

工作機械の精密制御のための最適パラメータ探索法

立田昌也*, 呉世訓, 堀洋一 (東京大学)

Optimal Parameter Search Method for Precision Control of Machine Tool

Masaya Tateda*, Sehoon Oh, Yoichi Hori (The University of Tokyo)

Abstract

For precision control of machine tool, a lot of control parameters must be tuned properly. So various novel search algorithm (for example, GA(Genetic Algorithm), PSO(Particle Swarm Optimization)) are used in auto-tuning. But they are not very fitted for auto-tuning because of characteristics of machine tool.

In this paper, we propose new novel search algorithm better than conventional search algorithm in tuning control parameter of machine tool. One of new novel search algorithm is improved PSO, and the other is PSO using NN(Neural Network). Effectiveness of improved PSO is verified by experimental results, but that of PSO using NN is not so clear.

キーワード：粒子群最適化, 神経回路網, 遺伝的アルゴリズム, 自動チューニング
(particle swarm optimization, neural network, genetic algorithm, auto-tuning)

1. はじめに

近年, 多くの場所で工作機械が活躍するようになってきている。そして, それに大きく貢献したものが NC(Numerical Control) 装置である。NC 装置は工作機械の位置, 運動などを数値的に制御する装置であり, 産業用モータの精密制御を実現している。

NC 装置は精密制御に欠かすことができないが, NC 装置を伴った工作機械は購入すればすぐに使えるものではない。なぜなら, 多くのパラメータの調整が必要となるからである。現在は専門家の手で調整を行っているが, それには多大な時間と労働力が必要となってしまう。

そこで, 最近では制御器のパラメータ自動チューニングが着目されてきている。制御器自身が自動的に最適なパラメータを見つけることができれば, 今まで人間が行っていたチューニングの作業が必要なくなり, そして信頼できるパラメータを設定することができる。

本論文では, 多目的最適化手法としてよく使われている GA(genetic algorithm), PSO(particle swarm optimization) を HIL で実際確かめ, それらの問題点を考慮に入れた上で, 工作機械の最適パラメータ探索に有効なアルゴリズムを提案することで, 工作機械をさらに便利なものにすることを目的とする。

ここで, まず HIL システムにおけるパラメータ自動チューニングの特徴・留意点を挙げておく。

- 制御器の性能を正確に表す適合度指標は一意的ではないが, 高い適合度を持つところが何箇所もある多峰性問題とみるよりは, いくつかの山はあるかもしれないが大きなピークを一つ持つ単峰性問題とみてよい。これは, 人間が行ってきたパラメータチューニング作業の経験から確かめられる。
- 実際のハードウェアを動かす最適化作業であるので, 特

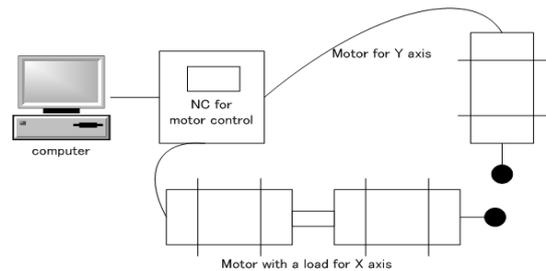


図 1 実験に使用する HIL システム
Fig. 1. HIL system for experiments

別な理由もなくハードウェアを壊す可能性がある極端的な値, または空間を探索してはいけない。また, 同じ理由から出来るだけ少ない試行回数で確実に最適化を行う必要がある。

- ある程度適合度の高い領域を探索するようになってからは, 最適解を求めるために微調整を行わなければならない。人間の場合は, 適合度をあげる為にどのパラメータを調整すればいいのかは経験的に予測できるが, オートチューニングの場合は制御器の特性モデルが分かっていないためモデルに基づく最適化は利用できない。などがある。

2. 実験環境

〈2・1〉 実験機について 今回使用する実験機は HIL システム (Hardware-In-the-Loop) に組み込まれている。HIL システムとは, 制御装置と計算機を接続して実際に制御装置を動かしながらパラメータ最適化をするためのシステムであり, 図 1 のように構成されている。また, システム全体のブロック線図は図 2 のようになっており, 調整可能なパラメータは V_{ff} や K_p, k など数多く存在している。

