

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

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
10 December 2009 (10.12.2009)

PCT

(10) International Publication Number
WO 2009/149113 A2

(51) International Patent Classification:
H04L 25/03 (2006.01) *H03M 13/39* (2006.01)

(21) International Application Number:
PCT/US2009/046014

(22) International Filing Date:
2 June 2009 (02.06.2009)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
61/058,456 3 June 2008 (03.06.2008) US
12/476,066 1 June 2009 (01.06.2009) US

(71) Applicant (for all designated States except US): **NTT DOCOMO, INC.** [JP/JP]; Sanno Park Tower, 11-1, Nagata-cho, 2-chome, Chiyoda-ku, Tokyo 100-6150 (JP).

(72) Inventors; and

(75) Inventors/Applicants (for US only): **PAPADOPOULOS, Haralabos** [GR/US]; 390 South Cypress Avenue, San Jose, CA 95117 (US). **SUNDBERG, Carl-Erik, W.** [SE/US]; 395 Ano Nuevo Avenue, Apt. 1107, Sunnyvale, CA 94085 (US).

(74) Agents: **VINCENT, Lester, J.** et al.; Blakely, Sokoloff, Taylor & Zafman LLP, 1279 Oakmead Parkway, Sunnyvale, CA 94085-4040 (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, CL, 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, MK, 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:

— without international search report and to be republished upon receipt of that report (Rule 48.2(g))

(54) Title: SOFT OUTPUT M-ALGORITHM RECEIVER STRUCTURES WITH GENERALIZED SURVIVOR SELECTION CRITERIA FOR MIMO SYSTEMS

(57) Abstract: A method and apparatus is disclosed herein for a modified soft output M-algorithm. In one embodiment, the soft output M-algorithm is employed by a receiver in a wireless communication system to receive information-bearing signals wirelessly transmitted from the transmitter wirelessly transmitted, the receiver comprising: an inner decoder structure having a multiple-in multiple-out (MIMO) joint demapper to perform joint inner demapping over each tone, the joint demapper being operable to apply a soft-output M-type algorithm to identify survivor candidates at each depth in a detection tree being searched for each tone, including surviving full-length candidates, based on at least one metric and at least one other criterion, where a number of best alternatives from every level of the tree are expanded along with one or more alternatives selected meeting the at least one other criterion and where soft-output related information is collected and stored for each bit, and an outer decoder operable with the inner decoder to perform iterative decoding.



WO 2009/149113 A2

SOFT OUTPUT M-ALGORITHM RECEIVER STRUCTURES WITH GENERALIZED SURVIVOR SELECTION CRITERIA FOR MIMO SYSTEMS

PRIORITY

[0001] The present patent application claims priority to and incorporates by reference the corresponding provisional patent application serial no. 61/058,456, titled, "Adaptive Soft Output Generalized M-Algorithm Receiver Structures for MIMO/OFDM/QAM Systems with BICM/ID," filed on June 3, 2008.

FIELD OF THE INVENTION

[0002] Embodiments of the present invention relate to the field of adaptive reduced-complexity receiver structures for receiving information over wireless systems with multiple transmit antennas and multiple receive antennas; more particularly, embodiments of the present invention relates to inner/outer decoder structures with an optimal outer decoder and inner decoders based on a forward-backward version of the soft output M-algorithm are employed.

BACKGROUND OF THE INVENTION

[0003] Future wireless systems require efficient utilization of the radio frequency spectrum in order to increase the data rate achievable within a given transmission bandwidth. This can be accomplished by employing multiple transmit and receive antennas combined with signal processing. A number of recently developed techniques and emerging standards are based on employing multiple antennas at a base station to improve the reliability of data communication over wireless media without compromising the effective data rate of the wireless systems. So called space-time block-codes (STBCs) are used to this end. Specifically, recent advances in wireless communications have demonstrated that by jointly encoding symbols over time and transmit antennas at a base station, one can obtain reliability (diversity) benefits as well as increases in the effective data rate from the base station to each user. These multiplexing (throughput) gain and

diversity benefits depend on the space-time coding techniques employed at the base station. The multiplexing gains and diversity benefits are also inherently dependent on the number of transmit and receive antennas in the system being deployed, in the sense that they are fundamentally limited by the multiplexing-diversity trade-offs curves that are dictated by the number of transmit and the number of receive antennas in the system.

[0004] For high data rates and wideband transmission the use of OFDM makes the equalizer unnecessary. With multilevel modems, coded modulation systems can easily be designed by use of an outer binary code, e.g., a convolutional code and an interleaver in a so called bit-interleaved coded modulation (BICM) system. One such class of systems employing BICM are MIMO/OFDM/BICM/ID systems. Such systems can also employ an inner orthogonal or quasi-orthogonal space-time block code, although typically they will not. Also, the transmit antennas need not be collocated, although, typically they are collocated.

[0005] A number of receiver structures exist as options for transmission systems. Many of these designs include an inner-outer decoder structure, whereby the outer decoder is optimally selected. The designs include iterative decoding (ID) receivers with a MAP-based inner decoder, ID systems with a MaxLogMAP-based inner decoder, receivers using QRD/M-Algorithm based inner decoder, MMSE-based inner decoders, tree-search based inner decoders based on conventional SOMA, on SOMA versions with tree reordering, and on forward-backward SOMA versions with tree ordering.

[0006] ID receivers that have a MAP-based inner decoder use the optimum inner decoder and have the optimum bit-error-rate performance among all inner/outer decoder structures. However, the MAP-based inner decoder becomes computationally intractable as the number of transmit antennas (which equals the number of QAM symbols that need to be jointly resolved), referred to herein as N , and the number of bits represented by each QAM symbol, referred to herein as B , increase.

[0007] ID systems with a MaxLogMAP-based inner decoder have less complexity than the MAP-based system and are asymptotically (high SNR) optimal

in that they have near optimum bit-error-rate performance at high SNR. However, the MaxLogMAP-based inner decoder also becomes computationally intractable as N and B increase.

[0008] Systems with receivers using QRD/M-Algorithm based inner decoder also use a variant of the M-algorithm to produce hard bit estimates along with reliability information. They perform a limited tree-search whereby at every level of the tree only the M best candidates are kept and expanded through the next level in the tree. As a result, they can yield drastic reductions in complexity by proper choice of the M parameter, at a cost in bit-error-rate performance. These methods directly employ the “hard-output” M-algorithm, to generate hard-output estimates, and then employ the resulting M full-length candidates to obtain soft information. However, to generate soft information for any bit location, both values of the bit must be available in the pool of the remaining M candidates. As a result, when one of the bit-values is missing for a given bit location in the set M full-length candidates, these methods resort to heuristic (and inferior) softify-ing techniques to generate soft output for each bit. Also, these methods do not exploit iterative decoding.

[0009] MMSE-based inner decoders have much lower complexity but suffer in bit-error-rate performance, especially, at higher outer-code rates.

[0010] Systems employing a tree search based on conventional SOMA where reliability values are calculated recursively in the forward direction only will sometimes yield reliability values which are not calculated relative to the globally best sequence estimating the MAP or MLD output sequence; these typically also use a large number of early terminated paths to collect soft output at the intermediate levels in the tree.

[0011] Schemes involving forward-backward SOMA versions with tree reordering correspond to the best soft-output algorithms subject to the constraint that the survivor list at each stage consists of the paths with the best set of metrics. Some proposed algorithms do not necessarily include all the paths with the best metrics and can outperform the forward-backward SOMAs by providing bit log-

likelihood ratios (bit-LLRs), i.e., soft-output information, based on longer sequences.

[0012] Note also that there exist many other inner decoder structures, including spherical decoders, soft-output Viterbi-algorithm (SOVA) based inner-decoders, etc.

SUMMARY OF THE INVENTION

[0013] A method and apparatus is disclosed herein for a soft output M-type algorithm. In one embodiment, the soft output M-type algorithm is employed by a receiver in a communication system to receive information-bearing signals transmitted from the transmitter, where the receiver comprises: an inner decoder structure having a multiple-in multiple-out (MIMO) joint demapper to perform joint inner demapping over each tone, the joint demapper being operable to apply a soft-output M-type algorithm to identify survivor candidates at each depth in a detection tree being searched for each tone, including surviving full-length candidates, based on at least one metric and at least one other criterion, where a number of best alternatives from every level of the tree are expanded along with one or more alternatives selected meeting the at least one other criterion and where soft-output related information is collected and stored for each bit, and an outer decoder operable with the inner decoder to perform iterative decoding.

BRIEF DESCRIPTION OF THE DRAWINGS

[0014] The present invention will be understood more fully from the detailed description given below and from the accompanying drawings of various embodiments of the invention, which, however, should not be taken to limit the invention to the specific embodiments, but are for explanation and understanding only.

Figure 1 is a flow diagram of one embodiment of a decoding process.

Figure 2 is a block diagram of one embodiment of a transmitter for space-time coding with bit-interleaved coded modulation (BICM) and OFDM where it is assumed as an example that the outer binary code is a convolutional code.

Figure 3 is a block diagram of one embodiment of a receiver having an iterative decoder for the coded OFDM system shown in Figure 2.

Figure 4 is a block diagram of one embodiment of the MIMO demapper having MIMO joint demapper units with distinct demappers for the different OFDM tones for the MIMO/OFDM system with BICM/ID.

Figure 5A illustrates a set partition type mapper for 16 QAM.

Figure 5B illustrates a Gray mapper for 16 QAM.

Figure 6 illustrates the decision tree that allows a recursive computation of metrics on a tree in the case that there are three transmit antennas.

Figure 7 illustrates an example of a decision tree.

Figure 8 is a flow diagram of one embodiment of a process for setting up the SOMA inner decoding operation on a tone.

Figure 9 illustrates the result of a QR decomposition.

Figure 10 illustrates the indeterminable LLR problem that arises in generating soft output for all bit locations based on full length candidates, where all full-length candidates in this example agree in their decisions for bit locations 1, 2, and 4.

Figure 11 illustrates a flow diagram describing the selection of survivors and terminated paths at tree depth n .

Figure 12 illustrates a process with an example involving a tree in which the best path is shown.

Figure 13 shows two tables and a list of ordered paths.

Figure 14 shows an alternative embodiment of the two tables and a list of ordered paths.

DETAILED DESCRIPTION OF THE PRESENT INVENTION

[0015] Embodiments of the present invention deal primarily with the forward link, i.e., the base-to-mobile direction of transmission and more specifically reduced complexity receiver structures for these systems. Methods and apparatuses are disclosed for adaptive reduced complexity receiver structures. These receiver structures consist of inner/outer decoder modules that exchange information. The

receiver structures can be single or multiple-iteration structures. In one embodiment, the inner/outer decoding structures exploit M-algorithm type searches. Specifically, in one embodiment, the inner/outer decoding structures use different criteria in selecting the survivor candidates at every stage in the M-type algorithm. This favorably improves the soft-output information provided by the inner decoder.

[0016] In one embodiment, the soft-output MIMO inner-decoder makes adaptive use of a new class of soft output M-algorithms, (SOMA). This soft-output MIMO detector is applied on every tone (or, sub-channel) in the OFDM system, as well as at every iteration in the decoding. The SOMA detector uses only a fraction of the total number of candidates in its MIMO detection process, thus a considerable complexity reduction. There is of course a tradeoff between the performance and the degree of complexity reduction. The number of candidates explored in the SOMA is controlled by the parameter M, the number of paths that are extended from each surviving node at every level in the detection tree. In the overall detection process, the number of inner/outer decoder iterations, I, also affects the total decoding complexity and the associated performance. Note that during any given iteration of the decoding operation, SOMA-based inner decoding is performed independently on each OFDM tone.

[0017] While a conventional SOMA conducts a reduced size tree search by keeping as its M survivors at each depth the best M candidates at that depth, other criteria can be employed in addition to the path quality metric to decide which paths are chosen as the survivor paths, so that the M survivor paths at any given stage may differ from the paths with the best-quality metrics. In one embodiment, the survivor list is augmented (some of the M best candidates in the survivor list are replaced) with paths that can improve the soft-output information provided on one or more bits.

[0018] In one embodiment, in the disclosed reduced-complexity algorithm, at each stage the list of paths, referred to as the “survivor list,” is selected subject to a modified set of criteria. These criteria allow the selection of some survivor paths that may have lower-quality metrics than some of the paths that are not included in the survivor list. The new criteria allow these paths to be selected as survivors

because, despite their lower-quality metric, they can improve the soft-output information that will be eventually produced by the algorithm. In particular, due the bit combinations these paths represent and their relationship to the bits-combinations of the rest of the surviving paths, their inclusion in the list allows soft-output information to be collected for some bit locations based on longer paths (with more reliable metrics) in the tree.

[0019] The techniques described herein can be utilized to improve “softified” M-algorithms, SOMA and forward-backward SOMA (FB-SOMA) algorithms and can also be used in conjunction with channel-adaptive methods that have been proposed as modifications to the basic SOMA algorithm for providing further complexity/performance benefits. In the FB-SOMA, some information is stored for soft-output calculations in the forward pass. Once the forward pass is complete and the best full-length path has been chosen, the algorithm also goes through a backward pass. During the backward pass, the reliability (soft-output) information on each of the bits is computed based on and the quality of the (available) best full-length path and the information stored at the various stages of the forward pass.

[0020] The new schemes described herein are referred to herein as SOMA with survivor selection (SuSe-SOMA) and FB-SOMA with survivor selection (SuSe-FB-SOMA).

[0021] In the following description, numerous details are set forth to provide a more thorough explanation of the present invention. It will be apparent, however, to one skilled in the art, that the present invention may be practiced without these specific details. In other instances, well-known structures and devices are shown in block diagram form, rather than in detail, in order to avoid obscuring the present invention.

[0022] Some portions of the detailed descriptions which follow are presented in terms of algorithms and symbolic representations of operations on data bits within a computer memory. These algorithmic descriptions and representations are the means used by those skilled in the data processing arts to most effectively convey the substance of their work to others skilled in the art. An algorithm is here,

and generally, conceived to be a self-consistent sequence of steps leading to a desired result. The steps are those requiring physical manipulations of physical quantities. Usually, though not necessarily, these quantities take the form of electrical or magnetic signals capable of being stored, transferred, combined, compared, and otherwise manipulated. It has proven convenient at times, principally for reasons of common usage, to refer to these signals as bits, values, elements, symbols, characters, terms, numbers, or the like.

