

局所解回避構造を取り入れた Particle Swarm Optimizationの適用による 非線形系の制御系パラメータ設定

M2009MM001 青木 孝生

指導教員:高見 勲

1 はじめに

近年, 工業製品の機能や性能, コストに対する要求はますます高度化し, 一層の効率化・高信頼化・高機能化のための実用的な最適化手法に対するニーズが更に高まっている. 特にGA(遺伝的アルゴリズム), NN(ニューラルネットワーク), AC(蟻コロニー最適化), PSO(粒子群最適化)等を代表とする進化型アルゴリズムが注目されており, コンピュータパワーの飛躍に伴い, 非線形アルゴリズムを効率的に解くことのできる最適化手法として, システムの設計・製造・運用に利用されてきた [1].

その中で, 本研究では Particle Swarm Optimization(PSO)に局所解回避構造を取り入れ, 非線形系の制御系パラメータ設定に適用することをテーマとし, より信頼できる最適解を発見するために, 従来PSOの問題点を解消するPSOアルゴリズムの開発を目指す.

また本研究では, 状態フィードバックを用いた2リンク回転型倒立振り子 Pendubot の制御系設計を一例とし, 最適な制御応答を実現するパラメータ設定を改良したPSOにより求めていくが, 制御系設計にPSOを適用する観点からも修正を加えて, 従来法よりも良い制御結果を得ることも目標としている.

PSOにおいては, 最終的に発見された解が必ずしも最適解とは限らず, 探索の過程において局所解に陥らないように探索させる必要がある. 一方, より良い結果を得るためには, 局所的探索も重要である. これらの事から現在, PSOにおいて大域的探索と局所的探索の両立が問題点として理解されており, 各論文ではその実現の為, 様々な議論がなされている [2].

例えば, PSOにGAの突然変異の概念を取り入れる事でより大域的な探索を目指したEvolutionaryPSO(EPSO)[3]や, 近傍粒子同士がペアを組んだり解消しながら探索を行う事で, 大域的探索から局所探索に移行するPairPSO[4]などがある. 近年では, 制御系設計への適用例も増えており, PSOをシステム同定に利用した例 [5]や, PIDパラメータをPSOで求めている例 [6]などがある.

本研究では, PSOの情報交換形態において一般的な手法である, 情報を常に全体で共有しているGbestモデル [2]ではなく, 情報の伝達経路を段階化したモデルを用いることで, 情報伝達速度を故意に遅くし大域的探索を実現し, 最適解発見の可能性を高める. また, 探索の過程では特定の区域にparticleが集まってしまい探索が滞る傾向があるので, particleの堆積数が一定値以上になったらparticleを散らすアルゴリズムを導入し, 探索の効率化を図る.

2 PSOとは

粒子群最適化(PSO)は生物の採餌行動を模擬した最適化手法で, 微分可能性などの条件を必要とせず, 目的関数値の情報のみで非線形最適化問題を解く事ができる直接探索型のアルゴリズムである [7]. PSOは多次元空間において, 位置ベクトルと移動ベクトルを持つparticle(探索点)の群でモデル化される. これらのparticleが情報交換をし, それに基づいて自身の位置と速度を調整しながら解空間を探索し, 最適な位置を探す.

2.1 PSOの概要

n 次元空間におけるparticleは, 位置ベクトル $x_i = (x_{i1}, x_{i2}, \dots, x_{ij}, \dots, x_{in})^T$, 移動ベクトル $v_i = (v_{i1}, v_{i2}, \dots, v_{ij}, \dots, v_{in})^T$ で表される. ここで, 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はそれまでの探索で発見した最良解 $pbest_i = (pbest_{i1}, pbest_{i2}, \dots, pbest_{ij}, \dots, pbest_{in})^T$ と, その評価値である $f(pbest_i)$ を記憶している. 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解 $gbest = (gbest_1, gbest_2, \dots, gbest_j, \dots, gbest_n)^T$ と, その評価値である $f(gbest)$ を記憶している.

これらのベクトルを用いて, 各particleは新たな移動ベクトル v_i^{k+1} を生成し, 現在の位置 x_i^k から次の位置 x_i^{k+1} へ移動する. ここで, k は反復回数を表す.

2.2 PSOのアルゴリズム

以下に一般的なPSOのアルゴリズムを示す [1]. ここで, $i = 1, 2, \dots, m, j = 1, 2, \dots, n$ とする. ただし, m はparticleの数である.

Step 0: 準備

m , および最大反復回数 T_{max} を与え, $k = 0$ とおく.

