(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) **International Patent Classification:**
H03H 7/30 (2006.01)

(21) **International Application Number:**
PCT/US2008/069433

(22) **International Filing Date:** 8 July 2008 (08.07.2008)

(25) **Filing Language:** English

(26) **Publication Language:** English

(30) **Priority Data:**
11/776,233 11 July 2007 (11.07.2007) US

(71) **Applicant** *(for all designated States except US)*: **TECHWELL , INC.** [US/US]; 408 East Plumeria Drive, San Jose, CA 95134 (US).

(72) **Inventors; and**
(75) **Inventors/Applicants** *(for US only)*: **FIMOFF, Mark** [US/US]; 1020 Concord Lane, Hoffman Estates, IL 60192 (US). **KIM, Jin** [US/US]; 550 Timberwood Drive, Lake Zurich, IL 60047 (US). **TOMEZAK, Gregory, J.** [US/US]; 601 Checker Drive, Buffalo Grove, IL 60090 (US).

(74) **Agent: COTRONE, Carlo, M.**; Michael Best & Friedrich LLP, 100 East Wisconsin Avenue, Suite 3300, Milwaukee, WI 53202-4108 (US).

(81) **Designated States** *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) **Designated States** *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MT, NL, NO, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*

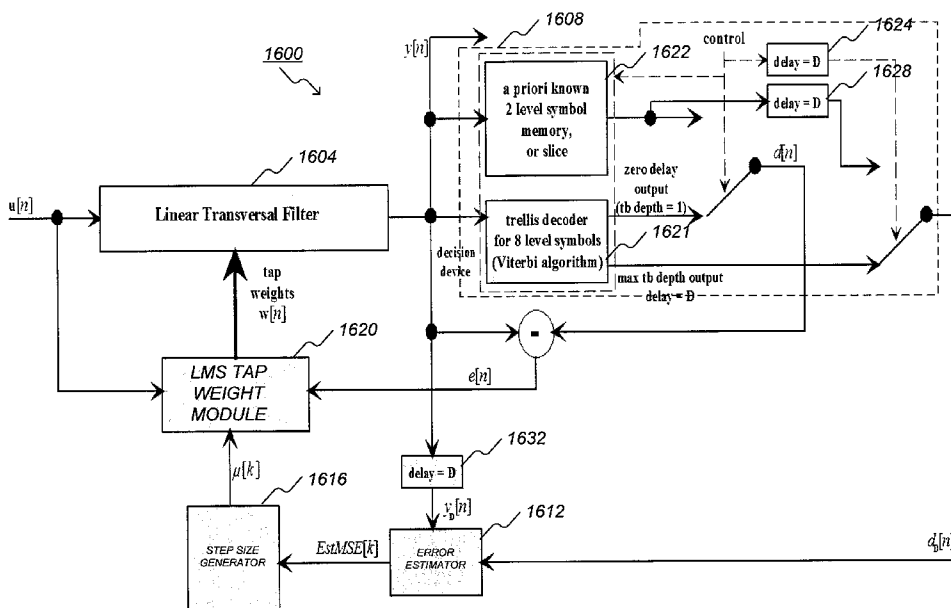(54) **Title:** METHOD FOR CHANNEL TRACKING IN AN LMS ADAPTIVE EQUALIZER FOR 8VSB

FIG. 16

(57) **Abstract:** A method and system for channel tracking in an adaptive equalizer for a digital data receiver. The method includes determining a first set of tap weights and a second set of tap weights of the equalizer, and determining a difference between the first and second sets of tap weights. The method also includes determining an error estimate based on symbols received at the receiver, comparing the error estimate with a divergence threshold, and determining a step size factor based on the difference and the comparison.

METHOD FOR CHANNEL TRACKING IN
AN LMS ADAPTIVE EQUALIZER FOR 8VSB

RELATED APPLICATIONS

[0001]   This application is a continuation-in-part of prior-filed copending U.S. Patent
Application Serial No. 11/687,909, filed on March 19, 2007, which claims the benefit of
prior-filed U.S. Provisional Patent Application Serial No. 60/885,692, filed on January 19,
2007, the entire contents of both of which are incorporated by reference herein.

FIELD

[0002]   Embodiments of the invention relate generally to digital communication systems
and methods, and particularly to digital television receivers.

BACKGROUND

[0003]   In 1996, the Advanced Television Systems Committee, Inc. ("ATSC") adopted an
ATSC digital television ("DTV") terrestrial transmission standard. Several generations of
receivers have been developed since adoption of the ATSC DTV standard. Generally, each
generation of receivers was developed to improve reception performance over previous
generations of receivers. A main impediment to good reception is severe multipath
interference. Hence, complicated equalizers were developed for receivers in order to improve
receiver performance by mitigating the effects of the multipath interference.

[0004]   Terrestrial broadcast DTV channel presents quite a difficult multipath
environment. Relatively strong duplicates of the transmitted signal may arrive at a receiver
via various reflected signal paths as well as via the direct path from transmitter to receiver. In
some cases, there is no direct path from transmitter to receiver, and all received signal paths
are via reflection. If the path carrying the strongest signal is regarded as the main signal path,
reflected signals may arrive at the receiver both prior to or subsequent to the main signal.
The arrival time differences among various signal paths, compared to that of the main signal
path, can be large. Also, these reflected signals may vary in time, both in terms of amplitude
and delay relative to the main signal path.

[0005]   During a typical transmission, data is transmitted in frames 100 as shown in FIG. 1. Each frame 100 is composed of two fields 104, 108. Each of the fields 104, 108 includes 313 segments. Each of the segments includes 832 symbols. As such, each of the fields 104, 108 includes a total of 260,416 symbols. Each of the segments begins with a four-symbol sequence, referred to as a segment sync, which comprises four symbols [+5, -5, -5, +5]. The first segment in each field is a field sync segment.

[0006]   FIG. 2 shows an exemplary field sync segment 200 of the field 104 or 108 of FIG. 1. The field sync segment 200 includes a segment sync 204, a pseudo noise sequence 208 that comprises 511 symbols (PN511), a pseudo noise sequence 212 that comprises 63 symbols (PN63), a second PN63 sequence 216, and a third PN63 sequence 220. The third PN63 sequence 220 is followed by a mode sequence 224 that comprises 24 symbols to indicate a transmitting mode of 8-level vestigial sideband ("8VSB"). In alternate fields, the three PN63 sequences 212, 216, 220 are the same. In the remaining fields, the first and third PN63 sequences 212, 220 are the same while the second PN63 sequence 216 is inverted. In either case, the first 728 symbols of the field sync segment 200 are a priori known to a receiver and may be used for equalizer training. The mode sequence 224 is followed by a reserved mode sequence 228 of 92 symbols composing various mode and reserved fields that are not a priori known to the receiver. The sequences 204, 208, 212, 216, 220, 224, and 228 symbols use a symbol set of {+5, -5}. The field sync segment 200 ends with a precode sequence 232 comprising 12 symbols that use a symbol set of {-7, -5, -3, -1, +1, +3, +5, +7}, and are duplicates of the last 12 symbols of the preceding data field. These are thus called precode symbols.

[0007]   The remaining 312 segments of each field 104, 108 are referred to as data segments. An exemplary data segment 300 is shown in FIG. 3. After the segment sync symbols 204, the data segment 300 includes a data sequence 304 that comprises 828 symbols. The symbols are trellis encoded by a 12 phase trellis encoder that results in 8-level symbols from a symbol set of {-7, -5, -3, -1, +1, +3, +5, +7}.

[0008]   FIG. 4 shows a digital data (e.g., 8VSB) transmitter 400. The transmitter 400 includes a randomizer 404 that randomizes data to be transmitted, a Reed-Solomon encoder 408 that encodes the randomized data from the randomizer 404, and an interleaver 412 that interleaves Reed-Solomon byte-wise encoded data. The transmitter 400 also includes a trellis encoder 416 that encodes the interleaved data. An exemplary trellis encoder 416 is a 12-

phase trellis encoder. A data frame formatter 420 subsequently adds segment sync symbols and field sync symbols to the trellis coded data at appropriate times to create a data frame structure like that of FIG. 1. A pilot insertion module 424 then inserts a pilot carrier frequency signal by adding a fixed DC level to each of the symbols.

[0009]   A modulator 428 then implements root raised cosine pulse shaping and modulates the signal for RF transmission as an 8VSB signal at a symbol rate of 10.76 MHz. The 8VSB signal differs from other commonly used linear modulation methods such as quadrature amplitude modulation ("QAM") in that the 8VSB symbols are real, but have a pulse shape that is complex with only the real part of the pulse having a Nyquist shape.

[0010]   FIG. 5 shows a block diagram of a digital data (e.g., 8VSB) receiver 500. The receiver 500 includes a tuner 504 to receive RF signals transmitted from the transmitter 400, and a demodulator 508 to demodulate the RF signal to baseband. The receiver 500 also includes a sync and timing recovery module 512 to perform symbol clock timing and frame synchronization recovery on the demodulated signals. The receiver 500 also includes a matched filter 516 to filter the recovered signals, an equalizer 520 that equalizes the filtered signals, a phase tracker 524 that reduces the phase noise of the equalized signals, a trellis decoder 528 that decodes the noise-reduced equalized signals, a deinterleaver 532 that deinterleaves the decoded signals, a Reed-Solomon decoder 536 that decodes the deinterleaved signals, and a derandomizer 540 that derandomizes the decoded signals.

[0011]   The multipath RF channel between the transmitter 400 and the receiver 500 can be viewed in its baseband equivalent form. For example, the transmitted signal has a root raised cosine spectrum with a nominal bandwidth of 5.38 MHz and an excess bandwidth of 11.5% centered at one fourth of the symbol rate (i.e., 2.69 MHz). Thus, the transmitted pulse shape or pulse $q(t)$ is complex and given by EQN. (1):

$$q(t) = e^{j\pi F_s t/2} q_{RRC}(t) \qquad\qquad (1)$$

where F_s is a symbol frequency, and $q_{RRC}(t)$ is a real square root raised cosine pulse with an excess bandwidth of 11.5% of the multipath RF channel. The pulse $q(t)$ is referred to as a "complex root raised cosine pulse." For an 8VSB system, the transmitted pulse shape $q(t)$ and the received and matched filter pulse shape $q^*(-t)$ are identical since $q(t)$ is conjugate-

symmetric. Thus, the raised cosine pulse *p(t)*, referred to as the "complex raised cosine pulse," is given by EQN. (2):

$$p(t) = q(t) * q^*(-t) \tag{2}$$

where * denotes convolution, and * denotes complex conjugation.

[0012]   The transmitted baseband signal with a data rate of 1/T symbols/sec can be represented by EQN. (3):

$$s(t) = \sum_k I_k q(t - kT), \tag{3}$$

where $\{I_k \in A \equiv \{\alpha_1, \dots \alpha_8\} \subset R^1\}$ is a transmitted data sequence, which is a discrete 8-ary sequence taking values of the real 8-ary alphabet A. For 8VSB, the alphabet set is {-7, -5, -3, -1, +1, +3, +5, +7}.

[0013]   A physical channel between the transmitter 400 and the receiver 500 is denoted *c(t)* and can be described by

$$c(t) = \sum_{k=-L_{ha}}^{L_{hc}} c_k \delta(t - \tau_k) \tag{4}$$

where $\{c_k(\tau)\} \subset C^1$, and L_{ha} and L_{hc} are the maximum number of anti-causal and causal multipath delays, respectively. Constant τ_k is a multipath delay, and variable $\delta(t)$ is a Dirac delta function. Hence, the overall channel impulse response is given by EQN. (5):

$$h(t) = p(t) * c(t) = \sum_{-L_{ha}}^{L_{hc}} c_k p(t - \tau_k) \tag{5}$$

[0014]   The matched filter output *y(t)* in the receiver prior to equalization is given by EQN. (6):

$$y(t) = \left(\sum_k \delta(t - kT) \right) * h(t) + v(t), \tag{6}$$

where *v(t)* is given by EQN. (7):

4

$$v(t) = \eta(t) * q^*(-t) \tag{7}$$

which denotes a complex or colored noise process after the pulse matched filter, with $\eta(t)$ being a zero-mean white Gaussian noise process with spectral density σ_n^2 per real and imaginary part. Sampling the matched filter output $y(t)$ at the symbol rate produces a discrete time baseband representation of the input to the equalizer 520, as shown in EQN. (8):

$$y[n] \equiv y(t)\big|_{t=nT} = \sum_k I_k h[n-k] + v[n]. \tag{8}$$

[0015]   As stated above, for each data field of 260,416 symbols, only 728 symbols, which reside in the field sync segment 200, are a priori known and thus available for equalizer training. Furthermore, conditions of the multipath channel are generally not known a priori. As such, the equalizer 520 in the receiver 500 is so configured to adaptively identify and combat various multipath channel conditions.

[0016]   In the following discussion, n represents a sample time index, regular type represents scalar variables, bold lower case type represents vector variables, bold upper case type represents matrix variables, a * superscript indicates complex conjugation, and the ^{H} superscript indicates conjugate transposition (Hermitian).

[0017]   The equalizer 520 may be implemented as, or employ equalization techniques relating to, linear equalizers ("LEs"), decision feedback equalizers ("DFEs"), and predictive decision feedback equalizers ("pDFEs"). Equalizer tap weight adaptation is often achieved via a least mean square ("LMS") algorithm or system, which is a low complexity method for adaptively approximating a minimum mean squared error ("MMSE") tap weight solution, or equivalently a solution to the Weiner Hopf equations, described below.

