

METHOD AND APPARATUS FOR SYMBOL TIMING ESTIMATION IN A WIRELESS COMMUNICATIONS SYSTEM

Claim of Priority under 35 U.S.C. §119

[0001] The present Application for Patent claims priority to Provisional Application No. 60/854,877, entitled "Signal Acquisition" filed October 26, 2006, and assigned to the assignee hereof and hereby expressly incorporated by reference herein.

BACKGROUND

Field

[0002] The present disclosed systems relates generally to a system for signal acquisition in a wireless communication system, and, more specifically, to a packet detection system for detecting packets in a received signal.

Background

[0003] Wireless networking systems have become a prevalent means by which a large number of people worldwide communicate. Wireless communication devices have become smaller and more powerful to meet consumer needs, which include improved portability and convenience. Users have found many uses for wireless communication devices, such as cellular telephones, personal digital assistants (PDAs), notebooks, and the like, and such users demand reliable service and expanded coverage areas.

[0004] Wireless communications networks are commonly utilized to communicate information regardless of where a user is located (inside or outside a structure) and whether a user is stationary or moving (*e.g.*, in a vehicle, walking). Generally, wireless communications networks are established through a mobile device communicating with a base station or access point. The access point covers a geographic region or cell and, as the mobile device is operated, it may move in and out of these geographic cells. To achieve uninterrupted communication the mobile device is assigned resources of a cell it has entered and de-assigned resources of a cell it has exited.

[0005] A network can also be constructed utilizing solely peer-to-peer communication without utilizing access points. In further embodiments, the network can include both access points (infrastructure mode) and peer-to-peer communication. These types of networks are referred to as ad hoc networks). Ad hoc networks can be self-configuring

whereby when a mobile device (or access point) receives communication from another mobile device, the other mobile device is added to the network. As the mobile devices leave the area, they are dynamically removed from the network. Thus, the topography of the network can be constantly changing. In a multihop topology, a transmission is transferred through a number of hops or segments, rather than directly from a sender to a recipient.

[0006] Ultra-wideband technology such as the WiMedia ultra-wideband (UWB) common radio platform has the inherent capability to optimize wireless connectivity between multimedia devices within a wireless personal area network (WPAN). The goals of the wireless standard is to fulfill requirements such as low cost, low power consumption, small-form factor, high bandwidth and multimedia quality of service (QoS) support.

[0007] The WiMedia UWB common radio platform presents a distributed medium-access technique that provides a solution to operating different wireless applications in the same network. The WiMedia UWB common radio platform incorporates media access control (MAC) layer and physical (PHY) layer specifications based on multi-band orthogonal frequency-division multiplexing (MB-OFDM). The WiMedia MAC and PHY specifications are intentionally designed to adapt to various requirements set by global regulatory bodies. Manufacturers needing to meet regulations in various countries can thus do so easily and cost-effectively. Some other application-friendly features that WiMedia UWB attempts to implement include the reduced level of complexity per node, long battery life, support of multiple power management modes and higher spatial capacity.

[0008] WiMedia UWB-compliant receivers have to cope with interference from existing wireless services while providing large bandwidth. At the same time, they have to perform with very low transmit power. Thus, one challenge faced by receivers in an operational environment is the acquisition of a signal and, as a part thereof, establishing time synchronization with the transmitted signal. Further, being able to reliably optimize the timing estimation efficiently and with a small design footprint is a challenge.

[0009] There is therefore a need in the art for meeting the challenges noted above.

