

- 卒業論文 -

工作機械の精密制御のための
最適パラメータ探索法

平成19年2月13日

指導教員：堀 洋一 教授

東京大学工学部電子工学科

50420 立田 昌也

内容梗概

現在 NC 装置を用いた工作機械が世に数多く流通されている。NC 装置を用いることで工作機械の高速かつ高精度な制御を実現することができるのだが、そのために数多くのパラメータを調整する必要があり、調整方法も複雑なため一般人には非常に困難なものとなっている。

そこで本論文では、その調整を工作機械が自動で行えるようにするためのパラメータ探索アルゴリズムを提案し、人にかかる労力を低減し工作機械をさらに便利なものとすることを目的とした。

まず初めに人の手での調整を行い各パラメータの最適値を定め、次に従来行われてきた GA、PSO によるパラメータ最適化を行った。そこで生じた問題点を明らかにし、これらを克服するために PSO に改良を加えた手法と PSO に NN を組み合わせた手法を提案する。それらを実験することで理論通りの動作ができるかを確かめ、有効性があるのかどうかを示した。

目次

第 1 章	研究の背景と目的	3
1.1	NC 装置の特徴	3
1.2	工作機械に残された問題	4
1.3	本論文の構成	4
第 2 章	実験機の構成について	5
2.1	NC 工作機械を取り入れた HIL(Hardware-In-the-Loop) システムについて	5
2.2	実験機のシステム	5
第 3 章	実験機の人の手によるパラメータ最適化	6
3.1	パラメータ調整方法	6
3.2	FF タイミング調整係数の設定	6
3.3	速度 FF ゲインの設定	8
3.4	補間前加減速、円弧の速度の設定	9
3.5	人の手によるパラメータ調整結果	11
第 4 章	従来の最適化手法を利用した NC パラメータ最適化	14
4.1	自動チューニングの特徴・留意点	14
4.2	自動チューニングの際の実験方法	14
4.3	GA(Genetic Algorithm) を利用した NC パラメータ最適化実験	15
4.4	PSO(Particle Swarm Optimization) を利用した NC パラメータ最適化実験	16
4.5	GA、PSO の比較・考察	19
第 5 章	NC パラメータ最適化に適した新たな最適化手法の提案と検討	20
5.1	改良 PSO による NC パラメータ最適化	20
5.1.1	改良 PSO の原理・特徴	20
5.1.2	改良 PSO による実験・考察	20
5.2	NN(ニューラルネットワーク) を利用した NC パラメータ最適化	23
5.2.1	NN の学習システム	23
5.2.2	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	23
5.2.3	PSO に NN の学習能力を組み合わせた手法による実験・考察	24
第 6 章	本論文のまとめと今後の課題	26
6.1	本論文のまとめ	26
6.2	今後の課題	26
	参考文献	27

第1章 研究の背景と目的

1.1 NC装置の特徴

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。昔は人の手で行われた作業も徐々に機械が代行するようになり、工作機械の高品質・低価格化もあいまって、その飛躍ぶりは留まる所を知らない。そして、工作機械の高品質化に大きく貢献したものがNC(Numerical Control)装置である。外観の一例として図1.1のようなものがある。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、一般にNCともいわれ、そのNC機能はコントロールプログラムと呼ばれるソフトウェアによって規定される。NCの技術開発は高速・高精度・高効率の加工を目標に進められている。

最近のNC装置では、高速の64ビットRISCプロセッサを使用し微小線分が連続する高速輪郭切削での加工形状誤差を大幅に低減できる。また、インポリュート補間などの補間機能やカスタマイズのための各種の指令言語、自己診断機能、高速・高精度に必要なサーボシステムなどの機能が付けられている。NC装置を導入することで、同じ製品を複数作成する際に、同じ寸法のものと同じ作業時間で、大量に作ることができる。人手による作業では、作業者のスキルレベルによって精度や作業時間に大きなばらつきが生じやすく、品質管理が難しいが、それらの問題点を解消することができる。



図 1.1: NC 装置

1.2 工作機械に残された問題

上記のような特徴を持つ NC 装置は製品製造の低コスト・高品質化には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、大抵の工作機械はパラメータ調整をしなければならないからである。現在は顧客に物を売った際、その製品に対する必要なパラメータ設定のため、その製品に関する専門家を出張などの形で直接派遣し、必要なパラメータを調整するサービスを行うのが一般的である。だが人の手によるパラメータチューニングには限界がある。それというのも、調整しなければならないパラメータは1つだけではない。速度ループゲイン、電流ループゲイン、ポジションゲイン、負荷イナーシャ比、外乱除去フィルタゲインなど 10、20 のパラメータを調整するのは当たり前で、場合によっては 100 を超えるパラメータを調整しなければならない。そうなるともはや人の手でパラメータを調整するのは不可能であり、そういった状況下では高性能化が著しい現代の工作機械の性能を活かしきることができない。そして、たとえ人間の手で調整できたとしても、それまでに莫大な時間がかかってしまうし、調整した結果が最適解である保証はどこにもない。

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。

本研究では、工作機械が高速高精度で動作できるようなパラメータを効率良く探索するアルゴリズムを提案することで、人の手にかかる労力・作業時間を削減し、工作機械をさらに便利なものにすることを目的とする。

1.3 本論文の構成

本論文では、工作機械のパラメータを自動でチューニングするに当たって様々な最適解探索法を用いた結果生じた問題を考察し、工作機械の自動チューニングに適した探索アルゴリズムに改良することを目的としている。

まず第2章では本論文で用いる実験機のシステムについて説明し、世の中に多く流通している工作機械は複雑な構成をしていることを理解する。第3章では実際に実験機を手動で調整することで、各パラメータの最適値の目安と人の手による調整の煩雑さを知る。第4章では従来最適化手法である GA、PSO を用いた自動チューニングをし、そこで生じた問題点を考察する。そして第5章では、第4章で生じた問題点を改善するために改良した新たな最適化手法を提案し、実際に実験をしてその実用性を確認する。

第2章 実験機の構成について

2.1 NC工作機械を取り入れたHIL(Hardware-In-the-Loop)システムについて

HIL(Hardware-In-the-Loop)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る。今回使用するNC工作機械もHILシステムに組み込まれており、図2.1のように構成されている。

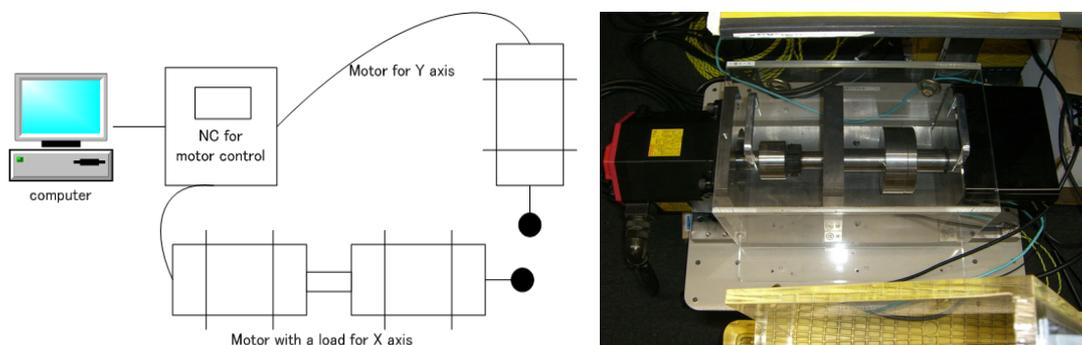


図 2.1: 実験装置の HIL システム, 外観

2.2 実験機のシステム

本実験で用いる機器のシステム全体のブロック線図は図2.2のようにになっている。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。