[0023] It should be borne in mind, however, that all of these and similar terms are to be associated with the appropriate physical quantities and are merely convenient labels applied to these quantities. Unless specifically stated otherwise as apparent from the following discussion, it is appreciated that throughout the description, discussions utilizing terms such as "processing" or "computing" or "calculating" or "determining" or "displaying" or the like, refer to the action and processes of a computer system, or similar electronic computing device, that manipulates and transforms data represented as physical (electronic) quantities within the computer system's registers and memories into other data similarly represented as physical quantities within the computer system memories or registers or other such information storage, transmission or display devices.

[0024] The present invention also relates to apparatus for performing the operations herein. This apparatus may be specially constructed for the required purposes, or it may comprise a general purpose computer selectively activated or reconfigured by a computer program stored in the computer. Such a computer program may be stored in a computer readable storage medium, such as, but is not limited to, any type of disk including floppy disks, optical disks, CD-ROMs, and magnetic-optical disks, read-only memories (ROMs), random access memories (RAMs), EPROMs, EEPROMs, magnetic or optical cards, or any type of media suitable for storing electronic instructions, and each coupled to a computer system bus.

[0025] The algorithms and displays presented herein are not inherently related to any particular computer or other apparatus. Various general purpose systems may be used with programs in accordance with the teachings herein, or it

may prove convenient to construct more specialized apparatus to perform the required method steps. The required structure for a variety of these systems will appear from the description below. In addition, the present invention is not described with reference to any particular programming language. It will be appreciated that a variety of programming languages may be used to implement the teachings of the invention as described herein.

[0026] A machine-readable medium includes any mechanism for storing or transmitting information in a form readable by a machine (e.g., a computer). For example, a machine-readable medium includes read only memory (“ROM”); random access memory (“RAM”); magnetic disk storage media; optical storage media; flash memory devices; etc.

Overview

[0027] As set forth above, embodiments of the present invention perform an adaptive soft-output M-algorithm that uses modified criteria for selecting the survivors. Such criteria, for example, may be used for both SOMA, FB-SOMA, and softified SOMA. More specifically, the survivor-selection method in the M-algorithm has been modified to include soft-output information considerations. In particular, additional information is exploited, in addition to the metric for selecting the survivors at each depth in the tree, and as a result, the survivors are not necessarily the M best alternatives in terms of the metric of interest. That is, in order to obtain reliable soft information by the decoding soft-output algorithm, other modified selection criteria are employed to choose the survivor set that limits the search on the tree.

[0028] Figure 1 is a flow diagram of one embodiment of a decoding process for producing a first set of output data representing most likely transmitted bit estimation values and information about the reliability of each of these estimates. The process may be performed by processing logic that may comprise hardware (e.g., dedicated logic, circuitry, etc.), software (such as is run on a general purpose computer system or a dedicated machine), or a combination of both. In one

embodiment, the decoding process is performed by a receiver in the wireless communication system.

[0029] Referring to Figure 1, the process begins by performing a first decoding operation to produce a first set of output data representing most likely transmitted bit estimation values and information about the reliability of each of these estimates, including performing a detection process over each tone for joint inner demapping, by applying a modified soft-output M algorithm to identify survivor candidates at each depth in a detection tree being searched for each tone, including surviving full-length candidates, based on at least one metric and at least one other criterion, where the best from every level of the tree are expanded along with one or more alternatives selected meeting the at least one other criterion and where soft-output related information is collected and stored for each bit (processing block 102).

[0030] In one embodiment, at any given tree depth in the M-algorithm, the demapper orders all paths according to their metric, and uses the additional criteria, besides the metric order, to select the survivor paths. The demapper takes into account the bit locations for which all paths in the survivor list agree in value to select the survivor paths. In one embodiment, the demapper constructs the survivor list by first adding the candidate with the best metric to the list (or more generally, by adding the M candidates with the best metrics to the list), and then iteratively adding members in the list based on their relative metrics, as well as the effect their addition to the survivor list would have on the number of bit locations for which all paths in the survivor list agree in value.

[0031] The process also includes calculating a soft output value for each bit by comparing a metric of partial path from an estimated best path with the metric of the longest and best visited path with an opposite decision on that bit (processing block 103).

[0032] In one embodiment, a selection process is used for a modified version of a forward-backward SOMA algorithm, whereby the survivor list at each depth consists of the union of the M' paths with the best metrics and another set of

M_e candidates that are selected based on their metrics as well bit-LLR considerations. This will be described in greater detail below.

[0033] Figure 10 provides an illustration of an indeterminable LLR problem. This is an important problem that arises in many soft-output algorithms relying on a reduced tree-search based on an M-algorithm. In particular, the problem arises in the process of generating soft output for all bit locations based on full-length candidates. All full-length candidates in the example agree in their decisions for bit locations 1, 2, and 4. As a result, there are no full-length candidates available with alternative decisions for bits 1, 2, and 4 (i.e., candidates with decisions 0, 1, and 1, respectively, on those bits).

[0034] The SOMA and FB-SOMA deal with this problem by computing bit-LLRs (soft output information) for those bit locations based on metric comparisons of shorter-length paths. In particular, as explained above, each bit-LLR is computed as a difference between the metrics of two paths. One of the paths is an early-terminated path at some depth “n” in the tree, while the other corresponds to a survivor path at the same depth. Both the SOMA and the FB-SOMA have some very attractive features in the way they obtain reliable soft output information based on full-length and early-terminated paths. However, they are limited in that they rely on an M-algorithm to generate the paths for which metrics will be computed throughout the tree. Although such an M-algorithm keeps as survivors at each depth the set of paths with the best metrics, and it can in principle be tuned to find the best full-length path with high probability, unfortunately, it makes no provisions to have long paths with alternative bit decisions (other than indirectly through increasing the number of survivors at each depth). As a result, it is likely that for a subset of the bit locations, reliability is computed at small depths (based on short paths). This implies that only a subset of the measurements is used for computing reliability for these bits. The problem is somewhat alleviated if a tree-preordering step is applied as described in U.S. serial no. 12/335,389, entitled “Tree Position Adaptive Soft Output M-Algorithm Receiver Structures”, filed December 15, 2008, which is incorporated by reference. Specifically, the quality of the information provided by the processed measurements employed at different tree depths, degrades with tree

depth. As a result, although the SOMA and FB-SOMA have to rely on partial paths for some bit-LLR calculations, tree preordering alleviates the effect of the metric approximation based on partial (as opposed to full-length) paths. On the other hand, making provisions to include members in the survivor list not just based on metric quality but by also trying to increase the length of paths used to compute bit-LLRs will in general boost the quality of the soft output information provided by the inner decoder.

[0035] One embodiment for the reduced complexity receiver is a modified version of a soft output M-algorithm (SOMA), which is used adaptively with a forward-backward tree search, and whereby the survivor-selection method in the M-algorithm has been modified to include soft-output information considerations. In particular, additional information is exploited, besides the metric for selecting the survivors at each depth in the tree, and as a result, the survivors are not necessarily the M best alternatives in terms of the metric of interest. Specifically, keeping in mind that it is of interest to obtain reliable soft information by the decoding soft-output algorithm, other modified selection criteria are employed to choose the survivor set that limits the search on the tree.

[0036] Figure 11 depicts a flow diagram of one embodiment of a selection process for a modified version of a forward-backward SOMA algorithm, whereby the survivor list at each depth consists of the union of the M' paths with the best metrics and another set of M_e candidates that are selected based on their metrics as well bit-LLR considerations. The process is performed by processing logic that may comprise hardware (circuitry, dedicated logic, etc.), software (such as is run on a general purpose computer system or a dedicated machine), or a combination of both. The results of the process are the survivor list and the early-terminated list at a fixed but arbitrary depth “n”.

[0037] Referring to Figure 11, the inputs to the process are the following: (i) the list of survivors at depth “n-1” and their metrics; (ii) the “effective” observation/measurement at depth “n” (i.e., the nth output sample from the QR decomposition; see Figure 9); (iii) the set of Q QAM constellation symbols. First, for each survivor at depth “n-1”, processing logic uses these inputs to compute

metrics for all Q paths emanating as length-1 extensions of the given survivor (one path per constellation symbol). Given there are M_p survivors at depth “ $n-1$ ”, path metrics are thus computed for a total of $Q M_p$ paths. That is, the paths of the M_p survivors at depth $n-1$ are extended and their length- n metrics are computed. This is performed in a manner well known in the art. Each length- n metric is computed as the sum of length-“ $n-1$ ” metric of the parent node (survivor node at previous depth) and the “branch” metric of the associated length-1 extension.

[0038] Next, processing logic orders (sorts) these paths based on their metric with the first element in the list corresponding to the path with the best metric and the last one corresponding to the path with the worst path metric. After ordering the paths, processing logic splits this sorted list into two sets. The top M' elements (where $M' \geq 1$ may be a preselected or adaptively chosen parameter, similarly to the M parameter in the conventional M -algorithm) are added to the survivor list, while the rest of paths are kept in a “Rest of Paths” list.

[0039] After splitting the sorted list, the binary representations of the M' length- n survivor paths are examined (they are all nB -bits long). Processing logic compares these binary codewords in order to determine the set of bit locations where all M' paths agree in value. The resulting list of bit locations where all paths agree is referred to herein as the “Common-Bits” list. If this list is empty, the survivor-selection process terminates at depth “ n ”. In the case that this list is not empty, for each bit location in this list, processing logic also examines the bit values of each of the paths in the “Rest of the Paths” list, one at a time, starting from the top (best) of the “Rest of the Paths” list. In one embodiment, this is accomplished by the following. First the “Common-Bits” list is copied onto a new list, referred to herein as the “Common-Bits-Temp” list. Starting from the first path in the “Rest of the Paths” list, processing logic compares its bit-values in the bit locations listed in the “Common-Bits-Temp” list against the (common) values of the survivor list. If there is agreement in value in all these bit locations between this path and the survivor list, then processing logic dismisses the path. If, however, there are one or more bit-locations of disagreement, then processing logic copies the path (from the “Rest of the Paths” list) to the “To Keep” list. In that case, the bit locations of

disagreement are also removed from the “Common-Bits-Temp” list. The process continues with the next path in the “Rest of Paths” list. The process terminates if there are no more candidates in the “Rest of Paths” list, or the “Common-Bits-Temp” list is empty.

[0040] Then, letting “R” denote the number of elements in the “To Keep” list, processing logic appends a subset of M_e elements from the “To Keep” list to the survivor list, and the rest ($R-M_e$) of the elements in the “To Keep” list are kept as the early terminated paths. Note that the original SOMA (and the original FB-SOMA depending on how soft-output information is computed) employ $M_e=0$. In one embodiment, the set M' is predetermined and $M_e=R$. In another embodiment, the set M' is predetermined and M_e includes no more than K elements for some preset value K . In these two embodiments, the value of survivors M is variable, as it depends on R , the number of elements in the “To Keep” list.

[0041] There are also many other embodiments that employ iterative versions of the above process in which the list of M' “Survivors” and the R paths in the “To Keep” list are updated in an iterative fashion. For illustration, a couple of representative example processes are provided. In these examples, the number of survivors that will be kept in the end at depth n is a fixed, but arbitrary value, of M where $M \geq 1$ (the value M can vary from depth to depth). In one example process, some value for M' is set, e.g., $M'=1$. In this case, the example process selects as the only survivor in the list the best candidate (in terms of its length- n metric). Then the “To Keep” list is generated. If the size, R , of the “To Keep” list is such that $R+1 \geq M$, then the top $M_e=M-1$ members from the “To Keep” list are used to append the survivor list (total of M members) and the rest ($R-M_e$) paths from the “To Keep” list comprise the list of early terminated paths. If $M > R+1$, however, then appending all R paths in the “To Keep” list yields a list of survivors of size $R+1 < M$, so there is space for another $M-(R+1)$ candidates to be added to the survivor lists. These survivors are added from the original sorted path lists as follows: (i) the survivor set is deleted from the sorted list; (ii) the first $M-(R+1)$ paths in the resulting list are added to the survivor list. The process naturally, generalizes to the case that $M' > 1$. In that case, if the size of the “To Keep” list is such that $R+M' \geq M$, then $M_e=M-M'$

members from the “To Keep” list are used to append the survivor list (total of M members) and the rest $R-M_e$ are used to update the list of early terminated paths. Similarly, if $M > R+M'$, appending all R paths in the “To Keep” list yields a list of survivors of size $R+M' < M$, so there is space for another $M-(R+M')$ candidates to be added to the survivor lists. These survivors are added from the original sorted path lists as follows: (i) the survivor set is deleted from the sorted list; and (ii) the first $M-(R+M')$ paths in the resulting list are added to the survivor list.

[0042] In another embodiment, survivors are added to the list iteratively, one by one. At first the two candidates with the best two metrics are selected and put in the survivor list (assuming total number of survivors allowed, M , is at least as large as 2). In preparation for adding the next survivor to the list, first the “Rest of Paths” and “To Keep” lists are generated. Two candidates are considered for inclusion in the survivor list: (i) the best candidate in the set of “Rest of Paths” list; and (ii) the best candidate in the “To Keep” list (this candidate is also a member of the “Rest of Paths” list, although not necessarily its best candidate). If (i) and (ii) correspond to the same candidate, then that candidate is added to the survivor list and the process of adding the next survivor is repeated. If candidates (i) and (ii) differ, a criterion is employed for selecting the next candidate. One example involves the case where the criterion is to always select candidate (i), in which case, the method reduces to the traditional M algorithm, which selects as survivors, the M paths with the best metrics. When the criterion is to always choose candidate (ii) we obtain the embodiment of the previous paragraph. In general the (i) vs. (ii) selection criterion can take into account multiple criteria such as the relative metric differences between candidate (i) and candidate (ii); the depth of the tree; the size of the resulting “Common-Bits” list (one based on the survivor list and candidate (i), and another based on the survivor list and candidate (ii)), as well as other parameters. A generalized version of this embodiment includes joint comparison/selection of one or more candidates in an iterative fashion based on comparisons of multiple elements from the “Rest of Paths” and “To Keep” lists.

