

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2012-165468
(P2012-165468A)

(43) 公開日 平成24年8月30日(2012.8.30)

(51) Int.Cl. F I テーマコード (参考)
HO4J 13/00 (2011.01) HO4J 13/00 100

審査請求 有 請求項の数 18 O L (全 28 頁)

(21) 出願番号 特願2012-118591 (P2012-118591)
(22) 出願日 平成24年5月24日 (2012.5.24)
(62) 分割の表示 特願2009-149732 (P2009-149732)
の分割
原出願日 平成13年11月8日 (2001.11.8)
(31) 優先権主張番号 60/246,947
(32) 優先日 平成12年11月9日 (2000.11.9)
(33) 優先権主張国 米国 (US)

(71) 出願人 596008622
インターデジタル テクノロジー コー
ポレーション
アメリカ合衆国 19810 デラウェア
州 ウィルミントン シルバーサイド ロ
ード 3411 コンコルド プラザ ヘ
イグリー ビルディング スイート 10
5
(74) 代理人 110001243
特許業務法人 谷・阿部特許事務所
(72) 発明者 デ, パルタプラティム
アメリカ合衆国 ニューヨーク州 117
35 ファーミングデイル, メルヴィル
ロード 1233-8, アpartment
8

最終頁に続く

(54) 【発明の名称】 単一ユーザ検出

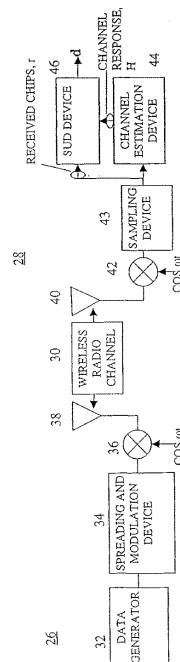
(57) 【要約】

【課題】 TDD/CDMA通信方式など共用周波数スペクトラムによる多重通信のユーザ装置を単純化する。

【解決手段】 CDMA通信システムにおいて共用周波数スペクトラム経由で送信側から複数のデータ信号を送信する。これら送信されたデータ信号は相似のチャネル応答を受ける。

受信側では送信されてきたデータ信号の合成信号を受信する。この合成信号をチップ速度の倍数の速度でサンプリングする。合成信号に対するチャネル応答を算出する。合成信号サンプル値とチャネル応答推算値とを用いて拡散データベクトルの第1の要素を算定する。この第1の要素の算定からの係数を用いて、拡散データベクトルの残余の要素を算定する。この拡散データベクトルの算定済みの要素を用いてデータ信号のデータを算定する。

【選択図】 図2



【特許請求の範囲】

【請求項 1】

符号分割多元接続通信を処理するための方法であって、
 通信局からチップ速度で同時に送信された複数の異なるデータ信号の合成信号を共用スペクトラム経路で受信するステップと、
 前記チップ速度の倍数の速度で前記合成信号をサンプリングするステップと、
 前記合成信号に対するチャンネル応答を前記チップ速度の倍数の速度で推算するステップと、
 前記合成信号のサンプルと前記推算されたチャンネル応答を用いて拡散データベクトルを算定するステップと、
 前記拡散データベクトルを用いて前記異なるデータ信号のそれぞれのデータを推算するステップと
 を備えたことを特徴とする方法。

10

【請求項 2】

拡散データベクトルを算定する前記ステップは、
 前記合成信号のサンプルと前記推算されたチャンネル応答とを用いて拡散データベクトルの第 1 の要素を算定するステップ
 前記第 1 の要素の算定からの係数を用いて前記拡散データベクトルの残余の要素を算定するステップ
 を含むことを特徴とする請求項 1 に記載の方法。

20

【請求項 3】

前記データの推算は、前記拡散データベクトルの逆拡散によって行われることを特徴とする請求項 1 に記載の方法。

【請求項 4】

拡散データベクトルを算定する前記ステップは、
 前記推算されたチャンネル応答を用いて相互相関行列を算定するステップと、
 前記相互相関行列の部分ブロックを選択するステップと、
 前記部分ブロックについてコレスキー係数を算定するステップと、
 前記コレスキー係数を拡張するステップと、
 前記拡張されたコレスキー係数を用いて前記拡散データベクトルを算定するステップと
 を含むことを特徴とする請求項 1 に記載の方法。

30

【請求項 5】

拡散データベクトルを算定する前記ステップは、
 前記推算されたチャンネル応答を用いて相互相関行列を算定するステップ
 を含むことを特徴とする請求項 1 に記載の方法。

【請求項 6】

拡散データベクトルを算定する前記ステップは、
 前記推算されたチャンネル応答を用いてチャンネル相関行列の列を算定するステップと、
 前記算定した列、前記推算されたチャンネル応答、前記受信した合成信号及びフーリエ変換を用いて前記拡散データベクトルを算定するステップと
 を含むことを特徴とする請求項 1 に記載の方法。

40

【請求項 7】

拡散データベクトルを算定する前記ステップは、
 前記倍数の速度でサンプリングしたサンプルを実効チップ速度サンプルとして合成するステップと、
 前記倍数の速度で推算したチャンネル応答を実効チップ速度チャンネル応答として合成するステップと、
 前記実効チップ速度サンプル、前記実効チップ速度チャンネル応答及びフーリエ変換を用いて拡散データベクトルを算定するステップと
 を含むことを特徴とする請求項 1 に記載の方法。

50

【請求項 8】

前記チャンネル応答を推算するステップは、前記チップ速度の倍数の速度で前記合成信号に対するチャンネル応答行列として推算し、拡散データベクトルを算定する前記ステップは、

前記チャンネル応答行列の列、前記推算されたチャンネル応答行列、前記サンプル及びフーリエ変換を用いて前記倍数のチップ速度に対応する大きさの拡散データベクトルの修正バージョンを算定するステップと、

前記推算された拡散データベクトルが前記チップ速度に対応する大きさになるように前記修正バージョンの要素を消去することによって前記拡散データベクトルを推算するステップと

10

を含むことを特徴とする請求項 1 に記載の方法。

【請求項 9】

符号分割多元接続通信のための無線通信装置であって、

チップ速度で同時に送信された複数の異なるデータ信号の合成信号を共用スペクトラム経路で受信するように構成された受信機と、

前記チップ速度の倍数の速度で前記合成信号をサンプルするように構成されたサンプリング装置と、

前記合成信号に対するチャンネル応答を前記チップ速度の倍数の速度で推算するように構成されたチャンネル応答推算回路と、

前記合成信号のサンプルと前記推算されたチャンネル応答を用いて拡散データベクトルを算定するように構成された拡散データベクトル算定回路と、

20

前記拡散データベクトルを用いて前記データ信号のデータを推算するように構成されたデータ信号推算回路と

を備えたことを特徴とする通信装置。

【請求項 10】

前記拡散データベクトル算定回路は、前記合成信号のサンプルと前記推算されたチャンネル応答とを用いて拡散データベクトルの第 1 の要素を算定すること、及び前記第 1 の要素の算定からの係数を用いて前記拡散データベクトルの残余の要素を算定することにより、拡散データベクトルを算定するように構成されていることを特徴とする請求項 9 に記載の通信装置。

30

【請求項 11】

前記データ信号推算回路は、前記拡散データベクトルの逆拡散によってデータを推算するように構成されていることを特徴とする請求項 9 に記載の通信装置。

【請求項 12】

前記拡散データベクトル算定回路は、前記推算されたチャンネル応答を用いて相互相関行列を算定し、前記相互相関行列の部分ブロックを選択し、前記部分ブロックについてコレスキー係数を算定し、前記コレスキー係数を拡張し、前記拡張されたコレスキー係数を用いて前記拡散データベクトルを算定するように構成されていることを特徴とする請求項 9 に記載の通信装置。

40

【請求項 13】

前記拡散データベクトル算定回路は、前記推算されたチャンネル応答を用いて相互相関行列を算定するように構成されていることを特徴とする請求項 9 に記載の通信装置。

【請求項 14】

前記拡散データベクトル算定回路は、前記推算されたチャンネル応答を用いてチャンネル相関行列の列を算定し、前記算定した列、前記推算されたチャンネル応答、前記受信した合成信号及びフーリエ変換を用いて前記拡散データベクトルを算定するように構成されていることを特徴とする請求項 9 に記載の通信装置。