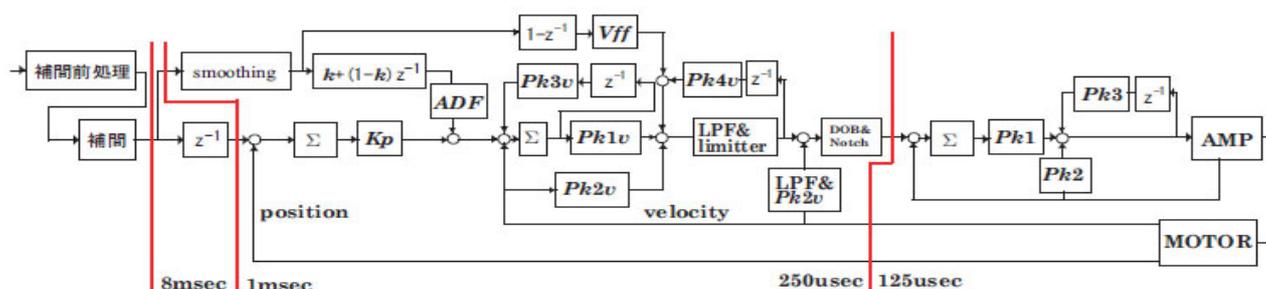


図 2.2: システム全体のブロック線図

第3章 実験機の人の手によるパラメータ最適化

3.1 パラメータ調整方法

パラメータ自動チューニングを行う前に、手動で実験機を調整して最適値の目安をつけないといけない。まず、調整するパラメータは以下の5つである。

- 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の中でリスケールしているため、値それ自体は特に意味を持たない

そして、調整パラメータがそれぞれ特性が異なるため以下のように多くの段階を踏んで調整をしなければならない。

1. フィードバック系の調整 (適正な共振回避フィルタを入れた上で、速度ゲインをできるだけ高く設定する。速度ゲインに合わせて位置ループゲインを設定する)
2. フィードフォワードを有効にし、円弧形状を見ながら FF タイミング調整係数 (P3) を設定
3. Rのない四角の形状で、停止する軸のオーバーシュート量を見ながら速度 FF ゲイン (P4,5) を各軸調整
4. 再度円弧形状を見て、FF タイミング調整係数 (P3) を調整 (3 の設定が4 の最適値に影響を与えるため)
5. R付きの円弧で、補間前加減速 (P1)、円弧の速度 (P2) を輪郭形状誤差とのトレードオフで決定 (接線方向加速度は、法線方向加速度の1.5~2倍程度を目安として、一緒に決める)
6. 3 の速度 FF ゲイン (P4,5) を再調整 (補間前加減速 (P1) の影響を多少受けるため)

3.2 FF タイミング調整係数の設定

FF タイミング調整係数 (P3) とはFFのタイミングをずらすパラメータである。4096 と入れると1ms分時間的に前に進み、-4096を入れると1ms分時間的に遅れる。そしてFF

が進むほど、円弧の半径は減少する。

このパラメータの調整は円弧形状を見て形状誤差が最も小さくなるように設定をする。円弧の半径は10mmとし、FF タイミング調整係数は-8192 から 4096 の間で、その他のパラメータは補間前加減速:280、円弧の速度:3000、速度 FF ゲイン (x 軸):800、速度 FF ゲイン (y 軸):130 で調整を行った。FF タイミング調整係数と形状誤差の関係は図 3.1 のようになり-4000 程度で形状誤差が最小になっていることが分かる。実際の軌道は精密制御を心掛けた調整で μm 単位の誤差となってしまうので一部拡大した結果を図 3.2 に示した。図 3.1 での結果の通り、-4000 での実軌道は内回り過ぎず外回り過ぎずの軌道を実現できている。

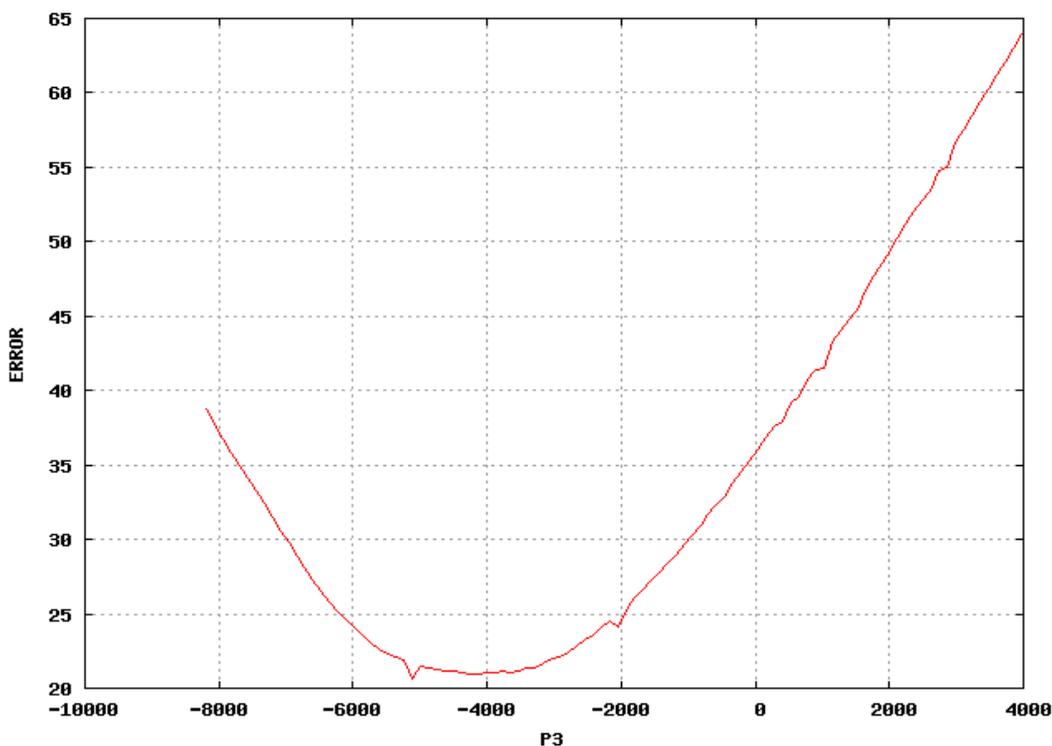


図 3.1: FF タイミング調整係数 (P3) と形状誤差の特性

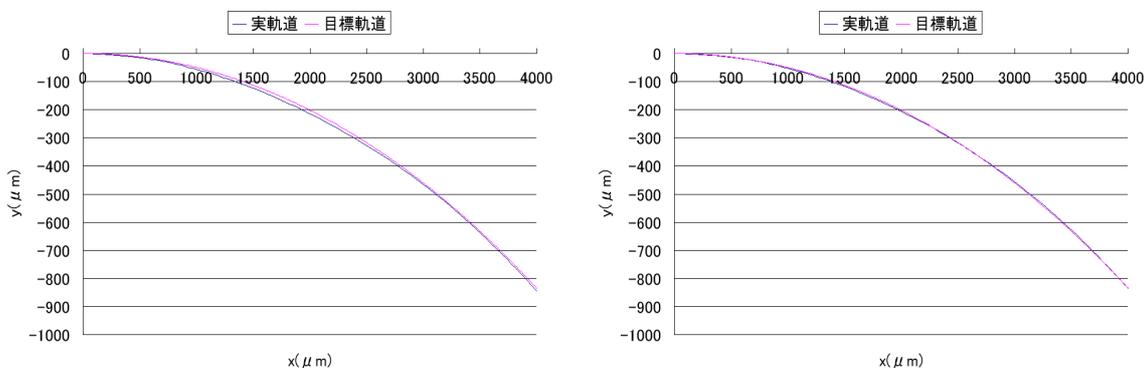


図 3.2: 軌跡拡大 (左:非最適値 4096、右:最適値 - 4000)

3.3 速度 FF ゲインの設定

速度 FF ゲイン (x 軸:P4, y 軸:P5) とは指令を加速度の次元に変換したものに、この係数を掛けてトルク指令に加算するものである。コーナ部での形状としては、不足するとオーバーシュート、過剰だとアンダーシュートという傾向になる。

