

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
28 December 2006 (28.12.2006)

PCT

(10) International Publication Number
WO 2006/138135 A2

- (51) International Patent Classification:
H04J 1/00 (2006.01) H04L 1/02 (2006.01)
- (21) International Application Number:
PCT/US2006/022228
- (22) International Filing Date: 7 June 2006 (07.06.2006)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:
60/691,756 16 June 2005 (16.06.2005) US
11/158,586 21 June 2005 (21.06.2005) US
- (71) Applicant (for all designated States except US): QUALCOMM INCORPORATED [US/US]; 5775 Morehouse Drive, San Diego, California 92121 (US).
- (72) Inventors; and
- (75) Inventors/Applicants (for US only): HOWARD, Steven J. [US/US]; 75 Heritage Avenue, Ashland, Massachusetts 01721 (US). KETCHUM, John W. [US/US]; 37 Candleberry Lane, Harvard, Massachusetts 01451 (US).

WALLACE, Mark S. [US/US]; 4 Madel Lane, Bedford, Massachusetts 01730 (US). MONSEN, Peter [US/US]; 15 Oakcroft Road, Wellesley, Massachusetts 02482 (US). WALTON, Jay Rodney [US/US]; 85 Highwoods Lane, Carlisle, Massachusetts 01741 (US).

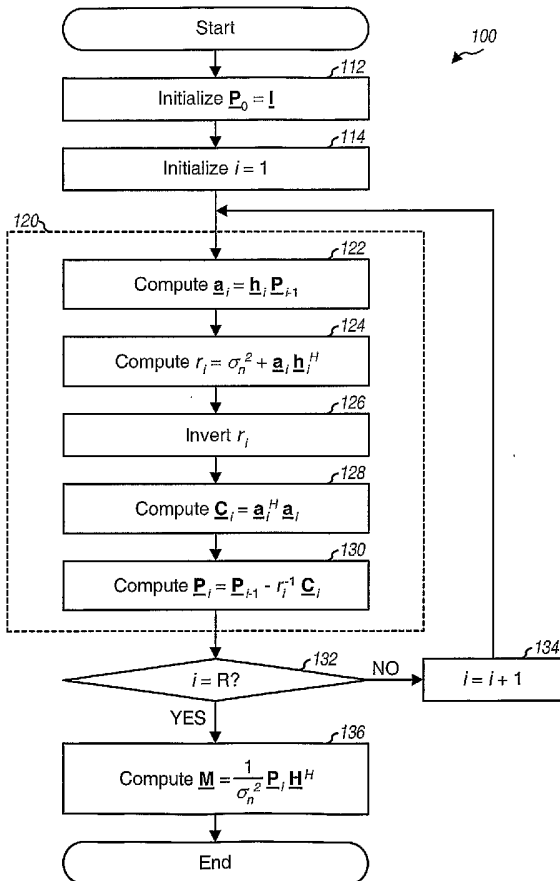
(74) Agents: WADSWORTH, Philip R. et al.; 5775 Morehouse Drive, San Diego, California 92121 (US).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, LY, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, 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,

[Continued on next page]

(54) Title: EFFICIENT FILTER WEIGHT COMPUTATION FOR A MIMO SYSTEM



(57) Abstract: Techniques to efficiently derive a spatial filter matrix are described. In a first scheme, a Hermitian matrix is iteratively derived based on a channel response matrix, and a matrix inversion is indirectly calculated by deriving the Hermitian matrix iteratively. The spatial filter matrix is derived based on the Hermitian matrix and the channel response matrix. In a second scheme, multiple rotations are performed to iteratively obtain first and second matrices for a pseudo-inverse matrix of the channel response matrix. The spatial filter matrix is derived based on the first and second matrices. In a third scheme, a matrix is formed based on the channel response matrix and decomposed to obtain a unitary matrix and a diagonal matrix. The spatial filter matrix is derived based on the unitary matrix, the diagonal matrix, and the channel response matrix.

WO 2006/138135 A2



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, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

— *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))*

Published:

— *without international search report and to be republished upon receipt of that report*

Declarations under Rule 4.17:

— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

EFFICIENT FILTER WEIGHT COMPUTATION FOR A MIMO SYSTEM

BACKGROUND

I. Field

[0001] The present disclosure relates generally to communication, and more specifically to techniques for computing filter weights in a communication system.

II. Background

[0002] A multiple-input multiple-output (MIMO) communication system employs multiple (T) transmit antennas at a transmitting station and multiple (R) receive antennas at a receiving station for data transmission. A MIMO channel formed by the T transmit antennas and the R receive antennas may be decomposed into S spatial channels, where $S \leq \min\{T, R\}$. The S spatial channels may be used to transmit data in a manner to achieve higher overall throughput and/or greater reliability.

[0003] The transmitting station may simultaneously transmit T data streams from the T transmit antennas. These data streams are distorted by the MIMO channel response and further degraded by noise and interference. The receiving station receives the transmitted data streams via the R receive antennas. The received signal from each receive antenna contains scaled versions of the T data streams sent by the transmitting station. The transmitted data streams are thus dispersed among the R received signals from the R receive antennas. The receiving station would then perform receiver spatial processing on the R received signals with a spatial filter matrix in order to recover the transmitted data streams.

[0004] The derivation of the weights for the spatial filter matrix is computationally intensive. This is because the spatial filter matrix is typically derived based on a function that contains a matrix inversion, and direct calculation of the matrix inversion is computationally intensive.

[0005] There is therefore a need in the art for techniques to efficiently compute the filter weights.

SUMMARY

[0006] Techniques for efficiently computing the weights for a spatial filter matrix are described herein. These techniques avoid direct computation of matrix inversion.

[0007] In a first embodiment for deriving a spatial filter matrix $\underline{\mathbf{M}}$, a Hermitian matrix $\underline{\mathbf{P}}$ is iteratively derived based on a channel response matrix $\underline{\mathbf{H}}$, and a matrix inversion is indirectly calculated by deriving the Hermitian matrix iteratively. The Hermitian matrix may be initialized to an identity matrix. One iteration is then performed for each row of the channel response matrix, and an efficient sequence of calculations is performed for each iteration. For the i -th iteration, an intermediate row vector $\underline{\mathbf{a}}_i$ is derived based on a channel response row vector $\underline{\mathbf{h}}_i$, which is the i -th row of the channel response matrix. A scalar r_i is derived based on the intermediate row vector and the channel response row vector. An intermediate matrix $\underline{\mathbf{C}}_i$ is also derived based on the intermediate row vector. The Hermitian matrix is then updated based on the scalar and the intermediate matrix. After all of the iterations are completed, the spatial filter matrix is derived based on the Hermitian matrix and the channel response matrix.

[0008] In a second embodiment, multiple rotations are performed to iteratively obtain a first matrix $\underline{\mathbf{P}}^{1/2}$ and a second matrix $\underline{\mathbf{B}}$ for a pseudo-inverse matrix of the channel response matrix. One iteration is performed for each row of the channel response matrix. For each iteration, a matrix $\underline{\mathbf{Y}}$ containing the first and second matrices from the prior iteration is formed. Multiple Givens rotations are then performed on matrix $\underline{\mathbf{Y}}$ to zero out elements in the first row of the matrix to obtain updated first and second matrices for the next iteration. After all of the iterations are completed, the spatial filter matrix is derived based on the first and second matrices.

[0009] In a third embodiment, a matrix $\underline{\mathbf{X}}$ is formed based on the channel response matrix and decomposed (e.g., using eigenvalue decomposition) to obtain a unitary matrix $\underline{\mathbf{V}}$ and a diagonal matrix $\underline{\mathbf{\Lambda}}$. The decomposition may be achieved by iteratively performing Jacobi rotations on matrix $\underline{\mathbf{X}}$. The spatial filter matrix is then derived based on the unitary matrix, the diagonal matrix, and the channel response matrix.

[0010] Various aspects and embodiments of the invention are described in further detail below.

BRIEF DESCRIPTION OF THE DRAWINGS

[0011] The features and nature of the present invention will become more apparent from the detailed description set forth below when taken in conjunction with the drawings in which like reference characters identify correspondingly throughout.

[0012] FIGS. 1, 2 and 3 show processes for computing an MMSE spatial filter matrix based on the first, second, and third embodiments, respectively.

[0013] FIG. 4 shows a block diagram of an access point and a user terminal.

DETAILED DESCRIPTION

[0014] The word “exemplary” is used herein to mean “serving as an example, instance, or illustration.” Any embodiment or design described herein as “exemplary” is not necessarily to be construed as preferred or advantageous over other embodiments or designs.

[0015] The filter weight computation techniques described herein may be used for a single-carrier MIMO system and a multi-carrier MIMO system. Multiple carriers may be obtained with orthogonal frequency division multiplexing (OFDM), interleaved frequency division multiple access (IFDMA), localized frequency division multiple access (LFDMA), or some other modulation technique. OFDM, IFDMA, and LFDMA effectively partition the overall system bandwidth into multiple (K) orthogonal frequency subbands, which are also called tones, subcarriers, bins, and frequency channels. Each subband is associated with a respective subcarrier that may be modulated with data. OFDM transmits modulation symbols in the frequency domain on all or a subset of the K subbands. IFDMA transmits modulation symbols in the time domain on subbands that are uniformly spaced across the K subbands. LFDMA transmits modulation symbols in the time domain and typically on adjacent subbands. For clarity, much of the following description is for a single-carrier MIMO system with a single subband.

[0016] A MIMO channel formed by multiple (T) transmit antennas at a transmitting station and multiple (R) receive antennas at a receiving station may be characterized by an $R \times T$ channel response matrix \mathbf{H} , which may be given as:

$$\underline{\mathbf{H}} = \begin{bmatrix} h_{1,1} & h_{1,2} & \cdots & h_{1,T} \\ h_{2,1} & h_{2,2} & \cdots & h_{2,T} \\ \vdots & \vdots & \ddots & \vdots \\ h_{R,1} & h_{R,2} & \cdots & h_{R,T} \end{bmatrix} = \begin{bmatrix} \underline{\mathbf{h}}_1 \\ \underline{\mathbf{h}}_2 \\ \vdots \\ \underline{\mathbf{h}}_R \end{bmatrix}, \quad \text{Eq (1)}$$

where $h_{i,j}$, for $i=1, \dots, R$ and $j=1, \dots, T$, denotes the coupling or complex channel gain between transmit antenna j and receive antenna i ; and

$\underline{\mathbf{h}}_i$ is a $1 \times T$ channel response row vector for receive antenna i , which is the i -th row of $\underline{\mathbf{H}}$.

For simplicity, the following description assumes that the MIMO channel is full rank and that the number of spatial channels (S) is given as: $S = T \leq R$.

[0017] The transmitting station may transmit T modulation symbols simultaneously from the T transmit antennas in each symbol period. The transmitting station may or may not perform spatial processing on the modulation symbols prior to transmission. For simplicity, the following description assumes that each modulation symbol is sent from one transmit antenna without any spatial processing.

