

(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2014-153949

(P2014-153949A)

(43) 公開日 平成26年8月25日(2014.8.25)

(51) Int.Cl. F I テーマコード (参考)
G05B 13/02 (2006.01) G05B 13/02 L 5H004

審査請求 未請求 請求項の数 5 O L (全 16 頁)

(21) 出願番号 特願2013-23652 (P2013-23652)
 (22) 出願日 平成25年2月8日(2013.2.8)

(71) 出願人 504136568
 国立大学法人広島大学
 広島県東広島市鏡山1丁目3番2号
 (74) 代理人 110001427
 特許業務法人前田特許事務所
 (72) 発明者 山本 透
 広島県東広島市鏡山一丁目4番1号 国立
 大学法人広島大学大学院工学研究院内
 (72) 発明者 脇谷 伸
 広島県東広島市鏡山一丁目4番1号 国立
 大学法人広島大学大学院工学研究院内
 Fターム(参考) 5H004 GA34 GB01 KA01 KB01 KC27
 KD32 KD47 MA36

最終頁に続く

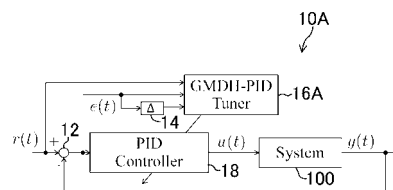
(54) 【発明の名称】 GMDHを用いた非線形PIDコントローラおよびGMDH重み係数決定方法

(57) 【要約】

【課題】より少ない容量のメモリで動作可能な非線形PIDコントローラを提供する。

【解決手段】非線形PIDコントローラ(10A)は、制御対象の非線形システム(100)の出力値と目標値との偏差を算出する偏差演算部(12)と、偏差の微分値を算出する微分演算部(14)と、目標値、偏差、および微分値の3値からGMDH理論に基づいて非線形システムのPID制御に係る比例ゲイン、積分ゲイン、および微分ゲインを算出するGMHD-PID調整部(16A)と、偏差に基づいて、算出された比例ゲイン、積分ゲイン、および微分ゲインで非線形システムをPID制御するPID制御部(18)とを備えている。

【選択図】図5



【特許請求の範囲】

【請求項 1】

制御対象の非線形システムの出力値と目標値との偏差を算出する偏差演算部と、
前記偏差の微分値を算出する微分演算部と、
前記目標値、前記偏差、および前記微分値の 3 値から G M D H (Group Method of Data Handling) 理論に基づいて前記非線形システムの P I D 制御に係る比例ゲイン、積分ゲイン、および微分ゲインを算出する G M D H - P I D 調整部と、
前記偏差に基づいて、前記算出された比例ゲイン、積分ゲイン、および微分ゲインで前記非線形システムを P I D 制御する P I D 制御部とを備えている
ことを特徴とする非線形 P I D コントローラ。

10

【請求項 2】

請求項 1 に記載の非線形 P I D コントローラにおいて、
前記 G M D H - P I D 調整部が、各 G M D H ネットワークで前記 3 値から前記各ゲインを個別に算出するものであり、
前記各 G M D H ネットワークが、アダライン演算子が階層的に接続されてなる
ことを特徴とする非線形 P I D コントローラ。

【請求項 3】

請求項 2 に記載の非線形 P I D コントローラにおいて、
前記各 G M D H ネットワークが、前記 3 値のいずれか二つが入力される第 1 および第 2 のアダライン演算子と、前記第 1 のアダライン演算子の出力および前記第 2 のアダライン演算子の出力が入力される第 3 のアダライン演算子とからなる
ことを特徴とする非線形 P I D コントローラ。

20

【請求項 4】

請求項 1 から 3 のいずれか一つに記載の非線形 P I D コントローラにおける G M D H の重み係数を決定する方法において、
前記 3 値から前記非線形システムの P I D 制御に係る比例ゲイン、積分ゲイン、および微分ゲインを個別に算出する各 C M A C (Cerebellar Model Articulation Controller) をコンピュータ上に構築し、
前記各 C M A C に前記非線形システムの P I D 制御に係る学習をさせ、
学習済みの前記各 C M A C の入出力サンプルデータに基づいて、前記 3 値から前記各ゲインを算出する各 G M D H ネットワークの重み係数を決定する
ことを特徴とする G M D H 重み係数決定方法。

30

【請求項 5】

請求項 4 に記載の G M D H 重み係数決定方法において、
前記 3 値のいずれか二つが入力される 2 個のアダライン演算子が 1 階層目に配置され、前記 1 階層目の 2 個のアダライン演算子の出力が入力されるアダライン演算子が 2 階層目に配置された 3 つの G M D H ネットワークのそれぞれについて、前記入出力サンプルデータに基づいて、前記第 1 階層目の 2 個のアダライン演算子の重み係数を決定し、
前記 1 階層目の 2 個のアダライン演算子の重み係数の決定後に、再度前記入出力サンプルデータに基づいて、前記 2 階層目のアダライン演算子の重み係数を決定し、
すべてのアダライン演算子の重み係数の決定後に、前記 3 つの G M D H ネットワークのそれぞれの寄与率を計算し、
前記 3 つの G M D H ネットワークの中から前記寄与率が高いいずれか一つを選択し、当該選択した G M D H ネットワークの重み係数を、前記 3 値から前記各ゲインを算出する各 G M D H ネットワークの重み係数として決定する
ことを特徴とする G M D H 重み係数決定方法。

40

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、P I D コントローラに関し、特に、非線形システムを P I D 制御する非線形

