GA と GMDH ネットワークを用いた P I D 制御系の一設計法

正 員 坂口 彰浩* 学生員 片山 優**
正 員 山本 透**

A Design Scheme of PID Control Systems Using a GA and a GMDH Network

Akihiro Sakaguchi*, Member, Masaru Katayama**, Student Member, Toru Yamamoto**, Member

PID control scheme has been widely used in most process control system. Because the PID controller has a simple structure, and have three parameters at the most. However, since it is difficult to determine suitable parameters, a lot of researches have been reported with respect to tuning scheme of PID parameters. In this paper, a design scheme of PID control system using a GMDH (Group Method of Data Handling) network and a genetic algorithm (GA) is proposed. Furthermore, the effectiveness of the newly proposed scheme is evaluated on a simulation example.

キーワード：遺伝的アルゴリズム, GMDH ネットワーク, PID 制御, 一般化予測制御

Keywords: Genetic Algorithm, GMDH Network, PID Control, Generalized Predictive Control

1. 緒 言

モデル予測制御法⁽¹⁾の一つとして、一般化予測制御 (GPC) 法⁽²⁾が提案されている。この方法は、多段予測を行うため、むだ時間先の出力の予測値のみを用いるのではなく、数段にわたって予測される出力が、参照モデルの応答に追従するように制御入力を得る。このことから、むだ時間が曖昧である制御系のみならず、むだ時間が変動するような制御系においてもロバストであるため、プロセス制御系においては有効な手法の一つとされている。しかしながら、この手法は制御系の構造が複雑であり、また、多くの設計パラメータを調整しなければならない。それゆえに、これらの最適なパラメータの調整に当たり、多くの試行錯誤が必要とされている。

他方、現在に至っても、PID 制御法⁽³⁾は、大部分のプロセス制御系を扱う現場において、幅広く用いられている。これは、PID 制御系の構造が非常に簡単であることや、パラメータの数がたかだか 3 個であり、それらの物理的意味が明確であるためである。しかしながら、実システムにおける多くの場合、十分に制御系が安定となる範囲内のパ

ラメータを用いて制御を行っており、必ずしも最適なパラメータを与えているとは限らない。実際、この 3 つのパラメータの最適な値を見つけることは、非常に難しい問題である。したがって、現在までに様々なパラメータ決定法が提案されている。この問題に対して、GPC 法との関連に基いて PID パラメータを調整する方法⁽⁴⁾が考察され、その有効性が検証されている。しかしながら、この方法においても、最適な設計パラメータを与えるには、設計パラメータの設計に際し、試行錯誤を必要とする。

プロセス制御系には、非線形特性を有した制御系が多いことから、本論文では非線形系の制御対象を扱う。このような系に対して、ロバスト PID 制御法やセルグチューニング PID 制御法が有効な手法として提案されている。しかしながら、ロバスト PID 制御法では、制御性能を大きく左右する振動範囲の決定は非常に難しい問題となっており、また、保守的な制御系設計となってしまうため、高精度の制御性能を得ることが難しい場合がある。他方、セルフチューニング PID 制御法は、逐次的にパラメータを推定し、制御対象の変動に応じて制御機が適応調整される。しかし、制御性能はその推定精度に依存することになり、場合によっては不安定な制御系に陥ってしまうことがある。本論文では、GMDH(Group Method of Data Handling) ネットワーク⁽⁵⁾⁻⁽⁸⁾と遺伝的アルゴリズム (GA)⁽⁹⁾を用いることで、非線形システムに対する PID 制御系の一設計法を提案する。具体的には、GPC 法との関連に基いて PID パラメータを調整するが、この手法に含まれる設計パラメータの最適値を、GA を用いて探索する。このとき、非線形系を対象と

* 佐世保工業高等専門学校

〒 857-1193 長崎県佐世保市沖新町 1-1
Sasebo National College of Technology
1-1, Okishincho, Sasebo, Nagasaki 857-1193

** 広島大学

〒 739-8524 広島県東広島市鏡山 1-1-1
Hiroshima University
1-1-1 Kagamiyama, Higashi-Hiroshima, Hiroshima, 739-8524