[0043] In one embodiment, the set of survivors selected at depth n include the *smallest* set of n -length paths visited with the following properties: (i) the best

M' candidates are included as survivors; (ii) for each bit-value and each of the nB bit locations (in each visited length-n path), the best M'' visited n-length candidates are also included. In one embodiment, to implement the case M'=1 and M''=1, two tables are used to determine which survivors other than the best path to keep in the survivor list, as each path is processed in an ordered fashion from the best path to the worst path. Figure 12 illustrates the process with an example involving a tree in which the best path is shown as 111100. Two other paths are identified that are not in the list of top paths. There are paths 101100 and 011110. Figure 13 shows two tables and a list of ordered paths. The tables are stored in a memory in the receiver and are used to make sure all the bit locations of the codewords are included and ensure that paths are selected where bit locations differ for each bit location. Note that the tables are only shown with four locations. In general the number of columns in each table is equal to the number of bits in the codewords at the given tree depth. The ordered paths are ordered in terms of their path metrics from best path at the top of the list to worse paths as one goes down the list. The process begins by traversing the list and checking paths one at a time to decide whether or not to append the path to the survivor list. In particular, a path is appended to the survivor list if the path "contributes" a bit-value b (with b=0,1) at some bit-location c (c is a number from the list {1, 2, ..., nB}) which is not in the survivor list (i.e., no other path in the survivor list has bit-value b at bit location c). For example, in the table of Figure 13, the best path, 1111, contributes all of its bits, which happen to all be 1's and completely fills the 1's table. Therefore, it is kept as a path and all the 1-value table entries are marked with an "x" (or equivalently, if they were initialized with a "0" value, they are incremented to "1" denoting that collectively in the current list of survivor paths, there is a single 1-value at each bit-location). The next path is processed and it has one bit, the fourth position being a 0, that is not present in the table. Therefore, this path is kept, and the fourth entry of the 0-value table is marked with an "x" (or, equivalently, it is incremented to "1" if it was initialized with a 0 value). The same occurs for the next path which has a 0 bit in its third position and no such bit exist yet in the 0's table. However, the next path, 1100, contains bits that are all duplicative of bits that have been in the previously

processed paths; therefore, it is not kept. This process continues through the list until all the memory locations in both the 1's and the 0's tables are covered, and the remaining paths are discarded.

[0044] In one embodiment implementing a generalization of this case involving $M' \geq 1$ and $M'' \geq 1$ (the M'' best candidates are kept for each bit location and bit value), two tables are also used to determine which survivors other than the best path to keep in the survivor list, as each path is processed in an ordered fashion from the best path to the worst path. In that case, all entries of the 1-value and 0-value tables are initialized with a "0" value. All the paths are considered again in progression, from best to worst as candidates for the survivor list. Let b_k denote the value of the k th bit of the path being considered for addition in the survivor list. At any given time in the algorithm the k th entry in the 0-value (1-value) table shows how many candidates in the survivor list have 0 (1) in their k th location. When a given candidate path is added in the list, for each bit location k from 1 to nB , if the path has a zero in location k , the k th entry of the 0-value table is incremented by 1, else the k th entry of the 1-value table is incremented by 1. One algorithm that accomplishes generating the desired list works as follows: first each of the first M' candidates is added in the survivor list, one at a time, and each time a candidate is added to the survivor list the associated entries of the 0-value 1-value tables are incremented by 1. Once the first M' candidates have been all added to the survivor list, the algorithm calculates the minimum value among all entries of the 0-value and the 1-value tables. If that value equals or exceeds M'' the algorithm terminates (and the survivor list corresponds to the best M' paths). If not, the algorithm proceeds through the remaining visited paths from best to worst. When a candidate path is considered, its value at each bit location (from location 1 to location nB) is checked against the tables. Let b_k denote the value of the k th bit of the path being considered for addition in the survivor list. If there exists a bit location k , such that $b_k=0$ ($b_k=1$), i.e., the candidate has a 0 (1) value at location k , *and* the k th 0-value (1-value) entry is less than M'' , then the candidate is added to the survivor set (and all the corresponding entries of the 0-value and 1-value tables are incremented by 1). Else, if for every bit location k , the value of the b_k -table is M'' or larger, then the

path is discarded. The process is repeated until all entries of the 0-value and the 1-value tables are at least as large as M'' . Note that this case allows for the use of improved MaxLogMap metrics in the soft-output computations when $M'' \geq 2$.

Figure 14 shows such an embodiment that uses two tables and a list of ordered paths for $M''=2$. Referring to Figure 14, as with the tables in Figure 13, each of the order paths is examined and the table is marked until all positions in the tables have been marked at least two times.

[0045] In one embodiment, a MaxLogMAP approximation is used to compute bit-LLR values for each bit location. In one embodiment, bit-LLRs are computed using the MaxLogMAP approximation based on candidates selected via an M-algorithm tree search, whereby the survivor selection criterion at each depth in the tree is such that at least one candidate is kept for each possible bit-value and bit-location. In this case the bit-LLR values are computed based on full-length sequences. In another embodiment, bit-LLRs are computed using an improved MaxLogMAP approximation that is also based on candidates selected via an M-algorithm search, whereby the survivor selection criterion at each depth in the tree is such that at least one candidate is kept for each possible bit-value and bit-location. In this case the bit-LLR values are computed based on an improved MaxLogMAP approximation using full-length sequences. In this case, the well-known improved MaxLogMAP approximation, using J surviving candidates per bit-value and bit locations, is as follows; for a given bit location, let $\alpha_1, \alpha_2, \dots, \alpha_J$ denote the (log-posterior) path metrics of the best full-length surviving paths that have bit-value 1 at the given bit location; also let $\beta_1, \beta_2, \dots, \beta_J$ denote the (log-posterior) path metrics of the best full-length surviving paths that have bit-value 0 at the given bit location; the bit-LLR for the given bit is then computed as follows:

$$\log \left[\frac{\sum_{j=1}^J e^{\alpha_j}}{\sum_{j=1}^J e^{\beta_j}} \right]$$

One way of computing the above bit-LLR estimates exploits the $\max^*$ term relationship of the following equation:

$$\max^*(x,y)=\max(x,y)+\log(1+e^{-|x-y|})$$

Example Embodiments of Wireless Communication Systems

[0046] A wireless communication system comprising a first device (e.g., a base station) having a transmitter and a second device having a receiver (e.g., a mobile terminal) to receive information-bearing signals from the transmitter wirelessly transmitted using OFDM and bit interleaved coded modulation is described. In one embodiment, the communication system described herein is a coded modulation system that includes transmitters that apply space-time coding with bit-interleaved coded modulation that is combined with a multi-carrier OFDM modulation and receivers that apply OFDM demodulation with iterative demapping and decoding. The systems described herein have N_t transmit antennas and N_r receive antennas. Each of the N_r receive antennas receives signals that are the sum of channel-distorted versions of the signals transmitted from the N_t transmit antennas. Such coded modulation systems in accordance with the present invention may be advantageously employed in wireless local/wide area network (LAN/WAN) applications.

[0047] In one embodiment, the space-time coding system described herein comprises OFDM for wideband transmission, MIMO and large QAM constellations for high spectral efficiency, a bit interleaver for the bit-interleaved coded modulation scheme (BICM) and an outer binary code. The overall detection is typically performed iteratively. This requires that both the inner MIMO demapper and the outer decoder perform soft in soft out (SISO) detection/decoding. One system component contributing to the complexity is typically the joint demapper as explained above. The outer decoder is less critical in terms of complexity. In one embodiment, the MIMO detector in principle works with any binary outer code. This code could be a turbo code, an LDPC code, a regular convolutional code or an RCPC code. The decoder for the outer code is preferably a soft in soft out (SISO) type decoder, for example a MAP decoder. The outer decoder supplies soft information to the inner MIMO detector for iterative decoding.

[0048] While the exemplary embodiment is described for space-time coding with bit-interleaved coded modulation, other types of coded modulation for space-time coding may be used. In addition, the exemplary embodiments are described for a mapping of the bit-interleaved coded data into symbols using QAM; however, other modulation schemes may be used, such as, for example, but not limited to phase-shift keying (PSK).

[0049] Generally, the receiver includes circuitry that estimates the values for the elements in channel response matrix $H[f]$, and such estimates may be generated using periodic test (pilot) signals transmitted by the transmitter to the receiver. Such a priori information of the channel impulse response may also be generated via simulations. The matrix $H[f]$ denotes the channel response over the f th OFDM tone and is a matrix of dimensions N_r by N_t .

[0050] When combined with signal processing, multiple transmit and receive antennas can yield communication links with increased bandwidth efficiency (data rate), extended power efficiency (range), or both. Embodiments of the invention deal primarily with the forward link, i.e., the base-to-mobile transmission direction of transmission. Methods and apparatuses are disclosed for adaptive soft output M-algorithm based receiver structures.

[0051] In one embodiment, a reduced complexity soft output MIMO detector in the receiver makes use of a soft output M-algorithm (SOMA) with modified survivor selection criteria, which is described in greater detail below. In one embodiment, the soft output MIMO demapper is applied for every tone or subchannel in the OFDM system, as well as at every iteration in the decoding process. To illustrate the advantages of the SOMA demapper, consider for illustration the optimum MIMO detector, referred to as a maximum a posteriori probability (MAP) detector. The MAP performs a joint demapping function over all the transmit antennas and over all the involved QAM constellation symbols and bits. Consider also for illustration the asymptotically optimum but simpler (with respect to the MAP) exhaustive MaxLogMAP detection algorithm. Even in the simpler MaxLogMAP detector an exhaustive demapping operation is required and it involves a search space that is growing exponentially with the product of the

number of transmit antennas (N_t) and the number of bits per QAM constellation point (B). For example, with a 6 by 6 MIMO system (6 transmit antennas and 6 receive antennas) using 64 QAM modulation (6 bits per constellation point) this product is 36. In such a case, the decoding complexity is of the order of 2^{36} , and the MaxLogMAP cannot be implemented with the technology of today. In contrast, in one embodiment, the SOMA detector only uses a small fraction of the total number of candidates in its MIMO detection process, thus the considerable complexity reduction. There is of course a tradeoff between the performance and the degree of complexity reduction.

[0052] In one embodiment, as described in further detail below, the SOMA is used adaptively, in that the tree-search symbol order when performing a detection search is adapted by each SOMA module (one per OFDM tone) according to the channel conditions (on the given OFDM tone) in order to optimize the overall complexity-performance trade-offs of the receiver. More specifically, during every inner/outer decoder iteration, a SOMA detector performs a SOMA detection process on each OFDM tone. In one embodiment, the number of candidates explored in the SOMA detection process is controlled by the parameter (M) that indicates the number of paths that are extended from each node or level in the detection tree. In particular, at any given level in the detection tree, only a subset M of the visited candidates are kept as survivors and are going to be extended at the next level. The rest of the candidates tested at this level are referred to as early-terminated paths. The early terminated paths are used by the SOMA for performing soft-output calculations. In one embodiment, the number of early terminated paths that are explored in the SOMA detection process is also an adaptation parameter as these paths also play a role in the soft output calculations. For purposes herein, this value is denoted T and is used in the soft output value calculation by the algorithm. In the overall detection process the number of inner/outer decoder iterations, I, also affects the total decoding complexity and the associated performance.

[0053] In one embodiment, the SOMA process performed by the receiver is referred to herein as a forward-backward SOMA (FB-SOMA), and is based on a forward pass and a backward pass through a detection tree. The FB-SOMA process

performs a search through the detection tree via a modified M-algorithm, while recursively collecting soft-output information. The FB-SOMA process also uses the value that the best visited full-length codeword takes on any given bit location as its hard-estimate for that bit. Furthermore, the soft information on this bit is computed as the difference between two, same-length path metrics, that differ in their value on the given bit location. Unlike the SOMA, however, in the FB-SOMA the soft information on this bit is computed as the difference between a metric of the best visited full-length codeword and the metric of the longest alternative-decision path visited, i.e., the best visited path with a different decision on the given bit from that of the best visited full-length path. Thus, the FB-SOMA soft-output on any bit is the metric difference between the best and longest paths visited with decisions 1 and 0 on that bit location.

[0054] More specifically, in the forward pass, the FB-SOMA process collects the best and longest path metrics associated with all the alternative bit decisions. In particular, assume that at depth $i - 1$, there are survivor paths for each bit value at a given bit location. If the depth- i survivor paths all agree in value at that bit location, then the best early terminated path with an alternative decision at depth i is used as the metric for the alternative decision path in the bit soft-output calculation. The other path used in this soft-output calculation is the length- i partial-metric of the best full-length path. To ensure that this partial metric is available at the end of the forward search, at depth i (for each i) all partial paths (lengths 1 to i) of all survivors at length i are stored and propagated through the tree. Then, in the backward pass, the nodes in the best full-length path are traced backwards and all the soft-output computations are computed against its partial metrics and the associated alternative-decision metrics.

[0055] In one embodiment, involving an OFDM based system with one MIMO detector for each tone, the complexity reduction in the SOMA based detector is implemented adaptively for each tone by use of symbol reordering followed by a SOMA-based decoder. In one embodiment, the reordering process is a relatively simple procedure based on CSI that is collected at the receiver (regardless of whether or not tree reordering is performed). The SOMA decoder

employed over each tone can be fixed (using fixed M, I, T parameters) or it can be channel adaptive. The M value used in the SOMA tree may be fixed or it may vary over the decoding tree. In such a system, a MaxLogMAP metric can be employed.

[0056] In one embodiment, the parameters M and T and/or I are selected adaptively for the best overall performance for a given total complexity level, with the quantity that guides the adaptivity being the quality of the different OFDM tones. For example, a high signal level or alternatively a large signal to noise ratio (SNR) for a certain tone means a good quality level for that tone. In such a case, the SOMA detection process performs decoding with a lower value of M, a lower value of T and potentially a lower value of I. On the other hand, for a tone of poor quality, i.e. a tone with low signal level or low SNR, the SOMA detection process performs decoding with higher values of M, T and I for the best use of the overall complexity. The adaptivity can also be extended over time, i.e. over successive OFDM symbols.

Transmitter and Receiver Embodiments

[0057] Figures 2 and 3 show the transmitter and receiver block diagrams for a MIMO/OFDM system with BICM and ID. More specifically, Figure 2 is a block diagram of one embodiment of a transmitter for space-time coding with bit-interleaved coded modulation (BICM) with OFDM modulation for wideband frequency selective channels. Referring to Figure 2, transmitter 200 comprises (as an example) convolutional encoder 201, bit interleaver 202, serial-to-parallel converter 203, mapper modems 207_1 - 207_{N_t} , inverse fast Fourier transform (IFFT) modules 208_1 - 208_{N_t} , and transmit antennas 209_1 - 209_{N_t} . Note that IFFT modules 208_1 - 208_{N_t} also include circular-prefix operations, which are performed in a manner that is well known in the art. Also the resulting sequences generated by the IFFT/circular prefix modules are converted into electrical continuous-time signals via pulse-amplitude modulation in a manner that is well known in the art.

[0058] To perform BICM encoding to the data, convolutional coder 201 applies a binary convolutional code to the input bits (input data) 210. Bit interleaver 202 then interleaves the encoded bits from convolutional coder 201 to generate bit-

interleaved encoded bits. This bit interleaving de-correlates the fading channel, maximizes diversity, removes correlation in the sequence of convolutionally encoded bits from convolutional coder 201, and conditions the data for increased performance of iterative decoding. Convolutional coder 201 and bit interleaver 202 may typically operate on distinct blocks of input data, such as data packets.

