

(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2015-135621

(P2015-135621A)

(43) 公開日 平成27年7月27日(2015.7.27)

(51) Int.Cl.	F 1	テーマコード (参考)
G06F 17/12 (2006.01)	G06F 17/12	5B056
H04B 7/005 (2006.01)	H04B 7/005	5K046

審査請求 未請求 請求項の数 5 O L (全 21 頁)

(21) 出願番号	特願2014-7112 (P2014-7112)	(71) 出願人	000005223 富士通株式会社 神奈川県川崎市中原区上小田中4丁目1番1号
(22) 出願日	平成26年1月17日 (2014.1.17)	(74) 代理人	100104190 弁理士 酒井 昭徳
		(72) 発明者	長谷川 剛 神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内
		Fターム(参考)	5B056 BB02 5K046 EE06 EF11

(54) 【発明の名称】 演算装置、演算方法および無線通信装置

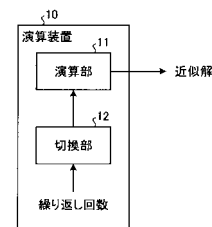
(57) 【要約】

【課題】 発散を抑えつつ並列効率の向上を図ること。

【解決手段】 演算装置 10 は、無線通信システムにおける無線信号の等化処理を行うイコライザの重みを演算する。演算装置 10 は、演算部 11 と、切換部 12 と、を備える。演算部 11 は、並列演算器を用いた繰り返し演算により一次方程式の近似解を得る。切換部 12 は、繰り返し演算の繰り返し回数に応じて、演算部 11 による繰り返し演算の第 1 アルゴリズムと、第 1 アルゴリズムより発散しやすく並列演算効率が高い第 2 アルゴリズムと、を切り換える。

【選択図】 図 1 B

図 1A に示した演算装置における信号の流れの一例を示す図



【特許請求の範囲】

【請求項 1】

並列演算器を用いた繰り返し演算により一次方程式の近似解を得る演算部と、
前記演算部による前記繰り返し演算を、前記繰り返し演算の繰り返し回数に応じて、第 1 アルゴリズムと、前記第 1 アルゴリズムより発散しやすく並列演算効率が高い第 2 アルゴリズムと、に交互に切り換える切換部と、
を備えることを特徴とする演算装置。

【請求項 2】

前記第 1 アルゴリズムは、前記近似解を示す 1 つのベクトルに対応する演算式を用いる更新アルゴリズムであり、

前記第 2 アルゴリズムは、前記近似解を示すベクトルを分割した複数のベクトルに対応する各演算式を用いる更新アルゴリズムである、

ことを特徴とする請求項 1 に記載の演算装置。

【請求項 3】

前記切換部は、前記第 2 アルゴリズムを用いた場合に前記近似解が常に収束する条件を前記一次方程式が満たしているか否かを判定し、前記条件を前記一次方程式が満たしている期間は、前記繰り返し回数に関わらず前記第 2 アルゴリズムに維持することを特徴とする請求項 1 または 2 に記載の演算装置。

【請求項 4】

無線通信システムにおける無線信号の等化処理を行うイコライザの重みを演算する演算装置が、並列演算器を用いた繰り返し演算により一次方程式の近似解を得る演算方法であって、

前記繰り返し演算を、前記繰り返し演算の繰り返し回数に応じて、第 1 アルゴリズムと、前記第 1 アルゴリズムより発散しやすく並列演算効率が高い第 2 アルゴリズムと、に交互に切り換えることを特徴とする演算方法。

【請求項 5】

並列演算器を用いた繰り返し演算により一次方程式の近似解を得る演算装置であり、前記繰り返し演算を、前記繰り返し演算の繰り返し回数に応じて、第 1 アルゴリズムと、前記第 1 アルゴリズムより発散しやすく並列演算効率が高い第 2 アルゴリズムと、に交互に切り換える演算装置と、

前記演算装置によって得られた前記近似解に基づく重みにより無線信号の等化処理を行うイコライザと、

を備えることを特徴とする無線通信装置。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、演算装置、演算方法および無線通信装置に関する。

【背景技術】

【0002】

従来、連立一次方程式を反復法で解く手法として、ガウス - ザイデル法やガウス - ヤコビ法が知られている。また、回路シミュレータによる水道管網の解析において、ガウス - ザイデル法を用いて仮想状態方程式を解く技術が知られている（たとえば、下記特許文献 1 参照。）。

【先行技術文献】

【特許文献】

【0003】

【特許文献 1】特開平 10 - 320444 号公報

【発明の概要】

【発明が解決しようとする課題】

【0004】

10

20

30

40

50

しかしながら、上述した従来技術では、並列演算器を用いた繰り返し演算により一次方程式の近似解を得る場合に、近似解が発散しにくい反復法を用いると並列演算の効率が低く、並列演算の効率が高い反復法を用いると近似解が発散する場合がある。

【0005】

1つの側面では、本発明は、発散を抑えつつ並列効率の向上を図ることができる演算装置、演算方法および無線通信装置を提供することを目的とする。

【課題を解決するための手段】

【0006】

上述した課題を解決し、目的を達成するため、本発明の一側面によれば、並列演算器を用いた繰り返し演算により一次方程式の近似解を得る場合に、前記繰り返し演算を、前記繰り返し演算の繰り返し回数に応じて、第1アルゴリズムと、前記第1アルゴリズムより発散しやすく並列演算効率が高い第2アルゴリズムと、に交互に切り換える演算装置、演算方法および無線通信装置が提案される。

10

【発明の効果】

【0007】

本発明の一側面によれば、発散を抑えつつ並列効率の向上を図ることができるという効果を奏する。

【図面の簡単な説明】

【0008】

【図1A】図1Aは、実施の形態1にかかる演算装置の一例を示す図である。

20

【図1B】図1Bは、図1Aに示した演算装置における信号の流れの一例を示す図である。

【図2A】図2Aは、実施の形態2にかかる演算装置の一例を示す図である。

【図2B】図2Bは、図2Aに示した演算装置における信号の流れの一例を示す図である。

【図2C】図2Cは、演算装置の動作の一例を示すフローチャートである。

【図3A】図3Aは、演算装置を適用した無線受信機の一例を示す図である。

【図3B】図3Bは、図3Aに示した無線受信機における信号の流れの一例を示す図である。

【図4A】図4Aは、一次方程式のブロック分割の一例を示す図である。

30

【図4B】図4Bは、各アルゴリズムにおける近似解の一例を示す図である。

【図5A】図5Aは、演算装置のハードウェア構成の一例を示す図である。

【図5B】図5Bは、図5Aに示した演算装置のハードウェア構成における信号の流れの一例を示す図である。

【図6A】図6Aは、更新アルゴリズムの切り換えの一例を示す図である。

【図6B】図6Bは、更新アルゴリズムの切り換えの別の例を示す図である。

【図7】図7は、ガウス・ザイデル法とガウス・ヤコビ法とを切り換える場合の近似解の一例を示す図である。

【図8A】図8Aは、実施の形態3にかかる演算装置の一例を示す図である。

【図8B】図8Bは、図8Aに示した演算装置における信号の流れの一例を示す図である。

40

【図9】図9は、一次方程式のブロック分割の一例を示す図である。

【図10】図10は、更新アルゴリズムの切り換えの一例を示す図である。

【図11】図11は、ガウス・ザイデル法と変形ガウス・ザイデル法とを切り換える場合の近似解の一例を示す図である。

【図12A】図12Aは、実施の形態4にかかる演算装置の一例を示す図である。

【図12B】図12Bは、図12Aに示した演算装置における信号の流れの一例を示す図である。

【発明を実施するための形態】

【0009】

50

以下に図面を参照して、本発明にかかる演算装置、演算方法および無線通信装置の実施の形態を詳細に説明する。

【0010】

(実施の形態1)

図1Aは、実施の形態1にかかる演算装置の一例を示す図である。図1Bは、図1Aに示した演算装置における信号の流れの一例を示す図である。図1A、図1Bに示すように、実施の形態1にかかる演算装置10は、演算部11と、切換部12と、を備える。

【0011】

演算部11は、並列演算器を用いた繰り返し演算により(連立)一次方程式の近似解を得る演算部である。演算部11は、たとえば繰り返し演算のたびに、得られた近似解を出力する。

10

【0012】

切換部12は、演算部11による繰り返し演算を、繰り返し演算の繰り返し回数に応じて、第1アルゴリズムと第2アルゴリズムとに交互に切り換える切換部である。切り換えは、たとえば、一回の演算ごとの切り換えや、複数回の演算ごとの切り換えなどを含む。また、たとえば第1アルゴリズム、第2アルゴリズム、第2アルゴリズム、第1アルゴリズム、第2アルゴリズム、第2アルゴリズム、...のように、異なる回数ずつ交互に切り換えてもよい。また、第1アルゴリズムおよび第2アルゴリズムのみを用いる場合に限らず、たとえば第3アルゴリズムをさらに用いるようにしてもよい。