【請求項 15】

前記拡散データベクトル算定回路は、前記倍数の速度でサンプリングしたサンプルを実効チップ速度サンプルとして合成し、前記倍数の速度で推算したチャンネル応答を実効チッ

50

ブ速度チャンネル応答として合成し、前記実効チップ速度サンプル、前記実効チップ速度チャンネル応答及びフーリエ変換を用いて拡散データベクトルを算定するように構成されていることを特徴とする請求項 9 に記載の通信装置。

【請求項 16】

前記チャンネル応答推算回路は、前記チップ速度の倍数の速度で前記合成信号に対するチャンネル応答行列として前記チャンネル応答を推算するように構成されており、

前記拡散データベクトル算定回路は、前記チャンネル応答行列の列、前記推算されたチャンネル応答行列、前記サンプル及びフーリエ変換を用いて前記倍数のチップ速度に対応する大きさの拡散データベクトルの修正バージョンを算定し、前記推算された拡散データベクトルが前記チップ速度に対応する大きさになるように前記修正バージョンの要素を消去することによって前記拡散データベクトルを推算するように構成されていることを特徴とする請求項 9 に記載の通信装置。

10

【請求項 17】

第三代パートナシッププロジェクト(3GPP)のユニバーサル地上無線接続(UTRA)システムのための基地局として構成されていることを特徴とする請求項 9 から 16 のいずれか 1 つに記載の通信装置。

【請求項 18】

第三代パートナシッププロジェクト(3GPP)のユニバーサル地上無線接続(UTRA)システムのためのユーザ装置

【発明の詳細な説明】

20

【技術分野】

【0001】

この発明は概括的には無線通信システムに関する。さらに詳細にいうと、この発明は無線通信システムにおけるデータ検出に関する。

【背景技術】

【0002】

図 1 は無線通信システム 10 を示す。無線通信システム 10 はユーザ装置(UEs) 14₁ ~ 14₃(14)と通信する基地局 12₁ ~ 12₅(12)を有する。各基地局には対応する稼働範囲があり、その範囲内で基地局はユーザ装置 UEs 14 と通信する。

【0003】

30

幾つかの通信システム、例えば符号分割多元接続(CDMA)方式や符号分割多元接続を用いる時分割複信(TDD/CDMA)方式の通信システムでは、多重通信が同一の周波数スペクトラムでなされる。これらの多重通信信号はチャンネル毎に互いに異なる符号を用いることで区別されている。更に効率的に無線周波数スペクトラムを使用するため、TDD/CDMA 通信システムでは、通信用に複数の時間スロットに分割された繰返しフレームを使用する。このようなシステムで送られる通信信号では、それぞれが一つの或いは複数の対応する符号とそれに割当てられた時間スロットを有する。一つの時間スロットでの一つの符号使用は資源ユニットと呼ばれる。

【先行技術文献】

【非特許文献】

40

【0004】

【非特許文献 1】IEEE International Symposium on Personal, Indoor and Mobile RadioCommunications XX, DD vol.3, 1998, 1340 頁乃至 1345 頁

【非特許文献 2】IEEE Transactions on Information Theory, March 1994, vol.40, No. 2, 340 頁乃至 349 頁

【非特許文献 3】International Annual Conference of ICT, XX, XX, 1999, 1 頁乃至 7 頁

【発明の概要】

【発明が解決しようとする課題】

【0005】

50

多重通信は同一の無線周波数スペクトラムで同時になされることから、このような通信システムにおける受信機は多数の通信信号を相互に区別する必要がある。このような信号の検出のための一つの手法は多数ユーザ検出法である。多数ユーザ検出法では、全てのユーザ、すなわちユーザ装置 UEs 14 と対応する信号を同時に検出する。多数ユーザ検出を実行する手法は、コレスキー (Cholesky) 分解或いは近似コレスキー分解を用いる同時検出を基にするブロック線形等化 (BLE-JD) を含む。これらの検出手段は高度の複雑さを伴っている。この複雑さは電力消費を増加させ、ひいてはユーザ装置 UEs 14 の電池寿命の低下をもたらす。そこで、受信データを検出する代替手段の開発が求められている。

【課題を解決するための手段】

【0006】

符号分割多元接続通信システムにおいては、送信側は複数のデータ信号を共用周波数スペクトラム経由で送信する。送信されるデータ信号の各々は同様のチャネル応答を受ける。受信側では、それら送信されてきたデータ信号の合成信号が受信される。その合成信号をチップ速度のある倍数の速度でサンプリングする。これにより、受信合成信号に対する共通のチャネル応答を推算する。次に、拡散データベクトルの第1の要素を、上記合成信号のサンプル値とチャネル応答推算値とを使用して算定する。第1のベクトル要素の算定からの係数を用いて、拡散データベクトルの残りのベクトル要素も算定する。データ信号のデータを、算定済みの拡散データベクトル要素を用いて算定する。

【発明の効果】

【0007】

CDMA方式と共用周波数スペクトラムによる多重通信の受信装置を単純化し、受信装置の小型化および消費電力の低減を可能にする。

【図面の簡単な説明】

【0008】

【図1】稼働範囲毎に配置される基地局とユーザ装置からなる無線通信システムの説明図

【図2】単純化して図示した送信機と単一ユーザの受信機

【図3】通信バーストの説明図

【図4】単一ユーザ検出 (SUD) のための拡張前進代入手法の流れ図

【図5】単一ユーザ検出 (SUD) のための帯状近似コレスキー手法の流れ図

【図6】単一ユーザ検出 (SUD) のためのトプリッツ手法の流れ図

【図7】単一ユーザ検出 (SUD) のため、チャネル相関行列に適用される高速フーリエ変換 (FFT) 手法の流れ図

【図8】効果的な合成方法を用いる単一ユーザ検出 (SUD) のための高速フーリエ変換 (FFT) 手法の流れ図

【図9】零埋め込みを用いる単一ユーザ検出 (SUD) のための高速フーリエ変換 (FFT) 手法の流れ図

【発明を実施するための形態】

【実施例1】

【0009】

図2には TDD / CDMA 通信システムにおける送信機 26 と単一ユーザ検出 (SUD) 手法を用いた受信機 28 とを単純化した形で示す。単一ユーザ検出手法は他の通信システム、周波数分割複信 (FDD) CDMA 方式などの通信システムにも適用できる。通常の通信システムにおいては、送信機 26 は各ユーザ装置 UE 14 内にあり、多重通信を行う多重送回路 26 は各基地局 12 内にある。SUD 受信機 28 は基地局 12、複数の UE 14、またはこれらの両者に配置できる。通常、SUD 手法はある特定の送信機からの単一符号或いは多符号伝送時におけるデータを検出するのに用いられる。全ての伝送信号が同一の送信機から送られる場合、個々のチャネル符号信号の各々は、多符号伝送では同一のチャネルインパルス応答を受ける。SUD 手法は特にダウンリンク、すなわち全ての送信信号が一つの基地局の一つのアンテナ或いはアンテナアレイから輻射されるダウンリンクに有用である。この手法は、単一ユーザが単一符号伝送または多符号伝送で一つのスロットを

10

20

30

40

50

占有するアップリンクにも有用である。

【 0 0 1 0 】

送信機 2 6 は無線伝送チャネル 3 0 経由でデータを送る。受信機 2 6 内のデータ発生器 3 2 は、受信機 2 8 に送信すべきデータを生ずる。変調 / 拡散系列挿入装置 3 4 はデータをスペクトラム拡散し、この拡散ずみ基準データを適切な割当て時間スロット内でミドアンブルトレーニング系列およびデータ拡散用の符号と時分割多重化し、一つの或いは複数の通信信号バーストを発生させる。

【 0 0 1 1 】

通常の通信信号バースト 1 6 は図 3 に示すとおり、一つのミドアンブル 2 0、一つのガード期間 1 8 および二つのデータ領域 2 2, 2 4 から成る。ミドアンブル 2 0 は二つのデータ領域を分離し、ガード期間 1 8 は、互いに異なる送信機 2 6 からのバーストの到着時間の差を許容するように通信信号バーストを分離している。二つのデータ領域 2 2、2 4 は通信信号バーストデータを含む。

10

【 0 0 1 2 】

これらの通信信号バーストで変調器 3 6 において無線周波数 (R F) を変調する。 R F 信号はアンテナ 3 8 により輻射され、無線伝送チャネル 3 0 経由で受信機 2 8 のアンテナに達する。この通信信号送信に用いられる変調方式は、直角位相偏位変調 (Q P S K) や M 次直角位相変調 (Q A M) など当業者に周知の方式のどれであっても差し支えない。