[0018] The receiving station obtains R received symbols from the R receive antennas in each symbol period. The received symbols may be expressed as:

$$\underline{\mathbf{r}} = \underline{\mathbf{H}} \cdot \underline{\mathbf{s}} + \underline{\mathbf{n}}, \quad \text{Eq (2)}$$

where $\underline{\mathbf{s}}$ is a $T \times 1$ vector with T modulation symbols sent by the transmitting station;

$\underline{\mathbf{r}}$ is an $R \times 1$ vector with R received symbols obtained by the receiving station from the R receive antennas; and

$\underline{\mathbf{n}}$ is an $R \times 1$ vector of noise.

For simplicity, the noise may be assumed to be additive white Gaussian noise (AWGN) with a zero mean vector and a covariance matrix of $\sigma_n^2 \cdot \mathbf{I}$, where σ_n^2 is the variance of the noise and \mathbf{I} is the identity matrix.

[0019] The receiving station may use various receiver spatial processing techniques to recover the modulation symbols sent by the transmitting station. For example, the receiving station may perform minimum mean square error (MMSE) receiver spatial processing, as follows:

$$\hat{\underline{\mathbf{s}}} = (\sigma_n^2 \cdot \underline{\mathbf{I}} + \underline{\mathbf{H}}^H \cdot \underline{\mathbf{H}})^{-1} \cdot \underline{\mathbf{H}}^H \cdot \underline{\mathbf{r}} = \underline{\mathbf{P}} \cdot \underline{\mathbf{H}}^H \cdot \underline{\mathbf{r}} = \underline{\mathbf{M}} \cdot \underline{\mathbf{r}}, \quad \text{Eq (3)}$$

where $\underline{\mathbf{M}}$ is a $T \times R$ MMSE spatial filter matrix;

$\underline{\mathbf{P}}$ is a $T \times T$ Hermitian covariance matrix for the estimation error $\underline{\mathbf{s}} - \hat{\underline{\mathbf{s}}}$;

$\hat{\underline{\mathbf{s}}}$ is a $T \times 1$ vector that is an estimate of $\underline{\mathbf{s}}$; and

“ H ” denotes a conjugate transpose.

The covariance matrix $\underline{\mathbf{P}}$ may be given as $\underline{\mathbf{P}} = E[(\underline{\mathbf{s}} - \hat{\underline{\mathbf{s}}}) \cdot (\underline{\mathbf{s}} - \hat{\underline{\mathbf{s}}})^H]$, where $E[\]$ is an expectation operation. $\underline{\mathbf{P}}$ is also a Hermitian matrix whose off-diagonal elements have the following properties $p_{j,i} = p_{i,j}^*$, where “ $*$ ” denotes a complex conjugate.

[0020] As shown in equation (3), the MMSE spatial filter matrix $\underline{\mathbf{M}}$ has a matrix inverse calculation. Direct calculation of the matrix inversion is computationally intensive. The MMSE spatial filter matrix may be more efficiently derived based on the embodiments described below, which indirectly calculate the matrix inversion with an iterative process instead of directly calculating the matrix inversion.

[0021] In a first embodiment of computing the MMSE spatial filter matrix $\underline{\mathbf{M}}$, the Hermitian matrix $\underline{\mathbf{P}}$ is computed based on the Riccati equation. Hermitian matrix $\underline{\mathbf{P}}$ may be expressed as:

$$\underline{\mathbf{P}} = (\sigma_n^2 \cdot \underline{\mathbf{I}} + \underline{\mathbf{H}}^H \cdot \underline{\mathbf{H}})^{-1} = \left(\sigma_n^2 \cdot \underline{\mathbf{I}} + \sum_{k=1}^R \underline{\mathbf{h}}_k^H \cdot \underline{\mathbf{h}}_k \right)^{-1}. \quad \text{Eq (4)}$$

[0022] A $T \times T$ Hermitian matrix $\underline{\mathbf{P}}_i$ may be defined as:

$$\begin{aligned} \underline{\mathbf{P}}_i &= \left(\sigma_n^2 \cdot \underline{\mathbf{I}} + \sum_{k=1}^i \underline{\mathbf{h}}_k^H \cdot \underline{\mathbf{h}}_k \right)^{-1}, \\ &= \left(\sigma_n^2 \cdot \underline{\mathbf{I}} + \sum_{k=1}^{i-1} \underline{\mathbf{h}}_k^H \cdot \underline{\mathbf{h}}_k + \underline{\mathbf{h}}_i^H \cdot \underline{\mathbf{h}}_i \right)^{-1}, \\ &= (\underline{\mathbf{P}}_{i-1}^{-1} + \underline{\mathbf{h}}_i^H \cdot \underline{\mathbf{h}}_i)^{-1}, \end{aligned} \quad \text{Eq (5)}$$

[0023] The matrix inversion lemma may be applied to equation (5) to obtain the following:

$$\underline{\mathbf{P}}_i = \underline{\mathbf{P}}_{i-1} - \frac{\underline{\mathbf{P}}_{i-1} \cdot \underline{\mathbf{h}}_i^H \cdot \underline{\mathbf{h}}_i \cdot \underline{\mathbf{P}}_{i-1}}{r_i} \quad \text{and} \quad r_i = 1 + \underline{\mathbf{h}}_i \cdot \underline{\mathbf{P}}_{i-1} \cdot \underline{\mathbf{h}}_i^H, \quad \text{Eq (6)}$$

where r_i is a real-valued scalar. Equation (6) is referred to as the Riccati equation.

Matrix $\underline{\mathbf{P}}_i$ may be initialized as $\underline{\mathbf{P}}_0 = \frac{1}{\sigma_n^2} \cdot \underline{\mathbf{I}}$. After performing R iterations of equation

(6), for $i=1, \dots, R$, matrix $\underline{\mathbf{P}}_R$ is provided as matrix $\underline{\mathbf{P}}$, or $\underline{\mathbf{P}} = \underline{\mathbf{P}}_R$.

[0024] Equation (6) may be factored to obtain the following:

$$\underline{\mathbf{P}}_i = \underline{\mathbf{P}}_{i-1} - \frac{\underline{\mathbf{P}}_{i-1} \cdot \underline{\mathbf{h}}_i^H \cdot \underline{\mathbf{h}}_i \cdot \underline{\mathbf{P}}_{i-1}}{r_i} \quad \text{and} \quad r_i = \sigma_n^2 + \underline{\mathbf{h}}_i \cdot \underline{\mathbf{P}}_{i-1} \cdot \underline{\mathbf{h}}_i^H, \quad \text{Eq (7)}$$

where matrix $\underline{\mathbf{P}}_i$ is initialized as $\underline{\mathbf{P}}_0 = \underline{\mathbf{I}}$ and matrix $\underline{\mathbf{P}}$ is derived as $\underline{\mathbf{P}} = \frac{1}{\sigma_n^2} \cdot \underline{\mathbf{P}}_R$.

Equations (6) and (7) are different forms of a solution to equation (5). For simplicity, the same variables $\underline{\mathbf{P}}_i$ and r_i are used for both equations (6) and (7) even though these variables have different values in the two equations. The final results from equations (6) and (7), i.e., $\underline{\mathbf{P}}_R$ for equation (6) and $\frac{1}{\sigma_n^2} \cdot \underline{\mathbf{P}}_R$ for equation (7), are equivalent.

However, the calculations for the first iteration of equation (7) are simplified because $\underline{\mathbf{P}}_0$ is an identity matrix.

[0025] Each iteration of equation (7) may be performed as follows:

$$\underline{\mathbf{a}}_i = \underline{\mathbf{h}}_i \cdot \underline{\mathbf{P}}_{i-1}, \quad \text{Eq (8a)}$$

$$r_i = \sigma_n^2 + \underline{\mathbf{a}}_i \cdot \underline{\mathbf{h}}_i^H, \quad \text{Eq (8b)}$$

$$\underline{\mathbf{C}}_i = \underline{\mathbf{a}}_i^H \cdot \underline{\mathbf{a}}_i, \quad \text{and} \quad \text{Eq (8c)}$$

$$\underline{\mathbf{P}}_i = \underline{\mathbf{P}}_{i-1} - r_i^{-1} \cdot \underline{\mathbf{C}}_i, \quad \text{Eq (8d)}$$

where $\underline{\mathbf{a}}_i$ is a $1 \times T$ intermediate row vector of complex-valued elements; and

$\underline{\mathbf{C}}_i$ is a $T \times T$ intermediate Hermitian matrix.

[0026] In equation set (8), the sequence of operations is structured for efficient computation by hardware. Scalar r_i is computed before matrix $\underline{\mathbf{C}}_i$. The division by r_i in equation (7) is achieved with an inversion and a multiply. The inversion of r_i may be performed in parallel with the computation of $\underline{\mathbf{C}}_i$. The inversion of r_i may be achieved

with a shifter to normalize r_i and a look-up table to produce an inverted r_i value. The normalization of r_i may be compensated for in the multiplication with \underline{C}_i .

[0027] Matrix \underline{P}_i is initialized as a Hermitian matrix, or $\underline{P}_0 = \underline{I}$, and remains Hermitian through all of the iterations. Hence, only the upper (or lower) diagonal matrix needs to be calculated for each iteration. After R iterations are completed, matrix \underline{P} is obtained as $\underline{P} = \frac{1}{\sigma_n^2} \cdot \underline{P}_R$. The MMSE spatial filter matrix may then be computed as follows:

$$\underline{M} = \underline{P} \cdot \underline{H}^H = \frac{1}{\sigma_n^2} \cdot \underline{P}_R \cdot \underline{H}^H . \quad \text{Eq (9)}$$

[0028] FIG. 1 shows a process 100 for computing the MMSE spatial filter matrix \underline{M} based on the first embodiment. Matrix \underline{P}_i is initialized as $\underline{P}_0 = \underline{I}$ (block 112), and index i used to denote the iteration number is initialized as $i = 1$ (block 114). R iterations of the Riccati equation are then performed.

[0029] Each iteration of the Riccati equation is performed by block 120. For the i -th iteration, the intermediate row vector \underline{a}_i is computed based on the channel response row vector \underline{h}_i and the Hermitian matrix \underline{P}_{i-1} from the prior iteration, as shown in equation (8a) (block 122). The scalar r_i is computed based on the noise variance σ_n^2 , the intermediate row vector \underline{a}_i , and the channel response row vector \underline{h}_i , as shown in equation (8b) (block 124). Scalar r_i is then inverted (block 126). Intermediate matrix \underline{C}_i is computed based on the intermediate row vector \underline{a}_i , as shown in equation (8c) (block 128). Matrix \underline{P}_i is then updated based on the inverted scalar r_i and the intermediate matrix \underline{C}_i , as shown in equation (8d) (block 130).

[0030] A determination is then made whether all R iterations have been performed (block 132). If the answer is 'No', then index i is incremented (block 134), and the process returns to block 122 to perform another iteration. Otherwise, if all R iterations have been performed, then the MMSE spatial filter matrix \underline{M} is computed based on the Hermitian matrix \underline{P}_R for the last iteration, the channel response matrix \underline{H} , and the noise variance σ_n^2 , as shown in equation (9) (block 136). Matrix \underline{M} may then be used for receiver spatial processing as shown in equation (3).