Step 1: 初期化

各particleの初期位置 x_i^0 と初期速度 v_i^0 をランダムに与える.

$$pbest_i^0 = x_i^0$$

$$gbest^0 = pbest_{i_g}^0$$

ただし, $i_g = \arg \min_i f(pbest_i^0)$

Step 2: 位置と速度の更新

particleの位置と速度を以下の式に従って更新する. ただし, $\text{rand}_1()_{ij}, \text{rand}_2()_{ij}$ は0から1の間に分布する一様乱数であり, w, c_1, c_2 は各項に対する重み係

数である．

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

$$x_{ij}^{k+1} = x_{ij}^k + v_{ij}^{k+1} \quad (2)$$

Step 3 : pbest と gbest の更新

$$I = \{i \mid f(x_i^{k+1}) < f(pbest_i^k)\}$$

$$pbest_i^{k+1} = x_i^{k+1}, i \in I$$

$$pbest_i^{k+1} = pbest_i^k, i \notin I$$

$$gbest^{k+1} = pbest_{i_g}^{k+1}$$

とおく．ただし， $i_g = \arg \min_i f(pbest_i^{k+1})$

Step 4 : 終了判定

$k = k+1$ としてStep 2へ戻る．もし $k = T_{max}$ であるならば，最適解を $gbest^{k+1}$ ，最適値を $f(gbest^{k+1})$ として終了する．

3 PSOの大域探索性能の向上

3.1 情報伝達遅延モデル

PSOでは一般に，群としての最良解である $gbest$ の情報を群全体で共有するモデルを採用している．しかし，最良解に向かうベクトルを強くすると，収束速度が速くなる一方で大域的探索の可能性は失われてしまう．すると一度収束した解が局所解であった場合，そこから抜け出すのは困難である．

そこで本研究では，最良解の情報伝播経路を段階化し情報伝達速度を遅くすることで，大域的探索の可能性を広げる．いくつかのparticleの群れが局所解に収束してしまっても，その情報を共有していない群れが存在するため，局所解に捕らわれている群れに先導されて全体が局所解に陥ってしまう危険を回避し，最適解発見の可能性を高めることができる．

まず，particleの全体 R を， L 個の個別探索群に以下の様に分ける．ここで， L は情報伝達をどれほど段階化するかの定数である．

$$R_l = \{i \mid \frac{(l-1)m}{L} < i \leq \frac{lm}{L}\}$$

ただし， $l = 1, 2, \dots, L$

ここで， L 個に分けられた各群が保有する最良解を $gbest_{R_l} = gbest_{R_{l1}}, gbest_{R_{l2}}, \dots, gbest_{R_{lL}}$ とし，移動するparticleが，所属する群 R_l が保持している最良解 $gbest_{R_l}$ を参照して移動するように，更新式(1)を下式に変更する．

$$v_{ij}^{k+1} = w \cdot v_{ij}^k + c_1 \cdot \text{rand}_1()_{ij} \cdot (pbest_{ij}^k - x_{ij}^k) + c_2 \cdot \text{rand}_2()_{ij} \cdot (gbest_{R_l j}^k - x_{ij}^k) \quad (3)$$

次に，前節のStep3における $gbest$ の更新式を以下のように変更する．

- $(1 - \frac{1}{2^{(\alpha-1)}})T_{max} \leq k < (1 - \frac{1}{2^\alpha})T_{max}$ のとき，
($\alpha = 1, 2, \dots, L-1$)

$$gbest_{R_l}^{k+1} = pbest_{i_g}^{k+1}$$

ここで $i_g = \arg \min_{i \in R_l} f(pbest_i^{k+1})$

$$\tilde{R}_l = \bigcup_{\beta=1}^{\alpha} R_{\psi} \quad \psi = l + \beta - 1$$

$$\text{ただし, } R_{\psi} = \begin{cases} R_{\psi-L} & \text{if } \psi > L \\ R_{\psi} & \text{if } \psi \leq L \end{cases}$$

- $(1 - \frac{1}{2^{(L-1)}})T_{max} \leq k < T_{max}$ のとき，

$$gbest_{R_l}^{k+1} = pbest_{i_g}^{k+1}, i \in \bigcup_{\beta=1}^L R_{\beta}$$

ここで $i_g = \arg \min_i f(pbest_i^{k+1})$

ここで，上記の更新式の変更は，探索の試行回数が重なるにつれ，分けられた群各々の最良解情報の共有範囲を拡大することを表している．すなわち，探索初期では各群が個別に探索を行い，終盤になるに従って隣の群，そのまた隣の群，と情報共有範囲を拡大し，最終的にはすべてのparticleが情報を共有することで，局所解に陥る可能性を排除しながら局所的探索へ移行している(図1)．