このパラメータの調整は四角の形状を見て形状誤差が最も小さくなるように調整する。四角は辺長が 30mm の正方形を用いて、速度 FF ゲインは x 軸と y 軸と分けてそれぞれ 200 から 1500、50 から 700 の間で、その他のパラメータは補間前加減速:280、円弧の速度:3000、FF タイミング調整係数:- 4000 で調整を行った。速度 FF ゲインと形状誤差の関係を示した結果は図 3.3 のようになり、 x 軸は 1175、 y 軸は 50 で最適な結果を得た。そして実際の軌跡の一部を拡大すると図 3.4 のようになった。

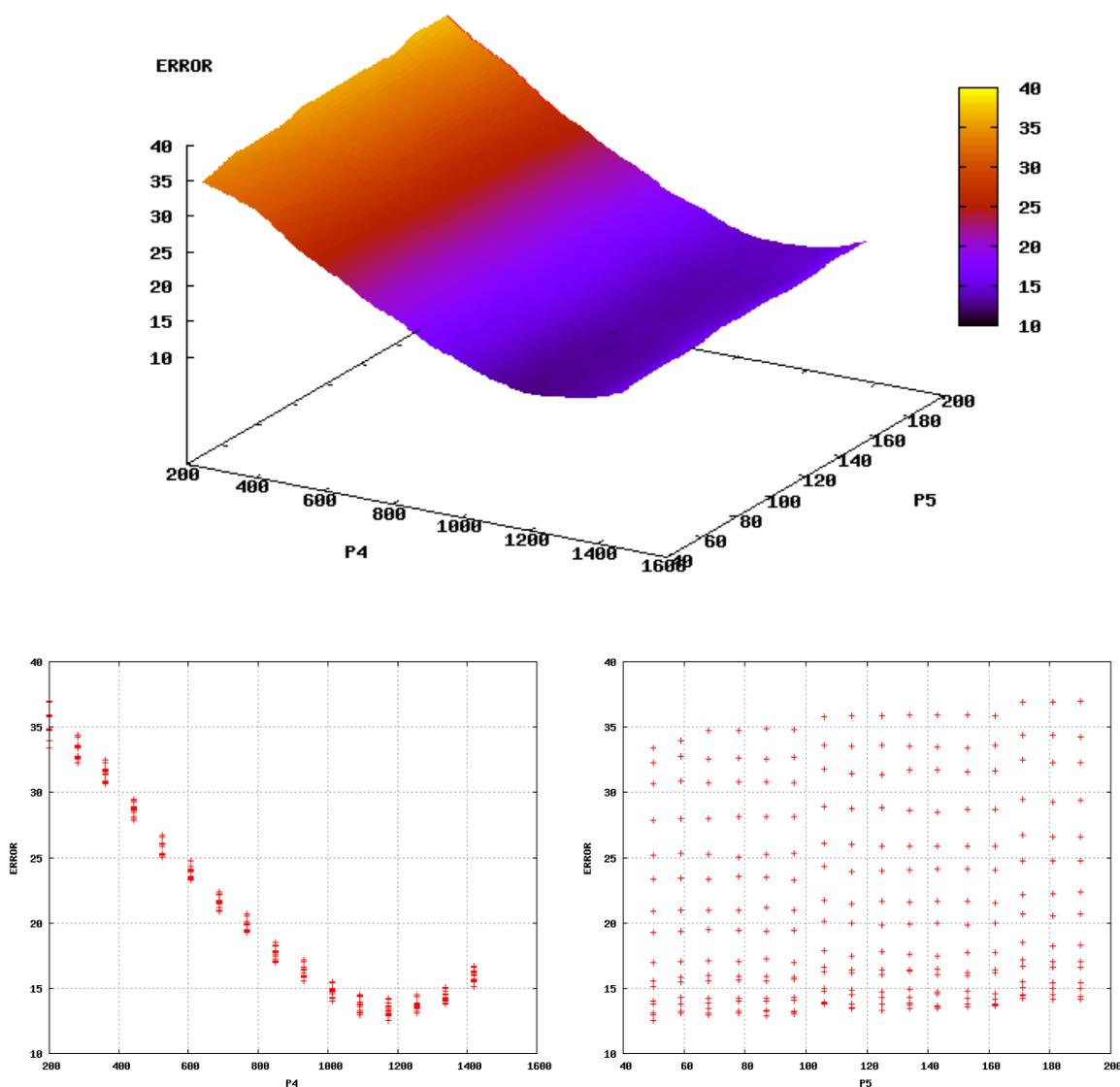


図 3.3: 速度 FF ゲイン (P4,P5) と形状誤差の特性

速度 FF ゲインを調整した後は再度円弧形状で FF タイミング調整係数の調整を行う。これは速度 FF ゲインが FF タイミング調整係数の調整に影響を与える為である。実験条件は

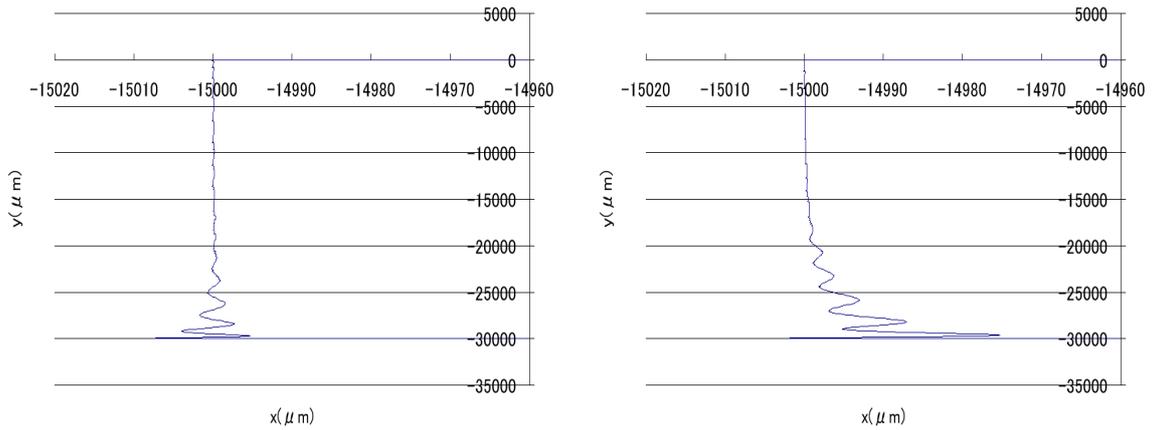


図 3.4: 軌跡拡大 (左:最適値、右:非最適値)

最初と同様に円弧の半径は10mmとし、FF タイミング調整係数は-8192 から 4096 の間で、その他のパラメータは補間前加減速:280、円弧の速度:3000、速度 FF ゲイン (x 軸):1175、速度 FF ゲイン (y 軸):50 で調整を行った。FF タイミング調整係数と形状誤差の関係を示した結果は図 3.5 のようになり、 -4500 程度で最適値を得た。

