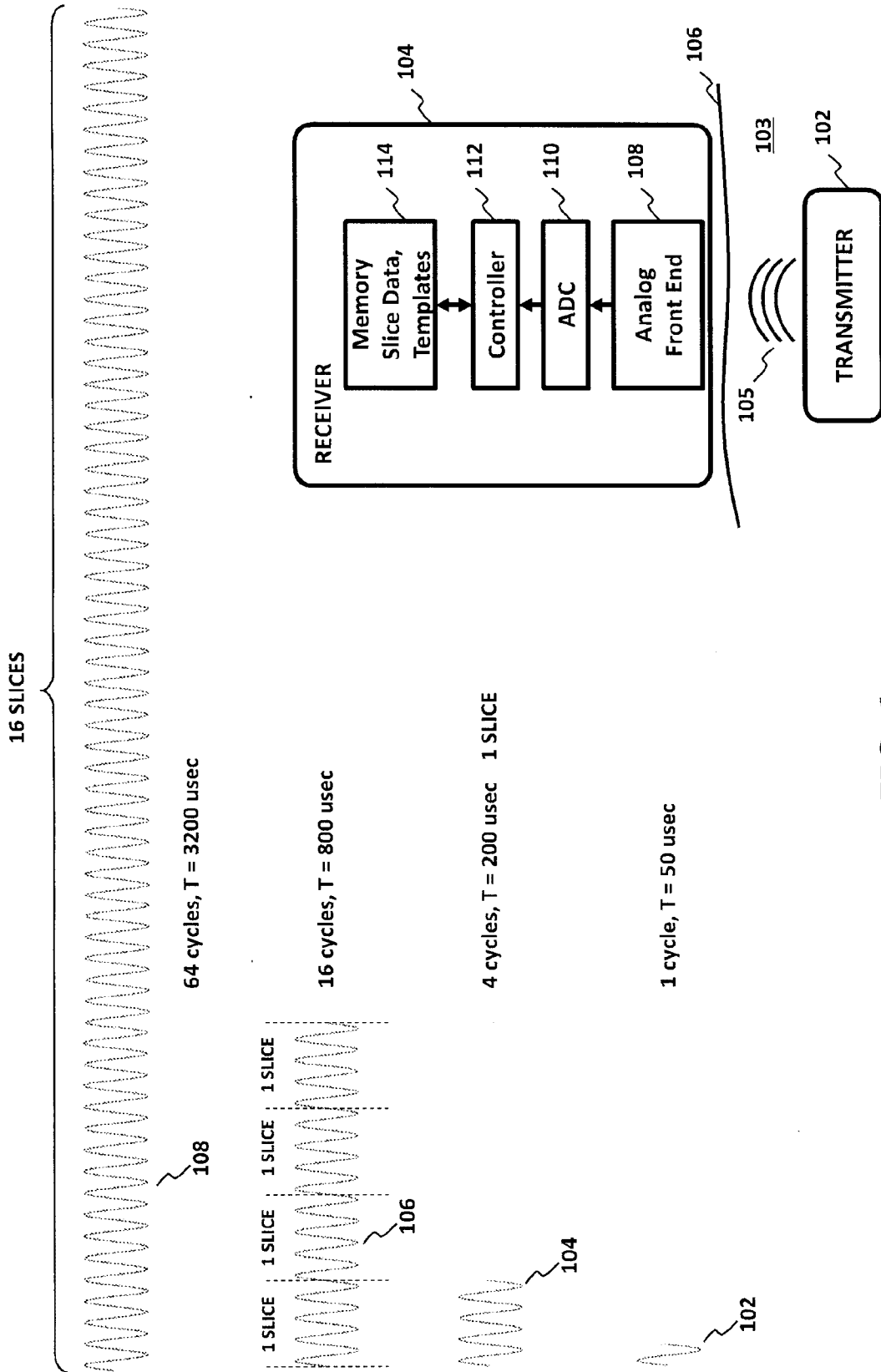
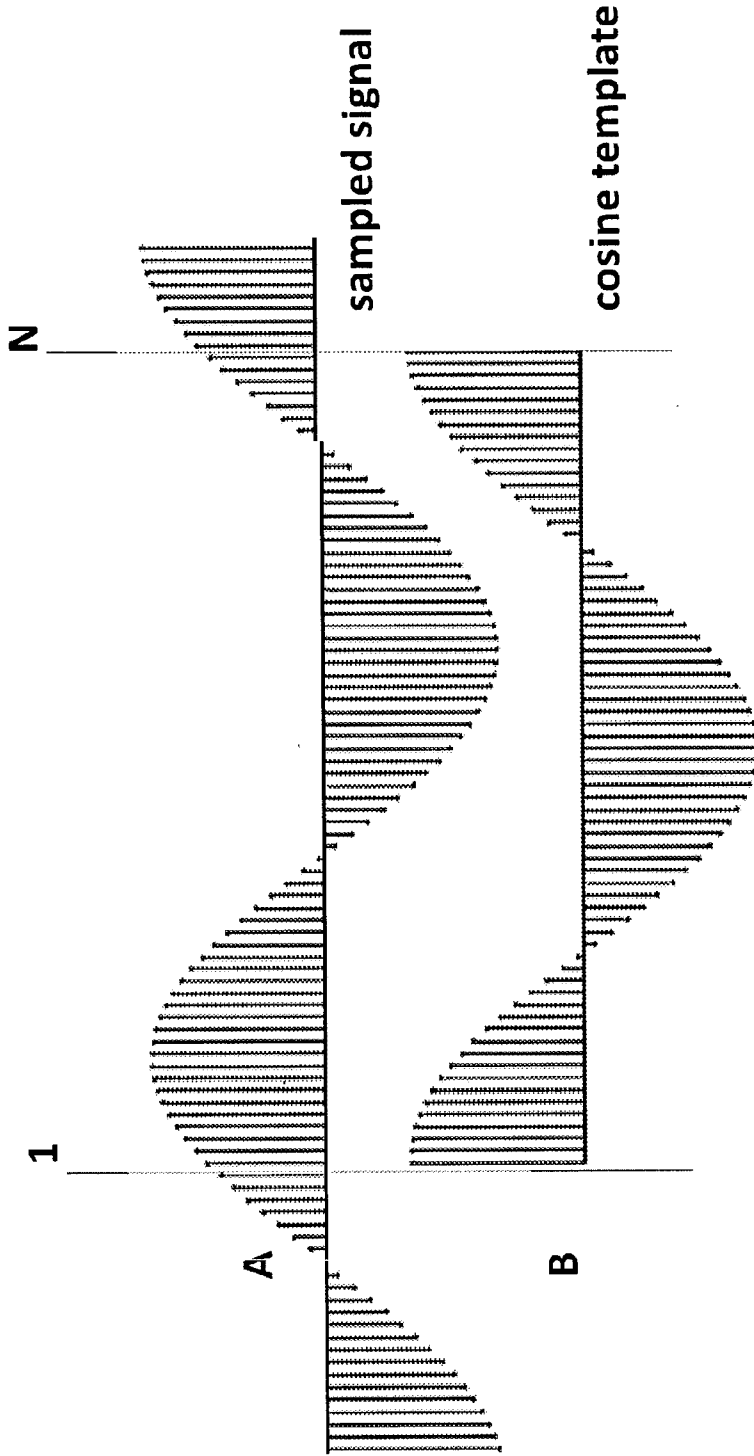