【 0 0 1 3 】

受信機 2 8 のアンテナ 4 0 は種々の R F 信号を受信する。受信された R F 信号を復調器 4 2 で復調してベースバンド信号を再生する。このベースバンド信号を、単一型または複合型の A - D 変換器などのサンプリング装置 4 3 によって、送信バースト信号のチップ速度で、あるいはそのチップ速度のある倍数の速さでサンプリングする。サンプリングされた出力を、チャンネル推算装置 4 4 や S U D 装置 4 6 などの装置により、受信信号バーストに割当てられたその時間スロット内に適切な符号で処理される。チャンネル推算装置 4 4 は、ベースバンドサンプルの中のミドアンブルトレーニング系列コンポーネントを用い、チャンネルインパルス応答などのチャンネル情報を提供する。このチャンネルインパルス応答は行列 H として表わされる。このチャンネル情報は S U D 装置 4 6 で用いられ、それにより、ソフトシンボル受信通信バーストの中の送信されてきたデータが推算される。

20

【 0 0 1 4 】

S U D 装置 4 6 は、チャンネル推算装置 4 4 からのチャンネル情報および送信機 2 6 の用いた既知の拡散符号を使用して、所望の受信した通信バーストのデータを推算する。ここで、S U D は、第三代パートナシッププロジェクト (3 G P P) のユニバーサル地上無線接続 (U T R A) T D D システムを背景の通信システムとして説明するが、この S U D は他のシステムにも適用できる。そのシステムとは、直接拡散方式の広帯域 C D M A (W - C D M A) システム、すなわちアップリンク信号およびダウンリンク伝送信号を互いに別々の時間スロットに制限した広帯域 C D M A (W - C D M A) である。

30

受信機 2 8 はそのアンテナ 4 0 を用いて同時到達の合計で K 個のバースト信号を受信する (4 8) 。これら K 個のバースト信号を一つの観察間隔内で互いに重畳させる。3 G P P U T R A T D D システムについては、時間スロットの各データ領域が一つの観察間隔に対応する。

40

【 0 0 1 5 】

一つの観察間隔に対し、データ検出課題は式 1 により表わされる。

【 0 0 1 6 】

【 数 1 】

$$r = H \cdot d + n$$

Equation 1

【 0 0 1 7 】

ここで、r は受信したサンプル値で、H はチャンネル応答行列、d は拡散データベクトルで

50

ある。拡散データ行列は、各チャネルで伝送されてきたデータとそのチャネルの拡散符号とを含んでいる。

【 0 0 1 8 】

受信信号をオーバーサンプリングした場合は、各送信チップの複数サンプル値を生じ、結果として受信ベクトル r_1, \dots, r_N を生ずる(48)。同様に、チャネル推算装置44は、受信ベクトル r_1, \dots, r_N に対応するチャネル応答 H_1, \dots, H_N を算定する(50)。チップ速度の2倍でのサンプリングでは、式1は式2に書き替えられる。

【 0 0 1 9 】

【 数 2 】

$$\begin{bmatrix} r_1 \\ r_2 \end{bmatrix} = \begin{bmatrix} H_1 \\ H_2 \end{bmatrix} \cdot \mathbf{d} + \mathbf{n}$$

Equation 2

10

【 0 0 2 0 】

ここで、 r_1 は偶数番目のサンプル(チップ速度でのサンプリングに対応)で、 r_2 は奇数番目のサンプル(r_1 サンプルからチップ間隔の二分の一離れた所でのサンプリングに対応)である。 H_1 は奇数番目のサンプルに対するチャネル応答で、 H_2 は偶数番目に対するその応答である。

【 0 0 2 1 】

式1は、チップ速度のN倍の速度でのサンプリングでは、式3となる。

【 0 0 2 2 】

【 数 3 】

$$\begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_N \end{bmatrix} = \begin{bmatrix} H_1 \\ H_2 \\ \vdots \\ H_N \end{bmatrix} \cdot \mathbf{d} + \mathbf{n}$$

Equation 3

20

30

【 0 0 2 3 】

ここで、 r_1, r_2, \dots, r_N は、チップ速度のそれぞれの倍数でのサンプルである。各サンプルはチップ間隔のN分の1だけずれている。 H_1, H_2, \dots, H_N は対応するチャネル応答である。以下の説明はチップ速度の2倍の速度でのサンプリングを実行する受信機に絞るが、同様な手法はチップ速度の任意の倍数でのサンプリングに対しても適用できる。

【 0 0 2 4 】

チップ速度の2倍の速度でのサンプリングについて、チャネル応答行列 H_1 および H_2 の大きさは $(Ns+W-1) \times Ns$ となる。ここで、 Ns は観察期間内に送信される拡散チップの数であり、 W はチャネルインパルス応答の長さであり、例えば長さ57チップと表わす。受信信号は Ns 個の拡散チップを有するので、 r_1 及び r_2 の長さは Ns である。式2は式4のように書き替えられる。

【 0 0 2 5 】

40

【数 4】

$$\begin{array}{l}
 r_1(0) \\
 r_1(1) \\
 | \\
 r_1(W-1) \\
 r_1(N_s-1) \\
 \text{-----} \\
 r_2(0) \\
 r_2(1) \\
 | \\
 r_2(W-1) \\
 r_2(N_s-1)
 \end{array}
 \left[\begin{array}{cccccccc}
 h_1(0) & 0 & 0 & \text{--} & \text{--} & \text{--} & \text{--} & \text{--} \\
 h_1(1) & h_1(0) & 0 & \text{--} & & & & \\
 | & | & & & & & & \\
 h_1(W-1) & h_1(W-2) & \text{--} & \text{--} & h_1(1) & h_1(0) & 0 & 0 \\
 | & | & & & | & | & | & | \\
 0 & 0 & \text{--} & h_1(W-1) & h_1(W-2) & \text{--} & h_1(1) & h_1(0) \\
 \text{--} & \text{--} & \text{--} & \text{--} & \text{--} & \text{--} & \text{--} & \text{--} \\
 h_2(0) & 0 & 0 & \text{--} & \text{--} & \text{--} & \text{--} & \text{--} \\
 h_2(1) & h_2(0) & 0 & \text{--} & & & & \\
 | & | & & & & & & \\
 h_2(W-1) & h_2(W-2) & \text{--} & \text{--} & h_2(1) & h_2(0) & 0 & 0 \\
 | & | & & & | & | & | & | \\
 0 & 0 & \text{--} & h_2(W-1) & h_2(W-2) & \text{--} & h_2(1) & h_2(0) \\
 \text{--} & \text{--} & \text{--} & \text{--} & \text{--} & \text{--} & \text{--} & \text{--}
 \end{array} \right] \cdot \mathbf{d+n}$$

Equation 4

【0026】

$r_1(i)$ 、 $r_2(i)$ 、 $h_1(i)$ 及び $h_2(i)$ は、それぞれ対応するベクトル行列 r_1 、 r_2 、 H_1 及び

H_2 の i 番目の要素である。

拡散データベクトルを算定する手法の一つは、拡張前進代入手法であり、その手法を図4を参照して説明する。拡張前進代入のために、受信したデータベクトルを再配列し、各偶数番目のサンプルの次にそれに対応する奇数番目のサンプルを伴うようにする。同様の再配列を式5aに示すとおりチャネル応答行列についても行う。

【0027】

【数 5 (a)】

$$\begin{array}{l}
 r_1(0) \\
 r_2(0) \\
 r_1(1) \\
 r_2(1) \\
 | \\
 r_1(W-1) \\
 r_2(W-1) \\
 | \\
 r_1(N_s-1) \\
 r_2(N_s-1)
 \end{array}
 \left[\begin{array}{cccccccc}
 h_1(0) & 0 & 0 & \text{--} & & & & \\
 h_2(0) & 0 & 0 & \text{--} & & & & \\
 h_1(1) & h_1(0) & 0 & \text{--} & & & & \\
 h_2(1) & h_2(0) & 0 & \text{--} & & & & \\
 | & | & | & \text{--} & & & & \\
 h_1(W-1) & h_1(W-2) & \text{--} & \text{--} & h_1(1) & h_1(0) & 0 & 0 \\
 h_2(W-1) & h_2(W-2) & \text{--} & \text{--} & h_2(1) & h_2(0) & 0 & 0 \\
 | & | & & & | & | & | & | \\
 0 & 0 & \text{--} & h_1(W-1) & h_1(W-2) & \text{--} & h_1(1) & h_1(0) \\
 0 & 0 & \text{--} & h_2(W-1) & h_2(W-2) & \text{--} & h_2(1) & h_2(0)
 \end{array} \right] \cdot \begin{array}{l}
 d(0) \\
 d(1) \\
 | \\
 | \\
 d(N_s-1)
 \end{array} + \mathbf{n}$$

