制御知識を取り入れたParticle Swarm Optimizationによる PID制御系設計

M2007MM001 安藤貴彦
指導教員：高見勲

1 はじめに

PID制御は古典制御の代表で長い歴史を持ち，現在でも広範囲の制御対象に適用されている．その主な理由として，制御系が単純であること，比例，積分，微分の機能の物理的な意味が分かっており，現場調整が容易であること，他の複雑な制御方式を適用しなくても必要な制御性能を得ることが出来る場合が多いといった点が挙げられる．よって，数式モデルが困難なプロセス系から，設計，実装が容易で数式モデルが得やすいサーボ系までPID制御は適用され，工作機械やロボット [1]，自動車 [2] など多くの事例がある．

PID制御の比例，積分，微分の各ゲイン（これをPIDパラメータと呼ぶことにする）は制御系の性能を大きく左右するため，その適切な設定が重要な課題である．また，システムの設計・解析・制御に対する要求が高度化し，効率化・高信頼化・高機能化といったことが求められており，実用的な最適化に対するニーズが高まってきている．したがって本研究では，PIDパラメータを設定する手段として，メタヒューリスティクスの一種のParticle Swarm Optimization（以下，PSO）[3][4][5] を適用する．従来のパラメータの最適化では評価関数が微分可能，連続といった制約を前提としたが，本手法のPSOはオーバーシュート量，整定時間を評価関数として扱うことができるため，応答を直接評価できる．

制御系に対してのPSOの適用は幾つか報告されており，PID制御 [6][7]，H_∞ 制御 [8] といったものがある．[6][7] では基本的なPSOを用いてPID制御設計を行い，PSOのアルゴリズム自体の改良は行っていない．本研究では，制御知識をPSOに取り入れたアルゴリズムを提案する．これは，制御系が不安定であったり，応答が振動的であったりするようなパラメータ領域を探索していても意味が無いので，制御知識の観点から望ましい方向へ探索を誘導してやり，早い段階で最良解に近い領域に到達させるようにする．さらに，PIDパラメータには非負という領域制約があるので，上下限制約を付け，PSOの探索において局所解に陥らないように，定期的に解を分散させる分散アルゴリズムもPSOに追加する．また，本論文では，複数回試行した中で評価値が最小になった解を最良解とする．

2 Particle Swarm Optimization

PSOは鳥の群れ，魚の群れ，昆虫の群れなどといった集団での探索行動に基づいた多点型最適化手法であり，Particle(微粒子) と呼ばれるランダムに配置された探索点がSwarm(群れ) を構成し，Particleの独自情報と，群全体の共通情報を組み合わせて解空間を良い方へと移動する探索アルゴリズムである．また，この手法は非線形最適問題を説く有力な手法として知られている．以下にPSOの概要をまとめる [4].

PSOは次のようなベクトルから構成されている．n 次元空間における particle(探索点) は，以下のベクトルを持つ．

位置ベクトル $\boldsymbol{x}_i^k := (x_{i,1}^k, x_{i,2}^k, \cdots, x_{i,j}^k, \cdots x_{i,n}^k)^T \in \mathbb{R}^n$

移動ベクトル $\boldsymbol{v}_i^k := (v_{i,1}^k, v_{i,2}^k, \cdots, v_{i,j}^k, \cdots v_{i,n}^k)^T \in \mathbb{R}^n$

k は反復回数，i はparticleの番号，j はベクトルの要素番号を表す．

さらに，各particleはこれまでの探索で発見した最良ベクトル $\boldsymbol{pbest}_i^k := (pbest_{i,1}^k, pbest_{i,2}^k, \cdots, pbest_{i,j}^k, \cdots pbest_{i,n}^k)^T \in \mathbb{R}^n$ と $\boldsymbol{pbest}_i^k$ の評価関数 $f(\boldsymbol{pbest}_i^k)$ を記憶する．群としては，すべてのparticleがこれまでの探索で発見した最良ベクトル $\boldsymbol{gbest}^k := (gbest_1^k, gbest_2^k, \cdots, gbest_j^k, \cdots gbest_n^k)^T \in \mathbb{R}^n$ と $\boldsymbol{gbest}^k$ の評価関数 $f(\boldsymbol{gbest}^k)$ を記憶する．