〈2・2〉 実験条件 次節以降で行う自動チューニング

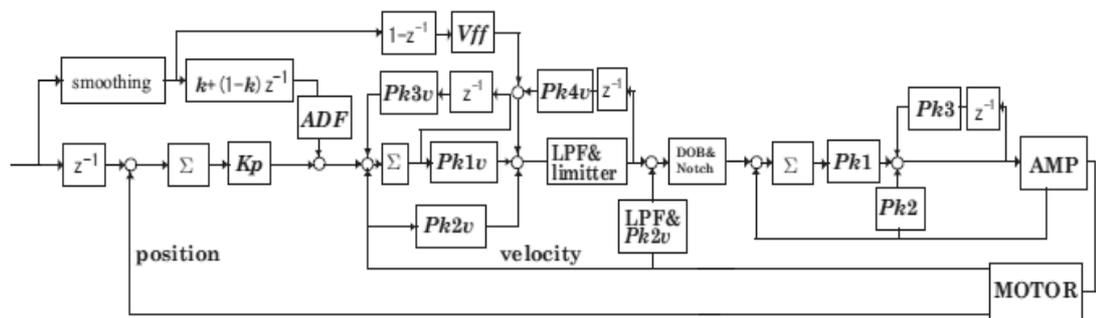


図 2 システムのブロック線図
Fig. 2. Block diagram of system

は、調整するパラメータを以下のようにした。

- P1:直線加減速の加速度 50~550
- P2:円弧指令に対する許容速度 1000~4000
- P3:FF タイミング調整計数 -8192~4096
- P4:速度 FF ゲイン x 軸 200~1500
- P5:速度 FF ゲイン y 軸 50~700

(注) 各パラメータの値は NC の中でリスケーリングしているため、値それ自体は特に意味を持たない
そして、制御入力に 4 隅が円弧の正方形の位置指令信号を送り、性能評価は形状誤差 (Error) と軌道生成にかかった時間 (Time), それらを元に式 (1) で算出した適合度 (Fitness) をもって行う。

$$Fitness = S_e(e_{bias} - \sum trajectory\ error(t)) + S_t(t_{bias} - Elapsed\ time) \dots\dots\dots (1)$$

- S_e : 形状誤差に対する重み係数
- e_{bias} : 形状誤差基準値
- S_t : 時間に対する重み係数
- t_{bias} : 時間基準値

各パラメータの最適値は手で調整を行った結果、表 1 のようになり、次節以降の自動チューニングの実験ではこの値に近づけることを目標とする。

3. 既存の多目的最適化手法による自動チューニング

〈3・1〉 GA による自動チューニング GA は最適化のオペレーションとして選択 (selection), 交叉 (crossover), 突然変異 (mutation) を用いている。生態系の最適化を取り入れた点で理解しやすい面もあるが、初期値への依存性が高い問題点を持つ。初期値、生態系にたとえて言えば種が悪ければ交叉による最適化を行っても適合度の低いパラメータしか得られない時がしばしばある。突然変異がその初期値依存性を解決できる手段であるが、この突然変異も必ず適合度を向上させる方向に起きるとは保証できない。⁽²⁾

ここで、実際に GA を用いてパラメータチューニングをし GA の特性を検証してみる。集団数 : 24, 世代数 : 16, 交叉率 : 0.9, 突然変異率 : 0.1 として実験を行った。結果は図 3 のようになり、各世代の適合度、形状誤差と軌道生成にかかった時間の更新結果を表示している。適合度に関してみると、それぞれ世代が増えれば確かに適合度が増えていることがわかるが、まだ値のバラつきが残っていて完全に収束し切っていない。これも突然変異や交叉が戦略性を持って行われていないためであり、このような特性では最適値に行き着くには確率的な問題となり、制御器のパラメータチューニングには受け入れ難い結果である。

〈3・2〉 PSO による自動チューニング 粒子群最適化 (Particle Swarm Optimization, PSO) とは群知能の一種であり、魚や動物の群れを例えに挙げると、一匹が良さそうな経路を発見すると (すなわち、食料を発見したとか安全であるという場合), 群れの残りはどこにいても素早くそれを追うことができる。