SUMMARY

- [0010] The presently described approaches are directed to timing estimation. In one approach, a method is described for performing symbol timing estimation. The method including defining a search space in a plurality of estimated magnitudes of channel taps; defining a search window in the search space; and, locating a symbol timing estimate index in the search space corresponding to a maximum value of an energy of the plurality of estimated magnitudes of channel taps within the search window.
- [0011] In another approach, an apparatus for performing symbol timing estimation is described. The apparatus includes means for defining a search space in a plurality of estimated magnitudes of channel taps; means for defining a search window in the search space; and, means for locating a symbol timing estimate index in the search space corresponding to a maximum value of an energy of the plurality of estimated magnitudes of channel taps within the search window.
- [0012] In yet another approach, a wireless communications apparatus is disclosed. The wireless communications apparatus includes an antenna configured to receive a signal having a symbol; and, a control processor coupled to the antenna for performing a method for symbol timing estimation of the symbol. The method includes defining a search space in a plurality of estimated magnitudes of channel taps; defining a search window in the search space; and, locating a symbol timing estimate index in the search space corresponding to a maximum value of an energy of the plurality of estimated magnitudes of channel taps within the search window.
- [0013] In still yet another approach, a computer program product is disclosed. The computer program product includes computer-readable medium having code for causing a computer to define a search space in a plurality of estimated magnitudes of channel taps; code for causing the computer to define a search window in the search space; and, code for causing the computer to locate a symbol timing estimate index in the search space corresponding to a maximum value of an energy of the plurality of estimated magnitudes of channel taps within the search window.
- [0014] In still yet a further approach, a processor is disclosed, the processor having a memory, the memory configured to cause the processor to implement a method for performing symbol timing estimation. The method including defining a search space in a plurality of estimated magnitudes of channel taps; defining a search window in the search space; and, locating a symbol timing estimate index in the search space

corresponding to a maximum value of an energy of the plurality of estimated magnitudes of channel taps within the search window.

BRIEF DESCRIPTION OF THE DRAWINGS

- [0015] FIG. 1 is a block diagram of an exemplary ad hoc wireless network;
- [0016] FIG. 2 is a block diagram of an exemplary wireless terminal device;
- [0017] FIG. 3 is a packet structure conforming to the WiMedia Ultra-Wideband (UWB) standard;
- [0018] FIG. 4 is a chart of the worldwide allocation of the UWB spectrum;
- [0019] FIG. 5 is a preamble structure of the packet of FIG. 3;
- [0020] FIG. 6 is a block diagram of a packet/frame synchronization sequence generator for the preamble structure of FIG. 5;
- [0021] FIG. 7 is a plot of an aperiodic auto-correlation function of a base sequence used to generate a preamble pattern;
- [0022] FIG. 8 is a block diagram of a hierarchical base sequence generator used to generate a base sequence;
- [0023] FIG. 9 is a plot of the aperiodic cross-correlation between the base sequence of FIG. 7 and the corresponding hierarchical base sequence of FIG. 8;
- [0024] FIG. 10 is a plot of the aperiodic cross-correlation between the base sequence of FIG. 7 and a rounded version of the corresponding base sequence;
- [0025] FIG. 11 is a timeline illustrating the acquisition/synchronization process for time-frequency code (TFC)-1 and TFC-2;
- [0026] FIG. 12 is a timeline illustrating the acquisition/synchronization process for TFC-3 and TFC-4;
- [0027] FIG. 13 is a timeline illustrating the acquisition/synchronization process for TFC-5, TFC-6 and TFC-7;
- [0028] FIG. 14 is a timeline illustrating the acquisition/synchronization process for TFC-8, TFC-9 and TFC-10;
- [0029] FIG. 15 is a block diagram of a synchronizer, which includes a packet detection module, a timing estimation module, and a carrier frequency offset (CFO) estimation and frame synchronization module;
- [0030] FIG. 16 is a timing estimator implementing the timing estimation module of the synchronizer of FIG. 15;

- [0031] FIG. 17 is a timing diagram illustrating the operation of the timing estimator of FIG. 16;
- [0032] FIG. 18 is a flow diagram of a timing index estimation process of the timing estimator of FIG. 16;
- [0033] FIG. 19 is a first exemplary implementation of the matched filter of the synchronizer of FIG. 15;
- [0034] FIG. 20 is a second exemplary implementation of the matched filter of the synchronizer of FIG. 15; and,
- [0035] FIG. 21 is an exemplary implementation of a L-tap multipath energy combiner used to implement a sliding window.