[0031] In a second embodiment of computing the MMSE spatial filter matrix $\underline{\mathbf{M}}$, the Hermitian matrix $\underline{\mathbf{P}}$ is determined by deriving the square root of $\underline{\mathbf{P}}$, which is $\underline{\mathbf{P}}^{1/2}$, based on an iterative procedure. The receiver spatial processing in equation (3) may be expressed as:

$$\begin{aligned}\hat{\underline{\mathbf{s}}} &= (\sigma_n^2 \cdot \underline{\mathbf{I}} + \underline{\mathbf{H}}^H \cdot \underline{\mathbf{H}})^{-1} \cdot \underline{\mathbf{H}}^H \cdot \underline{\mathbf{r}} = \left(\begin{bmatrix} \underline{\mathbf{H}}^H & \sigma_n \cdot \underline{\mathbf{I}} \end{bmatrix} \cdot \begin{bmatrix} \underline{\mathbf{H}} \\ \sigma_n \cdot \underline{\mathbf{I}} \end{bmatrix} \right)^{-1} \begin{bmatrix} \underline{\mathbf{H}}^H & \sigma_n \cdot \underline{\mathbf{I}} \end{bmatrix} \cdot \begin{bmatrix} \underline{\mathbf{r}} \\ \underline{\mathbf{0}}_{T \times 1} \end{bmatrix}, \\ &= (\underline{\mathbf{U}}^H \cdot \underline{\mathbf{U}})^{-1} \cdot \underline{\mathbf{U}}^H \cdot \begin{bmatrix} \underline{\mathbf{r}} \\ \underline{\mathbf{0}}_{T \times 1} \end{bmatrix} = \underline{\mathbf{U}}^p \cdot \begin{bmatrix} \underline{\mathbf{r}} \\ \underline{\mathbf{0}}_{T \times 1} \end{bmatrix}, \\ &= \underline{\mathbf{H}}_{\sigma_n}^p \cdot \underline{\mathbf{r}},\end{aligned}\tag{Eq (10)}$$

where $\underline{\mathbf{U}} = \begin{bmatrix} \underline{\mathbf{H}} \\ \sigma_n \cdot \underline{\mathbf{I}} \end{bmatrix}$ is a $(R+T) \times T$ augmented channel matrix;

$\underline{\mathbf{U}}^p$ is a $T \times (R+T)$ pseudo-inverse matrix obtained from a Moore-Penrose inverse or a pseudo-inverse operation on $\underline{\mathbf{U}}$, or $\underline{\mathbf{U}}^p = (\underline{\mathbf{U}}^H \cdot \underline{\mathbf{U}})^{-1} \cdot \underline{\mathbf{U}}^H$;

$\underline{\mathbf{0}}_{T \times 1}$ is a $T \times 1$ vector of all zeros; and

$\underline{\mathbf{H}}_{\sigma_n}^p$ is a $T \times R$ sub-matrix containing the first R columns of $\underline{\mathbf{U}}^p$.

[0032] QR decomposition may be performed on the augmented channel matrix, as follows:

$$\begin{bmatrix} \underline{\mathbf{H}} \\ \sigma_n \cdot \underline{\mathbf{I}} \end{bmatrix} = \underline{\mathbf{Q}} \cdot \underline{\mathbf{R}} = \begin{bmatrix} \underline{\mathbf{B}} \\ \underline{\mathbf{Q}}_2 \end{bmatrix} \cdot \underline{\mathbf{R}},\tag{Eq (11)}$$

where $\underline{\mathbf{Q}}$ is a $(R+T) \times T$ matrix with orthonormal columns;

$\underline{\mathbf{R}}$ is a $T \times T$ matrix that is non-singular;

$\underline{\mathbf{B}}$ is an $R \times T$ matrix containing the first R rows of $\underline{\mathbf{Q}}$; and

$\underline{\mathbf{Q}}_2$ is a $T \times T$ matrix containing the last T rows of $\underline{\mathbf{Q}}$.

[0033] The QR decomposition in equation (11) decomposes the augmented channel matrix into an orthonormal matrix $\underline{\mathbf{Q}}$ and a non-singular matrix $\underline{\mathbf{R}}$. An orthonormal matrix $\underline{\mathbf{Q}}$ has the following property: $\underline{\mathbf{Q}}^H \cdot \underline{\mathbf{Q}} = \underline{\mathbf{I}}$, which means that the columns of the orthonormal matrix are orthogonal to one another and each column has unit power. A non-singular matrix is a matrix that an inverse can be computed for.

[0034] The Hermitian matrix $\underline{\mathbf{P}}$ may then be expressed as:

$$\begin{aligned}\underline{\mathbf{P}} &= (\sigma_n^2 \cdot \underline{\mathbf{I}} + \underline{\mathbf{H}}^H \cdot \underline{\mathbf{H}})^{-1} = \left(\begin{bmatrix} \underline{\mathbf{H}}^H & \sigma_n \cdot \underline{\mathbf{I}} \\ \underline{\mathbf{H}} & \sigma_n \cdot \underline{\mathbf{I}} \end{bmatrix} \right)^{-1}, \\ &= ((\underline{\mathbf{Q}} \cdot \underline{\mathbf{R}})^H \cdot \underline{\mathbf{Q}} \cdot \underline{\mathbf{R}})^{-1} = (\underline{\mathbf{R}}^H \cdot \underline{\mathbf{Q}}^H \cdot \underline{\mathbf{Q}} \cdot \underline{\mathbf{R}})^{-1}, \quad \text{Eq (12)} \\ &= (\underline{\mathbf{R}}^H \cdot \underline{\mathbf{R}})^{-1} = \underline{\mathbf{R}}^{-1} \cdot \underline{\mathbf{R}}^{-H} = \underline{\mathbf{P}}^{1/2} \cdot \underline{\mathbf{P}}^{H/2}.\end{aligned}$$

$\underline{\mathbf{R}}$ is the Cholesky factorization or matrix square root of $\underline{\mathbf{P}}^{-1}$. Hence, $\underline{\mathbf{P}}^{1/2}$ is equal to $\underline{\mathbf{R}}^{-1}$ and is called the square-root of $\underline{\mathbf{P}}$.

[0035] The pseudo-inverse matrix in equation (10) may then be expressed as:

$$\underline{\mathbf{U}}^p = \underline{\mathbf{P}} \cdot \begin{bmatrix} \underline{\mathbf{H}}^H & \sigma_n \cdot \underline{\mathbf{I}} \end{bmatrix} = (\underline{\mathbf{R}}^{-1} \cdot \underline{\mathbf{R}}^{-H}) \cdot (\underline{\mathbf{R}}^H \cdot \underline{\mathbf{Q}}^H) = \underline{\mathbf{R}}^{-1} \cdot \underline{\mathbf{Q}}^H = \underline{\mathbf{P}}^{1/2} \cdot \begin{bmatrix} \underline{\mathbf{B}} \\ \underline{\mathbf{Q}}_2 \end{bmatrix}^H. \quad \text{Eq (13)}$$

Sub-matrix $\underline{\mathbf{H}}_{\sigma_n}^p$, which is also the MMSE spatial filter matrix, may then be expressed as:

$$\underline{\mathbf{H}}_{\sigma_n}^p = \underline{\mathbf{M}} = \underline{\mathbf{P}}^{1/2} \cdot \underline{\mathbf{B}}^H. \quad \text{Eq (14)}$$

[0036] Equation (10) may then be expressed as:

$$\hat{\underline{\mathbf{s}}} = \underline{\mathbf{H}}_{\sigma_n}^p \cdot \underline{\mathbf{r}} = \underline{\mathbf{P}}^{1/2} \cdot \underline{\mathbf{B}}^H \cdot \underline{\mathbf{r}} = \underline{\mathbf{M}} \cdot \underline{\mathbf{r}}. \quad \text{Eq (15)}$$

[0037] Matrices $\underline{\mathbf{P}}^{1/2}$ and $\underline{\mathbf{B}}$ may be computed iteratively as follows:

$$\underline{\mathbf{Y}}_i \cdot \underline{\Theta}_i = \underline{\mathbf{Z}}_i, \quad \text{or} \quad \text{Eq (16)}$$

$$\begin{bmatrix} 1 & \underline{\mathbf{h}}_i \cdot \underline{\mathbf{P}}_{i-1}^{1/2} \\ \underline{\mathbf{0}}_{T \times 1} & \underline{\mathbf{P}}_{i-1}^{1/2} \\ -\underline{\mathbf{e}}_i & \underline{\mathbf{B}}_{i-1} \end{bmatrix} \cdot \underline{\Theta}_i = \begin{bmatrix} r_i^{1/2} & \underline{\mathbf{0}}_{1 \times T} \\ \underline{\mathbf{k}}_i & \underline{\mathbf{P}}_i^{1/2} \\ \underline{\mathbf{l}}_i & \underline{\mathbf{B}}_i \end{bmatrix}, \quad \text{Eq (17)}$$

where $\underline{\mathbf{Y}}_i$ is a $(T+R+1) \times (T+1)$ matrix containing elements derived based on $\underline{\mathbf{P}}_{i-1}^{1/2}$,

$\underline{\mathbf{B}}_{i-1}$ and $\underline{\mathbf{h}}_i$;

$\underline{\Theta}_i$ is a $(T+1) \times (T+1)$ unitary transformation matrix;

$\underline{\mathbf{Z}}_i$ is a $(T + R + 1) \times (T + 1)$ transformed matrix containing elements for $\underline{\mathbf{P}}_i^{1/2}$,

$\underline{\mathbf{B}}_i$ and r_i ;

$\underline{\mathbf{e}}_i$ is an $R \times 1$ vector with one (1.0) as the i -th element and zeros elsewhere; and

$\underline{\mathbf{k}}_i$ is a $T \times 1$ vector and $\underline{\mathbf{l}}_i$ is an $R \times 1$ vector, both of which are non-essential.

Matrices $\underline{\mathbf{P}}^{1/2}$ and $\underline{\mathbf{B}}$ are initialized as $\underline{\mathbf{P}}_0^{1/2} = \frac{1}{\sigma_n} \cdot \underline{\mathbf{I}}$ and $\underline{\mathbf{B}}_0 = \underline{\mathbf{0}}_{R \times T}$.

[0038] The transformation in equation (17) may be performed iteratively, as described below. For clarity, each iteration of equation (17) is called an outer iteration. R outer iterations of equation (17) are performed for the R channel response row vectors $\underline{\mathbf{h}}_i$, for $i = 1, \dots, R$. For each outer iteration, the unitary transformation matrix $\underline{\Theta}_i$ in equation (17) results in the transformed matrix $\underline{\mathbf{Z}}_i$ containing all zeros in the first row except for the first element. The first column of the transformed matrix $\underline{\mathbf{Z}}_i$ contains $r_i^{1/2}$, $\underline{\mathbf{k}}_i$, and $\underline{\mathbf{l}}_i$. The last T columns of $\underline{\mathbf{Z}}_i$ contain updated $\underline{\mathbf{P}}_i^{1/2}$ and $\underline{\mathbf{B}}_i$. The first column of $\underline{\mathbf{Z}}_i$ does not need to be calculated since only $\underline{\mathbf{P}}_i^{1/2}$ and $\underline{\mathbf{B}}_i$ are used in the next iteration. $\underline{\mathbf{P}}_i^{1/2}$ is an upper triangular matrix. After R outer iterations are completed, $\underline{\mathbf{P}}_R^{1/2}$ is provided as $\underline{\mathbf{P}}^{1/2}$, and $\underline{\mathbf{B}}_R$ is provided as $\underline{\mathbf{B}}$. The MMSE spatial filter matrix $\underline{\mathbf{M}}$ may then be computed as based on $\underline{\mathbf{P}}^{1/2}$ and $\underline{\mathbf{B}}$, as shown in equation (14).