これは多次元空間において位置と速度を持つ粒子群でモデル化される。これらの粒子はハイパー空間を飛びまわり、最善な位置を探す。群れのメンバーは良い位置について情報交換し、それに基づいて自身の位置と速度を調整する。そして、その位置と速度の更新は式 (2),(3) で行われる。

$$x_i(k+1) = x_i(k) + v_i(k+1) \dots\dots\dots (2)$$

$$v_i(k+1) = \alpha(v_i(k) + \beta_1 rand_1(x_{pbest_i} - x_i(k)) + \beta_2 rand_2(x_{gbest} - x_i(k))) \dots\dots\dots (3)$$

世代が変わるときに, PSO は式 (3) で計算された $v_i(k+1)$ だけ各パラメータは変動する。ただし, 式 (3) における x_{gbest} とはその世代までもっとも最適だったパラメータ, x_{pbest_i} は個体 i がその世代まで探索した中での最も最適だったパラメータを意味し, $rand_1, rand_2$ はそれぞれ 0 と 1 の間の乱数を意味する。^{(3) (4)}

実際に実験をして PSO の検証をしてみる。 $\alpha = 0.8, \beta_1 =$

表 1 パラメータの最適値
Table 1. Best value of parameter

P1	P2	P3	P4	P5	Error($\mu m \cdot s$)	Time(sec)	Fitness
237	3768	-4500	1175	50	33.9208	1.949	45.4081