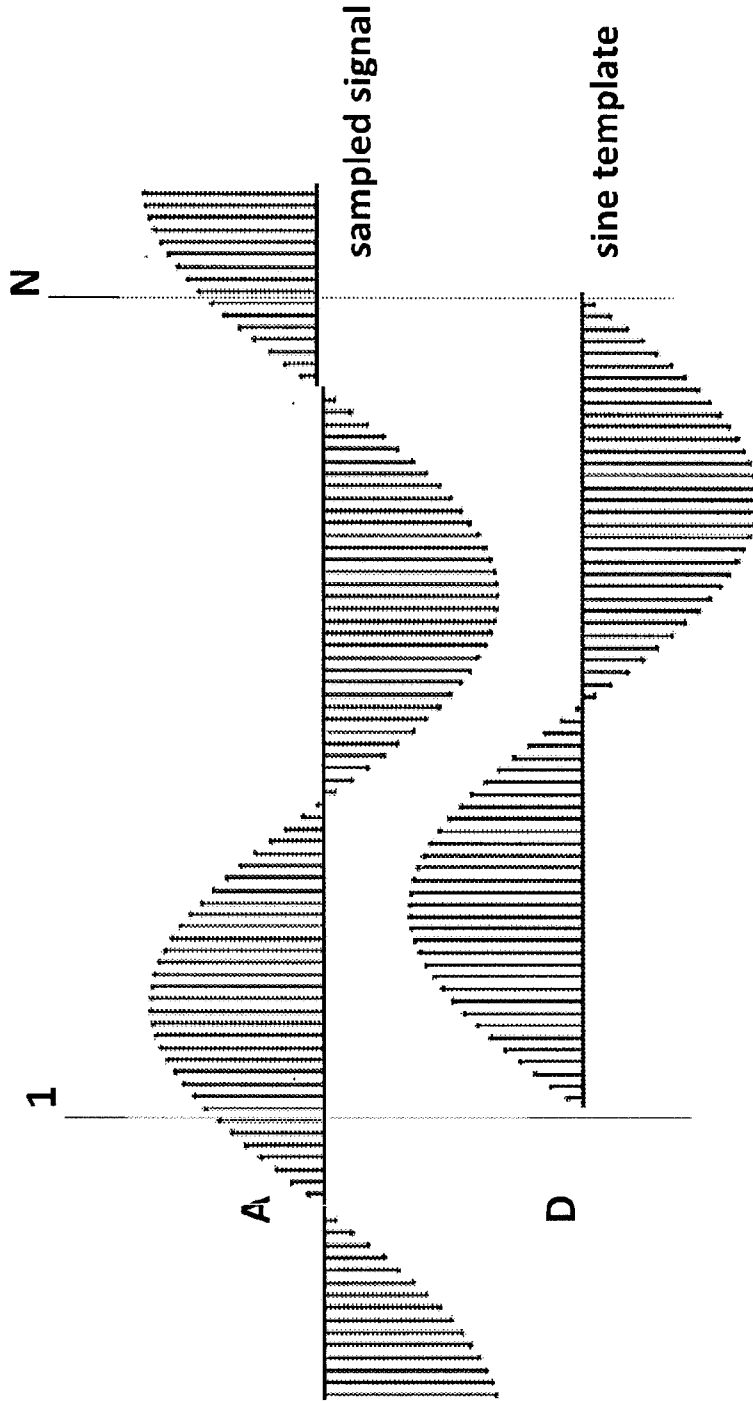
FIG. 1



$$C = A_1 \times B_1 + A_2 \times B_2 + A_3 \times B_3 + \dots + A_N \times B_N$$

$$C = \sum_{n=1}^N A_n \times B_n$$

FIG. 2A



$$S = A_1 \times D_1 + A_2 \times D_2 + A_3 \times D_3 + \dots + A_N \times D_N$$