[0059] After performing bit interleaving, bit-mapping and modulation and OFDM are applied to the bit-interleaved encoded bits. Serial-to-parallel converter 203 receives the serial bit-interleaved encoded bit stream from bit interleaver 202. Note that serial-to-parallel converter 203 may include a framing module (not shown) to insert framing information into the bit stream, which allows a receiver to synchronize its decoding on distinct blocks of information. Serial-to-parallel converter 203 generates a word of length N_t long, with each element of the word provided to a corresponding one of mapper modems 207_1 - 207_{N_t} . Elements of the word may be single bit values, or may be B bit values where B is the number of bits represented by each modem constellation symbol.

[0060] Each of mapper modems 207_1 - 207_{N_t} converts B bits to corresponding symbols (of the Q -ary symbol space, with $Q=2^B$). The output of each modem mapper 207 is a complex-valued symbol (or equivalently two real-valued samples, representing the real and imaginary parts of the complex-valued symbol). Each of IFFT modules 208_1 - 208_{N_t} collects up to F symbols, and then applies the IFFT operation of length F to the block of F symbols. F is an integer whose value can typically range from 64 to 4096, or larger and depends on the available transmission bandwidth, the carrier frequency, and the amount of Doppler shifts that need to be accommodated by the system. Thus, each of IFFT modules 208_1 - 208_{N_t} generates F parallel subchannels that may be transmitted over the corresponding antenna among 209_1 - 209_{N_t} . Each subchannel is a modulated subcarrier that is transmitted over the channel.

[0061] In embodiment, the transmitter and receivers have an equal number of transmit and receive antennas, *i.e.*, $N_t = N_r = N$. The binary information-bearing signal, hereby denoted as u_k , is encoded first at the transmitter by an outer binary code using convolutional coder 201, generating a coded sequence c_k . This

sequence is interleaved by a bit interleaver 202. Then, each of mapper modems 207₁-207_{N_t} maps groups of B interleaved bits at a time into 2^B -QAM symbols. The resulting QAM symbols are multiplexed through the N transmit antennas 209₁-209_{N_t} in a round-robin fashion and OFDM transmission is applied over each antenna using IFFT modules 208₁-208_{N_t}. For convenience, for purposes herein, $s_k[n]$, the QAM symbol transmitted by antenna k on tone n , and via $b_k^\ell[n]$ the ℓ th out of the B bits is used as input in one of mapper modems 207₁-207_{N_t} to produce $s_k[n]$. Letting $\mathbf{b}_k[n] = [b_k^1[n], b_k^2[n], \dots, b_k^B[n]]$, then,

$$s_k[n] = \text{map}(b_k[n]) \quad (1)$$

where map denotes the mapper operation. Figures 5A and 5B show two such representative mappings for $B=4$. Figure 5A illustrates one embodiment of a so called set partition type mapper for 16QAM for use in iterative decoding. This type of mapper is sometimes suitable for BICM with iterative decoding (ID) in contrast to the Gray mapper which is always suitable for a non-iterative decoding process, and is shown in Figure 5B.

[0062] Figure 3 is a block diagram of one embodiment of a receiver having an iterative decoder for the space-time code for the OFDM system. Referring to Figure 3, receiver 300 comprises receive antennas 301₁-301_{N_r}, fast Fourier transform (FFT) modules 302₁-302_{N_r}, demodulator/detector 303, parallel-to-serial converter 307, bit deinterleaver 308, maximum a posteriori (MAP) decoder 309, bit interleaver 310, and serial-to-parallel converter 311. Although not shown, each of the FFT modules 302₁-302_{N_r} is preceded by front end that performs filtering, sampling, and a circular-prefix-removal operation.

[0063] For a wideband system, receiver 300 performs OFDM demodulation for each of receive antennas 301₁-N_r, and the demodulation and demapping is performed over F parallel subchannels. The i th receive antenna 301(i) senses a signal made up of various contributions of the signals transmitted from the N_t transmit antennas (i.e., contributions of the multiple F parallel, narrowband, flat fading subchannels transmitted over corresponding antennas 209₁-209_{N_t} of Figure

2). Each of FFT modules 302₁-302_{N_r} apply an F-point FFT to the corresponding signals of receive antennas 301₁-301_{N_r}, generating N_r parallel sets of F subchannels.

[0064] In one embodiment, demodulator/detector 303 estimates bits in each of the F subchannels. Multi-input multi-output (MIMO) demapper 305, based on the N_r parallel sets of F subchannels from FFT modules 302₁-302_{N_r} produces soft estimates of the demapped bits (i.e., bits mapped from the constellation symbol) in each of the F subchannels from the N_t antennas in the transmitter. MIMO demapper 305 produces the estimates of the demapped bits and reliability information about these bits using reliability information generated by soft-output decoding (followed by reinterleaving) by MAP decoder 309.

[0065] In one embodiment, MIMO demapper 305 computes hard values (i.e., 0/1 estimates) for the bits transmitted on the overlapping F subchannels, along with an estimate (approximation) of the a posteriori probability of the hard value being correct. Figure 4 is a block diagram of one embodiment of MIMO demapper 305 having MIMO joint demapper units for the different OFDM tones/subchannels. Referring to Figure 4, each signal of the N_r receive antennas 301₁-301_{N_r} is divided into F subchannels (via demodulator 304, not shown in Figure 4) by applying an FFT operation and sending its output to corresponding subchannel MIMO demappers 401₁-401_F. The signal outputs of the kth subchannel for all N_r receive antennas are provided to the kth subchannel MIMO demapper 401(k). After the first iteration reliability information is provided to the kth subchannel MIMO demapper 401(k), using extrinsic information generated by the output of MAP decoder 309 at the previous iteration. The extrinsic information is exchanged between MIMO demapper 305 and MAP decoder 309 to improve the bit error rate performance at each iteration. Methods for computing the extrinsic information in such inner/outer decoder settings are well-known in the art. In the first iteration, there is no extrinsic information input to the soft demapper. In subsequent iterations, in one embodiment, the extrinsic information is computed as follows. First, the soft-output is computed by the MAP outer decoder, and from it the input reliability information (input to the same outer decoder) is subtracted off in order to compute the extrinsic information produced by MAP decoder 309. This extrinsic

information is deinterleaved and passed as input to MIMO demapper 305 in the next iteration.

[0066] Returning to Figure 3, the estimates of bits in F parallel streams from MIMO demapper 305 together with reliability values for those bits are provided to parallel-to-serial converter 307 along with the extrinsic reliability information on each one of these bits. The reliability information is computed as the difference between the output reliability values for those bits (produced by demapper 305) and the input reliability values for those bits (inputs to demapper 305). The converter 307 reconstitutes the estimate of the BICM encoded bit stream generated by the transmitter, which was estimated by the receiver 300. The estimated BICM encoded bit stream (and the extrinsic reliability information) is then deinterleaved by bit deinterleaver 308 and applied to MAP decoder 309 to reverse the convolutional encoding applied by the transmitter. The reverse operation in this case corresponds to generating estimates of the bit values of the information bit stream that is the input to convolutional coder 201, and also producing extrinsic information that is to be passed back (after reinterleaving) to MIMO demapper 303 as new reliability information.

[0067] In one embodiment, the MAP decoding process generates soft output values for the transmitted information bits in a manner that is well known in the art.

[0068] The extrinsic information from MAP decoder 309 is first applied to bit interleaver 310. Bit interleaving aligns elements of the extrinsic information with the interleaved estimated BICM encoded bitstream from MIMO demapper 305. In addition, the interleaved extrinsic information is applied to serial-to-parallel converter 311, which forms N_t parallel streams of extrinsic information corresponding to the parallel bit streams formed at the transmitter.

The extrinsic information is exchanged between MIMO joint demapper 305 and MAP decoder 309 to improve the bit error rate performance at each iteration. In one embodiment, a MaxLogMAP-type approximation is used by the outer decoder to compute bit-LLR values for each bit location. In another embodiment, an improved Max-Log approximation for calculation of LLRs can be used in MAP decoder 309 associated with the convolutional code used as an outer encoder in the transmission

scheme. The improved Max-Log approximation for calculation of a posteriori LLR values may employ a max* term relationship as described above when calculating updated forward recursive, reverse recursive, and branch metrics sequences to calculate the LLR. Each constituent MIMO demapper 305 or MAP decoder 309 thus calculates the max* term by separate calculation of a max term (max(x,y)) and a logarithmic correction term ($\log(1+e^{-|x-y|})$).

An Example of an Inner Decoder Structure

[0069] After OFDM front-end preprocessing, the samples from each receive antenna and on each tone are passed through an inner/outer soft-in soft-out decoder structure for decoding shown in Figures 3 and 4, which are described above. Also as described above, in one embodiment, the outer decoder is an optimal (soft-in soft-out) MAP decoder. The complexity of the near-optimal receivers associated with these types of coded OFDM/BICM/OFDM systems resides in the inner decoder of the receiver structure in Figure 3. The received signal sample on the m th receive antenna and the n th tone can be expressed as

$$y_m[n] = \sum_k h_{km}[n] s_k[n] + w_m[n] \quad (2)$$

where $h_{mk}[n]$ denotes the effective channel gain between the k th transmit and the m th receive antenna on the n th tone, $w_m[n]$ denotes the associated thermal noise term on the m th antenna and n th tone. Alternatively, (2) can be compactly re-expressed as follows,

$$y[n] = H[n] s[n] + w[n] \quad (3)$$

where $h[n] = [h_1[n] \ h_2[n] \ \dots \ h_N[n]]^T$ with $h_m[n] = [h_{1m}[n] \ h_{2m}[n] \ \dots \ h_{Nm}[n]]^T$, and where $s[n] = [s_1[n] \ s_2[n] \ \dots \ s_N[n]]^T$, and $y[n]$ and $w[n]$ are similarly defined and where it is assumed that $N_t=N_r=N$.

[0070] In one embodiment, the channel state information (CSI) is not available at the transmitter, but CSI is fully available at the receiver; that is, the set of $H[n]$'s is known at the receiver but not at the transmitter.

[0071] On each OFDM tone, N QAM symbols are transmitted simultaneously and each of the N receive antennas receives a linear combination of these N symbols (whereby the linear combination is dictated by the instantaneous channel coefficients).

[0072] As stated above, in one embodiment, the receiver uses a version of the soft output M-algorithm (SOMA) that employs modified survivor selection criteria in the M-algorithm search. In one embodiment, the joint demapper uses the modified SOMA for finding the best alternative among an exponentially growing population of candidates by doing a reduced search in a detection tree. This is done by expanding only the M best alternatives from every level of the tree and one or more alternatives, rather than all alternatives. In one embodiment, the M best alternatives are determined using a metric and another criterion besides the metric. In one embodiment, the metric is so called MaxLogMAP type metric, such as described in Lin & Costello, "Error Control Coding, 2nd Edition," Prentice Hall, New York, 2003, which is well-known in the art.

[0073] Based on the search through the detection tree, the joint demapper calculates soft output values by comparing the estimated best path (the best visited path) with the longest opposite decision path visited (the best visited path with a different decision on the given bit from that of the best visited full-length path). These paths through the levels of the tree could be terminated at the end of the tree (there are M such paths) or non-terminated at every level (there are T early-terminated paths).

[0074] The soft output values from the inner SOMA-based MIMO joint demapper are then used by the soft in soft out (SISO) decoder for the outer binary code. This decoder in turn feeds soft values back to the inner decoder in an iterative turbo-type iterative decoding. In another embodiment, a soft-input hard-output Viterbi decoder (i.e., a simpler outer decoder) uses the soft output values from the inner SOMA-based MIMO joint demapper to produce hard output data for non-iterative decoding.

[0075] In one embodiment, the parameters in the SOMA-based inner decoder are channel-adaptive. Such channel-adaptive versions of SOMA inner

decoders save in complexity (with respect to the base SOMA designs) without appreciable reduction in performance, as well as being optimizable for a given channel realization to a desired target BER performance.

[0076] In one embodiment, the SOMA algorithm computes (estimated) symbol decision values and reliability information for the associated bit estimates by first turning the computation above into a computation on a tree and then performing approximate maximization computations by limiting the search through the tree.

[0077] Next the focus is on the SOMA operating on a fixed but arbitrary OFDM tone n . For convenience, the dependence of all variables vectors and matrices on the OFDM index, n , has been omitted. In one embodiment, the mapping of the MaxLogMAP demapper computations on a tree structure is based on exploiting the QR-type decompositions of the channel matrix is described. Let $\pi : \{1, \dots, N\} \rightarrow \{1, \dots, N\}$ denote a permutation function,

$s^{(\pi)} = [s_{\pi(1)} s_{\pi(2)} \dots s_{\pi(N)}]^T$ denote the associated N -symbol permutation of s , and

$P^{(\pi)}$ denote the associated permutation matrix, *i.e.*, the matrix yielding $s^{(\pi)} = P^{(\pi)} s$.

[0078] Associated with any fixed order π , the decomposition expresses the channel matrix H from equation (3) as $H^{(\pi)} = H [P^{(\pi)}]^T$ in the form

$H^{(\pi)} = Q^{(\pi)} L^{(\pi)}$ with $Q^{(\pi)}$ unitary and $L^{(\pi)}$ lower triangular. As a result, the

information lossless projection operation of y onto $[Q^{(\pi)}]^H$ yields a vector $\tilde{y}$ that constitutes a set of measurements that are equivalent to those in y from equation (3) and which can be represented as follows

$$\tilde{y} = L^{(\pi)} s^{(\pi)} + \tilde{w} . \quad (4)$$

whereby $L_{ij}^{(\pi)} = \{L^{(\pi)}\}_{i,j}$, and $L_{ij}^{(\pi)} = 0$ when $i > j$. See Figure 11 for an example, where the structure of the first term in the right hand side of equation (4) is illustrated for $N_t = N_r = N = 3$.