DETAILED DESCRIPTION

- [0036] Various embodiments are now described with reference to the drawings. In the following description, for purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of one or more aspects. It may be evident, however, that such embodiment(s) may be practiced without these specific details. In other instances, well-known structures and devices are shown in block diagram form in order to facilitate describing these embodiments.
- [0037] As used in this application, the terms “component,” “module,” “system,” and the like are intended to refer to a computer-related entity, either hardware, firmware, a combination of hardware and software, software, or software in execution. For example, a component may be, but is not limited to being, a process running on a processor, a processor, an object, an executable, a thread of execution, a program, and/or a computer. By way of illustration, both an application running on a computing device and the computing device can be a component. One or more components can reside within a process and/or thread of execution and a component may be localized on one computer and/or distributed between two or more computers. In addition, these components can execute from various computer readable media having various data structures stored thereon. The components may communicate by way of local and/or remote processes such as in accordance with a signal having one or more data packets (e.g., data from one component interacting with another component in a local system, distributed system, and/or across a network such as the Internet with other systems by way of the signal). The word “exemplary” is used herein to mean “serving as an

example, instance, or illustration.” Any embodiment described herein as “exemplary” is not necessarily to be construed as preferred or advantageous over other embodiments.

[0038] Furthermore, various embodiments are described herein in connection with a user device. A user device can also be called a system, a subscriber unit, subscriber station, mobile station, mobile device, remote station, access point, remote terminal, access terminal, terminal device, handset, host, user terminal, terminal, user agent, wireless terminal, wireless device, or user equipment. A user device can be a cellular telephone, a cordless telephone, a Session Initiation Protocol (SIP) phone, a wireless local loop (WLL) station, a Personal Digital Assistant (PDA), a handheld device having wireless connection capability, or other processing device(s) connected to a wireless modem. In certain embodiments, the user device may be a consumer electronics device with a UWB modem attached, such as printer, camera / camcorder, music player, standalone magnetic or flash storage device, or other AV equipment with content storage, for example.

[0039] Moreover, various aspects or features described herein may be implemented as a method, apparatus, or article of manufacture using standard programming and/or engineering techniques. The term “article of manufacture” as used herein is intended to encompass a computer program accessible from any computer-readable device, carrier, or media. For example, computer readable media can include but are not limited to magnetic storage devices (*e.g.*, hard disk, floppy disk, magnetic strips...), optical disks (*e.g.*, compact disk (CD), digital versatile disk (DVD)...), smart cards, and flash memory devices (*e.g.*, card, stick, key drive...).

[0040] Various embodiments will be presented in terms of systems that may include a number of devices, components, modules, and the like. It is to be understood and appreciated that the various systems may include additional devices, components, modules, and the like, and/or may not include all of the devices, components, modules and so forth, discussed in connection with the figures. A combination of these approaches may also be used.

[0041] With reference now to the drawings, FIG. 1 illustrates example ad hoc wireless network 100. Wireless network 100 can include any number of mobile devices or nodes, of which four are illustrated for ease of illustration, that are in wireless communication. Mobile devices can be, for example, cellular phones, smart phones, laptops, handheld communication devices, handheld computing devices, satellite radios,

global positioning systems, Personal Digital Assistants (PDAs), and/or other suitable devices for communicating over wireless network 100. Wireless network 100 can also include one or more base stations or access points (not shown).

[0042] In wireless network 100, terminal device 112 is shown communicating with terminal device 114 via communication link 120 and with terminal device 116 via communication link 112. Terminal device 116 is also shown communicating with terminal device 118 via communication link 124. Terminal devices 112, 114, 116 and 118 may be structured and configured in accordance with the exemplary simplified block diagram of a possible configuration of a terminal device 200 as shown in FIG. 2. As those skilled in the art will appreciate, the precise configuration of terminal device 200 may vary depending on the specific application and the overall design constraints. Processor 202 can implement the systems and methods disclosed herein.

[0043] Terminal device 200 can be implemented with a front-end transceiver 204 coupled to an antenna 206. A baseband processor 208 can be coupled to the transceiver 204. The baseband processor 208 can be implemented with a software based architecture, or other type of architectures, such as hardware or a combination of hardware and software. A microprocessor can be utilized as a platform to run software programs that, among other functions, provide control and overall system management function. A digital signal processor (DSP) can be implemented with an embedded communications software layer, which runs application specific algorithms to reduce the processing demands on the microprocessor. The DSP can be utilized to provide various signal processing functions such as pilot signal acquisition, time synchronization, frequency tracking, spread-spectrum processing, modulation and demodulation functions, and forward error correction.

[0044] Terminal device 200 can also include various user interfaces 210 coupled to the baseband processor 208. User interfaces 210 can include a keypad, mouse, touch screen, display, ringer, vibrator, audio speaker, microphone, camera, storage and/or other input/output devices.