するため、GMDH ネットワークにより詳細モデルを構築し、これを用いて適応度関数を計算する。以上により、得られた最適な設計パラメータを用いることで、GPC 則に基づいて PID パラメータを計算することができる。本提案手法では、設計パラメータを GA により調整するため試行錯誤を必要とせず、また、オフライン学習により調整するために信頼性の高い PID パラメータを決定することができる。最後に、数値例により本手法の有効性を定量的に検証する。

2. GA による PID パラメータの調整

〈2・1〉 PID パラメータの調整法の概要 化学プロセスなどに代表されるプロセス制御系の大部分において PID 制御法が用いられている。しかしながら、PID 制御系を設計する上で、PID パラメータをどのように決定するかは、制御性能を大きく左右するために重要な問題とされている。したがって、多くの方法がこれまでに考察されているが、ここでは、GA を用いて最適なパラメータを探索する。これまでも、直接 PID パラメータを探索する方法が提案されているが、その探索範囲が非常に広いため、多くの時間が必要となる。これに対し、文献 (10) では、一般化最小分散制御 (GMVC) 法との関連に基いて PID パラメータを調整する方法が提案されている。しかしながら、この手法によると、比例ゲインは GA によって選択された設計パラメータによって調整されるものの、積分パラメータ、微分パラメータは、それとは関係なく設計用モデルによって決定される。したがって、微分ゲイン、積分ゲインは設計パラメータによって調整されないため、制御性能は、初期の段階で調整される微分ゲイン、積分ゲインによって制約されてしまう。一方、GPC 法との関連に基いて PID パラメータを調整する方法⁽⁴⁾が考察され、その有効性が検証されている。この手法では、設計パラメータによって比例・微分・積分パラメータの全てが調整されるため、より良好な制御性能を得ることができる。しかしながら、設計パラメータの調整には試行錯誤を必要とする。

そこで、本論文では、GPC 則との関連に着目し、GPC 法の評価規範に含まれる設計パラメータを GA により探索する。そして、それを伸立ちとし PID パラメータを調整する。その際、設計用モデルと詳細モデルが必要となる。ここでは、設計用モデルとして 2 次の線形モデルを、詳細モデルとして GMDH ネットワークを用いる。

〈2・2〉 設計用モデル 設計用モデル (ノミナルモデル) として、次式で表されるような線形 2 次系を考える。

$$A(z^{-1})y(t) = z^{-k}B(z^{-1})u(t-1) \dots\dots\dots (1)$$

ただし、 $y(t)$ 、 $u(t)$ は、システム出力、制御入力を示している。また、 k はむだ時間を表している。さらに、多項式 $A(z^{-1})$ と $B(z^{-1})$ は次式で与えられる多項式である。

$$\left. \begin{aligned} A(z^{-1}) &= 1 + a_1z^{-1} + a_2z^{-2} \\ B(z^{-1}) &= b_0 + b_1z^{-1} + \dots + b_mz^{-m} \end{aligned} \right\} \dots\dots (2)$$

なお、(2) 式に含まれるパラメータは、制御対象のヒストリカルデータをもとに最小二乗法等を用いて、オフラインで決定する。

〈2・3〉 詳細モデル (1) 式的设计用モデルに対し、GA において適応度を計算する際、制御対象の動特性を詳細に記述した詳細モデルが必要である。ここでは、文献 (11) で提案されている改良型 GMDH ネットワークを用いて構成する。GMDH 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簡単であるという特徴をもっている。したがって、制御対象の動特性を把握することが容易であるという利点がある。以下に、簡単にその手順を示す。詳細に関しては、文献 (11) を参照されたい。

GMDH ネットワークは、図 1 に示すように N-Adaline 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。N-Adaline は 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 $z(t)$ とすると $z(t)$ は、次式のように表される。

$$z(t) = w_0x_1(t) + w_1x_1^2(t) + w_2x_1(t)x_2(t) + w_3x_2^2(t) + w_4x_2(t) + w_5 \dots\dots\dots (3)$$