【0013】

第2アルゴリズムは、第1アルゴリズムより発散しやすい(収束性が悪い)並列演算効率が高い更新アルゴリズムである。発散しやすい更新アルゴリズムとは、たとえば、反復行列のスペクトル半径が大きい更新アルゴリズムである。反復行列のスペクトル半径については後述する。また、並列演算効率が高い更新アルゴリズムとは、たとえば、並列演算器により同時実行が可能な演算の数が多い更新アルゴリズムである。

20

【0014】

このように、繰り返し演算の繰り返し回数に応じて収束性および並列演算効率が異なる複数のアルゴリズムを交互に切り換えることにより、近似解の発散を抑えつつ、並列演算効率の向上を図ることができる。

【0015】

<更新アルゴリズムの一例>

一例としては、第1アルゴリズムをガウス-ザイデル法の更新アルゴリズムとし、第2アルゴリズムをガウス-ヤコビ法の更新アルゴリズムとすることができる。繰り返し演算の繰り返し回数に応じてガウス-ザイデル法およびガウス-ヤコビ法の各更新アルゴリズムを切り換えることにより、近似解の発散を抑えつつ、並列演算効率の向上を図ることができる。

30

【0016】

または、第1アルゴリズムを、近似解を示す1つのベクトルに対応する演算式を用いる更新アルゴリズムとし、第2アルゴリズムを、近似解を示すベクトルを分割した複数のベクトルに対応する各演算式を用いる更新アルゴリズムとしてもよい。近似解を示す1つのベクトルに対応する演算式を用いる更新アルゴリズムは、一例としてはガウス-ザイデル法の更新アルゴリズムである。

40

【0017】

近似解を示すベクトルを分割した複数のベクトルに対応する各演算式を用いる更新アルゴリズムは、たとえばガウス-ザイデル法の更新アルゴリズムを変形したものとすることができる。近似解を示すベクトルを分割した複数のベクトルに対応する各演算式を用いる更新アルゴリズムは、近似解を示す1つのベクトルに対応する演算式を用いる更新アルゴリズムに比べて、発散しやすく並列演算効率が高い更新アルゴリズムとなる。また、たとえばガウス-ヤコビ法の更新アルゴリズムより収束しやすい更新アルゴリズムとなる。このため、近似解の発散をより抑えつつ、並列演算効率の向上を図ることができる。

50

【 0 0 1 8 】

< 変形例 >

また、切換部 1 2 は、第 2 アルゴリズムを用いた繰り返し演算による近似解が常に収束する所定条件を一次方程式が満たしているか否かを判定してもよい。そして、切換部 1 2 は、判定結果に基づいて、所定条件を一次方程式が満たしている期間は、演算部 1 1 の繰り返し回数に関わらず第 2 アルゴリズムを維持する。これにより、第 2 アルゴリズムを用いても近似解が発散しない状況において並列演算効率をより向上させることができる。

【 0 0 1 9 】

(実施の形態 2)

(実施の形態 2 にかかる演算装置)

10

図 2 A は、実施の形態 2 にかかる演算装置の一例を示す図である。図 2 B は、図 2 A に示した演算装置における信号の流れの一例を示す図である。図 2 A , 図 2 B に示すように、実施の形態 2 にかかる演算装置 2 0 は、更新部セレクタ 2 1 (Updater Selector) と、ガウス - ザイデル更新部 2 2 (Gauss - Seidel Updater) と、ガウス - ヤコビ更新部 2 3 (Gauss - Jacobi Updater) と、並列 M A C 演算部 2 4 と、重み保持部 2 5 ($w^{(n)}$) と、を備える。演算装置 2 0 は、たとえば無線受信機に設けられ、受信信号の等化処理を行うイコライザの合成重みを算出する演算装置である。

【 0 0 2 0 】

図 1 A , 図 1 B に示した演算装置 1 0 は、たとえば演算装置 2 0 によって実現することができる。図 1 A , 図 1 B に示した演算部 1 1 は、たとえばガウス - ザイデル更新部 2 2 、ガウス - ヤコビ更新部 2 3 および並列 M A C 演算部 2 4 によって実現することができる。図 1 A , 図 1 B に示した切換部 1 2 は、たとえば更新部セレクタ 2 1 によって実現することができる。

20

【 0 0 2 1 】

更新部セレクタ 2 1 には、重み保持部 2 5 から出力された重みベクトル $w^{(n)}$ と、重みベクトル $w^{(n)}$ の演算の繰り返し回数 n と、が入力される。更新部セレクタ 2 1 は、ガウス - ザイデル更新部 2 2 およびガウス - ヤコビ更新部 2 3 のいずれかを繰り返し回数 n に応じて周期的に選択する。そして、更新部セレクタ 2 1 は、選択した更新部へ、重み保持部 2 5 から出力された重みベクトル $w^{(n)}$ を出力する。

30

【 0 0 2 2 】

ガウス - ザイデル更新部 2 2 は、更新部セレクタ 2 1 から重みベクトル $w^{(n)}$ が出力されると、並列 M A C 演算部 2 4 における繰り返し演算における演算データを、正方行列 R 、ステアリングベクトル h および重みベクトル $w^{(n)}$ に基づいて更新する。また、ガウス - ザイデル更新部 2 2 は、演算データをガウス - ザイデル法の更新アルゴリズムによって更新する。そして、ガウス - ザイデル更新部 2 2 は、更新した演算データに基づいて並列 M A C 演算部 2 4 によって算出された重みベクトル $w^{(n+1)}$ を取得し、取得した重みベクトル $w^{(n+1)}$ を重み保持部 2 5 へ出力する。

【 0 0 2 3 】

ガウス - ヤコビ更新部 2 3 は、更新部セレクタ 2 1 から重みベクトル $w^{(n)}$ が出力されると、並列 M A C 演算部 2 4 における繰り返し演算における演算データを、正方行列 R 、ステアリングベクトル h および重みベクトル $w^{(n)}$ に基づいて更新する。また、ガウス - ヤコビ更新部 2 3 は、演算データをガウス - ヤコビ法の更新アルゴリズムによって更新する。そして、ガウス - ヤコビ更新部 2 3 は、更新した演算データに基づいて並列 M A C 演算部 2 4 によって算出された重みベクトル $w^{(n+1)}$ を取得し、取得した重みベクトル $w^{(n+1)}$ を重み保持部 2 5 へ出力する。

40

【 0 0 2 4 】

並列 M A C 演算部 2 4 は、並列の M A C (Multiply and Accumulation : 積和) 演算器を備える。並列 M A C 演算部 2 4 は、並列の M A C 演算器を用いて、ガウス - ザイデル更新部 2 2 またはガウス - ヤコビ更新部 2 3 によって更新された

50

演算データに基づく重みベクトル $w^{(n+1)}$ を算出する。そして、算出結果をガウス - ザイデル更新部 22 またはガウス - ヤコビ更新部 23 へ返す。

【0025】

重み保持部 25 は、ガウス - ザイデル更新部 22 またはガウス - ヤコビ更新部 23 から出力された重みベクトル $w^{(n+1)}$ を保持する。そして、重み保持部 25 は、保持した重みベクトル $w^{(n+1)}$ を、つぎの演算周期において重みベクトル $w^{(n)}$ として出力する。重み保持部 25 から出力された重みベクトル $w^{(n)}$ は、演算装置 20 による演算結果として出力されるとともに更新部セレクタ 21 へ入力される。

【0026】

図 2A, 図 2B に示す演算装置 20 によれば、ガウス - ザイデル法の更新アルゴリズムと、ガウス - ヤコビ法による更新アルゴリズムと、を周期的に切り換えることができる。これにより、ガウス - ヤコビ法の更新アルゴリズムを部分的に用いることにより並列演算の効率を上げることができる。また、ガウス - ヤコビ法では発散してしまうような条件においても、途中でガウス - ザイデル法の更新アルゴリズムを用いることにより、安定に収束する状態を維持することができる。このため、並列演算の効率および収束性を向上させることができる。

10

【0027】

(演算装置の動作)

図 2C は、演算装置の動作の一例を示すフローチャートである。図 2A, 図 2B に示した演算装置 20 は、たとえば図 2C に示す各ステップを実行する。まず、演算装置 20 は、繰り返し回数 n および重みベクトル $w^{(0)}$ を初期化 ($n = 0$, $w^{(0)} = (0, 0, \dots, 0)^T$) する (ステップ S201)。