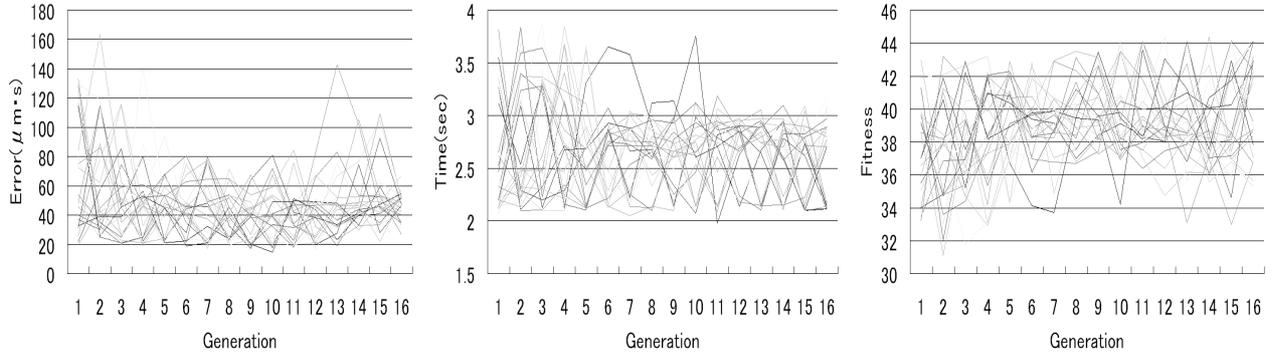


図 3 GA によるパラメータチューニング
Fig. 3. Tuning of control parameter using GA

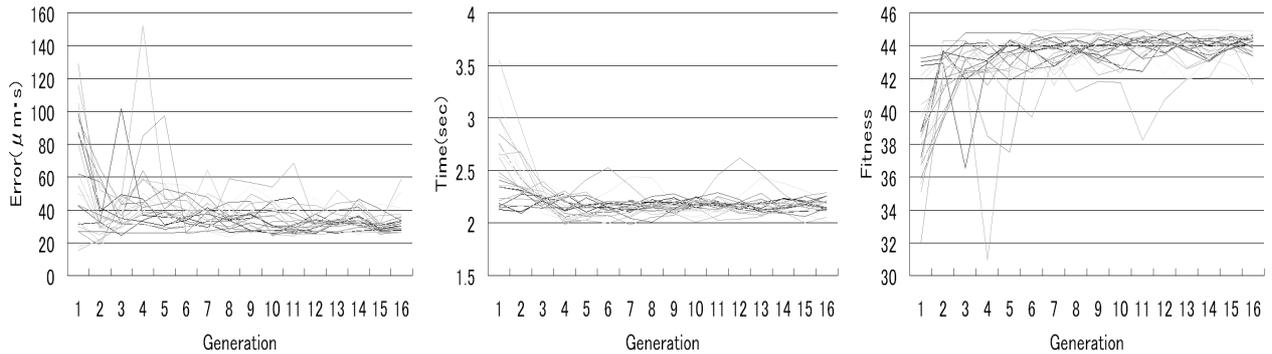


図 4 PSO によるパラメータチューニング
Fig. 4. Tuning of control parameter using PSO

1.0, $\beta_2 = 1.0$, 集団数 : 24, 世代数 : 16 とし 3 度実験を行った。1 度目に行った結果を表すと図 4 のようになり, GA の時と同様に各世代の適合度, 形状誤差と軌道生成にかかった時間の更新結果を表示している。適合度に関して見てみると, ランダムに決められた初期値から 3 世代程更新することで収束性の高い値を得られているが, そこからの値の更新が上手く行われておらず探索が止まったような状態となってしまう。

4. 改良 PSO による自動チューニング

〈4・1〉 改良 PSO の原理・特徴 従来の PSO では収束性は非常に高いものの, ある一定値以上になると探索が止まってしまう。それは式 (3) の α の値が大きいためである。だがしかし, α を小さくしてしまうとパラメータの更新幅が小さくなるため細かな調整は可能になるが, 最適値への収束を遅くしてしまう原因となってしまう。

そこで, 最初に定める集団を α の大きい集団, α の小さい

集団とに分けてグローバルな探索能力を有し, 最適解近傍での微調整にも対応できる手法を提案する。従来の PSO と比べて, 少し多めに集団数を設定しなければならないという欠点もあるが工作機械のパラメータチューニングでは集団数が増えても許容範囲内の試行回数で実現できるので問題ないと言える。

〈4・2〉 改良 PSO による実験・考察 実際に実験をして上記手法の有効性を検証してみる。まず実験条件として集団数 : 24, 世代数 : 16, $\alpha : 0.3, 0.6, 0.9$ とし, それぞれ集団数を 8 で分配する。その実験を 3 度行った。

その結果, 適合度, 形状誤差, 軌道生成にかかった時間の更新の特性を表すと 1 度目に行った実験の結果は図 5 のようになり, 改良法は少しずつ値を更新する集団と大幅に更新する項と分かれてグローバルな探索能力を有し, 最適解近傍での微調整にも対応できていると言える。改良法は従来法よりも収束性こそ悪いが, 最終結果は表 2 を見れば分かるように, 改良法の方が最適値に近い値を導き出している。

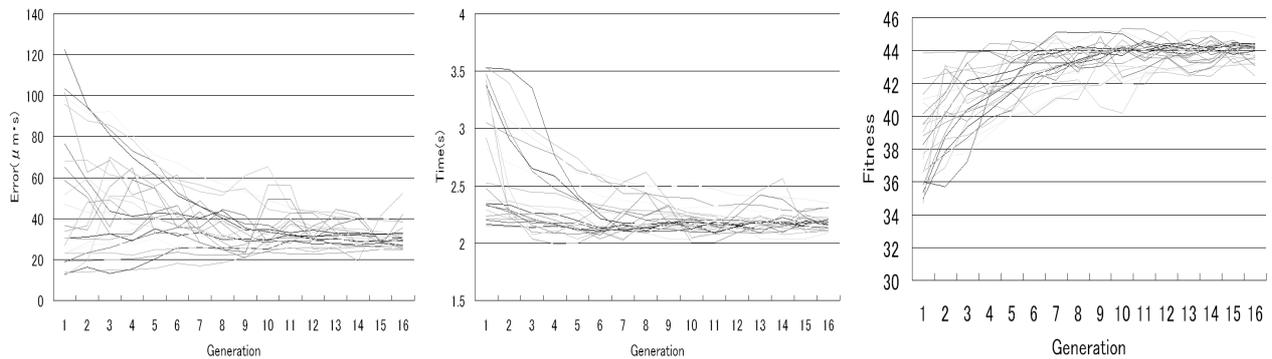


図 5 改良 PSO によるパラメータチューニング
Fig. 5. Tuning of control parameter using improved PSO

表 2 GA, PSO, 改良 PSO による調整結果
Table 2. Result of tuning using GA, PSO, improved PSO