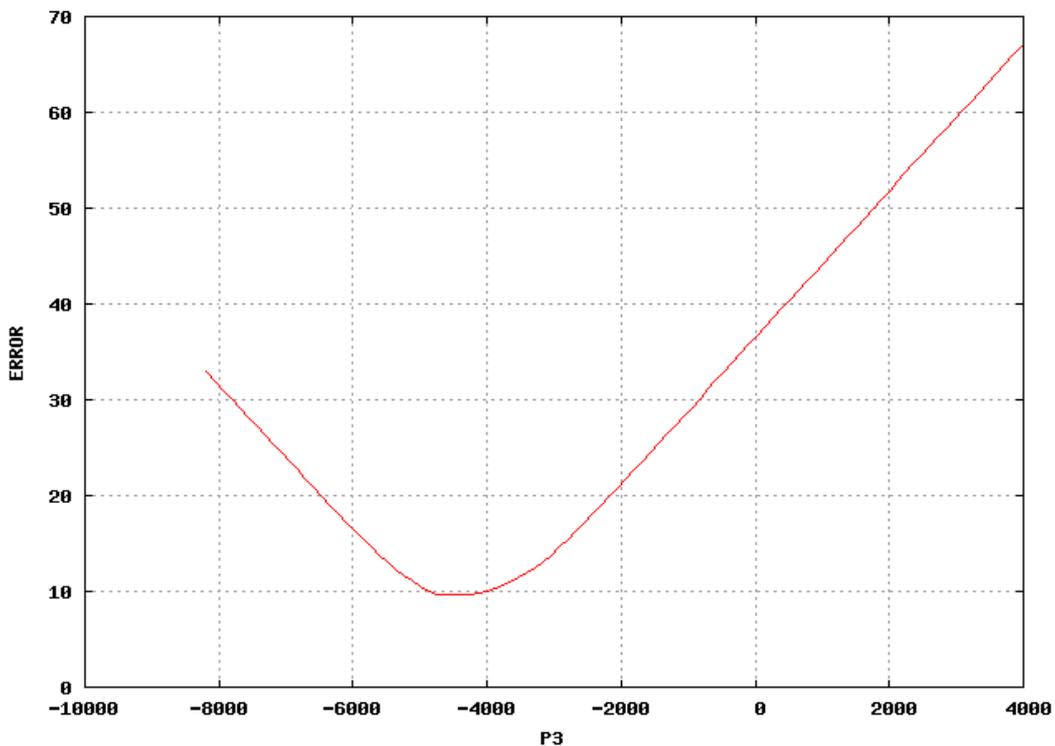


図 3.5: FF タイミング調整係数 (P3) と形状誤差の特性

3.4 補間前加減速、円弧の速度の設定

補間前加減速 (P1) とは接線方向の加速度を決めるパラメータである。これは例えば、直線部分から円弧部分の軌道を描く際、形状誤差を小さくするために速度を円弧の許容速

度まで減速させなければならないのだが、その減速の為の加速度を決めるパラメータとなる。今回調整するのは決まった速度を加速・減速させるのに要する時間のパラメータとしたので、この値が大きいほど緩やかな加減速となる。

円弧の速度 (P2) とは円弧部分での許容速度のことである。接線方向加速度と法線方向加速度が、直線⇒円弧⇒直線の繋ぎ目で変化するため、形状誤差が発生し、コーナ部では指令或いはサーボ系の遅れに起因する形状誤差（内回り誤差）が発生する。この誤差は、補間後の時定数（直線）を T_1 とするとこの処理は入力に対して過去 T_1 だけ平均した処理に相当する。即ち入力を $u(t)$ 、出力を $y(t)$ とすると

$$y(t) = \frac{1}{T_1} \int_0^{T_1} u(t - \tau) d\tau$$

$$Y(s) = \frac{1}{sT_1} (1 - e^{-sT_1}) U(s)$$

より伝達関数 $H(s)$ は

$$H(s) = \frac{1}{sT_1} (1 - e^{-sT_1})$$

と表すことができる。

今、円弧の指令 $u(t) = Re^{j\omega t}$ ($\omega = \frac{v}{R}$) を考えると

$$H(s) = H(j\omega) = \frac{1}{j\omega T_1} (1 - e^{-j\omega T_1})$$

ここで、 $e^{-j\omega T_1} = 1 - j\omega T_1 + \frac{1}{2}(-j\omega T_1)^2 - \frac{1}{6}(j\omega T_1)^3$ と近似すると

$$H(j\omega) \simeq 1 - j\frac{1}{2}\omega^2 T_1^2 - \frac{1}{6}\omega^2 T_1^2$$

となる。このゲインは

$$\begin{aligned} |H(j\omega)| &= \left| \left(1 - \frac{1}{6}\omega^2 T_1^2 \right) - j\frac{1}{2}\omega^2 T_1^2 \right|^{\frac{1}{2}} \\ &= \left| \left(1 - \frac{1}{6}\omega^2 T_1^2 \right)^2 + \frac{1}{4}\omega^2 T_1^2 \right|^{\frac{1}{2}} \\ &\simeq \left| 1 - \frac{1}{12}\omega^2 T_1^2 \right|^{\frac{1}{2}} \\ &\simeq 1 - \frac{1}{24}\omega^2 T_1^2 \quad (\text{いずれも}\omega T_1\text{の4次以上を無視}) \end{aligned}$$

よって半径は $R(1 - \frac{1}{24}\omega^2 T_1^2)$ となり、形状誤差となる半径減少 ΔR は

$$\Delta R = \frac{R}{24}\omega^2 T_1^2 = \frac{1}{24} \frac{v^2}{R} T_1^2$$

となる。

このパラメータの調整は4隅が円弧の四角の形状を見て形状誤差と軌道達成にかかった時間がトレードオフの関係になるが、これらを考慮した適合度という形状誤差と軌道達成にかかった時間が小さければ大きな値を示す関数にて調整する。算出式は第4章の式(4.1)のように定めた。形状は辺長が30mmの正方形の4隅を半径5mmの円弧に変更し、補間前加減速は50から550、円弧の速度は300から4000の間で、FFタイミング調整係数:-

4500、速度 FF ゲイン (x 軸):1175、速度 FF ゲイン (y 軸):50 で調整を行った。補間前加減速、円弧の速度と適合度の関係を示した結果は図 3.6、形状誤差との関係を示した結果は図 3.7、軌道達成にかかった時間との関係を示した結果は図 3.8 のようになり、円弧の速度が上がっても形状誤差があまり増加しないため、最適値は補間前加減速:237、円弧の速度:3768 のように得られた。