[0079] Given an equation for $\tilde{y}$ given above, the full-search MaxLogMAP can be readily implemented based on the above set of measurements via a search on

a tree. At depth k in the tree, only the k first equations are considered from equation (4) to rank candidates. As these equations depend only on the k first symbols in $s^{(\pi)}$, the sets of candidates are ranked in groups whereby each group corresponds to all the N -symbol candidates that have the same symbol values in the first k symbols in the order described by π . In particular, letting $\tilde{s}$, denote an arbitrary $N \times 1$ vector of 2^B QAM symbol values, $\tilde{s}_m = [\tilde{s}]_m$, and $\{\tilde{b}_m^1, \tilde{b}_m^2, \dots, \tilde{b}_m^B\}$ denote the associated values of the k th bits that map to $\tilde{s}_m$, the MaxLogMAP computation reduces to

$$L(\tilde{b}_m^k) = \max_{\tilde{s}: \tilde{b}_m^k=1} \Gamma(\tilde{s}, \hat{s}) - \max_{\tilde{s}: \tilde{b}_m^k=0} \Gamma(\tilde{s}, \hat{s}) \quad (5)$$

where $\hat{s} = [L^{(\pi)}]^{-1} y$ and

$$\Gamma(\tilde{s}, \hat{s}) = \sum_{m=1}^N \left\| l_{mm}^{(\pi)} (\tilde{s}_m - \hat{s}_m) + \sum_{n=1}^{m-1} l_{mn}^{(\pi)} (\tilde{s}_n - \hat{s}_n) \right\|^2 \quad (6)$$

[0080] The quantities $\Gamma(\tilde{s}, \hat{s})$ can be readily implemented recursively via a full tree-search on a tree of depth N and 2^B branches per node.

[0081] The conventional SOMA algorithm, in essence, obtains soft-output that approximates the MaxLogMAP-metric via an M-algorithm based search on the tree. Like any M -algorithm, from all surviving candidates at any given level, all possible candidates are expanded to the next level ($2^B M$ in this case), but only a subset M of those is kept for search at higher depths in the tree. An important element of the SOMA is that it recursively generates and updates quality metric estimates for each value of each of the NB bits represented on the tree. In particular, it exploits the use of two $N \times B$ matrices $\Delta^{(0)}$ and $\Delta^{(1)}$, whereby the relative reliability metrics associated with the values 0 and the 1 of the k th bit in s_m are given by $\delta_{m,k}^{(0)} = [\Delta^{(0)}]_{m,k}$ and $\delta_{m,k}^{(1)} = [\Delta^{(1)}]_{m,k}$, respectively. The scheme relies on recursively extending each surviving path at level m to its 2^B path extensions at the next level, computing the cumulative metrics for the new paths and sorting the paths in the order of decreasing metrics. If $p_{[l,i],r}$ denote the r th ranked path at depth l , then the

M top paths, i.e., the paths in the set $\{p_{[l,i],r}; 1 \leq r \leq M\}$ are retained, while the paths in $\{p_{[l,i],r}; r > M\}$ are terminated.

[0082] In contrast, in the modified SOMA, the survivors are selected based on additional criteria and not just the metric; that is, some of the best M paths are discarded and some other paths are kept in their place. This selection is based on soft-output type criteria as described above. Therefore, in the case of the modified SOMA, $p_{[l,i],r}$ denotes the r th survivor path at depth l , for r between 1 and M , and denotes the $(r-M)$ th best rejected path for r greater than M .

[0083] However, a subset of the best N_{term} terminated paths $\{p_{[l,i],r}; M+1 \leq r \leq M+N_{\text{term}}\}$ are still used before they are discarded for producing relative reliability updates for the bits and bit values they represent by updating the associated locations in $\left[\Delta^{(0)}\right]$ and $\left[\Delta^{(1)}\right]$ (note that T and N_{term} are used interchangeably to denote the number of terminated paths). After the completion at depth N , the SOMA first chooses the surviving length- N path with the best accumulated metric as the hard estimate. This $N \times 1$ vector of QAM symbol estimates is used to directly demap and obtain hard estimates for the NB bits $\{\tilde{b}_m^k; 1 \leq k \leq B, 1 \leq m \leq N\}$. Reliability metrics are updated in the two matrices based on all length N tree candidates $2 \leq r \leq M+N_{\text{term}}$. Then the relative reliability information for the k th bit represented in the m th QAM symbol is given by

$$L(b_m^k) = \left[2\tilde{b}_m^k - 1 \right] \delta_{m,k}^{(1-\tilde{b}_m^k)} \quad (7)$$

The values of M (surviving candidates per depth) and N_{term} (the number of candidates used for gathering soft information based on early terminated paths) can be varied to trade off computation complexity with bit-error-rate performance. In the iterative decoding setting, in each iteration cycle, each decoder computes extrinsic information that is passed as input (appropriately deinterleaved in the case of the MIMO demapper, and reinterleaved in the case of the outer MAP decoder) to the other decoder. The extrinsic information is computed as the difference between

the soft output information produced by the decoder (e.g., in the case of MIMO demapper see equation (7)), and the input intrinsic information to the decoder. Typically, the extrinsic information passed between decoders for any given particular bit location is in the form of differential values, that is the difference between the “bit=1” value and the “bit=0” reliability value. In one embodiment, if iterative decoding is used, the metric used for SOMA decoding shown in equation (6) is modified to include an extrinsic term. In particular, another term is added to the right hand side of equation (6), which is a sum of terms, one term for each bit location in the binary representation of the symbol $\tilde{s}$. When differential reliability values are employed, the term added that corresponds to any given, but fixed, bit location, equals zero if the bit-value of that bit location in $\tilde{s}$ is 0, and equal to the differential input reliability value otherwise.

[0084] The MIMO demapper receives y_1 and y_2 signals and returns estimates of the bits represented by the symbols s_1 and s_2 , for one subtone and reliability (soft) information on each bit representing each of the symbols s_1 and s_2 . That is, soft-output (reliability information) is provided on the set of estimated bits.

[0085] After the QR-decomposition, two scalar measurements are obtained in the form described in equation (4). For illustration purposes, consider the permutation order π corresponding to the order s_1, s_2 . Due to the structure of the L^π matrix in (4), the first measurement in $\tilde{y}$ only depends on s_1 while the second depends on both s_1 and s_2 . Next, the metric in equation (6) is considered, which in this case is a sum of two terms. The first term ($m=1$) is the term due to the first measurement in $\tilde{y}$ and only depends on s_1 . The second term ($m=N=2$) is term due to the second measurement (and consists of an l_{12} and an l_{22} term). This structure allows the computation of each of the metrics in (6) to be performed on a tree. At the first level of the tree only the first terms ($m=1$) in the sums in (6) are computed. Since these depend only on s_1 the number of terms computed (and thus number of level-1 nodes in the tree) equals the number of possible values s_1 can take. In the second step, from each node at level one (each corresponding to a distinct value of s_1), leafs for each possible value of s_2 are extended, and the second term (branch metric) in the sum in equation (6) is computed and added to 1st term corresponding

to the particular value of s_1 . In the end (level 2 in this case), there are as many end nodes as there are candidate vectors of symbols, and each node represents a computation of (6) for a specific vector symbol candidate. Those can thus be compared as in (5) to provide bit estimates and reliability information for all the bits represented by the QAM symbol vector.

[0086] Figure 6 illustrates the decision tree that allows this recursive computation of metrics on a tree in the case that there are three transmit antennas and thus $N_t = N_r = N=3$. The SOMA algorithm does not search the full tree, but rather a limited set of paths. The way the paths are limited is to start expanding paths from the root of the tree and at each level keeping only a subset of the paths as surviving paths (i.e., as paths that will be further extended). Referring to Figure 6, at each new level of the tree, a decision is made to expand the tree for only the M best branches and a subset of other branches as described in more detail above. This decision can be based, at least in part, by calculating the partial distance metric for each candidate. Specifically, at level “n”, the distance metric used for comparisons corresponds to the first “n” terms of equation (6) (i.e., the sum of all terms in equation (6) for $m=1, 2, \dots, n$). Then based on this metric the best M metrics are selected as survivors. Thus, the tree is pruned at each depth by keeping the best M paths, as well as other additional candidates. This is represented in Figure 7.

[0087] More specifically with respect to searching the tree, the SOMA operates its search on a tree that is associated with a specific symbol order. The full tree without pruning represents all possible combinations of sequences of QAM symbols where the length of the sequences (or the depth of the tree) is equal to the number of transmit antennas. Let Q denote the number of possible values of a single QAM symbol ($Q=2^B$). At depth one in the tree, there are Q candidates each corresponding to the Q values associated with “so-called” ordered 1st symbol. At the next depth, Q candidates are expanded from each of the candidates of level 1, each candidate corresponding to the Q values associated with “so-called” ordered 2nd QAM symbol, and so on, until level N, in which case all symbol values for all N transmitted QAM symbols (on the given OFDM tone) have been represented. For example, for a 6x6 MIMO antenna system with 64QAM, the length of the

sequences is 6 QAM symbols. The SOMA prunes the tree in decoding and only searches the most promising part of the tree according to a metric at each level.

[0088] When applying SOMA on a specific OFDM tone, there are N measurements (as many as there are receive and transmit antennas) and N QAM symbols that need to be resolved (these are the symbols transmitted on this OFDM tone over the N transmit antennas). In one embodiment, in order to apply the SOMA on a tree representing a specific symbol order, a preprocessing technique is applied on the N measurements, which is based on the well-known QR decomposition (performed by a QR decomposition front end of the joint demapper). The N -dimensional column vector representing all N measurements is projected to a unitary transformation. Specifically, the vector is left-multiplied by a matrix A that has dimensions N rows by N columns, and satisfies $AA^H = I$, where I is the identity matrix and the superscript “ H ” represents the transposition and element-wise conjugation operation. Any such unitary transformation of the measurements is invertible and therefore information preserving. As a result, the new processed N measurements, i.e., the result of the left multiplication of the original measurements, are information preserving. When the unitary matrix A is chosen based on the QR decomposition of the channel matrix, the resulting processed measurements have the following property: the first measurement only depends on the 1st ordered symbol; the 2nd measurement only depends on the 1st and the 2nd ordered symbols; the k th measurement depends only the first k ordered symbols. These processed measurements are then used to implement the SOMA algorithm on a tree with the specific symbol order. Note that to operate the SOMA on a tree formed by a different symbol order, a different unitary matrix A should be used. The new unitary transformation would then generate a different set of processed measurements, so that the k th measurement only depends on the k first symbols in the new symbol order.

[0089] In one embodiment, the decoding tree is organized based on reordering the symbols according to their received energy levels, or signal to noise ratio levels. In one embodiment, these received signal or SNR levels are estimated from the channel state information at the receiver, and potentially taking into

account extrinsic information provided by the outer decoder. To compute the received symbol energy associated with the k th symbol (i.e., the QAM symbol transmitted by the k th transmit antenna), the receiver computes the total received channel-coefficient energy from the k th transmit antenna to all receive antennas. The symbol corresponding to the highest received energy level is placed at the root of the tree and is processed first in the decoding algorithm. The second symbol in the tree is ordered such that it has the second highest received energy level. The remaining symbols are selected according to decreasing received energy level values. This ranking yields better detection efficiency. Then, following the required QR decomposition, the modified SOMA in its simplest form can be operated with a fixed value of M (or M' and M''), as well as fixed T and I parameters. In another embodiment, the signal energies of the symbols that are relevant during the tree search are instead used for ordering. First, all symbols are tested as the first symbols in the tree order and the symbol with the highest signal energy on the first-level tree is chosen. Then all remaining symbols are considered as the second symbol in the order, and the symbol with the highest signal level on the second level on the tree is chosen as the second symbol in the order. The process continues until all symbols have been ordered. In one embodiment, the signal energy on the tree of each of the symbols tested can be computed as follows. Assuming a symbol is considered as the " m "th ordered symbol at level m on the tree (after symbols 1, 2, ..., $m-1$ in the order have already been selected). An order π is chosen, whereby the first m symbols are the given ordered symbols and where an arbitrary order is placed on the remaining symbols. Then the QR decomposition for that symbol order is performed and an equation of the form (4) is obtained. Then the magnitude of the " m "-th entry on the main diagonal of the matrix L^π in Equation (4) denotes the signal-level on the " m "-th symbol in the tree. Note that only the magnitude of the " m "-th entry on the main diagonal is needed, and the value of that entry can be computed in computationally efficient manner, without performing the full QR decomposition. Specifically, consider selecting the " m "-th ordered symbol. In the described embodiment, the first $m-1$ symbols in the order have already been selected and all remaining symbols are compared as the " m "-th symbol in the order.

For each of these symbols, a QR decomposition is made for a (symbol-specific) order π , according to which the first $m-1$ symbols are the already chosen symbols, the “ m ”-th symbol is the symbol in comparison, and the rest of the order is arbitrary. Consider a fixed but arbitrary row among the first $m-1$ rows of the matrix A in the QR decomposition. Any such row of the matrix A is the same for each of these QR decompositions that test symbols as the m -th ordered symbols. Also, only the “ m ”-th row of the A matrix associated with any such order π needs to be determined, in order to obtain the required SNR information, i.e., in order to obtain the “ m ”-th element along the main diagonal of the matrix L^π in Eqn. (4). This can be computed quite efficiently in a manner well known in the art.

[0090] Note that, as described above, the symbol ordering for setting up the SOMA search tree is conducted on a per-tone basis, i.e., the ordering is performed independently for every tone in the OFDM system. In principle, however, this is not required. In another embodiment, other block-reordering schemes are used in which the same tree-search order is chosen for a block of tones by comparisons of the received symbol levels over the given block of tones.

[0091] The tree-search symbol reordering technique has a number of advantages. One advantage is that it provides a method for a high performing inner joint demapper with soft output, with an overall complexity that makes it implementable in an iterative decoding setting. Furthermore, tree ordering requires minimal additional complexity, and can yield improved performance for the same complexity, or allow achieving the same bit-error-rate performance with lower receiver complexity.

[0092] The SOMA detection process also calculates the soft (reliability) information on each bit, besides choosing the best path as described herein.

[0093] Figure 8 is a flow diagram of one embodiment of a process for setting up the SOMA inner decoding operation on tone f . The process is performed by processing logic that may comprise hardware (e.g., circuitry, dedicated logic, etc.), software (such as is run on a general purpose computer system or a dedicated machine), or a combination of both. The process is performed on each tone.

[0094] Referring to Figure 8, first channel measurements are used (based on pilot signals) to estimate the channels between all transmit-receive antenna pairs on any given tone f (this would correspond to estimating the channel matrix $H[f]$) (801). The channel estimation and SNR computation are based on pilot measurements on OFDM tone f (831). The channel estimates and lookup table (LUT) 805 are used to set up the adaptivity (e.g., changing the tree-search symbol order, changing M' , M'' , T of the I, etc.) of SOMA.

[0095] More specifically, these measurements are used to set up the QR-decomposition of the channel matrix, set up the SOMA detection tree (803) using tree-search symbol order selector/controller 850 (optional), and select (e.g., by use of a lookup table) the parameters of the SOMA algorithm (804). As the flow diagram reveals, the selection of these parameters depends on the channel conditions. Then, once the QR-decomposition (802), detection tree (803), and SOMA parameters have been set (804), the measured data on all receive antennas on tone f (832) are processed through a QR-decomposition (802) to generate a set of effective channel measurements, the tree is constructed (803) and then the SOMA inner detection algorithm is implemented (804).