[0018]   In the case of an LE, let $\mathbf{u}[n]$ be an N long equalizer input vector, $\mathbf{y}[n]$ be the equalizer output $\mathbf{w}^H[n]\mathbf{u}[n]$, where $\mathbf{w}^H[n]$ is an N long equalizer tap weight vector of a linear transversal filter or an adaptive filter,

$$\mathbf{R}_{uu}[n] = E\big(\mathbf{u}[n]\mathbf{u}^H[n]\big) \text{ has a size of } N \times N, \text{ and}$$

$$\mathbf{r}_{du} = E\big(\mathbf{u}[n]d^*[n]\big)$$

Then $e[n] = d[n] - y[n]$ where $d[n]$ is the desired symbol.

[0019]   The mean squared error ("MSE") is given by $J = E(e[n]e^*[n])$. It can be shown that the MSE as a function of filter taps $\mathbf{w}$, $J(\mathbf{w})$, is given by (n index omitted for clarity) EQN. (9):

$$J(\mathbf{w}) = \sigma_d^2 - \mathbf{w}^H \mathbf{r}_{du} - \mathbf{r}_{du}^H \mathbf{w} + \mathbf{w}^H \mathbf{R}_{uu} \mathbf{w} \tag{9}$$

A gradient vector of $J(\mathbf{w})$ is given by EQN. (10):

$$\nabla J(\mathbf{w}) = 2 \frac{\partial J(\mathbf{w})}{\partial \mathbf{w}^*} = \begin{bmatrix} \dfrac{\partial J}{\partial w_0^R} + j \dfrac{\partial J}{\partial w_0^I} \\ \dfrac{\partial J}{\partial w_1^R} + j \dfrac{\partial J}{\partial w_1^I} \\ - \\ - \\ \dfrac{\partial J}{\partial w_{N-1}^R} + j \dfrac{\partial J}{\partial w_{N-1}^I} \end{bmatrix} = 2(\mathbf{R}_{uu}\mathbf{w} - \mathbf{r}_{du}) \tag{10}$$

[0020]   An optimal MMSE tap vector $\mathbf{w}_{opt}$ is found by setting $\nabla J(\mathbf{w}) = 0$, yielding the Weiner Hopf tap weight solution given by EQN. (11):

$$\mathbf{w}_{opt}[n] = \mathbf{R}_{uu}^{-1}[n]\mathbf{r}_{du}[n] \tag{11}$$

The MSE is generally a measure of the closeness of $\mathbf{w}$ to $\mathbf{w}_{opt}$. As a function of the tap weight vector $\mathbf{w}$, the MSE is then given by EQN. (12):

$$J(\mathbf{w}) = J_{\min} + (\mathbf{w} - \mathbf{w}_{opt})^H \mathbf{R}_{uu} (\mathbf{w} - \mathbf{w}_{opt}) \tag{12}$$

where

$$J_{\min} = \min_{\mathbf{w}} J(\mathbf{w}) = \sigma_d^2 - \mathbf{r}_{du}^H \mathbf{R}_{uu}^{-1} \mathbf{r}_{du} = J(\mathbf{w}_{opt}) \tag{12}$$

In practice, for large N, inverting $\mathbf{R}_{uu}$ is prohibitively complicated. So a less complicated iterative solution is desirable. A steepest descent method ("SD") provides such a solution. It is given by EQN. (13):

$$\mathbf{w}[n+1] = \mathbf{w}[n] - \mu\{\nabla J(\mathbf{w}[n])\} = \mathbf{w}[n] - \mu[\,\mathbf{R}_{uu}[n]\mathbf{w}[n] - \mathbf{r}_{du}[n]\,]$$

$$(13)$$

where μ is a step size parameter. However, estimating and updating $\mathbf{R}_{uu}$ and $\mathbf{r}_{du}$ can also be complicated.

[0021]   By using instantaneous approximations for $\mathbf{R}_{uu}$ and $\mathbf{r}_{du}$, EQN. (13) can be greatly simplified for practical applications. For example, as shown in EQN. (14) and EQN. (15),

$$\mathbf{R}_{uu}[n] \approx \mathbf{u}[n]\mathbf{u}^{H}[n]$$

$$(14)$$

and

$$\mathbf{r}_{du} \approx \mathbf{u}[n]d^{*}[n],$$

$$(15)$$

the gradient can be given by EQN. (16):

$$\nabla J(\mathbf{w}[n]) = 2(\mathbf{R}_{uu}[n]\mathbf{w}[n] - \mathbf{r}_{du}[n])$$
$$\approx -2\mathbf{u}[n]\big[d^{*}[n] - \mathbf{u}^{H}[n]\mathbf{w}[n]\big]$$
$$\approx -2\mathbf{u}[n]e^{*}[n]$$

$$(16)$$

A practical LMS algorithm for the equalizer 520, as shown in EQN. (17), can then be determined from EQN. (13):

$$\mathbf{w}[n+1] = \mathbf{w}[n] - \mu\{\nabla J(\mathbf{w}[n])\}$$
$$= \mathbf{w}[n] - 2\mu\mathbf{u}[n]e^{*}[n]$$

$$(17)$$

where μ is a step size parameter.

[0022] FIG. 6 shows a signal-to-noise-plus-interference-ratio ("SINR") plot 600 depicting a plurality of SINRs obtained from a plurality of symbol blocks with an LMS-based LE as discussed above. Similarly, FIG. 7 shows an MSE plot 700 depicting a plurality of MSEs obtained from the plurality of symbol blocks with the LMS-based LE as discussed above. It is well known that SINR and MSE are related as shown in EQN. (18):

$$SINR = 10 \log_{10} \frac{signal\ power}{MSE}$$

(18)

As shown in FIG. 6, time (in terms of symbol blocks processed by the equalizer 520) is measured along an x-axis 604, and SINR is measured along a y-axis 608. FIG. 6 shows an SINR curve 612 for different times after symbols are equalized with the equalizer 520. FIG. 6 shows that the curve 612 converges to an SINR value of about 15 dB after about 3,000 symbol blocks have been equalized, where each block includes 512 symbols. Similarly, as shown in FIG. 7, time is measured along an x-axis 704, and MSE is measured along a y-axis 708. FIG. 7 shows an MSE curve 712 for different times after symbols are equalized with the equalizer 520. FIG. 7 shows that the MSE curve 712 converges to an MSE value of about 1 dB after about 3,000 symbol blocks have been equalized.

[0023] In general, equalizer convergence is achieved when the SINR rises above a prescribed value before approaching a SINR convergence value such that subsequent error correction modules, such as the trellis decoder 528 and the Reed-Solomon decoder 536, can nearly completely correct all data errors. For 8VSB, the prescribed value is about 15 dB, and the SINR convergence value, which will depend on channel conditions, must be larger than that prescribed value. An example is shown in FIG. 6, where values of SINR rise above 15 dB before approaching an SINR convergence value of about 16 dB.

[0024] FIG. 8 shows an LE system 800 that utilizes the LMS algorithm as discussed. The LE system 800 includes a linear transversal filter 804 with tap weights **w** fed by an input data vector **u**[n]. The filter 804 has an output $y[n]$ that feeds a non-linear decision device 808. The decision device 808 has an output $d[n]$ that is a set of likely symbols transmitted. The output $d[n]$ is subtracted from the output $y[n]$ to create an error signal $e[n]$. The error signal $e[n]$ is used by an LMS algorithm 812 to update the tap weights **w** of the filter 804 for time n + 1.

SUMMARY

[0025]   The following summary sets forth certain exemplary embodiments of the invention. It does not set forth all embodiments of the invention and should in no way be construed as limiting of embodiments of the invention.

[0026]   In one embodiment, the invention includes a method of channel tracking in an adaptive equalizer for a digital data receiver. The method includes determining a first set of tap weights and a second set of tap weights of the equalizer, and determining a difference between the first and second sets of tap weights. The method also includes determining an error estimate based on symbols received at the receiver, comparing the error estimate with a divergence threshold, and determining a step size factor based on the difference and the comparison.

[0027]   In another embodiment, the invention includes an adaptive equalizer for a digital data receiver, wherein data received by the receiver includes coded symbols. The equalizer includes a tap weight module, a difference module, an error estimator, a comparison module, and a step size factor module. The tap weight module determines a first set of tap weights and a second set of tap weights of the equalizer. The difference module determines a difference between the first and second sets of tap weights. The error estimator determines an error estimate based on symbols received at the receiver. The comparison module compares the error estimate with a divergence threshold, and the step size factor module determines a step size factor based on the difference and the comparison.

[0028]   In another embodiment, the invention includes a device configured to process digital television signals. The device includes a receiver that includes a demodulator, a decoder, and an equalizer. The receiver receives radio frequency signals modulated with data including coded symbols and uncoded symbols, the demodulator demodulates the received radio frequency signals to produce the coded symbols and the uncoded symbols, and the decoder decodes the coded symbols to produce corresponding decoded symbols. The equalizer includes a tap weight module, a difference module, an error estimator, a comparison module, and a step size factor module. The tap weight module determines a first set of tap weights and a second set of tap weights of the equalizer. The difference module determines a difference between the first and second sets of tap weights. The error estimator determines an error estimate based on symbols received at the receiver. The comparison module

compares the error estimate with a divergence threshold, and the step size factor module determines a step size factor based on the difference and the comparison.

[0029]   Other aspects of the invention will become apparent by consideration of the detailed description and accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

[0030]   FIG. 1 shows an exemplary 8VSB data frame.

[0031]   FIG. 2 shows an exemplary field sync segment.

[0032]   FIG. 3 shows an exemplary data segment.

[0033]   FIG. 4 shows an exemplary digital data transmitter.

[0034]   FIG. 5 shows an exemplary digital data receiver.

[0035]   FIG. 6 shows a signal-to-noise-plus-interference-ratio ("SINR") plot depicting a plurality of SINRs obtained from a plurality of symbol blocks with the equalizer of FIG. 5.

[0036]   FIG. 7 shows a mean-squared-error ("MSE") plot depicting a plurality of MSEs obtained from a plurality of symbol blocks with the LMS-based linear equalizer of FIG. 5.

[0037]   FIG. 8 shows a typical linear equalizer system having a decision device.

[0038]   FIG. 9A shows a digital communications device according to an embodiment of the invention.

[0039]   FIG. 9B shows a method according to an embodiment of the invention.

[0040]   FIG. 9C shows a linear equalizer according to an embodiment of the invention.

[0041]   FIG. 10 shows an MSE estimate plot based on coded symbols.

[0042]   FIG. 11 shows an estimate plot based on uncoded symbols.

[0043]   FIG. 12 shows an estimate plot showing a portion of the plot of FIG. 10.

[0044]   FIG. 13 shows an estimate plot showing a portion of the plot of FIG. 11.

[0045] FIG. 14 shows an estimate plot based on selective use of coded and uncoded symbols.

[0046] FIG. 15 shows an estimate plot showing a portion of the plot of FIG. 14.

[0047] FIG. 16 shows a linear equalizer system according to an embodiment of the invention.

[0048] FIG. 17 shows a digital communications device according to an embodiment of the invention.

[0049] FIG. 18 shows a method according to an embodiment of the invention.

[0050] FIG. 19 shows a linear equalizer according to an embodiment of the invention.

DETAILED DESCRIPTION

[0051] Before any embodiments of the invention are explained in detail, it is to be understood that the invention is not limited in its application to the details of construction and the arrangement of components set forth in the following description or illustrated in the following drawings. The invention is capable of other embodiments and of being practiced or of being carried out in various ways. Also, it is to be understood that the phraseology and terminology used herein are for the purpose of description and should not be regarded as limiting. The use of "including," "comprising," or "having" and variations thereof herein is meant to encompass the items listed thereafter and equivalents thereof as well as additional items.

[0052] As should also be apparent to one of ordinary skill in the art, the systems shown in the figures are models of what actual systems might be like. Many of the modules and logical structures described are capable of being implemented in software executed by a microprocessor or a similar device or of being implemented in hardware using a variety of components including, for example, application specific integrated circuits ("ASICs"). Terms like "equalizer" or "decoder" may include or refer to both hardware and/or software. Furthermore, throughout the specification capitalized terms are used. Such terms are used to conform to common practices and to help correlate the description with the coding examples, equations, and/or drawings. However, no specific meaning is implied or should be inferred simply due to the use of capitalization. Thus, the claims should not be limited to the specific

examples or terminology or to any specific hardware or software implementation or combination of software or hardware.

[0053]   As noted above, the step size μ of EQN. (17) controls a rate at which the LMS adaptive equalizer tap weights $\mathbf{w}$ converge to near an optimum $\mathbf{w}_{opt}$. It is desirable to use a larger step size to decrease the amount of time needed until convergence is obtained. However, a larger step size leads to a larger steady state MSE, or lower SINR, at the output of an equalizer after convergence. Hence, after the equalizer is close to convergence, a smaller step size is desirable. Therefore, it is generally advantageous to have a variable step size whose value depends on a "closeness" of $\mathbf{w}$ to $\mathbf{w}_{opt}$, thereby enabling a receiver to use a larger step size while the adaptive filter is converging and a smaller step size after convergence.

[0054]   The ability of an adaptive equalizer to track a nonstationary channel is also a concern in appropriately choosing a step size μ. If the equalizer converges close to the optimal tap weight vector $\mathbf{w}_{opt}$ and is running with a small step size, but then the channel conditions change, the equalizer must adequately track and adapt the tap weight vector $\mathbf{w}$. Detection of changes in channel conditions and a switch to a larger step size μ, even though the equalizer has previously converged, is advantageous in this situation.

[0055]   Embodiments of the invention include methods, systems, and devices for adaptively selecting a step size of an equalizer. In one specific embodiment, an adaptive equalizer selectively uses coded symbols or uncoded symbols to determine a step size by which to update tap weights used by a transversal filter. Selective use of coded symbols or uncoded symbols can enable a more accurate estimation of error throughout various states of the equalizer, which estimation in turn can be employed to determine an appropriate step size. For instance, uncoded symbols may be employed before a predetermined convergence state of the equalizer, and coded symbols may be employed once the predetermined convergence state is reached.