$$S = \sum_{n=1}^N A_n \times D_n$$

FIG. 2B

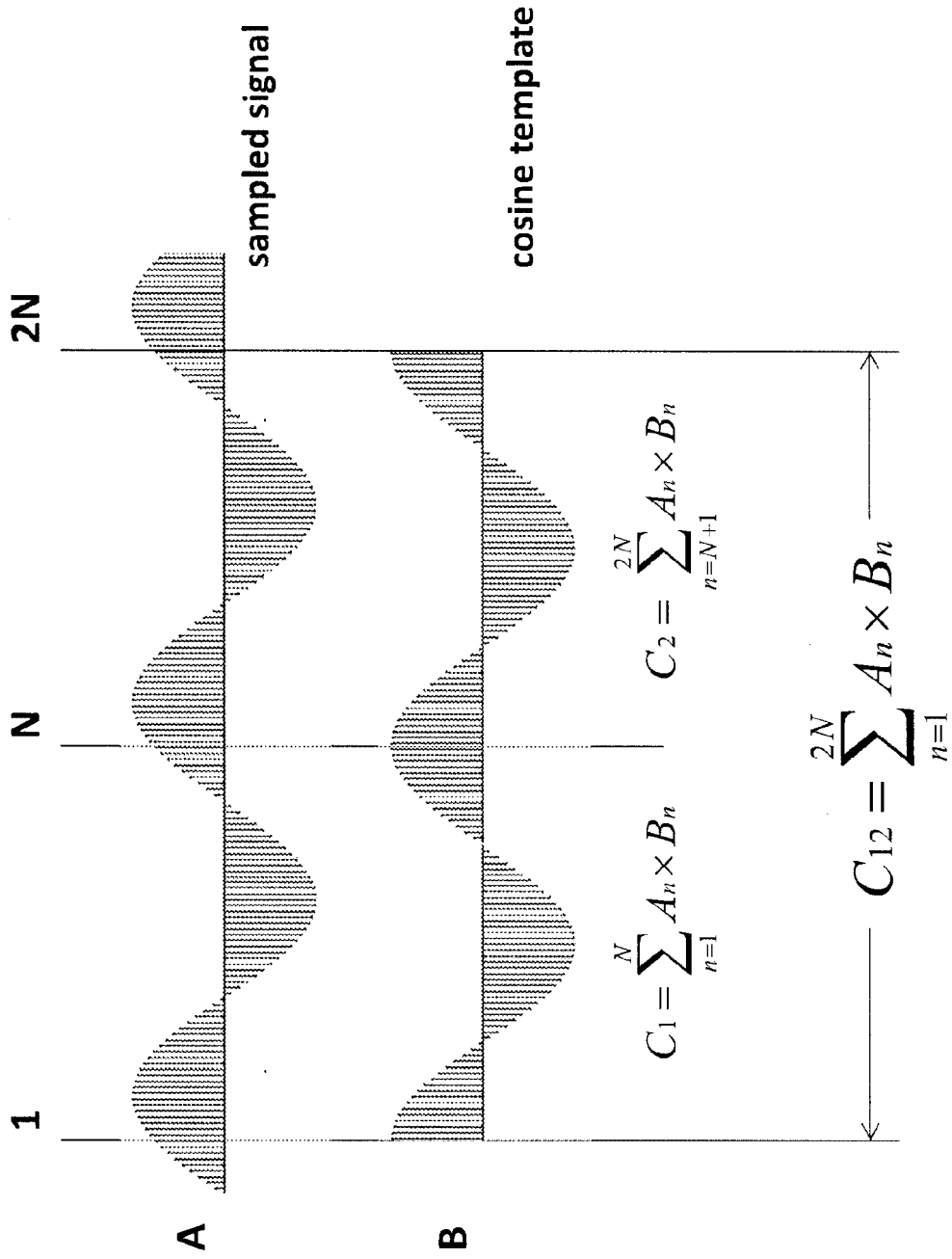
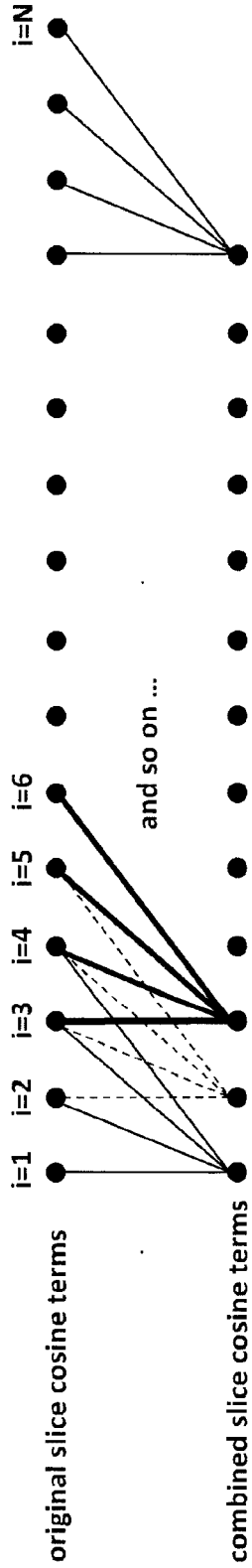