これらのベクトルを用いて，以下のような式が与えられ，この式により $\boldsymbol{x}_i, \boldsymbol{v}_i$ を更新し，particleを移動させる．

$$v_{i,j}^{k+1} = w \cdot v_{i,j}^k + c_1 \cdot \mathrm{rand}_1()_{i,j} \cdot (pbest_{i,j}^k - x_{i,j}^k) + c_2 \cdot \mathrm{rand}_2()_{i,j} \cdot (gbest_j^k - x_{i,j}^k) \quad (1)$$

$$x_{i,j}^{k+1} = x_{i,j}^k + v_{i,j}^{k+1} \quad (2)$$

ここで，w，c_1，c_2 は各項に対する重み係数，$\mathrm{rand}_1()_{i,j}$，$\mathrm{rand}_2()_{i,j}$ は0から1の間に分布する一様乱数である．

この更新を反復回数 T_{max} 回行い，最適解を求める．

3 PIDパラメータチューニング

PIDパラメータ K_P，K_I，K_D をPSOの位置ベクトルの要素とし，$\boldsymbol{x}_i := (K_P, K_I, K_D) \in \mathbb{R}^3$ と表す．

評価関数 $f(\boldsymbol{x}_i^k)$ は，ステップ応答において出力応答は目標値応答との誤差が少なく，オーバーシュートが小さく，整定時間が短いほど良いので，誤差面積とオーバーシュートと整定時間の和とする．よって，

$$f(\boldsymbol{x}_i^k) = a_{iae} \cdot \int_0^\infty |e(t)|\, dt + a_{mo} \cdot (y_{max} - y_\infty) + a_{st} \cdot T_s \quad (3)$$

と与える．$e(t)$ は偏差，y_{max} は応答 $y(t)$ の最大値，y_∞ は定常値，T_s は整定時間，また a_{iae}，a_{mo}，a_{st} は各項の重み係数である．この評価関数に従って，応答が最良な応答になるように $\boldsymbol{x}_i$ を移動させていく．

4 制御知識を取り入れたPSO

PSOによりPIDパラメータは探索されるが，応答が発散したり，振動的であったり，定常偏差があったりと制御性が悪い領域を探索しても意味が無いので，そのような領域から早く移動させる必要がある．そこで，パラメー

タをチューニングする際，応答に応じてゲインを大きくしたり，小さくしたりという制御知識の考え方を PSO に取り入れる．これにより，制御性の悪い領域から早く抜け出し，制御性の良い領域を探索させるようにする．

4.1 不安定を考慮した移動ベクトル

particle が不安定領域にある場合，その particle を安定領域を速く移動させ，安定領域内での探索を早く出来るようにする．不安定領域とは応答が発散してしまう PID パラメータの領域を意味する．このようにするために移動ベクトルである式 (1) を次のように改良する．

$$\begin{aligned} v_{i,j}^{k+1} &= w \cdot v_{i,j}^{k} + c_1 \cdot \mathrm{rand}_1()_i \cdot (pbest_{i,j}^{k} - x_{i,j}^{k}) \\ &\quad + c_2 \cdot \mathrm{rand}_2()_i \cdot (gbest_j^{k} - x_{ij}^{k}) \\ &\quad - c_3 \cdot \gamma^k \cdot \mathrm{rand}_3()_i \cdot \frac{1}{N_{us}} \sum_{z_{k\ i} \in US} (z_{k\ i,j} - x_{i,j}^{k}) \end{aligned} \tag{4}$$

γ は 0 ~ 1 までの定数である．$\boldsymbol{z}_{k\ i}$ は繰り返し回数 k までに存在した $\boldsymbol{x}_i$ で，不安定な領域を US とする．N_{us} は繰り返し回数 k までに不安定な領域に存在した particle の数である．右辺第 4 項は不安定な領域に $\boldsymbol{x}_i^k$ が存在したら，それまでの探索で不安定になった $\boldsymbol{z}_{k\ i}$ へ向かうベクトルの逆方向へ向かわせるための項である (図 1).