[0039] For each outer iteration i , the transformation in equation (17) may be performed by successively zeroing out one element in the first row of $\underline{\mathbf{Y}}_i$ at a time with a 2×2 Givens rotation. T inner iterations of the Givens rotation may be performed to zero out the last T elements in the first row of $\underline{\mathbf{Y}}_i$.

[0040] For each outer iteration i , a matrix $\underline{\mathbf{Y}}_{i,j}$ may be initialized as $\underline{\mathbf{Y}}_{i,1} = \underline{\mathbf{Y}}_i$. For each inner iteration j , for $j = 1, \dots, T$, of outer iteration i , a $(T + R + 1) \times 2$ sub-matrix $\underline{\mathbf{Y}}'_{i,j}$ containing the first and $(j + 1)$ -th columns of $\underline{\mathbf{Y}}_{i,j}$ is initially formed. The Givens rotation is then performed on sub-matrix $\underline{\mathbf{Y}}'_{i,j}$ to generate a $(T + R + 1) \times 2$ sub-matrix $\underline{\mathbf{Y}}''_{i,j}$ containing a zero for the second element in the first row. The Givens rotation may be expressed as:

$$\underline{\mathbf{Y}}''_{i,j} = \underline{\mathbf{Y}}'_{i,j} \cdot \underline{\mathbf{G}}_{i,j}, \quad \text{Eq (18)}$$

where $\underline{\mathbf{G}}_{i,j}$ is a 2×2 Givens rotation matrix for the j -th inner iteration of the i -th outer iteration and is described below. Matrix $\underline{\mathbf{Y}}_{i,j+1}$ is then formed by first setting $\underline{\mathbf{Y}}_{i,j+1} = \underline{\mathbf{Y}}_{i,j}$, then replacing the first column of $\underline{\mathbf{Y}}_{i,j+1}$ with the first column of $\underline{\mathbf{Y}}''_{i,j}$, and then replacing the $(j+1)$ -th column of $\underline{\mathbf{Y}}_{i,j+1}$ with the second column of $\underline{\mathbf{Y}}''_{i,j}$. The Givens rotation thus modifies only two columns of $\underline{\mathbf{Y}}_{i,j}$ in the j -th inner iteration to produce $\underline{\mathbf{Y}}_{i,j+1}$ for the next inner iteration. The Givens rotation may be performed in-place on two columns of $\underline{\mathbf{Y}}_i$ for each inner iteration, so that intermediate matrices $\underline{\mathbf{Y}}_{i,j}$, $\underline{\mathbf{Y}}'_{i,j}$, $\underline{\mathbf{Y}}''_{i,j}$ and $\underline{\mathbf{Y}}_{i,j+1}$ are not needed and are described above for clarity.

[0041] For the j -th inner iteration of the i -th outer iteration, the Givens rotation matrix $\underline{\mathbf{G}}_{i,j}$ is determined based on the first element (which is always a real value) and the $(j+1)$ -th element in the first row of $\underline{\mathbf{Y}}_{i,j}$. The first element may be denoted as a , and the $(j+1)$ -th element may be denoted as $b \cdot e^{j\theta}$. The Givens rotation matrix $\underline{\mathbf{G}}_{i,j}$ may then be derived as follows:

$$\underline{\mathbf{G}}_{i,j} = \begin{bmatrix} c & -s \\ s \cdot e^{-j\theta} & c \cdot e^{-j\theta} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & e^{-j\theta} \end{bmatrix} \cdot \begin{bmatrix} c & -s \\ s & c \end{bmatrix}, \quad \text{Eq (19)}$$

where $c = \frac{a}{\sqrt{a^2 + b^2}}$ and $s = \frac{b}{\sqrt{a^2 + b^2}}$ for equation (19).

[0042] FIG. 2 shows a process 200 for computing the MMSE spatial filter matrix $\underline{\mathbf{M}}$ based on the second embodiment. Matrix $\underline{\mathbf{P}}_i^{1/2}$ is initialized as $\underline{\mathbf{P}}_0^{1/2} = \frac{1}{\sigma_n} \cdot \underline{\mathbf{I}}$, and matrix $\underline{\mathbf{B}}_i$ is initialized as $\underline{\mathbf{B}}_0 = \underline{\mathbf{0}}$ (block 212). Index i used to denote the outer iteration number is initialized as $i=1$, and index j used to denote the inner iteration number is initialized as $j=1$ (block 214). R outer iterations of the unitary transformation in equation (17) are then performed (block 220).

[0043] For the i -th outer iteration, matrix $\underline{\mathbf{Y}}_i$ is initially formed with the channel response row vector $\underline{\mathbf{h}}_i$ and matrices $\underline{\mathbf{P}}_{i-1}^{1/2}$ and $\underline{\mathbf{B}}_{i-1}$, as shown in equation (17) (block 222). Matrix $\underline{\mathbf{Y}}_i$ is then referred to as matrix $\underline{\mathbf{Y}}_{i,j}$ for the inner iterations (block 224). T inner iterations of the Givens rotation are then performed on matrix $\underline{\mathbf{Y}}_{i,j}$ (block 230).

[0044] For the j -th inner iteration, the Givens rotation matrix $\underline{\mathbf{G}}_{i,j}$ is derived based on the first and $(j+1)$ -th elements in the first row of $\underline{\mathbf{Y}}_{i,j}$, as shown in equation (19) (block 232). The Givens rotation matrix $\underline{\mathbf{G}}_{i,j}$ is then applied to the first and $(j+1)$ -th columns of $\underline{\mathbf{Y}}_{i,j}$ to obtain $\underline{\mathbf{Y}}_{i,j+1}$, as shown in equation (18) (block 234). A determination is then made whether all T inner iterations have been performed (block 236). If the answer is 'No', then index j is incremented (block 238), and the process returns to block 232 to perform another inner iteration.

[0045] If all T inner iterations have been performed for the current outer iteration and the answer is 'Yes' for block 236, then the latest $\underline{\mathbf{Y}}_{i,j+1}$ is equal to $\underline{\mathbf{Z}}_i$ in equation (17). Updated matrices $\underline{\mathbf{P}}_i^{1/2}$ and $\underline{\mathbf{B}}_i$ are obtained from the latest $\underline{\mathbf{Y}}_{i,j+1}$ (block 240). A determination is then made whether all R outer iterations have been performed (block 242). If the answer is 'No', then index i is incremented, and index j is reinitialized as $j=1$ (block 244). The process then returns to block 222 to perform another outer iteration with $\underline{\mathbf{P}}_i^{1/2}$ and $\underline{\mathbf{B}}_i$. Otherwise, if all R outer iterations have been performed and the answer is 'Yes' for block 242, then the MMSE spatial filter matrix $\underline{\mathbf{M}}$ is computed based on $\underline{\mathbf{P}}_i^{1/2}$ and $\underline{\mathbf{B}}_i$, as shown in equation (14) (block 246). Matrix $\underline{\mathbf{M}}$ may then be used for receiver spatial processing as shown in equation (15).

[0046] In a third embodiment of computing the MMSE spatial filter matrix $\underline{\mathbf{M}}$, eigenvalue decomposition of $\underline{\mathbf{P}}^{-1}$ is performed as follows:

$$\underline{\mathbf{P}}^{-1} = \sigma_n^2 \cdot \underline{\mathbf{I}} + \underline{\mathbf{H}}^H \cdot \underline{\mathbf{H}} = \underline{\mathbf{V}} \cdot \underline{\mathbf{\Lambda}} \cdot \underline{\mathbf{V}}^H, \quad \text{Eq (20)}$$

where $\underline{\mathbf{V}}$ is a $T \times T$ unitary matrix of eigenvectors; and

$\underline{\mathbf{\Lambda}}$ is a $T \times T$ diagonal matrix with real eigenvalues along the diagonal.

[0047] Eigenvalue decomposition of a 2×2 Hermitian matrix $\underline{\mathbf{X}}_{2 \times 2}$ may be achieved using various techniques. In an embodiment, eigenvalue decomposition of $\underline{\mathbf{X}}_{2 \times 2}$ is achieved by performing a complex Jacobi rotation on $\underline{\mathbf{X}}_{2 \times 2}$ to obtain a 2×2 matrix $\underline{\mathbf{V}}_{2 \times 2}$ of eigenvectors of $\underline{\mathbf{X}}_{2 \times 2}$. The elements of $\underline{\mathbf{X}}_{2 \times 2}$ and $\underline{\mathbf{V}}_{2 \times 2}$ may be given as:

$$\underline{\mathbf{X}}_{2 \times 2} = \begin{bmatrix} x_{1,1} & x_{1,2} \\ x_{2,1} & x_{2,2} \end{bmatrix} \quad \text{and} \quad \underline{\mathbf{V}}_{2 \times 2} = \begin{bmatrix} v_{1,1} & v_{1,2} \\ v_{2,1} & v_{2,2} \end{bmatrix}. \quad \text{Eq (21)}$$

The elements of $\underline{\mathbf{V}}_{2 \times 2}$ may be computed directly from the elements of $\underline{\mathbf{X}}_{2 \times 2}$, as follows:

$$r = \sqrt{(\operatorname{Re}\{x_{1,2}\})^2 + (\operatorname{Im}\{x_{1,2}\})^2}, \quad \text{Eq (22a)}$$

$$c_1 = \frac{\operatorname{Re}\{x_{1,2}\}}{r} = \cos(\angle x_{1,2}), \quad \text{Eq (22b)}$$

$$s_1 = \frac{\operatorname{Im}\{x_{1,2}\}}{r} = \sin(\angle x_{1,2}), \quad \text{Eq (22c)}$$

$$g_1 = c_1 - js_1, \quad \text{Eq (22d)}$$

$$\tau = \frac{x_{2,2} - x_{1,1}}{2 \cdot r}, \quad \text{Eq (22e)}$$

$$x = \sqrt{1 + \tau^2}, \quad \text{Eq (22f)}$$

$$t = \frac{1}{|\tau| + x}, \quad \text{Eq (22g)}$$

$$c = \frac{1}{\sqrt{1 + t^2}}, \quad \text{Eq (22h)}$$

$$s = t \cdot c = \sqrt{1 - c^2}, \quad \text{Eq (22i)}$$

if $(x_{2,2} - x_{1,1}) < 0$

$$\text{then } \underline{\mathbf{V}}_{2 \times 2} = \begin{bmatrix} c & -s \\ g_1 \cdot s & g_1 \cdot c \end{bmatrix}, \quad \text{Eq (22j)}$$

$$\text{else } \underline{\mathbf{V}}_{2 \times 2} = \begin{bmatrix} s & c \\ g_1 \cdot c & -g_1 \cdot s \end{bmatrix}. \quad \text{Eq (22k)}$$

[0048] Eigenvalue decomposition of a $T \times T$ Hermitian matrix $\underline{\mathbf{X}}$ that is larger than 2×2 may be performed with an iterative process. This iterative process uses the Jacobi rotation repeatedly to zero out off-diagonal elements in $\underline{\mathbf{X}}$. For the iterative process, index i denotes the iteration number and is initialized as $i = 1$. $\underline{\mathbf{X}}$ is a $T \times T$ Hermitian matrix to be decomposed and is set as $\underline{\mathbf{X}} = \underline{\mathbf{P}}^{-1}$. Matrix $\underline{\mathbf{D}}_i$ is an approximation of diagonal matrix $\underline{\mathbf{\Lambda}}$ in equation (20) and is initialized as $\underline{\mathbf{D}}_0 = \underline{\mathbf{X}}$. Matrix $\underline{\mathbf{V}}_i$ is an approximation of unitary matrix $\underline{\mathbf{V}}$ in equation (20) and is initialized as $\underline{\mathbf{V}}_0 = \underline{\mathbf{I}}$.