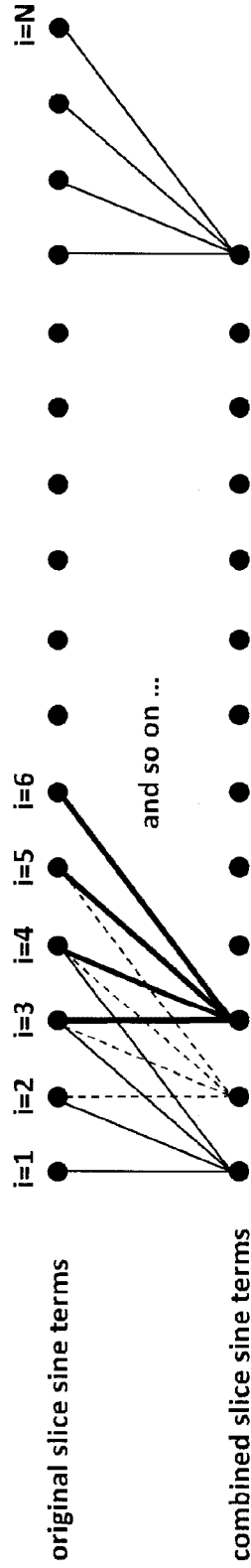
FIG. 3

$$C_{12} = C_1 + C_2$$

N is total number of slices in the slice record;
 M is the number of slices to be combined;
 index i runs from 1 to M-N+1;
 in this example, M=4 to combine 4 slices



$$\text{Combined Slice Cosine Term}_i = \sum_{k=i}^{i+M-1} \text{Original Slice Cosine Term}_k$$



$$\text{Combined Slice Sine Term}_i = \sum_{k=i}^{i+M-1} \text{Original Slice Sine Term}_k$$