[0056]   Embodiments herein can achieve improved performance than that achieved in existing digital communication receivers. For instance, embodiments herein can be employed to respond adaptively to changing channel conditions and more effectively select an appropriate step size. In one embodiment, iterative processes are employed to detect when current tap weights are no longer sufficiently close to optimal tap weights, and to modify the

step size based on coded symbols or uncoded symbols as appropriate, so as to move closer to the optimal tap weight solution.

[0057]   Although some embodiments herein focus on processing (e.g., reception) of digital television signals, the invention may be implemented in connection with other kinds of digital signals. Similarly, although some embodiments herein relate to the 8VSB RF modulation format, the invention may be implemented in connection with other modulation formats, such as formats that include coded information and a priori known information.

[0058]   Additionally, although some embodiments herein relate to linear equalizers ("LEs"), the invention may be implemented in connection with other equalizer architectures, such as, for example, decision feedback equalizers ("DFEs") and predictive decision feedback equalizers ("pDFEs").

[0059]   FIG. 9A shows a digital communications device 950 according to an embodiment of the invention. The device 950 can be implemented as, or in conjunction with, any of a host of devices, such as, for example, a receiver (e.g., digital communication receiver), tuner, PC adapter card, set top box, DVD recorder, HDTV recorder, television, phone, or handheld device. The device 950 can be implemented partially or entirely on a semiconductor (e.g., FPGA semiconductor) chip, such as a chip developed through a register transfer level (RTL) design process. The device 950 includes a receiver module 955 and optional hardware and/or software module(s) 990 that provide additional functions (e.g., display functions). In other embodiments, the device 950 includes more or fewer modules than those depicted. For example, certain of the depicted modules can be implemented on other devices that interface with the device 950 (e.g., the receiver module 955 can communicate with a display module incorporated in a separate device).

[0060]   The receiver module 955 includes a demodulator 960, a decoder 965, and an equalizer 970. In some embodiments, the receiver module 955 includes one or more additional modules, such as, for example, a tuner, a sync and timing recovery module, a matched filter, a phase tracker, a deinterleaver, a second decoder, a slicer, and/or a derandomizer. The equalizer 970 includes a selection module 975, an error estimator 980, and a step size generator 985. Exemplary implementations of the equalizer 970 are described in further detail below.

[0061]   FIG. 9B shows a method 991 according to an embodiment of the invention. The method 991 can be employed by, or in conjunction with, an equalizer for a digital data receiver in order to determine a step size of the equalizer. For instance, the method 991 can be employed by the receiver module 955 and/or equalizer 970 of FIG. 9A, as well as in connection with other embodiments described below. In task 993, data is received that includes coded symbols and uncoded (e.g., a priori known) symbols. In task 994, a first error estimate is determined based on decoded symbols corresponding to the coded symbols. In task 995, a second error estimate is determined based on the uncoded symbols. In task 996, the first error estimate or the second error estimate is adaptively selected based on a convergence criterion. In task 997, a step size is determined based on the selected error estimate. The method 991 can be iteratively performed as a datastream is received and processed.

[0062]   Variations of the method 991 are within the scope of embodiments of the invention. For instance, in one embodiment, a method selects either decoded symbols or uncoded symbols based on a convergence criterion; determines a signal estimate based on the selected symbols; determines an error estimate based on received data and the signal estimate; and determines a step size based on the error estimate.

[0063]   FIG. 9C shows a linear equalizer ("LE") 900 according to an embodiment of the invention that can be implemented in a digital receiver. The equalizer 900 includes a linear transversal filter 904 having tap weights $\mathbf{w}$ and an output $y[n]$, a selection module 908, an error estimator 912, a step size generator 916 that generates a step size parameter $\mu[k]$, and an LMS tap weight module 920 that adjusts the tap weights $\mathbf{w}$ based on the step size parameter $\mu[k]$. In some embodiments, the error estimator 912 includes an MSE estimator. In such cases, the step size generator 916 generates the step size parameter $\mu[k]$ based on error information provided by the MSE estimator. Although FIG. 9C shows an LE, other types of equalizers, such as DFEs and pDFEs, can be employed.

[0064]   As previously noted, 8VSB signals include a combination of 8-level trellis coded symbols and uncoded 2-level symbols. The selection module 908 generates an output $d[n]$, which in turn is subtracted from $y[n]$ to obtain $e[n]$ at a summing node 924. The selection module 908 also feeds the output $d[n]$ to the error estimator 912. The selection module 908 includes a trellis decoder 928 that decodes the coded symbols at the equalizer output $y[n]$ using the Viterbi algorithm. In the embodiment shown, the decoder 928 has a zero delay or a

traceback depth of one output. While longer traceback depth decisions are generally more reliable, they incur a longer delay, which can be unacceptable if an instantaneous $e[n]$ is needed for the LMS update.

[0065]   Most of the uncoded symbols including all segment sync symbols and the first 728 symbols of the field sync segment are a priori known. These are perfectly "decoded" by reading them out of a memory at appropriate times. The 92 unknown 2-level symbols at the end of the field sync segment are decoded by slicing at a midpoint with a slicer 932 since the unknown symbols are 2-level symbols.

[0066]   Through a control line 940, a synchronizing control signal indicates to the selection module 908 and a switch 936 which type of symbol is being decoded. Exemplary methods for deriving this control signal first require symbol clock recovery, then data field synchronization, both of which occur in the preceding sync and timing recovery block, and then a modulo 260,416 symbol clock rate counter feeding a comparator that activates the control line 940 according to the type of symbol.

[0067]   The error estimator 912 controls the adjustable step size parameter $\mu[k]$ that is being fed to the LMS module 920. In one embodiment, the error estimator 912 includes a high MSE indicator and a low MSE indicator. A high MSE (low SINR) indicates that $\mathbf{w}$ is not close to $\mathbf{w}_{opt}$, and thus a need for a larger step size. A low MSE (high SINR) indicates that $\mathbf{w}$ is close to $\mathbf{w}_{opt}$, and thus a desirability of a smaller step size. An exemplary method of MSE estimation (or, equivalently, SINR estimation) for the 8VSB signal is discussed below.

[0068]   In some embodiments, the error estimator 912 periodically estimates an error, such as an MSE, every block of M symbols times from the trellis decoded symbols outputted by the decoder 928 as follows. For example, in the case of an MSE estimate, an instantaneous MSE estimate at block time k is given by EQN. (19).

$$\xi_{dec}^2[k] = \left(\frac{1}{M}\right)\sum_{i=0}^{M-1}\left(y[m+i] - d[m+i]\right)^2$$

(19)

where k is a block index, symbol index base $m = (k\text{-}1)M$, y is the equalizer output, d is the zero delay output of the trellis decoder 928, and M is a selected block size. Similarly, in the case of an MSE estimate, an averaged MSE estimate is given by EQN. (20).

15

$$\xi^2_{dec,\beta}[k] = (1 - \beta_{dec})\xi^2_{dec}[k] + \beta_{dec}\xi^2_{dec,\beta}[k-1], \ 0 < \beta_{dec} < 1$$

(20)

Note that values of β_{dec} are typically close to 1, with a smaller value providing a noisier MSE estimate but faster tracking of a changing MSE.

[0069] FIG. 10 shows an estimate plot 1000. Times (in terms of symbol blocks at the output of the equalizer 900) are measured along an x-axis 1004, and values of the MSE are measured along a y-axis 1008. A curve 1012 shows actual MSEs obtained, and a curve 1018 shows values of the MSE estimate $\xi^2_{dec,\beta}$. FIG. 10 shows that when the actual MSEs (curve 1012) are above about 0.9 (equivalent to an SINR below 13.75 dB), the estimated MSE $\xi^2_{dec,\beta}$ is considerably below the actual MSE.

[0070] Alternatively, error values may be estimated using only the a priori known 2-level segment sync symbols. For example, in the case of an MSE estimate, whenever $d[p].....d[p+3]$ are segment sync symbols, then an instantaneous MSE estimate at segment j is given by EQN. (21).

$$\xi^2_{seg}[j] = \left(\frac{1}{4}\right)\sum_{i=0}^{3}\left(y[p+i]-d[p+i]\right)^2$$

(21)

where j is a segment index, and $p = 832(j-1)$ is a symbol index (note that blocks k and segments j are in general asynchronous). Similarly, in the case of an MSE estimate, whenever $d[p].....d[p+3]$ are segment sync symbols, then an averaged MSE estimate is given by EQN. (22).

$$\xi^2_{seg,\beta}[j] = (1 - \beta_{seg})\xi^2_{seg}[j] + \beta_{seg}\xi^2_{seg,\beta}[j-1], \ 0 < \beta_{seg} < 1$$

(22)

Note that values of β_{seg} are typically close to 1, with a smaller value providing a noisier MSE estimate but faster tracking of a changing MSE.

[0071] FIG. 11 shows an estimate plot 1100. Times (in terms of symbol blocks at the output of the equalizer 900) are measured along an x-axis 1104, and values of the MSE are

measured along a y-axis 1108. A curve 1112 shows actual MSEs obtained, and a curve 1118 shows values of the MSE estimate $\xi_{seg,\beta}^2$. FIG. 11 shows that values of $\xi_{seg,\beta}^2$ are more accurate than $\xi_{dec,\beta}^2$ at high MSE (low SINR).

[0072]   FIG. 12 shows an estimate plot 1200 showing a portion of the plot 1000 of FIG. 10 between symbol blocks 2600 and 3800. Times (in terms of symbol blocks at the output of the equalizer 900) are measured along an x-axis 1204, and values of the MSE are measured along a y-axis 1208. A curve 1212 shows actual MSEs obtained, and a curve 1218 shows values of the MSE estimate $\xi_{dec,\beta}^2$. Similarly, FIG. 13 shows an estimate plot 1300 showing a portion of the plot 1100 of FIG. 11 between symbol blocks 2600 and 3800. Times (in terms of symbol blocks at the output of the equalizer 900) are measured along an x-axis 1304, and values of the MSE are measured along a y-axis 1308. A curve 1312 shows actual MSEs obtained, and a curve 1318 shows values of the MSE estimate $\xi_{dec,\beta}^2$.

[0073]   As shown in FIG. 12 and FIG. 13, while the $\xi_{seg,\beta}^2$ estimate is better than $\xi_{dec,\beta}^2$ at high MSE, the $\xi_{seg,\beta}^2$ estimate is noisier than $\xi_{dec,\beta}^2$ as MSE decreases because ξ_{seg}^2 is averaged over fewer symbols than is ξ_{dec}^2 ($4 \ll M$). Thus, ξ_{seg}^2 represents a noisier estimate of the actual MSE. As such, each of the two methods provides relatively better estimates for respective MSE regions. Hence, according to various embodiments of the invention, these methods are selectively combined to provide a more accurate MSE estimate both at low and high MSE.

[0074]   For example, in one embodiment involving EQN. (23) and EQN. (24), if $\xi_{dec,\beta}^2$ is less than a predetermined MSE value (e.g., associated with a convergence state), the step size generator 916 selects $\xi_{dec,\beta}^2$ as the error estimate *EstMSE*. If $\xi_{dec,\beta}^2$ is greater than or equal to the predetermined MSE value, the step size generator 916 selects $\xi_{seg,\beta}^2$ as the error estimate *EstMSE*. In some embodiments, the predetermined MSE value is about 0.9. That is,

If $\xi_{dec,\beta}^2[k] < 0.9$, (equivalent to SINR > 13.75 dB)

$$EstMSE[k] = \xi_{dec,\beta}^2[k] \quad \text{(near or post convergence)}$$

$$(23)$$

Else

$$EstMSE[k] = \xi_{seg,\beta}^2[j] \quad \text{(pre convergence)}$$

$$(24)$$

[0075]   FIG. 14 shows an error estimate plot 1400 based on selectively combining use of coded and uncoded symbols. Times (in terms of symbol blocks at the output of the equalizer 900) are measured along an x-axis 1404, and values of the MSE in dB are measured along a y-axis 1408. A curve 1412 shows actual MSEs obtained, and a curve 1416 shows values of the MSE estimate *EstMSE* determined by EQNS. (21) – (24) and associated logic. FIG. 15 similarly shows a portion 1400' of the estimate plot 1400 from symbol blocks 2600 to 3800. The portion 1400' shows the error estimator 912 transitioning from use of uncoded symbols (i.e., $\xi_{seg,\beta}^2$) to use of coded symbols (i.e., $\xi_{dec,\beta}^2$) at a switch point 1420. In the embodiment shown, the switch point is set at about MSE=0.9.

[0076]   Once the error estimator 912 has determined an error estimate, the step size generator 916 uses the error estimate *EstMSE*[k] to select a variable LMS step size depending on a range within which the error estimate *EstMSE*[k] falls. For example, as shown in expression (25), if the error estimate *EstMSE*[k] falls within a range $\Re_r$, the step size generator 916 sets the step size parameter $\mu[k]$ equal to a predetermined step size μ_r.

$$\text{If } EstMSE[k] \in \Re_r$$

$$\mu[k] = \mu_r$$

$$(25)$$

end

In one embodiment, smaller values of *EstMSE* correspond to lower values of μ_r. For example, the step size generator 916 can use three *EstMSE*[k] ranges: a first range $\Re_1$, a second range $\Re_2$, and a third range $\Re_3$. If *EstMSE*[k] is within the first range $\Re_1$, which is less than or equal to 0.7 dB (equivalent to an estimated SINR of about 14.75 dB), the step

size generator 916 uses a step size of μ_1. Similarly, if $EstMSE[k]$ is within the second range $\mathfrak{R}_2$, which is between 0.7 dB and 0.9 dB (equivalent to an estimated SINR of about 13.75 dB), the step size generator 916 uses a step size of μ_2. If $EstMSE[k]$ is within the third range $\mathfrak{R}_3$, which is above 0.9 dB, the step size generator 916 uses a step size of μ_3. In such cases, μ_1 is the smallest among the three step sizes, and μ_3 is the largest among the three step sizes.