ただし、 $w_i (i = 0, \dots, 5)$ は重み係数を示し、最小二乗法により決定される。また、図中の sq. は 2 乗演算を示している。

GMDH ネットワークの構造は以下の手順により決定される。

- step1) 入力変数の中から、2 入力の全ての組み合わせに対して N-Adaline を作る。
- step2) N-Adaline の出力 $z(t)$ が真のシステムの出力となるように、次式の評価規範に基いて重み係数 w_i を推定し、各 N-Adaline の寄与率を計算する。

(a) N-Adaline

(b) Schematic diagram of GMDH network

図 1 GMDH ネットワーク

Fig. 1. GMDH network.

$$J(t) = \frac{1}{2} e^2(t) \dots\dots\dots (4)$$

ただし、重み係数はバックプロパゲーション法に基いて学習する。例えば、二層目では、次式により重み係数を学習する。

$$\begin{aligned} \mathbf{W}_{1j}^{new} &= \mathbf{W}_{1j}^{old} \\ &+ \eta_1 e(t) \mathbf{W}_{2j}^T \frac{\partial \mathbf{X}_{2j}}{\partial z_j} \mathbf{X}_{1j} \dots\dots\dots (5) \end{aligned}$$

$$\mathbf{W}_{2j}^{new} = \mathbf{W}_{2j}^{old} + \eta_2 e(t) \mathbf{X}_{2j} \dots\dots\dots (6)$$

ただし、

$$e(t) = y(t) - \hat{y}(t) \dots\dots\dots (7)$$

$$\hat{y} = \mathbf{W}_{21}^T \cdot \mathbf{X}_{21} \dots\dots\dots (8)$$

$$z_1 = \mathbf{W}_{11}^T \cdot \mathbf{X}_{11} \dots\dots\dots (9)$$

$$z_2 = \mathbf{W}_{12}^T \cdot \mathbf{X}_{12} \dots\dots\dots (10)$$

$$\mathbf{W}_{21} = [w_{21}^0, w_{21}^1, w_{21}^2, w_{21}^3, w_{21}^4, w_{21}^5]^T \quad (11)$$

$$\mathbf{W}_{11} = [w_{11}^0, w_{11}^1, w_{11}^2, w_{11}^3, w_{11}^4, w_{11}^5]^T \quad (12)$$

$$\mathbf{W}_{12} = [w_{12}^0, w_{12}^1, w_{12}^2, w_{12}^3, w_{12}^4, w_{12}^5]^T \quad (13)$$

$$\begin{aligned} \mathbf{X}_{21} &= [1, z_1(t), z_1^2(t), \\ & z_1(t)z_2(t), z_2^2(t), z_2(t)]^T \cdot (14) \end{aligned}$$

$$\begin{aligned} \mathbf{X}_{11} &= [1, x_1(t), x_1^2(t), \\ & x_1(t)x_2(t), x_2^2(t), x_2(t)]^T \quad (15) \end{aligned}$$

$$\begin{aligned} \mathbf{X}_{12} &= [1, x_1(t), x_1^2(t), \\ & x_1(t)x_3(t), x_3^2(t), x_3(t)]^T \quad (16) \end{aligned}$$

である。ここでは、簡単のために2階層目における学習則を導いたが、階層を積み重ねていくと同時に学習則も同様の考え方に基いて導出する必要がある。

step3) 現階層の全てのN-Adalineについて、それらの寄与率があらかじめ定めたしきい値よりも小さいものは捨て、大きい寄与率をもつN-Adalineの出力を次の階層の入力変数として用いる。

step4) (現階層の最大寄与率) > (前階層の最大寄与率) ならば階層を一つ進め、[step1]に戻る。

step5) 階層を一つ戻し、その階層の最大寄与率をもつN-Adalineを出力層とし、順次それに連なる前層のN-Adalineのみを残し、他のN-Adalineを捨てることでネットワークの構成を終了する。

ただし、step2) で用いる寄与率は次式で定義される。

$$R^2 = \sum_{t=1}^N (\hat{y}(t) - \bar{\hat{y}})^2 / \sum_{t=1}^N (y(t) - \bar{y})^2 \dots\dots\dots (17)$$

上式において、 $\hat{y}(t)$ は出力の推定値、 $\bar{\hat{y}}$ は出力の推定値の平均値、 $\bar{y}$ は出力の平均値を表している。

