

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第3915601号

(P3915601)

(45) 発行日 平成19年5月16日(2007.5.16)

(24) 登録日 平成19年2月16日(2007.2.16)

(51) Int. Cl.	F I	
HO4L 27/36 (2006.01)	HO4L 27/00	F
HO3F 1/32 (2006.01)	HO3F 1/32	
HO4B 1/04 (2006.01)	HO4B 1/04	R
HO4B 1/26 (2006.01)	HO4B 1/26	H

請求項の数 10 (全 34 頁)

(21) 出願番号	特願2002-158885 (P2002-158885)	(73) 特許権者	000002130
(22) 出願日	平成14年5月31日(2002.5.31)		住友電気工業株式会社
(65) 公開番号	特開2004-7139 (P2004-7139A)		大阪府大阪市中央区北浜四丁目5番33号
(43) 公開日	平成16年1月8日(2004.1.8)	(74) 代理人	100064746
審査請求日	平成17年1月17日(2005.1.17)		弁理士 深見 久郎
		(74) 代理人	100085132
			弁理士 森田 俊雄
		(74) 代理人	100083703
			弁理士 仲村 義平
		(74) 代理人	100091409
			弁理士 伊藤 英彦
		(74) 代理人	100099922
			弁理士 甲田 一幸

最終頁に続く

(54) 【発明の名称】 前置型非線形歪補償器

(57) 【特許請求の範囲】

【請求項1】

信号変換器の逆特性を用いて、入力信号を変換し、変換後の信号を信号変換器に供給する変換用サポートベクトルマシンと、

前記信号変換器で変換された変換出力信号を受け、前記変換出力信号列により構成される信号ベクトルと、過去に抽出された変換出力信号列により構成される信号ベクトルとに基づいて、前記変換用サポートベクトルマシンにおける変換後の入力信号を、第1の不感帯を有する損失関数に基づいて推定する推定用サポートベクトルマシンと、

前記変換用サポートベクトルマシンおよび前記推定用サポートベクトルマシンに接続され、前記変換用サポートベクトルマシンの出力および前記推定用サポートベクトルマシンの出力に基づいて、前記変換用サポートベクトルマシンおよび前記推定用サポートベクトルマシンの係数を更新するための係数更新手段とを含み、

前記係数更新手段は、

前記変換出力信号を受け、新たに受けた前記変換出力信号の信号列により構成されるベクトルであって、かつ、第2の不感帯の外にあるベクトルをサポートベクトルの候補として抽出し、既にサポートベクトルとして抽出されているベクトルの集合に加え、前記ベクトルの集合を新たにサポートベクトルの集合とみなし、前記ベクトルの集合の中から、前記第1の不感帯の外にあるベクトルをサポートベクトルとして再抽出する抽出手段と、

前記抽出されたサポートベクトルに対する係数を計算する計算手段とを有する、前置型非線形歪補償器。

10

20

**【請求項 2】**

前記第 2 の不感帯の大きさは、前記第 1 の不感帯の大きさと等しい、請求項 1 記載の前置型非線形歪補償器。

**【請求項 3】**

前記第 2 の不感帯は、前記第 1 の不感帯に対する拡張不感帯である、請求項 1 記載の前置型非線形歪補償器。

**【請求項 4】**

前記抽出手段は、サポートベクトルの係数の絶対値が、所定のしきい値を下回った場合には、当該ベクトルをサポートベクトルとして再抽出しない、請求項 1 ~ 3 のいずれか 1 項に記載の前置型非線形歪補償器。

10

**【請求項 5】**

前記係数更新手段は、各学習データセットとして与えられる前記信号変換器の入出力信号のうち、前記変換出力信号の信号列を前記推定用サポートベクトルマシンにより変換して得られる出力データに前記不感帯の片側幅を加減して得られた 2 つの不感帯境界値のうち、当該出力データに近い方の不感帯境界値を前記信号変換器入力信号から減算した値を損失と定義するとき、前記ベクトルが不感帯の外にあるか否かを判定する際に、損失の符号の変化および損失の変化量に基づいて判定する、請求項 1 ~ 3 のいずれか 1 項に記載の前置型非線形歪補償器。

**【請求項 6】**

前記新しいサポートベクトルの追加の頻度が低下するにつれて前記第 2 の不感帯を規定するしきい値を前記第 1 の不感帯を規定するしきい値に漸近させる、請求項 3 記載の前置型非線形歪補償器。

20

**【請求項 7】**

前記係数更新手段は、前記サポートベクトルを抽出する処理を計測する計測手段をさらに含み、

前記計測手段の計測結果に応じて、前記第 2 の不感帯を規定するしきい値を前記第 1 の不感帯を規定するしきい値に漸近させる、請求項 6 記載の前置型非線形歪補償器。

**【請求項 8】**

前記推定用サポートベクトルマシンは、前記信号変換器で変換された変換出力信号を受け、実数信号推定用サポートベクトルマシンを含み、

30

前記変換用サポートベクトルマシンは、

前記入力信号の実数部および虚数部を第 1 および第 2 のノードに受ける、実数信号変換用サポートベクトルマシンと、

前記入力信号の虚数部および極性を反転させた実数部を第 3 および第 4 のノードに受ける、虚数信号変換用サポートベクトルマシンとを含む、請求項 1 記載の前置型非線形歪補償器。

**【請求項 9】**

前記入力信号、前記変換用サポートベクトルマシンの出力および前記信号変換器の出力は、中間周波数信号である、請求項 1 に記載の前置型非線形歪補償器。

**【請求項 10】**

40

前記信号変換器は、増幅器である、請求項 1 ~ 9 のいずれかに記載の前置型非線形歪補償器。

**【発明の詳細な説明】****【0001】****【発明の属する技術分野】**

本発明は、前置型非線形歪補償器に関し、特に、サポートベクトルマシン（以下「SVM」という）を回帰分析に用いた前置型非線形歪補償器に関する。

**【0002】****【従来の技術】**

従来、通信系の非線形歪補償を実現するために、各種デジタル回路による前置型非線形歪

50

補償器が作成されている。デジタル回路による前置型非線形歪補償器として代表的なものにルックアップテーブルを用いたものと、増幅器の逆関数を推定するものがある。

【0003】

ルックアップテーブルを用いた前置型非線形歪補償器は、増幅器の特性を入力信号の振幅値に対して微小区間に分け、その区間の代表値をテーブルに格納しておき、送信信号とそれに対応するテーブルの補償値を乗算して出力信号とするものである。

【0004】

一方、増幅器の逆関数を推定する前置型非線形歪補償器として、`pth-order-predistorter`と呼ばれるものがある。この前置型非線形歪補償器は、信号変換器の逆関数を複素多項式近似するものである。複素多項式は、非線形関数の表現形式として最も一般的なボルテラ級数より導出され、歪補償が行なわれる。

10

【0005】

しかし、ルックアップテーブルを用いた前置型非線形歪補償器では、テーブル内の値は離散値をとるため、値の不連続性により、フロアノイズが上昇する。フロアノイズを解消するためには、データ数を増やす必要があるが、データ数を増やすとメモリ容量が多くなるという問題がある。

【0006】

`pth-order-predistorter`では、計算精度を上げるために複素多項式の次数を増やす必要がある。このため、計算時間がかかり、解が収束しにくいという問題がある。

20

【0007】

さらに、広帯域信号に対しては、ルックアップテーブル、`pth-order-predistorter`ともに、信号帯域内の各周波数ごとに振幅、位相の非線形性を補償する必要があるため、ハードウェア化が困難であるという問題がある。

【0008】

【発明が解決しようとする課題】

ところで、近年、入出力観測データからプロセスのモデリングを行うという問題は、非線形回帰問題やシステム同定などの多くの分野において研究の目的となっている。上記前置型非線形歪補償器の最適な特性を得るという問題も、ひずみを補償する対象となる増幅器を未知の非線形システムとみなせば、この補償する対象となる増幅器の特性の逆関数のモデリングの問題に帰着される。

30

【0009】

このようなプロセスモデリングの問題に取り組む方法は、大きく分けてグローバルモデリングとローカルモデリングの2つに分けることができる。

【0010】

ローカルモデリングは、重み平均法、局所重み回帰法やデータベース学習であるJust-in-time法などがある。

【0011】

これに対して、グローバルモデリングは、データベースの全てに適合する一つの関数を作る手法である。このような関数の作成は、ニューラルネットワークモデリングや非線形統計回帰の他の方法で使われている手法である。

40

【0012】

このようなグローバルモデリングに対する統計的学習理論は、1960年代から存在していたが、実用的なツールとして用いられるようになったのは、1990年半ば以降にV.N. Vapnik等によって、以下の文献に開示されるようなサポートベクトルマシンが提案されたことを契機とする。

【0013】

文献1：V.N.Vapnik：The Nature of Statistical Learning Theory, Springer-Verlag, New York(1995)

文献2：V.N.Vapnik：Statistical Learning Theory, J.Willey & Sons, New York(1998)

50

)

文献3 : B.Scholkopf et al.ed, "Advances in Kernel Methods", The MIT Press, 1999.  
 このようなサポートベクトルマシンを用いた統計的学習理論によるプロセスモデリングには、一般的に、以下のような特徴がある。

【0014】

(1) 他の手法に比べると、収束速度に勝る。

(2) 常に唯一の解を得ることができる。

【0015】

(3) さまざまな学習問題に容易に適用できる。

(4) 問題の複雑さを問題の次元とは独立であるサポートベクトルの個数によって特徴づけているために、いわゆる「次元の呪い (curse of dimensionality)」から逃れることができる。

【0016】

さらに、叙述したサポートベクトルマシンは、回帰問題の学習に用いることのできるサポートベクトル回帰 (Support Vector Regression (以下、「SVR」と呼ぶ)) に拡張されている。このようなSVRについては、以下の文献に開示がある。

【0017】

文献4 : N.Cristianini, J.Shawe-Taylor "An Introduction to Support Vector Machines", Cambridge University Press, 2000.

さらに、このようなサポートベクトルマシンを上述した前置型非線形歪補償器に適用する試みが、以下の文献に開示されている。

【0018】

文献5 : 伊藤隆徳, 大森浩充, 佐野昭 B-5-124 「サポートベクトル法を用いた適応非線形歪補償」2001年電子情報通信学会総合大会予稿集, p.522, 2001年  
 しかしながら、サポートベクトルマシンをデジタル信号受信装置の前置型非線形歪補償器に用いる場合には、以下のような点を考慮する必要がある。

【0019】

(1) 一般に、デジタル信号の変調においては、変調後の信号が、2次元のコンスタレーション上の点として表現される。このため、信号は、複素数として扱われる必要がある。

【0020】

(2) 前置型非線形歪補償器として使用するためには、大量のデータを用いてSVRを使い回帰問題を解くことになるため、上記のような複素数に対する問題を実時間で効率よく解くことができるアルゴリズムが必要である。

【0021】

しかしながら、従来のサポートベクトルマシンを用いた手法では、上記の点について、いかなる解決を図るべきかが、必ずしも明確とは言えないという問題があった。

【0022】

本発明は、上述の課題を解決するためになされたもので、その目的は、デジタル信号に対して実時間で効率よく解くことができるアルゴリズムに基づいて、広帯域な歪補償が可能な前置型非線形歪補償器を提供することである。

【0023】

本発明の他の目的は、処理対象となる信号が複素数で表現される場合でも、高速に歪補償のための演算を行うことが可能な前置型非線形歪補償器を提供することである。

【0024】

【課題を解決するための手段】

本発明のある局面に従う前置型非線形歪補償器は、信号変換器の逆特性を用いて、入力信号を変換し、変換後の信号を信号変換器に供給する変換用サポートベクトルマシンと、信号変換器で変換された変換出力信号を受け、変換出力信号列により構成される信号ベクトルと、過去に抽出された変換出力信号列により構成される信号ベクトルとに基づいて、変換用サポートベクトルマシンにおける変換後の入力信号を、第1の不感帯を有する損失関

10

20

30

40

50

数に基づいて推定する推定用サポートベクトルマシンと、変換用サポートベクトルマシンおよび推定用サポートベクトルマシンに接続され、変換用サポートベクトルマシンの出力および推定用サポートベクトルマシンの出力に基づいて、変換用サポートベクトルマシンおよび推定用サポートベクトルマシンの係数を更新するための係数更新手段とを含み、係数更新手段は、変換出力信号を受け、新たに受けた変換出力信号の信号列により構成されるベクトルであって、かつ、第2の不感帯の外にあるベクトルをサポートベクトルの候補として抽出し、既にサポートベクトルとして抽出されているベクトルの集合に加え、ベクトルの集合を新たにサポートベクトルの集合とみなし、ベクトルの集合の中から、第1の不感帯の外にあるベクトルをサポートベクトルとして再抽出する抽出手段と、抽出されたサポートベクトルに対する係数を計算する計算手段とを有する。