50

P I Dコントローラに関する。

【背景技術】

【0002】

産業プロセスにおいては、制御構造の簡便さから今なおP I D制御が幅広く用いられている。しかしながら、実システムの多くは非線形性を有しており、固定のP I Dパラメータによる制御では、常に良好な制御結果を得ることが困難である。このような非線形システムに対する有効な制御手法として、ニューラルネットワーク (Neural Network : NN) の一種である小脳演算モデル (Cerebellar Model Articulation Controller : CMAC) を用いたCMAC - P I D制御器が提案されている (たとえば、非特許文献1参照)。しかしながら、最適な制御パラメータを習得するためにオンライン学習が必要なことや、制御系の実現に多大なメモリを要することが実用化への大きな障害となっている。前者の問題の解決法として、CMAC - F R I Tがすでに報告されている (たとえば、非特許文献2参照)。これは対象とするシステムの閉ループデータから制御パラメータを直接的に算出するF R I T (Fictitious Reference Iterative Tuning) 法をCMACの学習に取り入れることで、システムのエミュレータを構築することなく、オフラインでCMACの学習をすることが可能となる手法である。

10

【先行技術文献】

【非特許文献】

【0003】

【非特許文献1】黒住亮太・山本透：「小脳演算モデルを用いたインテリジェントP I D制御系の一設計」、電気学会論文誌.C, Vol.125, No.4, 607-615, (2005)

20

【非特許文献2】脇谷・大西・山本：「非線形システムに対するF R I Tを用いたCMAC - P I D制御器の一設計」、計測自動制御学会論文集、Vol.48, No.12(2012)

【発明の概要】

【発明が解決しようとする課題】

【0004】

CMAC - P I D制御器はP I Dの各パラメータごとに3次元の入力空間を有するCMACを用いるためその荷重表は膨大なものとなる。このため、CMAC - P I D制御器をマイコンなどで実現する場合、CMACの荷重表を格納するために大容量のメモリが必要となる。しかしながら、大容量のメモリをマイコンに搭載することはコスト面から不利である。

30

【0005】

上記問題に鑑み、本発明は、CMAC - P I D制御器のメモリ容量の問題を解決することを目的とし、より少ない容量のメモリで動作可能な非線形P I Dコントローラを提供することを課題とする。

【課題を解決するための手段】

【0006】

本発明の一面に従った非線形P I Dコントローラは、制御対象の非線形システムの出力値と目標値との偏差を算出する偏差演算部と、前記偏差の微分値を算出する微分演算部と、前記目標値、前記偏差、および前記微分値の3値からG M D H (Group Method of Data Handling) 理論に基づいて前記非線形システムのP I D制御に係る比例ゲイン、積分ゲイン、および微分ゲインを算出するG M H D - P I D調整部と、前記偏差に基づいて、前記算出された比例ゲイン、積分ゲイン、および微分ゲインで前記非線形システムをP I D制御するP I D制御部とを備えている。

40

【0007】

これによると、G M D Hを用いてP I D制御に係るゲイン調整が行われる。G M D Hはネットワークモデルの一つであり、たとえば学習を通して構築されたCMACを非線形モデルとして近似表現することができる。G M D Hでは重み係数のみをメモリに記憶しておけばよいため、必要なメモリ容量を削減することができる。また、操業中のP I Dゲインの算出に関わる計算コストも大幅に削減することが可能となる。

50

【 0 0 0 8 】

具体的には、前記 G M D H - P I D 調整部が、各 G M D H ネットワークで前記 3 値から前記各ゲインを個別に算出するものであり、前記各 G M D H ネットワークが、アダライン演算子が階層的に接続されてなる。

【 0 0 0 9 】

さらに具体的には、前記各 G M D H ネットワークが、前記 3 値のいずれか二つが入力される第 1 および第 2 のアダライン演算子と、前記第 1 のアダライン演算子の出力および前記第 2 のアダライン演算子の出力が入力される第 3 のアダライン演算子とからなる。

【 0 0 1 0 】

また、本発明の別の局面に従うと、上記の非線形 P I D コントローラにおける G M D H の重み係数を決定する方法は、前記 3 値から前記非線形システムの P I D 制御に係る比例ゲイン、積分ゲイン、および微分ゲインを個別に算出する各 C M A C (Cerebellar Model Articulation Controller) をコンピュータ上に構築し、前記各 C M A C に前記非線形システムの P I D 制御に係る学習をさせ、学習済みの前記各 C M A C の入出力サンプルデータに基づいて、前記 3 値から前記各ゲインを算出する各 G M D H ネットワークの重み係数を決定する。

10

【 0 0 1 1 】

上記の G M D H 重み係数決定方法において、前記 3 値のいずれか二つが入力される 2 個のアダライン演算子が 1 階層目に配置され、前記 1 階層目の 2 個のアダライン演算子の出力が入力されるアダライン演算子が 2 階層目に配置された 3 つの G M D H ネットワークのそれぞれについて、前記入出力サンプルデータに基づいて、前記第 1 階層目の 2 個のアダライン演算子の重み係数を決定し、前記 1 階層目の 2 個のアダライン演算子の重み係数の決定後に、再度前記入出力サンプルデータに基づいて、前記 2 階層目のアダライン演算子の重み係数を決定し、すべてのアダライン演算子の重み係数の決定後に、前記 3 つの G M D H ネットワークのそれぞれの寄与率を計算し、前記 3 つの G M D H ネットワークの中から前記寄与率が高いいずれか一つを選択し、当該選択した G M D H ネットワークの重み係数を、前記 3 値から前記各ゲインを算出する各 G M D H ネットワークの重み係数として決定する。

20

【 発明の効果 】

【 0 0 1 2 】

本発明によると、C M A C - P I D 制御器のメモリ容量の問題が解決され、より少ない容量のメモリで動作可能な非線形 P I D コントローラを実現することができる。