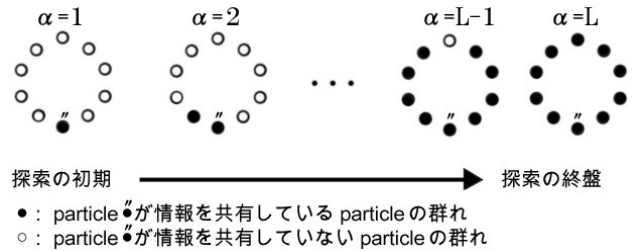


図1 変更したモデルの概略図

3.2 PSOの改良：堆積数の導入

次に，探索過程(特に後半の探索)においてparticleが過度に $gbest$ 付近に集まってしまう事が知られている[4]．密集したparticleは動きが鈍くなり，周辺探索を行わなくなるため，探索を効率的に行うには，避けるべき傾向であると言える．

この傾向を回避し，探索の後半においても効率的な最適解の探索をするために，particleの堆積数 ρ を導入する．そしてその時点での最良解である $gbest$ の位置からの距離が D_{zone} 以内にあるparticleの堆積数が，閾値 γ 以上であれば過密であると判断し，堆積したparticleを発散させる．これにより，particleの過密による探索の停滞を防ぎ，より効率的な探索の実現を目指す．堆積数を導入したアルゴリズムは，2.2節におけるStep3の次に，下記の操作の実行を挿入すればよい．

Step 3.1 : 準備・初期化

堆積と判断する範囲を決める D_{base}, D_{range} ，発散範

囲を決める $emit_{range}, emit_{base}$, 発散させる割合 η , 閾値 γ を与え, $\rho = 0$ とおく . また, $rand_3()_{ij}$ は-1 から 1 の間に分布する一様乱数である .

Step 3.2 : 堆積数のカウント

$$\rho = \begin{cases} \rho + 1 & \text{if } i \in S \\ \rho & \text{if } i \notin S \end{cases}$$

$$S = \{i \mid \sqrt{(gbest_1^k - x_{i1}^k)^2 + \dots + (gbest_j^k - x_{ij}^k)^2} \leq D_{zone}\}$$

$$D_{zone} = D_{base} + D_{range} \cdot \frac{T_{max} - k}{T_{max}}$$

Step 3.3 : 堆積した particle の発散

もし $\rho \geq \gamma$ であるなら下式を実行し, $\rho < \gamma$ であるなら 2.2 節の Step4 へ移動する . ここで ϕ は, どの程度 particle を発散させるかを定める係数である .

$$x_{ij}^k = (gbest_1^k + \phi \cdot rand_3()_{ij}, \dots, gbest_j^k + \phi \cdot rand_3()_{ij}), i \in S$$

$$\phi = emit_{base} + emit_{range} \cdot \frac{T_{max} - k}{T_{max}}$$

$$U = \{i \mid x_{ij}^k = (gbest_1^k + \phi \cdot rand_3()_{ij}, \dots, gbest_j^k + \phi \cdot rand_3()_{ij}), i \in S\}$$

U は発散させた particle の集合を表し, $i \in U$ となる particle の個数が $\rho \cdot \eta$ を上回ったなら終了し, 次の Step へ移動する .

Step 3.4 : 発散させた particle の初期化

発散させた particle の過去の $pbest$ 情報, 慣性項である移動ベクトル v_i を剥奪し, 初期化する .

$$pbest_{ij}^k = x_{ij}^k, i \in U$$

$$v_{ij}^k = v_{rand} \cdot rand_3()_{ij}, i \in U$$

ここで, v_{rand} は, 速度 v を与える際の重み係数であり, 終了したら 2.2 節の Step4 へ移動する .

4 PSO の制御系設計への適用

PSO は制御モデルの線形性, 評価関数の微分可能性も要求しないので, どのような対象にも適用でき, 非線形な対象についても問題なく適用できる . 本章では制御系のパラメータ設定に PSO を適用する観点から, PSO に修正を加え, 従来よりも効率的なパラメータチューニングの実現を目指す .

本研究では, 状態フィードバックで制御する際のフィードバックゲインを particle の位置として考えて PSO を適用する . また, PSO を制御系に適用するために必要な, 評価関数の設定を制御の観点から行い, その手法を以下に示す .