10

【0025】

好ましくは、第2の不感帯の大きさは、第1の不感帯の大きさと等しい。  
あるいは、好ましくは、第2の不感帯は、第1の不感帯に対する拡張不感帯である。

【0026】

さらに、好ましくは、第2の不感帯は、随時更新されるベクトルの集合に対応して縮小するように更新される。

【0027】

好ましくは、抽出手段は、サポートベクトルの係数の絶対値が、所定のしきい値を下回った場合には、当該ベクトルをサポートベクトルとして再抽出しない。

【0028】

20

好ましくは、係数更新手段は、各学習データセットとして与えられる信号変換器の入出力信号のうち、変換出力信号の信号列を推定用サポートベクトルマシンにより変換して得られる出力データに不感帯の片側幅を加減して得られた2つの不感帯境界値のうち、当該出力データに近い方の不感帯境界値を信号変換器入力信号から減算した値を損失と定義するとき、ベクトルが不感帯の外にあるか否かを判定する際に、損失の符号の変化および損失の変化量に基づいて判定する。

【0029】

好ましくは、新しいサポートベクトルの追加の頻度が低下するにつれて第2の不感帯を規定するしきい値を不感帯を規定するしきい値に漸近させる。

【0030】

30

好ましくは、係数更新手段は、サポートベクトルを抽出する処理を計測する計測手段をさらに含み、計測手段の計測結果に応じて、第2の不感帯を規定するしきい値を第1の不感帯を規定するしきい値に漸近させる。

【0031】

好ましくは、推定用サポートベクトルマシンは、信号変換器で変換された変換出力信号を受ける、実数信号推定用サポートベクトルマシンを含み、変換用サポートベクトルマシンは、入力信号の実数部および虚数部を第1および第2のノードに受ける、実数信号変換用サポートベクトルマシンと、入力信号の虚数部および極性を反転させた実数部を第3および第4のノードに受ける、虚数信号変換用サポートベクトルマシンとを含む。

【0032】

40

好ましくは、入力信号、変換用サポートベクトルマシンの出力および信号変換器の出力は、中間周波数信号である。

【0033】

好ましくは、信号変換器は、増幅器である。

【0034】

【発明の実施の形態】

以下、図面を参照しつつ本発明の実施の形態について詳細に説明する。以下の説明では、同一の構成部分には同一の符号を付してあり、それらの名称および機能も同じである。したがって、それらについての詳細な説明は繰り返さない。

【0035】

50

[ サポートベクトルマシンによる回帰問題の学習の一般論 ]

以下、本発明の説明を行うにあたり、その前提となるサポートベクトルマシンによる回帰問題の学習について、まとめておく。

【 0 0 3 6 】

1 - 1 . 線形回帰問題の学習

1 - 1 - 1 線形回帰問題

まず、サポートベクトルマシン ( S V M ) の考えに基づき、サポートベクトル回帰 ( Support Vector Regression : SVR ) の立場から線形回帰問題を解く方法について述べる。この問題は、以下のように定式化できる。

【 0 0 3 7 】

[ 問題 ]

m 入力 1 出力の未知線形回帰  $f : x \rightarrow y$  があり、トレーニングデータ

$(x_1, y_1), \dots, (x_n, y_n) \quad x_i \in \mathbb{R}^m, y_i \in \mathbb{R} \quad (1.1)$

が与えられているとする。このとき、トレーニングデータを用いてこの線形回帰を近似する関数

$f(x, w) = w^T x \quad (1.2)$

を求める。

【 0 0 3 8 】

1 - 1 - 2 損失関数

ここで、線形 不感損失関数 ( Linear -Insensitive Loss Function  $L_1(x, y, f(x, w))$  ) ならびに二次 不感損失関数 ( Quadratic -Insensitive Loss Function  $L_2(x, y, f(x, w))$  ) を次のように定義する。 20

【 0 0 3 9 】

【 数 1 】

$$\begin{aligned} L_1^{\epsilon}(x, y, f(x, w)) &= |y - f(x, w)|_{\epsilon} \\ &= \max(0, |y - f(x, w)| - \epsilon) \end{aligned} \quad (1.3)$$

$$\begin{aligned} L_2^{\epsilon}(x, y, f(x, w)) &= |y - f(x, w)|_{\epsilon}^2 \\ &= (\max(0, |y - f(x, w)| - \epsilon))^2 \end{aligned} \quad (1.4) \quad 30$$

【 0 0 4 0 】

図 1 は、このような線形 不感損失関数および二次 不感損失関数を示す図である。図 1 ( a ) は、線形 不感損失関数であり、図 1 ( b ) は、二次 不感損失関数を示す。

【 0 0 4 1 】

1 - 1 - 3 二次 不感損失関数による回帰問題

以下では、二次 不感損失を用いた場合のサポートベクトル回帰について考える。サポートベクトルマシンに基づいた主問題は、以下のように表現できる。 40

【 0 0 4 2 】

【 数 2 】

$$\min_w \|w\|^2 + C \sum_{i=1}^n (\max(0, |y_i - f(x_i, w)| - \epsilon))^2 \quad (1.5)$$

【 0 0 4 3 】

この主題は、次のように書き換えることができる。

【 0 0 4 4 】

10

20

30

40

50

【数3】

スラック変数  $\xi_i, \hat{\xi}_i (\geq 0)$  等を導入すると、

$$\min_{\mathbf{w}} \|\mathbf{w}\|^2 + C \sum_{i=1}^n (\xi_i^2 + \hat{\xi}_i^2) \quad (1.6)$$

$$\text{subject to} \quad -y_i + \mathbf{w}^T \mathbf{x}_i \leq \varepsilon + \xi_i \quad i=1, \dots, n \quad (1.7)$$

$$y_i - \mathbf{w}^T \mathbf{x}_i \leq \varepsilon + \hat{\xi}_i \quad i=1, \dots, n \quad (1.8)$$

$$\hat{\xi}_i \geq 0 \quad \xi_i \geq 0 \quad i=1, \dots, n \quad (1.9)$$

10

【0045】

ただし、 $C$  はモデルの複雑さと制約条件の緩和（トレーニング誤差）との間のトレードオフを設定する正則化パラメータである。

【0046】

【数4】

ラグランジュ乗数  $\alpha_i, \hat{\alpha}_i \geq 0$  を導入すると、この問題のラグラン

ジュ関数は以下のようになる。

20

$$L = \frac{1}{2} \|\mathbf{w}\|^2 + \frac{C}{2} \sum_{i=1}^n (\xi_i^2 + \hat{\xi}_i^2) \\ + \sum_{i=1}^n \alpha_i (\mathbf{w}^T \mathbf{x}_i - y_i - \varepsilon - \xi_i) + \sum_{i=1}^n \hat{\alpha}_i (y_i - \mathbf{w}^T \mathbf{x}_i - \varepsilon - \hat{\xi}_i)$$

$$\text{subject to} \quad \alpha_i, \hat{\alpha}_i \geq 0 \quad i=1, \dots, n$$

(1.10)

30

【0047】

式(1.10)に対するKarush-Kuhn-Tucker(KKT)条件により、以下の関係が成り立つ。

【0048】

【数5】

$$\frac{\partial L}{\partial \mathbf{w}} = \mathbf{w} + \sum_{i=1}^n (\alpha_i - \hat{\alpha}_i) \mathbf{x}_i = 0 \quad i=1, \dots, n \quad (1.11)$$

$$\frac{\partial L}{\partial \xi_i} = C \xi_i - \alpha_i = 0 \quad i=1, \dots, n \quad (1.12)$$

40

$$\frac{\partial L}{\partial \hat{\xi}_i} = C \hat{\xi}_i - \hat{\alpha}_i = 0 \quad i=1, \dots, n \quad (1.13)$$

【0049】

これらの関係式を式(1.10)に代入すると次の双村問題に書き換えられる。

【0050】

【数6】

$$\max_{\alpha} \sum_{i=1}^n y_i (\hat{\alpha}_i - \alpha_i) - \varepsilon \sum_{i=1}^n (\hat{\alpha}_i + \alpha_i) - \frac{1}{2} \sum_{i,j=1}^n (\hat{\alpha}_i - \alpha_i)(\hat{\alpha}_j - \alpha_j)(x_i^T x_j + \frac{1}{C} \delta_{ij}) \quad (1.14)$$

## 【0051】

ここで、係数  $\alpha_i$  を以下のように置き換えると、式(1.14)は、最終的に以下の式(1.15)のように書き換えることができる。また、式(1.11)により、線形回帰を近似する関数  $f(x, w) = w^T x$  は、以下の式(1.16)のようになる。

10

## 【0052】

## 【数7】

係数  $\alpha_i \leftarrow \hat{\alpha}_i - \alpha_i$  の置き換えを行い、さらに、 $\hat{\alpha}_i \alpha_i = 0$  の関係を用いると、

$$\max_{\alpha} \sum_{i=1}^n y_i \alpha_i - \varepsilon \sum_{i=1}^n |\alpha_i| - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j (x_i^T x_j + \frac{1}{C} \delta_{ij}) \quad (1.15)$$

20

$$f(x, \alpha) = \sum_{i=1}^n \alpha_i x_i^T x \quad (1.16)$$

## 【0053】

図2は、線形近似関数とサポートベクトルとの関係を示す概念図である。

図2において、「 $\alpha_i$ 」がサポートベクトルであり、これらの近似関数による誤差の大きさは  $\alpha_i$  よりも大きい。「 $\alpha_j$ 」は近似関数による誤差の大きさが  $\alpha_j$  よりも小さいデータであり、これらは直接近似関数には影響しない。

## 1-2 非線形回帰問題の学習

30

## 1-2-1 非線形回帰問題

以上の説明では、サポートベクトル回帰(SVR)を用いた線形回帰について説明した。以下では、それを応用したSVRを用いた非線形回帰について説明する。この非線形回帰問題は、以下のように定式化できる

## [問題]

$m$ 入力1出力の未知非線形回帰  $f: x \rightarrow y$  があり、トレーニングデータ

$$(x_1, y_1), \dots, (x_n, y_n) \quad x_i \in \mathbb{R}^m, y_i \in \mathbb{R} \quad (1.17)$$

が与えられているとする。このとき、トレーニングデータを用いてこの非線形回帰を近似する関数  $f(x, w) = w^T x$  を求める。ここでは、損失関数として、二次不感損失関数を用いることにする。

40

## 1-2-2 特徴空間

一般的に複雑な問題に応用するには、入力空間内で考えるよりも表現能力が優れている空間で考える必要がある。このような手法は高次元化と呼ばれている。

## 【0054】

ここで、非線形回帰問題に対応するために、入力空間から高次元特徴空間への非線形写像を導入する。入力ベクトル  $x$  を入力空間  $X$  から特徴空間  $F$  へ写像し、その特徴空間でのベクトル  $\phi(x)$  をこれまでの入力ベクトル  $x$  とみなすことにより入力空間ではなく特徴空間で線形回帰問題を取り扱う。ただし、 $\phi: X \rightarrow F$  は、入力空間からある特徴空間への非線形写像である。

## 【0055】

50



この作業は、実質的には元の入力空間において非線形回帰問題を解いていることになる。

【 0 0 5 6 】

ヒルベルトシュミット理論によるとヒルベルト空間における内積は次のような等価表現ができる。

【 0 0 5 7 】

【 数 8 】

$$(\phi(x_1), \phi(x_2)) = \sum_{i=1}^n a_i \phi_i(x_1) \phi_i(x_2) \leftrightarrow K(x_1, x_2) \quad (1.18)$$

$$a_i \geq 0 \quad (1.19)$$

$$K(x_i, x_j) = \phi(x_i)^T \phi(x_j) \quad (1.20)$$

【 0 0 5 8 】

以下に示すとおり、特徴空間で線形問題を考えると、式(1.21)となり、また、線形回帰を近似する関数  $f(x, w) = w^T \phi(x)$  は、式(1.22)のようになる。この式(1.22)は、入力空間における線形回帰問題となっており、式(1.20)の関係により、最終的に式(1.23)となる。さらに、式(1.23)をベクトル表記すると、式(1.24)となり、この非線形回帰問題を解くと、式(1.25)の非線形近似関数を得ることができる。

【 0 0 5 9 】

【 数 9 】

$$\max_{\alpha} \sum_{i=1}^n y_i \alpha_i - \varepsilon \sum_{i=1}^n |\alpha_i| - \frac{1}{2} \sum_{i,j=1}^n (\alpha_i \alpha_j \phi(x_i)^T \phi(x_j) + \frac{1}{C} \delta_{ij}) \quad (1.21)$$

$$f(\phi(x), \alpha) = \sum_{i=1}^n \alpha_i \phi(x_i)^T \phi(x) \quad (1.22)$$

$$\max_{\alpha} \sum_{i=1}^n y_i \alpha_i - \varepsilon \sum_{i=1}^n |\alpha_i| - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j (K(x_i, x_j) + \frac{1}{C} \delta_{ij}) \quad (1.23)$$

$$\max_{\alpha} y^T \alpha - \varepsilon \|\alpha\|_1 - \frac{1}{2} \alpha^T \left( K + \frac{I}{C} \right) \alpha \quad (1.24)$$

$$f(x, \alpha) = \sum_{i=1}^n \alpha_i K(x, x_i) \quad (1.25)$$

ただし、出力  $y = [y_1, \dots, y_n]^T$  , 係数  $\alpha = [\alpha_1, \dots, \alpha_n]^T$  ,

