

# Particle Swarm Optimization を用いたPID制御のパラメータチューニング

2003MM004 青木孝生

指導教員：高見勲

## 1 はじめに

現在, PID 制御は多くの場所で使用されており, 最も利用されている制御方法と言ってよい. しかし一方で, 専門家の手によるパラメータ調整が必要で, 多大な労力と時間を費やす必要がある. そのため近年, パラメータの自動チューニングが注目されている. そこで本研究では, 最適化手法として Particle Swarm Optimization(以下, PSO) を制御に導入する事でパラメータチューニングを行い最適な解を探索する. また, より良い解を発見するために, 探索の多様化を目的に PSO を改良していく.

## 2 Particle Swarm Optimization

粒子群最適化 (PSO) は群知能の一種であり, 例えば魚の群れの中の一匹が良さそうな経路を発見すると, 群れ全体にも情報が伝わり, 群れの残りの素早くそれを追う事が可能となる. 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 [1]. 以下に, PSO の概要をまとめる.

### 2.1 Particle Swarm Optimization 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.

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  $f(pbest_i)$  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  $f(gbest)$  を記憶している.

各 particle は更新式 (1) に従って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 [2]. ここで,  $i$  は particle の番号,  $j$  はベクトルの要素番号,  $k$  は反復回数,  $\text{rand}_1()_{ij}, \text{rand}_2()_{ij}$  は 0 から 1 の間の一様乱数,  $w, c_1, c_2$  は重み係数である.

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

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

## 3 PSO の改良

PSO において, 発見された解は最適解であるとは限らず, 探索の多様化を行う事が精度を高める為に必要とされる. そこで, 初回の探索情報から解空間を 4 分割 (自分の想像できる範囲で分割の割合を選んだ) する分割平面を

生成し, 二回目の探索で particle を 4 群に分けて各空間を探索させる事を考える. 最終的に各群が発見した最適解を比較し最も良い評価値を最適解とする. ただし本章では 3 次元空間内の探索を考える事とする.

初回の探索では, 評価値の良い順に 4 点  $num_i = (num_{i1}, num_{i2}, num_{i3})$  をとり, 各点を結んで作る四角形の辺の中点を求め, 向かい合う辺の中点同士を通る 2 直線  $line1, line2$  を生成する. ただし,  $num_i$  の距離が近すぎるのは好ましくないので, 各点間の距離  $distance_{ij}$  を一定値  $d$  以上とする制約条件を与える. ここで  $i = 1, 2, 3, 4$ ,  $j = 1, 2, 3, 4$  とし,  $i = j$  の場合は除く.

次に,  $line1, line2$  から分割平面を決定する第 3 点  $P$  を求めるが, まず  $num_i$  から  $line1, line2$  へ垂直に交わる点  $H1, H2$  を求める.  $line1$  上にある 2 つの中点を  $E = (e_1, e_2, e_3), F = (f_1, f_2, f_3)$  とおき,  $G = num_i$  とおき, 図 1 の様な配置の場合を考えると, 下式の様に求まる.

$$\vec{EG} = (num_{i1} - e_1, num_{i2} - e_2, num_{i3} - e_3)$$

$$\vec{EF} = (f_1 - e_1, f_2 - e_2, f_3 - e_3)$$

$$\gamma = \frac{EG_1 \cdot EF_1 + EG_2 \cdot EF_2 + EG_3 \cdot EF_3}{EF_1^2 + EF_2^2 + EF_3^2}$$

$$H1 = (\gamma f_1 + (1 - \gamma)e_1, \gamma f_2 + (1 - \gamma)e_2, \gamma f_3 + (1 - \gamma)e_3)$$

ただし, ここでの  $EF, EG$  の添え字は  $\vec{EF}, \vec{EG}$  の成分の要素番号を表す. 上式の  $H1 = (h_{11}, h_{12}, h_{13})$  とおき, 図 2 に示す角度  $\theta_i$  を求めると式 (3) のように求まる. そして  $\theta_i$  から, 分割平面を傾ける角度を  $\beta$  と選べば, 点  $P$  が式 (4) のように定まる. ここで図 2 は, 図 1 に示す  $line1$  と平行な矢印から見た図である.