[0045] The baseband processor 208 comprises a processor 202. In a software-based implementation of the baseband processor 208, the processor 202 may be a software program running on a microprocessor. However, as those skilled in the art will readily appreciate, the processor 202 is not limited to this embodiment, and may be implemented by any means known in the art, including any hardware configuration,

software configuration, or combination thereof, which is capable of performing the various functions described herein. The processor 202 can be coupled to memory 212 for the storage of data. An application processor 214 for executing application operating system and/or separate applications may also be provided as shown in FIG. 2. Application processor 214 is shown coupled to baseband processor 208, memory 212, and user interface 210.

[0046] FIG. 3 illustrates a packet structure 300 of a packet conforming with the WiMedia Ultra-Wideband (UWB) physical layer (PHY) and media access layer (MAC) standard for high rate, short range wireless communication as promulgated by ECMA International in Standard ECMA-368, "High Rate Ultra Wideband PHY and MAC Standard" (December 2005).

[0047] The ECMA Standard specifies a UWB PHY for a wireless personal area network (PAN) utilizing the unlicensed 3,100 – 10,600 MHz frequency band, supporting data rates of 53,3 Mb/s, 80 Mb/s, 106,7 Mb/s, 160 Mb/s, 200 Mb/s, 320 Mb/s, 400 Mb/s, and 480 Mb/s. The UWB spectrum is divided into 14 bands, each with a bandwidth of 528 MHz. The first 12 bands are then grouped into 4 band groups consisting of 3 bands, and the last two bands are grouped into a fifth band group. FIG. 4 illustrates a worldwide allocation of the UWB spectrum.

[0048] This ECMA Standard specifies a multiband orthogonal frequency division modulation (MB-OFDM) scheme to transmit information. A total of 110 sub-carriers (100 data carriers and 10 guard carriers) are used per band to transmit the information. In addition, 12 pilot subcarriers allow for coherent detection. Frequency-domain spreading, time-domain spreading, and forward error correction (FEC) coding are used to vary the data rates. The FEC used is a convolutional code with coding rates of 1/3, 1/2, 5/8 and 3/4.

[0049] The coded data is then spread using a time-frequency code (TFC). In one approach, as promulgated by the ECMA standard, there are two types of time-frequency codes (TFCs): one where the coded information is interleaved over three bands, referred to as Time-Frequency Interleaving (TFI); and, one where the coded information is transmitted on a single band, referred to as Fixed Frequency Interleaving (FFI).

[0050] Within each of the first four band groups, four time-frequency codes using TFI and three time-frequency codes using FFI are defined; thereby, providing support for up

to seven channels per band. For the fifth band group, two time-frequency codes using FFI are defined. This ECMA Standard specifies 30 channels in total.

[0051] FIG. 5 illustrates the standard preamble structure of the WiMedia UWB packet of FIG. 3. The preamble contains a total of 30 OFDM symbols. The first 24 preamble symbols are used for packet detection, timing estimation, CFO estimation and frame synchronization. Channel estimation uses the last 6 preamble symbols. In one approach, the first 24 symbols are of primary importance.

[0052] FIG. 6 is a block diagram of a preamble symbol generator 600, including a spreader 602, illustrating one approach of how preamble symbols may be generated, where:

[0053] 1. For a given a time-frequency code (TFC) (i.e., 1-10, referred to as TFC-1 to TFC-10), select the time-domain base sequence $s_{\text{base}}[m]$, $m = 0, 1, \dots, 127$ and the binary cover sequence $s_{\text{cover}}[n] = \pm 1$, $n = 0, 1, \dots, 23$. The binary cover sequence is used as a delimiter for determining the ending of the packet/frame synchronization sequence.

[0054] 2. Pad 37 zeros at the end of the base sequence to form the extended sequence $s_{\text{ext}}[k]$, $k = 0, 1, \dots, 164$.