カーネル行列  $K: K$  の  $i$  行  $j$  列目の要素  $K_{ij} = K(x_i, x_j)$  である。

【 0 0 6 0 】

ここで、注目すべき点は、特徴空間内での内積計算を入力空間内での計算に置き換えることができるカーネル関数  $K(x_i, x_j)$  を用いることにより、双対問題の式(1.21)やそれを解くことによって得られる近似関数には  $\phi(x)$  が含まれないので、実際には入力空間から特徴空間への非線形写像を計算する必要はない点である。これは、「カーネルトリック」と呼ばれている。

【 0 0 6 1 】

このような特徴があるため、高次元化を導入することにより、サポートベクトル回帰(SVR)では特徴空間の次元を非常に大きくとることもでき、さらにモデルの複雑さと特徴空間の次元とを独立に制御することが可能となる。このことより、SVRはいわゆる次元の呪いから逃れることができる。

10

20

30

40

50

## 【 0 0 6 2 】

図 3 は、非線形近似関数とサポートベクトルを示す概念図である。図 3 でも、「 $\xi_i$ 」がサポートベクトルであり、これらの近似関数による誤差の大きさは  $\xi_i$  よりも大きい。「 $\hat{\xi}_i$ 」は近似関数による誤差の大きさが  $\xi_i$  よりも小さいデータであり、これらは直接近似関数には影響しない。

1 - 2 - 3 サポートベクトルと係数の関係

二次 不感損失関数を用いた S V R 問題では、式 ( 1.14 ) から式 ( 1.15 ) に係数の置換えを行う前においては、式 ( 1.11 ) ~ ( 1.13 ) の K K T 条件が成り立っていることから、係数の置換え後においては結局以下の関係式が成り立つ。

## 【 0 0 6 3 】

【 数 1 0 】

$$\hat{\xi}_i - \xi_i = \frac{\alpha_i}{C} \quad i = 1, \dots, n \quad (1.26)$$

10

## 【 0 0 6 4 】

また、二次 不感損失関数は、不感帯  $\varepsilon$  を有するので、推定誤差の大きさが  $\xi_i$  よりも小さい、以下の式 ( 1.27 ) を満たすデータセット (  $x_i, y_i$  ) はサポートベクトルから外れる必要がある。このため、得られた解によるサポートベクトルの近似誤差の大きさは、式 ( 1.28 ) のようになる。

20

## 【 0 0 6 5 】

【 数 1 1 】

$$\left| y_i - \sum_{j=1}^n \alpha_j K(x_i, x_j) \right| \leq \varepsilon \quad (1.27)$$

$$\left| y_i - \sum_{j=1}^n \alpha_j K(x_i, x_j) \right| \leq \varepsilon \hat{\xi}_i - \xi_i = \frac{\alpha_i}{C} \quad i = 1, \dots, n \quad (1.28)$$

30

## 【 0 0 6 6 】

さらに、ラグランジェ関数を式 ( 1.10 ) のようにおいたので、以下の式 ( 1.29 ) および式 ( 1.30 ) の符号が等しくなる。このため、式 ( 1.23 ) の解であるサポートベクトルの係数  $\alpha_i$  は、式 ( 1.31 ) を満たすことになる。

## 【 0 0 6 7 】

【 数 1 2 】

$$y_i - \sum_{j=1}^n \alpha_j K(x_i, x_j) \quad (1.29)$$

40

$$\hat{\xi}_i - \xi_i = \frac{\alpha_i}{C} \quad (1.30)$$

$$y_i - \sum_{j=1}^n \alpha_j K(x_i, x_j) - \varepsilon \cdot \text{sign}(\alpha_i) = \frac{\alpha_i}{C} \quad i = 1, \dots, n \quad (1.31)$$

## 【 0 0 6 8 】

2 - 1 サポートベクトル回帰の学習に対する本発明の方法

以上の説明の方法をそのまま実行しようとする、S V R 問題においてトレーニングデータが多い場合、問題を解くために非常に多くの時間を費やしてしまう。そこで、本発明に

50

おいては、SVR問題を効率良く解くために、以下に説明するとおり、ニュートン・ラフソン (Newton - Raphason) 法を変形させた基本的な更新式を求め、その更新式を用いた逐次形アルゴリズムを採用する。

### 2 - 1 - 1 問題設定

ここでは、二次 不感損失関数を使用する次の問題を解く方法について説明する。

【0069】

[問題]

m入力 1出力の未知非線形回帰  $f: x \rightarrow y$  があり、トレーニングデータ

$(x_1, y_1), \dots, (x_n, y_n)$   $x_i \in R^m, y_i \in R$

が与えられているとする。このとき、トレーニングデータを用いてこの非線形回帰を近似する式 (2.1) の関数を求めるために、式 (2.2) で表される最大化問題を解く。 10

【0070】

【数13】

$$\text{SVR} : f(x, \alpha) = \sum_{i=1}^n \alpha_i K(x, x_i) \quad (2.1)$$

$$\max_{\alpha} W(\alpha) = \sum_{i=1}^n y_i \alpha_i - \varepsilon \sum_{i=1}^n |\alpha_i| - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j (K(x_i, x_j) + \frac{1}{C} \delta_{ij}) \quad (2.2)$$

20

【0071】

SVRでは、誤差を評価するときには不感帯を設けているので、この問題を解いて得られた  $f(x, \cdot)$  とトレーニングデータには、以下に述べるような関係が成り立っていないなければならない。

【0072】

すなわち、データセット  $(x_i, y_i)$  が不感帯の内側にあるならば、つまり、 $|y_i - f(x_i, \cdot)| \leq \varepsilon$  を満たすならば、係数は  $\alpha_i = 0$  となる。

【0073】

逆に、データセット  $(x_i, y_i)$  が不感帯の外側にあるならば、つまり、 $|y_i - f(x_i, \cdot)| > \varepsilon$  を満たすならば、係数  $\alpha_i$  は、上述した式 (1.31) を満たす。 30

【0074】

以下に述べるサポートベクトル回帰の学習では、上記の関係を考慮して、不必要となったデータセットをサポートベクトルから除外していく方法をとっている。

【0075】

### 2 - 1 - 2 バッチ処理

以下では、はじめに  $n$  個のトレーニングデータからいくつかのデータセットを取り出し、その取り出したデータセットだけを用いてSVRを求める方法について説明する。

【0076】

(ニュートン・ラフソン法の修正)

40

以下、二次 不感損失関数に対する双対問題をニュートン・ラフソン法を変形させて更新式を用いて解く方法について説明する。

【0077】

式 (2.2) の最大化問題をベクトル表記すると、式 (2.3) のようになり、かつ、評価関数  $W(\cdot)$  をテイラー展開すると、式 (2.4) のようになる。

【0078】

その上で、 $W(\cdot)$  を  $\alpha_i$  の近傍で二次関数近似を行ったとして、上式 (2.4) の第3項までで打ち切る。

【0079】

【数14】

50

$$\max_{\alpha} W(\alpha) = \mathbf{y}^T \alpha - \varepsilon \|\alpha\|_1 - \frac{1}{2} \alpha^T \left( \mathbf{K} + \frac{\mathbf{I}}{\mathbf{C}} \right) \alpha \quad (2.3)$$

$$W(\alpha) = W(\alpha^i) + \frac{\partial W}{\partial \alpha^T} (\alpha - \alpha^i) + \frac{1}{2} (\alpha - \alpha^i)^T \frac{\partial^2 W}{\partial \alpha \partial \alpha^T} (\alpha - \alpha^i) + \dots \quad (2.4)$$

【 0 0 8 0 】

この二次関数近似の最小値を与える  $\alpha = \alpha^i$  は、以下の式 (2.5) の条件により与えられ、そのような最小値を与える  $\alpha = \alpha^i$  は、逐次的に以下の式 (2.6) の関係を満たすことになる (ニュートン・ラフソン法)。 10

【 0 0 8 1 】

【 数 1 5 】

$$\frac{\partial W(\alpha^{i+1})}{\partial \alpha} = 0 \quad (2.5)$$

$$\frac{\partial W(\alpha^i)}{\partial \alpha} + \frac{\partial^2 W(\alpha^i)}{\partial \alpha \partial \alpha^T} (\alpha^{i+1} - \alpha^i) = 0$$

$$\alpha^{i+1} = \alpha^i - \left[ \frac{\partial^2 W(\alpha^i)}{\partial \alpha \partial \alpha^T} \right]^{-1} \frac{\partial W(\alpha^i)}{\partial \alpha} \quad (2.6)$$

ただし、

$$\frac{\partial W}{\partial \alpha} = \mathbf{y} - \left( \mathbf{K} + \frac{\mathbf{I}}{\mathbf{C}} \right) \alpha - \varepsilon \cdot \text{sign}(\alpha) \quad (2.7)$$

$$\frac{\partial^2 W}{\partial \alpha \partial \alpha^T} = -\mathbf{K} - \frac{\mathbf{I}}{\mathbf{C}} - \varepsilon \cdot \text{diag}(\text{sign}'(\alpha)) \quad (2.8)$$

$$\text{sign}(\alpha) = [\text{sign}(\alpha_1), \dots, \text{sign}(\alpha_n)]^T, \text{sign}'(\alpha) = [\text{sign}'(\alpha_1), \dots, \text{sign}'(\alpha_n)]^T \quad 30$$

$$\text{sign}(x) \approx \frac{2}{1 + \exp(-ax)} - 1, \quad a \gg 1 \quad (2.9)$$

$$\text{sign}'(x) = \frac{\partial \text{sign}(x)}{\partial x} \quad (2.10)$$

$$\approx 2a \left( \frac{1}{1 + \exp(-ax)} \right) \left( 1 - \frac{1}{1 + \exp(-ax)} \right) \quad (2.11)$$

【 0 0 8 2 】

( の更新方法 (初期値 0 の場合) )

ニュートン・ラフソン法に基づき  $\alpha$  を求める際、初期値  $\alpha^0 = 0$  の場合について考える。式 (2.6) で、 $\alpha^0 = 0$  とすると、以下の式 (2.12) および式 (2.13) を満たすことになるが、本発明においては、 $\alpha^0 = 0$  とおいて、式 (2.14) のように修正する。

【 0 0 8 3 】

【 数 1 6 】

$$\alpha^1 = \alpha^0 + \left( \mathbf{K} + \frac{\mathbf{I}}{\mathbf{C}} + \varepsilon \cdot \text{diag}(\text{sign}'(\alpha^0)) \right)^{-1} \left( \mathbf{y} - \left( \mathbf{K} + \frac{\mathbf{I}}{\mathbf{C}} \right) \alpha^0 - \varepsilon \cdot \text{sign}(\alpha^0) \right) \quad (2.12)$$

$$= \left( \mathbf{K} + \frac{\mathbf{I}}{\mathbf{C}} + \varepsilon \cdot \text{diag}(\text{sign}'(\alpha^0)) \right)^{-1} \mathbf{y} \quad (2.13)$$

$$\alpha^1 = \left( \mathbf{K} + \frac{\mathbf{I}}{\mathbf{C}} \right)^{-1} \mathbf{y} \quad (2.14)$$

10

【 0 0 8 4 】

その後、さらに式(2.6)を用いて更新すると、以下の式(2.15)および(2.16)のようになる。さらに、 $\alpha_i^1 = 0$  のとき、式(2.17)が成り立つので、式(2.16)は、式(2.18)および(2.19)で表されるようになる。

【 0 0 8 5 】

【 数 1 7 】

$$\alpha^2 = \alpha^1 + \left( \mathbf{K} + \frac{\mathbf{I}}{\mathbf{C}} + \varepsilon \cdot \text{diag}(\text{sign}'(\alpha^1)) \right)^{-1} \left( \mathbf{y} - \left( \mathbf{K} + \frac{\mathbf{I}}{\mathbf{C}} \right) \alpha^1 - \varepsilon \cdot \text{sign}(\alpha^1) \right) \quad (2.15)$$

20

$$= \alpha^1 - \varepsilon \left( \mathbf{K} + \frac{\mathbf{I}}{\mathbf{C}} + \varepsilon \cdot \text{diag}(\text{sign}'(\alpha^1)) \right)^{-1} \text{sign}(\alpha^1) \quad (2.16)$$

$$\text{sign}'(\alpha_i^1) = 0 \quad (2.17)$$

$$\alpha^2 = \alpha^1 - \varepsilon \left( \mathbf{K} + \frac{\mathbf{I}}{\mathbf{C}} \right)^{-1} \text{sign}(\alpha^1) \quad (2.18)$$