【0028】

チップ速度の N 倍の速度でのサンプリングについては、式 5 b が配列となる。

【 0 0 2 9 】

【 数 5 (b) 】

$$\begin{bmatrix} r_1(0) \\ r_2(0) \\ | \\ r_N(0) \\ r_1(1) \\ r_2(1) \\ | \\ r_N(1) \\ | \\ r_1(N_s-1) \\ r_2(N_s-1) \\ | \\ r_N(N_s-1) \end{bmatrix} = \begin{bmatrix} h_1(0) & 0 & \dots & 0 & 0 \\ h_2(0) & 0 & \dots & 0 & 0 \\ | & | & \dots & | & | \\ h_N(0) & 0 & \dots & 0 & 0 \\ h_1(1) & h_1(0) & \dots & 0 & 0 \\ h_2(1) & h_2(0) & \dots & 0 & 0 \\ | & | & \dots & | & | \\ h_N(1) & h_N(0) & \dots & 0 & 0 \\ | & | & \dots & | & | \\ 0 & 0 & \dots & h_1(1) & h_1(0) \\ 0 & 0 & \dots & h_2(1) & h_2(0) \\ | & | & \dots & | & | \\ 0 & 0 & \dots & h_N(1) & h_N(0) \end{bmatrix} \cdot \begin{bmatrix} d(0) \\ d(1) \\ | \\ d(N_s-1) \end{bmatrix} + \mathbf{n} \tag{10}$$

20

Equation 5b

【 0 0 3 0 】

ここで、 $d(i)$ は拡散ベクトル d の i 番目の要素である。拡散データベクトルの長さは N_s である。拡張前代入手法を用いると、 $d(0)$ 、 $d^\wedge(0)$ を算定するためのゼロ強制解が式 6 a および式 7 a により表わされる (5 2)。式 6 a は

【 0 0 3 1 】

【 数 6 (a) 】

$$\begin{bmatrix} h_1(0) \\ h_2(0) \end{bmatrix} \cdot d(0) = \begin{bmatrix} r_1(0) \\ r_2(0) \end{bmatrix} \tag{Equation 6a} \tag{30}$$

【 0 0 3 2 】

$d(0)$ についての一般式である。また、式 7 a は $d^\wedge(0)$ についての零強制解である。

【 0 0 3 3 】

【 数 7 (a) 】

$$d^\wedge(0) = \left\{ \begin{bmatrix} h_1(0) & h_2(0) \end{bmatrix} \begin{bmatrix} h_1(0) \\ h_2(0) \end{bmatrix} \right\}^{-1} \begin{bmatrix} h_1(0) & h_2(0) \end{bmatrix} \begin{bmatrix} r_1(0) \\ r_2(0) \end{bmatrix} \tag{Equation 7a} \tag{40}$$

【 0 0 3 4 】

同様にして、チップ速度の N 倍の速度については、式 6 b 及び式 7 b を用いる。

【 0 0 3 5 】

【数 8 (a)】

$$\begin{bmatrix} h_1(0) \\ h_2(0) \\ | \\ h_N(0) \end{bmatrix} d(0) = \begin{bmatrix} r_1(0) \\ r_2(0) \\ | \\ r_N(0) \end{bmatrix}$$

Equation 6b

【 0 0 3 6】

【数 9 (a)】

10

$$\hat{d}(0) = \left\{ \begin{bmatrix} h_1(0) \\ \vdots \\ h_N(0) \end{bmatrix} \right\}^{-1} \begin{bmatrix} r_1(0) \\ \vdots \\ r_N(0) \end{bmatrix}$$

Equation 7b

【 0 0 3 7】

式 7 a 及び式 7 b を解く際に、後での使用のため、図示のとおり v^H を算定するが、式 7 a についての v^H は式 8 により算定し、蓄積する (5 2)。

【 0 0 3 8】

20

【数 1 0】

$$v^H = \left\{ \begin{bmatrix} h_1(0) \\ h_2(0) \end{bmatrix} \right\}^{-1} \begin{bmatrix} h_1(0) \\ h_2(0) \end{bmatrix}$$

Equation 8

【 0 0 3 9】

$\hat{d}^{\wedge}(0)$ は式 9 により v^H を用いて算定される。

【 0 0 4 0】

【数 1 1】

30

$\hat{d}^{\wedge}(0)$ is determined using v^H per Equation 9.

$$\hat{d}^{\wedge}(0) = v^H \begin{bmatrix} r_1(0) \\ r_2(0) \end{bmatrix}$$

Equation 9

H 行列のトプリッツ (Toeplitz) 構造を用いて、残りの拡散データ要素を零強制手法で連鎖的に式 1 0 (a) により算定することができる (5 4)。

【 0 0 4 1】

40

【数 1 2 (a)】

$$\hat{d}^{\wedge}(i) = v^H \left\{ \begin{bmatrix} r_1(i) \\ r_2(i) \end{bmatrix} - \begin{bmatrix} h_1(i) \\ h_2(i) \end{bmatrix} \hat{d}^{\wedge}(0) + \sum_{j=1}^{i-1} \begin{bmatrix} h_1(j) \\ h_2(j) \end{bmatrix} \hat{d}^{\wedge}(i-j-1) \right\}$$

Equation 10a

【 0 0 4 2】

チップ速度の N 倍については、式 1 0 (b) を用いる。

【 0 0 4 3】

【数 1 2 (b)】

$$d^{(i)} = v^H \begin{bmatrix} r_1(i) \\ | \\ r_N(i) \end{bmatrix} - \begin{bmatrix} h_1(i) \\ | \\ h_N(i) \end{bmatrix} \hat{d}(0) - \sum_{j=1}^{i-1} \begin{bmatrix} h_1(j) \\ | \\ h_N(j) \end{bmatrix} \cdot d^{(i-j-1)} \tag{Equation 10b}$$

【 0 0 4 4 】

10

拡散データベクトルの算定のあと、それぞれの通信信号バーストデータを、例えば拡散データベクトルとそれぞれのバースト符号との混合などにより逆拡散する(56)。

【 0 0 4 5 】

拡張前進代入手法を用いる際の複雑度を、逆拡散の過程を除き、表1に要約する。

【 0 0 4 6 】

表 1

• v^H の計算	...	4 回の乗算と 1 回の逆数演算	
• $d^{(0)}$ の計算	...	2 回の乗算	
• $d^{(1)}$ の計算	...	4 回の乗算	20
• $d^{(W-1)}$ までの各々の計算	...	2 回の乗算	
• $d^{(w)}$ から $d^{(Ns-1)}$ まで各 $d^{(i)}$ を計算	...	(2W + 2) 回の乗算	
• 全乗算回数	...	$2Ns + (W - 1) \cdot W + 2W \cdot (Ns - W + 1)$	
• 全計算回数	...	$2Ns + (W - 1) \cdot W + 2W \cdot (Ns - W + 1) + 5$	

TDDバースト2型についてはNsは1104で、Wは57であるから、毎秒200回、拡張前進代入を用いてdを解くには、チップ速度の2倍でのサンプリングの場合で毎秒99.9016百万回の実演算(MROPS)、チップ速度でのサンプリングに対しては44.95 MROPS回の実演算を必要とする。

【 0 0 4 7 】

30

データを推算するもう一つの手法はコレスキー法を近似的に帯状化する方法で、これは図5で説明する。相互相関行列Rは正方形(Ns x Ns)になるように算定され、式11により帯状化される(58)。

【 0 0 4 8 】

【数 1 3】

$$R = H^H H \tag{Equation 11}$$

【 0 0 4 9 】

40

ここで(・)^Hはエルミート関数を示す。Hは2(Ns + W - 1) x Nsの大きさである。式11は、チップ速度の2倍の速度でサンプリングする場合は式12(a)に書き替えられる。

【 0 0 5 0 】

【数 1 4 (a)】

$$R = \begin{bmatrix} H_1^H & H_2^H \end{bmatrix} \cdot \begin{bmatrix} H1 \\ H2 \end{bmatrix} = H_1^H H_1 + H_2^H H_2 \tag{Equation 12a}$$

【 0 0 5 1 】

50

チップ速度の N 倍の速度でのサンプリングについては、式 1 2 (b) を用いる。

【 0 0 5 2 】

【 数 1 4 (b) 】

$$R = \begin{bmatrix} H_1^H & H_2^H & \dots & H_N^H \end{bmatrix} \begin{bmatrix} H_1 \\ H_2 \\ | \\ H_N \end{bmatrix} \text{ or}$$