[0055] 3. Spread the cover sequence with the extended based sequence using the spreader 602. The k th sample of the n th preamble symbol is given by:

$$s_{\text{sync},n}[k] = s_{\text{cover}}[n] \times s_{\text{ext}}[k], \quad k = 0, 1, \dots, 164, \quad n = 0, 1, \dots, 23.$$

[0056] FIG. 7 illustrates the aperiodic auto-correlation of the base sequence $s_{\text{base}}[m]$ corresponding to TFC-1. Other base sequences may have similar auto-correlation functions. In one synchronization approach, the excellent auto-correlation property is exploited. For example, the base sequence is generated from a hierarchical base sequence generator 800 as shown in FIG. 8. The basic premise behind using a hierarchical sequences is to partition the encoding process at the transmitter into a hierarchy so that the complexity of the decoding process at the receiver is reduced. Referring to the figure, a first binary sequence $\{a[k], k = 0, 2, \dots, 15\}$ is spread by a second binary sequence $\{b[k], k = 0, 2, \dots, 7\}$ with a spreader 802 to generate an intermediate sequence (also referred to as a binary hierarchical sequence) C $\{c[k], k = 0, 2, \dots, 127\}$ of length 128. Then, after taking a fast Fourier transform (FFT) of the intermediate sequence C using an FFT module 804 and shaping the

sequence in the frequency domain using a frequency domain shaping module 806, the sequence is transformed back to the time domain via an inverse FFT (IFFT) module 808 to obtain the base sequence $s_{\text{base}}[m]$. There is a unique set of binary sequences $\{a[k]\}$ and $\{b[k]\}$ corresponding to each of the ten base sequences.

[0057] FIG. 9 illustrates the aperiodic cross-correlation between the base sequence $s_{\text{base}}[m]$ for TFC-1 and the corresponding intermediate sequence $C \{c[k]\}$ generated using the hierarchical base sequence generator 800. This cross-correlation property indicates that when a matched filter is employed at the receiver, the base sequence can be replaced by the binary sequence C as the filter coefficients. In one approach, as illustrated below, the hierarchical structure of the binary sequence C can be efficiently used to simplify the hardware of the receiver used for synchronization. Further, it may be advantageous to use the rounded version of the preamble base sequence as the matched filter coefficients as well. FIG. 10 illustrates the aperiodic cross-correlation between the base sequence $s_{\text{base}}[m]$ for TFC-1 and the rounded version of the corresponding base sequence.

[0058] As a synchronization overview, FIG. 11 – FIG. 14 illustrate the synchronization and acquisition timelines for all the TFCs. Specifically, FIG. 11 illustrates an acquisition timeline 1100 for TFC-1 and TFC-2; FIG. 12 illustrates an acquisition timeline 1200 for TFC-3 and TFC-4; FIG. 13 illustrates an acquisition timeline 1300 for TFC-5, TFC-6 and TFC-7; and FIG. 14 illustrates an acquisition timeline 1400 for TFC-8, TFC-9 and TFC-10.

[0059] Referring initially to FIG. 11, the major synchronization tasks can be separated into three separate parts:

- [0060] 1. Packet detection.
- [0061] 2. Timing estimation.
- [0062] 3. Carrier frequency offset (CFO) estimation and frame synchronization.

[0063] As discussed above, the ECMA standard provides for multiple bands and, as seen from the timelines for all TFCs, a receiver will by default dwell on Band-1 before packet detection is asserted. This is because before packet detection, the receiver has no knowledge about the correct timing to switch to other bands (if it is in the TFI mode). Thus, the first three preamble symbols in Band-1 will be consumed for packet detection. Once packet detection has been completed, the next phase, timing estimation, is enabled and the receiver will scan for the next preamble symbol in Band-1 to determine the

optimal FFT window for the OFDM symbol. After timing estimation has been completed (e.g., the timing is recovered) for Band-1, the receiver will have enough information to know to switch to other bands according to the TFC, and automatic gain control (AGC) gain estimation will be performed. After AGC is settled, the rest part of the preamble symbols will be used for CFO estimation and frame sync detection. Whenever frame sync is detected, the final output of the CFO estimation will be sent to a phase rotator and the receiver will proceed with channel estimation.

[0064] FIG. 15 illustrates a synchronizer 1500 for performing the major synchronization tasks. The synchronizer 1500 includes a variable gain amplifier (VGA) module 1502, an analog-to-digital converter (ADC) 1504, a matched filter (MF) 1506, a squaring unit 1508, a packet detection module 1510, a timing estimation module 1540 and a CFO estimation and frame synchronization module 1570.