4.1 評価関数 J の設定

良い制御応答とは, 安定であり, 過渡応答が良いことを指す . これを評価するための基準として, 評価関数 J を導入して, シミュレーションと PSO を連動させて制御性能を評価し, 最適化する . 評価関数 J は式 (4) で与えられ, 図 2 に示すようなオーバーシュート量 (Z_{OS}), 誤差面積 (Z_{error}), 整定時間 (Z_{Ts}) を用いて設定している [8] . ここで, 誤差面積は制御偏差の絶対値の積分であり, 整

定時間は目標値からの誤差 $\pm 2\%$ として設定した . また, a_1, a_2, a_3 は各項に対する重み係数であり, この J を最小とするフィードバックゲインを求めていく事になる .

$$J = a_1 \cdot Z_{OS} + a_2 \cdot Z_{error} + a_3 \cdot Z_{Ts} \quad (4)$$

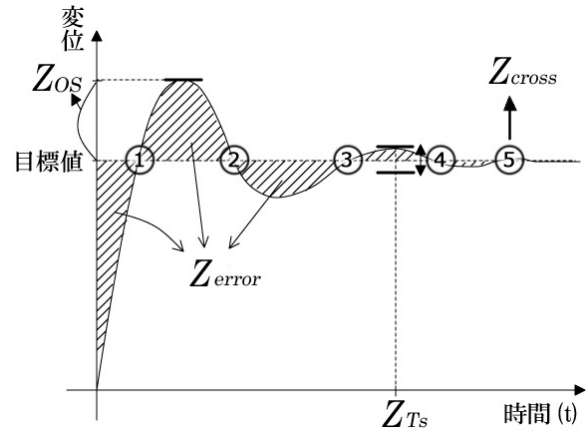


図 2 評価関数の設定

4.2 評価関数 J の修正 : 操作量の抑制

制御での問題点として, 操作量が過大になってしまい, 非現実的な例がしばしば見受けられる . この問題を防止する為に, 操作量の変動 (Z_{input}) とピーク値 (Z_{peak}) の指標を加えた式 (5) を評価関数 J に加え, 操作量の過大を抑制する . ピーク値を入れる事で非現実的なシミュレーション結果を排除でき, 操作量変動を入れる事でより少ない操作量での制御が実現できる .

$$J_1 = a_4 \cdot Z_{input} + Z_{peak} \quad (5)$$

$$Z_{peak} = \begin{cases} \infty & \text{if } u_{peak} \geq u_{lim} \\ 0 & \text{otherwise} \end{cases}$$

ここで, 操作量の変動 Z_{input} とは平衡状態からの, 操作量偏差の絶対値の積分である . また, a_4 は重み係数, u_{peak} は操作量の変動ピーク, u_{lim} は操作量変化の限界である .

4.3 評価関数 J の修正 : 目標値との交差制限

位置制御などのサーボ系制御では, 目標値付近での振動現象は避けるべき事柄であり, 少しい行き過ぎも認めない制御が要求される時もある . 式 (4) のままでは, このような事例に対処する事は難しい . そこで, 制御応答が目標値をまたぐ回数 (Z_{cross}) を測定し, その回数に比例してペナルティを与える式 (6) を追加する (図 2) . これにより, 目標値の行き過ぎを防ぐ事ができる .

$$J_2 = a_5 \cdot Z_{cross} \quad (6)$$

a_5 は重み係数であり, $a_5 = \infty$ とすれば, 行き過ぎは起こり得ない .

これらをまとめると, 修正された評価関数 J_{new} は下式のようになる .

$$J_{new} = J + J_1 + J_2 \quad (7)$$

5 シミュレーション・実験による検証

5.1 制御対象

本研究では図3の2リンク回転型倒立振りPendubotを制御対象とし、シミュレーションや実験を行って理論を検証する。pendubotは単独では不安定かつ非線形特性を有しているため、PSOによる非線形系の制御パラメータ設定を検証する上で最適である。

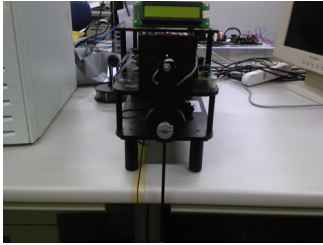


図3 pendubot

ここで、制御対象はリンク1,2と分かれている為、式(7)に基づいて各リンクで測定した評価値を J_{link1}, J_{link2} とし、最終的な評価関数 $J_{pendubot}$ を式(8)で与える。また、今回は $m = 200, T_{max} = 512$ として計算した。

$$J_{pendubot} = J_{link1} + J_{link2} \quad (8)$$

5.2 シミュレーション結果の比較・検討

PSOアルゴリズムを改良した結果、評価関数を修正した結果を図に示す。図4では、情報伝達遅延モデルの適用前後で探索を10回ずつ行い比較している。モデル変更により局所解を回避し、より良い解を発見できた。図5では、particleの移動速度の平均を比較している。堆積数を導入した事により、探索の後半においてもparticleが停滞する事なく探索を続けている。