$$D_s = h_{1s} - num_{is}, s = 1, 2, 3$$

$$l = \sqrt{(D_1)^2 + (D_2)^2}$$

$$\theta_i = \sin^{-1} \left( \frac{|D_3|}{\sqrt{(D_1)^2 + (D_2)^2 + (D_3)^2}} \right) \quad (3)$$

$$p_3 = h_{13} - \sqrt{\left( \frac{l}{\sin \beta} \right)^2 - (D_1)^2 - (D_2)^2}$$

$$P = (num_{i1}, num_{i2}, p_3) \quad (4)$$

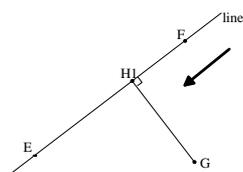


図 1 点 H の決定

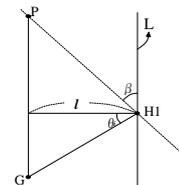


図 2 点 P の決定

さらに  $line1$  を作成する際に用いた中点を  $m_1 = (m_{11}, m_{12}, m_{13}), m_2 = (m_{21}, m_{22}, m_{23})$  とし,  $P = (p_1, p_2, p_3)$  とすると, 分割平面  $H1_d$  が以下の様に求まり, 同様の手順で  $line2$  に関する分割平面  $H2_d$  も作成する.

$$\begin{aligned} \delta_1 &= m_{12}(m_{23} - p_3) + m_{13}(p_2 - m_{22}) + m_{22}p_3 - m_{23}p_2 \\ \delta_2 &= m_{11}(p_3 - m_{23}) + m_{13}(m_{21} - p_1) - m_{21}p_3 + m_{23}p_1 \\ \delta_3 &= m_{11}(m_{22} - p_2) + m_{12}(p_1 - m_{21}) + m_{21}p_2 - m_{22}p_1 \end{aligned}$$

$$H1_d : \delta_1(x - m_{11}) + \delta_2(y - m_{12}) + \delta_3(z - m_{13}) = 0$$

### 3.1 particle の反射

次に, 解空間内を particle に探索させる. だが particle が移動する際, 担当の空間から別の解空間に飛び出る particle が出現する為, このような particle を担当の空間内におさめる必要がある. 本研究では分割平面に対して反射させる対処方法を取り (反射により particle が分割平面近傍に集まる傾向を弱められる), その様子を図 3 に示す.

ただし, 更新前の位置  $x_i^k = (x_{i1}^k, x_{i2}^k, x_{i3}^k)$  から更新後の位置  $x_i^{k+1} = (x_{i1}^{k+1}, x_{i2}^{k+1}, x_{i3}^{k+1})$  へ移動時に境界面を超えたとし, 修正後の位置が  $x_i^{k+1}$  である.  $x_H$  は  $x_i^{k+1}$  から平面  $H1_d$  と垂直に交わる直線と  $H1_d$  との交点である. 図 3 に基づき  $x_i^{k+1}$  を求めると,  $x_H$  に対応するパラメータの値  $t_H$  が求まり,  $x_H$  と  $x_i^{k+1}$  が次式の様に求まる.

$$t_H = \frac{\delta_1(x_{i1}^{k+1} - m_{11}) + \delta_2(x_{i2}^{k+1} - m_{12}) + \delta_3(x_{i3}^{k+1} - m_{13})}{-(\delta_1^2 + \delta_2^2 + \delta_3^2)}$$

$$x_H = (\delta_1 t_H + x_{i1}^{k+1}, \delta_2 t_H + x_{i2}^{k+1}, \delta_3 t_H + x_{i3}^{k+1})$$

$$x_i^{k+1} = (2\delta_1 t_H + x_{i1}^{k+1}, 2\delta_2 t_H + x_{i2}^{k+1}, 2\delta_3 t_H + x_{i3}^{k+1})$$