[0065] The coefficients $\{q[k], k = 0, 2, \dots, 127\}$ of the MF 1506 can be chosen either as the binary sequence $\{c[k], k = 0, 2, \dots, 127\}$ or the rounded preamble base sequence $\{\text{round}(s_{\text{base}}[k]), k = 0, 2, \dots, 127\}$, as discussed above. Due to the hierarchical structure of the binary sequence $\{c[k]\}$, however, the implementation of the MF 1506 may be simplified as shown in a binary hierarchical sequence MF 1900 of FIG. 19; while for the rounded version, a finite impulse response (FIR) implementation MF 2000 is shown in FIG. 20, which in one approach is an FIR filter with 127 tapped delay lines.

[0066] In the rounded approach, the matched filter coefficients $q[k], k = 0, 2, \dots, 127$ is set to the rounded version of the preamble base sequence $\text{Round}(s_{\text{base}}[k])$. As observed for all the preamble base sequences, $\text{Round}(s_{\text{base}}[k])$ only takes values from $\{\pm 2, \pm 1, 0\}$, which helps to reduce the hardware complexity as multiplication by 2 can be conveniently implemented as left shifting 1 bit. Also, as seen in FIG. 10, $\text{Round}(s_{\text{base}}[k])$ maintains good cross-correlation property with the base sequence $s_{\text{base}}[k]$. The complexity of the two different methods for the matched filter implementation is summarized in the following table:

Matched Filter Type	Number of Real Multiplications	Number of Real Additions	LUT Size (bits)
Binary Hierarchical	0	22	$10 \times (16+8) = 240$
Rounded Base Sequence	0	127	$10 \times 128 \times 3 = 3840$

[0067] **Table 1: Matched filter implementation comparison.**

[0068] The number of operations is for either I or Q branch within one sample duration $T_{\text{sample}} = 1/528\text{MHz} = 1.89\text{ns}$. For each approach, the reference sequences can be stored in a lookup table (LUT) of the size as listed in Table 1.

[0069] The output of the MF 1506 is processed by the squaring unit 1508. Denoting the received samples as $r[n]$, the magnitude square of the matched filter output may be expressed as:

$$[0070] \quad R[n] = \left| \sum_{k=0}^{127} r[n+k] \cdot q[k] \right|^2,$$

[0071] It is noted that an equal gain combining (EGC) operation may be performed to collect the energy from the multipath channels:

$$[0072] \quad D[n] = \sum_{m'=n}^{n+N-1} R[m'],$$

[0073] where N is the number of consecutive paths that are combined and $D[n]$ is the sliding window output. The EGC may be implemented as an L-tap multipath energy combiner 2100 as shown in FIG. 21. The L-tap multipath energy combiner 2100 allows a different weight to be assigned to each tap. The results of the EGC operation may be used by the packet detection module 1510 and the timing estimation module 1540.

[0074] As discussed, the first step in the synchronization process is for the packet detection module 1510 to detect the presence of a valid packet. The packet detection module 1510 will assert a packet detection signal to the timing estimation module 1540 after a valid packet has been detected. Specifically, once packet detection is asserted (i.e., the packet detection module 1510 has indicated that a packet has been detected by setting the *det_flag* to a logical true), the timing estimation module 1540 is enabled.

[0075] FIG. 16 illustrates an exemplary timing estimator 1600 that may be implemented for the timing estimation module 1540. The timing estimator 1600 includes a sliding window (SW) unit 1650 and a timing detector 1660. As will be explained using FIG. 17, the timing estimator 1600 attempts to locate the best location of an FFT window such that it can capture as much of the channel energy as possible.

[0076] As discussed above, the EGC operation may be performed to collect energy for multipath channels. In WiMedia UWB, the length of the zero padding (ZP) for each OFDM symbol is 32 paths, which means that there are at most $ZP+1=33$ consecutive paths that can be captured. Thus, a sliding window of 33 (i.e., $N = 33$) is used in the SW unit 1650 for the timing estimation and the EGC operation may be deployed using

the 33-unit wide SW unit 1650 implemented as the L-tap multipath energy combiner 2100.