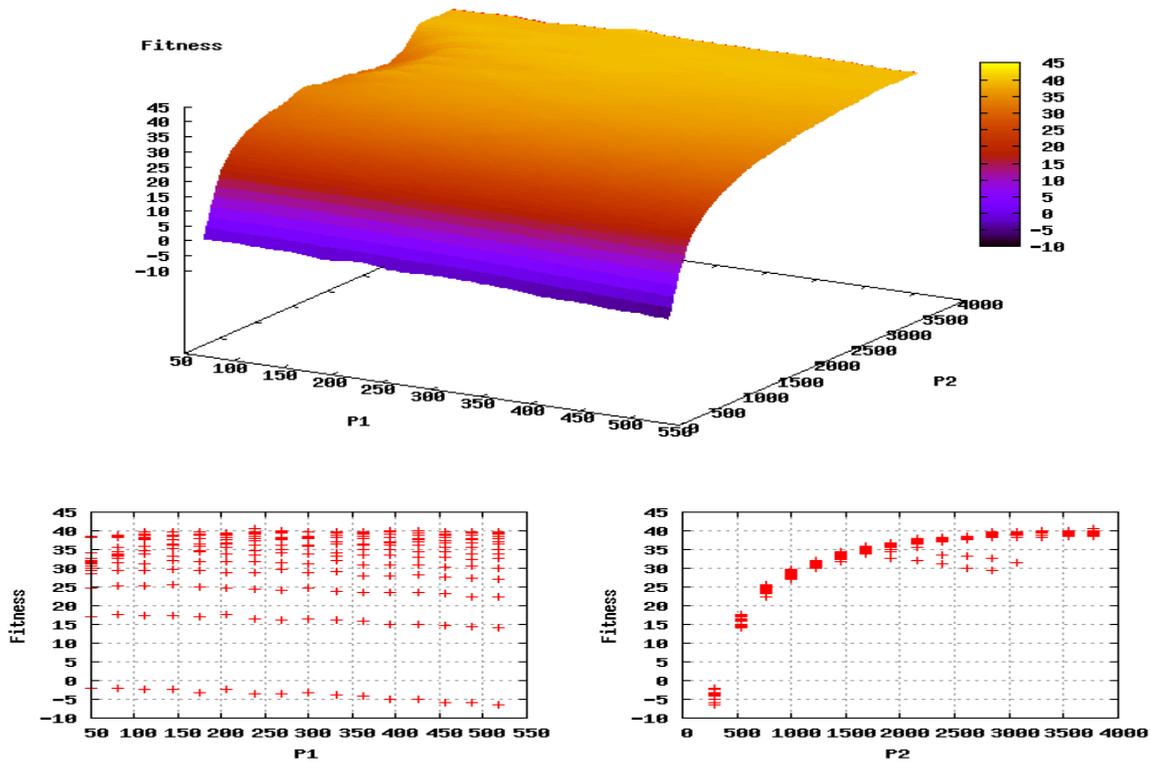


図 3.6: 補間前加減速 (P1)、円弧の速度 (P2) と適合度の特性

最後に、再度速度 FF ゲインの調整を行う。補間前加減速の調整で多少影響を受けるからである。調整方法は初回と同様に、辺長が 30mm の正方形を用いて、速度 FF ゲインは x 軸と y 軸と分けてそれぞれ 200 から 1500、50 から 700 の間で、その他のパラメータは補間前加減速:237、円弧の速度:3768、FF タイミング調整係数:- 4500 で調整を行った。速度 FF ゲインと形状誤差の関係を示した結果は図 3.9 のようになり、補間前加減速の影響はあまり受けず最適値は変わらなかった。