[0077]   In other embodiments, the step size generator 916 determines the step size differently. For instance, the step size generator 916 may employ a function (e.g., a continuous function) that computes the step size based on information received from the error estimator 912. In some embodiments, the step size is given by EQN. (25').

$$\mu[k] = \gamma EstMSE[k]$$

(25')

where γ is a predetermined positive real constant.

[0078]   FIG. 16 shows a second LE system 1600 according to an embodiment of the invention. The LE system 1600 is an alternative embodiment of the LE system 900 of FIG. 9C. Similar to the LE system 900, the LE system 1600 includes a linear transversal filter 1604 having tap weights **w** and an output $y[n]$, a selection module 1608, an error estimator 1612, a step size generator 1616 that generates a step size parameter $\mu[k]$, and an LMS tap weight module 1620 that adjusts the tap weights **w** based on the step size parameter $\mu[k]$. Furthermore, the selection module 1608 includes a decoder 1621 that decodes the coded symbols at the equalizer output $y[n]$ using the Viterbi algorithm, and a slicer 1622 that slices the unknown 2-level symbols at a midpoint.

[0079]   The LE system 1600 also includes a plurality of delay blocks 1624, 1628, 1632 that introduce delays during signal processing. In some embodiments, the error estimator 1612 includes an MSE estimator. In such cases, the step size generator 1616 generates the step size parameter $\mu[k]$ based on errors estimated by the MSE estimator. However, unlike the LE system 900 of FIG. 9C, the LE system 1600 uses a trellis decoder 1621 with a full traceback depth D+1 (which thus has a decode delay D) and delay blocks 1624, 1628, 1632, all with delay D. A full traceback depth of the trellis decoder 1621 can generate an output that has a higher reliability than a zero delay output generated by the trellis decoder 928 of FIG. 9C. As such, the trellis decoder 1621 with adjustable traceback depth controlled by the

delay blocks 1624, 1628, 1632 can provide a more accurate $\xi_{dec,\beta}^2$ value if both the a priori

known 2-level symbol output memory and the $y[n]$ signal are likewise delayed by D.

Accordingly, the input to the error estimator 1612 is $d_D[n]$, which is $d_D[n]$ delayed by an

amount D.

[0080]   In the LE system 1600, for the coded symbols, an instantaneous MSE estimate at

block time k given by EQN. (26).

$$\xi_{dec}^2[k] = \left(\frac{1}{M}\right)\sum_{i=0}^{M-1}\left(y_D[m+i] - d_D[m+i]\right)^2$$

(26)

where k is a block index, the symbol index base $m = (k-1)M$, y_D is a delayed equalizer output,

d is a full traceback output of the trellis decoder 1621 with a delay D, and M is a selected

block size. Similarly, in the LE system 1600, for the uncoded symbols, an instantaneous

MSE estimate at segment j is given by EQN. (27).

$$\xi_{seg}^2[j] = \left(\frac{1}{4}\right)\sum_{i=0}^{3}\left(y_D[p+i] - d_D[p+i]\right)^2$$

(27)

where j is a segment index, and $p = 832(j - 1)$ is a symbol index. Blocks k and segments j are

in general asynchronous.

[0081]   Various convergence criteria may be used in connection with embodiments of the

invention. In some embodiments, an initial error (e.g., MSE) estimate based on coded

symbols is used to determine if coded symbols should continue to be used for further error

estimates, or if uncoded symbols should be used for further error estimates. Other exemplary

embodiments use convergence criteria based on how often the sign of an error gradient

changes.

[0082]   Consider a case where the equalizer 970 of FIG. 9A has already converged and its

output MSE is low, such that a small LMS step size (μ) has been selected according to the

previously described methods. When the equalizer 970 has converged, its tap weights **w** can

be said to be adjusted properly for the current multipath channel conditions, and the tap

weights **w** can be seen to change very little as time progresses, as long as the channel

conditions are static. If the channel conditions begin to change with time, the channel is then considered to be dynamic.

[0083]   If the post convergence channel conditions change rapidly enough, the MSE at the equalizer output will increase because the above-described LMS algorithm, operating with a small step size (μ), cannot track the changing channel conditions fast enough. This may result in a selection of a larger LMS step size (μ) after the MSE has increased beyond a specific point, which may then aid the LMS algorithm in its attempt to track the changing channel conditions. However, this change in channel conditions may be so rapid that a switch to a larger step size is too late, resulting in an ultimate divergence of the equalizer 970 as it is unable to track a dynamic channel.

[0084]   In some embodiments of the invention, a larger step size is selected earlier in the channel tracking process, while the MSE is still relatively small, using a metric other than the MSE estimate *EstMSE*. Under dynamic channel conditions, even though the MSE remains low and thus the step size (μ) small, the equalizer tap weights **w** vary much more than under static channel conditions. Therefore, a measure of the difference of the tap weight vector values **w** at two points in time, if in excess of one or more predetermined thresholds, can be used to trigger an increase in LMS step size (μ) earlier in the process, while the MSE estimate *EstMSE* is still small.

[0085]   FIG. 17 shows a digital communications device 1700 according to an embodiment of the invention. The device 1700 can be implemented as, or in conjunction with, any of a host of devices, such as, for example, a receiver (e.g., digital communication receiver), tuner, PC adapter card, set top box, DVD recorder, HDTV recorder, television, phone, or handheld device. The device 1700 can be implemented partially or entirely on a semiconductor (e.g., FPGA semiconductor) chip, such as a chip developed through a register transfer level (RTL) design process. The device 1700 includes a receiver module 1704 and optional hardware and/or software module(s) 1706 that provide additional functions (e.g., display functions). In other embodiments, the device 1700 includes more or fewer modules than those depicted. For example, certain of the depicted modules can be implemented on other devices that interface with the device 1700 (e.g., the receiver module 1704 can communicate with a display module incorporated in a separate device).

[0086]   The receiver module 1704 includes a demodulator 1708, a decoder 1712, and an equalizer 1716. In some embodiments, the receiver module 1704 includes one or more additional modules, such as, for example, a tuner, a sync and timing recovery module, a matched filter, a phase tracker, a deinterleaver, a second decoder, a slicer, a derandomizer, a Fourier transformer, and/or an inverse Fourier transformer. The equalizer 1716 includes a tap weight module 1720, a difference module 1724, an error estimator 1728, a comparison module 1732, and a step size factor module 1736. In some embodiments, the tap weight module 1720 stores, in a memory, tap weights corresponding to multiple instants in time. For example, the tap weight module 1720 stores a first set of tap weights at time indices $k+L$, and a second set of tap weights at time index k. In some embodiments, the tap weight module 1720 constantly updates the stored tap weights.

[0087]   FIG. 18 shows a channel tracking method 1800 according to an embodiment of the invention. The method 1800 can be employed by, or in conjunction with, an equalizer for a digital data receiver. For instance, the method 1800 can be employed by the receiver module 1704 and/or equalizer 1716 of FIG. 17, as well as in connection with other embodiments described herein. In task 1804, a first set of equalizer tap weights ($\mathbf{w}[k]$) at a first point in time, and a second set of equalizer tap weights ($\mathbf{w}[k+L]$) at a second point in time, are determined (e.g., retrieved from the tap weight module 1720 or an associated memory). In task 1808, a difference between the first and second sets of tap weights is determined. In some embodiments, the difference is a metric ($\rho[k]$) determined by calculating a 2-norm of the difference of the tap weight vectors $\mathbf{w}$ at the respective points in time, given by EQN. (28) as follows.

$$\rho[k] = \left\| \mathbf{w}[k] - \mathbf{w}[k-L] \right\|_2^2$$

(28)

The metric determined by EQN. (28) is a time domain metric. In some embodiments, values of L are determined (e.g., selected) in advance. In general, the values of L are small enough so that the amount of tap weight variation (as represented by $\rho[k]$) can be determined quickly and updated frequently, but also large enough so that the amount of tap weight variation is not too small to be useful. For example, with k representing a block of symbols index, and with such block being 512 symbols in length, a value of $L = 50$ or $L = 100$ may be employed.

[0088] At task 1812, an error estimate *EstMSE*[*k*] is determined based on the decoded or the uncoded symbols (e.g., as described above). In some embodiments, an initial LMS step size is selected based on the error estimate, as shown in expression (25) described above. At task 1816, the error estimate determined at task 1812 is compared to a divergence threshold, γ. In some embodiments, the divergence threshold is generally a predetermined constant below which the equalizer is able to track a dynamic channel (e.g., a predetermined constant below which the equalizer is said to have converged to a relatively low MSE). At task 1820, a step size factor is determined based on the difference of task 1808 and the comparison of task 1816. In time domain processing, the step size factor can be, for example, a step size increment ($\Delta\mu_r$). If the error estimate *EstMSE*[*k*] is less than the divergence threshold, γ, the LMS step size (μ) is adjusted with the step size increment determined by expression (29) as follows.

$$\text{If } EstMSE[k] < \gamma \text{ and if } \rho[k] \in \Im_r,$$

$$\mu[k] = \mu[k] + \Delta\mu_r$$

$$(29)$$

where $\Im_r$ represents a plurality of difference ranges. In general, larger values of $\rho[k]$ correspond to larger values of the step size increment, $\Delta\mu_r$. For example, five regions may be established for $\rho[k]$, as shown in exemplary expression (30) as follows.

$$\begin{aligned}
\Im_1 &= (0, 20] & \Delta\mu_1 &= 0 \\
\Im_2 &= (20, 40] & \Delta\mu_2 &= 5.6 \times 10^{-3} \\
\Im_3 &= (40, 60] & \Delta\mu_3 &= 1.1 \times 10^{-2} \\
\Im_4 &= (60, 80] & \Delta\mu_4 &= 1.7 \times 10^{-2} \\
\Im_5 &= (80, \infty) & \Delta\mu_5 &= 2.3 \times 10^{-2}
\end{aligned}$$

$$(30)$$

That is, a different step size increment, $\Delta\mu_r$, is selected based on which of the difference ranges includes the determined difference. For example, for a difference of 55, which falls in the third difference range $\Im_3$, a step size increment, $\Delta\mu_3$, of 1.1×10^{-2} is selected. In other embodiments, the difference ranges may vary considerably depending on the particular

23

equalizer implementation and the selected value for L. Alternatively, $\Delta\mu$ can be set to a value that is directly proportional to $\rho[k]$ on a more continuous basis.

[0089]   Variations of the method 1800 are within the scope of embodiments of the invention. For instance, in one embodiment, a method processes the received symbols (decoded and/or uncoded) in the frequency domain.

[0090]   More specifically, each of the linear transversal filters 904 (of FIG. 9) and 1604 (of FIG. 16) has a particular filter length. In some applications, the length of such a filter can reach hundreds or thousands of taps. In such cases, more efficient hardware implementations are possible by utilizing a filter that operates in a block manner in the frequency domain. With the Fast Fourier Transform ("FFT") and its inverse ("IFFT"), the large computational overhead required for time domain convolution of the input data with the equalizer filter taps can be greatly reduced by implementing a frequency domain point wise vector multiply along with an overlap and save methodology. Additionally, the LMS iteration process as described above can be performed only once per block of symbols, instead of once per symbol. The block size is generally defined as the length of the FFT (N) minus an overlap.

[0091]   FIG. 19 shows an unconstrained frequency domain equalizer 1900 according to an embodiment of the invention. The frequency domain equalizer 1900 includes a FFT module 1904 that transforms an equalizer input vector, $\mathbf{u}[k]$, into its frequency domain equivalent, $\mathbf{U}_F[k]$, as defined in EQN. (31).

$$\mathbf{U}_F[k] = FFT(\mathbf{u}[k])$$

(31)

where k is the block index, and the subscript F indicates a frequency domain vector. The frequency domain equalizer 1900 also includes an IFFT module 1908 that inverse transforms the equalizer frequency domain output vector, $\mathbf{Y}_F[k]$, into its time domain equivalent, $\mathbf{y}[k]$, as defined in EQN. (32).

$$\mathbf{y}[k] = IFFT(\mathbf{U}_F[k] \otimes \mathbf{W}_F[k])$$

(32)

where $\otimes$ represents point wise vector multiply, and $\mathbf{W}_F[k]$ is the frequency domain tap weight vector at time index k. The output vector, $\mathbf{y}[k]$, is then fed to a decision device 1912,

such as the decision devices 908, 1608 described above. The decision device 1912 then generates a desired symbol vector, $\mathbf{d}[k]$.

[0092]   In addition, $\widetilde{\mathbf{y}}[k] = \begin{bmatrix} \mathbf{0} \\ \mathbf{y}[k] \end{bmatrix}$ and $\widetilde{\mathbf{d}}[k] = \begin{bmatrix} \mathbf{0} \\ \mathbf{d}[k] \end{bmatrix}$ are the zero padded filter output and

zero padded desired symbol vectors, respectively, with such zero padding as needed to properly execute an overlap and save methodology. An error vector ($\mathbf{e}[k]$) is then obtained by subtracting the output vector from the desired symbol vector, $\mathbf{e}[k] = \widetilde{\mathbf{d}}[k] - \widetilde{\mathbf{y}}[k]$. Similarly, a frequency domain error vector can then be defined as the FFT of a vector difference between the zero-padded desired symbol vectors and the zero-padded filter output vectors at an FFT module 1916, as shown in EQN. (33).

$$\mathbf{E}_F[k] = FFT(\widetilde{\mathbf{d}}[k] - \widetilde{\mathbf{y}}[k]) = FFT(\mathbf{e}[k])$$

(33)

Similar to the error estimators 912, 1612 described above, an error estimator 1920 estimates an error from the error vector ($\mathbf{e}[k]$).