30

$$= \left( \mathbf{K} + \frac{\mathbf{I}}{\mathbf{C}} \right)^{-1} \left( \mathbf{y} - \varepsilon \cdot \text{sign}(\alpha^1) \right) \quad (2.19)$$

【 0 0 8 6 】

このように求めた係数ベクトル  $\alpha^2$  で、以下の式(2.20)が成り立っている場合、式(2.21)のようになるので、上述した式(1.31)を満たしていることがわかる。

【 0 0 8 7 】

【 数 1 8 】

$$\text{sign}(\alpha^1) = \text{sign}(\alpha^2) \quad (2.20)$$

40

$$\alpha^2 = \left( \mathbf{K} + \frac{\mathbf{I}}{\mathbf{C}} \right)^{-1} \left( \mathbf{y} - \varepsilon \cdot \text{sign}(\alpha^2) \right) \quad (2.21)$$

【 0 0 8 8 】

しかし、求めた解が式(2.20)を満たしていない場合、その解は式(1.31)を満たさなく、解としてはふさわしくない。

【 0 0 8 9 】

(余分なデータセットの除去方法)

50

次に、式(2.20)を満たしていない場合について考えることにする。

【0090】

式(2.20)を満たしていない場合は、求めた解は式(1.31)を満たしてなく、KKT条件を満たさないで解ではないことになる。そこで、このような場合は、余分なデータセット $(x_i, y_i)$ が現在のサポートベクトルの候補の中に存在していると考え、現在のサポートベクトルの候補の中から余分なデータセットを取り除くことにする。

【0091】

以下では、さらに、余分なデータセットの選出方法について以下に述べる。

まずはじめに、先ほどの更新式を用いて、以下の式(2.22)に基づいて、係数ベクトル $n_{sv}$ を求める(初期値: 0)。

【0092】

次に、係数ベクトル $n_{sv}$ の各要素はゼロではない( $i = 0, i = 1, \dots, n_{sv}$ )と考え、さらに更新式(2.6)にステップサイズ( $0 < \eta < 1$ )を導入して、以下の式(2.23)のような新たな更新式をつくる。

【0093】

【数19】

$$\alpha_{n_{sv}} = \left( K_{n_{sv}} + \frac{I_{n_{sv}}}{C} \right)^{-1} y_{n_{sv}} \quad (2.22)$$

$$\alpha_{n_{sv}} \leftarrow \alpha_{n_{sv}} + \eta \left( K_{n_{sv}} + \frac{I_{n_{sv}}}{C} \right)^{-1} \left( y_{n_{sv}} - \left( K_{n_{sv}} + \frac{I_{n_{sv}}}{C} \right) \alpha_{n_{sv}} - \varepsilon \cdot \text{sign}(\alpha_{n_{sv}}) \right) \quad (2.23)$$

【0094】

この更新式を用いて係数ベクトル $n_{sv}$ を更新する際、更新前後で係数ベクトル $n_{sv}$ の各要素の符号が変化しない範囲で更新を行うことにする。

【0095】

次に、ステップサイズ $\eta$ の決め方について説明する。

ステップサイズは、1に近ければ近いほど評価関数が大きな値となるので、できる限り1に近いことが望ましい。そこで、次のようにしてそのステップサイズを求める。

【0096】

まず、以下の式(2.24)により、ステップサイズベクトルを計算する。

【0097】

【数20】

$$\eta = -\alpha_{n_{sv}} / \left( K_{n_{sv}} + \frac{I_{n_{sv}}}{C} \right)^{-1} \left( y_{n_{sv}} - \left( K_{n_{sv}} + \frac{I_{n_{sv}}}{C} \right) \alpha_{n_{sv}} - \varepsilon \cdot \text{sign}(\alpha_{n_{sv}}) \right) \quad (2.24)$$

ただし、記号 $/$ は、例えば $A/B$ の演算の場合に、行列 $A$ の

各要素を対応する行列 $B$ の各要素で割ることを意味する。

【0098】

ただし、ステップサイズベクトル $\eta = [\eta_1, \dots, \eta_{n_{sv}}]^T$ であり、 $i$ は、更新式(2.23)においてステップサイズ $\eta = \eta_i$ で、係数ベクトル $n_{sv}$ を更新すると $i = 0$ となる値である。

【0099】

10

20

30

40

50

つぎに、ステップサイズベクトル の要素の中で、0以上で、かつ、最も小さい要素  $i$  をステップサイズとして更新を行う。このとき、 $i = 0$  であり、データセット  $(x_i, y_i)$  を余分なデータセットであるとし、以下の式 (2.25) のようにサポートベクトルから外すことにする。

【0100】

【数21】

$$\alpha \leftarrow [\alpha_1, \dots, \alpha_{i-1}, \alpha_{i+1}, \dots, \alpha_{n_{sv}}] \quad (2.25)$$

10

【0101】

これに伴いサポートベクトルの個数が減少することになる ( $n_{sv} \rightarrow n_{sv} - 1$ )。

【0102】

サポートベクトルの個数が減少した後、再び更新式 (2.23) より係数ベクトル  $n_{sv}$  を更新する。このときも先ほどと同じように、更新前後で係数ベクトル  $n_{sv}$  の各要素の符号が変化しない範囲で更新を行うために式 (2.24) からステップサイズ を求める。

【0103】

ステップサイズベクトル の要素の中で0以上で最も小さい要素  $i$  が1以上の場合、更新式 (2.23) において  $= 1$  で更新した係数ベクトル  $n_{sv}$  は更新前後で各要素の符号が変化しないことがわかる。

20

【0104】

このとき、係数ベクトル  $n_{sv}$  が決定する。逆にステップサイズベクトル の要素の中で0以上で最も小さい要素  $i$  が1よりも小さい場合、先ほどと同じく更新式 (2.23) においてステップサイズ  $= i$  で係数ベクトル  $n_{sv}$  を更新し、データセット  $(x_i, y_i)$  をサポートベクトルから外すことにする。以降、係数ベクトル  $n_{sv}$  が決定するまでこの作業を繰り返し行う。

### 2 - 1 - 3 逐次処理

以上の説明では、 $n$  個のトレーニングデータからいくつかのデータセットを取り出し、その取り出したデータセットだけからSVRを求める方法について説明した。しかし、バッチ処理では逆行列の計算を行っているため、扱うデータ数には限界がある。そこで、この

30

ような問題を回避するために、次に逐次処理について説明する。

【0105】

ここでは、既に以下の式 (2.26) のようなSVRによる近似関数が存在しており、新たなトレーニングデータ  $(x_s, y_s)$  を用いて、このSVRによる近似関数を更新する方法について説明する。

【0106】

【数22】

$$SVR : f(x, \alpha) = \sum_{i=1}^{n_{sv}} \alpha_i K(x, x_i) \quad (2.26)$$

40

【0107】

まずはじめに、新たなトレーニングデータ  $(x_s, y_s)$  が不感帯の内側にあるかどうかを調べる。

【0108】

もし、以下の式 (2.27) が成り立つならば、不感帯の内側にあることになりSVRの更新は必要でないが、新たなトレーニングデータ  $(x_s, y_s)$  が不感帯の外側にある場合、つまり以下の式 (2.28) が成り立つ場合、SVRの更新が必要となる。

【0109】

【数23】

$$\left| y_s - \sum_{i=1}^{n_{sv}} \alpha_i K(x_s, x_i) \right| \leq \varepsilon \quad (2.27)$$

$$\left| y_s - \sum_{i=1}^{n_{sv}} \alpha_i K(x_s, x_i) \right| > \varepsilon \quad (2.28)$$

【0110】

そこで、新たなトレーニングデータ  $(x_s, y_s)$  が不感帯の外側にある場合について考えることにする。

【0111】

このとき、 $(x_1, y_1), \dots, (x_{n_{sv}}, y_{n_{sv}}), (x_s, y_s)$  をサポートベクトルであるとみなす。

【0112】

先ほど示した式(2.23)を用いて、係数ベクトル  $\alpha$  を更新しようとした場合、新たなトレーニングデータ  $(x_s, y_s)$  の係数  $\alpha_s$  が決まっていなないのでこの値の初期値を0とする。

【0113】

しかし、このままでは、式(2.23)を計算する際、 $\text{sign}(\alpha_s) = 0$  となり、このまま係数ベクトル  $\alpha$  を更新して求めても、KKT条件を満たさない。

【0114】

そこで、更新式(2.23)を次のように変更する。

【0115】

【数24】

$$\alpha_{n_{sv}} \leftarrow \alpha_{n_{sv}} + \eta \left( \mathbf{K}_{n_{sv}} + \frac{\mathbf{I}_{n_{sv}}}{C} \right)^{-1} \left( y_{n_{sv}} - \left( \mathbf{K}_{n_{sv}} + \frac{\mathbf{I}_{n_{sv}}}{C} \right) \alpha_{n_{sv}} - \varepsilon \cdot \text{Sign} \right) \quad (2.29)$$

ただし、

$$\alpha \leftarrow [\alpha_1 \dots \alpha_{n_{sv}} \ 0]^T \quad (2.30)$$

$$\text{Sign} \leftarrow \left[ \text{sign}([\alpha_1 \dots \alpha_{n_{sv}}]) \ \text{sign} \left( y_s - \sum_{i=1}^{n_{sv}} \alpha_i K(x_s, x_i) \right) \right]^T \quad (2.31)$$

【0116】

この後は、更新式(2.29)を用いて  $\alpha$  を更新することを考える。ここでも、更新前後で、係数ベクトル  $\alpha_{n_{sv}}$  の各要素の符号が変化しない範囲で更新を行うために式(2.24)を次のように変更し、ステップサイズ  $\eta$  を求める。

【0117】

【数25】

10

20

30

40



$$\eta = -\alpha_{n_{sv}} / \left( \mathbf{K}_{n_{sv}} + \frac{\mathbf{I}_{n_{sv}}}{C} \right)^{-1} \left( \mathbf{y}_{n_{sv}} - \left( \mathbf{K}_{n_{sv}} + \frac{\mathbf{I}_{n_{sv}}}{C} \right) \alpha_{n_{sv}} - \varepsilon \cdot \text{Sign} \right) \quad (2.32)$$

## 【0118】

これ以降は、バッチ処理のところで行ったように の更新を行い、 を決定する。

## 【0119】

実際には、本発明においては、以下の実施の形態の前置型非線形歪補償システムの処理において説明するとおり、大量のトレーニングデータセットが与えられている場合についてのSVRを求めるアルゴリズムを用いる。

## 【0120】

このアルゴリズムは、バッチ処理・逐次処理を一つにまとめた計算方法である。

## 【0121】

[実施の形態1]

図4は、本発明の実施の形態1の前置型非線形歪補償システム1000の構成を説明するための概略ブロック図である。

## 【0122】

図4に示した例では、増幅器16の出力の歪を前置型非線形歪補償器により補償する構成となっている。

## 【0123】

図4を参照して、本発明の実施の形態1に係る前置型非線形歪補償システム1000は、無線周波数信号 $S_{IN}$ を中間周波数信号に変換するダウンコンバータ2と、ダウンコンバータ2に接続され、ダウンコンバータ2の出力する中間周波数信号をサンプリングしてデジタル信号に変換するA/D変換器4と、A/D変換器4に接続され、サンプリングされた中間周波数信号をベースバンド信号 $x(t)$ に変換する直交検波器6と、直交検波器6に接続され、信号 $x(t)$ を、後述する増幅器16の逆特性を用いて、ベースバンド信号 $y(t)$ の実数部 $\text{Re}\{y(t)\}$ に変換するSVM54と、直交検波器6に接続され、信号 $x(t)$ を、増幅器16の逆特性を用いて、ベースバンド信号 $y(t)$ の虚数部 $\text{Im}\{y(t)\}$ に変換するSVM58と、SVM54および58に接続され、変換後のベースバンド信号 $y(t)$ を中間周波数信号に変換する直交変調器10と、直交変調器10に接続され、直交変調器10の出力する中間周波数信号をアナログ信号に変換するD/A変換器12とを含む。

## 【0124】

なお、本願発明は、図4において、ダウンコンバータ2、A/D変換器4および直交検波器6からなる受信部1の代わりに、無線送信機のベースバンド信号の出力部が接続されるような態様にも適用可能である。

## 【0125】

前置型非線形歪補償システム1000は、さらに、D/A変換器12に接続され、アナログの中間周波数信号を無線周波数信号 $S_0$ に変換し、増幅器16に供給するアップコンバータ14と、増幅器16に接続され、増幅器16より出力される出力信号 $S_{OUT}$ を中間周波数信号に変換するダウンコンバータ18と、ダウンコンバータ18に接続され、中間周波数信号をデジタル信号に変換するA/D変換器20と、A/D変換器20に接続され、デジタル信号をベースバンド信号 $z(t)$ に変換する直交検波器22とを含む。