[0049] A single iteration of the Jacobi rotation to update matrices $\underline{\mathbf{D}}_i$ and $\underline{\mathbf{V}}_i$ may be performed as follows. First, a 2×2 Hermitian matrix $\underline{\mathbf{D}}_{pq}$ is formed based on the current matrix $\underline{\mathbf{D}}_i$, as follows:

$$\underline{\mathbf{D}}_{pq} = \begin{bmatrix} d_{p,p} & d_{p,q} \\ d_{q,p} & d_{q,q} \end{bmatrix}, \quad \text{Eq (23)}$$

where $d_{p,q}$ is the element at location (p,q) in $\underline{\mathbf{D}}_i$, $p \in \{1, \dots, T\}$, $q \in \{1, \dots, T\}$, and $p \neq q$. $\underline{\mathbf{D}}_{pq}$ is a 2×2 submatrix of $\underline{\mathbf{D}}_i$, and the four elements of $\underline{\mathbf{D}}_{pq}$ are four elements at locations (p,p) , (p,q) , (q,p) and (q,q) in $\underline{\mathbf{D}}_i$. Indices p and q may be selected as described below.

[0050] Eigenvalue decomposition of $\underline{\mathbf{D}}_{pq}$ is then performed as shown in equation set (22) to obtain a 2×2 unitary matrix $\underline{\mathbf{V}}_{pq}$ of eigenvectors of $\underline{\mathbf{D}}_{pq}$. For the eigenvalue decomposition of $\underline{\mathbf{D}}_{pq}$, $\underline{\mathbf{X}}_{2 \times 2}$ in equation (21) is replaced with $\underline{\mathbf{D}}_{pq}$, and $\underline{\mathbf{V}}_{2 \times 2}$ from equation (22j) or (22k) is provided as $\underline{\mathbf{V}}_{pq}$.

[0051] A $T \times T$ complex Jacobi rotation matrix $\underline{\mathbf{T}}_{pq}$ is then formed with $\underline{\mathbf{V}}_{pq} \cdot \underline{\mathbf{T}}_{pq}$ is an identity matrix with four elements at locations (p,p) , (p,q) , (q,p) and (q,q) replaced with elements $v_{1,1}$, $v_{1,2}$, $v_{2,1}$ and $v_{2,2}$, respectively, in $\underline{\mathbf{V}}_{pq}$.

[0052] Matrix $\underline{\mathbf{D}}_i$ is then updated as follows:

$$\underline{\mathbf{D}}_{i+1} = \underline{\mathbf{T}}_{pq}^H \cdot \underline{\mathbf{D}}_i \cdot \underline{\mathbf{T}}_{pq} \quad \text{Eq (24)}$$

Equation (24) zeros out two off-diagonal elements at locations (p,q) and (q,p) in $\underline{\mathbf{D}}_i$. The computation may alter the values of other off-diagonal elements in $\underline{\mathbf{D}}_i$.

[0053] Matrix $\underline{\mathbf{V}}_i$ is also updated as follows:

$$\underline{\mathbf{V}}_{i+1} = \underline{\mathbf{V}}_i \cdot \underline{\mathbf{T}}_{pq} \quad \text{Eq (25)}$$

$\underline{\mathbf{V}}_i$ may be viewed as a cumulative transformation matrix that contains all of the Jacobi rotation matrices $\underline{\mathbf{T}}_{pq}$ used on $\underline{\mathbf{D}}_i$.

[0054] Each iteration of the Jacobi rotation zeros out two off-diagonal elements of $\underline{\mathbf{D}}_i$. Multiple iterations of the Jacobi rotation may be performed for different values of indices p and q to zero out all of the off-diagonal elements of $\underline{\mathbf{D}}_i$. A single sweep across all possible values of indices p and q may be performed as follows. Index p is stepped from 1 through $T-1$ in increments of one. For each value of p , index q is stepped from $p+1$ through T in increments of one. The Jacobi rotation is performed for each different combination of values for p and q . Multiple sweeps may be performed until $\underline{\mathbf{D}}_i$ and $\underline{\mathbf{V}}_i$ are sufficiently accurate estimates of $\underline{\mathbf{\Lambda}}$ and $\underline{\mathbf{V}}$, respectively.

[0055] Equation (20) may be rewritten as follows:

$$\underline{\mathbf{P}} = (\sigma_n^2 \cdot \underline{\mathbf{I}} + \underline{\mathbf{H}}^H \cdot \underline{\mathbf{H}})^{-1} = \underline{\mathbf{V}} \cdot \underline{\mathbf{\Lambda}}^{-1} \cdot \underline{\mathbf{V}}^H, \quad \text{Eq (26)}$$

where $\underline{\mathbf{\Lambda}}^{-1}$ is a diagonal matrix whose elements are the inverse of the corresponding elements in $\underline{\mathbf{\Lambda}}$. The eigenvalue decomposition of $\underline{\mathbf{X}} = \underline{\mathbf{P}}^{-1}$ provides estimates of $\underline{\mathbf{\Lambda}}$ and $\underline{\mathbf{V}}$. $\underline{\mathbf{\Lambda}}$ may be inverted to obtain $\underline{\mathbf{\Lambda}}^{-1}$.

[0056] The MMSE spatial filter matrix may then be computed as follows:

$$\underline{\mathbf{M}} = \underline{\mathbf{P}} \cdot \underline{\mathbf{H}}^H = \underline{\mathbf{V}} \cdot \underline{\mathbf{\Lambda}}^{-1} \cdot \underline{\mathbf{V}}^H \cdot \underline{\mathbf{H}}^H. \quad \text{Eq (27)}$$

[0057] FIG. 3 shows a process 300 for computing the MMSE spatial filter matrix $\underline{\mathbf{M}}$ based on the third embodiment. Hermitian matrix $\underline{\mathbf{P}}^{-1}$ is initially derived based on the channel response matrix $\underline{\mathbf{H}}$, as shown in equation (20) (block 312). Eigenvalue decomposition of $\underline{\mathbf{P}}^{-1}$ is then performed to obtain unitary matrix $\underline{\mathbf{V}}$ and diagonal matrix $\underline{\mathbf{\Lambda}}$, as also shown in equation (20) (block 314). The eigenvalue decomposition may be iteratively performed with a number of Jacobi rotations, as described above. The MMSE spatial filter matrix $\underline{\mathbf{M}}$ is then derived based on the unitary matrix $\underline{\mathbf{V}}$, the diagonal matrix $\underline{\mathbf{\Lambda}}$, and the channel response matrix $\underline{\mathbf{H}}$, as shown in equation (27) (block 316).

[0058] The MMSE spatial filter matrix $\underline{\mathbf{M}}$ derived based on each of the embodiments described above is a biased MMSE solution. The biased spatial filter matrix $\underline{\mathbf{M}}$ may be scaled by a diagonal matrix $\underline{\mathbf{D}}_{mmse}$ to obtain an unbiased MMSE

spatial filter matrix $\underline{\mathbf{M}}_{mmse}$. Matrix $\underline{\mathbf{D}}_{mmse}$ may be derived as $\underline{\mathbf{D}}_{mmse} = [\text{diag} [\underline{\mathbf{M}} \cdot \underline{\mathbf{H}}]]^{-1}$, where $\text{diag} [\underline{\mathbf{M}} \cdot \underline{\mathbf{H}}]$ is a diagonal matrix containing the diagonal elements of $\underline{\mathbf{M}} \cdot \underline{\mathbf{H}}$.

[0059] The computation described above may also be used to derive spatial filter matrices for a zero-forcing (ZF) technique (which is also called a channel correlation matrix inversion (CCMI) technique), a maximal ratio combining (MRC) technique, and so on. For example, the receiving station may perform zero-forcing and MRC receiver spatial processing, as follows:

$$\hat{\underline{\mathbf{s}}}_{zf} = (\underline{\mathbf{H}}^H \cdot \underline{\mathbf{H}})^{-1} \cdot \underline{\mathbf{H}}^H \cdot \underline{\mathbf{r}} = \underline{\mathbf{P}}_{zf} \cdot \underline{\mathbf{H}}^H \cdot \underline{\mathbf{r}} = \underline{\mathbf{M}}_{zf} \cdot \underline{\mathbf{r}}, \quad \text{Eq (28)}$$

$$\hat{\underline{\mathbf{s}}}_{mrc} = [\text{diag} (\underline{\mathbf{H}}^H \cdot \underline{\mathbf{H}})^{-1}] \cdot \underline{\mathbf{H}}^H \cdot \underline{\mathbf{r}} = [\text{diag} (\underline{\mathbf{P}}_{zf})] \cdot \underline{\mathbf{H}}^H \cdot \underline{\mathbf{r}} = \underline{\mathbf{M}}_{mrc} \cdot \underline{\mathbf{r}}, \quad \text{Eq (29)}$$

where $\underline{\mathbf{M}}_{zf}$ is a $T \times R$ zero-forcing spatial filter matrix;

$\underline{\mathbf{M}}_{mrc}$ is a $T \times R$ MRC spatial filter matrix;

$\underline{\mathbf{P}}_{zf} = (\underline{\mathbf{H}}^H \cdot \underline{\mathbf{H}})^{-1}$ is a $T \times T$ Hermitian matrix; and

$[\text{diag} (\underline{\mathbf{P}}_{zf})]$ is a $T \times T$ diagonal matrix containing the diagonal elements of $\underline{\mathbf{P}}_{zf}$.

A matrix inversion is needed to compute $\underline{\mathbf{P}}_{zf}$ directly. $\underline{\mathbf{P}}_{zf}$ may be computed using the embodiments described above for the MMSE spatial filter matrix.