FIG. 4

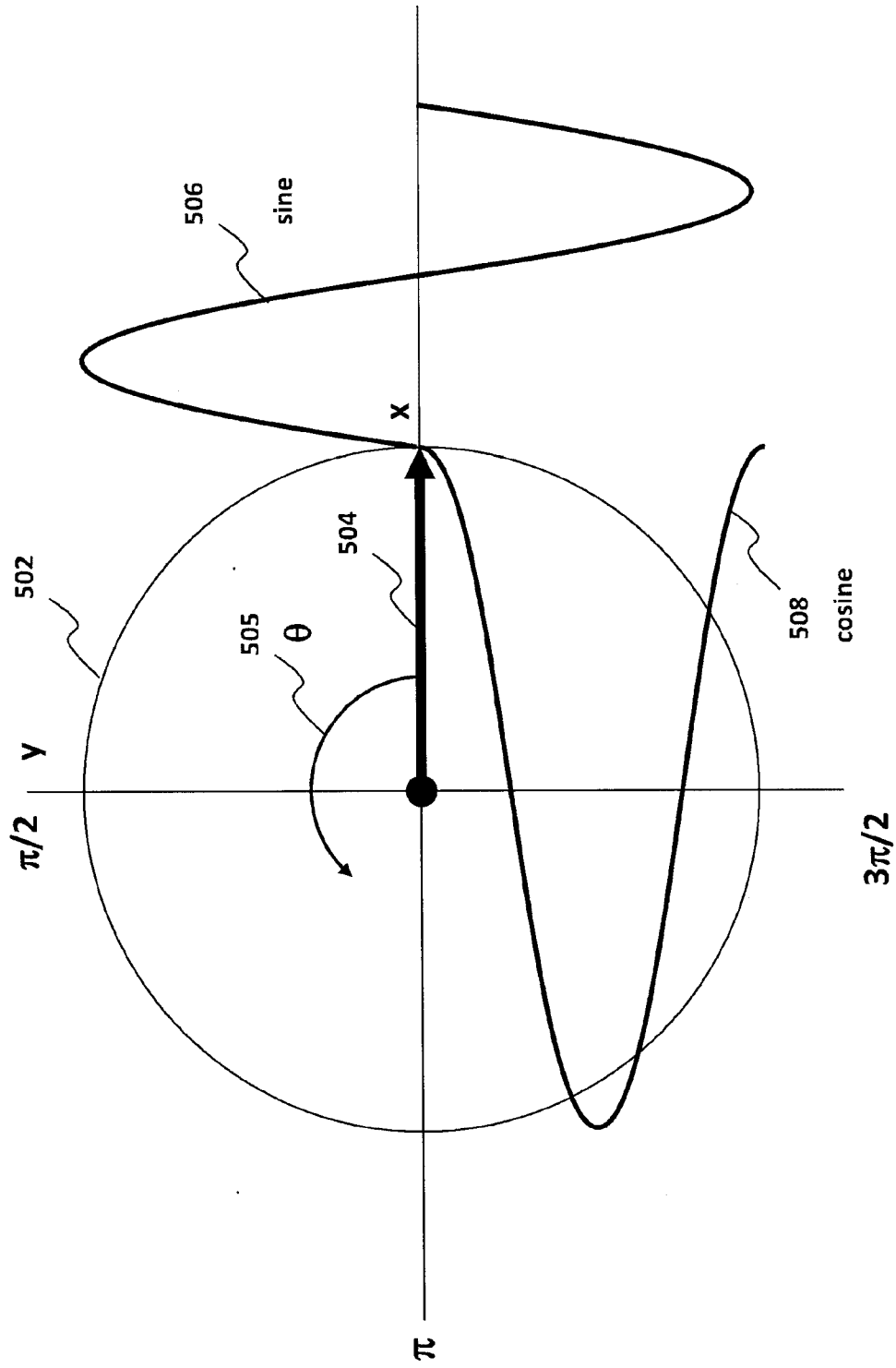


FIG. 5

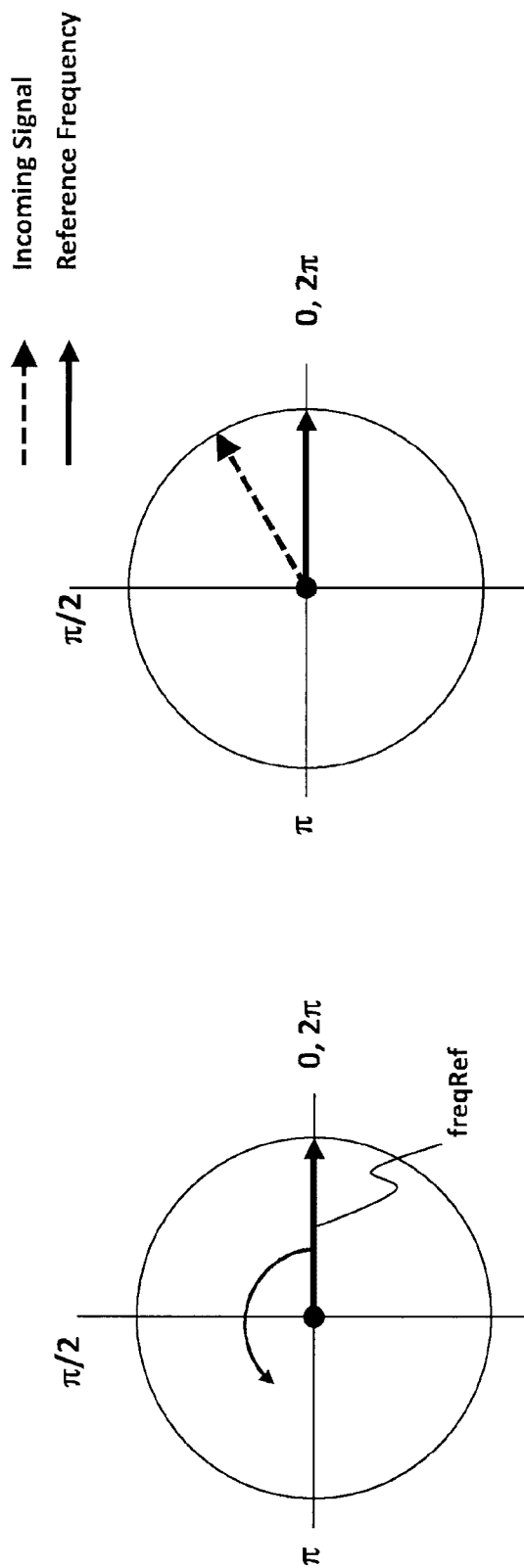


FIG. 6B

FIG. 6A