## 【0126】

前置型非線形歪補償システム1000は、さらに、直交検波器22の出力する信号 $z(t)$ と学習に用いたベースバンド信号ベクトル $V_z(p_i)$  ( $i = 1, \dots, n_r$ ) とに基づいて、SVM54より出力されるベースバンド信号の実数部の推定値 $\text{Re}\{y(t)$

10

20

30

40

50

} を出力する SVM26 と、直交検波器22の出力する信号  $z(t)$  とベースバンド信号ベクトル  $V_z(qj)$  ( $j = 1, \dots, n_1$ ) とに基づいて、SVM58より出力されるベースバンド信号の虚数部の推定値  $Im\{y(t)\}$  を出力する SVM62 と、SVM54 および SVM26 に接続され、ベースバンド信号の実数部の推定値  $Re\{y(t)\}$  とベースバンド信号  $y(t)$  の実数部  $Re\{y(t)\}$  との差分を計算する減算器30 と、SVM58 および SVM62 に接続され、ベースバンド信号の虚数部の推定値  $Im\{y(t)\}$  とベースバンド信号  $y(t)$  の虚数部  $Im\{y(t)\}$  との差分を計算する減算器64 と、直交検波部6 および直交検波部22に接続され、信号ベクトルおよびサポートベクトルのサイズ  $m$  と増幅器16による遅延時間  $d$  とを定める  $m \cdot d$  決定部44 と、減算器30、SVM54、SVM26、直交検波部22 および  $m \cdot d$  決定部44 に接続され、減算器30の出力、直交検波部22の出力および  $m \cdot d$  決定部44の出力に基づいて、SVM54 および SVM26 における、増幅器16の逆特性を有する関数の係数  $r(i)$  を更新する係数更新部42 と、減算器64、SVM58、SVM62、直交検波部22の出力および  $m \cdot d$  決定部44 に接続され、減算器64の出力、直交検波部22の出力および  $m \cdot d$  決定部44の出力に基づいて、SVM58 および SVM62 における、増幅器16の逆特性を有する関数の係数  $l(j)$  を更新する係数更新部66 とを含む。

10

#### 【0127】

SVM26 は、後に説明する方法に従い、ベースバンド信号の実数部の推定値  $Re\{y(t)\}$  を求める。SVM62 は、SVM26 と同様の方法に従い、ベースバンド信号の虚数部の推定値  $Im\{y(t)\}$  を求める。

20

#### 【0128】

係数更新部42 は、 $Re\{y(t)\}$  と  $Re\{y(t)\}$  との誤差  $e_r(t)$  が最小になるように係数  $r(i)$  を制御する。係数更新部66 は、係数更新部42 と同様に、 $Im\{y(t)\}$  と  $Im\{y(t)\}$  との誤差  $e_l(t)$  が最小になるように  $l(j)$  を制御する。

#### 【0129】

なお、直交検波器6をアナログで実現し、ダウンコンバータ2の出力を直交検波器6で受け、直交検波器6の出力をA/D変換器4でデジタルに変換し、その出力をSVM8に供給する構成であっても構わない。また、直交変調器10をアナログで実現し、SVM54 および SVM58 の出力をD/A変換器12が受け、D/A変換器12の出力を直交変調器10で受け、直交変調器10の出力をアップコンバータ12に供給する構成であっても構わない。さらに、直交検波器22をアナログで実現し、ダウンコンバータ18の出力を直交検波器22で受け、直交検波器22の出力をA/D変換器20で受け、A/D変換器20の出力をSVM26 および SVM62 に供給する構成であっても構わない。

30

#### 【0130】

(前置型非線形歪補償システム1000の動作)

以下、前置型非線形歪補償システム1000の動作について説明する。

#### 【0131】

まず、図4において、信号  $S_{IN}$ 、 $S_0$  および  $S_{OUT}$  は、式(1)~(3)でそれぞれ表わされる。

40

#### 【0132】

$$S_{IN} = Re\{x(t) \exp(j2\pi(f_{RF}t))\}$$

: 前置型非線形歪補償器への入力信号 ... (1)

$$S_0 = Re\{y(t) \exp(j2\pi(f_{RF}t))\}$$

: 歪補償された信号 ... (2)

$$S_{OUT} = Re\{z(t) \exp(j2\pi(f_{RF}t))\}$$

: 増幅器16からの出力信号 ... (3)

$x(t)$ 、 $y(t)$  および  $z(t)$  は、それぞれ帯域信号  $S_{IN}$ 、 $S_0$  および  $S_{OUT}$  の複素包絡線(ベースバンド信号)である。 $j$  は虚数を表わす。 $f_{RF}$  は無線周波数(Radio Frequency)を表わす。 $t$  は時刻を表わす。

50

## 【 0 1 3 3 】

[ 非線形歪補償器の動作原理 ]

$P = |y(t)|^2 / 2$  として、増幅器 16 のベースバンド領域における入出力特性を  $f$  を複素関数として  $f(P)$  と置くと、ベースバンド信号  $y(t)$  および  $z(t)$  の関係は、下記の式 (4) のように表すことができる。

## 【 0 1 3 4 】

さらに、一般に、非線形歪補償器では、 $f(P)$  に対し、次式 (5) の関係を満たす入出力特性  $g(P)$  を求めることを目的とする。仮に、入出力特性  $g(P)$  を得ることができれば、次式 (6) および (7) が成立し、 $S_{IN}$  に対して  $S_{OUT}$  を線形化することができる。

10

## 【 0 1 3 5 】

## 【 数 2 6 】

$$z(t) = f(P)y(t) \quad (4)$$

：ただし、 $P = \frac{1}{2}|y(t)|^2$ 、 $f(P)$  は複素関数。

$f(P) \cdot g(P) = k$   $k$  は増幅器 16 の利得 (実数) を表わす。

(5)

$$y(t) = g(P)x(t) \quad (6)$$

$$z(t) = f(P)g(P)x(t) = kx(t) \quad (7)$$

20

## 【 0 1 3 6 】

本実施の形態では、 $g(P)$  を、SVM 26 および係数更新部 42 ならびに SVM 62 および係数更新部 66 にて推定する。具体的には、 $y(t)$  と  $z(t)$  より推定した  $y(t)$  との誤差  $e(t)$  が、実部および虚部のそれぞれにおいて最小になるように、SVM 26 および SVM 62 の関数の係数を更新する。この計算が収束すれば、SVM 26 および SVM 62 により近似される特性は、増幅器 16 の逆特性に相当することとなる。

30

## 【 0 1 3 7 】

[ 逆特性関数の計算 ]

上述したような逆特性関数の計算のための条件を以下に整理しておく。

- ・非線形の歪補償のための計算は時間連続信号ではなく離散信号に対して行なわれる。ハードウェア化の際は、A/D変換器を使用し、連続信号を離散信号に変換し、計算後、離散信号を連続信号にD/A変換器により復元する。

- ・信号処理は帯域系ではなくベースバンド系または中間周波数信号にて行なわれる。ベースバンド系の場合には、増幅器 16 は等価的に複素ベースバンド信号  $y(t)$  を受け、複素信号  $z(t)$  を出力する系と見なすことができる。

## 【 0 1 3 8 】

[ サポートベクトルマシンの動作説明 ]

時刻  $kT$  ( $T$ : A/D変換およびD/A変換のサンプリング間隔) のとき、サポートベクトルマシン SVM 54 の出力  $Re\{y(kT)\}$  および SVM 58 の出力  $Im\{y(kT)\}$ 、ならびにサポートベクトルマシン SVM 26 の出力  $Re\{y_s(kT)\}$  および SVM 62 の出力  $Im\{y_s(kT)\}$  は、次式 (8) ~ (11) により得られる。

## 【 0 1 3 9 】

## 【 数 2 7 】

40

SVM54

$$\operatorname{Re}\{y(kT)\} = \sum_{i=1}^{n_R} \alpha_R(i) K\{V\_z(p_i T), V\_x(kT)\} \quad (8)$$

SVM58

$$\operatorname{Im}\{y(kT)\} = \sum_{j=1}^{n_I} \alpha_I(j) K\{V\_z(q_j T), V\_x(kT)\} \quad (9)$$

SVM26

$$\operatorname{Re}\{y_s'(kT)\} = \sum_{i=1}^{n_R} \alpha_R(i) K\{V\_z(p_i T), V\_z(kT)\} \quad (10)$$

10

SVM62

$$\operatorname{Im}\{y_s'(kT)\} = \sum_{j=1}^{n_I} \alpha_I(j) K\{V\_z(q_j T), V\_z(kT)\} \quad (11)$$

ただし、

$$V\_z(p_i T) = [z(p_i T), z(\{p_i - 1\} T), \dots, z(\{p_i - m + 1\} T)]^T \quad (12)$$

(i = 1, \dots, n) 20

$$V\_z(kT) = [z(kT), z(\{k - 1\} T), \dots, z(\{k - m + 1\} T)]^T \quad (13)$$

$$y_s(kT) = y(\{k - d\} T) \quad d: \text{増幅器 16 の入出力間の遅延} \quad (14)$$

【 0 1 4 0 】

ここで、変数の接頭辞 " V \_ " は、その変数がベクトルであることを表わす。また、m は信号ベクトルおよび後述するサポートベクトルのサイズ、n<sub>R</sub>、n<sub>I</sub> は、それぞれ実部および虚部のサポートベクトルマシンのサポートベクトルの数であり、p<sub>i</sub> は i 番目のサポートベクトルを取得した時間であり、q<sub>j</sub> は j 番目のサポートベクトルを取得した時間である。

30

【 0 1 4 1 】

V<sub>z</sub>(p<sub>i</sub>T) (i = 1, ..., n<sub>R</sub>), V<sub>z</sub>(q<sub>j</sub>T) (j = 1, ..., n<sub>I</sub>) は増幅器 16 の入出力特性を学習した結果、時刻 k T までに抽出されたサポートベクトルである。

【 0 1 4 2 】

$\alpha_R(i)$ ,  $\alpha_I(j)$  は、それぞれ V<sub>z</sub>(p<sub>i</sub>T), V<sub>z</sub>(q<sub>j</sub>T) に対応した重み係数 (実数) である。一般に、 $\alpha_R(i)$  (または、 $\alpha_I(j)$ ) が零の時、対応する V<sub>z</sub>(p<sub>i</sub>T) (または、V<sub>z</sub>(q<sub>j</sub>T)) は計算に加える必要がなく廃却しても構わない。また、非零の係数  $\alpha_R(i)$  (または、 $\alpha_I(j)$ ) の個数をサポートベクトルマシンの次元と呼ぶ。例えば、非零の係数の個数が 73 であればサポートベクトルマシンの次元数は 73 である。さらに、この次元数は学習データを増やしても一定値以上に増えることがない。この上界値のことを VC (Vapnik and Chervonenkis) 次元と呼ぶ。

40

【 0 1 4 3 】

K(V<sub>x</sub>, V<sub>y</sub>) は「内積カーネル」の関数であり、本実施の形態では、内積カーネルの線形和である式 (8) ~ (11) を、適切な係数  $\alpha_R(i)$ ,  $\alpha_I(j)$  を選択することにより、増幅器 16 の入出力特性の逆特性 g(P) に適合させることを目標とする。このように未知の関数にシステムの入出力関数を適合させるような問題全般を総称して「回帰問題」という。次式 (15) ~ (17) は、一般的に用いられる内積カーネルの例である。

【 0 1 4 4 】

【 数 2 8 】

50

$$K(V\_x, V\_y) = \exp(-\|V\_x - V\_y\|^2 / 2\sigma^2) \quad : \text{Gaussian} \quad (15)$$

$$K(V\_x, V\_y) = (aV\_x^T V\_y + 1)^P \quad : \text{Polynomial} \quad (16)$$

$$K(V\_x, V\_y) = \tanh(aV\_x^T V\_y - b) \quad : \text{Sigmoid} \quad (17)$$

## 【 0 1 4 5 】

どの内積カーネルが逆関数を表現するのに最も適しているかはシステムにより異なり、実際には動作前に何度か試験をした上で最も特性のよいものを選択する。もちろん、上記以外の関数でも逆関数  $g(P)$  をよりよく推定でき、かつ期待する効果が得られるのであれば、その関数を使用してもよい。

10

## 【 0 1 4 6 】

[ 係数更新部 4 2 の動作説明 ]

以下では、サポートベクトルに対応する係数  $r(i)$  ,  $l(j)$  の更新計算を行う係数更新部 4 2 および係数更新部 6 6 の動作について説明する。

## 【 0 1 4 7 】

上述したとおり、係数更新部 4 2 は、 $\text{Re}\{y(t)\}$  と  $\text{Re}\{y'(t)\}$  との誤差  $e_R(t)$  が最小になるように係数  $r(i)$  を制御する。一方、係数更新部 6 6 は、 $\text{Im}\{y(t)\}$  と  $\text{Im}\{y'(t)\}$  との誤差  $e_l(t)$  が最小になるように係数  $l(j)$  を制御する。