30

【 図面の簡単な説明 】

【 0 0 1 3 】

【 図 1 】 C M A C - P I D 制御器のブロック線図

【 図 2 】 C M A C - P I D 調整部の構成図

【 図 3 】 C M A C を示す模式図

【 図 4 】 C M A C - F R I T のブロック線図

【 図 5 】 本発明の一実施形態に係る非線形 P I D コントローラ (G M D H - P I D 制御器) のブロック線図

40

【 図 6 】 G M D H ネットワークを示す模式図

【 図 7 】 アダライン演算子の構造を示す模式図

【 図 8 】 B i l l i n e a r モデルの静特性を示すグラフ

【 図 9 】 固定 P I D 制御器による制御結果を示すグラフ

【 図 1 0 】 C M A C - P I D 制御器による制御結果を示すグラフ

【 図 1 1 】 図 1 0 の制御結果に対応する P I D ゲインの推移を示すグラフ

【 図 1 2 】 G M D H - P I D 制御器による制御結果を示すグラフ

【 図 1 3 】 図 1 2 の制御結果に対応する P I D ゲインの推移を示すグラフ

【 図 1 4 】 固定 P I D 制御器による別の制御結果を示すグラフ

【 図 1 5 】 C M A C - P I D 制御器による汎化能力検証の制御結果を示すグラフ

50

【図16】GM DH - PID制御器による汎化能力検証の制御結果を示すグラフ

【図17】図15の制御結果に対応するPIDゲインの推移を示すグラフ

【図18】図16の制御結果に対応するPIDゲインの推移を示すグラフ

【発明を実施するための形態】

【0014】

本発明の一実施形態に係る非線形PIDコントローラは、まずCMAC - PID制御器を生成し、当該CMAC - PID制御器の学習を行った後、CMACをGM DH理論に基づく非線形関数で近似することで生成される。以下、図面を参照しながら、本発明の一実施形態に係る非線形PIDコントローラの生成過程および構成について説明する。なお、同様の構成要素には同様の符号を付してその説明は繰り返さない。

10

【0015】

CMAC - PID制御系の設計

制御則として、次式で与えられる速度型PID制御則を考える。

【0016】

【数1】

$$u(t) = u(t-1) + K_I(t)e(t) + K_P(t)\Delta e(t) + K_D(t)\Delta^2 e(t) \quad (1)$$

20

【0017】

ここで、 $u(t)$ は制御入力であり、 $e(t)$ は次式で定義される制御偏差を示す。 $K_P(t)$ 、 $K_I(t)$ 、 $K_D(t)$ はそれぞれ各時刻における比例ゲイン、積分ゲイン、微分ゲインである。

【0018】

【数2】

$$e(t) := r(t) - y(t) \quad (2)$$

30

【0019】

(2)式において、 $r(t)$ および $y(t)$ はそれぞれ目標値、システム出力を示している。さらに Δ は差分演算子であり $\Delta := 1 - z^{-1}$ で定義される。また、 z^{-1} は遅延演算子を表し $z^{-1}e(t) = e(t-1)$ を意味する。CMAC - PID制御器では、時刻 t におけるPIDゲイン $K_P(t)$ 、 $K_I(t)$ 、 $K_D(t)$ をCMAC調節器によって調整する。

【0020】

<CMAC>

CMAC - PID制御器のブロック線図を図1に示す。CMAC - PID制御器10は、 $r(t)$ から $y(t)$ を減算して $e(t)$ を算出する偏差演算部12と、 $e(t)$ の微分値 $\Delta e(t)$ を算出する微分演算部14(上記の差分演算子に相当)と、 $r(t)$ 、 $e(t)$ 、 $\Delta e(t)$ から $K_P(t)$ 、 $K_I(t)$ 、 $K_D(t)$ を算出するCMAC - PID調整部16と、 $e(t)$ に基づいて、算出された $K_P(t)$ 、 $K_I(t)$ 、 $K_D(t)$ で非線形システム100をPID制御するPID制御部18とを備えている。

40

【0021】

CMAC - PID調整部16は、図2に示すように、 $K_P(t)$ 、 $K_I(t)$ 、 $K_D(t)$ のそれぞれを学習する3つのCMAC162によって構成される。

【0022】

図3にCMACの具体的な構造を示す。まず、入力空間に(3, 6)という入力を与えられると、その入力をラベル集合{B, F, J}、{c, g, k}に基づいて、荷重表が

50

ら 8, 9, 3 を参照し、その合計 20 を出力する。この入力に対して、たとえば教師信号が 14 であった場合、出力と教師信号の差を荷重表の枚数で割った値すなわち $(14 - 20) / 3$ を参照した荷重表に加え、学習を行う。ここでは説明の簡単化のために 2 次元の入力空間について述べたが、CMAC 162 の入力空間は $r(t)$ 、 $e(t)$ 、 $e(t)$ の 3 次元で構成し、(3) 式に示すように N 枚の荷重表 W に離散化する。詳細については非特許文献 1 を参照されたい。

【0023】

【数 3】

$$\left. \begin{aligned} K_P(t) &= \sum_{h=1}^N W_{P,h} \\ K_I(t) &= \sum_{h=1}^N W_{I,h} \\ K_D(t) &= \sum_{h=1}^N W_{D,h} \end{aligned} \right\} \quad (3)$$