20

【0028】

つぎに、演算装置 20 は、更新部セレクタ 21 がガウス - ザイデル (ガウス - ザイデル更新部 22) を選択したか否かを判断する (ステップ S202)。更新部セレクタ 21 がガウス - ザイデルを選択した場合 (ステップ S202: Yes) は、演算装置 20 は、ガウス - ザイデル更新部 22 で重みベクトル $w^{(n)}$ より重みベクトル $w^{(n+1)}$ を求め (ステップ S203)、ステップ S205 へ移行する。更新部セレクタ 21 がガウス - ヤコビ (ガウス - ヤコビ更新部 23) を選択した場合 (ステップ S202: No) は、演算装置 20 は、ガウス - ヤコビ更新部 23 で重みベクトル $w^{(n)}$ より重みベクトル $w^{(n+1)}$ を求める (ステップ S204)。

30

【0029】

つぎに、演算装置 20 は、並列 MAC 演算部 24 において、ステップ S203 またはステップ S204 によって求めた重みベクトル $w^{(n+1)}$ によって重みベクトル $w^{(n)}$ を更新する (ステップ S205)。つぎに、演算装置 20 は、繰り返し回数 n をインクリメント ($n = n + 1$) する (ステップ S206)。

【0030】

つぎに、演算装置 20 は、所定の終了条件を満たしたか否かを判断する (ステップ S207)。所定の終了条件は、たとえば、繰り返し回数 n が所定回数に達したとすることができる。所定の終了条件を満たしていない場合 (ステップ S207: No) は、演算装置 20 は、ステップ S202 へ戻る。終了条件を満たした場合 (ステップ S207: Yes) は、演算装置 20 は、重みベクトル $w^{(n)}$ を出力し (ステップ S208)、一連の動作を終了する。

40

【0031】

または、ステップ S207 とステップ S208 の順序を入れ替え、演算装置 20 は所定の終了条件を満たすまで、繰り返し演算のたびに重みベクトル $w^{(n)}$ を出力するようにしてもよい。

【0032】

更新部セレクタ 21 はガウス - ザイデル更新部 22 およびガウス - ヤコビ更新部 23 のいずれかを繰り返し回数 n に応じて選択するため、以上の各ステップにより、繰り返し回

50

数 n に応じて各更新アルゴリズムを切り換えることができる。

【 0 0 3 3 】

(演算装置を適用した無線受信機)

図 3 A は、演算装置を適用した無線受信機の一例を示す図である。図 3 B は、図 3 A に示した無線受信機における信号の流れの一例を示す図である。図 2 A , 図 2 B に示した演算装置 2 0 は、たとえば図 3 A , 図 3 B に示す無線受信機 3 0 に適用することができる。無線受信機 3 0 は、アンテナ 3 1 と、A G C アンプ 3 2 と、A / D コンバータ 3 3 と、復調部 3 4 と、復号部 3 5 と、を備える無線通信装置である。

【 0 0 3 4 】

アンテナ 3 1 は、無線送信された信号を受信し、受信した信号を A G C アンプ 3 2 へ出力する。A G C アンプ 3 2 は、アンテナ 3 1 から出力された信号を一定利得で増幅する A G C (A u t o m a t i c G a i n C o n t r o l : 利得一定制御) を行う。そして、A G C アンプ 3 2 は、増幅した信号を A / D コンバータ 3 3 へ出力する。

10

【 0 0 3 5 】

A / D コンバータ 3 3 は、A G C アンプ 3 2 から出力された信号をアナログ信号からデジタル信号に変換する。そして、A / D コンバータ 3 3 は、デジタル信号に変換した信号を復調部 3 4 へ出力する。

【 0 0 3 6 】

復調部 3 4 は、A / D コンバータ 3 3 から出力された信号の復調処理を行う。そして、復調部 3 4 は、復調処理によって得られた軟判定ビット列を復号部 3 5 へ出力する。復調部 3 4 による復調処理には、たとえば等化処理が含まれる。たとえば、復調部 3 4 は、チャンネル推定部 3 6 と、相関行列算出部 3 7 と、イコライザ重み算出部 3 8 と、イコライザ 3 9 と、を備える。

20

【 0 0 3 7 】

チャンネル推定部 3 6 は、A / D コンバータ 3 3 から出力された信号に基づくチャンネル推定を行う。そして、チャンネル推定部 3 6 は、チャンネル推定結果を上述したステアリングベクトル h としてイコライザ重み算出部 3 8 へ出力する。

【 0 0 3 8 】

相関行列算出部 3 7 は、A / D コンバータ 3 3 から出力された信号に基づく相関行列を算出する。そして、相関行列算出部 3 7 は、算出した相関行列を上述した正方行列 R としてイコライザ重み算出部 3 8 へ出力する。

30

【 0 0 3 9 】

イコライザ重み算出部 3 8 は、チャンネル推定部 3 6 から出力されたステアリングベクトル h と、相関行列算出部 3 7 から出力された正方行列 R と、に基づいて、イコライザ 3 9 における合成重みを算出する。そして、イコライザ重み算出部 3 8 は、算出した合成重みをイコライザ 3 9 へ出力する。図 2 A , 図 2 B に示した演算装置 2 0 は、たとえばイコライザ重み算出部 3 8 に適用することができる。

【 0 0 4 0 】

イコライザ 3 9 は、A / D コンバータ 3 3 から出力された信号に対して、イコライザ重み算出部 3 8 から出力された合成重みに基づく等化処理を行う。そして、イコライザ 3 9 は、等化処理を行った信号を復号部 3 5 へ出力する。

40

【 0 0 4 1 】

復号部 3 5 は、復調部 3 4 から出力された軟判定ビット列に基づく復号処理を行う。復号部 3 5 による復号処理には、たとえば誤り訂正処理などが含まれる。復号部 3 5 は、復号処理によって得られた “ 0 ” , “ 1 ” のビット列を出力する。

【 0 0 4 2 】

なお、無線受信機 3 0 は、たとえばアンテナ 3 1 と A / D コンバータ 3 3 との間に、デュプレクサ、フィルタ、ダウンコンバータ、L N A (L o w N o i s e A m p l i f i e r) などのアナログ素子をさらに備えていてもよい。

【 0 0 4 3 】

50

(各アルゴリズムによる一次方程式の近似解法)

つぎに、一次方程式である $Rw = h$ のガウス - ザイデル法およびガウス - ヤコビ法による近似解法を説明する。 w は、たとえば近似解となる重みベクトル (たとえばイコライザの合成重みベクトル) である。 R は、相関行列であって $N \times N$ の正方行列である。 h はステアリングベクトルである。相関行列 R およびステアリングベクトル h は、たとえば演算装置 20 の外部から与えられる。

【0044】

図 4 A は、一次方程式のブロック分割の一例を示す図である。一次方程式 $Rw = h$ は、たとえば図 4 A に示す式 4 1 によって示すことができる。式 4 1 の D は、相関行列 R の対角成分のみを取り出した対角行列である。 E は、相関行列 R の上三角行列である。 F は、相関行列 R の下三角行列である。

10

【0045】

<ガウス - ザイデル法による近似解法>

まず、一次方程式 $Rw = h$ のガウス - ザイデル法による近似解法について説明する。ガウス - ザイデル法においては、式 4 1 を下記 (1) 式のように変換した重みベクトル w の更新式 (演算式) が用いられる。そして、適当な初期値の重みベクトル $w^{(0)}$ から始めて、重みベクトル $w^{(n)}$ を順次算出することにより近似的に重みベクトル w が求められる。

【0046】

【数 1】

$$w^{(n+1)} = D^{-1}(h - EW^{(n)} - FW^{(n+1)}) \quad \dots(1)$$

20

【0047】

上記 (1) 式の更新式は、重みベクトル $w^{(n)}$ から左辺の重みベクトル $w^{(n+1)}$ を求める式である一方、右辺に重みベクトル $w^{(n+1)}$ が含まれている。ここで、上記 (1) 式は下記 (2) 式のように変形することができる。 $w_0^{(n+1)}$, $w_1^{(n+1)}$, $w_2^{(n+1)}$, ... は、重みベクトル $w^{(n+1)}$ の要素である。

【0048】

【数 2】

$$\begin{aligned} w_0^{(n+1)} &= (1/R_{00})(h_0 - R_{01}w_1^{(n)} - R_{02}w_2^{(n)} - R_{03}w_3^{(n)} \dots - R_{0N}w_N^{(n)}) \\ w_1^{(n+1)} &= (1/R_{11})(h_1 - R_{10}w_0^{(n+1)} - R_{12}w_2^{(n)} - R_{13}w_3^{(n)} \dots - R_{1N}w_N^{(n)}) \\ w_2^{(n+1)} &= (1/R_{22})(h_2 - R_{20}w_0^{(n+1)} - R_{21}w_1^{(n+1)} - R_{23}w_3^{(n)} \dots - R_{2N}w_N^{(n)}) \\ &\vdots \\ &\dots(2) \end{aligned}$$