[0096] In one embodiment, the receiver performs a modified soft output M-algorithm (SOMA) that is used with or without a forward-backward tree search. The base SOMA operates only in the forward direction, where at each level the locally best path is recursively estimated and then the bit reliability values are recursively estimated based on the locally best path and taking into account the terminated paths at that level of the tree. As a result, the reliability information for each bit is not necessarily gathered with respect to the full-length path and the associated longest early-terminated path. In one embodiment, the SOMA process is the FB-SOMA, which is a two-pass version of the soft-output M-algorithm that has a forward pass followed by a backward pass. One embodiment of the FB-SOMA is described in U.S. patent application serial no. 12/335,409, entitled "Adaptive Forward-Backward Soft Output M-Algorithm Receiver Structures", filed December 15, 2008.

[0097] The outputs of the SOMA inner detection algorithm are bit estimates, bit reliability information and bit extrinsic information 834. The hard output is generated thereafter, in a manner well known in the art, based on the best path.

[0098] The SOMA may be implemented on OFDM tone f using SOMA parameters on tone f provided by LUT 805. The output of LUT 805 may provide one or more variable values of the SOMA parameters on tone f , namely M , T (832), and the number of inner-outer soft-output decoder iterations, namely I (833). That is, LUT 805 may specify the value of M (when M is adaptive) while the values of T and I are unchanged (non-adaptable), or the value of T (where T is adaptable) while the value of M and I are unchanged (non-adaptable). The same could occur for 2 or more of the values of M , T and I . These values may be changed at different depths/levels of the tree, such that adaptation occurs over different levels. In such a case, adaptation occurs based on tone quality and on depth. In one embodiment, other variations may be used that include adaptive use of M' and M'' over different tones. In an alternative embodiment, a LUT is not used and the values are changed in the SOMA algorithm itself. In such a case, in one embodiment, the values are thresholded in the algorithm. For example, if the channel estimate for the tone falls within a first range, a certain value of M is used (e.g., $M=8$), but if the channel estimate of the tone falls in another range, then a different value of M is used (e.g., $M=12$). In one embodiment, involving modified SOMAs, for example M' and M'' , or M_e , and even J (the number of terms used in the improved MaxLogMAP soft output approximation) are adapted. These changes in value may also be based on the number of transmit antennas, the rate of the outer binary code, etc. Note that certain values may change based on whether the group of the paths with the best metrics have metrics that are clustered together, such that those in the cluster (more or less than M) or having values within a certain percentage of the worst metric in the cluster (e.g., with 95% of the value of the lowest quality metric in the cluster) are permitted to continue to the next level/depth in the tree. The resulting set of survivors may have cardinality more or less than M . Alternatively, the process may keep as survivor paths, all the paths whose relative metrics are within a certain percentage (e.g., 95%) of the metric of the best path.

[0099] Tree search order selector controller 850 and LUT 805 exchange information. For example, tree search order selector controller 850 selects an order but it also specifies a decomposition of the form of equation (4) that is used to set up the tree. The SNR values at level m on the tree are given by the m -th row of the L matrix in equation (4). These SNR values can be also used to choose via the LUT 805 values for e.g., the M parameter at each level of the tree.

Alternative Embodiments

[00100] There are a number of alternative embodiments. In one embodiment, the survivor-selection modified SOMA and FB-SOMA algorithms are implemented based on approximate metric values (at least for some of the paths) for calculating and sorting using e.g., the method described in K. Higuchi, et al., "Adaptive selection of surviving symbol replica candidates based on maximum reliability in QRM-MLD for OFCDM MIMO multiplexing", in Proc. Globecom, pp 2480-2486, Dec. 2004. This yields further savings in computational complexity.

[00101] In yet another embodiment, the proposed modified versions of the SOMA and the FB-SOMA are combined with the hierarchical tree based SOMA, the H-SOMA, described in Kitty Wong, "The Soft Output M-algorithm and its applications," PhD Thesis, Queens University, Kingston, Canada, August 2006. Described therein is the use of the hierarchical tree can yield significant savings for the base SOMA, in particular, for large QAM constellations, like 64 QAM. The same relative savings are also obtained by the FB-SOMA on the hierarchical tree.

[00102] In U.S. serial no. 12/121,649, entitled "Adaptive Soft Output M-Algorithm Receiver Structures for MIMO/OFDM/QAM Systems with BICM/ID", filed May 15, 2008, alternative methods for adaptive SOMA receivers are described, e.g., SOMA algorithms that have variable M in the tree and adapt the M , I , T parameters to the quality of the OFDM tones. Embodiments of the invention can be used independently or optionally be combined with those adaptation methods.

[00103] It should also be noted that, in the described form, the symbol ordering for setting up the modified searches on the SOMA and FB-SOMA search tree is conducted on a per-tone basis, i.e., the ordering is performed independently

for every tone in the OFDM system. In principle, however, the approach can be generalized to include block-reordering schemes, whereby the same tree-search order is chosen for a block of tones by comparisons of the received symbol levels over the given block of tones, such as described in U.S. serial no. 61/031,077, entitled "Tree Position Adaptive Soft Output M-Algorithm Receiver Structures for MIMO/OFDM/QAM Systems with BICM/ID", filed February 25, 2008.

[00104] In yet another embodiment, a metric correction term is applied for the soft output algorithm, much the same as the one used in the corrected SOVA algorithm, such as described in, for example, S. Lin and D.J. Costello Jr., "Error Control Coding, 2nd Edition", Prentice Hall, New York, 2003.

[00105] It should also be pointed out that embodiments of this invention can be combined with the hierarchical tree based on hierarchical QAM constellations as described in Kitty Wong, "The Soft Output M-algorithm and its applications," PhD Thesis, Queens University, Kingston, Canada, August 2006.

Advantages of Embodiments of the Invention

[00106] Embodiments of this invention have a number of attractive attributes. One advantage of at least one embodiment is that it provides for a high performing inner joint demapper with soft output, and an overall complexity that makes it implementable in an iterative decoding setting. For example, in one embodiment, survivor selection is modified to allow inclusion of paths in the survivor list, which, although potentially inferior in terms of their path metric value when compared to some of the excluded paths, they allow the collection of more reliable soft-output information for some bits in the information-bearing signal. The calculations are very efficient, especially when approximate metric and sorting calculations are used.

[00107] Furthermore, embodiments of the invention can be combined with tree ordering that requires minimal additional complexity, and can yield improved performance for the same complexity, or allow achieving the same bit-error-rate performance with lower receiver complexity. Hierarchical FB-SOMAs can also be used with constructions similar to the hierarchical SOMAs, when mappers with the multilevel bit property are employed.

[00108] Other advantages include the ability to obtain soft-output based on improved MaxLogMAP metrics and reduced size tree-searches.

[00109] Whereas many alterations and modifications of the present invention will no doubt become apparent to a person of ordinary skill in the art after having read the foregoing description, it is to be understood that any particular embodiment shown and described by way of illustration is in no way intended to be considered limiting. Therefore, references to details of various embodiments are not intended to limit the scope of the claims which in themselves recite only those features regarded as essential to the invention.

CLAIMS

We claim:

1. A receiver for use in a communication system to receive information-bearing signals transmitted from the transmitter, the receiver comprising:

an inner decoder structure having a multiple-in multiple-out (MIMO) joint demapper to perform joint inner demapping over each tone, the joint demapper being operable to apply a soft-output M-algorithm to identify survivor candidates at each depth in a detection tree being searched for each tone, including surviving full-length candidates, based on at least one metric and at least one other criterion, where a number of best alternatives from every level of the tree are expanded along with one or more alternatives selected meeting the at least one other criterion and where soft-output related information is collected and stored for each bit, and

an outer decoder operable with the inner decoder to perform iterative decoding.

2. A method comprising:

performing a first decoding operation to produce a first set of output data representing most likely transmitted bit estimation values and information about the reliability of each of these estimates, including

performing a detection process over each tone for joint inner demapping, by applying a soft-output M-algorithm to identify survivor candidates at each depth in a detection tree being searched for each tone, including surviving full-length candidates, based on at least one metric and at least one other criterion, where a number of best alternatives from every level of the tree are expanded along with one or more alternatives selected meeting the at least one other criterion and where soft-output related information is collected and stored for each bit; and

calculating a soft output value for each bit by comparing an estimated best path with the longest and best visited path with an opposite decision on the bit.

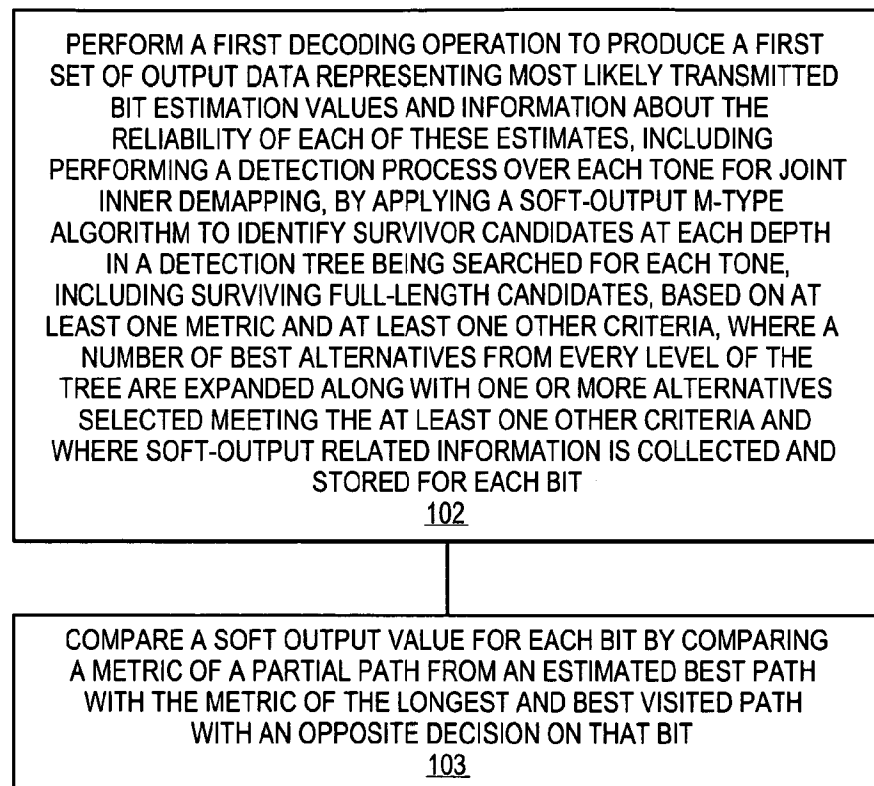
3. An article of manufacture having one or more computer readable storage media storing instructions thereon which, when executed by a system, cause the system to perform a method comprising:

- performing a first decoding operation to produce a first set of output data representing most likely transmitted bit estimation values and information about the reliability of each of these estimates, including

- performing a detection process over each tone for joint inner demapping, by applying a soft-output M-algorithm to identify survivor candidates at each depth in a detection tree being searched for each tone, including surviving full-length candidates, based on at least one metric and at least one other criterion, where a number of best alternatives from every level of the tree are expanded along with one or more alternatives selected meeting the at least one other criterion and where soft-output related information is collected and stored for each bit; and

- calculating a soft output value for each bit by comparing an estimated best path with the longest and best visited path with an opposite decision on the bit.