Equation 12b

$$R = \sum_{i=1}^N H_i^H H_i$$

10

【 0 0 5 3 】

式 1 2 a または式 1 2 b を用いて得られる R は、大きさ $N_s \times N_s$ 型の正方行列で、かつ帯状に区分けされ、チップ速度の 2 倍の速度でのサンプリングについては、式 1 3 に示すように、 $W=3$ で $N_s=10$ である。

【 0 0 5 4 】

【 数 1 5 】

$$R = \begin{bmatrix} R_0 & R_1 & R_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ R_1 & R_0 & R_1 & R_2 & 0 & 0 & 0 & 0 & 0 & 0 \\ R_2 & R_1 & R_0 & R_1 & R_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & R_2 & R_1 & R_0 & R_1 & R_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & R_2 & R_1 & R_0 & R_1 & R_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & R_2 & R_1 & R_0 & R_1 & R_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & R_2 & R_1 & R_0 & R_1 & R_2 & 0 \\ 0 & 0 & 0 & 0 & 0 & R_2 & R_1 & R_0 & R_1 & R_2 \\ 0 & 0 & 0 & 0 & 0 & 0 & R_2 & R_1 & R_0 & R_1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & R_2 & R_1 & R_0 \end{bmatrix}$$

Equation 13

20

30

【 0 0 5 5 】

一般に R の帯域幅は式 1 4 により得られる。

【 0 0 5 6 】

【 数 1 6 】

$$p = W - 1$$

Equation 14

40

【 0 0 5 7 】

近似コレスキー手法を用いるとともに、R の部分ブロック R_{sub} に $N_{col} \times N_{col}$ の大きさの行列を用いる。 R_{sub} の通常の大きさは $(2W - 1) \times (2W - 1)$ であるが、他のサイズの行列も使用できる。部分ブロック行列 R_{sub} は式 1 5 によるコレスキー分解を用いて分解される (6 0)。

【 0 0 5 8 】

【 数 1 7 】

$$R_{sub} = G G^H$$

Equation 15

50

【数 2 0 (b)】

$$G_{full}y = H_1^H r_1 + H_2^H r_2 + \dots + H_N^H r_N$$

【 0 0 6 6】

後退代入手法は、引き続き式 1 9 により拡散データベクトルを求めるのに用いられる。

【 0 0 6 7】

【数 2 1】

$$G_{full}^H d = y$$

Equation 19

10

【 0 0 6 8】

拡散データベクトル d を算定したあと、各バーストデータを逆拡散処理により算定する (6 6)。

【 0 0 6 9】

近似コレスキー分解手法の複雑度を、逆拡散処理を除いて、チップ速度の二倍でのサンプリングについて表 2 に示す。

表 2

演算項目	計算回数
・ $H^H H$ の演算	... $W(W + 1)$
・ コレスキー分解演算	... $N_{c.o.i}(W - 1)^2 / 2 + 3$
	$N_{c.o.i}(W - 1) / 2 - (W - 1) / 3 -$
	$(W - 1)^2 - 2(W - 1) / 3$
・ $H^H r$ の演算	... $2 N_s W$
・ 前進代入	... $[N_s - (W - 1) / 2] W$ 、及び N_s (逆数演算)
・ 後退代入	... $[N_s - (W - 1) / 2] W$ 、及び N_s (逆数演算)

20

T D D バースト 2 型については、 N_s は 1 1 0 4 で W は 5 7 であるから、チップ速度の 2 倍の速度でのサンプリングに対し、毎秒 2 0 0 回、帯状化近似コレスキー手法を実行するには、2 7 2 . 5 6 MROPS の実演算が必要とされる。これに对比して、帯状化コレスキー手法を厳密に実行すると、9 0 6 . 9 2 MROPS が必要とされる。また、チップ速度でのサンプリングに対しては、帯状化近似コレスキー手法を用いると 2 2 1 . 5 MROPS が必要である。

30

【 0 0 7 0】

データ検出のためのさらに他の手法、トプリッツ(Toeplitz)手法を用い(レビンソン - ダービン(Levinson-Durbin)型アルゴリズム)、これは図 6 で説明する。式 1 2 (a) および式 1 2 (b) をここで再記述する。チップ速度の N 倍については、式 1 2 b を用いる。

【 0 0 7 1】

【数 2 2 (a)】

$$R = \begin{bmatrix} H_1^H & H_2^H \end{bmatrix} \cdot \begin{bmatrix} H_1 \\ H_2 \end{bmatrix} = H_1^H H_1 + H_2^H H_2$$

Equation 12a

40

【 0 0 7 2】

【数 2 2 (b)】

$$R = \left[H_1^H, H_2^H, \dots, H_N^H \right] \begin{bmatrix} H_1 \\ H_2 \\ | \\ H_N \end{bmatrix} \text{ or}$$

Equation 12b

$$R = \sum_{i=1}^N H_i^H H_i$$

10

【 0 0 7 3】

R 行列は対称で、 $p = W - 1$ の帯域幅を有するトブリッツ型である(68)。R 行列の最上部の左角にある要素 $R(k)$ は、 $k \times k$ 行列で式 20 に示すように算定される。

【 0 0 7 4】

【数 2 3】

$$R(k) = \begin{bmatrix} R_0 & R_1 & \dots & R_{k-1} \\ R_1 & R_0 & & \vdots \\ & & \ddots & \\ R_{k-1} & R_{k-2} & \dots & R_0 \end{bmatrix}$$

Equation 20

20

【 0 0 7 5】

また、もう一つのベクトル R_k は、R 行列の要素を用いて式 21 により算定される(70)。

【 0 0 7 6】

【数 2 4】

30

$$R_k = \begin{bmatrix} R_1 \\ R_2 \\ \vdots \\ R_k \end{bmatrix}$$

Equation 21

【 0 0 7 7】

太字はその添字までの全ての要素を含む行列であることを示す。 $(k+1)$ 次の段階では、この系は式 22 により解かれる。

40

【 0 0 7 8】

【数 2 5】

$$R(k+1)d(k+1) = \left[H^H r \right]_{k+1}$$

Equation 22

【 0 0 7 9】

$[H^H r]_{k+1}$ は $H^H r$ の最初の $(k+1)$ 番目の要素である。 $d(k+1)$ は式 23 に示すように、長さ k のベクトル要素 $d(k+1)$ とスカラー量 $d_2(k+1)$ とに分解される。

【 0 0 8 0】

50

【数 2 6】

$$d(k+1) = \begin{bmatrix} d_1(k+1) \\ d_2(k+1) \end{bmatrix} \quad \text{Equation 23}$$

【0 0 8 1】

行列 $R(k+1)$ は式 2 4 のように分解される。

【0 0 8 2】

【数 2 7】

10

$$R(k+1) = \begin{bmatrix} R(k) & \vdots & E_k R_k \\ \cdots & \vdots & \cdots \\ R_k^H E_k & \vdots & R_0 \end{bmatrix} \quad \text{Equation 24}$$

【0 0 8 3】

ここで、 E_k は交換行列 (exchanger matrix) である。交換行列のベクトルへの作用は、そのベクトル要素の逆置換により生ずる。

20

【0 0 8 4】

線形予測のためにユール - ウォーカー (Yule - Walker) 等式を用いると、式 2 5 が得られる。

【0 0 8 5】

【数 2 8】

$$\begin{bmatrix} R(k-1) & \vdots & E_{k-1} R_{k-1} \\ \cdots & \vdots & \cdots \\ R_{k-1}^H E_{k-1} & \vdots & R_0 \end{bmatrix} \begin{bmatrix} y_1(k) \\ y_2(k) \end{bmatrix} = - \begin{bmatrix} R_{k-1} \\ R_k \end{bmatrix} \quad \text{Equation 25}$$

30

【0 0 8 6】

次数再帰 (recursion) 手法を用いると、式 2 6、式 2 7 及び式 2 8 が得られる。

【0 0 8 7】

【数 2 9】

$$y_1(k) = y(k-1) + y_2(k) E_{k-1} y(k-1) \quad \text{Equation 26}$$

【0 0 8 8】

【数 3 0】

40

$$y_2(k) = - \frac{[R_k + R_{k-1}^H E_{k-1} y(k-1)]}{[1 + R_{k-1}^H y(k-1)]} \quad \text{Equation 27}$$