30

40

【0049】

上記 (2) 式に示すように、たとえば重み $w_0^{(n+1)}$ は、右辺に重み $w_x^{(n+1)}$ が含まれていないため算出可能である。また、重み $w_1^{(n+1)}$ は、右辺に重み $w_0^{(n+1)}$ を含むが、重み $w_0^{(n+1)}$ の算出後であれば算出できる。

【0050】

このように、ガウス - ザイデル法においては、上記 (2) 式の各式を上から順次計算することにより、右辺に重みベクトル $w^{(n+1)}$ を含む上記 (1) 式であっても重みベクトル $w^{(n)}$ の算出が可能である。ただし、上記 (2) 式の各式を順次計算する必要があるということは同時に計算できないということの意味するため、並列演算器による並列演算の効率は低い。

50

【 0 0 5 1 】

< ガウス - ヤコビ法による近似解法 >

つぎに、一次方程式 $R w = h$ のガウス - ヤコビ法による近似解法について説明する。ガウス - ヤコビ法においては、式 4 1 を下記 (3) 式のように変換した重みベクトル w の更新式が用いられる。そして、適当な初期値の重みベクトル $w^{(0)}$ から始めて、重みベクトル $w^{(n)}$ を順次算出することにより近似的に重みベクトル w が求められる。

【 0 0 5 2 】

【数 3】

$$w^{(n+1)} = D^{-1} \{ h - (E + F) w^{(n)} \} \quad \dots(3)$$

10

【 0 0 5 3 】

上記 (3) 式の更新式は、右辺に重みベクトル $w^{(n+1)}$ が含まれていない。したがって、ガウス - ヤコビ法においては、重みベクトル $w^{(n+1)}$ の要素 $w_0^{(n+1)}$, $w_1^{(n+1)}$, $w_2^{(n+1)}$, ... の演算の順序に制約がない。このため、並列演算器による並列演算の効率が高い。ただし、ガウス - ヤコビ法は、ガウス - ザイデル法に比べて発散しやすい。

【 0 0 5 4 】

< 更新アルゴリズムの収束性 >

たとえば上記 (1) 式に示したガウス - ザイデル法の更新式は、たとえば $M^{(N+1)} = M w^{(n)} + C$ (ただし $M = - (D + F)^{-1} E$) のように変換することができる。この M を反復行列、 M の最大固有値の絶対値を「スペクトル半径」と称する。「スペクトル半径」が 1 より小さければ w は収束する。また、スペクトル半径が小さいほど、速く w は収束する。したがって、発散しやすい更新アルゴリズムとは、たとえば反復行列のスペクトル半径が大きい更新アルゴリズムである。

20

【 0 0 5 5 】

(各アルゴリズムにおける近似解)

図 4 B は、各アルゴリズムにおける近似解の一例を示す図である。図 4 B において、横軸 (Number of Iterations) は、演算の繰り返し回数 n を示している。縦軸 (Weight value) は、近似解として算出された合成重みを示している。演算結果 4 2, 4 3 は、たとえば図 3 A, 図 3 B に示した無線受信機 3 0 のイコライザ重み算出部 3 8 に各アルゴリズムを適用した場合において、伝搬環境が 3 パス等レベルのときの、繰り返し回数に対する近似解 (合成重み) の演算結果を示している。

30

【 0 0 5 6 】

演算結果 4 2 (Gauss - Seidel) は、ガウス - ザイデル法における演算結果を示している。演算結果 4 3 (Gauss - Jacobi) は、ガウス - ヤコビ法における演算結果を示している。演算結果 4 2, 4 3 に示すように、ガウス - ヤコビ法では、ガウス - ザイデル法に比べて近似解が発散しやすい。このため、仮に、ガウス - ザイデル法またはガウス - ヤコビ法のみを用いる場合は、近似解の収束性と並列演算の効率を両立できない。

【 0 0 5 7 】

これに対して、演算装置 2 0 は、ガウス - ザイデル法の更新アルゴリズムと、ガウス - ヤコビ法による更新アルゴリズムと、を繰り返し回数に応じて周期的に切り換えることができる。これにより、近似解の収束性と並列演算の効率を向上させることができる。

40

【 0 0 5 8 】

(演算装置のハードウェア構成)

図 5 A は、演算装置のハードウェア構成の一例を示す図である。図 5 B は、図 5 A に示した演算装置のハードウェア構成における信号の流れの一例を示す図である。図 2 A, 図 2 B に示した演算装置 2 0 は、たとえば図 5 A, 図 5 B に示す DSP 5 0 (Digital Signal Processor) によって実現することができる。

【 0 0 5 9 】

50

DSP50は、制御回路51と、MAC演算器52~54と、メモリ55(Memory)と、を備える。図2A,図2Bに示した更新部セクタ21、ガウス-ザイデル更新部22およびガウス-ヤコビ更新部23は、たとえば制御回路51によって実現することができる。図2A,図2Bに示した並列MAC演算部24は、たとえばMAC演算器52~54によって実現することができる。図2A,図2Bに示した重み保持部25は、たとえばメモリ55によって実現することができる。制御回路51の制御により、メモリ55に格納されたデータがMAC演算器52~54によって演算され、演算結果がメモリ55に格納される。

【0060】

ここでは演算装置20を実現する回路としてDSPを用いる例について説明したが、演算装置20は、DSPに限らず、FPGA(Field Programmable Gate Array)など各種のプロセッサによって実現することができる。また、ここでは3個のMAC演算器52~54を備える構成について説明したが、並列演算器は2個の演算器であってもよいし、4個以上(たとえば64個)の演算器であってもよい。また、DSP50がMAC演算器52~54を備える構成について説明したが、MAC演算器52~54はDSP50の外部に設けられていてもよい。

10

【0061】

(更新アルゴリズムの切り換え)

図6Aは、更新アルゴリズムの切り換えの一例を示す図である。図6Aに示す切り換え順序61は、更新部セクタ21による、ガウス-ザイデル法の更新アルゴリズムと、ガウス-ヤコビ法による更新アルゴリズムと、の切り換えの順序を示している。

20

【0062】

切り換え順序61に示す例では、更新部セクタ21は、ガウス-ザイデル法の更新アルゴリズム(GS)1回と、ガウス-ヤコビ法による更新アルゴリズム(GJ)2回と、を交互に切り換える。これにより、ガウス-ヤコビ法をそのまま用いたときに発散していた状況でも、途中で挟んだガウス-ザイデル法の更新アルゴリズムにより近似解が収束半径に留まり、結果も収束することが期待できる。

【0063】

たとえば、更新部セクタ21は、切り換え順序61を示す情報(テーブル等)を記憶しており、記憶した情報に基づいて、ガウス-ザイデル法の更新アルゴリズムと、ガウス-ヤコビ法による更新アルゴリズムと、を切り換える。

30

【0064】

図6Bは、更新アルゴリズムの切り換えの別の例を示す図である。図6Bに示す切り換え順序62は、更新部セクタ21による、ガウス-ザイデル法の更新アルゴリズムと、ガウス-ヤコビ法による更新アルゴリズムと、の切り換えの順序を示している。

【0065】

切り換え順序62に示す例では、更新部セクタ21は、所定期間Tの前半でガウス-ザイデル法の更新アルゴリズム(GS)を多く用い、所定期間Tの後半ではガウス-ヤコビ法の更新アルゴリズム(GJ)を多く用いる。これにより、所定期間Tの前半である程度近似解を収束させることができれば、後半は収束性の悪いガウス-ヤコビ法を用いても正しく収束することが期待できる。所定期間Tは、たとえば無線受信機による等化処理の一周期である。

40

【0066】

たとえば、更新部セクタ21は、切り換え順序62を示す情報(テーブル等)を記憶しており、記憶した情報に基づいて、ガウス-ザイデル法の更新アルゴリズムと、ガウス-ヤコビ法による更新アルゴリズムと、を切り換える。

【0067】

(ガウス-ザイデル法とガウス-ヤコビ法とを切り換える場合の近似解)

図7は、ガウス-ザイデル法とガウス-ヤコビ法とを切り換える場合の近似解の一例を示す図である。図7において、横軸(Number of Iterations)は、

50

演算の繰り返し回数 n を示している。縦軸 (Weight value) は、近似解として算出された合成重みを示している。演算結果 7 1, 7 2 のそれぞれは、繰り返し回数に対する近似解 (合成重み) の演算結果を示している。

【0068】

演算結果 7 1 (Gauss-Seidel) は、ガウス-ザイデル法における演算結果を参考として示している。演算結果 7 2 (GS1+GJ2) は、ガウス-ザイデル法の更新アルゴリズム (GS) 1 回と、ガウス-ヤコビ法による更新アルゴリズム (GJ) 2 回と、を交互に切り換えた場合 (たとえば図 6 A 参照) における演算結果を示している。