10

20

【0024】

< CMAC - FRIT によるオフライン学習 >

CMAC のオフライン学習には、モデルフリー型制御設計法の一つである FRIT を適用する。FRIT では、1 回の実験データ（閉ループデータ）を用いて、制御パラメータを直接的に算出することができる。FRIT を用いた CMAC のオフライン学習のブロック線図を図 4 に示す。図 4 において、 $r_0(t)$ 、 $e_0(t)$ 、 $u_0(t)$ 、 $y_0(t)$ は初期実験によって得られた操業データを示している。さらにチルダ $r(t)$ は擬似参照入力を示し、次式で表される。

【0025】

30

【数 4】

$$\tilde{r}(t) = C^{-1}(z^{-1})u_0(t) + y_0(t) \quad (4)$$

【0026】

ここで $C(z^{-1})$ は PID コントローラの多項式を表している。CMAC - FRIT では、PID ゲインの計算に必要な CMAC の荷重を以下の調整則によりオフライン学習を行う。

【0027】

40

【数 5】

$$\left. \begin{aligned} W_{P,h}^{new} &= W_{P,h}^{old}(t) - \eta_P \frac{\partial J(t+1)}{\partial K_P^{old}(t)} \frac{1}{N} \\ W_{I,h}^{new} &= W_{I,h}^{old}(t) - \eta_I \frac{\partial J(t+1)}{\partial K_I^{old}(t)} \frac{1}{N} \\ W_{D,h}^{new} &= W_{D,h}^{old}(t) - \eta_D \frac{\partial J(t+1)}{\partial K_D^{old}(t)} \frac{1}{N} \end{aligned} \right\} \quad (5)$$

10

【0028】

(5)式において η_P 、 η_I 、 η_D は学習係数を表している。また、 $J(t)$ は評価関数で次式で表される。

【0029】

【数 6】

$$J(t) = \frac{1}{2} \{y_0(t) - y_r(t)\}^2 \quad (6)$$

20

【0030】

(6)式において、 $y_r(t)$ は(7)式で表される参照モデル $G_m(z^{-1})$ からの出力である。

【0031】

【数 7】

$$G_m(z^{-1}) = \frac{z^{-1}P(1)}{P(z^{-1})} \quad (7)$$

30

【0032】

ここで $P(z^{-1})$ は参照モデルの特性多項式であり、次式で表される。

【0033】

【数 8】

$$P(z^{-1}) = 1 + p_1 z^{-1} + p_2 z^{-2} \quad (8)$$

【0034】

40

【数 9】

$$\left. \begin{aligned} p_1 &= -2 \exp\left(-\frac{\rho}{2\mu}\right) \cos\left(\frac{\sqrt{4\mu-1}}{2\mu} \rho\right) \\ p_2 &= \exp\left(-\frac{\rho}{\mu}\right) \\ \rho &:= T_s / \sigma \\ \mu &:= 0.25(1 - \delta) + 0.51\delta \end{aligned} \right\} \quad (9)$$

50

【 0 0 3 5 】

(9) 式において T_s はサンプリング時間、 τ 、 σ はそれぞれ制御系の立ち上がり特性、減衰特性に関係するパラメータを示しており、設計者が任意に設定する。特に τ は制御系の出力がステップ状の目標値の約 60% に達するまでの時間を示している。 σ は減衰特性であり、0 ~ 1.0 程度で設計するのがよいとされている。C M A C - F R I T の詳細については、非特許文献 2 を参照されたい。

【 0 0 3 6 】

G M D H を用いた非線形 P I D 制御器の構築

< 非線形 P I D コントローラ >

上述したように、本実施形態に係る非線形 P I D コントローラは、C M A C - P I D で構築された各 P I D ゲインを、(1 0) 式で与えられる非線形モデルとして記述することで生成される。つまり各時刻の P I D ゲインを $r(t)$ 、 $e(t)$ 、 $\Delta e(t)$ の非線形関数として記述する。その際、ネットワークモデルの一つである G M D H (Group Method of Data Handling) を用いる。

【 0 0 3 7 】

【 数 1 0 】

$$\left. \begin{aligned} K_P(t) &= f_P(r(t), e(t), \Delta e(t)) \\ K_I(t) &= f_I(r(t), e(t), \Delta e(t)) \\ K_D(t) &= f_D(r(t), e(t), \Delta e(t)) \end{aligned} \right\} \quad (10) \quad 20$$

【 0 0 3 8 】

< G M D H >

G M D H はシステムの入出力データを用いて、非線形モデルを構成する際に用いられるネットワークモデルの一つである。G M D H ネットワークは、モデルを多項式として表現することが比較的容易であるという特徴をもっている。したがって、非線形コントローラである図 1 の C M A C - P I D 制御器 1 0 を近似的に表現することができる。

【 0 0 3 9 】

本実施形態に係る非線形 P I D コントローラである G M D H - P I D 制御器のブロック線図を図 5 に示す。G M D H - P I D 制御器 1 0 A は、 $r(t)$ から $y(t)$ を減算して $e(t)$ を算出する偏差演算部 1 2 と、 $e(t)$ の微分値 $\Delta e(t)$ を算出する微分演算部 1 4 (上記の差分演算子に相当) と、 $r(t)$ 、 $e(t)$ 、 $\Delta e(t)$ から G M D H 理論に基づいて $K_P(t)$ 、 $K_I(t)$ 、 $K_D(t)$ を算出する G M D H - P I D 調整部 1 6 A と、 $e(t)$ に基づいて、算出された $K_P(t)$ 、 $K_I(t)$ 、 $K_D(t)$ で非線形システム 1 0 0 を P I D 制御する P I D 制御部 1 8 とを備えている。

【 0 0 4 0 】

G M D H ネットワークは、図 6 に示すようにアダライン演算子 (N-Adaline) 1 6 4 の適切な組み合わせによって構成される階層型のネットワークである。アダライン演算子 1 6 4 は図 7 に示すように 2 入力 1 出力からなるユニットで、入力変数を $x_1(t)$ 、 $x_2(t)$ 、出力変数を $z(t)$ とする。 $z(t)$ は次式のように表現される。