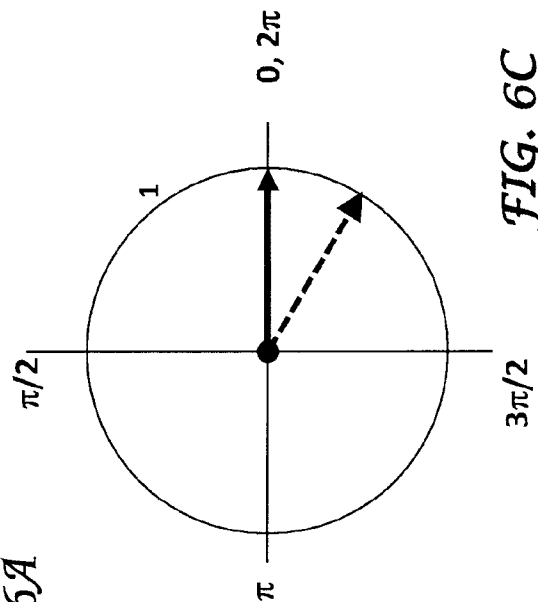


FIG. 6C

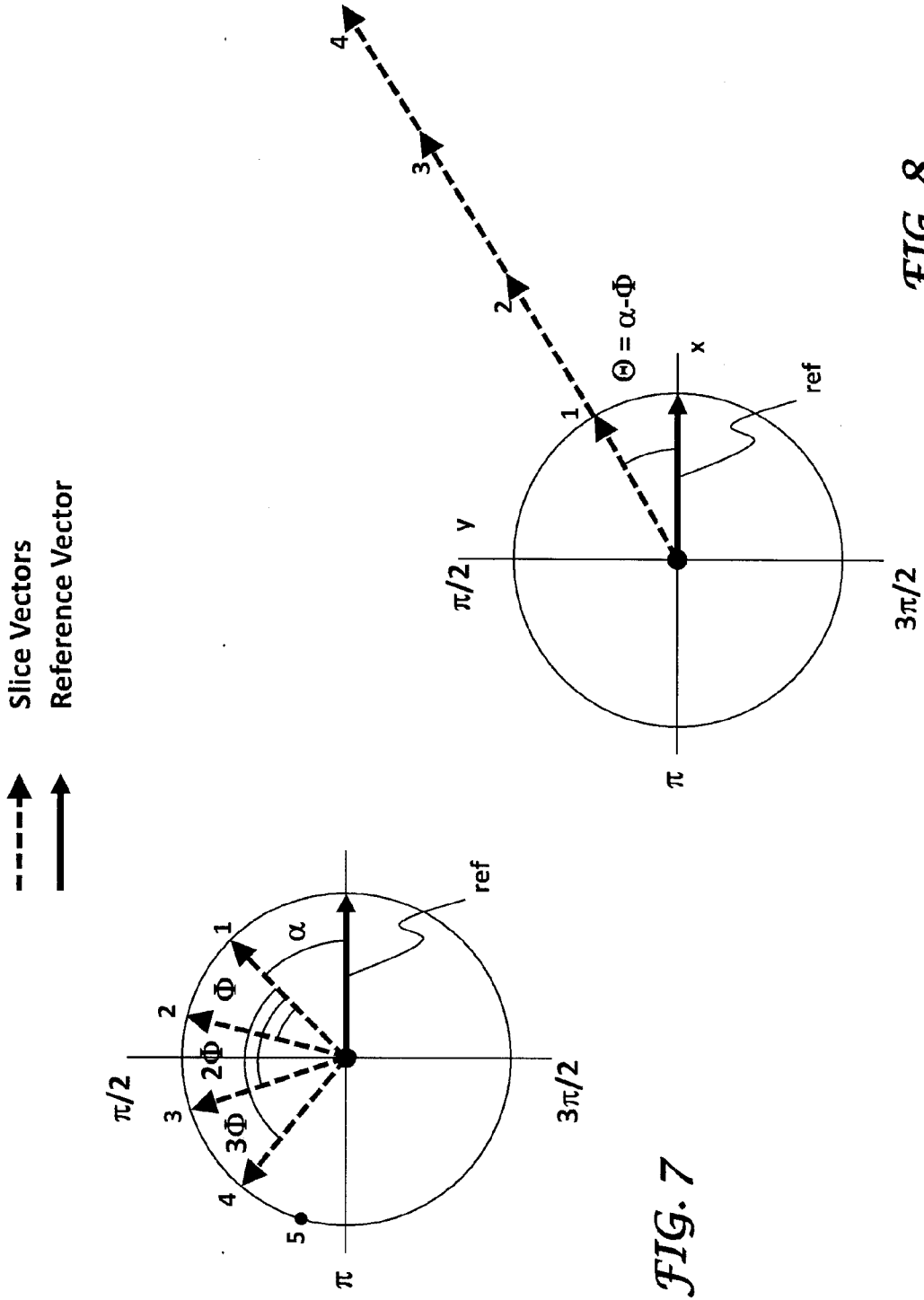


FIG. 8

FIG. 7