【0069】

演算結果 7 1, 7 2 に示すように、ガウス-ザイデル法の更新アルゴリズム (GS) と、ガウス-ヤコビ法による更新アルゴリズム (GJ) と、を交互に切り換えることにより、ガウス-ザイデル法単体の場合と同様に良好な収束性を得ることができる。そして、ガウス-ザイデル法の更新アルゴリズム (GS) と、ガウス-ヤコビ法による更新アルゴリズム (GJ) と、を交互に切り換えることにより、ガウス-ザイデル法単体の場合に比べて並列演算の効率を向上させることができる。

10

【0070】

このように、実施の形態 2 によれば、繰り返し演算の繰り返し回数に応じてガウス-ザイデル法およびガウス-ヤコビ法の各更新アルゴリズムを交互に切り換えることにより、近似解の発散を抑えつつ、並列演算効率の向上を図ることができる。

【0071】

(実施の形態 3)

図 8 A は、実施の形態 3 にかかる演算装置の一例を示す図である。図 8 B は、図 8 A に示した演算装置における信号の流れの一例を示す図である。図 8 A, 図 8 B において、図 2 A, 図 2 B に示した部分と同様の部分については同一の符号を付して説明を省略する。

20

【0072】

図 8 A, 図 8 B に示すように、実施の形態 3 にかかる演算装置 2 0 は、図 2 A, 図 2 B に示したガウス-ヤコビ更新部 2 3 に代えて変形ガウス-ザイデル更新部 8 1 (Modified Gauss-Seidel Updater) を備える。変形ガウス-ザイデル更新部 8 1 は、たとえば図 5 A, 図 5 B に示した制御回路 5 1 によって実現することができる。

30

【0073】

変形ガウス-ザイデル更新部 8 1 は、更新部セレクタ 2 1 から重みベクトル $w^{(n)}$ が出力されると、並列 MAC 演算部 2 4 における繰り返し演算における演算データを、正方行列 R 、ステアリングベクトル h および重みベクトル $w^{(n)}$ に基づいて更新する。また、変形ガウス-ザイデル更新部 8 1 は、演算データを変形ガウス-ザイデル法の更新アルゴリズムによって更新する。変形ガウス-ザイデル法の更新アルゴリズムについては後述する (たとえば図 9 参照。) そして、変形ガウス-ザイデル更新部 8 1 は、更新した演算データに基づいて並列 MAC 演算部 2 4 によって算出された重みベクトル $w^{(n+1)}$ を取得し、取得した重みベクトル $w^{(n+1)}$ を重み保持部 2 5 へ出力する。

【0074】

(変形ガウス-ザイデル法の更新アルゴリズムによる一次方程式の近似解法)

図 9 は、一次方程式のブロック分割の一例を示す図である。一次方程式 $Rw = h$ は、たとえば図 9 に示す式 9 0 によって示すことができる。変形ガウス-ザイデル法の更新アルゴリズムにおいては、重みベクトル w が重みベクトル z_1, z_2 の 2 つのブロックに分割される。これに対応して、ステアリングベクトル h も 2 つのブロックに分割される。また、相関行列 R は 4 つのブロックに分割される。

40

【0075】

変形ガウス-ザイデル法においては、式 9 0 を下記 (4) 式および下記 (5) 式のように変換した重みベクトル z_1, z_2 の更新式が用いられる。そして、適当な初期値の重みベクトル $z_1^{(0)}, z_2^{(0)}$ から始めて、重みベクトル $z_1^{(n)}, z_2^{(n)}$ を順次算出することに

50

より近似的に重みベクトル w が求められる。

【 0 0 7 6 】

【 数 4 】

$$z_1^{(n+1)} = D_1^{-1} \left(h_1 - (E_1 + F_1) z_1^{(n)} - G_1 z_2^{(n)} \right) \quad \dots(4)$$

【 0 0 7 7 】

【 数 5 】

$$z_2^{(n+1)} = D_2^{-1} \left(h_2 - (E_2 + F_2) z_2^{(n)} - G_2 z_1^{(n+1)} \right) \quad \dots(5)$$

10

【 0 0 7 8 】

上記 (4) 式は、下記 (6) 式のように変形することができる。また、上記 (5) 式は、下記 (7) 式のように変形することができる。なお、 $K = N / 2$ である。

【 0 0 7 9 】

【 数 6 】

$$w_0^{(n+1)} = (1/R_{00}) \left(h_0 - R_{01} w_1^{(n)} \dots - R_{0K-1} w_{K-1}^{(n)} - R_{0,K} w_K^{(n)} \dots - R_{0N} w_{N-1}^{(n)} \right)$$

$$w_1^{(n+1)} = (1/R_{11}) \left(h_1 - R_{10} w_0^{(n)} \dots - R_{1K-1} w_{K-1}^{(n)} - R_{1,K} w_K^{(n)} \dots - R_{1N} w_{N-1}^{(n)} \right)$$

⋮

$$w_{K-0}^{(n+1)} =$$

$$(1/R_{K-1,K-1}) \left(h_{K-1} - R_{K-1,1} w_0^{(n)} \dots - R_{K-1,K-1} w_{K-2}^{(n)} - R_{K-1,K} w_K^{(n)} \dots - R_{K-1N} w_{N-1}^{(n)} \right)$$

⋯(6)

【 0 0 8 0 】

【 数 7 】

$$w_K^{(n+1)} = (1/R_{KK}) \left(h_K - R_{K0} w_0^{(n+1)} \dots - R_{KK-1} w_{K-1}^{(n+1)} - R_{K,K} w_K^{(n)} \dots - R_{KN} w_{N-1}^{(n)} \right)$$

$$w_{K+1}^{(n+1)} =$$

$$(1/R_{K+1,K+1}) \left(h_{K+1} - R_{K+1,0} w_0^{(n+1)} \dots - R_{K+1,K-1} w_{K-1}^{(n+1)} - R_{K+1,K} w_K^{(n)} \dots - R_{K+1,N} w_{N-1}^{(n)} \right)$$

⋮

$$w_{N-1}^{(n+1)} =$$

$$(1/R_{N-1,N-1}) \left(h_{N-1} - R_{N-1,0} w_0^{(n+1)} \dots - R_{N-1,K-1} w_{K-1}^{(n+1)} - R_{N-1,K} w_K^{(n)} \dots - R_{N-1N} w_{N-1}^{(n)} \right)$$

⋯(7)

【 0 0 8 1 】

上記 (6) 式に示すように、重みベクトル w の前半の重みベクトル z_1 を算出する各式には未知の変数は含まれていないため、全て並列で実行することが可能である。また、上記 (7) 式に示すように、重みベクトル w の後半の重みベクトル z_2 を算出する各式には未知の変数は含まれるが、これらの未知の変数は上記 (6) 式で求められる値である。このため、上記 (6) 式の演算後であれば、重みベクトル w の後半の重みベクトル z_2 を算

50

出する各式は全て並列で実行することが可能である。

【0082】

このように、変形ガウス - ザイデル法の更新アルゴリズムは、近似解を示すベクトルを分割した複数のベクトルに対応する各更新式を用いる更新アルゴリズムである。変形ガウス - ザイデル法の更新アルゴリズムによれば、通常の高ス - ザイデル法の更新アルゴリズムに比べて多くの演算を同時に実行できるようになり、並列演算の効率が向上する。また、変形ガウス - ザイデル法は、高ス - ヤコビ法よりも重みベクトル w の更新を速く行うことができるため収束が速い。ただし、変形ガウス - ザイデル法は、単体では発散することもある。

【0083】

(更新アルゴリズムの切り換え)

図10は、更新アルゴリズムの切り換えの一例を示す図である。図10に示す切り換え順序100は、実施の形態3にかかる更新部セレクタ21による、高ス - ザイデル法の更新アルゴリズムと、変形高ス - ザイデル法による更新アルゴリズムと、の切り換えの順序を示している。

【0084】

切り換え順序100に示す例では、更新部セレクタ21は、高ス - ザイデル法の更新アルゴリズム(GS)1回と、変形高ス - ザイデル法による更新アルゴリズム(M-GS)2回と、を交互に切り換える。これにより、変形高ス - ザイデル法をそのまま用いたときに発散していた状況でも、途中で挟んだ高ス - ザイデル法の更新アルゴリズムにより近似解が収束半径に留まり、結果も収束することが期待できる。

【0085】

たとえば、更新部セレクタ21は、切り換え順序100を示す情報(テーブル等)を記憶しており、記憶した情報に基づいて、高ス - ザイデル法の更新アルゴリズムと、変形高ス - ザイデル法による更新アルゴリズムと、を切り換える。

【0086】

(高ス - ザイデル法と変形高ス - ザイデル法とを切り換える場合の近似解)