〈2・4〉 PID 制御則 ここでは、次式で表されるI-PD制御則を考える。

$$\Delta u(t) = \frac{k_c T_s}{T_I} e(t) - k_c \left(\Delta + \frac{T_D}{T_s} \Delta^2 \right) y(t) \quad (18)$$

(18) 式で、 Δ は差分演算子 ($\Delta := 1 - z^{-1}$) を、 $e(t)$ は制御誤差を示しており、目標値 $w(t)$ に対して次式により定義される。

$$e(t) = w(t) - y(t) \dots\dots\dots (19)$$

また、(18) 式において、 k_c は比例ゲイン、 T_I は積分ゲイン、 T_D は微分ゲインを示し、 T_s はサンプリング間隔を表している。ここで、以下の考察を簡単にするために、次式で表される $L(z^{-1})$ を導入する。

$$\begin{aligned} L(z^{-1}) &:= k_c \left\{ \left(1 + \frac{T_s}{T_I} + \frac{T_D}{T_s} \right) \right. \\ & \quad \left. - \left(1 + \frac{2T_D}{T_s} \right) z^{-1} + \frac{T_D}{T_s} z^{-2} \right\} \dots\dots\dots (20) \end{aligned}$$

このとき、(18) 式は次式のように書き換えられる。

$$L(z^{-1})y(t) + \Delta u(t) - L(1)w(t) = 0 \dots\dots\dots (21)$$

(18) 式や (21) 式のPIDパラメータは制御性能に大きな影響を与える。したがって、その調整は重要な問題である。とくに、多くのプロセス系においては、むだ時間が大きく、また、その大きさを正確に把握することが難しいため、最適なPIDパラメータを得ることは非常に困難である。そこで、次節で説明するGPC法との関連に基いてPIDパラメータを調整する。

〈2・5〉 GPC 法 GPC 則は、次式の評価規範の最小化に基づいて導かれる⁽¹⁾。

$$\begin{aligned} J = E \left[\sum_{i=1}^N \{y(t+i) - w(t+i)\}^2 \right. \\ \left. + \lambda \sum_{j=1}^N \{\Delta u(t+j-1)\}^2 \right] \dots\dots\dots (22) \end{aligned}$$

ここで、 $E[\cdot]$ は期待値を示している。また、 λ は操作量の重み係数を示し、 N は予測区間を示している。このとき、(22) 式の評価規範を最小化する制御入力は、次式で与えられる⁽¹⁾。

$$\begin{aligned} \sum_{i=1}^N p_i F_i(z^{-1}) y(t) \\ + \left\{ 1 + z^{-1} \sum_{i=1}^N p_i S_i(z^{-1}) \right\} \Delta u(t) \\ - \sum_{i=1}^N p_i w(t) = 0 \dots\dots\dots (23) \end{aligned}$$

ここで、 $F_i(z^{-1})$ は次式の Diophantine 方程式を解くことによって得られる。

$$1 = \Delta A(z^{-1})E_i(z^{-1}) + z^{-i}F_i(z^{-1}) \dots\dots\dots (24)$$

ただし、 $E_i(z^{-1})$ と $F_i(z^{-1})$ は次式であらわされる多項式である。

$$E_i(z^{-1}) = 1 + e_1 z^{-1} + \dots + e_{i-1} z^{-(i-1)} \dots (25)$$

$$F_i(z^{-1}) = f_{i,0} + f_{i,1} z^{-1} + f_{i,2} z^{-2} \dots\dots\dots (26)$$

また、 $S_i(z^{-1})$ は次式を解くことで得られる多項式である。

$$E_i(z^{-1})B(z^{-1}) = R_i(z^{-1}) + z^{-i}S_i(z^{-1}) \dots (27)$$

ただし、 $S_i(z^{-1})$ と $R_i(z^{-1})$ は次式で表される多項式である。

$$R_i(z^{-1}) = r_0 + r_1 z^{-1} + \dots + r_{i-1} z^{-(i-1)} \dots (28)$$

$$S_i(z^{-1}) = s_{i,0} + s_{i,1} z^{-1} + \dots + s_{i,m-1} z^{m-1} \dots (29)$$