	P1	P2	P3	P4	P5	Error($\mu \text{ m} \cdot \text{s}$)	Time(sec)	Fitness
GA	158	3639	-4948	1283	87	34.3097	2.117	44.1452
PSO 1st	232	3183	-4524	1310	159	26.2347	2.146	44.7414
PSO 2nd	280	3441	-4729	1262	163	28.0774	2.125	44.7101
PSO 3rd	185	2935	-5188	1206	196	29.6395	2.194	44.0512
Improved PSO 1st	228	3351	-4501	1216	139	29.8052	2.037	45.1785
Improved PSO 2nd	318	3596	-4032	1131	91	32.5374	2.041	44.9941
Improved PSO 3rd	270	3600	-4532	1230	169	32.8267	2.015	45.0366

5. PSO と NN(Neural Network) を組み合わせた自動チューニング

〈5・1〉 NN の学習システム NN は脳機能に見られるいくつかの特性を計算機上のシミュレーションによって表現することを目指した数学モデルである NN の特徴は、代数方程式で記述される静的モデルであること、非線形関数の写像能力を有すること、NN の学習法として誤差逆伝播法 (error back propagation algorithm) が広く採用されていることなどである。(5) (6)

NN が多くの分野で応用される第 1 の理由は、それが任意の写像や連続関数を近似できるからである。NN の関数近似能力については、近似定理として各種の数学的証明が与えられている。

誤差逆伝播法のアルゴリズムは、 m 層のネットワークを考え、 k 層の i 番目のユニットへの総入力を x_i^k 、このユニットの出力を y_i^k 、 $k-1$ 層の i 番目のユニットから k 層の j 番目のユニットへの結合係数を $w_{ij}^{k-1,k}$ 、ある入力に対する出力の値 (教師信号) を t_j 、学習係数を η と表記すると、以下の通りである。

- (1) 中間層と出力層の重みを乱数値によって初期化する
- (2) 入力パターン (合計 n 個) の各パターン p について、下記の処理を行う
 - i. パターン p を入力層に入力し、 $y_i^k = f(\sum_i w_{ij}^{k-1,k} y_i^{k-1})$ を計算機上でマッピングをし最適解を計算する。そうやって各ユニットの出力値を求める
 - ii. 教師信号と出力との 2 乗誤差 $E = \frac{1}{2} \sum_j (y_j^m -$

$t_j)^2$ を求める

iii. 出力層の重みの更新を $\Delta w_{ij}^{m-1,m} = -\eta \frac{\partial E}{\partial w_{ij}^{m-1,m}}$

を用いて行う

iv. 中間層以下の重みの更新を $\Delta w_{ij}^{k-1,k} = -\eta \delta_j^k y_i^{k-1}$ 、 $\delta_j^k = (\sum_i \delta_i^{k+1} w_{ji}^{k,k+1}) y_j^k (1 - y_j^k)$ を用いて行う

(3) E がある設定値以上である場合は、処理 2 に戻って繰り返す。設定値より小さい場合は終了

誤差逆伝播法の特徴は、入力パラメータと出力パラメータの相関関係が全く分からなかったとしても、教師信号をいくつか与えて学習を繰り返せば、どのようなモデルでも近似が可能という点である。

〈5・2〉 PSO に NN の学習能力を組み合わせた NC パラメータ最適化手法

〈3・2〉より PSO は最適解の近傍への収束性は良いが、最適解への収束性が悪いことが分かる。これは試行回数を少なくして最適解を見つけるのに不都合な性質である。しかし、それさえ解決できれば非常に有用なアルゴリズムとなる。そこで、PSO の問題改善法として NN との併用法を提案する。NN には非線形関数の写像能力があるので、入力パラメータが複数存在し、互いの関係性が明確でない今回の NC 工作機械の場合でも適用が可能であるので、パラメータ空間の同定ができる。同定ができれば、やっあって最適解に当たりをつけてパラメータ探索をすることで、PSO のみによるパラメータ探索よりもさらなる改善が