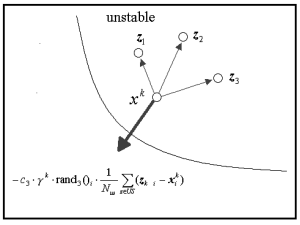


図 1 不安定を考慮した移動ベクトル

4.2 PID パラメータチューニングの制御知識を取り入れた PSO

PID パラメータをチューニングするための制御知識は次のようなものが挙げられる．

応答が振動的である K_P の値を小さくする．

$$K_{P_{kh}} = (1 - \beta_{K_P}) \cdot K_P \quad \text{if } y_{max} > 1.2 \cdot y_\infty \tag{5}$$

オフセットが残る K_I の値を大きくする．

$$K_{I_{kh}} = (1 + \beta_{K_I}) \cdot K_I \quad \text{if } r(t) \neq y_\infty \tag{6}$$

$K_{P_{kh}}$，$K_{I_{kh}}$ は制御知識により更新した K_P，K_I ゲインであり，β_{K_P},β_{K_I} は 0 ~ 1 までの定数である．この制御知識を $\boldsymbol{x}_i$ の更新時に追加させる．

5 制御知識を取り入れた PSO の解析

制御知識を取り入れた PSO の性能として，応答が安定している領域に速く移動するということが挙げられる．これを particle の反復回数ごとの位置と $f(\boldsymbol{gbest}^k)$ の変動を解析し，標準的な PSO との性能比較を行う．

5.1 制御対象

制御対象としてフレキシブルアームを用いる．この制御対象はアームの厚みが薄いので剛性が低く，応答が振動的になりアームの先端が振動してしまう．その振動を考慮して先端角度を制御する．この制御対象のモデリングの概略図を図 2 に示す．図 2 で θ がアームの根元の角度，α がアームの歪み角である．

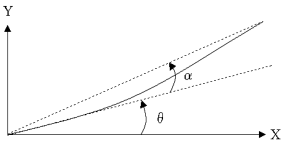
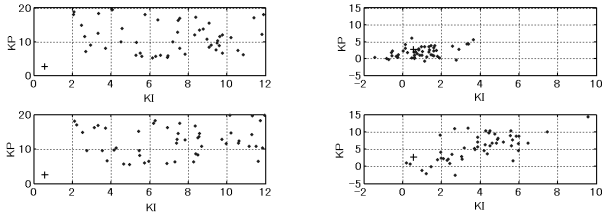


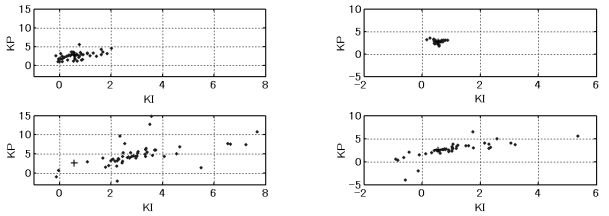
図 2 フレキシブルアーム概略図

5.2 解析結果

図 3 に反復回数ごとの各 particle の K_P と K_I の位置を示す．上段は制御知識を取り入れた PSO で行ったものであり，下段は一般的な PSO で行ったものである．図中の「●」は各 particle の位置,「＋」は最良解の位置を表し，particle 数 $m = 50$ とする．



(a) 初期値 (b) 反復回数:5

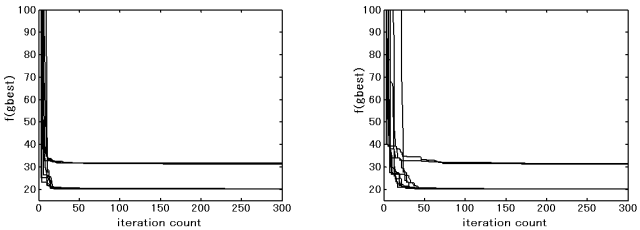


(c) 反復回数:10 (d) 反復回数:20

図 3 K_P, K_I の変動

図で分かるように制御知識を取り入れることにより，探索しても意味のない particle(振動的な応答を与える particle) は早い段階でその要素の値が下げられ，応答が穏やかな限定した領域に移動されている．

次に繰り返し回数 300 回においての $f(\boldsymbol{gbest}^k)$ の変動を図 4 に示す．この図は PSO を 10 回試行したものである．



(a) 制御知識付 PSO (b) 標準的な PSO

図 4 $f(\boldsymbol{gbest}^k)$ の変動

このように，制御知識を取り入れることにより，収束する速度は向上したといえる．これは前にも述べたように，制御知識により制御性能の悪い応答となる PID パラメータの領域からの脱出が速く，解に近い領域を集中的に探索しているためであると考えられる．しかし，図を見て分かるように 2 箇所に収束してしまっている．つまり，局所解に陥っている．

6 上下限制約と大域的解の探索

6.1 反射による上下限制約

標準的な PSO では，$\boldsymbol{x}_i$ と $\boldsymbol{v}_i$ は制約がないため，探索領域は無限になり，探索領域の制約を考慮していない．領域を外れると評価関数として重いペナルティを科してその領域の探索を極力避けるという考え方は容易に考えられるが，particle は結局領域外も探索してしまうことになるので，効率が悪い．また，村田らは，$x_j = f_j(y_j) = \frac{q_j + p_j \exp(y_j)}{1+\exp(y_j)}, q_j \le x_j \le p_j$ とおき，変数 y に対する無制約問題とする研究も報告されている [10].