さらに、次式により p_i が計算される。

$$[p_1, p_2, \dots, p_N] := \underbrace{[1, 0, \dots, 0]}_{N-1} (\mathbf{R}^T \mathbf{R} + \Lambda)^{-1} \mathbf{R} \dots\dots\dots (30)$$

$$\mathbf{R} := \begin{bmatrix} r_0 & & & \mathbf{0} \\ r_1 & r_0 & & \\ \vdots & \vdots & \ddots & \\ r_{N-1} & r_{N-2} & \dots & r_0 \end{bmatrix} \dots\dots\dots (31)$$

また、 Λ は次式で表される。

$$\Lambda = \lambda \mathbf{I} \dots\dots\dots (32)$$

なお、(23) 式の詳細な導出については、文献 (1) を参照されたい。

〈2・6〉 PID パラメータの調整法 ここでは、PID パラメータを GPC 則との関係に基いて調整する方法について考察する。

まず、(23) 式において定常特性を重視して、(23) 式の左辺第二項の定数多項式を定常ゲインに置き換えた次式を考える。

$$\sum_{i=1}^N p_i F_i(z^{-1}) y(t) + \left\{ 1 + \sum_{i=1}^N p_i S_i(1) \right\} \Delta u(t) - \sum_{i=1}^N p_i w(t) = 0 \dots\dots\dots (33)$$

ただし、(33) 式の左辺第 2 項を次式のように定義する。

$$X = 1 + \sum_{i=1}^N p_i S_i(1) \dots\dots\dots (34)$$

図 2 (35) に対する制御応答 ($k=3, \lambda=1.0$)

Fig. 2. Control result for (35), where $k=3$ and $\lambda=1.0$.

図 3 (35) に対する制御応答 ($k=3, \lambda=5.0$)

Fig. 3. Control result for (35), where $k=3$ and $\lambda=5.0$.

(33) 式において係数多項式を定常ゲインに置き換えた場合、制御性能が変化すると考えられる。そこで、このことについて以下の数値例を用いて考察する。

まず、制御対象が次式で与えられるとする。

$$y(t) = 0.4y(t-1) + 0.1y(t-2) + 0.3u(t-k-1) + 0.1u(t-k-2) \dots\dots (35)$$

ただし、 k はむだ時間を表している。この制御対象に対して、予測区間 $N=8$ の GPC 法を適用する。図 2 は、 $k=3$ の場合の応答 ($\lambda=2.0$) を、図 3 は、 $k=3$ の場合の応答 ($\lambda=5.0$) を示している。図 2、図 3 において破線が GPC 法、実線が $\Delta u(t)$ の係数多項式を定常ゲインに置き換えた応答を示している。以上の結果から、係数多項式を定常ゲインに置き換えた場合、図 2 では、応答の立ち上がりは劣化しているが、オーバーシュート量が抑えられ、整定時間も短くなっていることがわかる。GPC でこのオーバーシュートを抑えるためには、図 3 のように λ を大きくする必要がある。図 3 から、 λ を大きくすると、GPC による応答と定常ゲインに置き換えた応答には、ほとんど差がないことがわかる。もっとも、プロセス制御系においては、ひとつのループにオーバーシュートが生じると、下位ループなど他のループの応答に大きな影響を与え、結果として、制御系全体の安定性を損なうことになる。GPC がむだ時間システムに対して広く用いられていることを考えると、図 2 に示すようなむだ時間系に対して、 $\Delta u(t)$ の係数多項式を定

常ゲインに置き換えたことでオーバーシュートを抑えられるということは、プロセス制御系においては好都合なことであると考えられる。ところが、図2と図3から明らかなように、 λ の値によって即応性が変化するため、後に述べるGAの適応度関数において、再度過渡特性を考慮して設計パラメータを調整する。

以下のもとの、次式の関係が成立するとき、(21)式と(33)式は等価となる。

$$L(z^{-1}) = \frac{1}{X} \sum_{i=1}^N p_i F_i(z^{-1}) \\ = \tilde{f}_0 + \tilde{f}_1 z^{-1} + \tilde{f}_2 z^{-2} \dots \dots \dots (36)$$