図6は、式(5)の操作量抑制の有無の比較で、操作量抑制を設けたことで、過大でない現実的な解を見つけられるようになった。図7は $a_5 = \infty$ として式(6)の交差制限を加えた例で、これにより行き過ぎがなくなった。

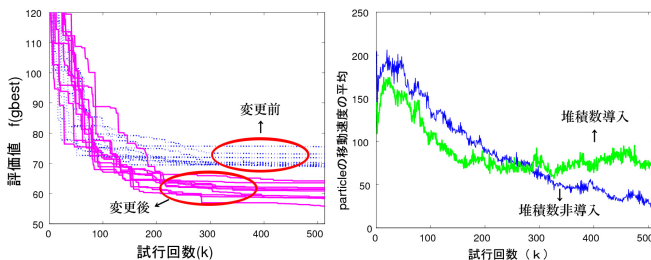


図4 情報モデル変更による効果 図5 堆積数導入の効果

5.3 実験結果の比較・検討

最終的に得られた実験結果を、以下に示す。図8、図9から、改良したPSOによって得られた制御応答は、立ち上がりが速くオーバーシュートも少なく、従来法よりも良い制御応答が得ることができたと言える。

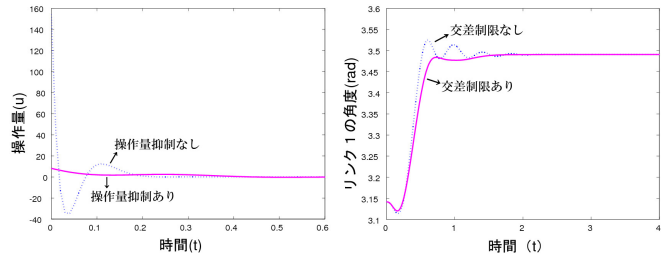


図6 操作量抑制による効果 図7 交差制限による効果

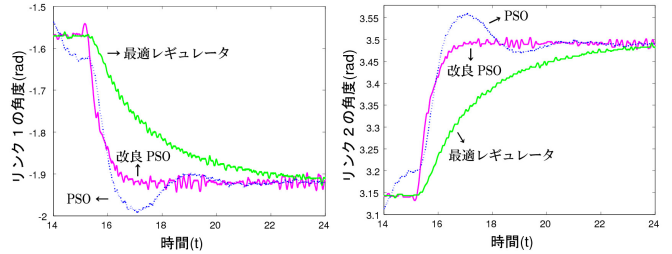


図8 実験(リンク1)の比較 図9 実験(リンク2)の比較

6 おわりに

本研究では以下の成果を得た。

- PSOを非線形特性を持つ対象の制御パラメータ設定に適用した。
- PSOアルゴリズムを改良する事で、局所解を回避し、従来法よりも良い解に到達できた。
- 制御の観点から評価関数に修正を加えることで、より実用的な解を得られるようになった。
- 実験を行い、理論の有効性を検証した。

参考文献

- [1] 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/情報第50巻第9号, システム制御情報学会, 2006
- [2] 石亀篤司, Particle Swarm Optimization-群れでの探索-計測と制御, 第47巻第6号, pp459-465, 2008
- [3] Miranda and N.fonseca, EPSCO-Evolutionary Particle Swarm Optimization, a New Algorithm with Applications in Power Systems, IEEE/PES Transmission and Distribution Conference and Exhibition 2002, 745/750, 2002
- [4] 緒方芳道, 小野智司, 中山茂, Pair Particle Swarm Optimizationによる複数解探索法の基礎検討, システム制御情報学会論文誌, Vol.19, No.8, pp.330-332, 2006
- [5] 和田堯, 杉江俊治, Particle Swarm Optimizationに基づく周波数領域におけるシステム同定, システム制御情報学会論文誌, Vol.23/No.10, pp236-242, 2010
- [6] S.Easter Selvan, Sethu Subramanian, S.Theban Solomon, Novel Technique for PID Tuning by Particle Swarm Optimization, in Proc.7th Annu.Swarm Users/Researchers Conf.(SwarmFest 2003), Notre Dame, IN, 2003.
- [7] J.Kennedy, R.Eberhart Particle Swarm Optimization, IEEE Int. Conf. on Neural Networks, Vol.4, pp1942-1948, 1995
- [8] 安藤貴彦, 高見勲, 制御知識を取り入れた Particle Swarm OptimizationによるPID制御系設計, 高速信号処理応用技術学会誌, Vol.11/No.2, 59-64, 2008