20

## 【 0 1 4 8 】

## 【 数 2 9 】

SVM54, SVM26

$$e_R(p_n) = \text{Re}\{y(p_n)\} - \text{Re}\{ys'(p_n)\} \quad (18)$$

(ここで、 $ys(p_n)$ ,  $ys'(p_n)$  は、 $ys(p_n T)$ ,  $ys'(p_n T)$  で、

$T=1$  とした例の表現。以下同じ)

30

SVM58, SVM62

$$e_l(q_n) = \text{Re}\{y(q_n)\} - \text{Re}\{ys'(q_n)\} \quad (19)$$

## 【 0 1 4 9 】

なお、以下では、表記を簡単にするため  $T=1$  としている。このようにしても一般性を失うことはない。

## 【 0 1 5 0 】

係数  $r(i)$  は、次式 (20) で表わされる評価関数  $W$  を最大化するという最適化問題を解くことにより得られる。また、係数  $l(j)$  についても同様である。

40

## 【 0 1 5 1 】

## 【 数 3 0 】

評価関数  $W$

$$W = \sum_{i=1}^{n_R} \alpha_R(i) \operatorname{Re}\{y_s(p_i)\} - \varepsilon \sum_{i=1}^{n_R} |\alpha_R(i)| - \frac{1}{2} \sum_{i,j=1}^{n_R} \alpha_R(i) \alpha_R(j) \{K(\operatorname{Re}(V_{-z}(p_i T)), \operatorname{Re}(V_{-z}(p_j T))) + \frac{\delta_{ij}}{C}\} \quad (20)$$

$$\text{ただし、} \delta_{ij} = \begin{cases} 1 & (i=j) \\ 0 & (i \neq j) \end{cases}$$

10

$\varepsilon$ : 雑音に対するマージン (定数)

$C$ : モデルの複雑さと制約条件の緩和 (トレーニング誤差)

との間のトレードオフを設定する正則化パラメータ

$$y_s(p_i) = y(p_i - d)$$

20

#### 【 0 1 5 2 】

ただし、定数  $C$  と定数  $\varepsilon$  は、通常、実験的に求められる。

増幅器 16 の入出力特性の学習が完了するまでの間はサポートベクトルの追加・削除が繰返されるが、これと対応して係数  $\alpha_R(i)$ 、 $\alpha_R(j)$  の値を更新する必要がある。

#### 【 0 1 5 3 】

以下では、係数更新部 42 の動作を例にとって説明する。係数更新部 66 の動作も、入力される変数が異なるのみで、基本的に係数更新部 42 と同様である。

#### 【 0 1 5 4 】

なお、後に詳しく説明するように、係数更新部 42 の更新の計算の特徴の 1 つは、以下のようにして逐次的に係数  $\alpha_R(i)$  の値を更新することである。

30

(1) 更新の計算の途中において、係数更新部 42 は、 $n-1$  組のサポートベクトル  $V_{-z}(p_i)$  および係数  $\alpha_R(i)$  を取得している状態となる。

(2) 係数更新部 42 は、さらに、 $n$  個目のサポートベクトルの候補としてデータ  $V_{-z}(p_n)$  を入手する。

(3)  $V_{-z}(p_n)$  をサポートベクトルと見なし、その係数を  $\alpha_R(n) = 0$  とおく。

(4)  $V_{-z}(p_n)$  が加わったことによる最適解からのずれ (摂動) を補正するため、 $\alpha_R(i)$  ( $i = 1, \dots, n$ ) に関して誤差分 (摂動分)  $\Delta \alpha_R(i)$  を計算する。

#### 【 0 1 5 5 】

上述のように逐次サポートベクトルを抽出することにより、1 回の係数更新において計算対象となる係数の数を削減できるので、計算量を削減することができる。

40

#### 【 0 1 5 6 】

[ 係数の更新の具体的な計算方法 ]

係数の更新方法の具体例を以下に示す。ただし、この具体例は実現の一例であって、必ずしもこの通りである必要はない。

#### 【 0 1 5 7 】

図 5 および図 6 は、このような係数の更新の具体的な計算方法を説明するためのフローチャートである。

#### 【 0 1 5 8 】

図 5 を参照して、係数更新部 42 は、まず、取得済みの  $N$  個の学習データセットから  $n$  個のデータセット ( $V_{-z}(P_i)$ ,  $\operatorname{Re}\{y_s(P_i)\}$ ) ( $i = 1 \sim n$ ) を取り出し、

50

これらをサポートベクトルに採用する（ステップ100）。

【0159】

次に、係数更新部42は、上記サポートベクトルに対応する係数の第1次近似と、その符号ベクトルSRを、以下の式(21)および(22)により計算する（ステップ102）。

【0160】

【数31】

$$\begin{aligned} \hat{\alpha}_R &= [\alpha_R(1), \alpha_R(2), \dots, \alpha_R(n)]^T \\ &= \text{inv} \left( \hat{K}_R + \frac{\hat{I}(n)}{C} \right) \cdot \hat{y}_R \end{aligned} \quad (21)$$

$$\hat{S}_R = [\text{sign}(\alpha_R(1)), \dots, \text{sign}(\alpha_R(n))]^T \quad (22)$$

ただし、

$[\dots]^T$ : 行ベクトルを転置して列ベクトルに変換する操作 20

$\text{inv}[\dots]$ : 行列 $[\dots]$ に対する逆行列

$\cdot$ : 行列の積

$$\hat{K}_R = \begin{pmatrix} K_R(1,1) & K_R(1,2) & \dots & K_R(1,n) \\ K_R(2,1) & K_R(2,2) & \dots & K_R(2,n) \\ \vdots & & & \vdots \\ K_R(n,1) & K_R(n,2) & \dots & K_R(n,n) \end{pmatrix} \quad (30)$$

$K_R(i,j) = K\{V_{-z}(p_i), V_{-z}(p_j)\}$

$\hat{I}(n)$ :  $n$ 行 $\times$  $n$ 列の単位行列

$$\hat{y}_R = [\text{Re}\{y_s(p_1), y_s(p_2), \dots, y_s(p_n)\}]^T \quad (23)$$

【0161】

さらに、ステップサイズベクトル $[r(1), \dots, r(n)]^T$ を以下の式(24)および(25)により計算する（ステップS104）。

【0162】

【数32】

$$\hat{\eta}_R = [\eta_R(1), \eta_R(2), \dots, \eta_R(n)]^T \quad (24)$$

ただし、 $\eta_R(i) = -\alpha_R(i) / \Delta \alpha_R(i)$

$$\begin{aligned} & [\Delta \alpha_R(1), \Delta \alpha_R(2), \dots, \Delta \alpha_R(n)]^T \\ &= \text{inv} \left( \hat{K}_R + \frac{\hat{I}(n)}{C} \right) \cdot (\hat{y}_R - \varepsilon \hat{S}_R) - \hat{\alpha}_R \end{aligned} \quad (25)$$

10

【0163】

ステップサイズベクトルの要素の中で0以上でもっとも小さい要素を  $R(i)$  とし(ステップS106)、要素  $R(i)$  が1以上であるか否かを判定する(ステップS108)。

【0164】

まず、要素  $R(i)$  が1未満である場合、続いて、要素  $R(i)$  を用いて、以下の式(26)により、係数ベクトル  $[\alpha_R(1), \dots, \alpha_R(n)]^T$  を更新する(ステップS110)。

20

【0165】

【数33】

$$\hat{\alpha}_R \leftarrow \hat{\alpha}_R + \eta_R(i) \left\{ \text{inv} \left( \hat{K}_R + \frac{\hat{I}(n)}{C} \right) \cdot (\hat{y}_R - \varepsilon \hat{S}_R) - \hat{\alpha}_R \right\} \quad (26)$$

【0166】

その上で、 $(V_z(P), \text{Re}\{y_s(P)\})$  を余分なデータセットとして、サポートベクトルから削除し、係数  $\alpha_R(P)$  も削除する(ステップS112)。このような削除に対応して、 $i > P$  について、以下の表にしたがって、 $i$  番目の学習データを取得した時間、および係数  $\alpha_R(i)$  の番号を読替える。

30

【0167】

【表1】

読替え前	読替え後
$P_i$	$P_{i-1}$
$\alpha_R(i)$	$\alpha_R(i-1)$

40

【0168】

さらに、 $(n-1)$  を新たに  $n$  と読替える処理を行う(ステップS114)。以上の読替えの手続きの後、処理はステップS104に復帰する。

【0169】

一方、ステップS108において、要素  $R(i)$  が1以上である場合、以下の式(27)により、改めて係数ベクトルを計算する(ステップS116)。

【0170】

【数34】



$$\hat{\alpha}_R = \text{inv} \left( \hat{K}_R + \frac{\hat{I}(n)}{C} \right) \cdot (\hat{y}_R - \epsilon \hat{S}_R) \quad (27)$$

## 【 0 1 7 1 】

次に、図 6 を参照して、新しい学習データセット (  $V_{-z}(k)$  ,  $\text{Re}\{y_s(k)\}$  ) の取得が行われる (ステップ S 1 1 8 )。

## 【 0 1 7 2 】

さらに、以下の式 ( 2 8 ) により、誤差  $e_R$  が計算される (ステップ 1 2 0 )。続いて、誤差  $e_R$  としきい値  $\epsilon'$  との比較が行われて (ステップ S 1 2 2 )、 $|e_R| < \epsilon'$  であれば、処理は、ステップ S 1 1 8 に復帰する。なお、ここで、 $\epsilon'$  である。

## 【 0 1 7 3 】

## 【 数 3 5 】

$$e_R = \text{Re}\{y_s(k)\} - \sum_{i=1}^n \alpha_R(i) K\{V_{-z}(k), V_{-x}(p_i)\} \quad (28)$$

$$|e_R| < \epsilon'$$

10

20

## 【 0 1 7 4 】

一方、 $|e_R| < \epsilon'$  であれば、以下の式 ( 2 9 ) ~ ( 3 0 ) にしたがって、係数  $\alpha_R(1)$  , ... ,  $\alpha_R(n)$  , および誤差  $e_R$  のそれぞれについて、符号を計算し、符号ベクトルを置き換え (ステップ S 1 2 4 )、 $[\alpha_R(1), \dots, \alpha_R(n), 0]$  を新たに係数ベクトルに置き換えるとともに、 $(n+1)$  を新たに  $n$  と置き換える (ステップ S 1 2 6 )。

## 【 0 1 7 5 】

## 【 数 3 6 】

$$\hat{S}_R \leftarrow [\text{sign}(\alpha_R(1)), \dots, \text{sign}(\alpha_R(n)), \dots, \text{sign}(\alpha_R(n))]'^T \quad (29)$$

$$\alpha_R \leftarrow [\alpha_R(1), \dots, \alpha_R(n), 0]^T \quad (30)$$

$$n \leftarrow n+1$$

30

## 【 0 1 7 6 】

(  $V_{-z}(k)$  ,  $\text{Re}\{y_s(k)\}$  ) を  $n$  番目のサポートベクトル (  $V_{-z}(Pn)$  ,  $\text{Re}\{y_s(Pn)\}$  ) として採用して、処理はステップ S 1 0 4 に復帰する (ステップ S 1 2 8 )。

## 【 0 1 7 7 】

以上の処理により、係数更新部 4 2 は、逐次適応的に係数ベクトルの更新を行う。

## 【 0 1 7 8 】

[ サポートベクトルマシン SVM 5 4 , 5 8 の動作説明 ]

SVM 5 4 は、係数更新部 4 2 で更新された係数  $\alpha_R(i)$  を用いて、上述した式 ( 8 ) に従い、ベースバンド信号  $x(t)$  をベースバンド信号  $\text{Re}\{y(t)\}$  に変換する。サポートベクトルマシン SVM 5 8 は、上述したと同様の手続きで、係数更新部 6 6 で更新された係数  $\alpha_I(i)$  を用いて、上述した式 ( 9 ) に従い、ベースバンド信号  $x(t)$  をベースバンド信号  $\text{Im}\{y(t)\}$  に変換する。

## 【 0 1 7 9 】

[ 逐次計算における特徴的手続きの説明 ]

以上、図 5 および図 6 により、サポートベクトルマシンを用いて、非線形なプロセスであって、当該プロセスを表現する変数が複素数で表される場合に、回帰問題を解くことによ

40

50

り、前置型非線形歪補償システムを実現するための構成について説明した。特に、上記係数更新部42の行う処理において、特徴的な手続きについて、以下に説明を付け加えておく。

【0180】

図7は、1入力1出力の非線形回帰  $f: x \rightarrow y$  に関して、サポートベクトルマシンによる近似学習がある程度進んだ状態において、サポートベクトルによる学習データの近似特性を示す概念図である。