図11は、高ス - ザイデル法と変形高ス - ザイデル法とを切り換える場合の近似解の一例を示す図である。図11において、図7に示した部分と同様の部分については同一の符号を付して説明を省略する。演算結果111は、演算結果71, 72と同様に、繰り返し回数に対する近似解(合成重み)の演算結果を示している。

【0087】

また、演算結果111(GS1 + M-GS2)は、高ス - ザイデル法の更新アルゴリズム(GS)1回と、変形高ス - ザイデル法による更新アルゴリズム(M-GS)2回と、を交互に切り換えた場合(たとえば図10参照)における演算結果を示している。

【0088】

演算結果72, 111に示すように、高ス - ザイデル法と変形高ス - ザイデル法の各更新アルゴリズムを切り換えることにより、高ス - ザイデル法と高ス - ヤコビ法の各更新アルゴリズムを切り換える場合より、収束にかかる時間を短縮することができる。

【0089】

このように、実施の形態3によれば、繰り返し演算の繰り返し回数に応じて、高ス - ザイデル法と変形高ス - ザイデル法の各更新アルゴリズムを交互に切り換えることができる。すなわち、近似解を示す1つのベクトルに対応する更新式を用いる更新アルゴリズムと、近似解を示すベクトルを分割した複数のベクトルに対応する各更新式を用いる更新アルゴリズムと、を交互に切り換えることができる。これにより、近似解の発散を抑えつつ、並列演算効率の向上を図ることができる。

【0090】

(実施の形態4)

(実施の形態4にかかる演算装置)

図12Aは、実施の形態4にかかる演算装置の一例を示す図である。図12Bは、図1

10

20

30

40

50

2 A に示した演算装置における信号の流れの一例を示す図である。図 1 2 A , 図 1 2 B において、図 2 A , 図 2 B に示した部分と同様の部分については同一の符号を付して説明を省略する。図 1 2 A , 図 1 2 B に示すように、実施の形態 4 にかかる演算装置 2 0 は、図 2 A , 図 2 B に示した構成に加えて絶対収束判定部 1 2 1 を備える。絶対収束判定部 1 2 1 は、たとえば図 5 A , 図 5 B に示した制御回路 5 1 によって実現することができる。

【 0 0 9 1 】

絶対収束判定部 1 2 1 は、ガウス - ザイデル更新部 2 2 およびガウス - ヤコビ更新部 2 3 へ入力される相関行列 R に基づいて、ガウス - ヤコビ法において常に収束する条件（絶対収束条件と称する）を満たしているか否かを判定する。

【 0 0 9 2 】

絶対収束条件は、たとえば、相関行列 R の各行について、その行の対角成分の絶対値が、それ以外の成分の絶対値の和よりも大きいという条件である。たとえば、絶対収束判定部 1 2 1 は、下記 (8) 式に示す条件が満たされているか否かを判断する。絶対収束判定部 1 2 1 は、判定結果を更新部セレクタ 2 1 へ出力する。

【 0 0 9 3 】

【 数 8 】

$$|R_{ii}| > \sum_{j \neq i} |R_{ij}| \quad (i=1,2,\dots,n) \quad \dots(8)$$

【 0 0 9 4 】

更新部セレクタ 2 1 は、絶対収束判定部 1 2 1 から出力された判定結果に基づいて、絶対収束条件が満たされている場合は、繰り返し回数 n に関わらずにガウス - ヤコビ更新部 2 3（ガウス - ヤコビ法）を選択する。これにより、ガウス - ヤコビ法であっても常に収束する状況においてはガウス - ヤコビ法を用いることにより、並列演算の効率を向上させることができる。

【 0 0 9 5 】

このように、実施の形態 4 によれば、ガウス - ヤコビ法を用いた繰り返し演算による近似解が常に収束する絶対収束条件を一次方程式が満たしているか否かを判定することができる。そして、判定結果に基づいて、絶対収束条件を一次方程式が満たしている期間は、繰り返し回数に関わらず更新アルゴリズムをガウス - ヤコビ法に維持することができる。これにより、ガウス - ヤコビ法を用いても近似解が発散しない状況において並列演算効率をより向上させることができる。

【 0 0 9 6 】

なお、ここでは実施の形態 2 に示したようにガウス - ザイデル法およびガウス - ヤコビ法の各更新アルゴリズムを切り換える場合に、ガウス - ヤコビ法の絶対収束条件の判定を行う場合について説明した。ただし、これに限らず、たとえば実施の形態 3 のようにガウス - ザイデル法および変形ガウス - ザイデル法の各更新アルゴリズムを切り換える場合に変形ガウス - ザイデル法の絶対収束条件の判定を行うようにしてもよい。

【 0 0 9 7 】

この場合は、切換部 1 2 は、絶対収束条件を一次方程式が満たしている期間においては、繰り返し回数に関わらず更新アルゴリズムを変形ガウス - ザイデル法に維持する。これにより、変形ガウス - ザイデル法を用いても近似解が発散しない状況において並列演算効率をより向上させることができる。

【 0 0 9 8 】

（各更新アルゴリズムにおける並列演算の効率）

各更新アルゴリズムについて、30 元の連立方程式を解く場合に要するクロック数について説明する。ここでは M A C 演算機を 6 4 個並列に実行でき、そのときのパイプライン処理のレイテンシが 5 [c l o c k] であるとする。また、繰り返し回数 n を 3 とする。

【 0 0 9 9 】

まずガウス - ザイデル法の計算について説明する。いずれの方法でも 1 回の更新演算に

10

20

30

40

50

30 × 29 = 870回のMAC演算を要するが、ガウス - ザイデル法ではそのうちの約半分のみが並列に演算可能である。このため、 $870 / 2 / 64 = 6.79$ となり、7 [clock]を要する。この全ての計算結果を得るにはさらにレイテンシの5 [clock]を要するが、この場合はつぎの処理に隠されるため全体のクロック数への影響はない。
【0100】

残りの並列に演算できない部分は、29行を1行ずつ、結果を確認しながら順番に実行することになるため、レイテンシの影響により $(1 + 5) \times 29$ [clock]を要する。これを3回繰り返すと $(7 + (1 + 5) \times 29) \times 3 = 543$ [clock]となる。
【0101】

つぎに、ガウス - ヤコビ法では、870回のMAC演算が全て並列に演算可能である。このため、 $870 / 64 = 13.59$ となり、14 [clock]とレイテンシの5 [clock]で演算が完了する。これを3回繰り返すと $(14 + 5) \times 3 = 57$ [clock]となる。
【0102】

変形ガウス - ザイデル法では、870回のMAC演算を半分ずつ並列に演算可能である。後半の演算では前半の演算結果を使うので、前半の結果が出るまで待つことになるため、前半も後半もそれぞれ $(7 + 5)$ [clock]を要する。これを3回繰り返すと $((7 + 5) + (7 + 5)) \times 3 = 72$ [clock]となる。
【0103】

すなわち、ガウス - ザイデル法では543 [clock]、ガウス - ヤコビ法では57 [clock]、変形ガウス - ザイデル法では72 [clock]となる。これらの結果を比べると、ガウス - ザイデル法とガウス - ヤコビ法とで処理ステップ数に10倍近い差が出ており、ガウス - ザイデル法が十分に並列演算器を活用できず、レイテンシの影響が大きく出ていることが分かる。一方、変形ガウス - ザイデル法は、ガウス - ヤコビ法よりはクロック数が多いものの、26%程度の増加である。
【0104】

このように、ガウス - ヤコビ法や変形ガウス - ザイデル法を用いることにより所要ステップ数を削減でき、これらに加えてガウス - ザイデル法の更新アルゴリズムも混在させて使用することにより収束特性を維持することができる。すなわち、所要ステップ数が少なく、収束性のよい方式が実現できる。
【0105】

以上説明したように、演算装置、演算方法および無線通信装置によれば、発散を抑えつつ並列効率の向上を図ることができる。
【0106】

たとえば、無線受信機におけるシングルキャリア信号向けのイコライザでは、イコライザ重みを求めるためにサイズの大きな一次方程式を解くことになる。携帯端末においてはこの演算量が消費電力に影響するため、厳密に一次方程式を解くのではなく、繰り返し演算により近似的に解を求める場合がある。この繰り返し演算として、ガウス - ザイデル法やガウス - ヤコビ法が知られている。両者の繰り返し1回あたりの演算量は同じだが、ガウス - ザイデル法は収束性がよい。
【0107】

シングルキャリア信号をマルチパス環境下で受信する場合に、受信信号に適応等化処理を行うことにより受信感度を大きく向上できることが知られている。イコライザはFIR (Finite Impulse Response: 有限インパルス応答) フィルタによって実現されることが多いが、十分な性能を出すためにはFIRフィルタで用いられる合成重みを精度よく求めることが求められる。また携帯電話のような移動体通信の場合は、特に高速移動時に伝搬環境が急激に変化するため、比較的短時間で精度よく合成重みを求めることも求められる。
【0108】