【 0 0 4 1 】

【数 1 1】

$$\left. \begin{aligned}
 \mathbf{w} &= [w_0, w_1, w_2, w_3, w_4, w_5]^T \\
 \mathbf{x}(t) &= [x_1(t), x_1^2(t), x_1(t)x_2(t), x_2^2(t), x_2(t), 1]^T \\
 \mathbf{X} &= [\mathbf{x}(1), \mathbf{x}(2), \dots, \mathbf{x}(t), \dots, \mathbf{x}(n)]^T \\
 \mathbf{z} &= [z(1), z(2), \dots, z(t), \dots, z(n)]^T \\
 \mathbf{z} &= \mathbf{X} \cdot \mathbf{w}
 \end{aligned} \right\} \quad (11)$$

【0 0 4 2】

ただし、 w_i ($i = 0, \dots, 5$) は重み係数を示す。また図中の S_q は 2 乗演算を示している。

【0 0 4 3】

重みの算出には次式の最小二乗法を用いることができる。もちろん、これ以外の方法で重みを算出してよい。

【0 0 4 4】

【数 1 2】

$$\mathbf{w} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{z} \quad (12)$$

【0 0 4 5】

GMDH ネットワークは図 6 に示す通り 3 種類の組み合わせが存在する。その中から次式で定義される寄与率が最も高いもの（たとえば、図 6 において破線で囲んだネットワーク）を GMDH - PID 調整部 16A に搭載すべき GMDH ネットワーク 165 として決定する。

【0 0 4 6】

【数 1 3】

$$R^2 = 1 - \left\{ \sum_{t=1}^{N_s} (z(t) - \tilde{z}(t))^2 / \sum_{t=1}^{N_s} (z(t) - \bar{z})^2 \right\} \quad (13)$$

【0 0 4 7】

上式においては N_s は入出力データのデータ数、チルダ $z(t)$ は出力の推定値、 \bar{z} は出力データの平均値を表している。

【0 0 4 8】

< コントローラ的设计 >

上述の GMDH を用いて、以下の手順に従い PID ゲインの非線形モデルを設計する。

step 1) 学習済みの CMAC の荷重表に $r_0(t)$ 、 $e_0(t)$ 、 $e_0(t)$ を入力し、チルダ $K_p(t)$ 、チルダ $K_I(t)$ 、チルダ $K_D(t)$ を算出する。

step 2) (12) 式によって、重み w を決定する。式 (11) において、 $x_1(t)$ 、 $x_2(t)$ は $r_0(t)$ 、 $e_0(t)$ 、 $e_0(t)$ のうちの二つであり、 $z(t)$ はチルダ $K_p(t)$ 、チルダ $K_I(t)$ 、チルダ $K_D(t)$ である。重み w は PID ゲインそれぞれにおいて異なった重みである。

step 3) PID ゲインそれぞれの推定値を用いて 1 階層目の GMDH と最小二乗法に

10

20

30

40

50

より、2階層目の重み w を決定する。

step 4) それぞれの組み合わせにおける推定値を算出し、式 (13) により寄与率を計算する。この寄与率が最も高いものを PID ゲインの非線形モデルとする。

【0049】

本手法によって、CMAC を用いて算出していた各 PID ゲインを GMDH ネットワークにより表現することが可能となり、CMAC に必要であったメモリを大幅に削減することができる。

【0050】

CMAC - PID 制御器 10 では、 $K_p(t)$ 、 $K_I(t)$ 、 $K_D(t)$ を算出する3つの CMAC 162 が、それぞれ、3次元の入力空間（その大きさは、たとえば 11^3 である）および3枚の荷重表を有するため、メモリに記憶すべきパラメータは $11979 (= 11^3 \times 3 \times 3)$ 個もある。このため、CMAC - PID 制御器 10 をマイコンなどで実現する場合、たとえば、1パラメータを64ビットデータとして、マイコンにはおよそ93Kバイトものメモリ容量が必要となる。一方、GMHD - PID 制御器 10A では、GMHD - PID 調整部 16A が、 $r(t)$ 、 $e(t)$ 、 $e(t)$ から3つの GMDH ネットワーク 165 のそれぞれで $K_p(t)$ 、 $K_I(t)$ 、 $K_D(t)$ を算出する。したがって、メモリに記憶すべきパラメータは各 GMDH ネットワーク 165 における各アダプティブ演算子 164 の6つの重み係数、すなわち、 $54 (= 3 \times 3 \times 6)$ 個のパラメータだけでよい。このため、GMHD - PID 制御器 10A をマイコンなどで実現する場合、たとえば、1パラメータを64ビットデータとして、マイコンに必要なメモリ容量はわずか432バイトで済む。このように、本実施形態に係る GMHD - PID 制御器 10A では必要なメモリ容量を大幅に低減することができる。

10

20

【0051】

制御結果

本実施形態の有効性を検証するため、GMHD - PID 制御器 10A を次式で与えられる Bilinear モデルに適用する。

【0052】

【数14】

$$\begin{aligned}
 y(t) &= 0.4y(t-1) - 0.09y(t-2) \\
 &+ 0.3y(t-1) - 0.1u(t-2) \\
 &+ 0.1u(t-1)y(t-1) \\
 &+ 0.05y(t-2)u(t-2) \\
 &+ \xi(t)
 \end{aligned}
 \tag{14}$$