ただし、(36)式の $\tilde{f}_i$ は、次式により計算される。

$$\tilde{f}_i = \frac{1}{X} \sum_{k=1}^N p_k f_{k,i} \dots \dots \dots (37)$$

したがって、(20)式と(36)式より次式のような関係を得ることができる。

$$\left. \begin{aligned} k_c &= -(\tilde{f}_1 + 2\tilde{f}_2) \\ T_I &= -\frac{\tilde{f}_1 + 2\tilde{f}_2}{\tilde{f}_0 + \tilde{f}_1 + 2\tilde{f}_2} T_s \\ T_D &= -\frac{\tilde{f}_2}{\tilde{f}_1 + 2\tilde{f}_2} T_s \end{aligned} \right\} \dots \dots \dots (38)$$

(30)式により λ が変わると p_i が変わり、これにより(34)式が変化する。よって、(38)式のPIDパラメータは、 λ によってすべて調整される。文献10)のGMVC法を用いた手法では、設計パラメータの λ が変更されても、積分・微分パラメータは変化せず、比例ゲインだけが逐次計算される。つまり、GPC法を用いることにより、GMVC法では考慮されない位相保証を考慮することができ、この点においてGPC法を用いた方がよりよい制御結果を得られると考えられる。

〈2・7〉GAによる探索 先にも述べたように、GAを用いてGPC則に含まれる設計パラメータ λ を探索する。GAを用いた探索法は数多く存在するが、ここでは、より詳細なPIDパラメータの値を得るために、実数値GAを用いて最適な設計パラメータを探索する。以下にその一例を示す。

[GAによる探索アルゴリズム]

- step1) 初期個体: λ の探索範囲を決定し、初期個体をランダムに n 個体発生させる。
- step2) 世代数: g を設定する。
- step3) 適応度: 適応度関数として、2乗誤差積分ISE (Integral of Squared Error) 規範に基づいた次式を用いる。

$$f(p) = \frac{1}{1 + \sum_{t=1}^M \{y_m(t) - y_p(t)\}^2} \dots \dots \dots (39)$$

ただし、 $y_m(t)$ は参照モデルの出力を、 $y_p(t)$ は詳細

モデルの出力を、 p は個体番号を示している。

step4) 選択: エリート戦略に基づいて行う。すなわち、最良の $\alpha\%$ の個体を次世代に残す。

step5) 交叉: $\alpha\%$ の中から、残りの $(100 - \alpha)\%$ の個体を交叉により生成する。具体的には、選ばれた2つの個体“a”と“b”を、それぞれ $f(a)$ 、 $f(b)$ とすると、新しい個体 c を次式により生成する。

$$c = \frac{I(\max\{f(a), f(b)\}) + (a + b)/2}{2} \dots (40)$$

ここで、 $I(x)$ は適応度 x を与える個体(a か b のいずれか)を表している。

step6) 突然変異

ランダムに $\beta\%$ の個体を選択し、その値を探索範囲の最大値から引いた値と置きかえる。

3. 数 値 例

ここでは、提案手法の有効性を検証するために数値例に適用する。制御対象が次式で与えられるとする。

$$\begin{aligned} y(t) &= 0.4y(t-1) - 0.09y(t-2) \\ &+ 0.3u(t-3) - 0.1u(t-4) \\ &+ 0.05y(t-3)u(t-3) \\ &+ 0.03y(t-4)u(t-4) + \xi(t) \dots \dots \dots (41) \end{aligned}$$

ただし、 $\xi(t)$ はモデル化誤差等を表すガウス性白色雑音で、平均0、分散0.01で与えた。

まず、提案手法と比較するために、PID制御法を適用する。PIDパラメータは、事前情報に基づいて、CHR法⁽¹²⁾より $k_c = 1.061$ 、 $T_i = 3.529$ 、 $T_d = 0.985$ のように得られた。PID制御法の制御結果を図4に示す。図4より、制御対象

図4 PID制御法による制御結果

Fig.4. The control result by PID scheme using CHR method.