本研究では，変数 (particle) の上限下限の制約に反射の考え方を用いる．この考え方を取り入れた根拠は，生物は障害物をよける際，一般的に反射に近い動きを見せること，また，反射量をペナルティーとして考えることが出来るということである．これは，領域を外れた分のベクトルの長さ分を反射させることにより，境界から遠い点に行ってしまうと反射量は多く，境界に近いと反射量は少なくなるというものである．

反射による上下限制約のアルゴリズムは次のようになる．ここで，上下限領域を $low_j \le x_{i,j}^k \le high_j$ と表す．

step1 最初に境界から外れる変数を見つける．

境界を外れた変数の有無をまず判別し，境界を外れた変数があれば反射のアルゴリズムに入る．

1. $x_{i,j}^k$ から境界までの距離 $dis\_in_{i,j}^k$ を求める．

$$dis\_in_{i,j}^k = \begin{cases} |x_{i,j}^k - low_j|, \text{ if } x_{i,j}^{k+1} < low_j \\ |x_{i,j}^k - high_j|, \text{ if } x_{i,j}^{k+1} > high_j \end{cases} \tag{7}$$

2. $x_{i,j}^k$ から $x_{i,j}^{k+1}$ の距離に対しての $dis\_in_{i,j}^k$ の割合 $rate\_in_{i,j}^k$ を求める．

$$rate\_in_{i,j}^k = \begin{cases} \dfrac{dis\_in_{i,j}^k}{|x_{i,j}^{k+1} - x_{i,j}^k|}, \\ \quad \text{if } (x_{i,j}^{k+1} < low_j) \text{ or } (x_{i,j}^{k+1} > high_j) \\ 1, \text{ if } (x_{i,j}^{k+1} > low_j) \text{ and } (x_{i,j}^{k+1} < high_j) \end{cases} \tag{8}$$

3. $x_{i,j}^k$ の要素の中で最小の $rate\_in_{i,j}^k$ を求める．これを $boundary_i^k$ とする．

$$boundary_i^k = \min_j rate\_in_{i,j}^k \tag{9}$$

$boundary_i^k < 0$ となる変数は領域から外れる．また，$\min_j rate\_in_{i,j}^k$ となる変数の要素が最初に領域から外れることになる．よって，この要素の正負を反転することにより反射させる．

step2 境界点を求める．

$$x\_boundary_{i,j}^k = x_{i,j}^k + (x_{i,j}^{k+1} - x_{i,j}^k) \cdot boundary_i^k \tag{10}$$

$$x_{i,j}^k = x\_boundary_{i,j}^k \tag{11}$$

step3 反射による $x_{i,j}^{k+1}$，$v_{i,j}^{k+1}$ の更新

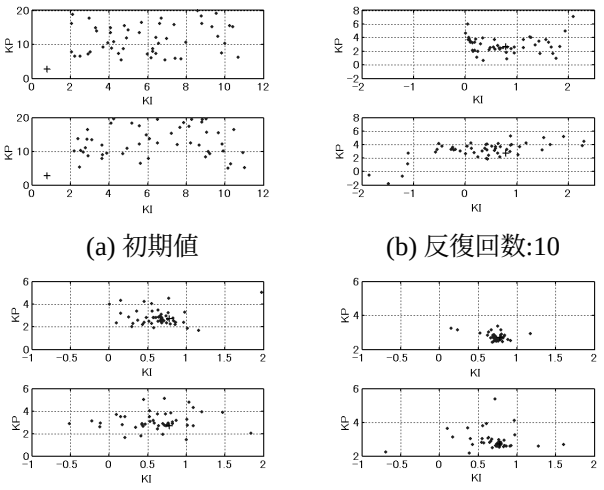
$$x_{i,j}^{k+1} = \begin{cases} low_j + |x_{i,j}^{k+1} - low_j|, \\ \quad \text{if } (boundary_i^k = rate\_in_{i,j}^k) \text{ and } (x_{i,j}^{k+1} < low_j) \\ high_j - |x_{i,j}^{k+1} - high_j|, \\ \quad \text{if } (boundary_i^k = rate\_in_{i,j}^k) \text{ and } (x_{i,j}^{k+1} > high_j) \end{cases} \tag{12}$$