ところで、最近の半導体製品は動作クロックがあまり向上していない。これは半導体技

10

20

30

40

50

術のかかえる課題ともなっている。そこで、最近では動作クロックを上げる代わりに並列処理数を増やすことにより処理能力を向上させる傾向がある。これは信号処理用に用いられるDSPなどでも同様であり、いかに並列処理を効率的に行うかが、演算効率向上のために求められている。

【0109】

ガウス-ザイデル法などの数値演算のアルゴリズムでは、「かけ算」をした後で「足し算」をする計算が多く行われる。このため、DSPではかけ算と足し算をセットにしたMAC演算器が用いられる場合が多い。

【0110】

ガウス-ザイデル法は比較的収束性のよい近似解法であるが、上述したように式を上から順番に求めることになるため、並列処理には向いていない。つまり、たとえ全ての行を同時に演算できる演算器を持っていたとしても、上から順番に計算しなければならないため、演算器が余ってしまう。すなわち、ガウス-ザイデル法では沢山ある演算器を十分に活用できない可能性がある。一方、ガウス-ヤコビ法は、上述したように、並列演算の効率は高いが、収束性がやや悪い。

10

【0111】

これに対して、上述した各実施の形態によれば、並列演算器を用いた繰り返し演算により1次方程式を解く回路において、繰り返し回数に応じて複数の更新アルゴリズムを切り換えることにより、収束性と並列演算効率との両立を図ることができる。

【0112】

なお、上述した各実施の形態においては、図1A、図1Bに示した演算装置10をイコライザの合成重みを演算する回路に用いる場合について説明したが、演算装置10の適用はこれに限らない。たとえば、演算装置10は、回路シミュレータ、画像処理、CT(Computed Tomography)スキャン画像生成、有限要素法解析など各種の演算に用いることができる。

20

【0113】

上述した各実施の形態に関し、さらに以下の付記を開示する。

【0114】

(付記1) 並列演算器を用いた繰り返し演算により一次方程式の近似解を得る演算部と、前記演算部による前記繰り返し演算を、前記繰り返し演算の繰り返し回数に応じて、第1アルゴリズムと、前記第1アルゴリズムより発散しやすく並列演算効率が高い第2アルゴリズムと、に交互に切り換える切換部と、を備えることを特徴とする演算装置。

30

【0115】

(付記2) 前記第2アルゴリズムは、前記第1アルゴリズムより反復行列のスペクトル半径が大きい更新アルゴリズムであることを特徴とする付記1に記載の演算装置。

【0116】

(付記3) 前記第2アルゴリズムは、前記第1アルゴリズムより同時実行が可能な演算の数が大きい更新アルゴリズムであることを特徴とする付記1または2に記載の演算装置。

【0117】

(付記4) 前記第1アルゴリズムはガウス-ザイデル法の更新アルゴリズムであり、前記第2アルゴリズムはガウス-ヤコビ法の更新アルゴリズムであることを特徴とする付記1~3のいずれか一つに記載の演算装置。

40

【0118】

(付記5) 前記切換部は、前記第1アルゴリズムへ周期的に切り換えることを特徴とする付記1~4のいずれか一つに記載の演算装置。

【0119】

(付記6) 前記第1アルゴリズムは、前記近似解を示す1つのベクトルに対応する演算式を用いる更新アルゴリズムであり、

前記第2アルゴリズムは、前記近似解を示すベクトルを分割した複数のベクトルに対応

50

する各演算式を用いる更新アルゴリズムである、

ことを特徴とする付記 1 ~ 5 のいずれか一つに記載の演算装置。

【 0 1 2 0 】

(付記 7) 前記切換部は、前記第 2 アルゴリズムを用いた場合に前記近似解が常に収束する条件を前記一次方程式が満たしているか否かを判定し、前記条件を前記一次方程式が満たしている期間は、前記繰り返し回数に関わらず前記第 2 アルゴリズムに維持することを特徴とする付記 1 ~ 6 のいずれか一つに記載の演算装置。

【 0 1 2 1 】

(付記 8) 無線通信システムにおける無線信号の等化処理を行うイコライザの重みを演算する演算装置が、並列演算器を用いた繰り返し演算により一次方程式の近似解を得る演算方法であって、

前記繰り返し演算を、前記繰り返し演算の繰り返し回数に応じて、第 1 アルゴリズムと、前記第 1 アルゴリズムより発散しやすく並列演算効率が高い第 2 アルゴリズムと、に交互に切り換えることを特徴とする演算方法。

【 0 1 2 2 】

(付記 9) 並列演算器を用いた繰り返し演算により一次方程式の近似解を得る演算装置であり、前記繰り返し演算を、前記繰り返し演算の繰り返し回数に応じて、第 1 アルゴリズムと、前記第 1 アルゴリズムより発散しやすく並列演算効率が高い第 2 アルゴリズムと、に交互に切り換える演算装置と、