[0077] The input to the SW unit 1650 is the magnitude square of the output, (i.e., the sequence $\{R[n]\}$ from the squaring unit 1508). Basically, each component in this sequence is an estimate to the squared amplitude of one of the channel tap coefficients. For the ideal scenario when SNR goes to infinity and the cross-correlation between the preamble pattern and the binary sequence is a perfect delta function, it becomes exactly the squared amplitude of the channel tap coefficient. FIG. 17 illustrates how the SW unit 1650 operates on this sequence for a number of $N3$ samples, which is a complete period in Band-1. $N3$ is a TFC-dependent parameter and is listed in the following table:

TFC Number	N3
1, 2	$165 \times 3 = 495$
3, 4	165
5, 6, 7	165
8, 9, 10	$165 \times 2 = 330$

[0078] **Table 2: TFC-Dependent Parameter for Timing Estimation.**

[0079] The maximum value among the $N3$ outputs from the SW unit 1650 is the most energy that can be captured, and the corresponding index is identified as the value of a variable *TIMING_INDEX*. In one approach, the operation of the timing detector 1660 may be implemented by the following pseudo code:

```
[0080]     TIMING_INDEX = 0;
[0081]     MAX_ENERGY_CAP = 0;
[0082]     for(step = 0; step < N3; step++)
[0083]     {
[0084]         if (input[step] > MAX_ENERGY_CAP)
[0085]         {
[0086]             MAX_ENERGY_CAP = input[step];
[0087]             TIMING_INDEX = step;
[0088]         }
[0089]     }
```

[0090] where input[step] is the output from the SW unit 1650 (which is the output of the multipath combiner for 33 samples), and MAX_ENERGY_CAP is the highest detected value of the captured energy from the SW unit 1650 in the $N3$ outputs encountered up to the current iteration.

- [0091] FIG. 18 illustrates a timing index detection process 1800 of the timing detector 1660, where, in step 1802, the variables of *TIMING_INDEX*, *MAX_ENERGY_CAP*, and *step* are set to 0. In step 1804, it is determined if the current value of the *step* variable is less than *N3*, which is the number of samples in a full period to be processed by the timing detector 1660 (e.g., 495 samples for Band-1 as illustrated in FIG. 17). If so, operation continues with step 1806. Otherwise, operation continues with step 1814.
- [0092] In step 1806, a value *EGC_out* is output from the multipath combiner (i.e., SW unit 1650), and operation continues with step 1808, where it is determined if the value of *EGC_out* is greater than the current value of *MAX_ENERGY_CAP*. If so, then *MAX_ENERGY_CAP* is set to *EGC_out* in step 1810. *TIMING_INDEX* is set to be the current value of *step* at this point.
- [0093] If the value of *EGC_out* is not greater than the current value of *MAX_ENERGY_CAP*, then operation continues with step 1812, where the window for the multipath combiner is slid for one sample, and *step* is incremented by 1. Operation then returns to step 1804.
- [0094] Returning to step 1804, if it is determined if the current value of the *step* variable is not less than *N3*, which indicates that the multipath combiner window has been slid across all *N3* outputs, and the value of the variable *TIMING_INDEX* is output.
- [0095] Using *TIMING_INDEX* and counting the group delay of the MF 1506 and the SW unit 1650, the receiver can easily locate the start of the FFT window (i.e., the starting point of one OFDM symbol) in Band-1. Then, the receiver will switch bands according to the TFC (if in TFI mode), which implies that the timing information obtained from Band-1 is directly applied to other bands.
- [0096] It is to be understood that the embodiments described herein may be implemented by hardware, software, firmware, middleware, microcode, or any combination thereof. When the systems and/or methods are implemented in software, firmware, middleware or microcode, program code or code segments, they may be stored in a machine-readable medium, such as a storage component. A code segment may represent a procedure, a function, a subprogram, a program, a routine, a subroutine, a module, a software package, a class, or any combination of instructions, data structures, or program statements. A code segment may be coupled to another code segment or a hardware circuit by passing and/or receiving information, data, arguments, parameters, or memory contents. Information, arguments, parameters, data, *etc.* may be

passed, forwarded, or transmitted using any suitable means including memory sharing, message passing, token passing, network transmission, *etc.*

[0097] For a software implementation, the techniques described herein may be implemented with modules (*e.g.*, procedures, functions, and so on) that perform the functions described herein. The software codes may be stored in memory units and executed by processors. The memory unit may be implemented within the processor or external to the processor, in which case it can be communicatively coupled to the processor through various means as is known in the art.