【0 0 8 9】

【数 3 1】

$$\mathbf{y}(k) = \begin{bmatrix} \mathbf{y}_1(k) \\ \mathbf{y}_2(k) \end{bmatrix} \quad \text{Equation 28}$$

【0 0 9 0】

$\mathbf{y}(k)$ を使用して、 $d(k+1)$ が式 2 9、式 3 0 及び式 3 1 により算定する (7 4)。

【0 0 9 1】

【数 3 2】

10

$$\mathbf{d}_1(k+1) = \mathbf{d}(k) + d_2(k+1)E_k\mathbf{y}(k) \quad \text{Equation 29}$$

【0 0 9 2】

【数 3 3】

$$d_2(k+1) = \left[\frac{(H^H\mathbf{r})_{k+1} - \mathbf{R}_k^H E_k \mathbf{d}(k)}{1 + \mathbf{R}_k^H \mathbf{y}(k)} \right] \quad \text{Equation 30}$$

【0 0 9 3】

20

【数 3 4】

$$\mathbf{d}(k+1) = \begin{bmatrix} \mathbf{d}_1(k+1) \\ d_2(k+1) \end{bmatrix} \quad \text{Equation 31}$$

【0 0 9 4】

ここで、 $(H^H\mathbf{r})_{k+1}$ は $H^H\mathbf{r}$ の $(k+1)$ 番目の要素である。

【0 0 9 5】

正確に再帰手法を開始したのち、再帰演算を順次 $k=1, 2, \dots, N_s$ と行う。 $d(N_s)$ が式 3 2 の解である (7 4)。

30

【0 0 9 6】

【数 3 5】

$$\mathbf{R}\mathbf{d} = H^H\mathbf{r} \quad \text{Equation 32}$$

【0 0 9 7】

拡散データベクトル \mathbf{d} をバーストのチャネルリゼーション符号で逆拡散し、データが復元する (7 6)。

【0 0 9 8】

40

\mathbf{R} 行列の帯状化構造は、次に述べるとおり再帰演算に影響する。ここで、 $\mathbf{R}(2)$ 、 \mathbf{R}_2 は式 3 3 で示される。

【0 0 9 9】

【数 3 6】

$$\mathbf{R}(2) = \begin{bmatrix} R_0 & R_1 \\ R_1 & R_0 \end{bmatrix}, \mathbf{R}_2 = \begin{bmatrix} R_1 \\ R_2 \end{bmatrix} \quad \text{Equation 33}$$

【0 1 0 0】

50

式 27 及び式 30 での内積計算は、各 2 回の乗算を必要とする。説明のため k=6 の場合での式 20 の R 行列を式 34 に示す。

【 0 1 0 1 】
【 数 3 7 】

$$R(6) = \begin{bmatrix} R_0 & R_1 & R_2 & 0 & 0 & 0 \\ R_1 & & & R_2 & 0 & 0 \\ R_2 & & & & R_2 & 0 \\ 0 & R_2 & & & & R_2 \\ 0 & 0 & R_2 & & & R_1 \\ 0 & 0 & 0 & R_2 & & R_0 \end{bmatrix}, R_6 = \begin{bmatrix} R_1 \\ R_2 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \text{Equation 34}$$

10

【 0 1 0 2 】

ベクトル R_6 の 0 値ではない要素数は、行列の帯幅 p に等しい。式 27 及び式 28 のそれぞれの内積、 $R_6^H E_6 y(k)$ 、 $R_6^H E_6 d(k)$ を計算する場合、 p 回 (k 回ではない) だけ乗算すればよい。式 26 及び式 29 の再帰計算に関しては、演算上の効率化は全く得られない。

【 0 1 0 3 】

表 3 にトブリッツ手法を実行する際の複雑度を示す。

20

【 0 1 0 4 】

表 3

演算項目	計算回数	MROPS
・バースト毎に 1 回実行される関数、 $H^H H$ の計算		
	1.3244	
・ y に対するユール - ウォーカ等式の計算	$672,888 \times 100/10^6$	269.1552
・バースト毎に 2 回実行される関数、 $H^H r$ の計算		100.68
・ $R(k+1)d(k+1)H^H r$ の計算	$672,888 \times 100/10^6$	538.3104

30

TDD バーストタイプのためのトブリッツ手法に対する MROPS 総計は、チップ速度の 2 倍の速度でのサンプリングに対しては、909.4566MROPS で、チップ速度でのそれに対しては 858.4688MROPS である。

【 0 1 0 5 】

データ検出用のその他の手段は、高速フーリエ変換 (FFT) を使用するもので、図 7 で説明する。チップ速度でのサンプリングを用いる場合、チャンネル行列 H は端部効果を除くと正方となる。 H 行列に対する循環型 (circulant) 近似、受信されたベクトル信号 r の FFT、及びチャンネルベクトル H を用いて、データ推算値を求めるやり方を探る。

40

【 0 1 0 6 】

チップ速度のある倍数の速度でのサンプリング、例えば 2 倍では、 H 行列は正方でもなく循環型でもない。しかしながら、式 13 に示す行列であるチャンネル相関行列 $R = H^H H(84)$ 、この部分行列は、式 35a の点線により示される様に循環型である。チップ速度の N 倍の速度でのサンプリングに対しては、チャンネル相関行列は式 35b で表わされる。

【 0 1 0 7 】

【数 3 8 (a)】

$$R = \begin{bmatrix} R_0 & R_1 & R_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ R_1 & R_0 & R_1 & R_2 & 0 & 0 & 0 & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ R_2 & R_1 & R_0 & R_1 & R_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & R_2 & R_1 & R_0 & R_1 & R_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & R_2 & R_1 & R_0 & R_1 & R_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & R_2 & R_1 & R_0 & R_1 & R_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & R_2 & R_1 & R_0 & R_1 & R_2 & 0 \\ 0 & 0 & 0 & 0 & 0 & R_2 & R_1 & R_0 & R_1 & R_2 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & R_2 & R_1 & R_0 & R_1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & R_2 & R_1 & R_0 \end{bmatrix} \quad 10$$

Equation 35a

【 0 1 0 8】

20

【数 3 8 (b)】

$$R = \begin{bmatrix} H_1^H H_2^H & \dots & H_N^H \end{bmatrix} \begin{bmatrix} H_1 \\ H_2 \\ \vdots \\ H_N \end{bmatrix} = \sum_{i=1}^N H_i^H H_i \quad \text{Equation 35b}$$

【 0 1 0 9】

R行列を循環型で近似することにより、式 3 6 , 式 3 7 及び式 3 8 が使用できる。

30

【 0 1 1 0】

【数 3 9】

$$R^H = D \Delta D^H \quad \text{Equation 36}$$

【 0 1 1 1】

【数 4 0】

$$\Delta = \text{diag}(D(R)_1) \quad 40$$

$$\hat{d} = \text{diag} \left(D \begin{bmatrix} R_0 \\ R_1 \\ R_2 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \right) \quad \text{Equation 37}$$

【 0 1 1 2】

50

【数 4 1】

$$R\hat{d} = H^H r \quad \text{Equation 38}$$

【0 1 1 3】

ここで、 $(R)_1$ は対角行列に拡張されたR行列の第1の列である。第1の列を用いて記述されるが、この手法はR行列のどの列でも使えるように修正できる(86)。しかし、0値ではない要素を最も多く有する列、例えば R_1 、 R_2 、 R_0 、 R_1 、 R_2 と要素が並ぶ様な列を用いるのが望ましい。これらの列は、通常、少なくとも、行列の両側からW番目にある列のどれかで、例えばW番目と $(N_s - W - 1)$ 番目の列を含めて、それらの間にある列である。式38及び式39が零強制等化手法で使用される。

10

【0 1 1 4】

【数 4 2】

$$\hat{d} = R^{-1}(H^H r) \quad \text{Equation 39}$$

【0 1 1 5】

Dは直交離散フーリエ変換(DFT)行列であることから、式40、式41、式42が導かれる。従って、 \hat{d} は式43、式44及び式45(a)により、フーリエ変換を

20

【0 1 1 6】

【数 4 3】

$$D^H D = N_s I \quad \text{Equation 40}$$

【0 1 1 7】

【数 4 4】

$$D^{-1} = (1/N_s) D^H \quad \text{Equation 41}$$

30

【0 1 1 8】

【数 4 5】

$$R^{-1} = \frac{1}{N_s} D^H \Delta^{-1} \frac{1}{N_s} D \quad \text{Equation 42}$$