[0093]   After the frequency domain input vector, $\mathbf{U}_F[k]$, has been Hermitian conjugated at a conjugate module 1924 to obtain $\mathbf{U}_F^H[k]$, it is point-wise vector multiplied by the frequency domain error vector, $\mathbf{E}_F[k]$. A variable step size module 1928 then determines a variable step size scalar $\mu[k]$ from the error estimate $EstMSE[k]$ as described. The equalizer 1900 then iteratively determines a new set of tap weights based on the current set of tap weights, the scalar $\mu[k]$, and a delay 1932 in a frequency block LMS ("FBLMS") iteration process, which is executed once per block. A FBLMS iteration process under a normal multipath condition is given by EQN. (34).

$$\mathbf{W}_F[k+1] = \mathbf{W}_F[k] + \mu[k](\mathbf{U}_F^H[k] \otimes \mathbf{E}_F[k])$$

(34)

Under difficult multipath conditions, the convergence properties of the equalizer 1900 can be improved if the variable step size module 1928 utilizes step size power normalization. An exemplary step size power normalization process is represented by EQN. (35) as follows.

$$\mathbf{W}_F[k+1] = \mathbf{W}_F[k] + \mu[k]\mathbf{D}^{-1}(\mathbf{U}_F^H[k] \otimes \mathbf{E}_F[k])$$

(35)

where **D** is a step size normalization matrix, as shown in EQN. (36).

$$\mathbf{D} = diag[\hat{\lambda}_0 .... \hat{\lambda}_{N-1}]$$

(36)

where $\hat{\lambda}_i$ provides a normalization factor for each equalizer tap i, where $i = 0....(N-1)$. $\hat{\lambda}_i$ is defined in EQN. (37) as follows.

$$\hat{\lambda}_i[k] = \beta_{step}\hat{\lambda}_i[k-1] + (1 - \beta_{step})|U_{F,i}[k]|^2$$

(37)

where β_{step} is a predetermined forgetting factor.

[0094]   The normalization factors $\hat{\lambda}_i$'s are a function of the frequency domain samples of the received data stream, $u_{F,i}$; hence, the $\hat{\lambda}_i$'s are a function of the channel conditions. The forgetting factor, β_{step}, controls the updating and averaging of the calculated normalizing factors. A larger β_{step} gives more weight to past values of $\hat{\lambda}_i$, enhancing the effect of averaging over time and providing a more accurate estimate of the ideal power normalization factor. A smaller β_{step} gives less weight to past values of $\hat{\lambda}_i$, resulting in a noisier estimate of the power normalization factor.

[0095]   In some embodiments, larger values of β_{step} are employed for static channels. However, in other embodiments, for dynamic channels, an optimal value for $\hat{\lambda}_i$ changes over time as the channel condition changes. Hence, in some embodiments, a smaller value of β_{step} is employed to more quickly "forget" past values.

[0096]   In the frequency domain, the channel tracking method 1800 of FIG. 18 can be employed to determine a step size factor that is, for example, the forgetting factor β_{step}. The difference of task 1808 can be determined, for example, by calculating a 2-norm of the

difference of the frequency domain or transformed tap weight vectors at different points in time k and $k - L$, as shown in EQN. (38) as follows.

$$\rho[k] = \left\| \mathbf{W}_F[k] - \mathbf{W}_F[k-L] \right\|_2^2$$

(38)

[0097]   In an embodiment, either a large or small β_{step} is selected depending on an estimate of changing channel conditions as shown in expressions (39) and (40).

If $EstMSE[k] < \gamma$

if $\rho[k] > \alpha$,

$$\beta_{step}[k] = \xi_{small}$$

(39)

else

$$\beta_{step}[k] = \xi_{large}$$

(40)

where α is a difference threshold, ξ_{small} and ξ_{large} are predetermined constants, and ξ_{small} $< \xi_{large}$. That is, if the error estimate is less than the divergence threshold, and if the difference is greater than the difference threshold, the forgetting factor is assigned a relatively small value. Conversely, if the error estimate is less than the divergence threshold, and if the difference is less than the difference threshold, the forgetting factor is assigned a relatively large value. Alternatively, $\beta_{step}[k]$ can be set to be inversely proportional to $\rho[k]$ in a continuous manner as shown in expression (41).

if $EstMSE[k] < \gamma$,

$$\beta_{step}[k] = \frac{\kappa}{\rho[k]}$$

(41)

where κ is a predetermined constant. That is, if the error estimate is greater than a divergence threshold, the forgetting factor increases as the difference decreases.

[0098]   Once the variable step size module 1928 determines the forgetting factor, $\beta_{step}[k]$, the variable step size module 1928 proceeds to determine the normalization factor $\hat{\lambda}_i$ for each equalizer tap, and to generate the step size normalization matrix, **D**, as described. The equalizer 1900 then iteratively generates a new set of frequency domain tap weights, $\mathbf{W}_F[k+1]$, based on EQN. (35). A delay is then applied to the new set of frequency domain tap weights at the delay module 1932. The delayed set of frequency domain tap weights are then used in filtering an input vector. The equalizer 1900 then iteratively applies the FBLMS process as described.

[0099]   It should be noted that the numerical values described above and illustrated in the drawings are exemplary values only. Other numerical values can also be used. Additionally, various embodiments above may be selectively and separately implemented in suitable contexts, or together (e.g., in the same equalizer).

[00100]   Various features and advantages of the invention are set forth in the following claims.

WHAT IS CLAIMED IS:

1.   A method of channel tracking in an adaptive equalizer for a digital data receiver, the method comprising:

determining a first set of tap weights and a second set of tap weights of the equalizer;

determining a difference between the first and second sets of tap weights;

determining an error estimate based on symbols received at the receiver;

comparing the error estimate with a divergence threshold; and

determining a step size factor based on the difference and the comparison.

2.   The method of claim 1, wherein the first and second sets of tap weights comprise respective frequency domain tap weights, and the step size factor comprises a forgetting factor.

3.   The method of claim 2, wherein determining a step size factor comprises:

assigning a first predetermined forgetting factor when the difference is greater than a difference threshold; and

assigning a second predetermined forgetting factor when the difference is less than or equal to the difference threshold,

wherein the first predetermined forgetting factor is less than the second predetermined forgetting factor.

4.   The method of claim 2, wherein the forgetting factor is inverse-proportionally related to the difference.

5.   The method of claim 2, wherein the difference comprises a 2-norm difference, the method further comprising:

determining a normalization factor based on the forgetting factor and the first and second sets of tap weights; and

iteratively determining a third set of frequency domain tap weights based on the normalization factor.

6.   The method of claim 1, wherein the step size factor comprises a step size increment, and wherein determining a step size factor comprises:

comparing the difference with a plurality of difference ranges including a first difference range, each of the difference ranges having a corresponding step size increment; and

selecting the corresponding step size increment of the first difference range if the difference is within the first difference range.

7.   The method of claim 6, further comprising incrementing a base step size based on the step size increment to determine a step size.

8.   The method of claim 7, further comprising:

comparing the error estimate with a plurality of error ranges including a first error range, each of the error ranges having a corresponding base step size; and

selecting a corresponding base step size of the first error range if the error estimate is within the first error range.

9.   The method of claim 1, wherein the difference comprises a 2-norm of the difference between the first and second set of tap weights.

10.   The method of claim 1, wherein the error estimate comprises a mean-squared-error estimate.

11.   The method of claim 1, wherein the first set of tap weights comprise tap weights of the equalizer at a first time, and the second set of tap weights comprise tap weights of the equalizer at a second time, wherein the first time is different from the second time.

12.   The method of claim 11, further comprising determining a third set of tap weights based on at least one of the first and second sets of tap weights, the difference, and the comparison, wherein the third set of tap weights comprise tap weights of the equalizer at a third time, wherein the third time is different from the first and second times.

13.   The method of claim 1, further comprising updating the step size based on the step size factor.

14.   The method of claim 1, wherein the equalizer is a linear-mean-squared (LMS) equalizer for 8-level-vestigial-sideband (8VSB) modulated signals.

15.   The method of claim 1, wherein the digital data receiver is a digital television receiver.

16.   An adaptive equalizer for a digital data receiver, data received by the receiver

including coded symbols, the equalizer comprising:

a tap weight module configured to determine a first set of tap weights and a second set

of tap weights of the equalizer;

a difference module configured to determine a difference between the first and second

sets of tap weights;

an error estimator configured to determine an error estimate based on symbols

received at the receiver;

a comparison module configured to compare the error estimate with a divergence

threshold; and

a step size factor module configured to determine a step size factor based on the

difference and the comparison.

17.   The equalizer of claim 16, wherein the equalizer is implemented on a semiconductor

chip.

18.   The equalizer of claim 16, wherein the equalizer is implemented in a digital television

receiver.

19.   The equalizer of claim 16, wherein the step size factor comprises a step size

increment.

20.   The equalizer of claim 16, wherein the equalizer is a frequency domain block linear-

mean-squared (FBLMS) equalizer for 8-level-vestigial-sideband (8VSB) modulated signals,

and wherein the step size factor comprises a forgetting factor.

21.   A device configured to process digital television signals, the device comprising:

a receiver including a demodulator, a decoder, and an equalizer,

the receiver configured to receive radio frequency signals modulated with data including coded symbols and uncoded symbols,

the demodulator configured to demodulate the received radio frequency signals to produce the coded symbols and the uncoded symbols,

the decoder configured to decode the coded symbols to produce corresponding decoded symbols, and

the equalizer including

a tap weight module configured to determine a first set of tap weights and a second set of tap weights of the equalizer,

a difference module configured to determine a difference between the first and second sets of tap weights,

an error estimator configured to determine an error estimate based on symbols received at the receiver,

a comparison module configured to compare the error estimate with a divergence threshold, and

a step size factor module configured to determine a step size factor based on the difference and the comparison.

22.   The device of claim 21, wherein the device is a tuner, a television, a PC adapter card, a set top box, a DVD recorder, a HDTV recorder, a phone, or a handheld device.

23.   A method of determining a step size for a linear-mean-squared (LMS) equalizer of 8-level-vestigial-sideband (8VSB) modulated signals having trellis-coded symbols and segment sync symbols, the method comprising:

(a)   determining a mean-squared-error estimate based on at least one of decoded trellis-coded symbols and the segment sync symbols;

(b)   determining a first set of tap weights and a second set of tap weights of the equalizer;

(c)   determining a 2-norm of the difference between the first and second set of tap weights;

(d)   comparing the error estimate with a divergence threshold;

(e)   if the error estimate is less than the divergence threshold, comparing the 2-norm with a plurality of difference ranges including a first difference range, each of the difference ranges having a corresponding step size increment, selecting the corresponding step size increment of the first difference range if the 2-norm is within the first difference range, and incrementing a base step size based on the step size increment to determine a step size, and determining a third set of tap weights based on the step size; and

(f)   iteratively performing acts (a)–(e).

24.   A digital communication receiver configured to receive radio frequency signals modulated with data including coded symbols and a priori known uncoded symbols, the receiver comprising:

        a demodulator configured to demodulate the received radio frequency signals to produce the coded symbols and the a priori known uncoded symbols;

        a decoder configured to decode the coded symbols to produce corresponding decoded symbols; and

        a frequency domain equalizer including

a difference module configured to determine a difference between a first tap weight vector and a second tap weight vector,

an error estimator configured to determine an error estimate based on the decoded symbols or the uncoded symbols,

a comparison module configured to compare the error estimate with a divergence threshold, and