[0098] What has been described above includes examples of one or more embodiments. It is, of course, not possible to describe every conceivable combination of components or methodologies for purposes of describing the aforementioned embodiments, but one of ordinary skill in the art may recognize that many further combinations and permutations of various embodiments are possible. Accordingly, the described embodiments are intended to embrace all such alterations, modifications and variations that fall within the spirit and scope of the appended claims. Furthermore, to the extent that the term “includes” is used in either the detailed description or the claims, such term is intended to be inclusive in a manner similar to the term “comprising” as “comprising” is interpreted when employed as a transitional word in a claim.

CLAIMS

WHAT IS CLAIMED IS:

1. A method for performing symbol timing estimation comprising:
 - defining a search space in a plurality of estimated magnitudes of channel taps;
 - defining a search window in the search space; and,
 - locating a symbol timing estimate index in the search space corresponding to a maximum value of an energy of the plurality of estimated magnitudes of channel taps within the search window.
2. The method of claim 1, wherein the search window has a size based on a length of a padding for an orthogonal frequency division multiplexing (OFDM) symbol.
3. The method of claim 2, wherein the padding is a zero padding.
4. The method of claim 1, wherein the search space is based on a frequency hopping pattern.
5. The method of claim 4, wherein the frequency hopping pattern comprises a time frequency code (TFC).
6. The method of claim 1, further comprising combining a predetermined number of estimated magnitude of channel taps within the search window to determine the energy.
7. The method of claim 6, further comprising assigning a weight to each of the predetermined number of estimated magnitude of channel taps within the search window.
8. An apparatus for performing symbol timing estimation comprising:
 - means for defining a search space in a plurality of estimated magnitudes of channel taps;
 - means for defining a search window in the search space; and,
 - means for locating a symbol timing estimate index in the search space corresponding to a maximum value of an energy of the plurality of estimated magnitudes of channel taps within the search window.

9. The apparatus of claim 8, wherein the search window has a size based on a length of a padding for an orthogonal frequency division multiplexing (OFDM) symbol.
10. The apparatus of claim 9, wherein the padding is a zero padding.
11. The apparatus of claim 8, wherein the search space is based on a frequency hopping pattern.
12. The apparatus of claim 11, wherein the frequency hopping pattern comprises a time frequency code (TFC).
13. The apparatus of claim 8, further comprising means for combining a predetermined number of estimated magnitude of channel taps within the search window to determine the energy.
14. The apparatus of claim 13, further comprising means for assigning a weight to each of the predetermined number of estimated magnitude of channel taps within the search window.
15. A wireless communications apparatus comprising:
 - an antenna configured to receive a signal having a symbol; and,
 - a control processor coupled to the antenna for performing a method for symbol timing estimation of the symbol, the method comprising:
 - defining a search space in a plurality of estimated magnitudes of channel taps;
 - defining a search window in the search space; and,
 - locating a symbol timing estimate index in the search space corresponding to a maximum value of an energy of the plurality of estimated magnitudes of channel taps within the search window.
16. The wireless communications apparatus of claim 15, wherein the search window has a size based on a length of a padding for an orthogonal frequency division multiplexing (OFDM) symbol.
17. The wireless communications apparatus of claim 16, wherein the padding is a zero padding.

18. The wireless communications apparatus of claim 15, wherein the search space is based on a frequency hopping pattern.

19. The wireless communications apparatus of claim 18, wherein the frequency hopping pattern comprises a time frequency code (TFC).

20. The wireless communications apparatus of claim 15, further comprising combining a predetermined number of estimated magnitude of channel taps within the search window to determine the energy.

21. The wireless communications apparatus of claim 20, further comprising assigning a weight to each of the predetermined number of estimated magnitude of channel taps within the search window.

22. A computer program product, comprising:

computer-readable medium comprising:

code for causing a computer to define a search space in a plurality of estimated magnitudes of channel taps;

code for causing the computer to define a search window in the search space; and,

code for causing the computer to locate a symbol timing estimate index in the search space corresponding to a maximum value of an energy of the plurality of estimated magnitudes of channel taps within the search window.

23. A processor, comprising:

a memory, the memory configured to cause the processor to implement a method for performing symbol timing estimation, the method comprising:

defining a search space in a plurality of estimated magnitudes of channel taps;

defining a search window in the search space; and,

locating a symbol timing estimate index in the search space corresponding to a maximum value of an energy of the plurality of estimated magnitudes of channel taps within the search window.