1/12

**FIG. 1**

2/12

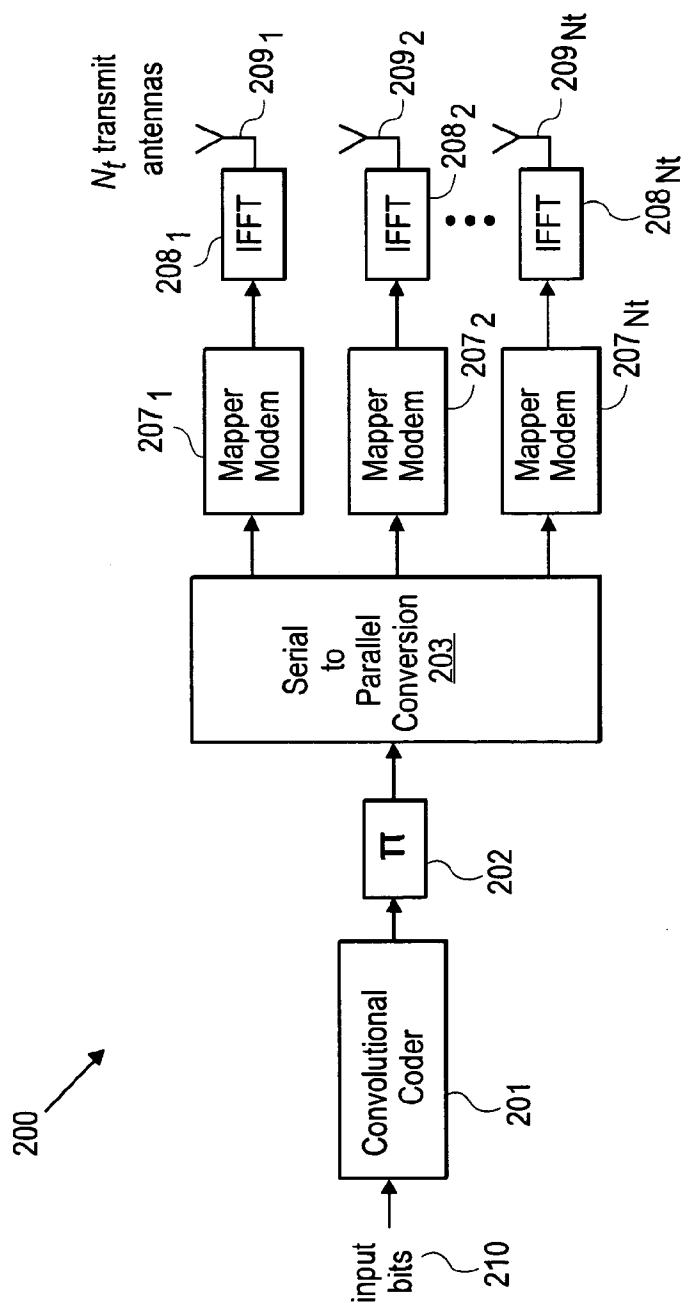


FIG. 2

3/12

300

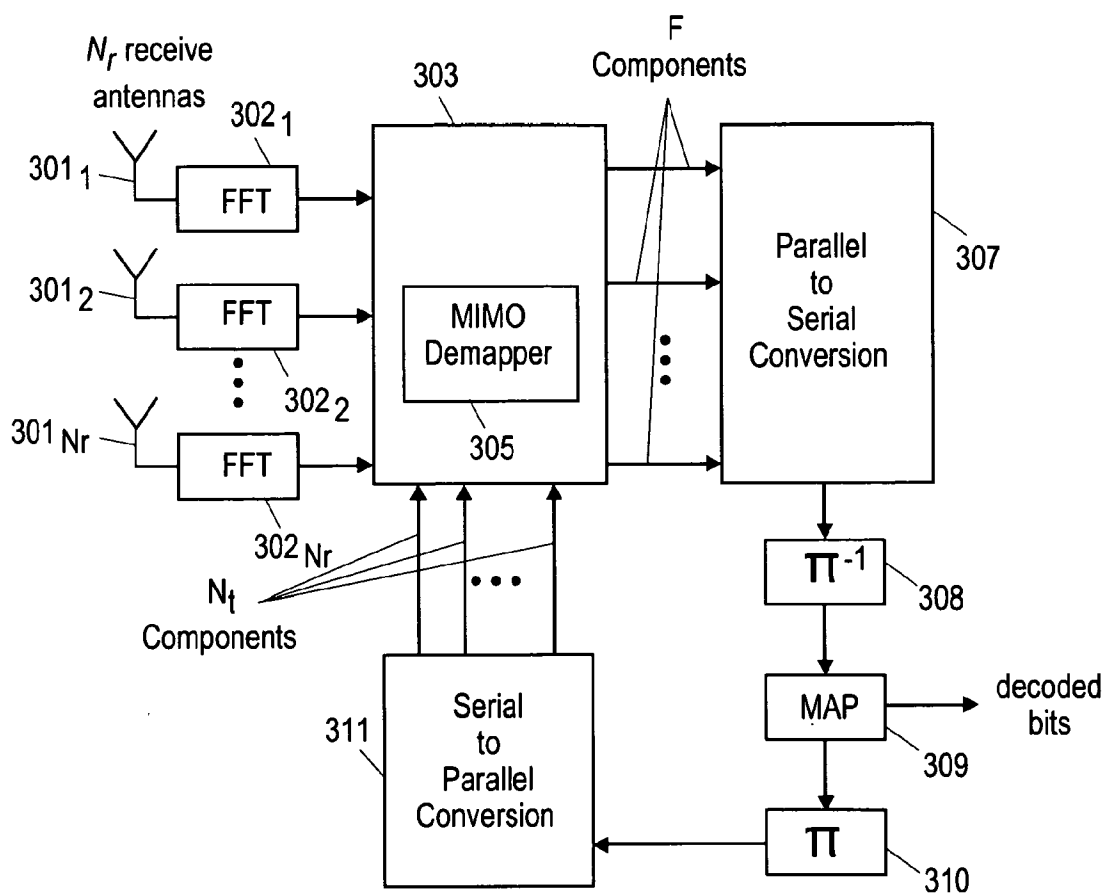
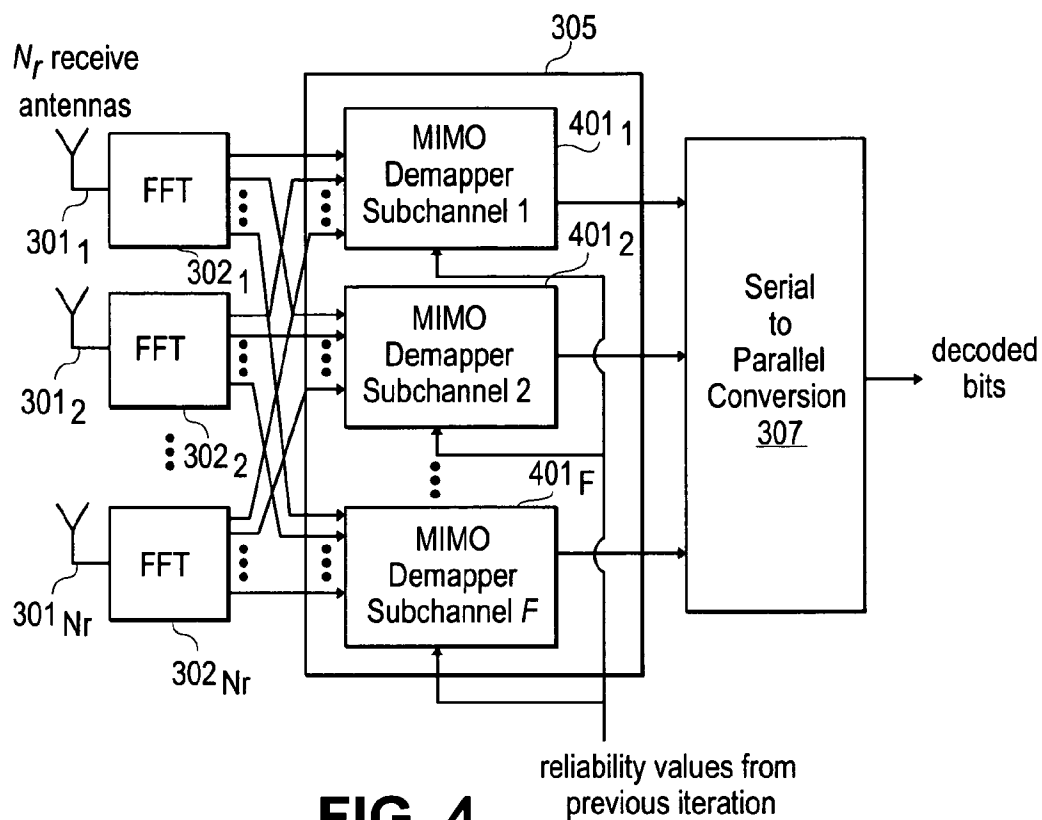


FIG. 3

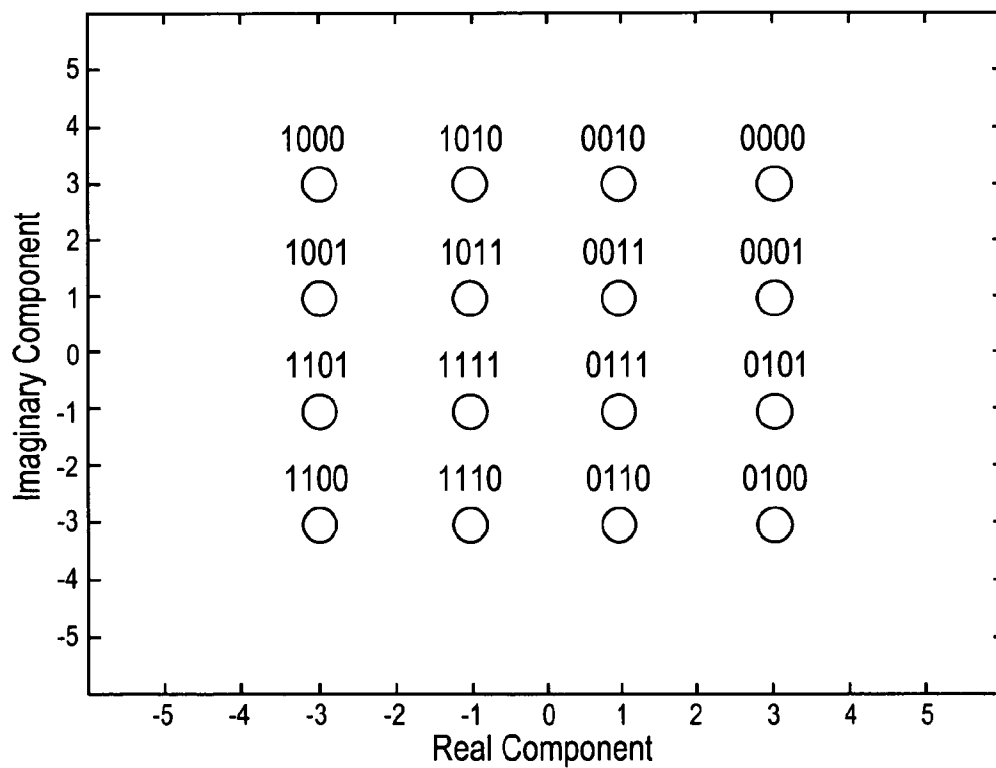
4/12

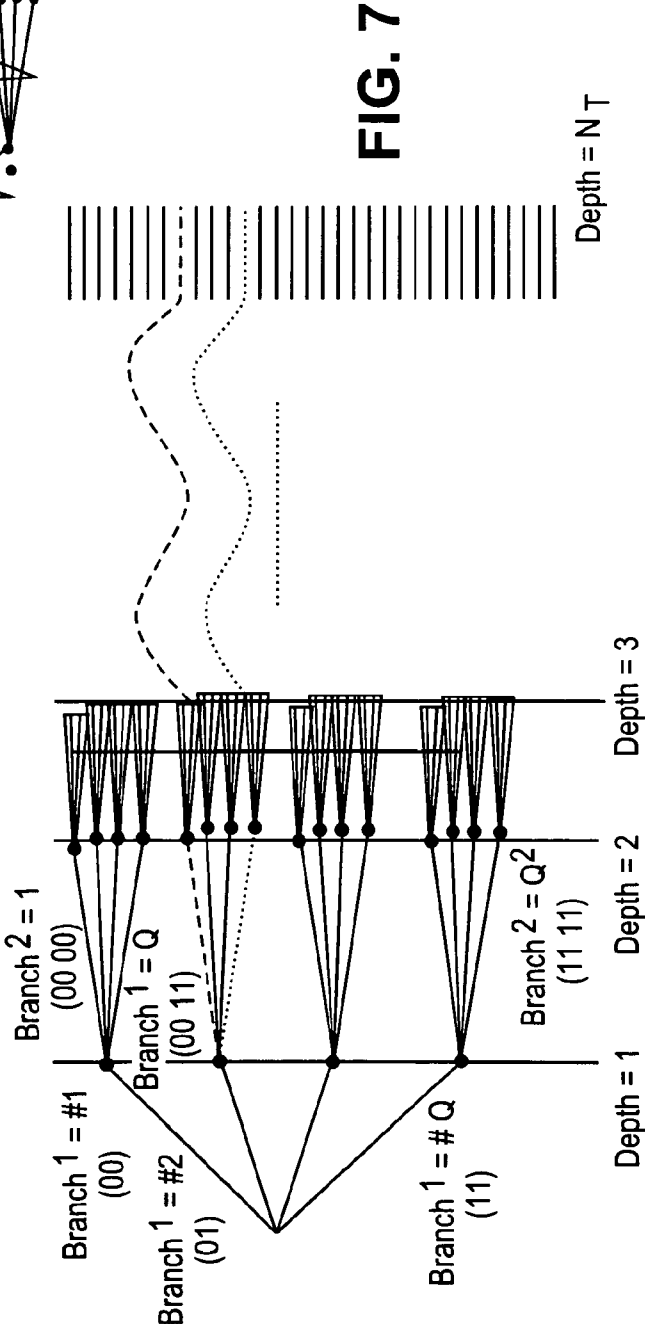
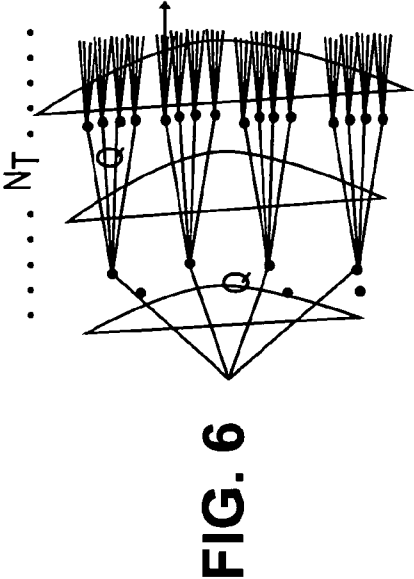
**FIG. 4**

1000	1011	1100	1111
○	○	○	○
0001	0010	0101	0110
○	○	○	○
0100	0111	0000	0011
○	○	○	○
1101	1110	1001	1010
○	○	○	○

FIG. 5A

5/12

**FIG. 5B**



7/12

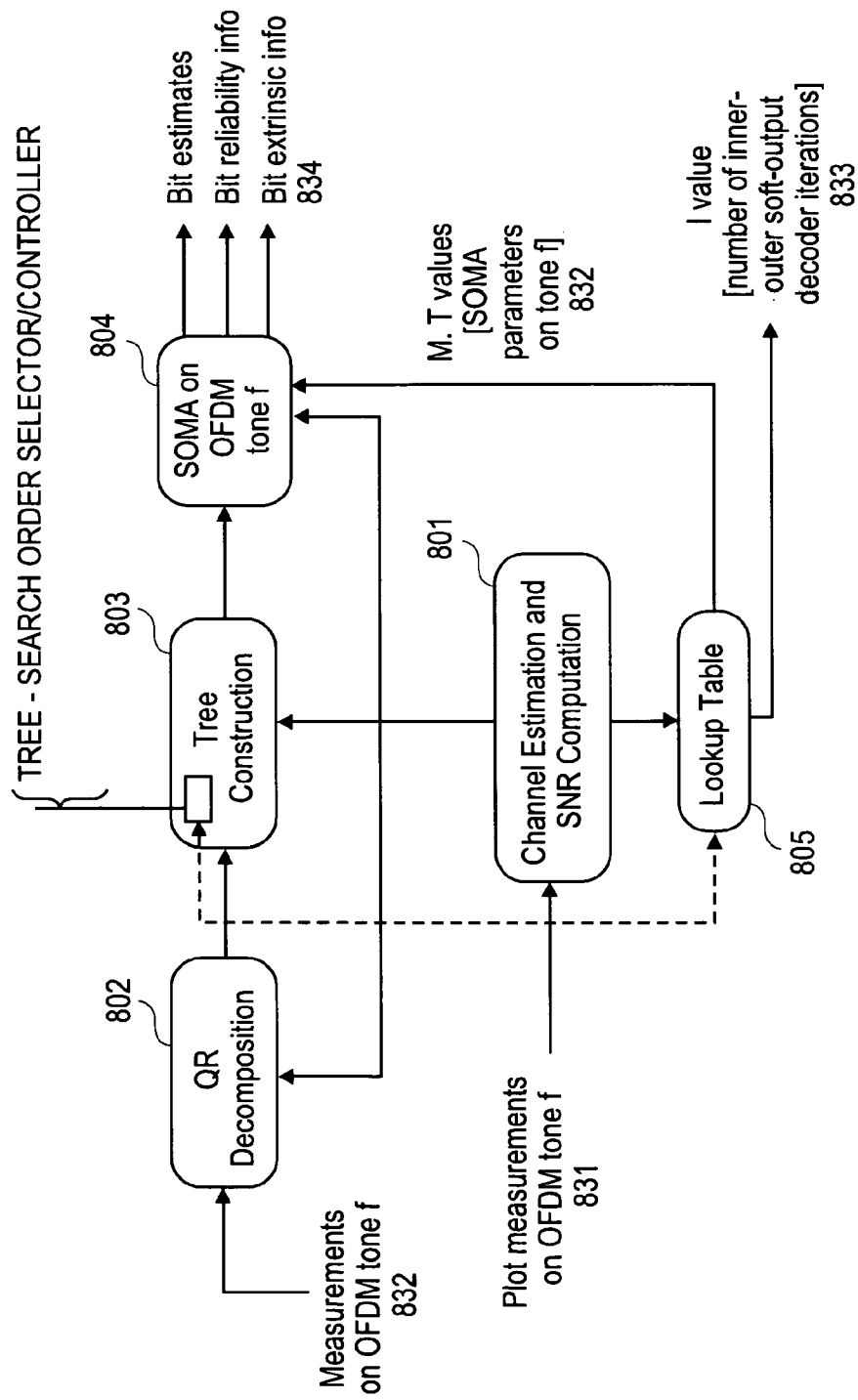


FIG. 8

8/12

$$\begin{bmatrix} \Lambda \\ S_1 \\ \Lambda \\ S_2 \\ \Lambda \\ S_3 \end{bmatrix} = \begin{bmatrix} \text{901} & & & \\ L_{11} & 0 & 0 & \text{904} \\ L_{21} & L_{22} & 0 & S_2 \\ L_{31} & L_{32} & L_{33} & S_3 \end{bmatrix}$$