の非線形特性のために、激しく振動していることがわかる。
次に、GMVC 法との関連に基づいて PID パラメータを調整する方法を適用する⁽¹⁰⁾。まず、設計用モデルを設計する。設計モデルは (1) 式を用い、そのパラメータは、事前情報により最小二乗法を用いて次式のように得られた。

$$\begin{aligned} \hat{y}(t) = & 0.685\hat{y}(t-1) + 0.109\hat{y}(t-2) \\ & + 0.004u(t-1) + 0.114u(t-2) \\ & + 0.346u(t-3) - 0.176u(t-4) \dots\dots\dots (42) \end{aligned}$$

図 5 に、GMVC 法との関連に基づいた方法の制御結果を示す。GMVC 系は、次式で表される評価規範の最小化に基づいて設計される。

$$J = [T(z^{-1})\hat{y}(t+1) + \omega\Delta u(t) - T(1)w(t)]^2 \quad (43)$$

ただし、 $T(z^{-1})$ は設計多項式で、次式のように与えた。

$$T(z^{-1}) = 1 - 1.8z^{-1} + 0.81z^{-2} \dots\dots\dots (44)$$

図 5 より明らかなように、オーバーシュートがでている。この手法では、事前情報により設計される設計用モデルに基づいて T_I , T_D が決定され、GA によって調整されるのは k_c のみである。したがって、位相補償が良好に機能したため、むだ時間のような位相遅れに起因して、オーバーシュートが生じていると考えられる。なお、GA により選択された設計パラメータ ω は 1.196 で、この値に基づいて計算される PID パラメータは、 $k_c = 0.399$, $T_I = 0.531$, $T_D = 1.341$ となった。

最後に、提案手法を適用する。まず、設計用モデルと詳細モデルを設計する。設計モデルは、(42) 式を用いた。詳細モデルは、GNDH ネットワークにより以下のように設計した。ネットワークの概要図を図 6 に、各 N-Adaline のパ

図 5 GMVC 法との関連に基づいた方法による制御結果

Fig. 5. The control result by PID scheme based GMVC.

図 6 ネットワークの概要図

Fig. 6. The diagram of the GMDH network.

表 1 各 N-Adaline のパラメータ

Table 1. The parameter of each N-Adaline.

	w_0	w_1	w_2	w_3	w_4	w_5
θ_1	-0.072	0.366	-0.001	-0.010	0.036	0.160
θ_2	0.130	0.307	-0.005	0.045	-0.008	0.027
θ_3	0.013	1.096	0.158	-0.669	0.543	-0.154

図 7 詳細モデルのモデル化誤差

Fig. 7. Modeling error of the detailed model for the same input.

ラメータの値は表 1 のように得られた。
真のシステムと詳細モデルの比較を、図 7 に示す。図 7 中の 1 段目は真のシステムの出力、2 段目は詳細モデルの出力、3 段目はこれらの誤差を示している。図 7 より、GMDH ネットワークによる詳細モデルが、十分に精度良く真のシステムの動特性を反映していることが分かる。以上の設計モデルと詳細モデルを用い、(40) 式の制御対象に提案手法を適用する。本数値例では、 λ の探索範囲を 0~10、個体数 g を 50、世代数 n を 100、次世代に残す個体の割合 α を 20%、突然変異率 β を 1% として探索を行った。図 8 に、提案手法による制御結果を示す。図から明らかなように、応答が振

図8 提案手法による制御結果

Fig. 8. The control result by the proposed method.

動することなく速やかに目標値に追従していることが分かる。本手法では、調整された設計パラメータによってPIDパラメータ全てを調整するため、ゲイン補償と位相補償が効果的に働き、良好な制御結果を得ることができた。なお、GAにより選択された設計パラメータ λ は0.126で、この値に基づいて計算されるPIDパラメータは、 $k_c = 0.195$, $T_I = 0.851$, $T_D = 0.212$ となった。

以上の結果から、本手法の非線形系に対する有効性を検証することができた。

4. 結 言

本論文では、PIDパラメータを調整する方法として、GPC則に含まれる設計パラメータを遺伝的アルゴリズムにより探索し、さらに、GPC則を仲立ちとし、これとの関連に基づいてPIDパラメータを調整する方法を提案した。また、本手法を数値例に適用し、その有効性を検証した。本論文で報告したように、遺伝的アルゴリズムを用いることで、GPC則に含まれるパラメータの決定のための試行錯誤が必要でなくなり、また、最適なパラメータが得られることで、結果として非線形系の制御対象に対しても最適なPIDパラメータを決定することが可能となった。今後は、本手法の実システムへの適用について考察を重ねる予定である。