【0181】

図7においては、「損失」は、不感帯2次損失関数に基づく近似特性を表しているものとする。つまり、「損失」は、不感帯の片側幅を近似特性に加減して得られた2つの不感帯境界値のうち、学習データに近い方の不感帯境界値を減算することにより得られる値と定義する。

10

【0182】

図7で、2重丸はサポートベクトルであるデータを表しており、1重丸は、サポートベクトルによる近似特性に対する不感帯の内側に存在し、サポートベクトルではないデータを表す。なお、サポートベクトルの定義により、図7中で2重丸で表されるデータが真のサポートベクトルであるときは、この真のサポートベクトルの外側には、サポートベクトルは存在しない。

【0183】

ここで、たとえば、サポートベクトルによる近似特性の上側であって、かつ、不感帯の外側にあるサポートベクトルのデータ  $D1 = (x_1, y_1)$  について見てみると、このデータ  $D1$  についての損失  $L1$  は、以下の式で表される。

20

【0184】

$$L1 = \gamma_1 / C$$

なお、ここで、 $\gamma_1$  はサポートベクトル  $x_1$  に対する係数である。このとき、データ  $D1$  の損失は、このデータがサポートベクトルによる近似特性の上側にあることに対応して、「上向き、かつ、外向き」であり、 $\gamma_1 > 0$  である。

【0185】

一方、たとえば、サポートベクトルによる近似特性の下側であって、かつ、不感帯の外側にあるサポートベクトルのデータ  $D2 = (x_2, y_2)$  について見てみると、このデータ  $D2$  についての損失  $L2$  も、 $L2 = \gamma_2 / C$  で表される。

30

【0186】

このとき、データ  $D2$  の損失は、このデータがサポートベクトルによる近似特性の下側にあることに対応して、「下向き、かつ、外向き」であり、 $\gamma_2 < 0$  である。

【0187】

(ステップS122における  $\gamma'$  の値の設定)

$\gamma'$  の値は、本来の不感帯を定義する値  $\gamma$  と同じ値としても計算を実行することは可能である。しかし、本願発明では、 $\gamma'$  の値を本来の不感帯を定義する値  $\gamma$  よりも大きくしている。

【0188】

そこで、以上の準備のもとに、まず、図6のステップS122において、 $\gamma'$  の値を本来の不感帯を定義する値  $\gamma$  よりも大きくする理由について、まず、説明する。

40

【0189】

図8は、このように  $\gamma'$  の値を本来の不感帯を定義する値  $\gamma$  よりも大きくすることの効果の説明するための概念図である。

【0190】

なお、 $\gamma'$  で決まる本来の不感帯に対し、 $\gamma$  をステップS122の  $\gamma'$  で置換えた場合に、 $\gamma'$  で決まる領域を本来の不感帯に対する「拡張不感帯」と定義する。

【0191】

拡張不感帯はステップS122のみにおいて用いられ、他のステップにおける不感帯は、

50

で決定される。

【0192】

図8においては、学習途中のサポートベクトルによる近似特性が直線LSVで表され、この近似特性よりもさらに近似が進んだ状態の近似特性が曲線CSVで表されているものとする。ただし、「学習途中のサポートベクトルによる近似特性」を直線としたのは、説明の便宜のためであって、より一般には、この特性も曲線で表される。

【0193】

まず、近似特性LSVについて、拡張不感帯をしきい値 $\theta$  ( $\theta = \theta_1$ )で定義した場合は、データD3は、最終的にはサポートベクトルにはならないデータであるものの、近似特性LSVの段階では、サポートベクトルとして選択されることになる。これは、拡張不感帯を定義するしきい値を、しきい値 $\theta$ よりも大きなしきい値 $\theta_1$  ( $\theta_1 > \theta$ )で定義した場合も同様である。

10

【0194】

これに対して、近似特性LSVについて、拡張不感帯をしきい値 $\theta_1$  ( $\theta_1 > \theta$ )よりもさらに大きなしきい値 $\theta_2$  ( $\theta_2 > \theta_1$ )で定義した場合は、データD3は、近似特性LSVの段階でも、サポートベクトルとして選択されない。

【0195】

上述したとおり、本発明においては、計算量を低減するために、逐次計算の途中において、サポートベクトルとして選択されなくなったデータを計算の対象から除外している。

【0196】

したがって、図6のステップS122において使用するしきい値 $\theta$ を、予め本来の不感帯に対応するしきい値 $\theta_1$ よりも大きなしきい値 $\theta_2$ としておけば、計算が進むにつれて、除外されていくはずのデータをそもそもサポートベクトルとして選択する確率が小さくなるので、一層の計算量の低減を期待できる。

20

【0197】

すなわち、1度、サポートベクトルとして抽出した古いデータセットが、学習(新しいサポートベクトルの追加)が進むにつれ、サポートベクトルとして不要となり除去されることが繰り返され、最終的には新しいサポートベクトルの追加が不要となって学習が終了することになる。しかしながら、「サポートベクトルの追加・除去の繰り返し」を低減することにより、計算量を削減するとともに学習の終了までの時間を短縮することが可能となる。

30

【0198】

なお、新しいサポートベクトルの追加の頻度が低下するにつれてしきい値 $\theta$  ( $\theta = \theta_1$ )に近づけていくという処理を行うことも可能である。この場合、ある程度の近似精度までは早期に学習することになり、かつ、最終的な精度 $\theta_2$ も確実に得ることが可能となる。

【0199】

また、係数更新部42や係数更新部66において、直交検波器22から受取る学習データの数をカウントするカウンタを備えて、一定数のデータを受取るごとにしきい値 $\theta$ を更新しつつ、 $\theta_1$ に近づけていくという処理を行うことも可能である。

【0200】

あるいは、係数更新部42や係数更新部66において、時間を計測するカウンタまたは抽出されたサポートベクトルの数をカウントするカウンタを有し、処理期間の長さに対応して、しきい値 $\theta$ を更新しつつ、 $\theta_1$ に近づけていくという処理を行うことも可能である。

40

【0201】

(サポートベクトルを削除する基準)

さらに、図5のステップS110からS114において、サポートベクトルを削除する基準について、さらに説明する。

【0202】

図9は、図8で示される状態から、近似特性の逐次計算を進めた場合に、余分となったサポートベクトルの除去を行う手続きを説明するための概念図である。

50

## 【0203】

以下の説明では、図8で示される状態で、しきい値と近似特性LSVにより決まる不感帯の外にデータD3があるため、データD3がサポートベクトルとして選択されており、図9に示す状態に近似を進めると、このデータD3がサポートベクトルから除去され、一方、データD1が新たにサポートベクトルとして追加されるものとする。

## 【0204】

図10は、図9において、データD3について、損失の変化を説明するための概念図である。

## 【0205】

なお、ここでは、直線LSVで表される近似特性で、このデータD3に対応する係数を  $i(1)$  とし、曲線CSVで表される近似特性でのデータD3に対応する係数を  $i(2)$  と表すものとする。 10

## 【0206】

直線LSVの不感帯の境界の外側にデータD3が存在することに対応して、直線LSVに対するデータD3の損失は、  $i(1)/C(i(1) > 0)$  となる。

## 【0207】

これに対して、曲線CSVの不感帯の境界の内側にデータD3が存在することに対応して、曲線CSVに対するデータD3の損失は、  $i(2)/C(i(2) < 0)$  となる。

## 【0208】

したがって、近似が進んで、サポートベクトルによる近似特性が、直線LSV（より一般的には曲線）から、曲線CSVへと変化することによる損失の変化量を、  $-i/C$  とすると、ステップS108で用いたステップサイズは、  $= -i(1)/i$  と表される。 20

## 【0209】

このとき、  $0 < 1$  が成り立っていれば、図10で示すように、直線LSV（より一般的には曲線）の不感帯の境界と曲線CSVの不感帯の境界との間にデータD3が存在することになる。これは、言い換えると、「データD3は、サポートベクトルによる近似特性が、直線LSV（より一般的には曲線）から、曲線CSVと変化することにより、サポートベクトルから除去されるデータとなった」ということができる。なお、以上の説明では、データが近似特性の上側にある場合について説明したが、データが近似特性の下側に 30

## 【0210】

したがって、図5のステップS108におけるような基準で判断することにより、それまでサポートベクトルとして選択していたデータをサポートベクトルから除外するべきか否かを判断することが可能となる。

## 【0211】

上述の例では、入出力特性が不明な、非線形の入出力特性を有する増幅器16の出力の補正に対して、サポートベクトルマシンを用いたが、本発明は増幅器16に対してのみ適用可能なわけではない。一般的に、入力される信号が複素数で表現でき、かつ入出力特性が不明な信号の変換回路に対しても本発明を適用することが可能である。 40

## 【0212】

以上説明したように、本実施の形態によると、SVM26およびSVM62を用いて、増幅器16の逆特性を有する信号を推定する。その信号とSVM54およびSVM58より出力される信号とに基づいて、係数更新部42および係数更新部66において、増幅器16の逆特性を有する関数の係数が更新される。

## 【0213】

この構成では、ルックアップテーブルを必要としないため、ノイズフロアを低く押さえることができ、かつ、入力信号が複素信号である場合に適用可能であるので、簡単な構成で前置型非線形歪補償器を提供することができる。また、サポートベクトルマシンは、 `pth-order-predistorter` に比べて、計算が簡単で収束時間が早い。さらに 50

、学習する過程で、一度、サポートベクトルに加えられた信号ベクトルにおいて、不要となったサポートベクトルが削除されるため、高速処理が可能な前置型非線形歪補償器を提供することができる。

【0214】

また、広帯域信号が入力された場合、増幅器の非線形性が信号帯域に比較して狭帯域の場合にもデータベクトルのサイズ $m$ を適切に設定することにより補償することが可能である。

【0215】

[実施の形態2]

図11は、本発明の実施の形態2の前置型非線形歪補償システム2000の構成を説明するための概略ブロック図である。 10

【0216】

図11に示した例でも、増幅器16の出力の歪を前置型非線形歪補償器により補償する構成となっている。

【0217】

図11に示した前置型非線形歪補償システム2000では、SVM54、SVM58およびSVM26において、サポートベクトルマシンの実数入力端子 $Re$ と虚数入力端子 $Im$ を明示的に示している。

【0218】

図11に示した前置型非線形歪補償システム2000の構成が、図1に示した前置型非線形歪補償システム1000の構成と異なる点は、以下のとおりである。 20

【0219】

まず、直交変調器6の出力のうち実数成分の $Re\{x(t)\}$ を受けて、 $(-1)$ を乗算した値を出力する乗算器7が設けられている。

【0220】

さらに、SVM58の実数入力端子 $Re$ へは、直交変調器6の出力のうち、虚数成分の $Im\{x(t)\}$ が入力され、SVM58の虚数入力端子 $Im$ には、乗算器7の出力が与えられる。

【0221】

これに応じて、SVM58の出力 $Im\{y(kT)\}$ を以下のような内積カーネルにより計算するものとする。 30

【0222】

【数37】

SVM58

$$\text{Im}\{y(kT)\} = \sum \alpha_R(i) K\{V_z(p_i T), V_w(kT)\}$$

$$\text{ただし、 } V_w(kT) = \text{Im}(V_x(kT)) - \sqrt{-1} \cdot \text{Re}(V_x(kT))$$

40

【0223】

さらに、SVM62、減算器64、係数更新部66が省略され、SVM54およびSVM58は、ともに係数更新部42において更新される係数 $\alpha_R(i)$ を用いて演算を行う構成となっている。

【0224】

前置型非線形歪補償システム2000の構成は、以上のような置き換えを行う以外は、前置型非線形歪補償システム1000の構成と同様であるので、同一部分には同一符号を付してその説明は繰り返さない。

【0225】

このような置き換えを行えば、前置型非線形歪補償システム1000で説明したのと同様 50

の手続きで、SVM26を用いて、増幅器16の逆特性を有する信号を推定することが可能である。

【0226】

したがって、以上のような構成により、前置型非線形歪補償システム1000よりも、より簡単な構成で、実施の形態1と同様の効果を奏することができる。

【0227】

今回開示された実施の形態はすべての点で例示であって制限的なものではないと考えられるべきである。本発明の範囲は上記した説明ではなくて特許請求の範囲によって示され、特許請求の範囲と均等の意味および範囲内でのすべての変更が含まれることが意図される。

10

【0228】

【発明の効果】

以上説明したとおり、本発明によれば、サポートベクトルマシンを用いて、増幅器等の信号の変換回路の逆特性を有する信号を推定するので、ルックアップテーブルを必要としないため、ノイズフロアを低く押さえることができる。