$$v_{i,j}^{k+1} = -v_{i,j}^{k+1} \quad \text{if case of (12)} \tag{13}$$

step4 終了判定

反射を行い領域内に $x_{i,j}^{k+1}$ が移動したら終了．そうでなかったら step1 に戻る．

この反射を導入した PSO により $K_P, K_I, K_D \ge 0$ という領域制約与えた上で，particle の変動を図 3 と同様な形で図 5 に示す．上段は制御知識を取り入れたに反射を導入した PSO で行ったものであり，下段は制御知識を取り入れた PSO で行ったものである．



(a) 初期値　(b) 反復回数:10

(c) 反復回数:20　(d) 反復回数:30

図 5 K_P, K_I の変動 (反射導入)

このように，非負の領域で探索を行い，最良解付近への収束が早い..

6.2 分散アルゴリズム

5 章の解析で示したように局所解が現れてしまった．この局所解から抜け出し，最適解を見つけ出すためには収束してしまった particle を再び飛び回るようにしてやらなければならない．

しかし，PSO では particle が収束してしまうということは $\boldsymbol{v}$ がほぼ 0 になってしまうことである．よって，particle の多様性の回復は困難である．筆者ら [9] は GA で多様性を回復させるため，ウイルス感染という考え方を取り入れ，定期的に適応度の悪い個体の遺伝子をウイルスにより変化させ，新しい個体を生成し，個体を分散させた．このようにして，定期的に個体が散らばらせ，多様性を維持した．この考え方を PSO に適用し，収束していた particle を大域的に分散させ，多様性を回復させる．次のような手順で PSO に適用する．

step1 m 個の particle を評価値 $f(\boldsymbol{x}_i)$ の良い d 個と，それ以外の $m-d$ 個に分け、それぞれで群を生成する．particle 数 d 個の群を群 1，particle 数 $m-d$ 個の群を群 2 とする．

step2 群 1 は step1 終了時の位置ベクトル $\boldsymbol{x}_i$，移動ベクトル $\boldsymbol{v}_i$，各 particle はこれまでの探索で発見した最良ベクトル $\boldsymbol{pbest}_i$，すべての particle がこれまでの探索で発見した最良ベクトル $\boldsymbol{gbest}$ を保持し，群 2 はウイルス $\boldsymbol{xvirus}_i, \boldsymbol{vvirus}_i$ を生成し，感染率 r の確率で，$\boldsymbol{x}_i, \boldsymbol{v}_i$ の要素と $\boldsymbol{xvirus}_i, \boldsymbol{vvirus}_i$ の要素を入れ換える．

そして，新たに $\boldsymbol{pbest}_i$，群 2 でのすべての particle の最良解 $\boldsymbol{gbest2}$ を決める．

step3 群 1，2 でそれぞれ PSO を反復回数 $N(N < T_{max})$ 回行う．その際，$f(\boldsymbol{gbest2}) < f(\boldsymbol{gbest})$ ならば，$\boldsymbol{gbest}$ を $\boldsymbol{gbest2}$ に更新する．

step4 群 1，2 をまとめ，step1 へ戻る．

このアルゴリズムを加えると $f(\boldsymbol{gbest})$ の変動は図 6 となる．$N = 40$ とする．

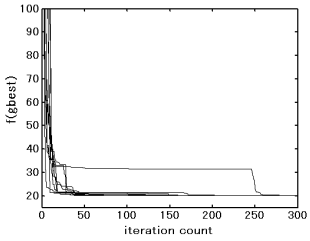
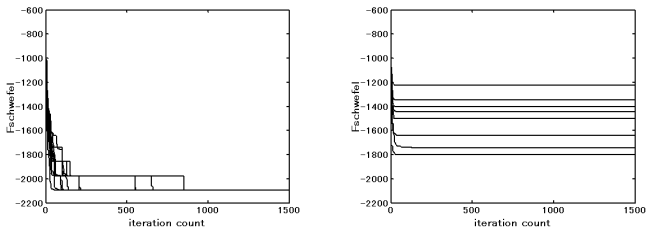