a forgetting factor module configured to determine a forgetting factor based on the difference and the comparison.
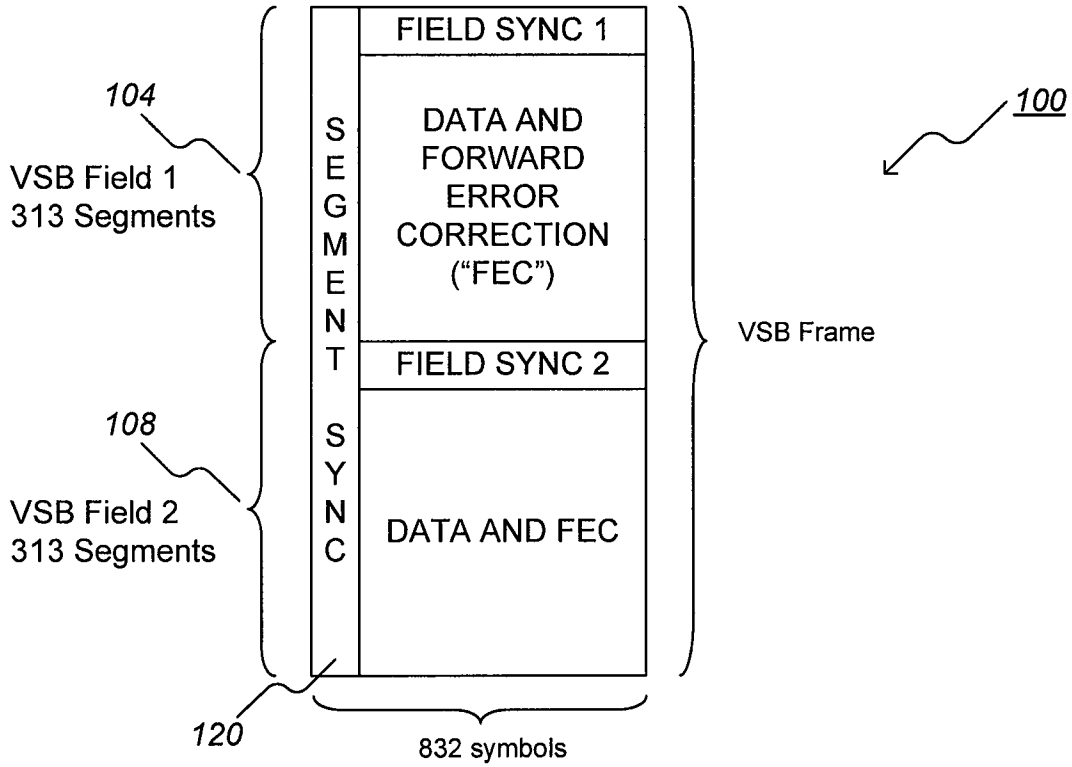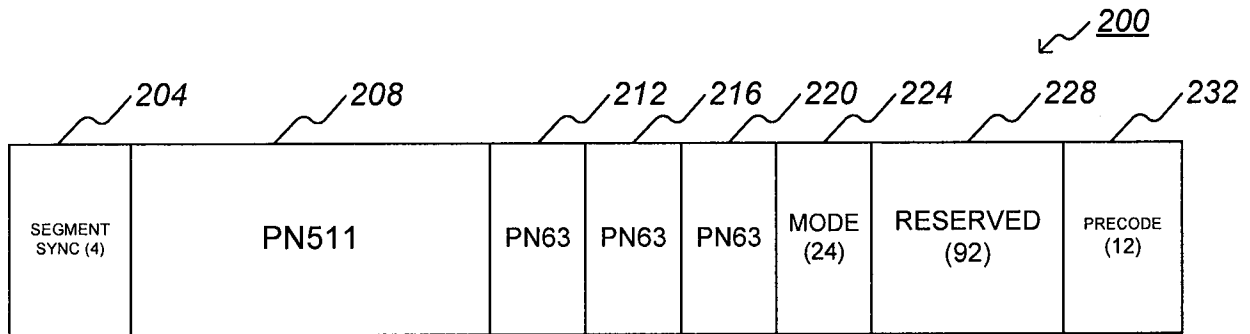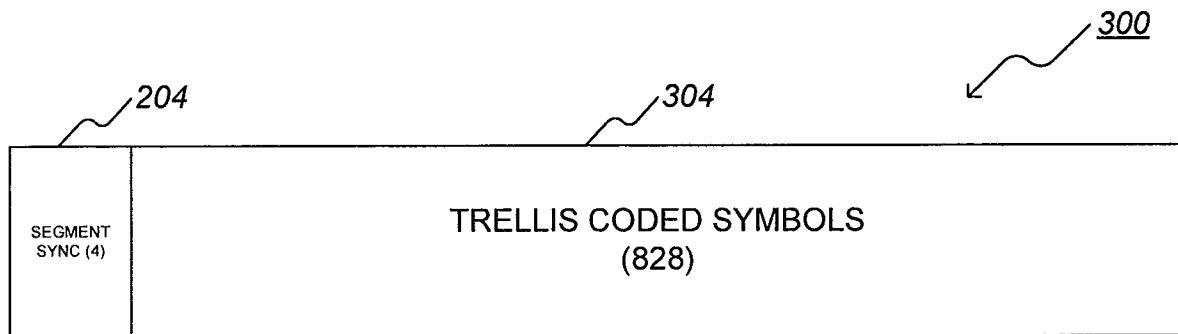
104

VSB Field 1
313 Segments

108

VSB Field 2
313 Segments

120

S E G M E N T S Y N C	FIELD SYNC 1
	DATA AND FORWARD ERROR CORRECTION ("FEC")
	FIELD SYNC 2
	DATA AND FEC

100

VSB Frame

832 symbols

FIG. 1

200

/204	/208	/212	/216	/220	/224	/228	/232
SEGMENT SYNC (4)	PN511	PN63	PN63	PN63	MODE (24)	RESERVED (92)	PRECODE (12)

FIG. 2

300

/204	/304
SEGMENT SYNC (4)	TRELLIS CODED SYMBOLS (828)

FIG. 3

RF

| randomizer | Reed-Solomon encoder | interleaver | trellis encoder | data frame formatter | pilot insertion | modulator |

404 · 408 · 412 · 416 · 420 · 424 · 428

field sync
segment sync

FIG. 4

400

RF

| tuner | demodulator | sync and timing recovery | matched filter | equalizer | phase tracker | trellis decoder | deinterleaver | Reed-Solomon decoder | derandomizer |

504 · 508 · 512 · 516 · 520 · 524 · 528 · 532 · 536 · 540

FIG. 5

500

FIG. 6

FIG.7

FIG. 8

FIG. 9A

_991_

_993_

RECEIVE DATA INCLUDING CODED SYMBOLS AND UNCODED SYMBOLS

_994_

DETERMINE FIRST ERROR ESTIMATE BASED ON DECODED SYMBOLS

_995_

DETERMINE SECOND ERROR ESTIMATE BASED ON UNCODED SYMBOLS

_996_

ADAPTIVELY SELECT FIRST ERROR ESTIMATE OR SECOND ERROR ESTIMATE BASED ON A CONVERGENCE CRITERION
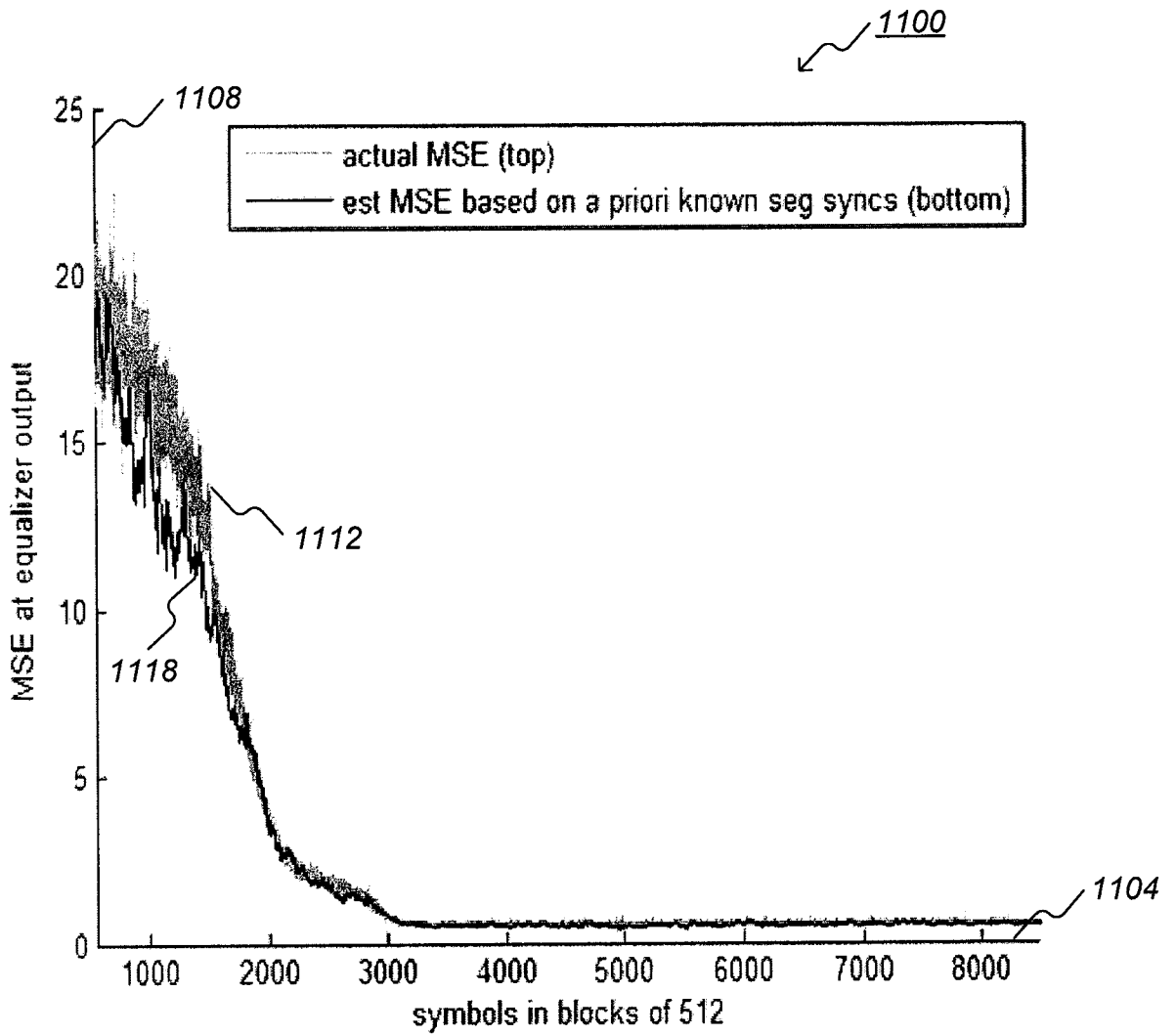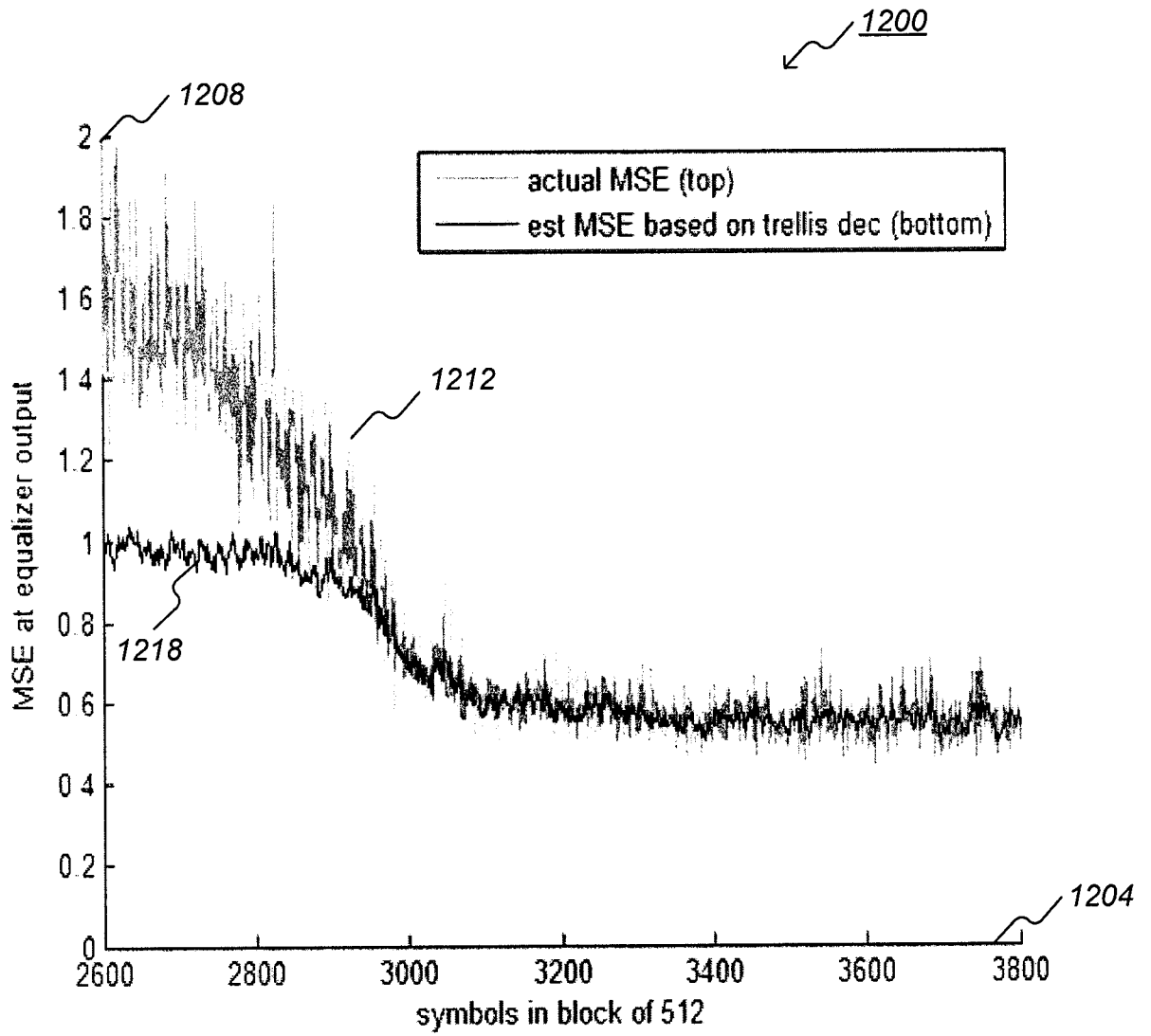
_997_