[0060] The description above assumes that T modulation symbols are sent simultaneously from T transmit antennas without any spatial processing. The transmitting station may perform spatial processing prior to transmission, as follows:

$$\underline{\mathbf{x}} = \underline{\mathbf{W}} \cdot \underline{\mathbf{s}}, \quad \text{Eq (30)}$$

where $\underline{\mathbf{x}}$ is a $T \times 1$ vector with T transmit symbols to be sent from the T transmit antennas; and

$\underline{\mathbf{W}}$ is a $T \times S$ transmit matrix.

Transmit matrix $\underline{\mathbf{W}}$ may be (1) a matrix of right singular vectors obtained by performing singular value decomposition of $\underline{\mathbf{H}}$, (2) a matrix of eigenvectors obtained by performing eigenvalue decomposition of $\underline{\mathbf{H}}^H \cdot \underline{\mathbf{H}}$, or (3) a steering matrix selected to spatially spread the modulation symbols across the S spatial channels of the MIMO channel. An effective channel response matrix $\underline{\mathbf{H}}_{eff}$ observed by the modulation

symbols may then be given as $\underline{\mathbf{H}}_{eff} = \underline{\mathbf{H}} \cdot \underline{\mathbf{W}}$. The computation described above may be performed based on $\underline{\mathbf{H}}_{eff}$ instead of $\underline{\mathbf{H}}$.

[0061] For clarity, the description above is for a single-carrier MIMO system with a single subband. For a multi-carrier MIMO system, a channel response matrix $\underline{\mathbf{H}}(k)$ may be obtained for each subband k of interest. A spatial filter matrix $\underline{\mathbf{M}}(k)$ may then be derived for each subband k based on the channel response matrix $\underline{\mathbf{H}}(k)$ for that subband.

[0062] The computation described above for the spatial filter matrix may be performed using various types of processors such as a floating-point processor, a fixed-point processor, a Coordinate Rotational Digital Computer (CORDIC) processor, a look-up table, and so on, or a combination thereof. A CORDIC processor implements an iterative algorithm that allows for fast hardware calculation of trigonometric functions such as sine, cosine, magnitude, and phase using simple shift and add/subtract hardware. A CORDIC processor may iteratively compute each of variables r , c_1 and s_1 in equation set (22), with more iterations producing higher accuracy for the variable.

[0063] FIG. 4 shows a block diagram of an access point 410 and a user terminal 450 in a MIMO system 400. Access point 410 is equipped with N_{ap} antennas and user terminal 450 is equipped with N_{ut} antennas, where $N_{ap} > 1$ and $N_{ut} > 1$. On the downlink, at access point 410, a transmit (TX) data processor 414 receives traffic data from a data source 412 and other data from a controller/processor 430. TX data processor 414 formats, encodes, interleaves, and modulates the data and generates data symbols, which are modulation symbols for data. A TX spatial processor 420 multiplexes the data symbols with pilot symbols, performs spatial processing with transmit matrix $\underline{\mathbf{W}}$ if applicable, and provides N_{ap} streams of transmit symbols. Each transmitter unit (TMTR) 422 processes a respective transmit symbol stream and generates a downlink modulated signal. N_{ap} downlink modulated signals from transmitter units 422a through 422ap are transmitted from antennas 424a through 424ap, respectively.

[0064] At user terminal 450, N_{ut} antennas 452a through 452ut receive the transmitted downlink modulated signals, and each antenna provides a received signal to a respective receiver unit (RCVR) 454. Each receiver unit 454 performs processing complementary to the processing performed by transmitter units 422 and provides

received pilot symbols and received data symbols. A channel estimator/processor 478 processes the received pilot symbols and provides an estimate of the downlink channel response $\underline{\mathbf{H}}_{dn}$. A processor 480 derives a downlink spatial filter matrix $\underline{\mathbf{M}}_{dn}$ based on $\underline{\mathbf{H}}_{dn}$ and using any of the embodiments described above. A receive (RX) spatial processor 460 performs receiver spatial processing (or spatial matched filtering) on the received data symbols from all N_{ut} receiver units 454a through 454ut with the downlink spatial filter matrix $\underline{\mathbf{M}}_{dn}$ and provides detected data symbols, which are estimates of the data symbols transmitted by access point 410. An RX data processor 470 processes (e.g., symbol demaps, deinterleaves, and decodes) the detected data symbols and provides decoded data to a data sink 472 and/or controller 480.

[0065] The processing for the uplink may be the same or different from the processing for the downlink. Data from a data source 486 and signaling from controller 480 are processed (e.g., encoded, interleaved, and modulated) by a TX data processor 488, multiplexed with pilot symbols, and possibly spatially processed by TX spatial processor 490. The transmit symbols from TX spatial processor 490 are further processed by transmitter units 454a through 454ut to generate N_{ut} uplink modulated signals, which are transmitted via antennas 452a through 452ut.

[0066] At access point 410, the uplink modulated signals are received by antennas 424a through 424ap and processed by receiver units 422a through 422ap to generate received pilot symbols and received data symbols for the uplink transmission. A channel estimator/processor 428 processes the received pilot symbols and provides an estimate of the uplink channel response $\underline{\mathbf{H}}_{up}$. Processor 430 derives an uplink spatial filter matrix $\underline{\mathbf{M}}_{up}$ based on $\underline{\mathbf{H}}_{up}$ and using any of the embodiments described above. An RX spatial processor 440 performs receiver spatial processing on the received data symbols with the uplink spatial filter matrix $\underline{\mathbf{M}}_{up}$ and provides detected data symbols. An RX data processor 442 further processes the detected data symbols and provides decoded data to a data sink 444 and/or controller 430.

[0067] Controllers 430 and 480 control the operation at access point 410 and user terminal 450, respectively. Memory units 432 and 482 store data and program codes used by controllers 430 and 480, respectively.

[0068] The blocks in FIGS. 1 through 4 represent functional blocks that may be embodied in hardware (one or more devices), firmware (one or more devices), software

(one or more modules), or combinations thereof. For example, the filter weight computation techniques described herein may be implemented in hardware, firmware, software, or a combination thereof. For a hardware implementation, the processing units used to compute the filter weights may be implemented within one or more application specific integrated circuits (ASICs), digital signal processors (DSPs), digital signal processing devices (DSPDs), programmable logic devices (PLDs), field programmable gate arrays (FPGAs), processors, controllers, micro-controllers, microprocessors, electronic devices, other electronic units designed to perform the functions described herein, or a combination thereof. The various processors at access point 410 in FIG. 4 may also be implemented with one or more hardware processors. Likewise, the various processors at user terminal 450 may be implemented with one or more hardware processors.

[0069] For a firmware or software implementation, the filter weight computation techniques may be implemented with modules (e.g., procedures, functions, and so on) that perform the functions described herein. The software codes may be stored in a memory unit (e.g., memory unit 432 or 482 in FIG. 4) and executed by a processor (e.g., processor 430 or 480). The memory unit may be implemented within the processor or external to the processor.

[0070] The previous description of the disclosed embodiments is provided to enable any person skilled in the art to make or use the present invention. Various modifications to these embodiments will be readily apparent to those skilled in the art, and the generic principles defined herein may be applied to other embodiments without departing from the spirit or scope of the invention. Thus, the present invention is not intended to be limited to the embodiments shown herein but is to be accorded the widest scope consistent with the principles and novel features disclosed herein.

[0071] WHAT IS CLAIMED IS:

CLAIMS

1. An apparatus comprising:
a first processor operative to derive a channel response matrix; and
a second processor operative to derive a first matrix iteratively based on the channel response matrix and to derive a spatial filter matrix based on the first matrix and the channel response matrix, wherein the second processor indirectly calculates a matrix inversion by deriving the first matrix iteratively.
2. The apparatus of claim 1, wherein the second processor is operative to initialize the first matrix to an identity matrix.
3. The apparatus of claim 1, wherein the second processor is operative, for each of a plurality of iterations, to derive an intermediate row vector based on the first matrix and a channel response row vector corresponding to a row of the channel response matrix, to derive a scalar based on the intermediate row vector and the channel response row vector, to derive an intermediate matrix based on the intermediate row vector, and to update the first matrix based on the scalar and the intermediate matrix.
4. The apparatus of claim 1, wherein the first matrix is for a minimum mean square error (MMSE) spatial filter matrix.
5. The apparatus of claim 1, wherein the second processor is operative to derive the first matrix based on the following equation:

$$\underline{\mathbf{P}}_i = \underline{\mathbf{P}}_{i-1} - \frac{\underline{\mathbf{P}}_{i-1} \cdot \underline{\mathbf{h}}_i^H \cdot \underline{\mathbf{h}}_i \cdot \underline{\mathbf{P}}_{i-1}}{r_i},$$

where $\underline{\mathbf{P}}_i$ is the first matrix for i -th iteration, $\underline{\mathbf{h}}_i$ is i -th row of the channel response matrix, r_i is a scalar derived based on $\underline{\mathbf{h}}_i$ and $\underline{\mathbf{P}}_{i-1}$, and “ H ” is a conjugate transpose.

6. The apparatus of claim 1, wherein the second processor is operative to derive the first matrix based on the following equations:

$$\underline{\mathbf{a}}_i = \underline{\mathbf{h}}_i \cdot \underline{\mathbf{P}}_{i-1},$$

$$r_i = \sigma_n^2 + \underline{\mathbf{a}}_i \cdot \underline{\mathbf{h}}_i^H ,$$

$$\underline{\mathbf{C}}_i = \underline{\mathbf{a}}_i^H \cdot \underline{\mathbf{a}}_i , \text{ and}$$

$$\underline{\mathbf{P}}_i = \underline{\mathbf{P}}_{i-1} - r_i^{-1} \cdot \underline{\mathbf{C}}_i ,$$

where $\underline{\mathbf{P}}_i$ is the first matrix for i -th iteration, $\underline{\mathbf{h}}_i$ is i -th row of the channel response matrix, $\underline{\mathbf{a}}_i$ is an intermediate row vector for the i -th iteration, $\underline{\mathbf{C}}_i$ is an intermediate matrix for the i -th iteration, r_i is a scalar for the i -th iteration, σ_n^2 is noise variance, and “ H ” is a conjugate transpose.

7. The apparatus of claim 1, wherein the second processor is operative to derive the spatial filter matrix based on the following equation:

$$\underline{\mathbf{M}} = \underline{\mathbf{P}} \cdot \underline{\mathbf{H}}^H ,$$

where $\underline{\mathbf{M}}$ is the spatial filter matrix, $\underline{\mathbf{P}}$ is the first matrix, $\underline{\mathbf{H}}$ is the channel response matrix, and “ H ” is a conjugate transpose.

8. A method of deriving a spatial filter matrix, comprising:

deriving a first matrix iteratively based on a channel response matrix, wherein a matrix inversion is indirectly calculated by deriving the first matrix iteratively; and

deriving the spatial filter matrix based on the first matrix and the channel response matrix.

9. The method of claim 8, further comprising:

initializing the first matrix to an identity matrix.

10. The method of claim 8, wherein the deriving the first matrix comprises, for each of a plurality of iterations,

deriving an intermediate row vector based on the first matrix and a channel response row vector corresponding to a row of the channel response matrix,

deriving a scalar based on the intermediate row vector and the channel response row vector,

deriving an intermediate matrix based on the intermediate row vector, and

updating the first matrix based on the scalar and the intermediate matrix.