図 6 $f(\boldsymbol{gbest}^k)$ の変動 (分散アルゴリズム追加)

このように，局所解に陥ることが多かったが，すべての particle が最良解に到達するようになった．

6.3 Schwefel 関数による解析

Schwefel 関数 (変数の数：5) という制約条件付の多峰性関数において，上下限制約と分散アルゴリズムを導入し PSO により最適化を行う．反復回数 1500 回，試行回数 10 回での $f(\boldsymbol{gbest}^k)$ の変動を図 7 に示す．縦軸は評価関数を表す．最適解の評価関数の値は-2094.914435 である．



(a) アルゴリズム導入　　(b) 標準的

図 7 $f(\boldsymbol{gbest}^k)$ の変動 (Schwefel 関数)

アルゴリズム導入により，試行ごとに評価関数が違い最適解に到達しなかったものが，全ての試行で最適解に到達するようになった．

7 シミュレーション，実験

制御対象であるフレキシブルアームのアームの先端を 45 度移動させた応答を示す．PSO により求めた PID パラメータにより出された応答において，局所解は図 8，最良解は図 9 に示す．破線はシミュレーション，実線は実験結果である．

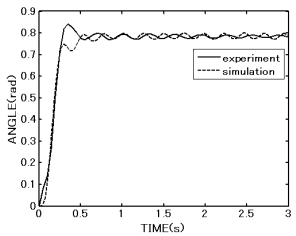


図 8 局所解

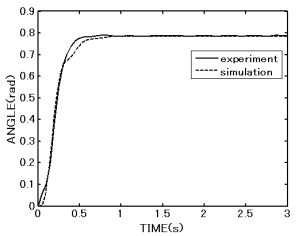


図 9 最良解

このように PSO を用いることにより，最適解ではオーバーシュートのない良い応答になったといえる．

8 おわりに

本研究では制御知識を取り入れた PSO により PID 制御系設計を提案し，以下の成果を得た．

- 制御知識を取り入れることにより，応答が不安定で発散したり，振動的になったり，オフセットが残ったりと制御性能の悪い応答となる探索領域を素早く脱出し，探索領域を限定することで，particle の収束を速くすることができた．
- 反射による探索領域を制約させるアルゴリズムにより，制約領域内での探索が可能になった．
- particle の多様性を回復させる分散アルゴリズムにより、解が局所的から大域的探索になり、解は最良解へ全て到達した。
- 実験による理論の有効性の確認を行った．

参考文献

[1] 小黒龍一：ロボット・工作機械における制御理論応用，計測と制御，**38**-1，42/46 (1999)

[2] 副島慎一，平野豊，岩崎尚：自動車における制御理論，計測と制御，**38**-1，51/54 (1999)

[3] 安田恵一郎，岩崎信弘，井出東：Partical Swarm Optimization:最適化手法としての解釈と解析，設計工学・システム部門講演会講演論文集，**2003**-13，120/123 (2003)

[4] 安田恵一郎，石亀篤司：非線形計画アルゴリズム－実用的観点から，システム/制御/情報，**50**-9，344/349 (2006)

[5] 石亀篤司：Partical Swarm Optimization －群れでの探索－，計測と制御，**47**-6，459/465 (2008)

[6] Mehdi Nasri, Hossein Nezamabadi-pour, and Malihe Maghfoori: A PSO-Based Optimum Design of PID Controller for a Linear Brushless DC Moter, PROCEEDINGS OF WORLD ACADEMY OF SCIENCE, ENGINEERING AND TECHNOLOGY, 20 APRIL 2007 ISSN 1307-6884, 211/215 (2007)

[7] Li Xu-Zhou, Yu Fei, Wang You-bo: PSO Algorithm based Online Self-Tuning of PID Controller, 2007 International Conference on Computational Interloligence and Security, 128/132 (2007)

[8] Ichiro Maruta, Tae-Hyoung Kim, Toshiharu Sugie: Synthesis of fixed-structure H_∞ controllers via Constrained Particle Swarm Optimization, The International Federation of Automatic Control, 7843/7847（2008)

[9] 安藤貴彦，高見勲：GA を用いた極配置による PID 制御，高速信号処理応用技術学会誌，**10**-2，74/80 (2007)

[10] 村田秀樹，相吉英太郎：上下限領域に閉じ込めた Particle Swarm Optimization の力学系の分岐特性と収束特性，電気学会論文誌 C，**126**-7，904/912 (2006)