30

【0053】

ここで $\xi(t)$ は平均 0、分散 1.0×10^{-3} のガウス性白色雑音を示している。また、Bilinear モデルの静特性を図 8 に示す。さらに、各ステップにおける目標値を (15) 式のように設定した。

40

【0054】

【数 15】

$$r(t) = \begin{cases} 1.0 & (0 < t \leq 50) \\ 3.0 & (50 < t \leq 100) \\ 5.0 & (100 < t \leq 150) \\ 3.5 & (150 < t \leq 200) \end{cases} \quad (15)$$

【0055】

10

< C M A C のオフライン学習 >

はじめに C M A C のオフライン学習に必要な入出力データを得るために、閉ループ系の安定性を十分に考慮した固定 P I D ゲインによる制御を行った。このとき、P I D ゲインの決定については $y = 5$ 付近のデータを用いて、C H R (Chien, Hrones, Reswick) 法を用いて算出した。P I D ゲインは次式の通りである。

【0056】

【数 16】

$$K_P = 0.5, K_I = 0.1, K_D = 0.02 \quad (16)$$

20

【0057】

このパラメータでの固定 P I D 制御器による制御結果を図 9 に示す。結果から低い目標値に対しては、目標値追従性においても必ずしも良好とはいえない。

【0058】

次に、固定 P I D 制御による実験データに基づいて、C M A C - F R I T により C M A C をオフライン学習する。その際の参照モデルは (17) 式のように設計した。

【0059】

【数 17】

$$P(z^{-1}) = 1 - 1.3406z^{-1} + 0.4493z^{-2} \quad (17)$$

30

【0060】

また、学習係数は P, I, D とともに 10^{-3} に設計した。図 10、図 11 に C M A C - P I D での制御結果、P I D ゲインの推移を示す。C M A C がオフラインで十分学習され、低い目標値においても良好な制御ができていることがわかる。

【0061】

< G M D H によるコントローラの設計 >

オフライン学習された C M A C を G M D H により、ネットワークとして構成し、(15) 式で与えられる目標値に対してシミュレーションを行った。

40

【0062】

シミュレーションの結果と P I D ゲインの推移を図 12、図 13 に示す。結果から G M D H - P I D が C M A C - P I D とほぼ同じ制御性能を得られていることがわかる。

【0063】

< 汎化能力の検証 >

(18) 式に示す未学習の目標値に対して制御を行うことで、汎化能力について考察する。固定 P I D での制御結果を図 14 に、C M A C - P I D での制御結果を図 15 に、本実施形態に係る G M D H - P I D での制御結果を図 16 に示す。また、C M A C - P I D での P I D ゲインの推移を図 17 に、本実施形態に係る G M D H - P I D での P I D ゲイ

50

ンの推移を図 18 に示す。

【 0 0 6 4 】

【 数 1 8 】

$$r(t) = \begin{cases} 1.2 & (0 < t \leq 50) \\ 3.0 & (50 < t \leq 100) \\ 4.5 & (100 < t \leq 150) \\ 2.5 & (150 < t \leq 200) \end{cases} \quad (18)$$

10

【 0 0 6 5 】

これらの制御結果により、本実施形態に係る GMDH - PID が CMAC - PID とほぼ同等の制御性能を有していると考えられる。このシミュレーションでは CMAC の荷重表に要したメモリ量が倍精度浮動小数点数で 264 K バイトだったことに対し、本実施形態での GMDH の重みでは 432 バイトとなり、99.8% メモリを削減した。また、制御対象によっては CMAC のメモリが増大する場合もあるため、メモリ容量を削減するために本実施形態は有用であるといえる。

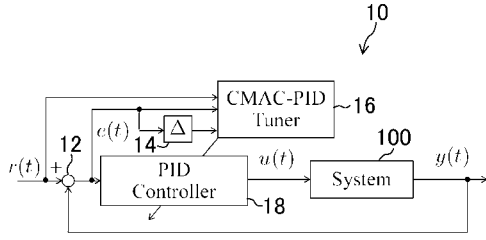
【 符号の説明 】

【 0 0 6 6 】

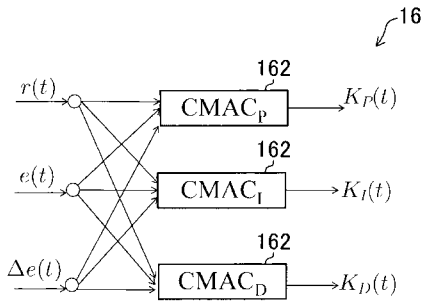
- 10A GMDH - PID 制御器 (非線形 PID コントローラ)
- 12 偏差演算部
- 14 微分演算部
- 16A GMDH - PID 調整部
- 162 CMAC
- 164 アダプティブ演算子
- 165 GMDH ネットワーク
- 18 PID 制御部
- 100 非線形システム