表 3 PSO,PSO+NN による調整結果
Table 3. Result of tuning using PSO, PSO+NN

	P1	P2	P3	P4	P5	Error(μ m · s)	Time(sec)	Fitness
PSO	232	3183	-4524	1310	159	26.2347	2.146	44.7414
PSO+NN	203	3312	-5124	1305	118	26.342	2.116	44.9492

できるものと思われる。

NC 工作機械のパラメータ探索への NN の適用法は,NC 工作機械のコントローラへの入力を \mathbf{P} (NC の調整可能なパラメータ),NC 工作機械の出力から計算された適合度を \mathbf{F} , 入力から出力への写像を f と表すとすると,

$$\mathbf{F} = f(\mathbf{P})$$

$$\mathbf{F} = (F_1, F_2, \dots)$$

$$\mathbf{P} = (P_1, P_2, \dots)$$

という形で表現されるが,この $f()$ を同定することができれば \mathbf{F} の最適解を計算機上で求めることができる。この同定手段として NN を使用する。前述の通り,NN は非線形関数の写像能力があるので問題無く使用できる。

〈5・3〉 PSO に NN の学習能力を組み合わせた手法による実験・考察 実際実験をして提案手法の有効性を検証してみる。実験条件として〈3・2〉で行った PSO の結果を用い,そのなかの適合度 43.0 以上のデータ 80 個を NN の教師信号として使用する。そして学習係数 η を 0.01 から 0.40 まで 0.01 刻みの 40 通りの最適解を求める。そのほかの条件として学習回数は 1 万回,中間層数は 50 とし,計算機上での最適解探索には PSO を用いた。

その結果求められた最適解は表 3 のようになり,PSO 単体による結果と大差ないものとなり,そして改良 PSO よりも悪い結果となってしまった。多次元のモデルの推定となるのに教師信号数が 80 と少なめであったのでモデリング精度が欠けてしまったというのが主な原因であろうが,NN の計算時間を考慮すると,このことを問題解決の糸口とするのはあまり賢明なことではない。その他の条件を見直してみる必要がある。

6. 結 論

本論文では,従来の多目的最適化手法である GA,PSO による工作機械のパラメータ自動チューニングを行い,GA には収束性がないこと,PSO には収束性は高いが最適解近傍での探索に難があることを示した。そして,それらの問題を解決するために,PSO の速度更新式をグローバルな探索能力を有するが微調整のできない集団,収束は遅いが最適解近傍での微調整が可能な集団とに分けるように変更し,ハイブリッドな探索を行える手法と,NN の学習能力に着目し,PSO でもって最適値のおおよその目安をつけ NN でその近辺を計算機上でモデル化し,最適値を計算機上で求める手法を提案した。

改良 PSO は実際実験を行った結果,従来の PSO よりも最適解に近い値を探索できており有効性を実証できたが,NN

を利用した PSO はほとんど性能向上が行われず前者の提案手法よりも悪い結果が得られてしまった。ただし,NN の設定条件が悪かった可能性もあるので,今後はそれらを見直すと同時に改良 PSO の速度更新式を世代毎に変化させるような手法を検討する予定である。

参考文献

- (1) 平野 廣美, "応用事例でわかる遺伝的アルゴリズムプログラミング", パーソナルメディア株式会社,2001 年.
- (2) Bunghoon Chang, Yoichi Hori, "Research related to the Parameter Auto-tuning of Two Mass Control System", in IEE of Japan Technical Meeting Record, IIC-05-47, 2005 年.
- (3) Sehoon Oh, Yoichi Hori, "Control Parameter Optimization in the Hardware-in-the-loop System using Novel Search Algorithm",IEEE-IECON 2006
- (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/情報第 50 巻第 9 号, システム制御情報学会,2006 年.
- (5) 志水 清孝, "ニューラルネットと制御", コロナ社,2002 年.
- (6) 電気学会 GA 等組合せ最適化手法応用調査専門委員会編, "遺伝的アルゴリズムとニューラルネット-スケジューリングと組合せ最適化-", コロナ社,1999 年.