The diagram illustrates a matrix equation. On the left, a column vector is shown with elements Λ , S_1 , Λ , S_2 , Λ , and S_3 enclosed in brackets. This is followed by an equals sign and a 4x4 block matrix. The block matrix is composed of four sub-matrices, each enclosed in brackets and labeled with a reference numeral: 901 (top-left), 902 (bottom-left), 903 (bottom-middle), and 904 (top-right). The sub-matrix 901 contains the element L_{11} . The sub-matrix 902 contains the elements L_{21} , L_{31} , and L_{32} . The sub-matrix 903 contains the elements L_{22} , L_{32} , and L_{33} . The sub-matrix 904 contains the elements S_1 , S_2 , and S_3 . The matrix is partitioned into four quadrants by a horizontal line between the first and second rows, and a vertical line between the third and fourth columns. The top-right quadrant (between the first and second rows, and between the third and fourth columns) contains the element 0. The bottom-right quadrant (between the third and fourth rows, and between the third and fourth columns) contains the element 0. The bottom-left quadrant (between the third and fourth rows, and between the first and second columns) contains the element 0. The top-left quadrant (between the first and second rows, and between the first and second columns) contains the element L_{11} . The top-middle quadrant (between the first and second rows, and between the second and third columns) contains the element 0. The top-right quadrant (between the first and second rows, and between the third and fourth columns) contains the element 0. The bottom-middle quadrant (between the third and fourth rows, and between the second and third columns) contains the element 0. The bottom-right quadrant (between the third and fourth rows, and between the third and fourth columns) contains the element 0.

FIG. 9

9/12

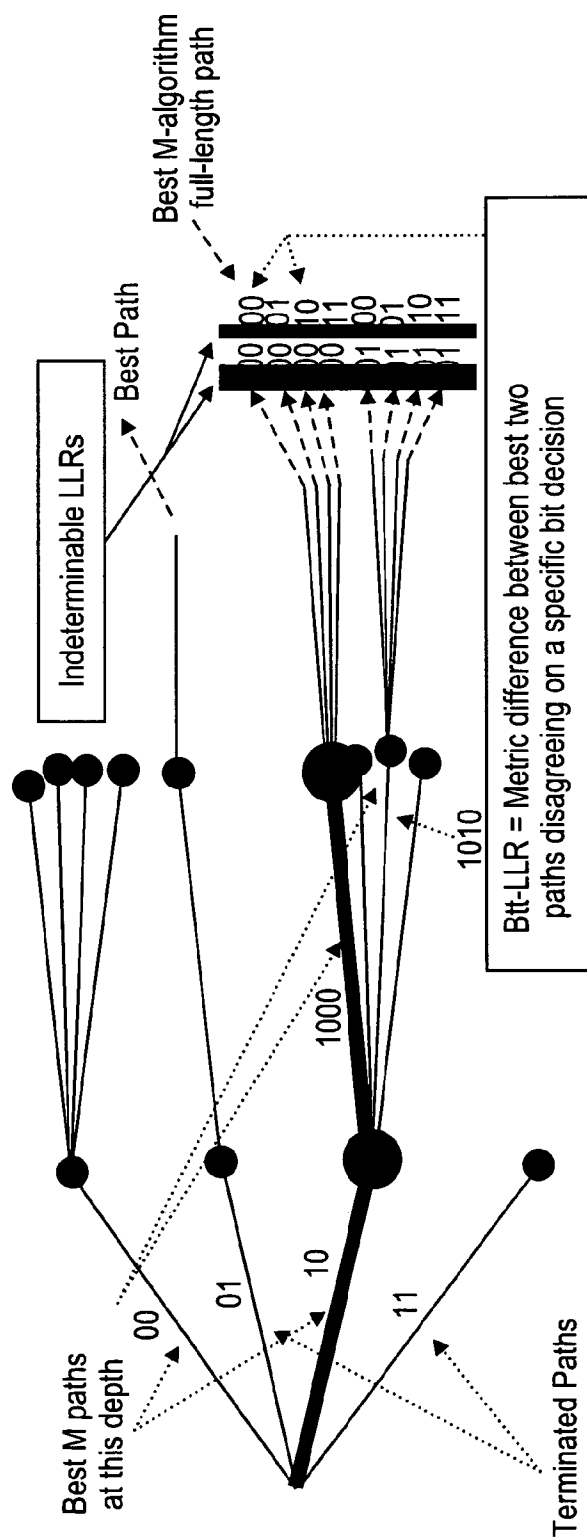
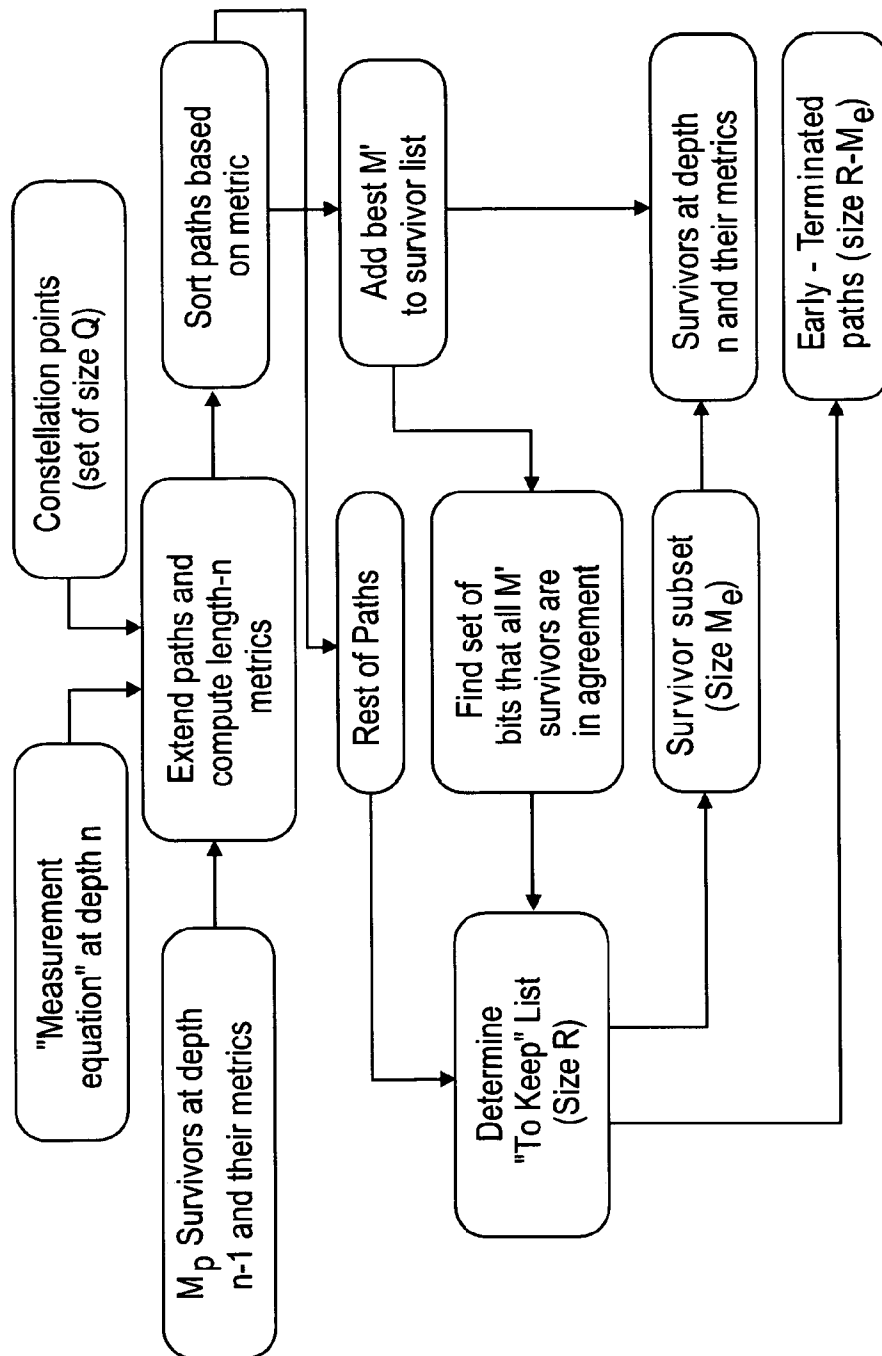


FIG. 10

10/12

**FIG. 11**

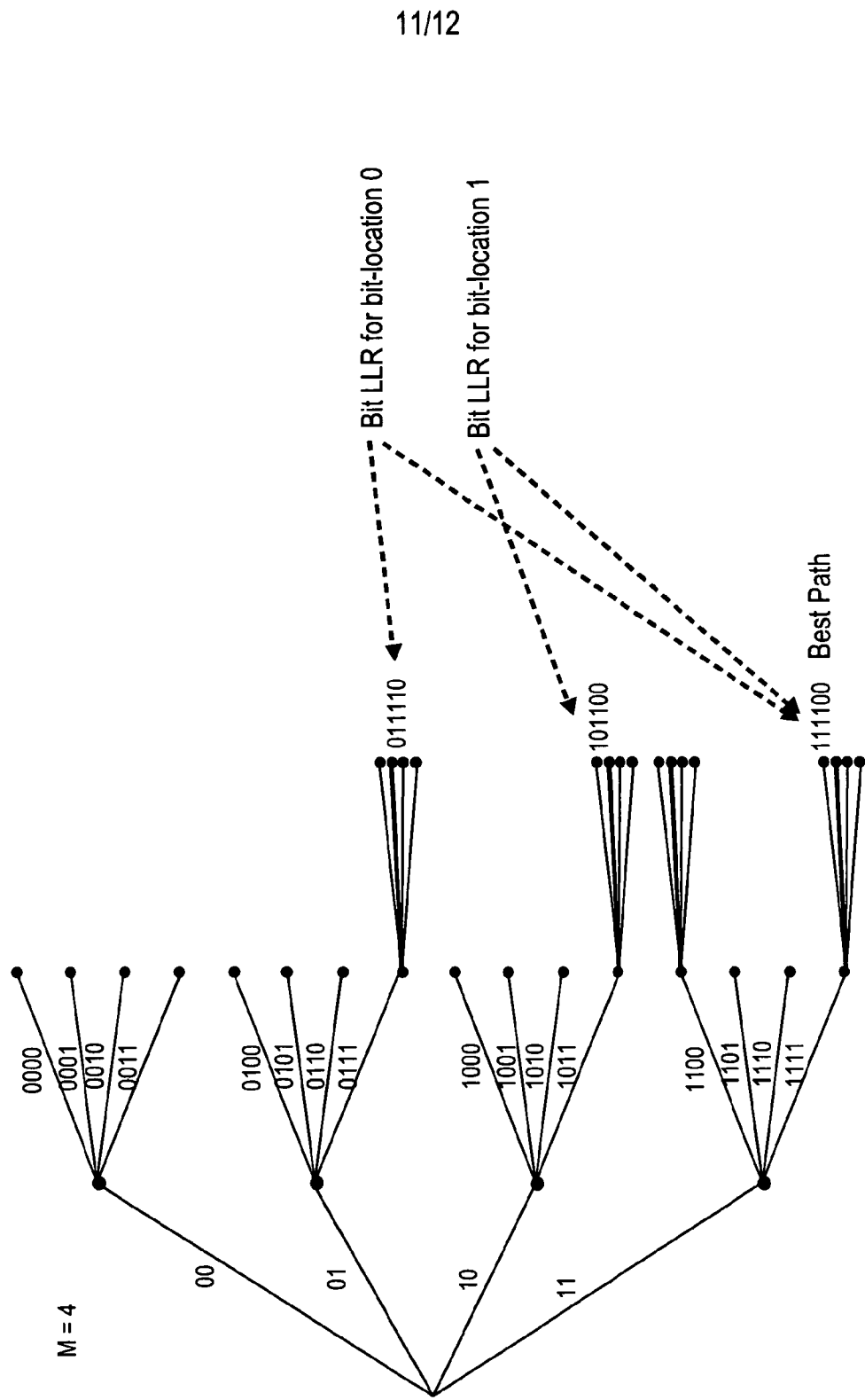


FIG. 12

12/12

0-value Table				1-value Table				Ordered Paths	Survivor?
								-	-
				X	X	X	X	1111	Yes
			X					1110	Yes
		X						1101	Yes
								1100	No
	X							1011	Yes
								1010	No
								1001	No
								1000	No
X								0111	Yes
								0110	No

FIG. 13

0-value Table				1-value Table				Ordered Paths	Survivor?
0	0	0	0	0	0	0	0	-	-
0	0	0	0	1	1	1	1	1111	Yes
0	0	0	1	2	2	2	1	1110	Yes
0	0	1	1	3	3	2	2	1101	Yes
0	0	2	2	4	4	2	2	1100	Yes
0	1	2	2	5	4	3	3	1011	Yes
0	2	2	3	6	4	4	3	1010	Yes
0	2	2	3	6	4	4	3	1001	No
0	2	2	3	6	4	4	3	1000	No
1	2	2	3	6	5	5	4	0111	Yes
2	2	2	4	6	6	6	4	0110	Yes

FIG. 14