(平成13年12月3日受付, 平成14年9月10日再受付)

文 献

- (1) D.W. Clarke, C. Mohtadi and P.S. Tuffs: "Generalized Predictive Control", *Automatica*, **23**, 2, pp.137~160 (1987)
- (2) E.F. Camacho and C. Bordons: *Model Predictive Control in the Process Industry*, Advances in Industrial Control, Springer, London (1995)
- (3) N. Suda: *PID Control*, Asakura Publishing (1992) (in Japanese)
須田信英: *PID制御*, 朝倉書店 (1992)

- (4) M. Asano and Toru Yamamoto: "A Design of Self-Tuning Predictive PID Controllers", *IEICE Trans. on Fundamentals*, E84A-7, pp.1779-1783 (xxxx)
- (5) A.G. Ivakhnenko: "Polynomial theory of complex systems", *IEEE Trans. Syst., Man, and Cybernetics*, **12**, pp.364-378 (1971)
- (6) S. Shrier, R.L. Barron, and L.O. Gilstrap: "Polynomial and neural networks: analogies and engineering applications", *Proc. IEEE International Conference on Neural Networks*, **2**, pp.431-439 (1987)
- (7) S.J. Farlow: *Self-Organizing Methods in Modeling*, Marcel Dekker Inc., (1984)
- (8) R. Hecht-Nielsen: *Neurocomputing*, Addison-Wesley Publishing Company, (1990)
- (9) D.E. Goldberg: *Genetic Algorithm in Search, Optimization & Machine Learning*, Addison-Wesley, (1992)
- (10) T. Yamamoto, Y. Mitsukura, and M. Kaneda: "A Design of PID Controllers by Using a Genetic Algorithm", *Trans. of SICE*, Vol.35, No.4, pp.531-537 (1999-4) (in Japanese)
山本・満倉・兼田: 「遺伝的アルゴリズムを用いたPID制御器の設計」, 計測自動制御学会論文集, **35**, 4, pp.531-537 (1999-4)
- (11) A. Sakaguchi, and Toru Yamamoto: "A Design of Generalized Minimum Variance Controllers Using a GMDH Network for Nonlinear Systems", *Trans. IEICE*, Vol.E84-A, No.11, pp.2901-2907 (2001)
- (12) K.L. Chien, J.A. Hrones, and J.B. Reswick: "On the Automatic Control of Generalized Passive Systems", *Trans. ASME*, **74**, pp.175-185 (1952)

坂口 彰 浩 (正員) 1976年3月20日生。2000年3月岡山県立大学大学院情報系工学研究科博士前期課程(電子情報通信工学専攻)修了。同年4月佐世保工業高等専門学校助手となり現在に至る。インテリジェント手法を用いたプロセス制御系の応用に関する研究に従事。計測自動制御学会、システム制御情報学会などの会員。

片山 優 (学生員) 1977年3月18日生。2001年3月岡山県立大学大学院情報系工学研究科博士前期課程(電子情報通信工学専攻)修了。同年4月広島大学大学院教育学研究科博士課程後期文化教育開発専攻(技術・情報教育学専修)に入学し現在に至る。計測自動制御学会の会員。

山本 透 (正員) 1961年8月29日生。1987年3月徳島大学大学院工学研究科修士課程(情報工学専攻)修了。同年4月高松工業高等専門学校助手, 1988年4月同講師, 1992年4月大阪大学基礎工学部助手, 1994年4月岡山県立大学情報工学部助教授, 1999年10月広島大学学校教育学部助教授, 2000年4月同教育学部助教授, 2001年4月同大学院教育学研究科技術・情報教育学講座助教授となり現在に至る。適応制御とくにセルフチューニングコントローラの設計と応用, ならびにPID制御系の高度化と知能化に関する研究に従事。工学博士。IEEE, 計測自動制御学会, システム制御情報学会, 電子情報通信学会などの会員。