### 3.2 評価関数 $J$ の設定

particle の位置を評価する評価関数  $J$  を図 4 に示すようなオーバーシュート ( $OS$ ), 誤差面積 ( $error$ ), 整定時間 ( $T_s$ ) を用いて設定すると, 下式ようになる. ここで, 誤差面積は単位時間ごとの目標値からのズレの和であり, 整定時間は目標値からの誤差  $\pm 2\%$  として設定した. また,  $c_{OS}, c_{error}, c_{T_s}$  は重み係数である.

$$J = c_{OS} \cdot OS + c_{error} \cdot error + c_{T_s} \cdot T_s$$

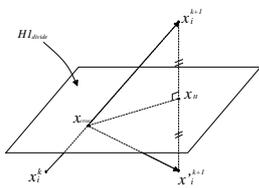


図 3 particle の反射

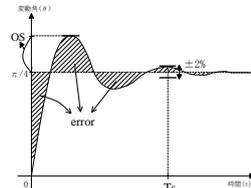


図 4 評価関数の設定

## 4 制御対象

本研究ではフレキシブルアームを制御対象とする. フレキシブルアームはアーム部の厚みが薄い為, アームの振る舞いは振動的である. モータの電圧を調整し, アームの角度を制御する [3].

## 5 シミュレーションと実験

アームの角度を  $\pi/4$  動かすシミュレーションと実験を PID 制御を用いて行った. 改良前後の PSO で得た最適解を PID パラメータとし, シミュレーションを行なった結果の比較を図 5, 実験結果の比較を図 6, PSO の適応前後の比較を図 7 に示す. ほぼシミュレーション通りの実験結果となっている. PSO の適応により制御結果がかなりよくなり, チューニングの労力が省けた. また, 改良後 PSO の方が立ち上がりが早く, 最適なチューニングができたといえる.

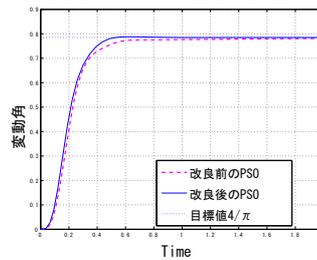


図 5 シミュレーション結果

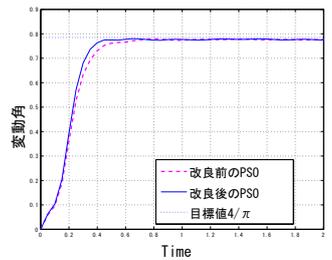


図 6 実験結果

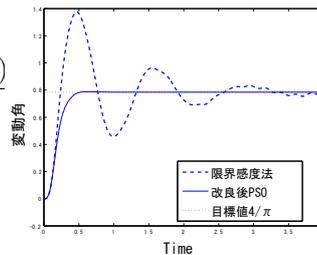


図 7 PSO の適応効果

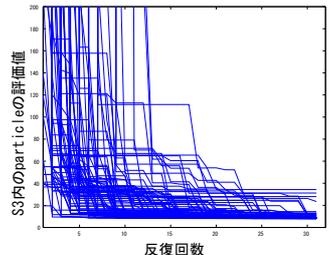


図 8 particle の動き

## 6 おわりに

本研究で得られた成果は以下の点である.

- PSO を制御に導入する事で, PID パラメータの自動チューニングを行い, 専門家の手による手間や労力を必要とせず最適な解を得ることができた.
- 解空間分割を行い, 各群が個別探索をするアルゴリズムを PSO に導入し, 探索の多様化を実現した.
- シミュレーションにかなった波形が実験で見られ, 改良前の PSO より良いパラメータの発見ができた.

## 参考文献

- [1] 立田昌也・呉世訓・堀洋一: 工作機械の精密制御のための最適パラメータ探索法. 産業計測制御研究会, IIC-07-95(2007)
- [2] 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)
- [3] 久米洋平: フレキシブルアームに対する ILQ 最適サーボ系設計, 南山大学数理情報学部数理科学科論文 (2006)