3.5 人の手によるパラメータ調整結果

以上の結果より、人の手による調整結果は表 3.1 のようになり、次章以降ではこの値に近づくことを目標に自動チューニングの実験を行っていく。

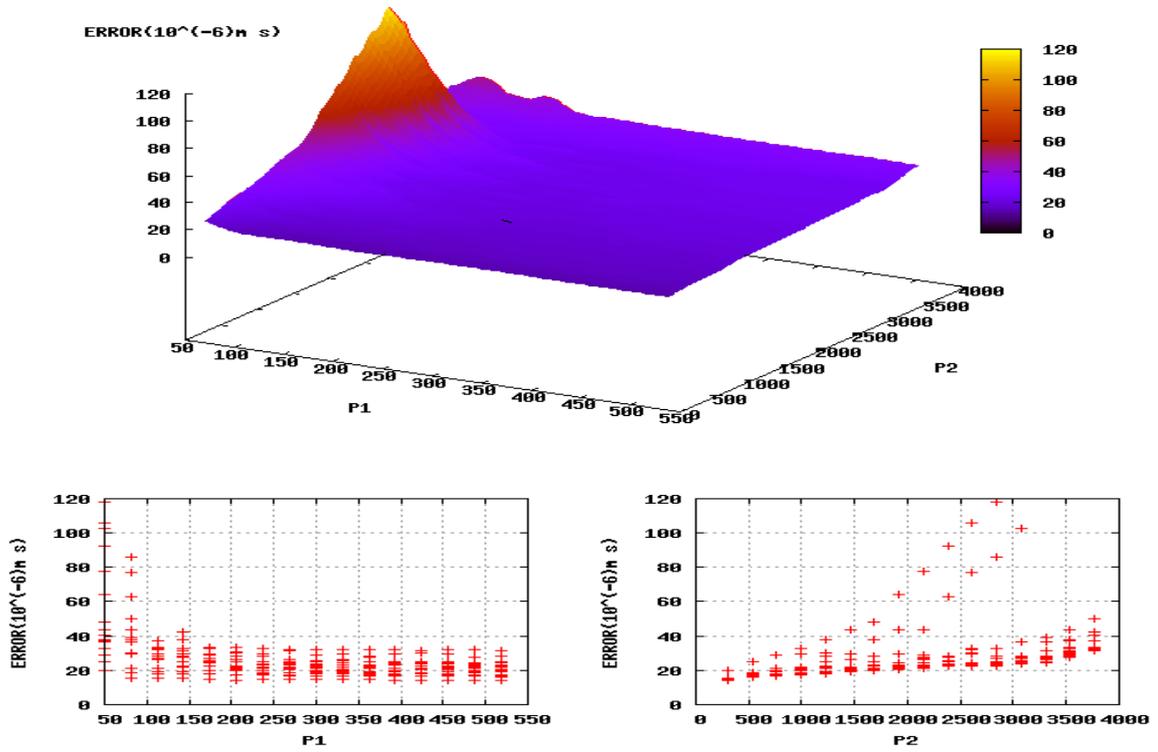


図 3.7: 補間前加減速 (P1)、円弧の速度 (P2) と形状誤差の特性

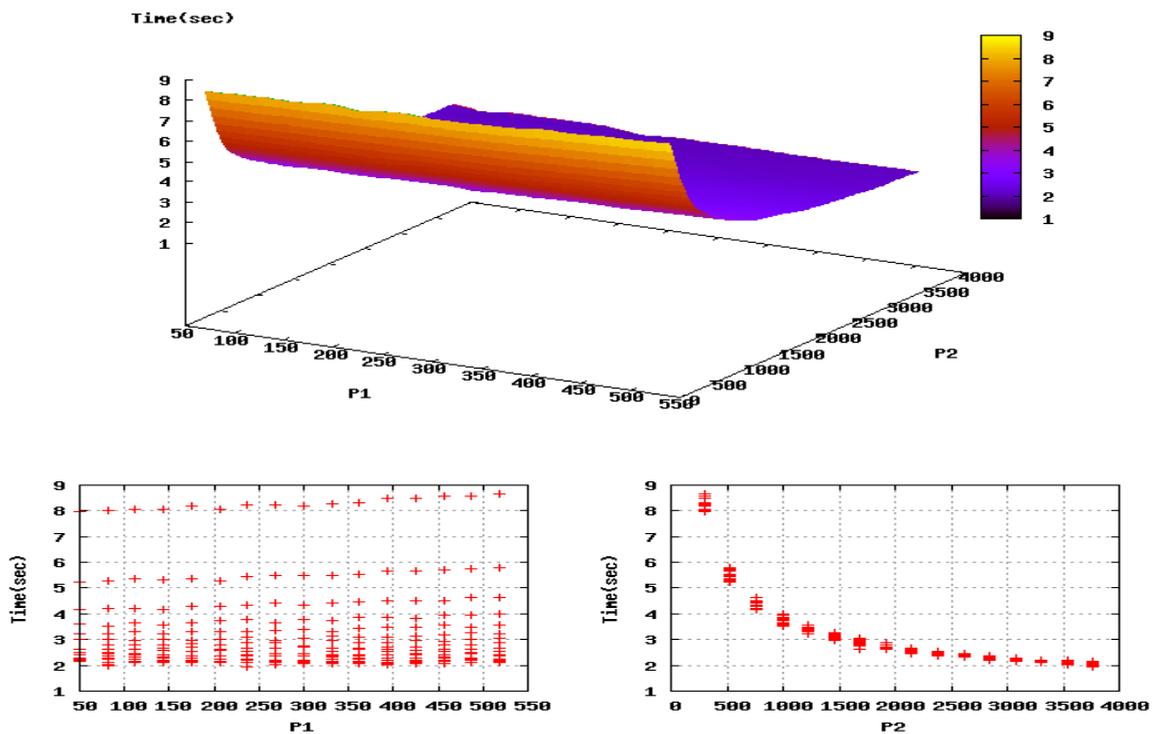


図 3.8: 補間前加減速 (P1)、円弧の速度 (P2) と軌道達成にかかった時間の特性

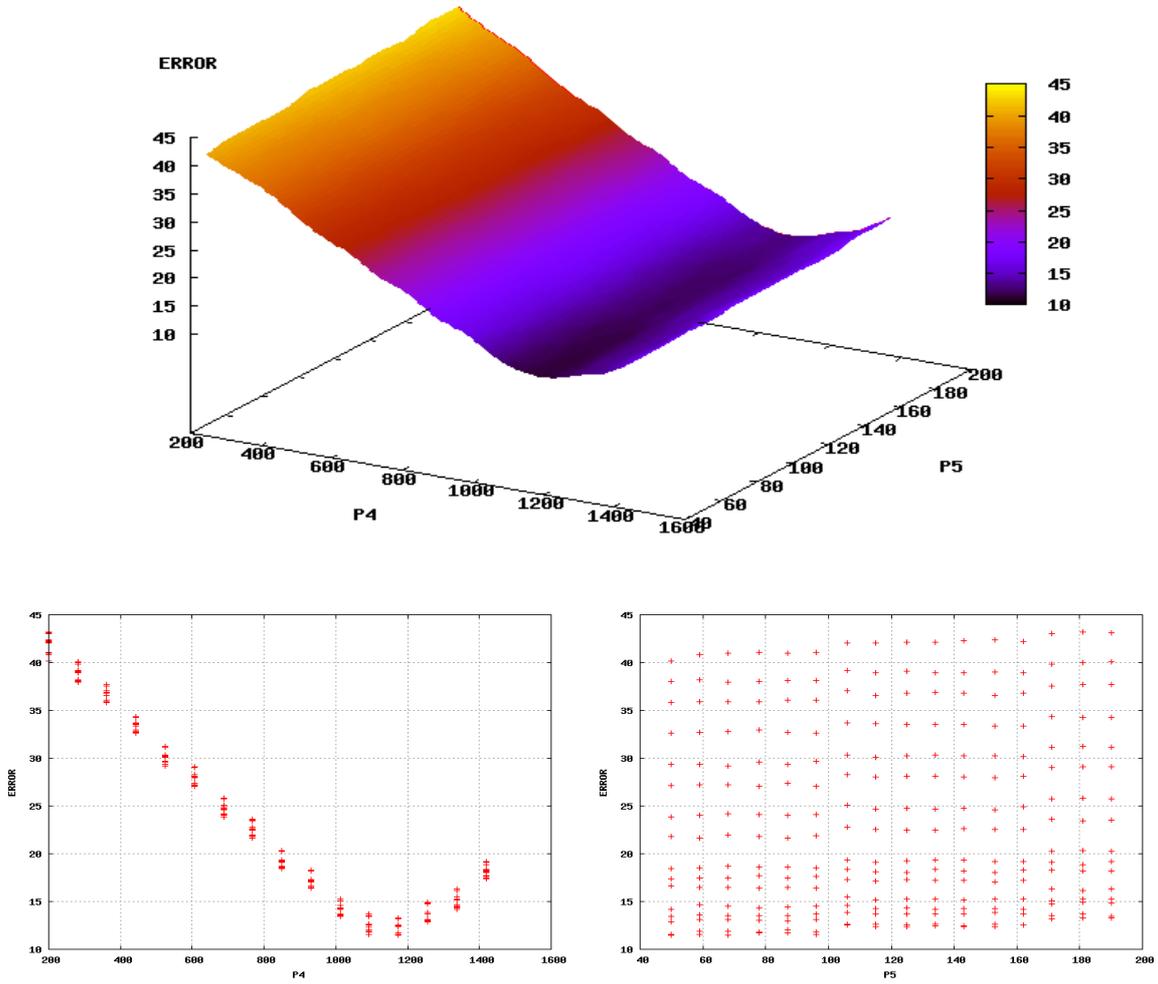


図 3.9: 速度 FF ゲイン (P4P5) と形状誤差の特性

表 3.1: 人の手による調整結果

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

第4章 従来の最適化手法を利用したNC パラメータ最適化

4.1 自動チューニングの特徴・留意点

ここで、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。

4.2 自動チューニングの際の実験方法

自動チューニングの際の実験方法であるが、3.1で説明した5つのパラメータの最適化を行う。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 (4.1) で算出した適合度 (Fitness) をもって行う。

$$Fitness = S_e(e_{bias} - \sum trajectory\ error(t)) + S_t(t_{bias} - Elapsed\ time) \quad (4.1)$$

- 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

4.3 GA (Genetic Algorithm) を利用した NC パラメータ最適化実験

遺伝的アルゴリズム (Genetic Algorithm, GA) とは、1975 年にミシガン大学のジョン・ホランド (John Holland) によって提案された近似解を探索するアルゴリズムの一つであり、データ (解の候補) を遺伝子で表現した「個体」を複数用意し、適応度の高い個体を優先的に選択して交叉・突然変異などの操作を繰り返しながら解を探索する。

この手法の利点は評価関数の可微分性や単峰性などの知識がない場合であっても、適用可能なことである。必要とされる条件は評価関数の全順序性と、探索空間が位相 (トポロジー) を持っていることである。

また遺伝子の表現の仕方によっては組合せ最適化問題や NP 困難な問題などのさまざまな問題に適用可能である。[1][2][3]

アウトラインは図 4.1 のようになる。

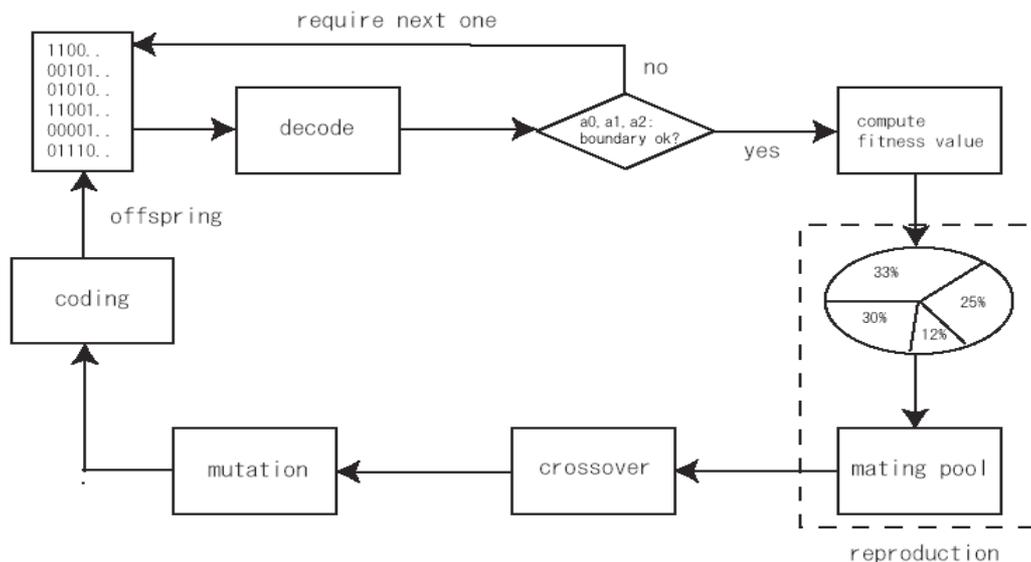


図 4.1: GA のアルゴリズム

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。[4]

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。集団の数を 10、世代の数を 10、交叉率を 0.9、突然変異率を 0.1 として 7 度実験を行った。結果は図 4.2, 4.3 のようになり、図 4.2 には GA による最適化を一度行った際の、各パラメータの値の変動と各世代の適合度、形状誤差と軌道生成にかかった時間の各個体の更新結果を、図 4.3 には各世代の適合度の最高値の推移を表示している。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、各パラメータの結果を見ても分かるように、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。