【0 1 1 9】

【数 4 6】

$$R^{-1} = \frac{1}{N_s^2} D^H \Delta^{-1} [D(H^H r)] \quad \text{Equation 43}$$

40

【0 1 2 0】

【数 4 7】

$$D^H \hat{d} = \frac{1}{N_s} \Delta^{-1} [F(H^H r)] \quad \text{Equation 44}$$

【0 1 2 1】

【数 4 8 (a)】

$$F(\hat{d}) = \frac{F(H^H r)}{N_s F((R)_1)} \quad \text{Equation 45a}$$

【 0 1 2 2】

用いて算定できる。ここで $(\cdot)_1$ は第 1 の列を示すが、相似の式では、R 行列のどの列でも使用できる。F(\cdot) はフーリエ変換関数を示す。F($H^H r$) は式 4 5 (b) により、

【 0 1 2 3】

【数 4 8 (b)】

$$F(H^H r) = N_c [F(h_1)F(r_1) + \dots + F(h_N)F(r_N)] \quad \text{Equation 45b}$$

【 0 1 2 4】

高速フーリエ変換(FFT)を行って適切に計算される。式 4 5 (a) の結果について逆フーリエ変換、 $F^{-1}(\cdot)$ を行うと、逆拡散ベクトルが得られる(8 8)。送信されたデータは、適合する符号を用いて逆拡散することにより復元される(9 0)。

【 0 1 2 5】

このFFT法の複雑度を表 4 に示す。

演算項目	計算回数	MROPS
・ バースト毎に 1 回実行 される関数計算、 $H^H H$ の計算		1.3224
・ $F([R]_1) \cdot N_s \log_2 N_s$	$11160 \times 100 / 10^6$	4.4640
・ バースト毎に 2 回実行 される関数計算、FFT によ る $H^H r$ の計算		
38		
・ 式 4 5 の演算		0.8832
・ $F^{-1}(d) \cdot N_s \log_2 N_s$		
8.9280		
総計		55MROPS

このFFTによる解法は他の手法よりも複雑ではない。しかしながら、精度の低下が循環型近似から生ずる。

チップ速度のある倍数でのサンプリングに対するデータベクトルを解くためにFETを応用する更なる手法では、図 8 で説明するようにサンプルを重み付けして合成する。チップ速度の 2 倍でのサンプリングに対する説明のため、 r_1 は奇数サンプル、 r_2 は偶数サンプルとする。 r_1 の各要素、例えば最初の要素 $r_1(0)$ は式 4 6 で示すように、重み付けするとともに、対応の r_2 の要素である $r_2(0)$ で合成される。 $r_{eff}(0)$ は実効的に合成された行列 r_{eff} の修正合成要素である。W1、W2

【 0 1 2 6】

10

20

30

40

【数 4 9】

$$r_{eff}(0) = W_1 r_1(0) + W_2 r_2(0) \quad \text{Equation 46}$$

【0 1 2 7】

は重みである。チップ速度のN倍でのサンプリングに対しては、式 4 7 を用いる。H₁ から H_n までのチャンネル応答行列に対し、類似の重み付けを行い、

【0 1 2 8】

【数 5 0】

10

$$r_{eff}(0) = W_1 r_1(0) + \dots + W_n r_n(0) \quad \text{Equation 47}$$

【0 1 2 9】

H_{eff}

が生成される(9 2)。その結果、式 3 は式 4 8 となる。もたらされた式は、

【0 1 3 0】

【数 5 1】

20

$$\mathbf{r}_{eff} = H_{eff} \mathbf{d} + \mathbf{n} \quad \text{Equation 48}$$

【0 1 3 1】

式 4 9 によるFFTで容易に解くことができるNs × Ns型の式である(9 4)。

【0 1 3 2】

【数 5 2】

$$F(\mathbf{d}) = \frac{F(\mathbf{r}_{eff})}{F((H_{eff})_1)} \quad \text{Equation 49}$$

30

【0 1 3 3】

逆フーリエ変換を用いて拡散データベクトルが算定される。次に、バーストデータが、バースト符号を用いた拡散データベクトルの逆拡散により得られる(9 6)。式 4 9 はH_{eff} 行列の最初の列を用いるが、この方法はH_{eff} の代表的な列を用いるように修正できる。高速フーリエ変換(FFT)を用いる更に他の解法は零埋め込みを利用するもので、これは図 9 を用いて説明する。式 5 はデータベクトルにおける全ての他の要素、例えば偶数要素のようなものを0値とする零埋め込みにより修正される(9 8)。修正されたd行列はd~で表す。H行列も拡張して、H~で示す。H行列の拡張は、各列をその列の右側に繰返し、かつ繰返された列の各要素を1行ずつ下にずらし、さらにずらした列の上端に0値を埋め込むことにより行う。式 4 9 aは、W = 3、Ns = 4 の場合で、チップ速度の2倍の速度でのサンプリングに対してのこのような系の表示である。

40

【0 1 3 4】

零強制解を式 50 に示す (1 0 0)。

【 0 1 3 8 】

【 数 5 4 】

$$F(\mathbf{d}^{\sim}) = \frac{F(\mathbf{r})}{F((H^{\sim})_1)}$$

Equation 50

【 0 1 3 9 】

第 1 列以外の列が類似の高速フーリエ変換(FFT)では使用できる。更に最初のチャネル応答行列 H の列、或いは H の列から導出された H[~] の推定拡張された列のいずれも使用できる。d[~] の N 番目ごとの値を用いて d を推算する。次に、適合する符号を用いて、拡散データベクトル d を逆拡散し、データを復元する (1 0 2)。

10

【 産業上の利用可能性 】

【 0 1 4 0 】

時分割複信 CDMA (TDD/CDMA) 通信システムなど大容量の通信システムの受信機に適用できる。

【 符号の説明 】

【 0 1 4 1 】

1 0 無線通信システム

20

1 2 基地局

1 4 ユーザ装置

1 6 通信信号バースト

1 8 ガード期間

2 0 ミドアンプル

2 2、2 4 データ領域

2 6 送信機

2 8 受信機

3 0 無線伝送チャネル

3 2 データ発生器

30

3 4 変調 / 拡散装置

3 6 変調器

3 8、4 0 アンテナ

4 2 復調器

4 3 サンプリング装置

4 4 チャネル推算装置

4 6 単一ユーザ検出(SUD)装置

4 8 チップ速度の倍数の速度で受信信号 r をサンプリングする

5 0 チップ速度の倍数の速度でのサンプリングに対応するチャネル応答行列の算定

40

5 2 拡散前進代入手法及び零強制手法を用いて拡散データベクトルの最初の要素 $\hat{d}^{\sim}(0)$ を算定。 $\hat{d}^{\sim}(0)$ 算定には、また、チャネル応答行列を用いて因数 v^H を求め、それを使用。 v^H は蓄積

5 4 残りの拡散データベクトル要素 $\hat{d}^{\sim}(1)$ 、...、 $\hat{d}^{\sim}(N_s - 1)$ を算定。算定には、蓄積した因数 v^H を利用しながら、零強制手法と拡散前進手法を連鎖的に使用