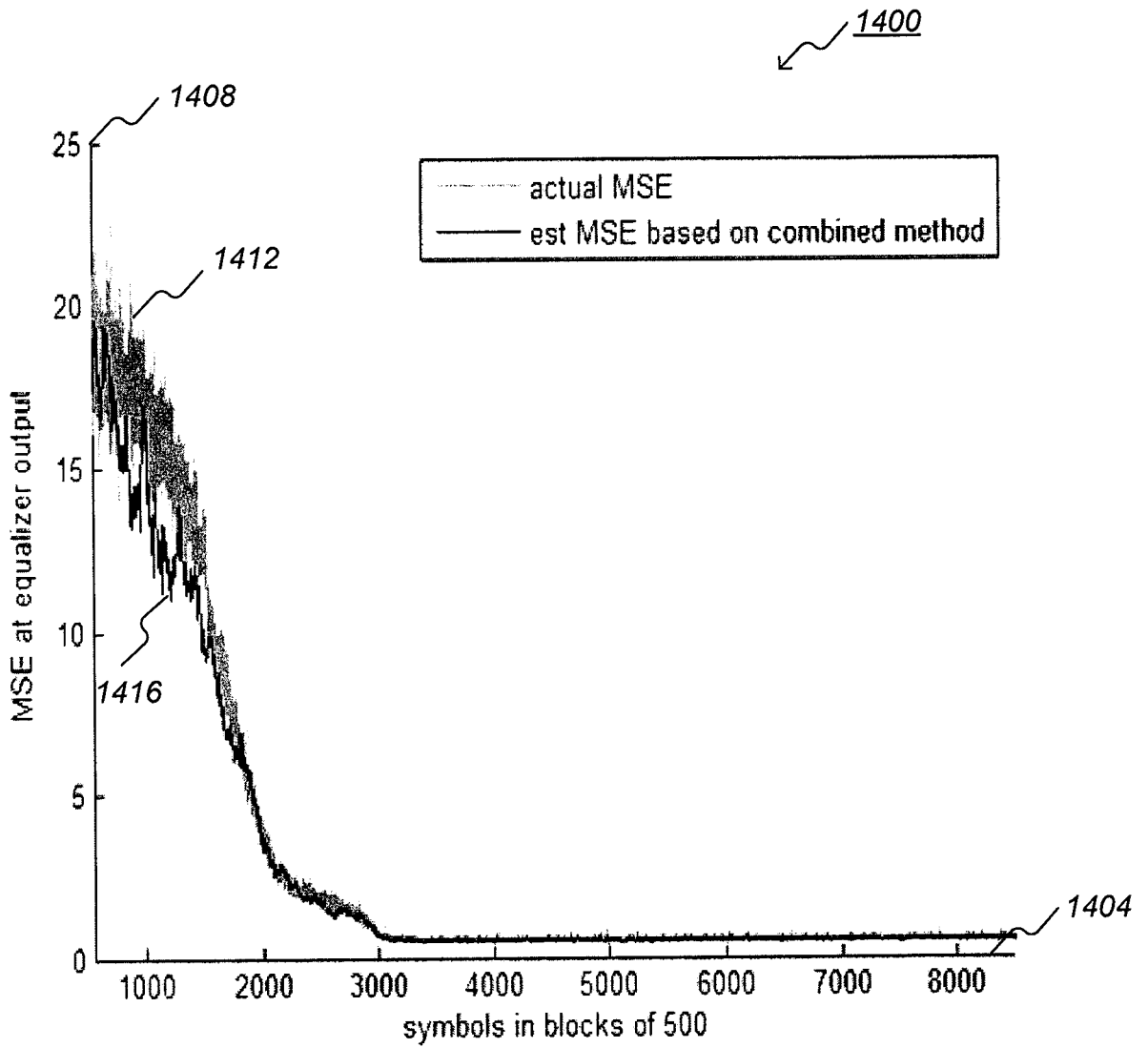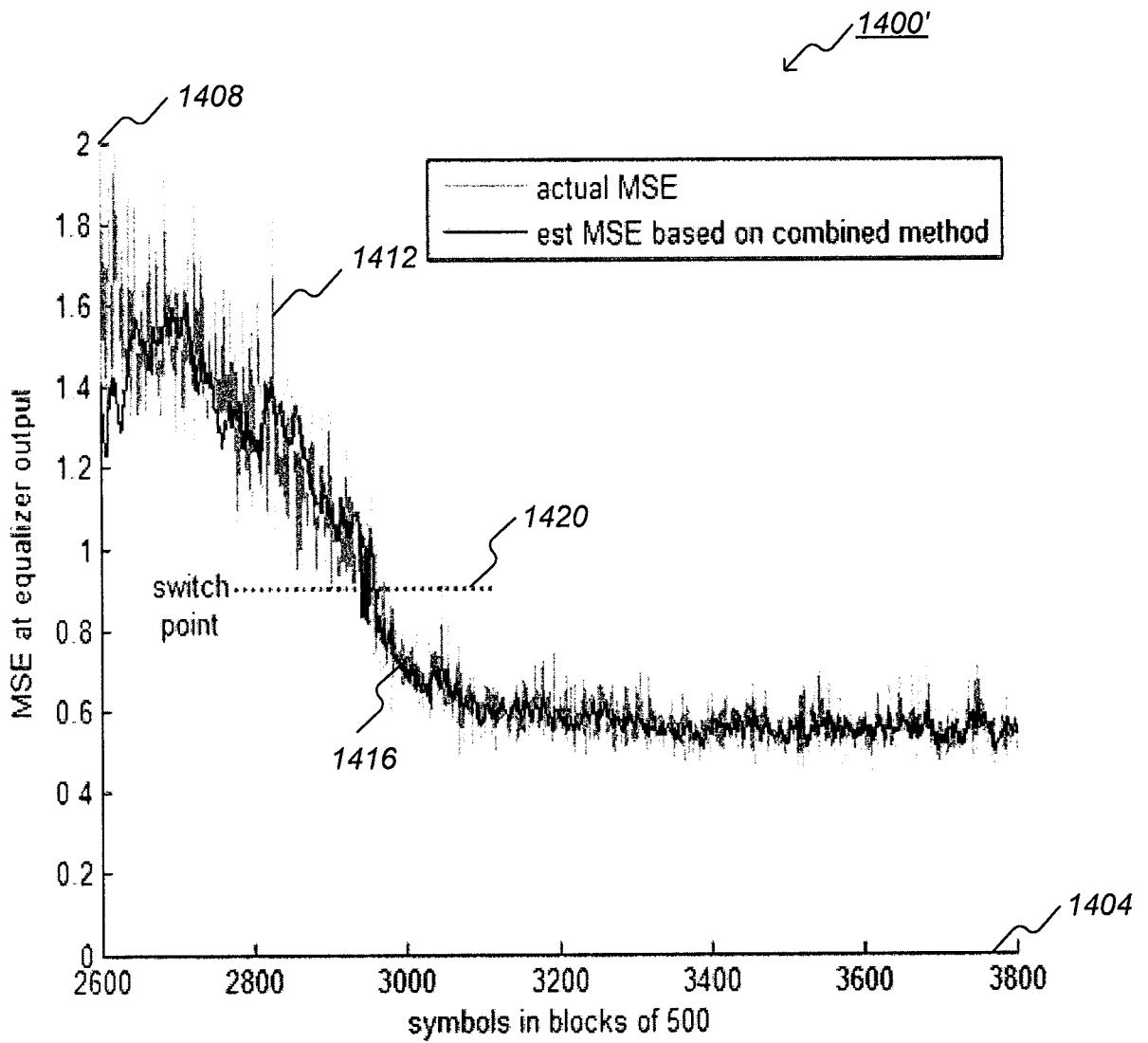DETERMINE STEP SIZE BASED ON THE SELECTED ERROR ESTIMATE