前記演算装置によって得られた前記近似解に基づく重みにより無線信号の等化処理を行うイコライザと、

を備えることを特徴とする無線通信装置。

【符号の説明】

【 0 1 2 3 】

1 0 , 2 0 演算装置

1 1 演算部

1 2 切換部

2 1 更新部セレクタ

2 2 ガウス - ザイデル更新部

2 3 ガウス - ヤコビ更新部

2 4 並列 M A C 演算部

2 5 重み保持部

4 1 , 9 0 式

4 2 , 4 3 , 7 1 , 7 2 , 1 1 1 演算結果

5 0 D S P

5 1 制御回路

5 2 ~ 5 4 M A C 演算器

5 5 メモリ

6 1 , 6 2 , 1 0 0 切り換え順序

8 1 変形ガウス - ザイデル更新部

1 2 1 絶対収束判定部

10

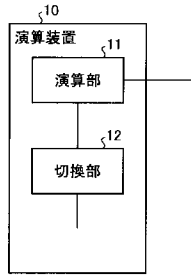
20

30

40

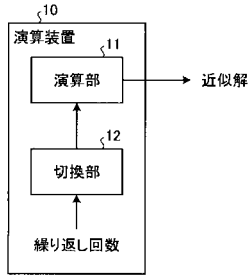
【 図 1 A 】

実施の形態1にかかる演算装置の一例を示す図



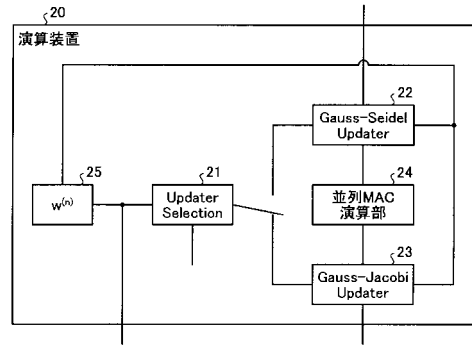
【 図 1 B 】

図1Aに示した演算装置における信号の流れの一例を示す図



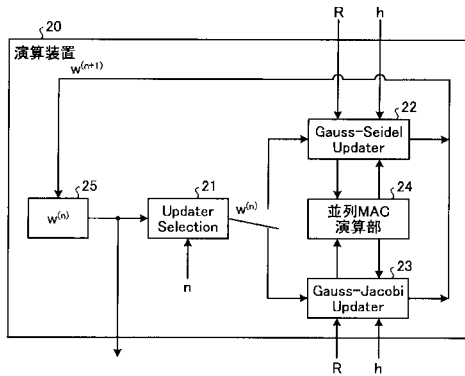
【 図 2 A 】

実施の形態2にかかる演算装置の一例を示す図



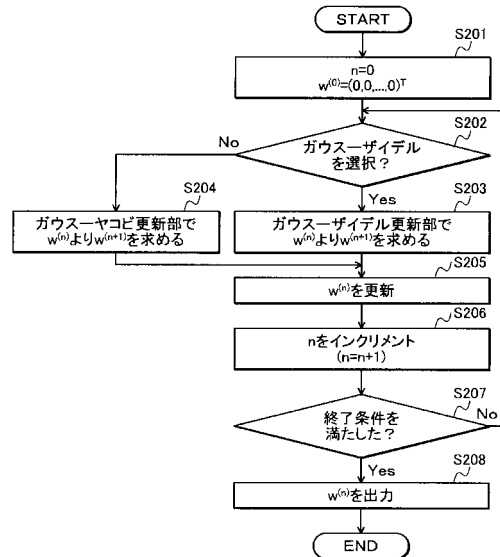
【 図 2 B 】

図2Aに示した演算装置における信号の流れの一例を示す図

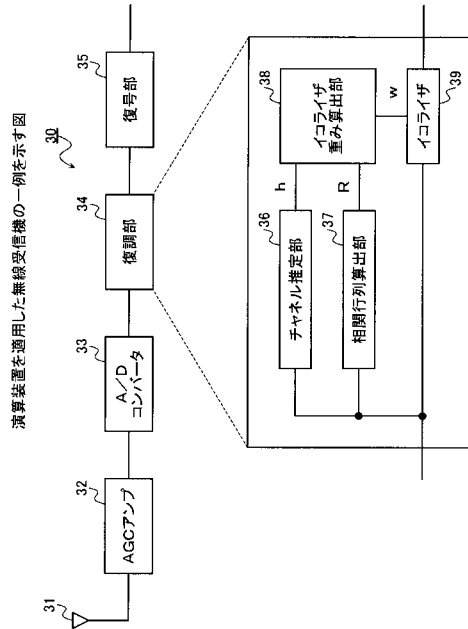


【 図 2 C 】

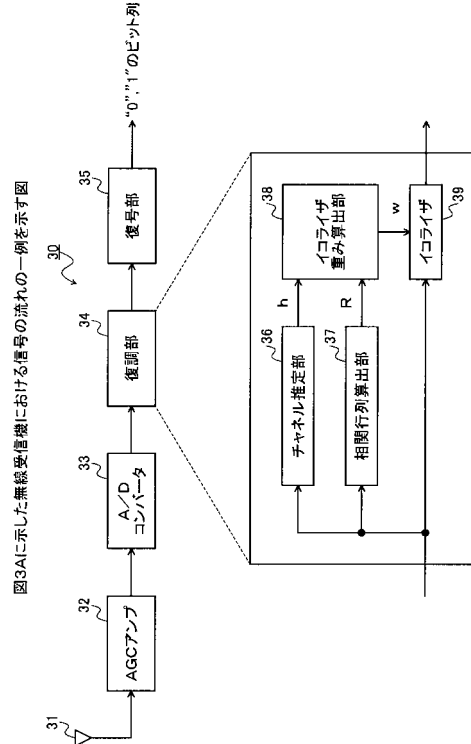
演算装置の動作の一例を示すフローチャート



【図3A】

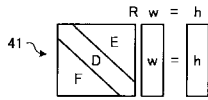


【図3B】



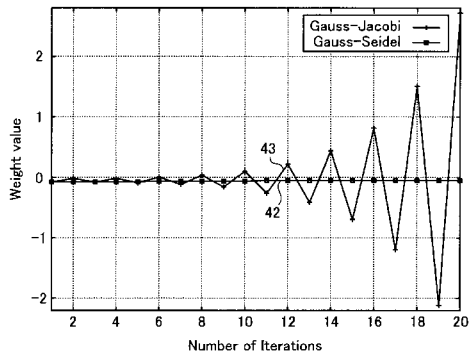
【図4A】

一次方程式のブロック分割の一例を示す図



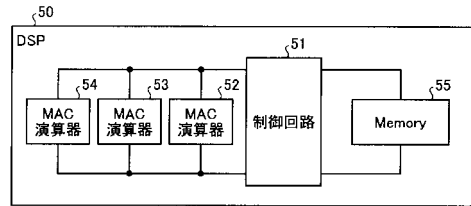
【図4B】

各アルゴリズムにおける近似解の一例を示す図



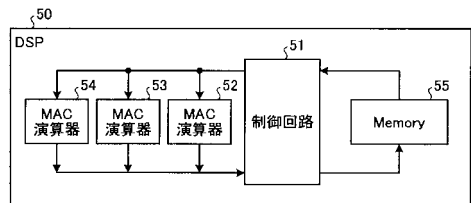
【図5A】

演算装置のハードウェア構成の一例を示す図



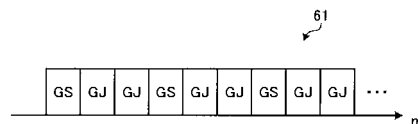
【図5B】

図5Aに示した演算装置のハードウェア構成における信号の流れの一例を示す図



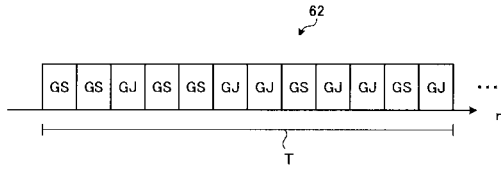
【図6A】

更新アルゴリズムの切り換えの一例を示す図



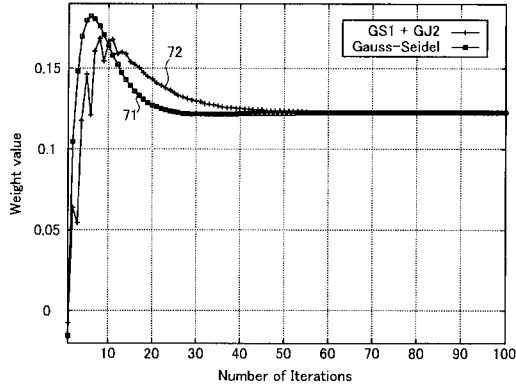
【 図 6 B 】

更新アルゴリズムの切り換えの別の例を示す図



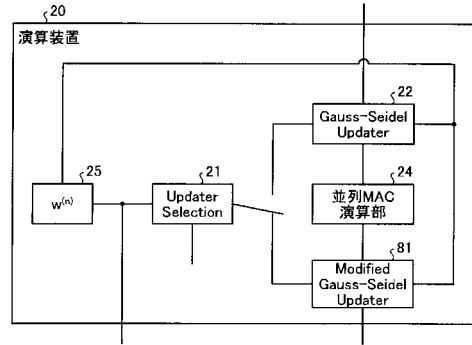
【 図 7 】

ガウス-ザイデル法とガウス-ヤコビ法とを切り換える場合の近似解の一例を示す図



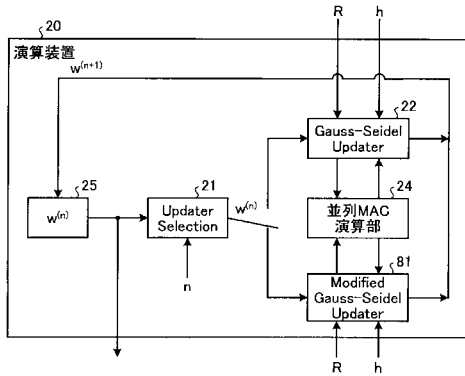
【 図 8 A 】

実施の形態3にかかる演算装置の一例を示す図



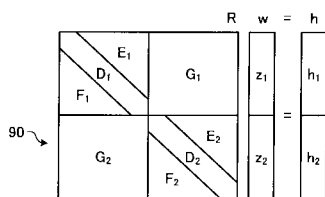
【 図 8 B 】

図8Aに示した演算装置における信号の流れの一例を示す図



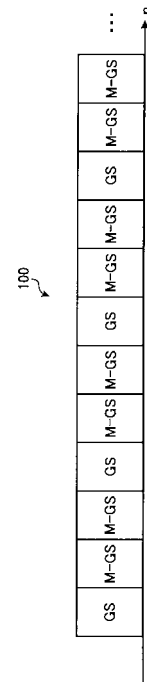
【 図 9 】

一次方程式のブロック分割の一例を示す図



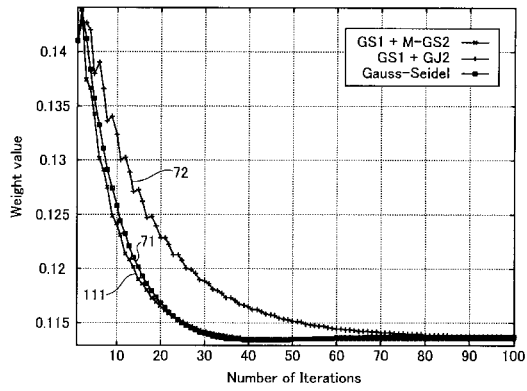
【 図 1 0 】

更新アルゴリズムの切り換えの一例を示す図



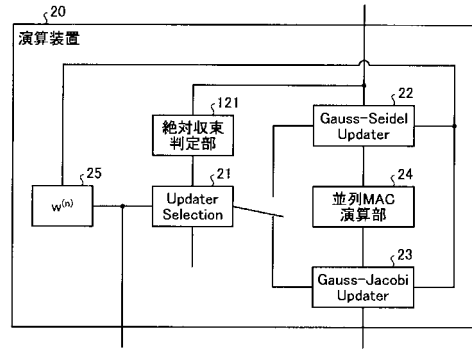
【 図 1 1 】

ガウス-ザイデル法と変形ガウス-ザイデル法とを切り換える場合の近似解の一例を示す図



【 図 1 2 A 】

実施の形態4にかかる演算装置の一例を示す図



【 図 1 2 B 】

図12Aに示した演算装置における信号の流れの一例を示す図