5 6 拡散データベクトル d の逆拡散処理によりデータを復元

5 8 チャネル応答行列を用いて、正方で、かつ帯状化された行列 R を算定

6 0 R 行列のサブブロック R_{sub} について、近似コレスキー分解を実行

6 2 コレスキー係数 G を拡張し、密行列(full matrix) G_{full} を生成

6 4 前進代入手法及び後進代入手法を用いて、拡散データベクトル d を算定

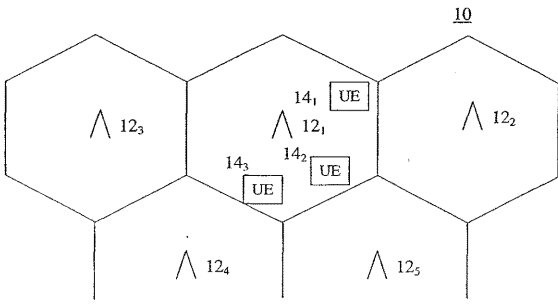
50

- 6 6 拡散データベクトル d から逆拡散によりデータを復元
- 6 8 チャンネル応答行列を用いて、対称で、かつトプリッツ型の R 行列を算定
- 7 0 R 行列を用いて、行列 R_k を算定
- 7 2 $(k + 1)$ 次の段階で、長さが k で、かつスカラー量がそれぞれ $d_2(k + 1)$ 、 $y_2(k)$ であるそれぞれのベクトル $d_1(k + 1)$ 、 $y_1(k)$ を算定することにより、ベクトル $d(k + 1)$ を算定。更に $d(k + 1)$ を決めてから、順次 $d(N_s)$ まで算定
- 7 4 $d(N_s)$ が推算される拡散データベクトル
- 7 6 拡散データベクトル d から、逆拡散によりデータを復元
- 8 4 チャンネル応答行列とそれ自身の転置行列を結合して、チャンネル相関行列 R を算定
- 8 6 R 行列のある列を用いて、 R 行列を循環型として近似
- 8 8 チャンネル相関行列のある列か、または、近似循環型行列及び受信されたベクトルを乗算したチャンネル応答行列のある列のフーリエ変換により、拡散データベクトル d を算定。得られた結果について逆フーリエ変換
- 9 0 拡散データベクトル d から、逆拡散によりデータを復元
- 9 2 チップ速度のある倍数の速度でのサンプルと対応するチャンネル応答行列の要素とを重み付けにより、実効的なチップ速度サンプルを合成し、その結果として、実効的な受信信号ベクトル及びチャンネル応答行列を生成
- 9 4 効果的なチャンネル応答行列の代表的な列及び受信されたベクトル、それぞれのフーリエ変換により、拡散データベクトル d を算定。次に、得られた結果を逆フーリエ変換
- 9 6 拡散データベクトル d から、逆拡散によりデータを復元
- 9 8 データベクトルが受信されたデータシンボルと同じ長さになるようにデータベクトルへの 0 値埋め込み
- 1 0 0 受信されたベクトル及び合成されたチャンネル相関行列の代表的な列、それぞれのフーリエ変換により、0 値埋め込みデータベクトルを決定。次に、得られた結果を逆フーリエ変換
- 1 0 2 拡散データベクトル d から、逆拡散によりデータを復元

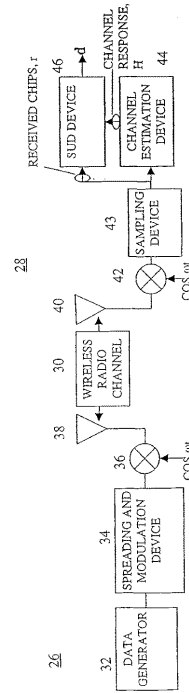
10

20

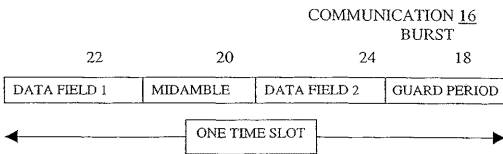
【 図 1 】



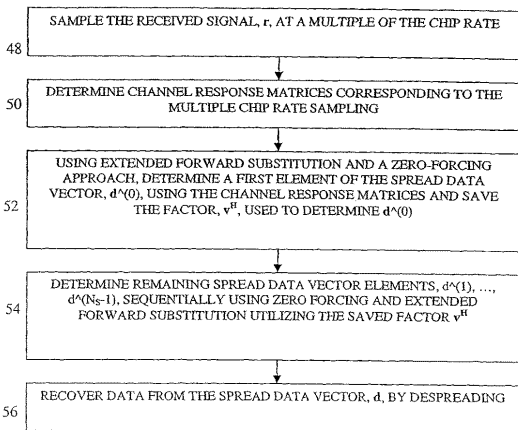
【 図 2 】



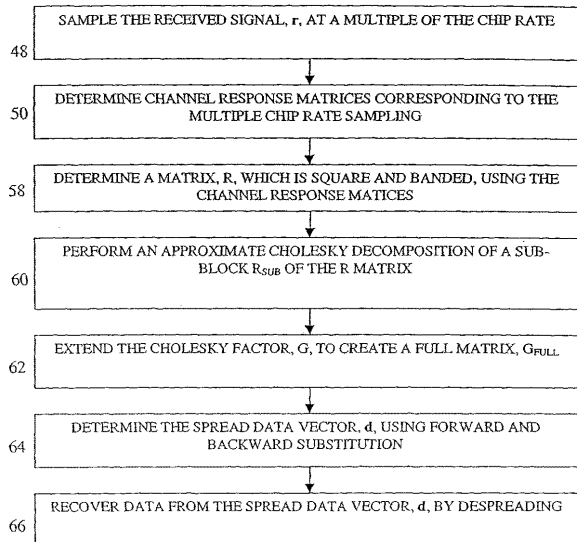
【 図 3 】



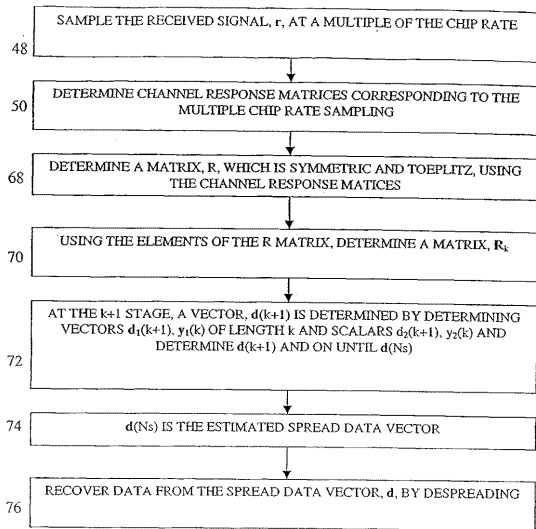
【 図 4 】



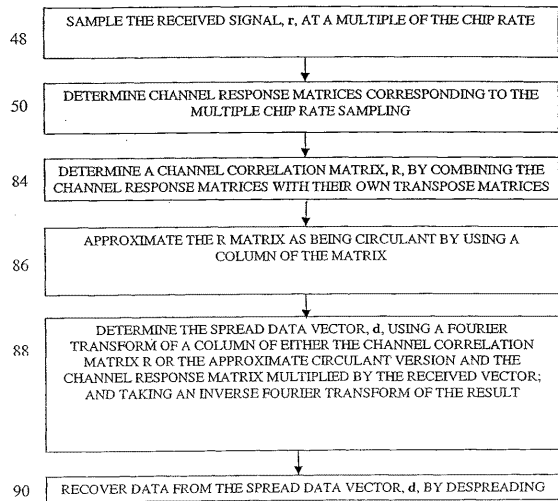
【 図 5 】



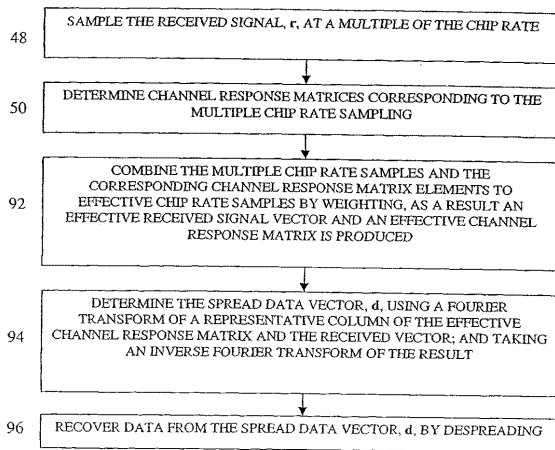
【 図 6 】



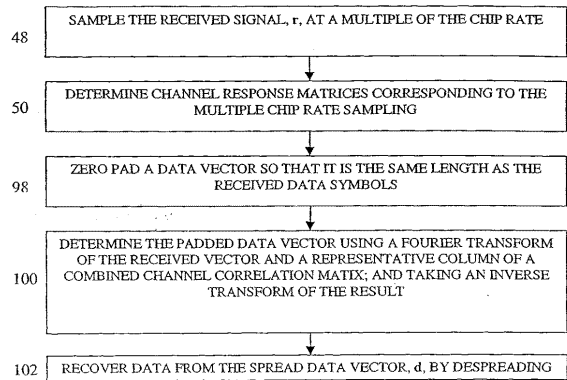
【 図 7 】



【 図 8 】



【 図 9 】



フロントページの続き

- (72)発明者 ゼイラ, アリエラ
アメリカ合衆国 ニューヨーク州 1 1 7 4 3 ハンティントン, ウェスト ネック ロード 2
3 9
- (72)発明者 パン, ジュン - リン
アメリカ合衆国 ニューヨーク州 1 1 7 8 4 セルデン, コート ストリート 1 5
- (72)発明者 ミスラ, ラジ マニ
アメリカ合衆国 ニューヨーク州 1 1 2 1 5 ブルックリン, セブンス アヴェニュー 3 5 8
, ナンバー 1 5 7