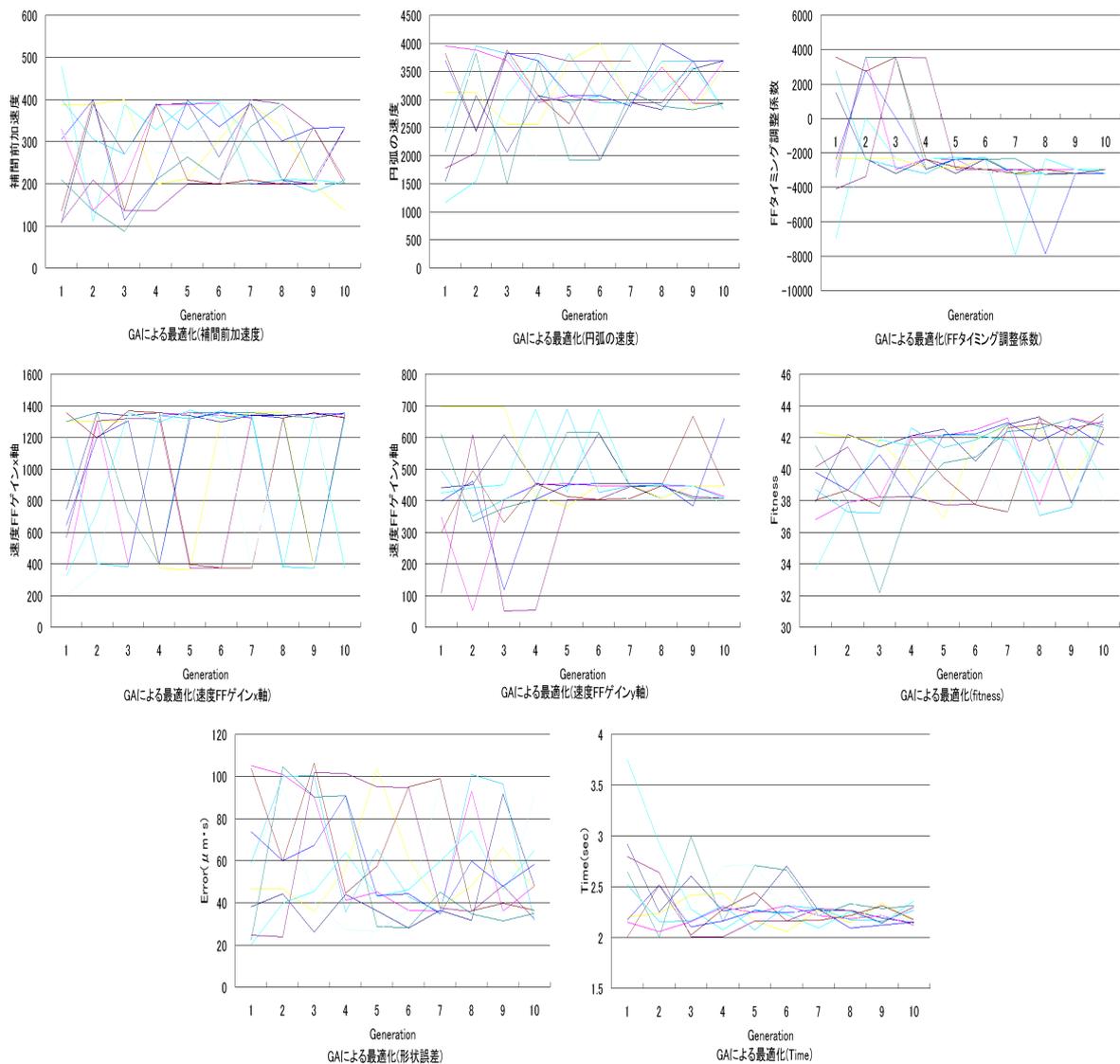


図 4.2: GA によるパラメータ最適化実験結果

4.4 PSO(Particle Swarm Optimization) を利用したNC パラメータ最適化実験

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

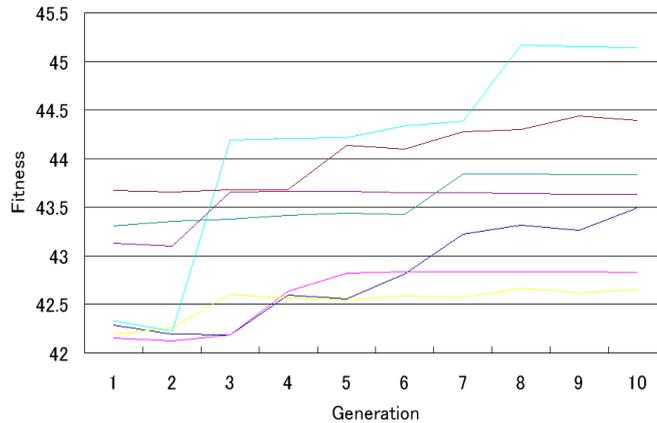


図 4.3: GA による各世代の適合度最大値の推移

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 (4.2),(4.3) で行われる。

$$x_i(k+1) = x_i(k) + v_i(k+1) \quad (4.2)$$

$$v_i(k+1) = \alpha \{v_i(k) + \beta_1 rand_1(x_{pbest_i} - x_i(k)) + \beta_2 rand_2(x_{gbest} - x_i(k))\} \quad (4.3)$$

世代が変わるときに、PSO は式 (4.3) 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 (4.3) 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。図にすると図 4.4 のようになる。[5][6]

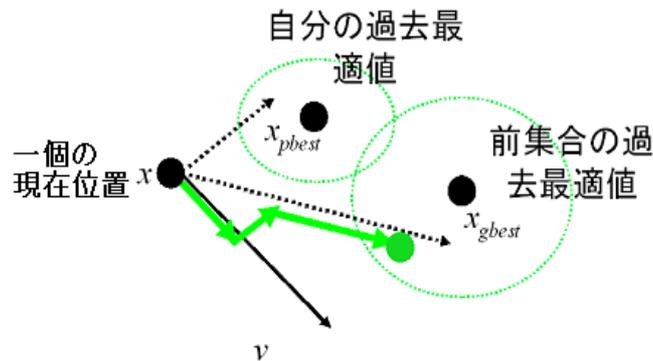


図 4.4: PSO におけるパラメータ更新

実際に実験をして PSO の検証を試みる。 $\alpha = 0.8, \beta_1 = 1.0, \beta_2 = 1.0$ 、集団の数 10、世代数 10 とし 8 回実験を行った。結果は図 4.5, 4.6 のようになり、GA の時と同様に図 4.5 には各パラメータの値の変動と各世代の適合度、形状誤差と軌道生成にかかった時間の更新結果を、図 4.6 には各世代の適合度の最高値の推移を表示している。図 4.5 の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いる。図 4.6 をみても分かるように時々最適値に非常に近い値を導き出してはいるが、それ以上に探索が途中で停滞してしまっている可能性の方が高い。そして、最適値に非常に近い値を導き出した結果も 1 世代更新することでかなりの伸び幅を示しており、これは偶然見つかったという可能性も否めない。