11. An apparatus comprising:

means for deriving a first matrix iteratively based on a channel response matrix, wherein a matrix inversion is indirectly calculated by deriving the first matrix iteratively; and

means for deriving a spatial filter matrix based on the first matrix and the channel response matrix.

12. The apparatus of claim 11, further comprising:

means for initializing the first matrix to an identity matrix.

13. The apparatus of claim 11, wherein the means for deriving the first matrix comprises, for each of a plurality of iterations,

means for deriving an intermediate row vector based on the first matrix and a channel response row vector corresponding to a row of the channel response matrix,

means for deriving a scalar based on the intermediate row vector and the channel response row vector,

means for deriving an intermediate matrix based on the intermediate row vector, and

means for updating the first matrix based on the scalar and the intermediate matrix.

14. An apparatus comprising:

a first processor operative to derive a channel response matrix; and

a second processor operative to perform a plurality of rotations to iteratively obtain a first matrix and a second matrix for a pseudo-inverse matrix of the channel response matrix and to derive a spatial filter matrix based on the first and second matrices.

15. The apparatus of claim 14, wherein the second processor is operative to initialize the first matrix to an identity matrix and to initialize the second matrix with all zeros.

16. The apparatus of claim 14, wherein the second processor is operative, for each of a plurality of rows of the channel response matrix, to form an intermediate matrix based on the first matrix, the second matrix, and a channel response row vector, and to perform at least two rotations on the intermediate matrix to zero out at least two elements of the intermediate matrix.

17. The apparatus of claim 14, wherein the second processor is operative to perform a Givens rotation for each of the plurality of rotations to zero out one element of an intermediate matrix containing the first and second matrices.

18. The apparatus of claim 14, wherein the pseudo-inverse matrix is for a minimum mean square error (MMSE) spatial filter matrix.

19. The apparatus of claim 14, wherein the second processor is operative to perform at least two rotations for each of a plurality of iterations based on the following equation:

$$\begin{bmatrix} 1 & \underline{\mathbf{h}}_i \cdot \underline{\mathbf{P}}_{i-1}^{1/2} \\ \underline{\mathbf{0}} & \underline{\mathbf{P}}_{i-1}^{1/2} \\ -\underline{\mathbf{e}}_i & \underline{\mathbf{B}}_{i-1} \end{bmatrix} \cdot \underline{\Theta}_i = \begin{bmatrix} r_i^{1/2} & \underline{\mathbf{0}} \\ \underline{\mathbf{k}}_i & \underline{\mathbf{P}}_i^{1/2} \\ \underline{\mathbf{l}}_i & \underline{\mathbf{B}}_i \end{bmatrix},$$

where $\underline{\mathbf{P}}_i^{1/2}$ is the first matrix for i -th iteration, $\underline{\mathbf{B}}_i$ is the second matrix for the i -th iteration, $\underline{\mathbf{h}}_i$ is i -th row of the channel response matrix, $\underline{\mathbf{e}}_i$ is a vector with one for i -th element and zeros elsewhere, $\underline{\mathbf{k}}_i$ and $\underline{\mathbf{l}}_i$ are non-essential vectors, $r_i^{1/2}$ is a scalar, $\underline{\mathbf{0}}$ is a vector with all zeros, and $\underline{\Theta}_i$ is a transformation matrix representing the at least two rotations for the i -th iteration.

20. The apparatus of claim 14, wherein the second processor is operative to derive the spatial filter matrix based on the following equation:

$$\underline{\mathbf{M}} = \underline{\mathbf{P}}^{1/2} \cdot \underline{\mathbf{B}}^H,$$

where $\underline{\mathbf{M}}$ is the spatial filter matrix, $\underline{\mathbf{P}}^{1/2}$ is the first matrix, $\underline{\mathbf{B}}$ is the second matrix, and “ H ” is a conjugate transpose.

21. A method of deriving a spatial filter matrix, comprising:
performing a plurality of rotations to iteratively obtain a first matrix and a second matrix for a pseudo-inverse matrix of a channel response matrix; and
deriving the spatial filter matrix based on the first and second matrices.

22. The method of claim 21, wherein the performing the plurality of rotations comprises, for each of a plurality of iterations,
forming an intermediate matrix based on the first matrix, the second matrix, and a channel response row vector corresponding to a row of the channel response matrix, and
performing at least two rotations on the intermediate matrix to zero out at least two elements of the intermediate matrix.

23. The method of claim 21, wherein the performing the plurality of rotations comprises
performing a Givens rotation for each of the plurality of rotations to zero out one element of an intermediate matrix containing the first and second matrices.

24. An apparatus comprising:
means for performing a plurality of rotations to iteratively obtain a first matrix and a second matrix for a pseudo-inverse matrix of a channel response matrix; and
means for deriving a spatial filter matrix based on the first and second matrices.

25. The apparatus of claim 24, wherein the means for performing the plurality of rotations comprises, for each of a plurality of iterations,
means for forming an intermediate matrix based on the first matrix, the second matrix, and a channel response row vector corresponding to a row of the channel response matrix, and
means for performing at least two rotations on the intermediate matrix to zero out at least two elements of the intermediate matrix.

26. The apparatus of claim 24, wherein the means for performing the plurality of rotations comprises

means for performing a Givens rotation for each of the plurality of rotations to zero out one element of an intermediate matrix containing the first and second matrices.

27. An apparatus comprising:

a first processor operative to derive a channel response matrix; and

a second processor operative to derive a first matrix based on the channel response matrix, to decompose the first matrix to obtain a unitary matrix and a diagonal matrix, and to derive the spatial filter matrix based on the unitary matrix, the diagonal matrix, and the channel response matrix.

28. The apparatus of claim 27, wherein the second processor is operative to perform eigenvalue decomposition of the first matrix to obtain the unitary matrix and the diagonal matrix.

29. The apparatus of claim 27, wherein the second processor is operative to perform a plurality of Jacobi rotations on the first matrix to obtain the unitary matrix and the diagonal matrix.

30. The apparatus of claim 27, wherein the second processor is operative to derive the first matrix based on the following equation:

$$\underline{\mathbf{X}} = \sigma_n^2 \cdot \underline{\mathbf{I}} + \underline{\mathbf{H}}^H \cdot \underline{\mathbf{H}} ,$$

where $\underline{\mathbf{X}}$ is the first matrix, $\underline{\mathbf{H}}$ is the channel response matrix, $\underline{\mathbf{I}}$ is an identity matrix, σ_n^2 is noise variance, and “ H ” is a conjugate transpose.

31. The apparatus of claim 27, wherein the second processor is operative to derive the spatial filter matrix based on the following equation:

$$\underline{\mathbf{M}} = \underline{\mathbf{V}} \cdot \underline{\mathbf{\Lambda}}^{-1} \cdot \underline{\mathbf{V}}^H \cdot \underline{\mathbf{H}}^H ,$$

where $\underline{\mathbf{M}}$ is the spatial filter matrix, $\underline{\mathbf{H}}$ is the channel response matrix, $\underline{\mathbf{V}}$ is the unitary matrix, $\underline{\mathbf{\Lambda}}$ is the diagonal matrix, and “ H ” is a conjugate transpose.

32. A method of deriving a spatial filter matrix, comprising:

deriving a first matrix based on a channel response matrix;
decomposing the first matrix to obtain a unitary matrix and a diagonal matrix;
and
deriving the spatial filter matrix based on the unitary matrix, the diagonal matrix,
and the channel response matrix.

33. The method of claim 32, wherein the decomposing the first matrix comprises

performing eigenvalue decomposition of the first matrix to obtain the unitary matrix and the diagonal matrix.

34. The method of claim 32, wherein the decomposing the first matrix comprises

performing a plurality of Jacobi rotations on the first matrix to obtain the unitary matrix and the diagonal matrix.

35. An apparatus comprising:

means for deriving a first matrix based on a channel response matrix;

means for decomposing the first matrix to obtain a unitary matrix and a diagonal matrix; and

means for deriving a spatial filter matrix based on the unitary matrix, the diagonal matrix, and the channel response matrix.

36. The apparatus of claim 35, wherein the means for decomposing the first matrix comprises

means for performing eigenvalue decomposition of the first matrix to obtain the unitary matrix and the diagonal matrix.

37. The apparatus of claim 35, wherein the means for decomposing the first matrix comprises

means for performing a plurality of Jacobi rotations on the first matrix to obtain the unitary matrix and the diagonal matrix.