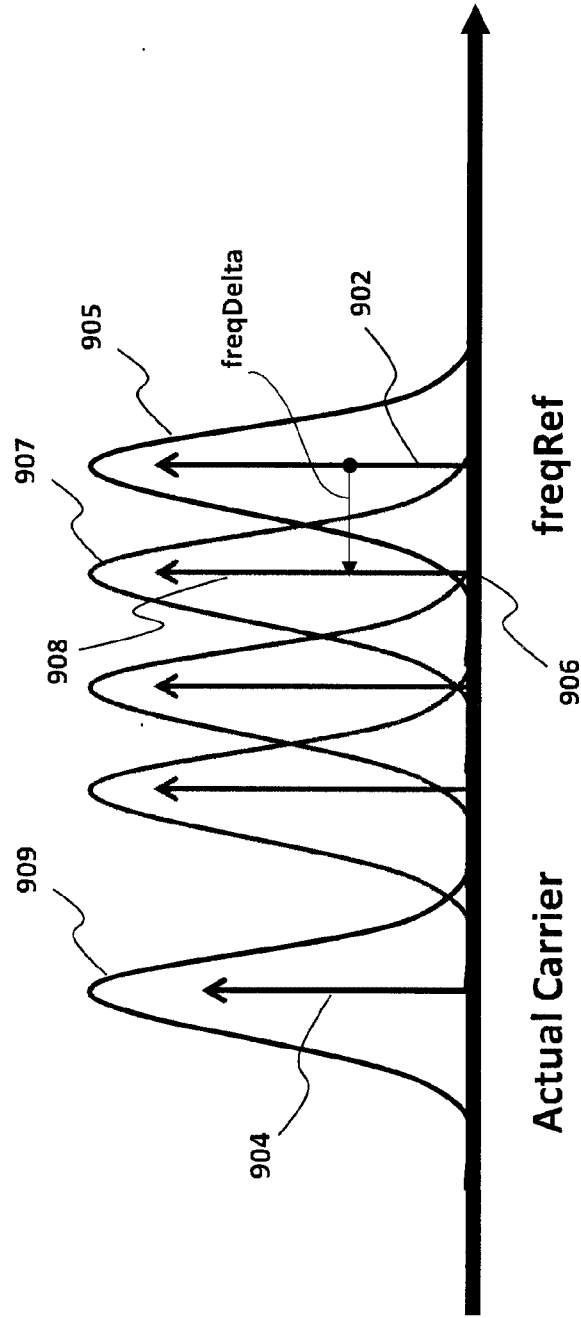
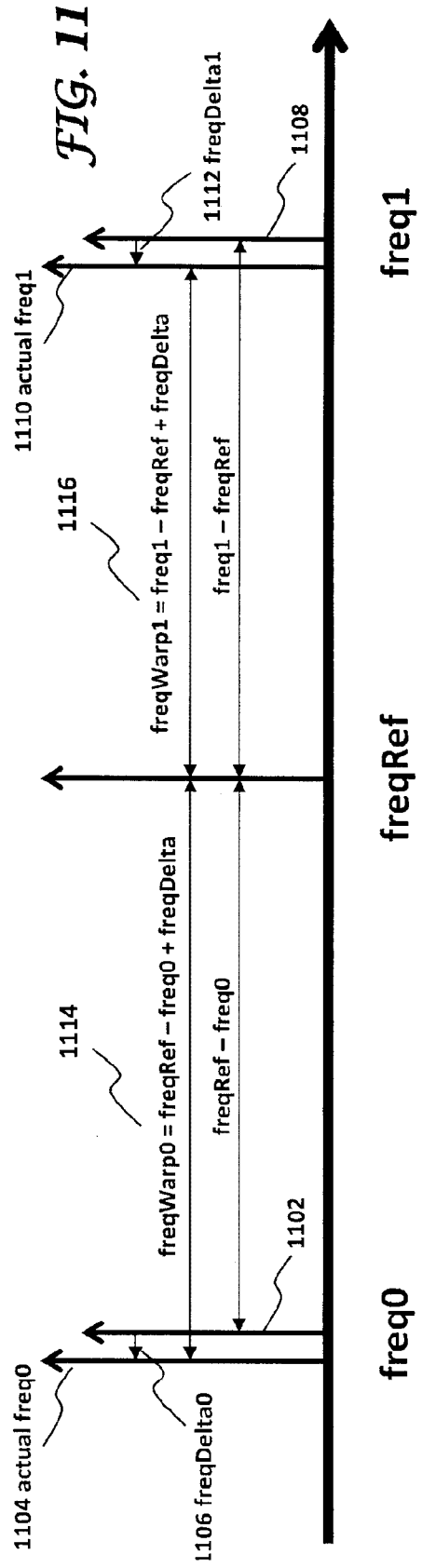
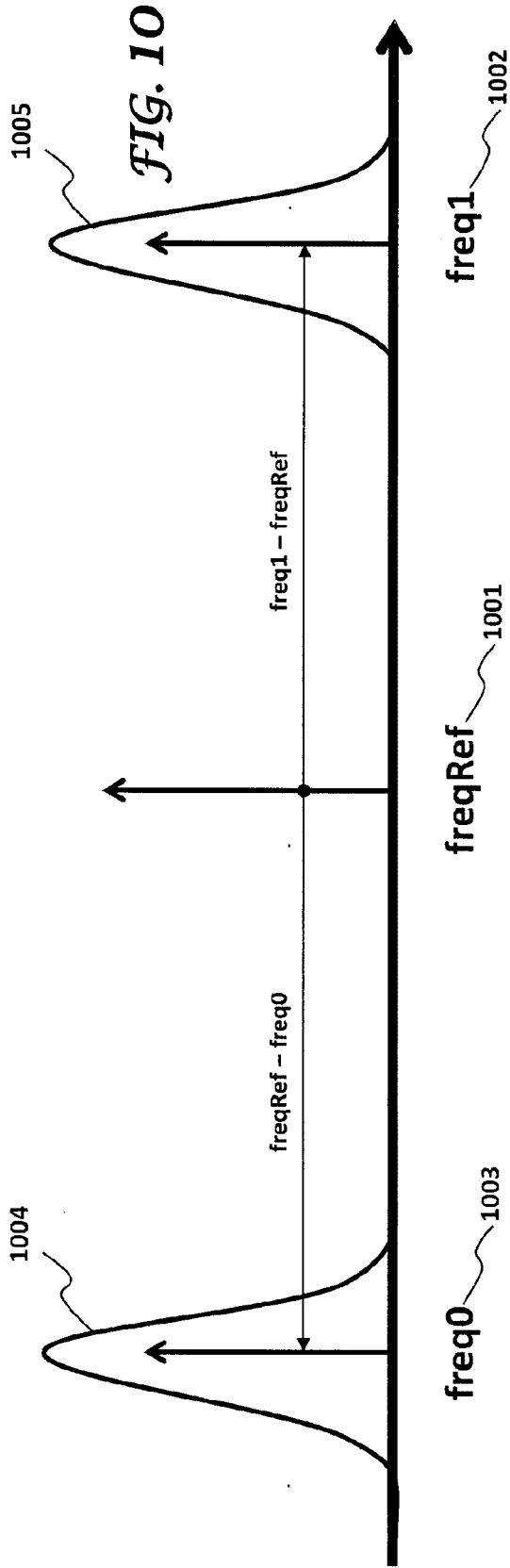