【0229】

しかも、入力が複素信号である場合に適用可能であるので、簡単な構成で前置型非線形歪補償器を提供することも可能となる。

【0230】

さらに、学習する過程で、一度、サポートベクトルに加えられた信号ベクトルにおいて、不要となったサポートベクトルが削除されるため、高速処理が可能である。

20

【0231】

また、広帯域信号が入力された場合、増幅器の非線形性が信号帯域に比較して狭帯域の場合にもデータベクトルのサイズを適切に設定することにより補償することが可能である。

【図面の簡単な説明】

【図1】 線形不感損失関数および二次不感損失関数を示す図である。

【図2】 線形近似関数とサポートベクトルとの関係を示す概念図である。

【図3】 非線形近似関数とサポートベクトルを示す概念図である。

【図4】 本発明の実施の形態1の前置型非線形歪補償システム1000の構成を説明するための概略ブロック図である。

30

【図5】 係数の更新の具体的な計算方法を説明するための第1のフローチャートである。

【図6】 係数の更新の具体的な計算方法を説明するための第2のフローチャートである。

【図7】 サポートベクトルマシンによる近似学習がある程度進んだ状態において、サポートベクトルによる学習データの近似特性を示す概念図である。

【図8】  $\gamma$  の値を本来の不感帯を定義する値よりも大きくすることの効果の説明するための概念図である。

【図9】 近似特性の逐次計算を進めた場合に、余分となったサポートベクトルの除去を行う手続きを説明するための概念図である。

40

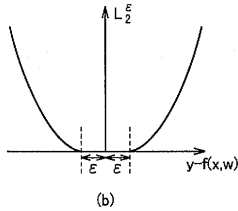
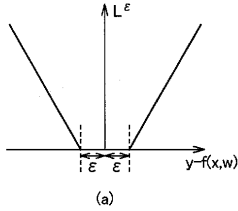
【図10】 図9において、データD3について、損失の変化を説明するための概念図である。

【図11】 本発明の実施の形態2の前置型非線形歪補償システム2000の構成を説明するための概略ブロック図である。

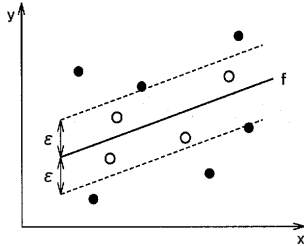
【符号の説明】

2, 18 ダウンコンバータ、4, 20 A/D変換器、6, 22 直交検波器、54, 58, 26, 62 SVM、10 直交変調器、12 D/A変換器、14 アップコンバータ、16 増幅器、30, 64 減算器、42, 66 係数更新部。

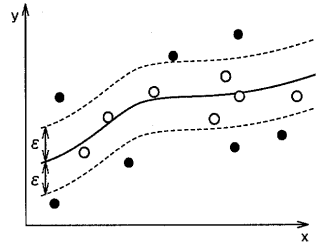
【 図 1 】



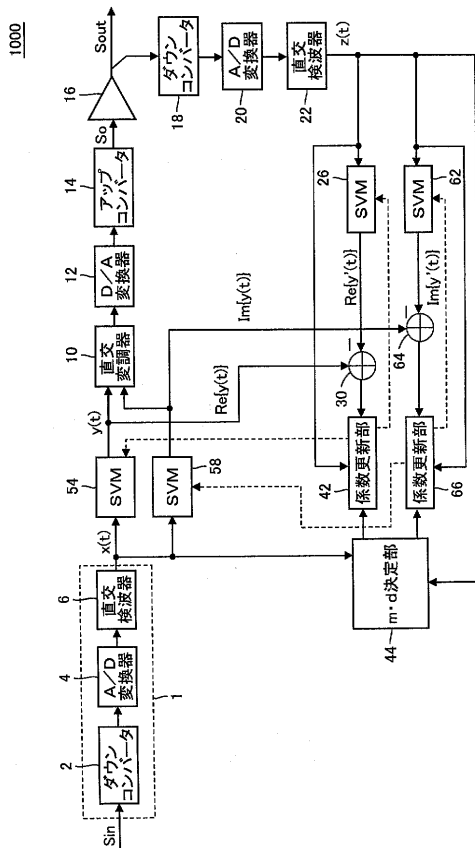
【 図 2 】



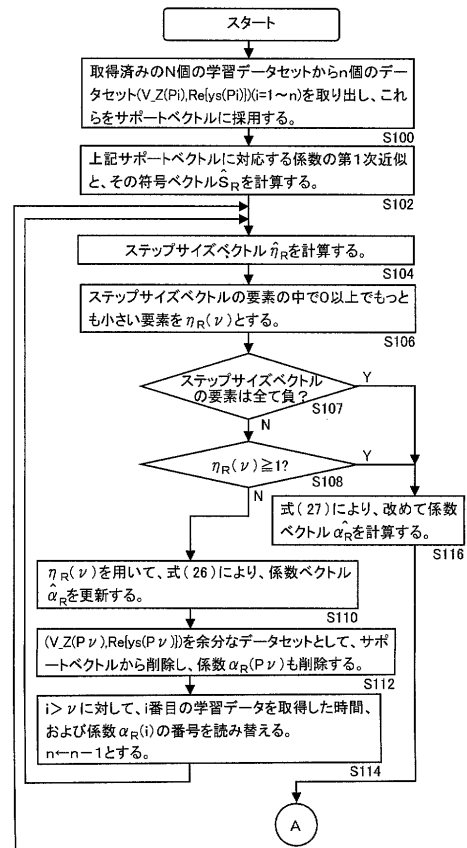
【 図 3 】



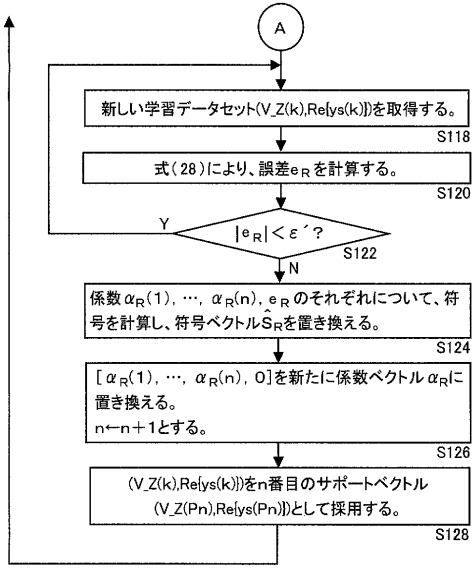
【 図 4 】



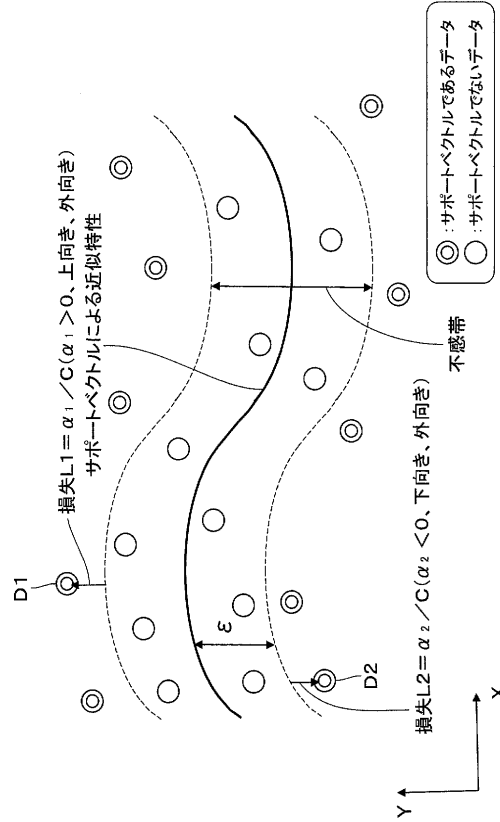
【 図 5 】



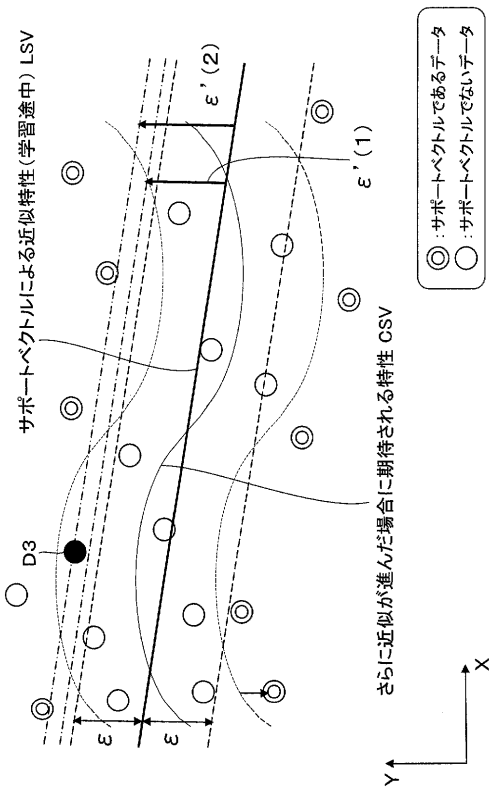
【 図 6 】



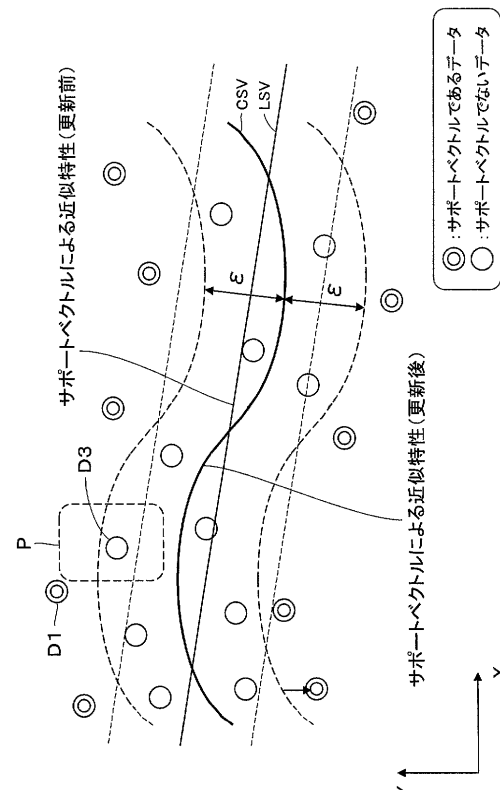
【 図 7 】



【 図 8 】

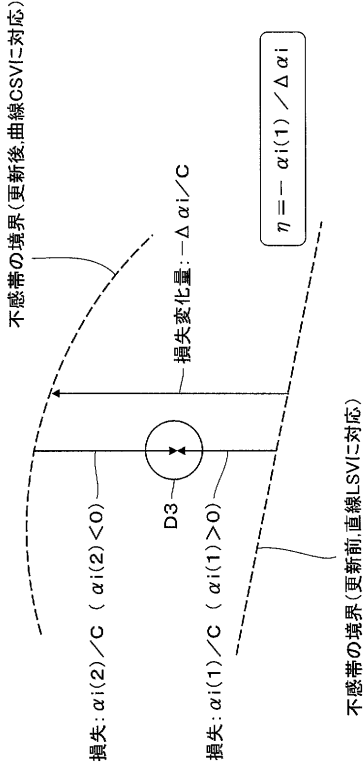


【 図 9 】

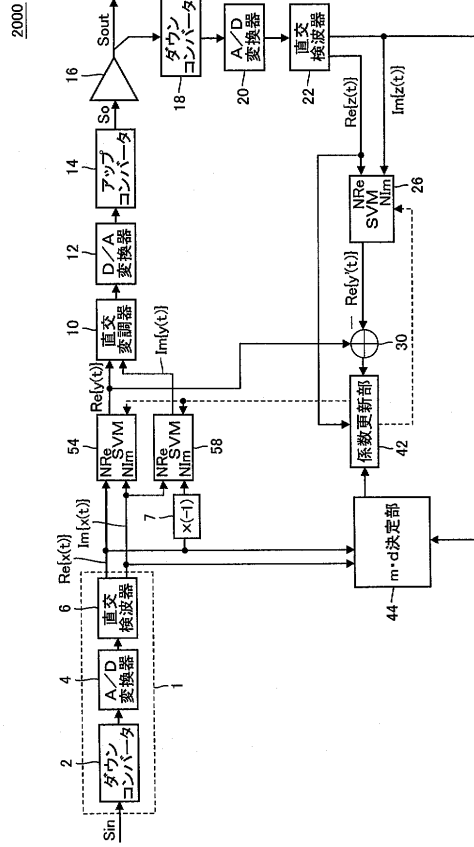




【 図 1 0 】



【 図 1 1 】



---

フロントページの続き

- (72)発明者 多湖 紀之  
大阪市此花区島屋一丁目1番3号 住友電気工業株式会社 大阪製作所内
- (72)発明者 岡田 洋侍  
大阪市此花区島屋一丁目1番3号 住友電気工業株式会社 大阪製作所内
- (72)発明者 佐野 昭  
神奈川県横浜市港北区日吉三丁目14番1号 慶應義塾大学内

審査官 藤井 浩

- (56)参考文献 特開2002-261848(JP,A)  
特開2003-087340(JP,A)

- (58)調査した分野(Int.Cl., DB名)
- H04L 27/00 - 27/38
  - H03F 1/32
  - H04B 1/04
  - H04B 1/26