FIG. 9B

FIG. 9C

FIG. 10

FIG. 11

FIG. 12
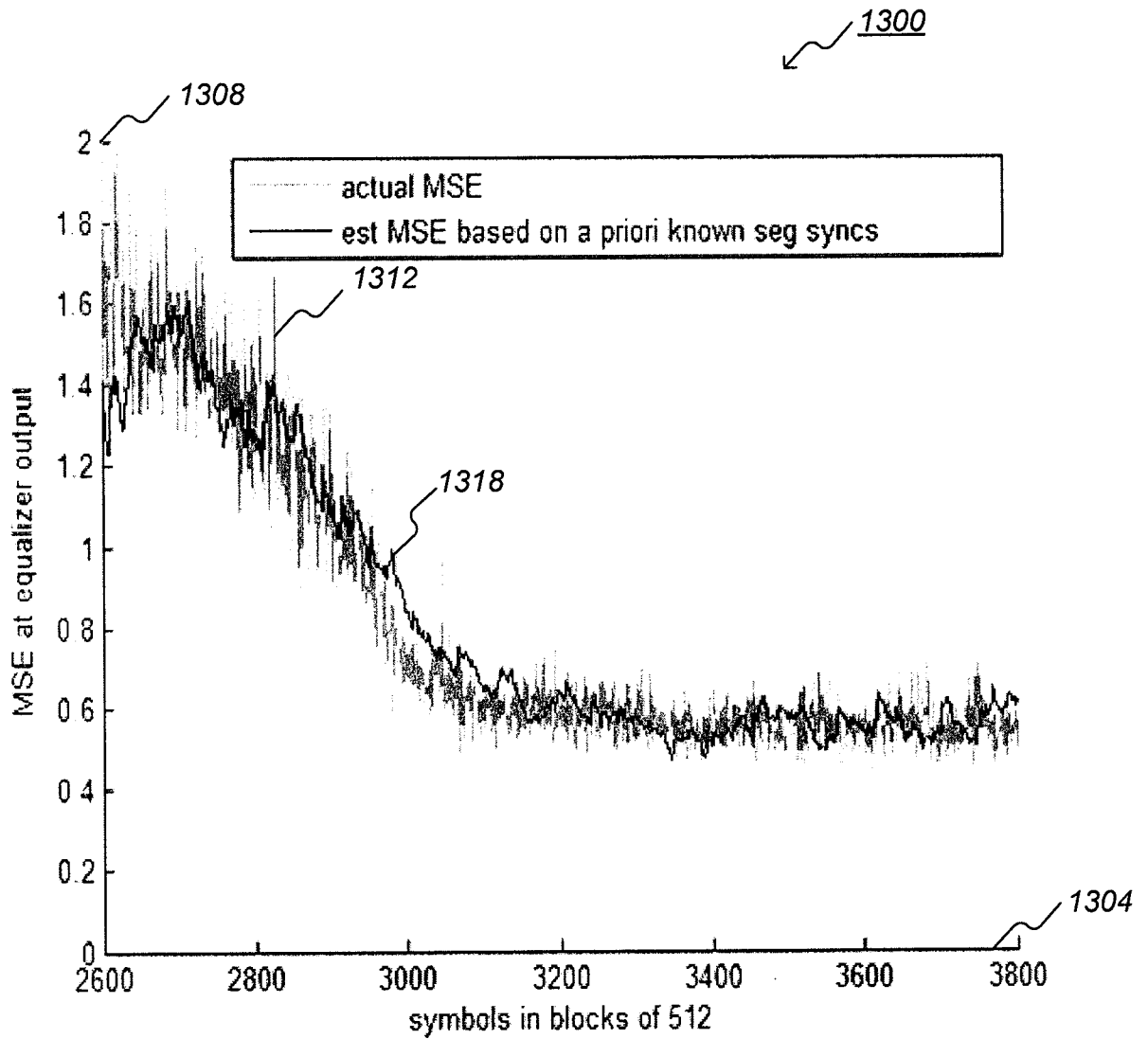
FIG. 13

FIG. 14

FIG. 15
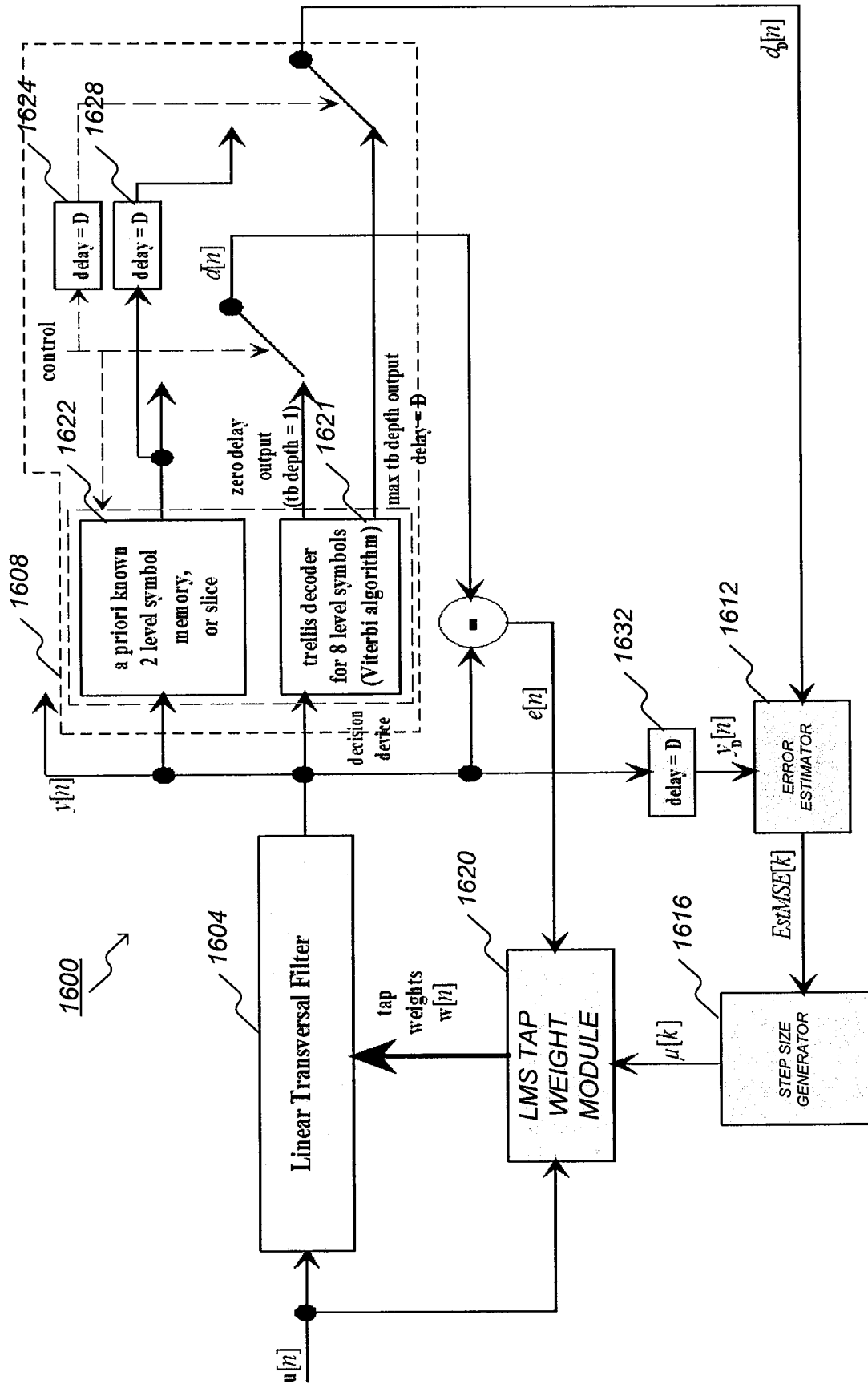
FIG. 16
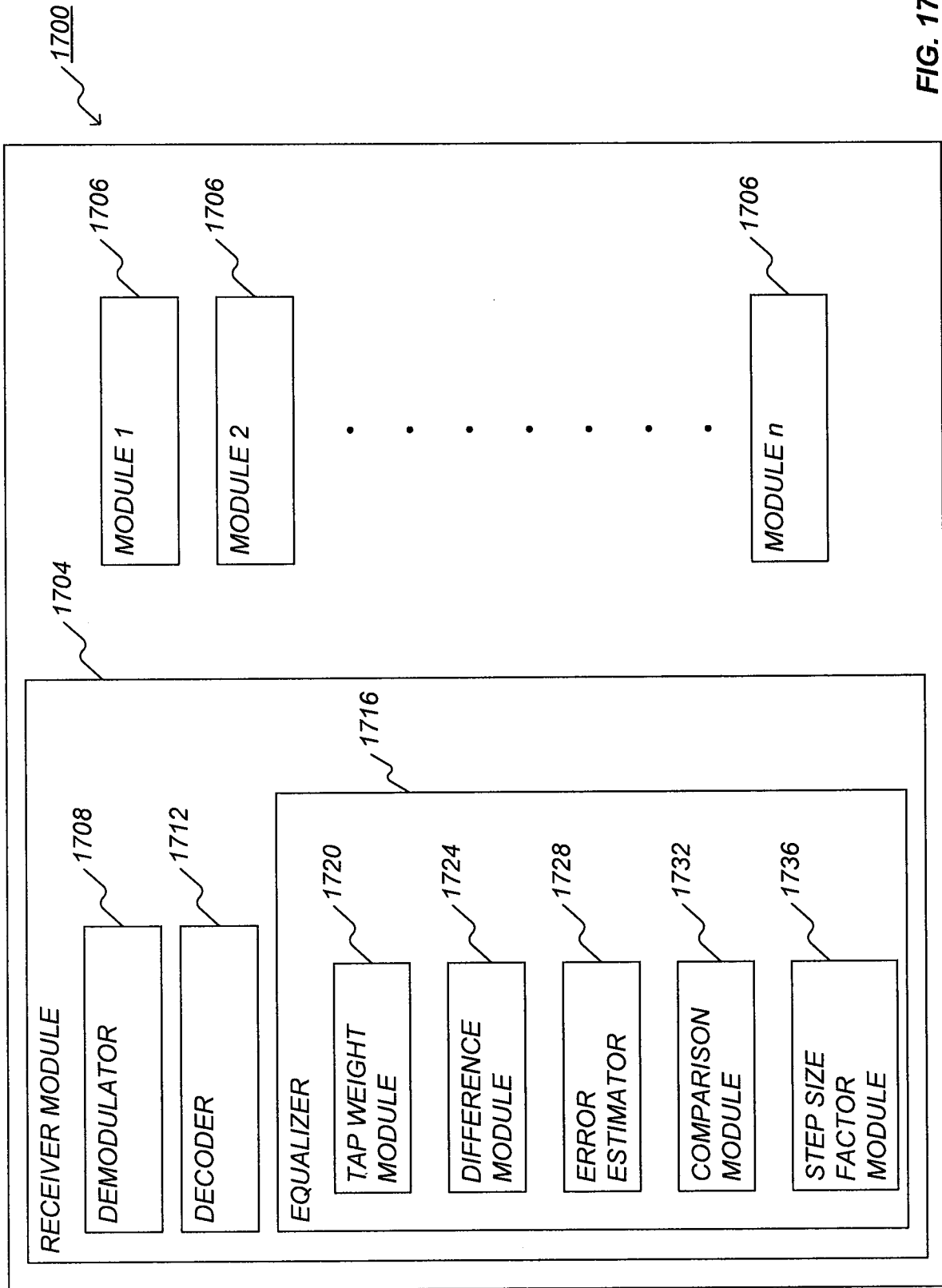
FIG. 17
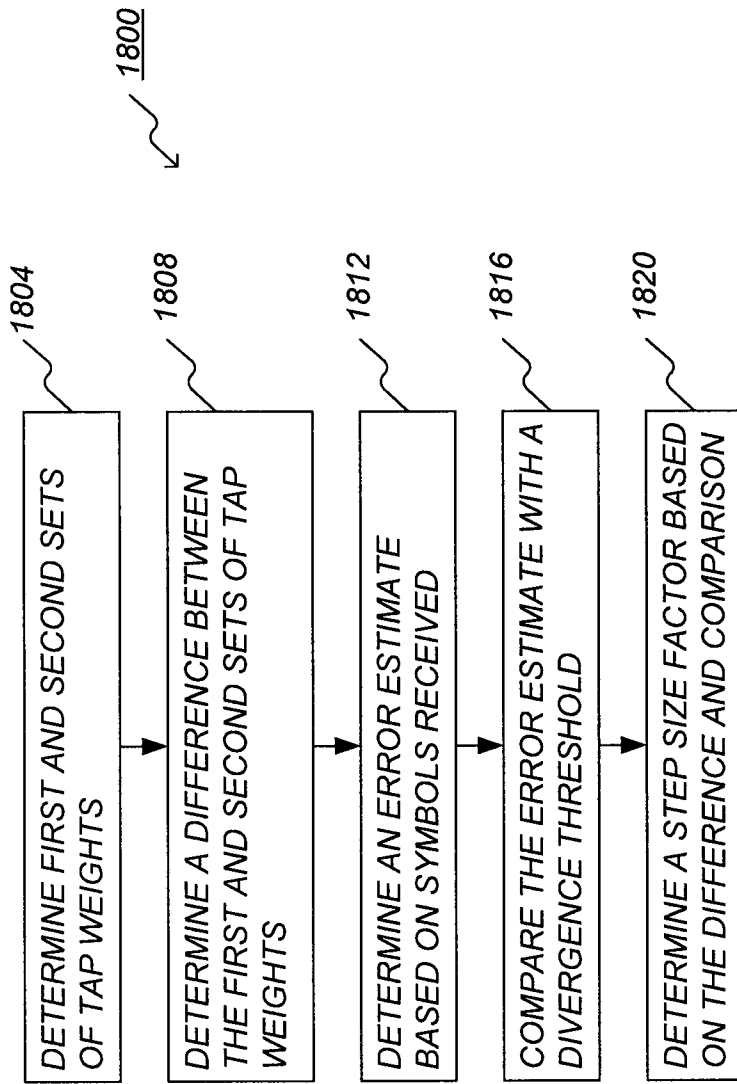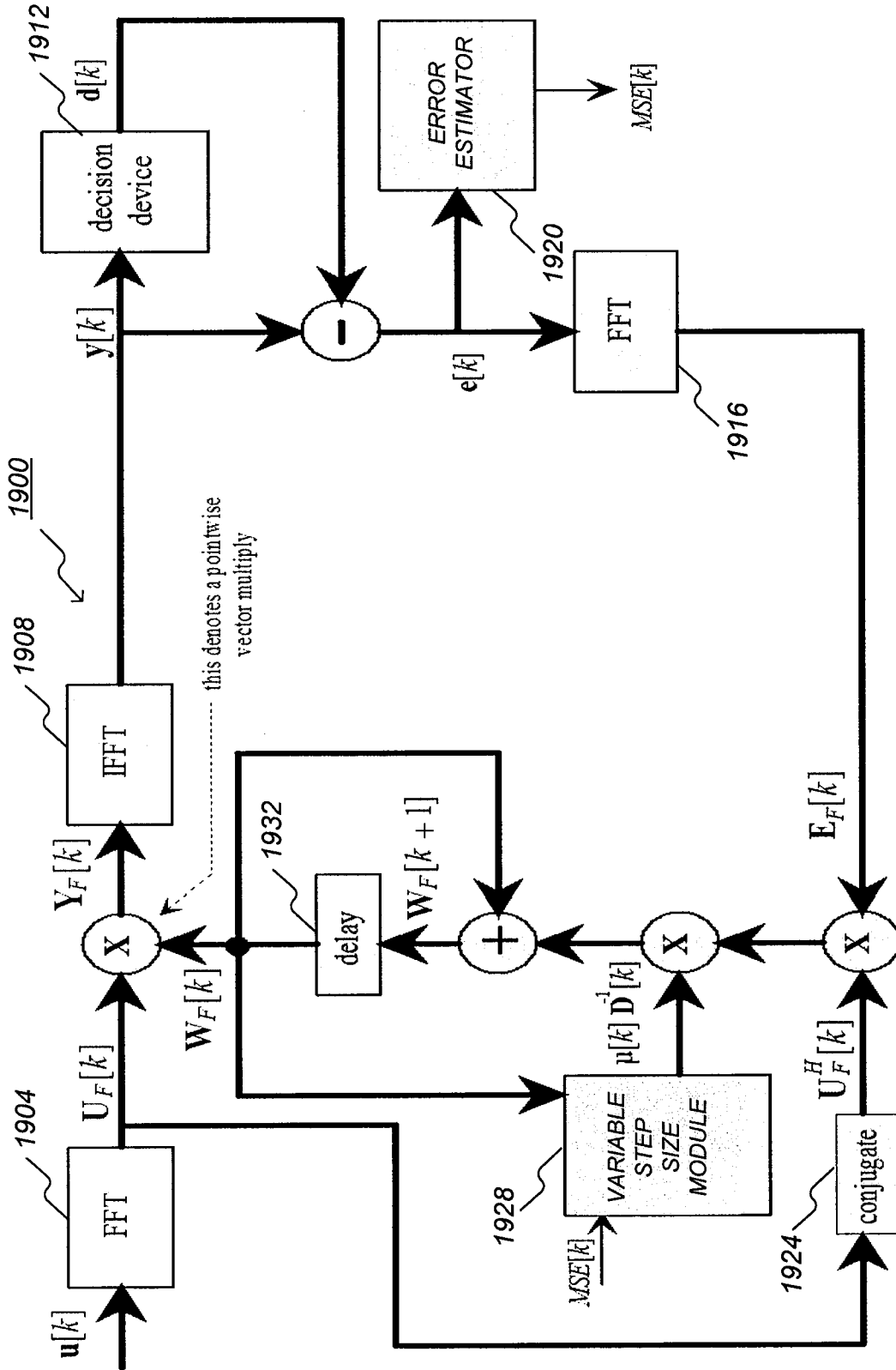
1800

1804 DETERMINE FIRST AND SECOND SETS OF TAP WEIGHTS

1808 DETERMINE A DIFFERENCE BETWEEN THE FIRST AND SECOND SETS OF TAP WEIGHTS

1812 DETERMINE AN ERROR ESTIMATE BASED ON SYMBOLS RECEIVED

1816 COMPARE THE ERROR ESTIMATE WITH A DIVERGENCE THRESHOLD

1820 DETERMINE A STEP SIZE FACTOR BASED ON THE DIFFERENCE AND COMPARISON

FIG. 18

FIG. 19

A. CLASSIFICATION OF SUBJECT MATTER

IPC: **H03H 7/30(2006.01)**

USPC: 375/232

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
    U.S. : 375/232,229,230,232,316

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
Please See Continuation Sheet

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 6,907,064 B1 (TOKUNAGA et al) 14 June 2005 (14.06.2005), figure 1, column 4, lines 42-55, column 4, line 56 to column 5, line 7, column 5, lines 23-40, column 3, lines 53-56	1, 6-10, 11-19
---		----------
Y	and column 1, lines 18-25.	2-5, 21-24
Y	US 4,027,258 (PERREAULT) 31 May 1977 (31.05.1977), abstract, column 1, line 60 to column 2, lines 48 and column 1, lines 42-47.	2-5, 20
Y	US 2003/0227968 A1 (KIM et al) 11 December 2003 (11.12.2003), paragraph 0012 and 0016).	21-24

☐ Further documents are listed in the continuation of Box C.   ☐ See patent family annex.

*	Special categories of cited documents:	"T"	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"A"	document defining the general state of the art which is not considered to be of particular relevance		
"E"	earlier application or patent published on or after the international filing date	"X"	document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"L"	document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"Y"	document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"O"	document referring to an oral disclosure, use, exhibition or other means		
"P"	document published prior to the international filing date but later than the priority date claimed	"&"	document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
28 August 2008 (28.08.2008)	18 SEP 2008

Name and mailing address of the ISA/US
        Mail Stop PCT, Attn: ISA/US
        Commissioner for Patents
        P.O. Box 1450
        Alexandria, Virginia 22313-1450
Facsimile No. (571) 273-3201
	Authorized officer
	
	  Kevin M. Burd
	
	Telephone No. 703-305-4900

Form PCT/ISA/210 (second sheet) (April 2007)

INTERNATIONAL SEARCH REPORT	International application No.
	PCT/US08/69433

Continuation of B. FIELDS SEARCHED Item 3:
EAST
search term: equalizer, step size, divergence, tap weights, equalizer coefficients