1/4

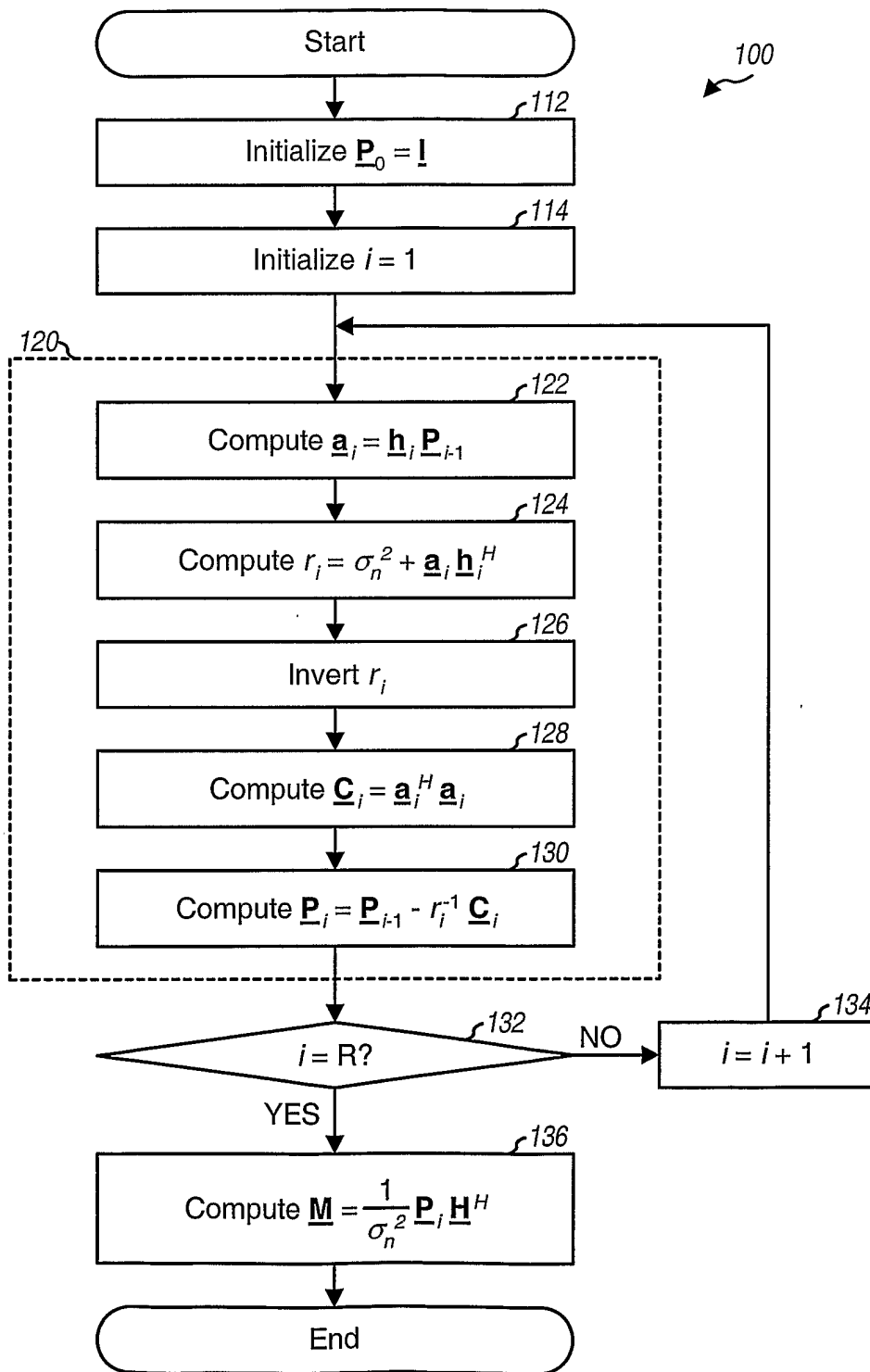


FIG. 1

2/4

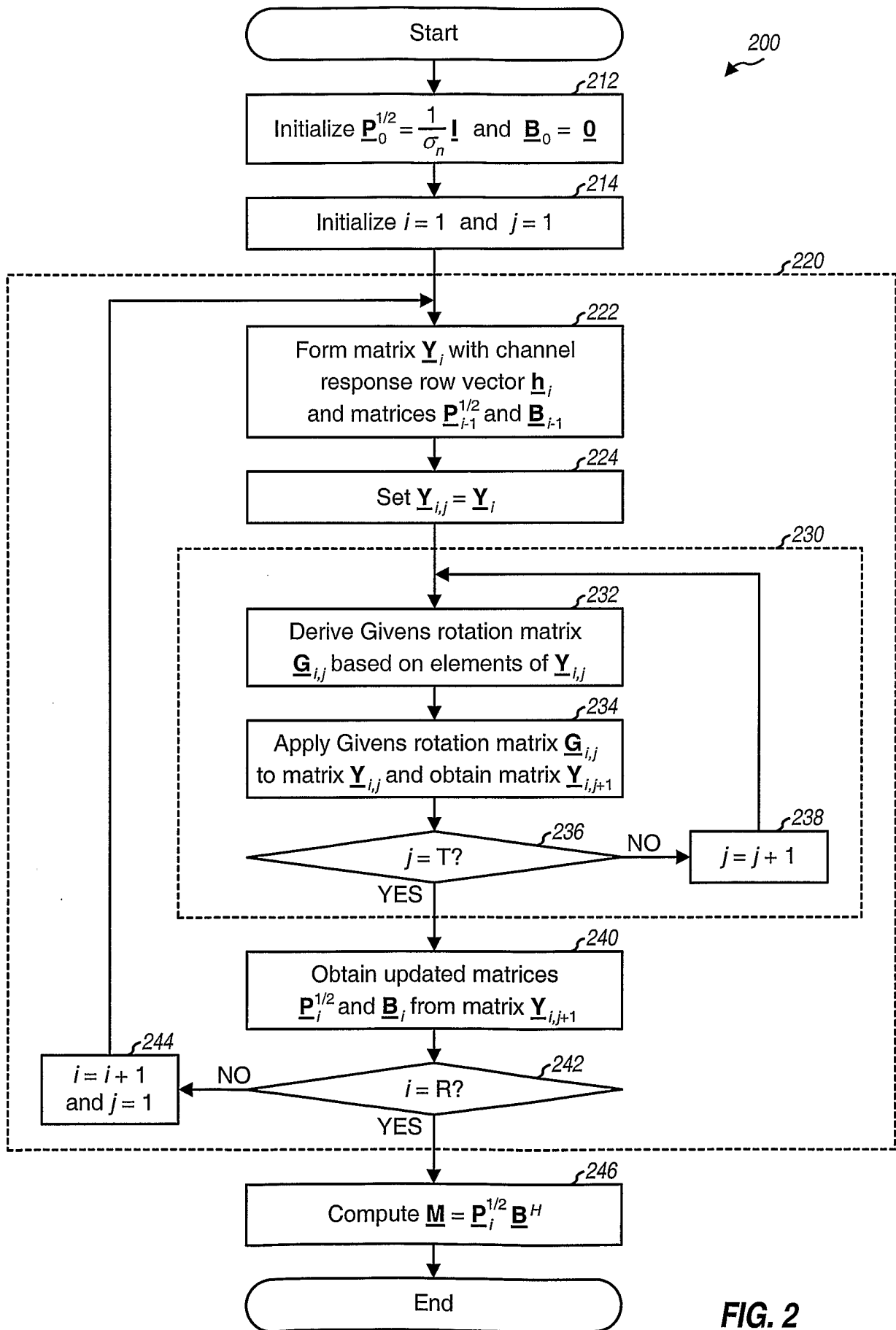


FIG. 2

3/4

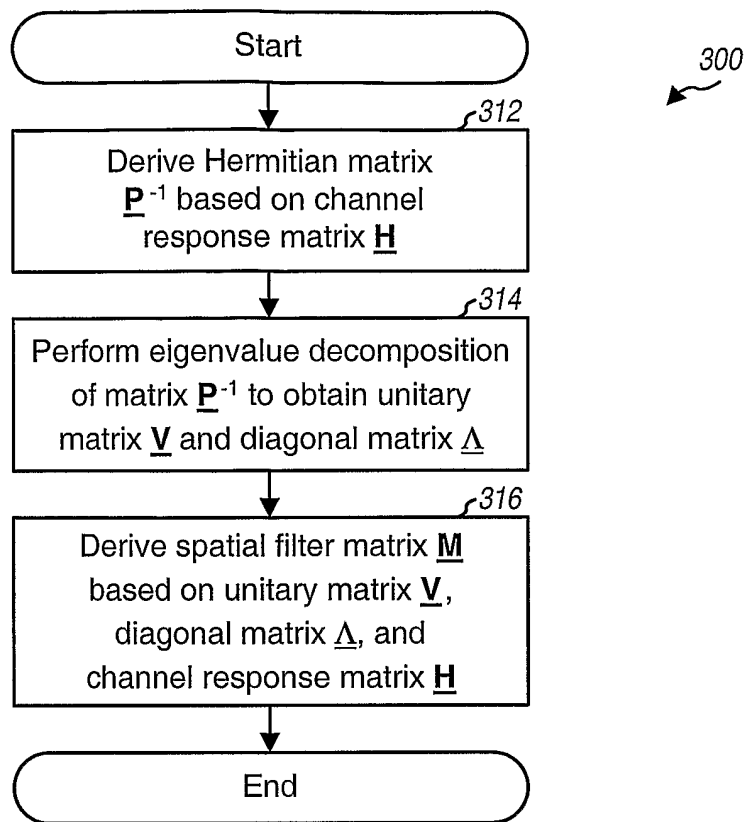


FIG. 3

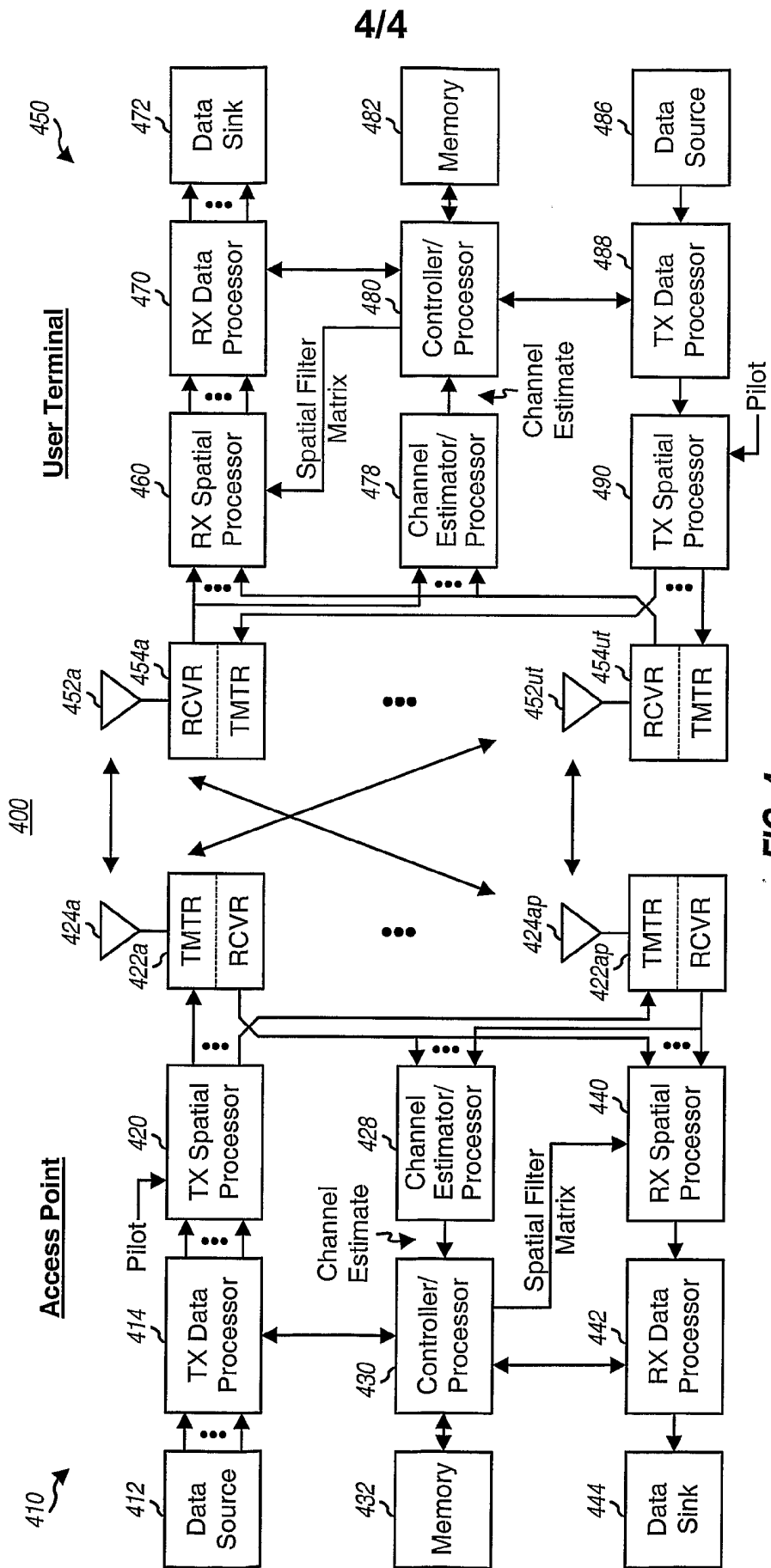


FIG. 4