20

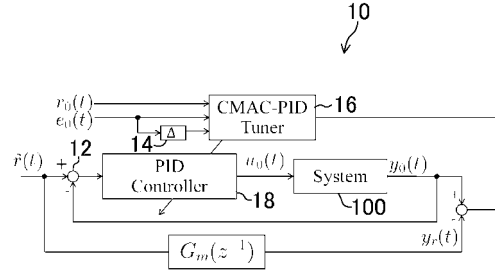
【 図 1 】



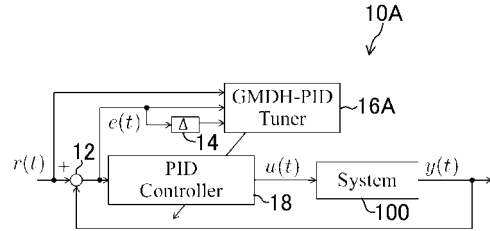
【 図 2 】



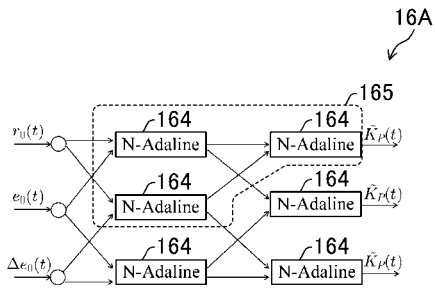
【 図 4 】



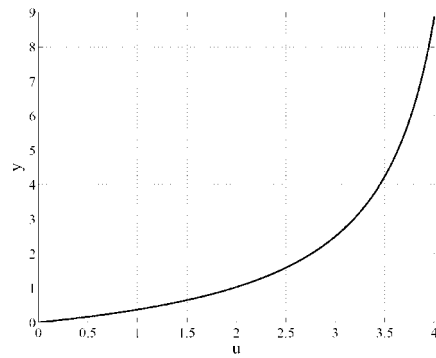
【 図 5 】



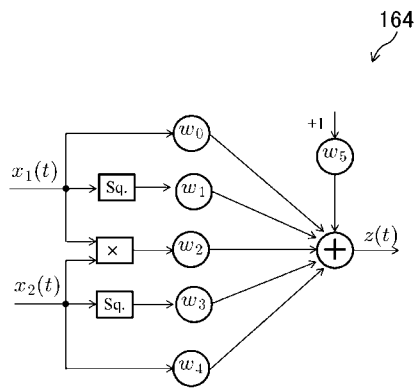
【 図 6 】



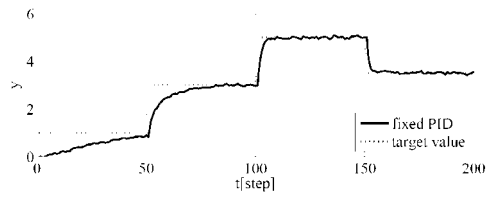
【 図 8 】



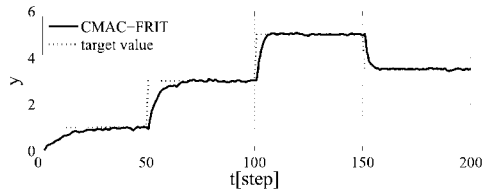
【 図 7 】



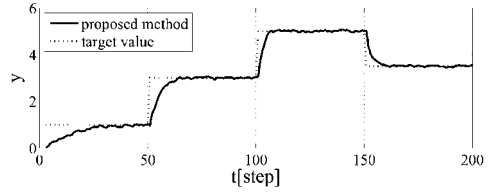
【 図 9 】



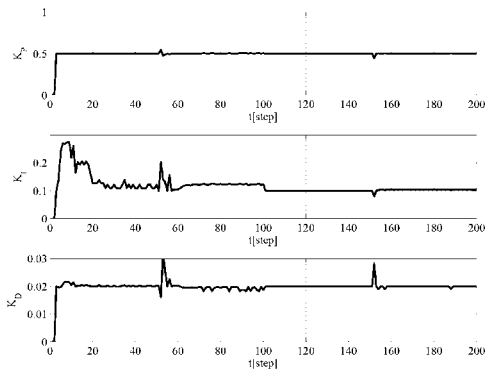
【 図 1 0 】



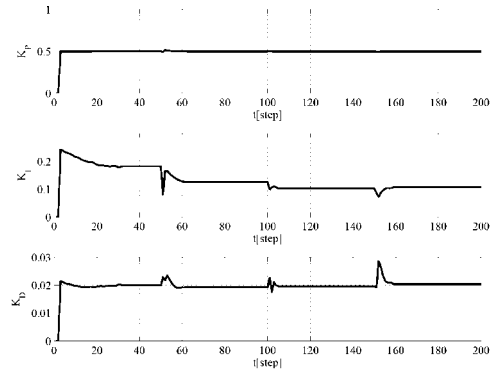
【 図 1 2 】



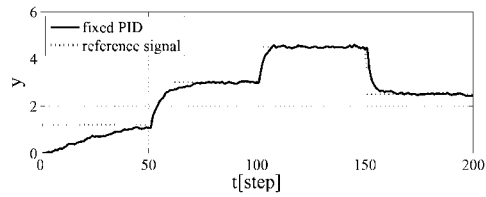
【 図 1 1 】



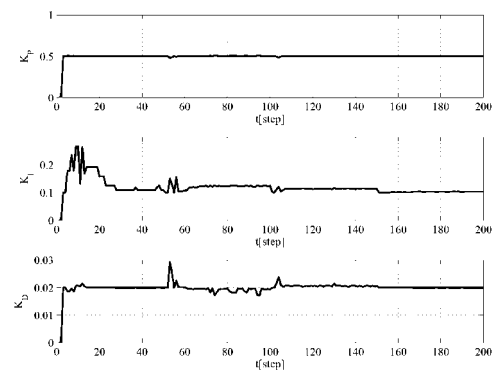
【 図 1 3 】



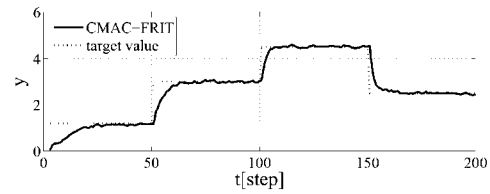
【 図 1 4 】