FIG. 9



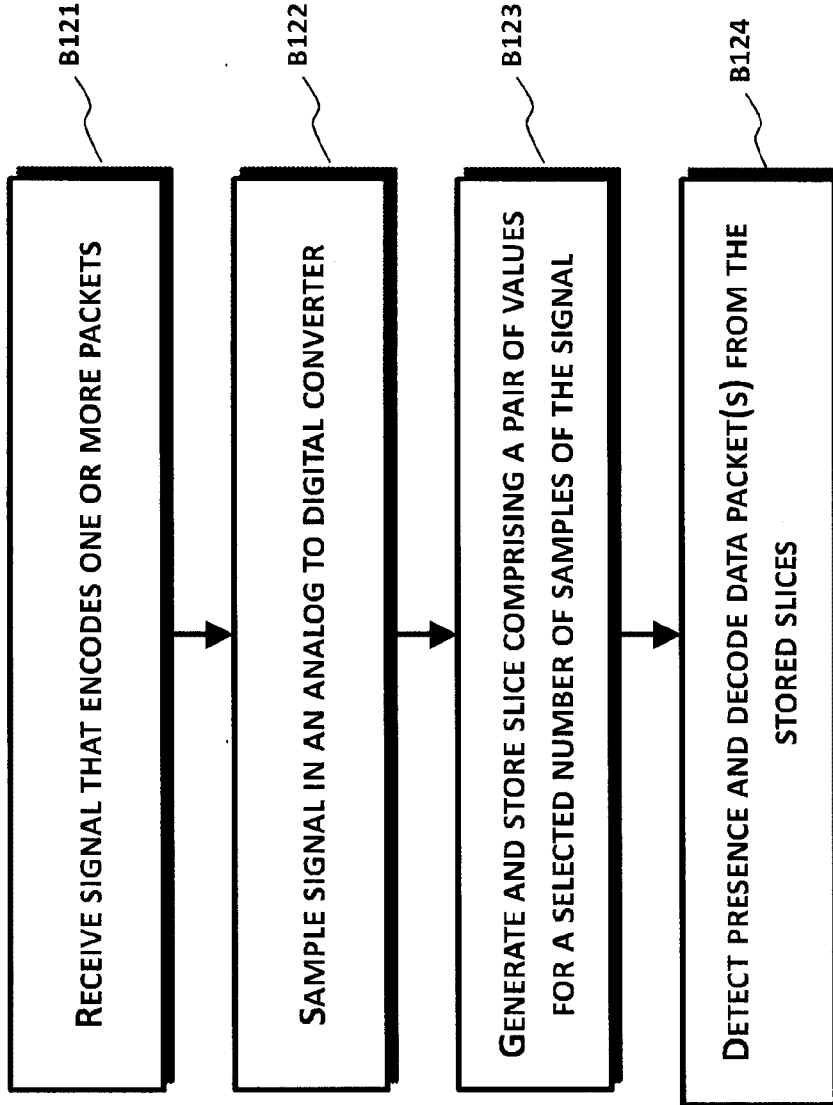


FIG. 12

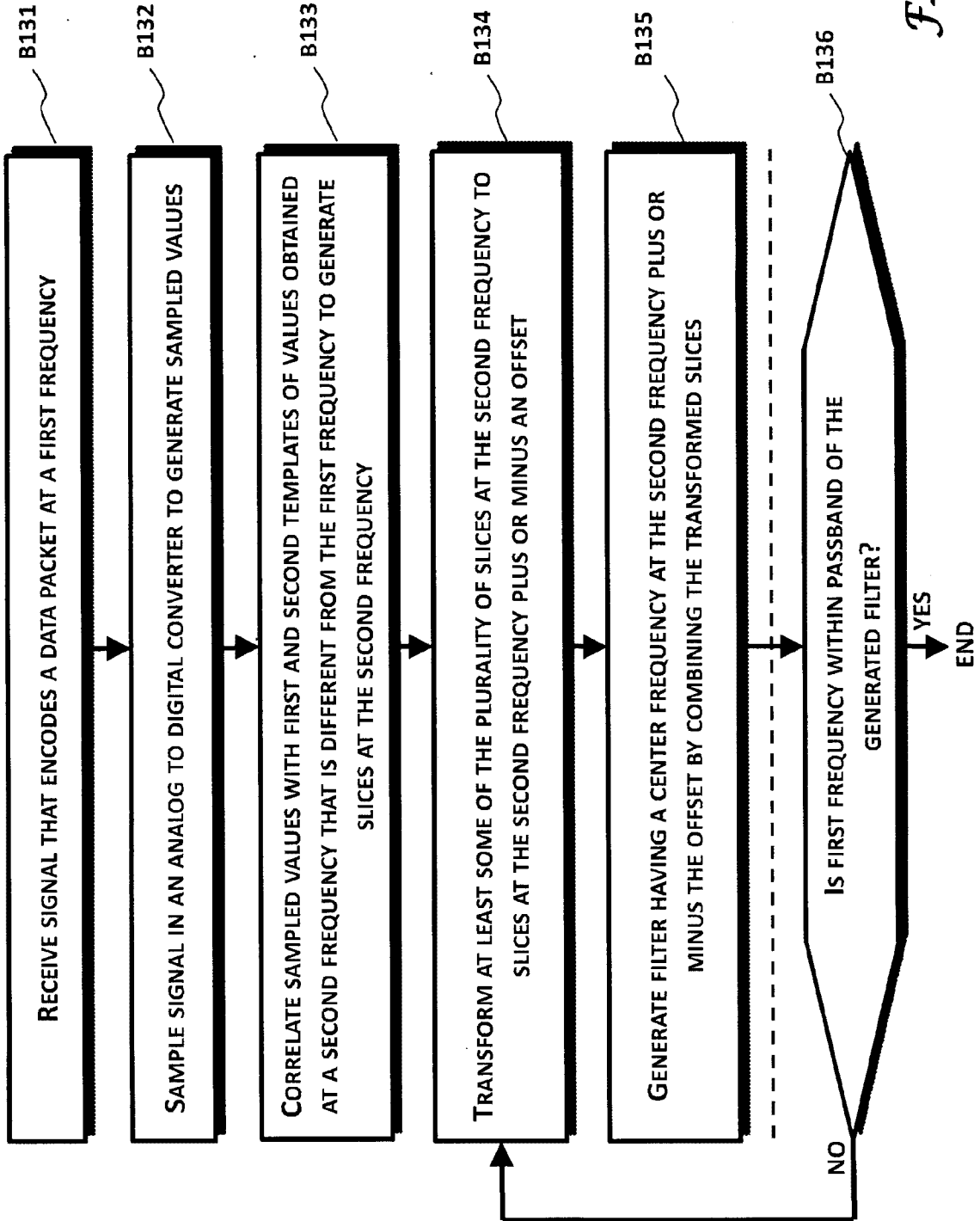


FIG. 13

RECEIVE SIGNAL THAT ENCODES ONE OR MORE PACKETS

B121

SAMPLE SIGNAL IN AN ANALOG TO DIGITAL CONVERTER

B122

GENERATE AND STORE SLICE COMPRISING A PAIR OF VALUES
FOR A SELECTED NUMBER OF SAMPLES OF THE SIGNAL

B123

DETECT PRESENCE AND DECODE DATA PACKET(S) FROM THE
STORED SLICES

B124