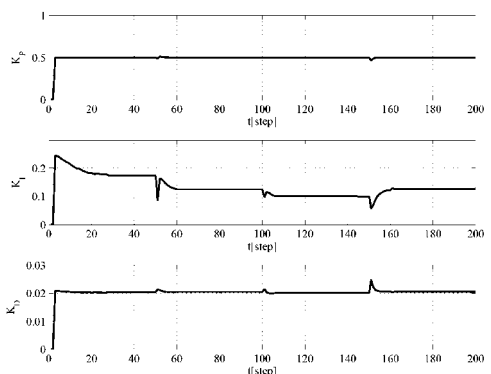
【 図 1 7 】



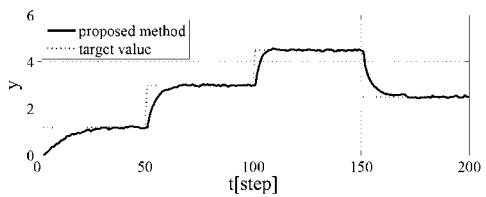
【 図 1 5 】



【 図 1 8 】

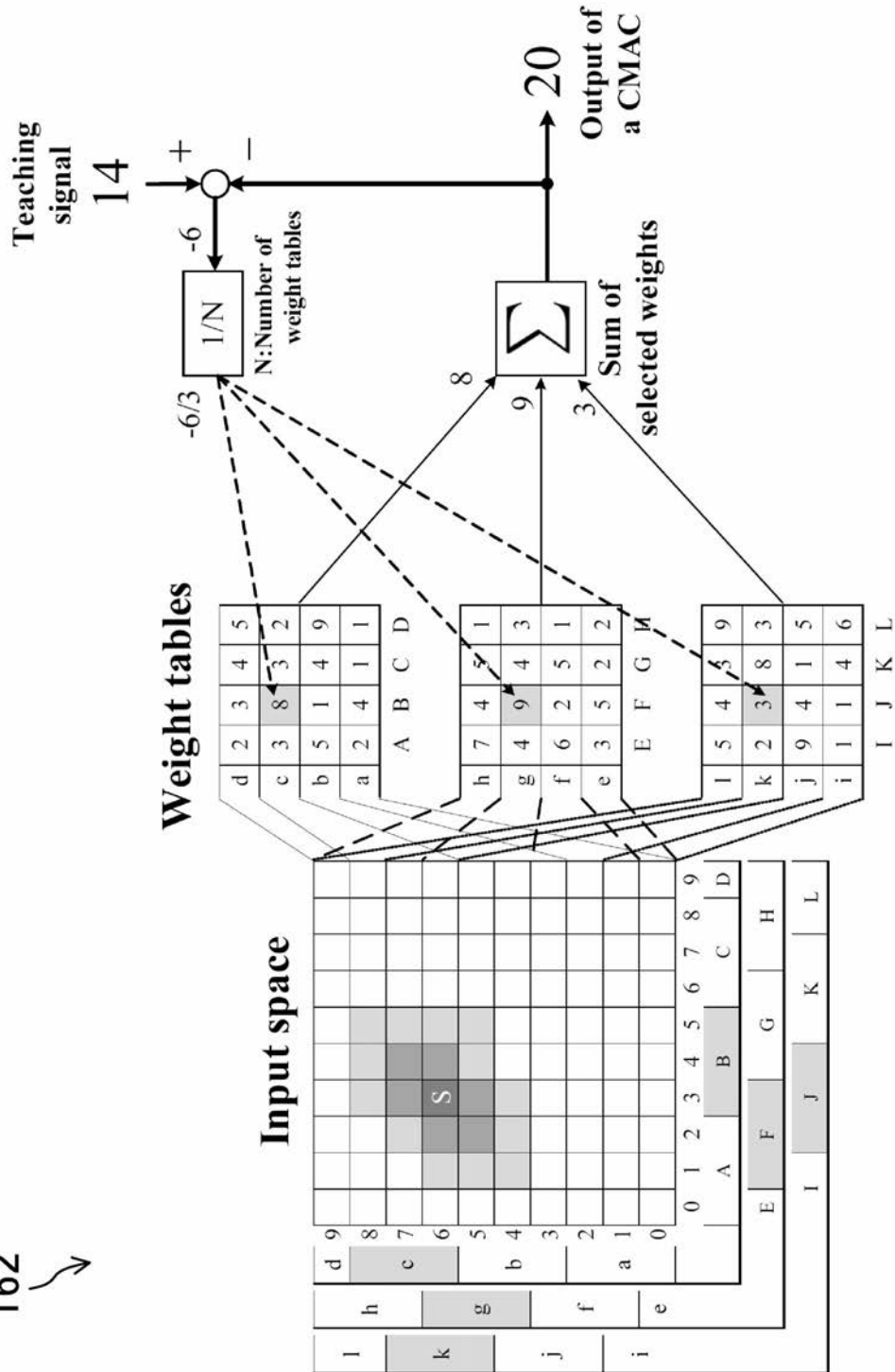


【 図 1 6 】



【 図 3 】

162 ↗



フロントページの続き

特許法第30条第2項適用申請有り 平成24年8月30日に公益社団法人 計測自動制御学会の第22回インテリジェント・システム・シンポジウム、Fuzzy, Artificial Intelligence, Neural Networks and Computational Intelligence (FAN Symposium 2012 in Okinawa) 講演予稿集にて発表 平成24年9月5日に一般社団法人 電気学会の平成24年 電気学会 電子・情報・システム部門大会の講演予稿集、第209 - 212頁にて発表 平成24年11月3日にVBL院生会研究シーズのビジネス化コンペにて発表 平成24年11月12日にICONIP (The International Conference on Neural Information Processing) 2012の19th International Conference 講演予稿集、第169 - 176頁にて発表 平成24年11月17日にIEEE広島支部の学生の、学生による、社会のための第14回広島支部学生シンポジウム論文集、第23 - 26頁にて発表