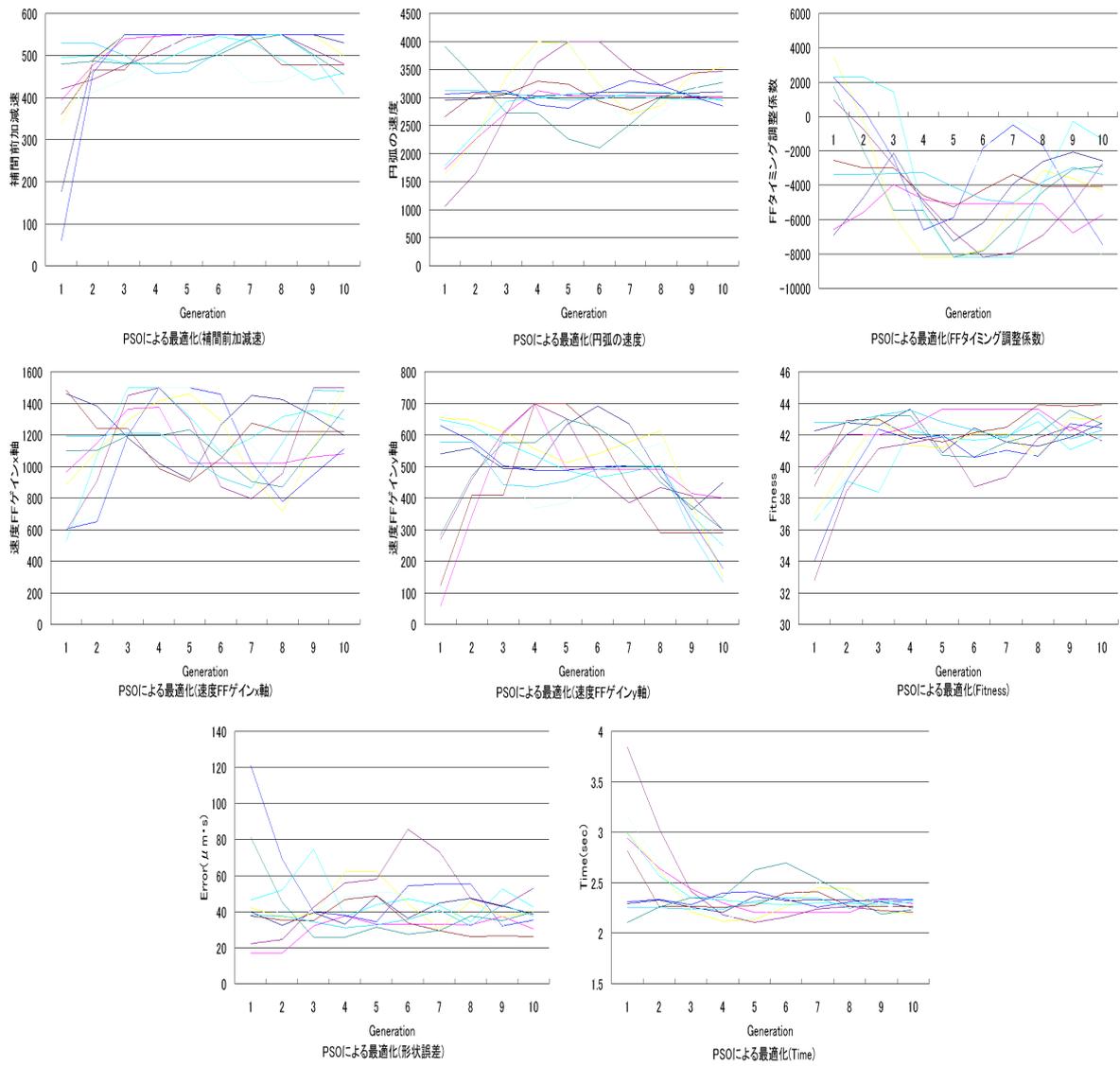


図 4.5: PSO によるパラメータ最適化実験結果

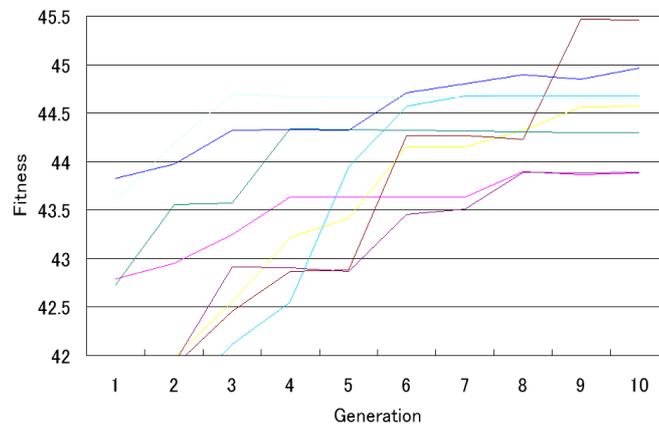


図 4.6: PSO による各世代の適合度最大値の推移

4.5 GA、PSOの比較・考察

GA、PSOによるパラメータ最適化の結果、表4.1,4.2のような結果を得た。適合度に関しては人の手による調整と比べるとかなりの隔りがある。試行回数を増やせばある程度は性能向上するだろうが、パラメータの推移の結果を考慮するとその可能性も低いと思われる。

それというのも、GAは図4.2からも分かるように最適解に収束していくという保証がどこにもない。確率に依存したアルゴリズムであるので、HILシステムにおけるパラメータチューニングには不向きである。

PSOは図4.5からも分かるように初期値がどのような値であっても最適値の方へ収束していつているのが分かる。これはパラメータチューニングにとって非常に有用な性質であり、大いに活用ができる。しかし、パラメータによっては局所解に陥り、値の更新が止まってしまい最適解への探索ができなくなってしまう。

次章ではこれらの問題点を克服できるような新たな最適化手法の提案と検討をする。

表 4.1: GAによる調整結果

	P1	P2	P3	P4	P5	Error(μ m · s)	Time(sec)	Fitness
GA 1st	209	2818	-3207	1325	407	31.4527	2.267	43.338
GA 2nd	501	2451	-5681	1219	110	18.9456	2.511	42.811
GA 3rd	77	3131	-1381	1174	180	47.5684	2.124	42.7683
GA 4th	228	3648	-5861	1197	55	33.8539	1.989	45.1233
GA 5th	291	2348	-5369	1219	140	19.6561	2.392	43.607
GA 6th	191	2519	-5465	1261	138	20.7543	2.276	44.3423
GA 7th	468	2923	-5561	1151	497	26.7376	2.265	43.8241

表 4.2: PSOによる調整結果

	P1	P2	P3	P4	P5	Error(μ m · s)	Time(sec)	Fitness
PSO 1st	478	3013	-4068	1221	290	26.0797	2.264	43.8972
PSO 2nd	251	3294	-4131	1259	284	29.7773	2.11	44.6494
PSO 3rd	339	3443	-4116	1246	198	30.8791	2.089	44.6922
PSO 4th	207	3560	-5884	1400	200	35.4456	2.042	44.578
PSO 5th	363	2921	-4777	1008	50	23.1687	2.303	43.8915
PSO 6th	199	3879	-5059	1209	50	36.2876	1.91	45.4555
PSO 7th	396	4000	-4770	1400	137	37.1603	2.062	44.2608
PSO 8th	210	4000	-4695	1400	92	34.7927	1.998	44.9639

第5章 NCパラメータ最適化に適した新たな最適化手法の提案と検討

5.1 改良PSOによるNCパラメータ最適化

5.1.1 改良PSOの原理・特徴

4章でも説明したように、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可能性が高い。それは4.3の式(4.3)の α の値が大きいというのが一つの原因である。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、それが逆に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、最適解近辺での微調整にも対応できる手法を提案する。これが実現できれば、局所解に陥りやすいというPSOの弱点を補うことが可能となる。

5.1.2 改良PSOによる実験・考察

5.1.1で提案した手法が実際に上手くいくかどうか実験することで検証してみる。

まず実験条件として集団数：10、世代数：10とし、 α は0.3,0.6,0.9、それぞれの集団数を3等分して分配し8回実験を行った。

その結果、各パラメータの値、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図5.1のようになり、全てに言えることが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、最適解近辺での微調整にも対応できていると言える。改良法は図5.2を見ても分かるように従来法よりも改良法の方が安定して最適値に近づいた値に収束していつている。

具体的な値は表5.1のようになり、表4.2のPSOによる適合度の結果と比較してみると分かるように改良したPSOの方が高い確率で良い値を示している。しかし、まだ人の手による調整結果には及んでいない。

そこで、その原因として試行回数が少なかったということも考えられるので、集団数を24、世代数を16と今までの実験条件よりも多くしてGA,PSO,改良PSOの検証をしてみた。その結果は表5.2のようになり、GA,PSOは試行回数が増えてもあまり性能向上は得られなかったが改良PSOは人の手による調整結果と同じレベルの結果をより高い確率で探索できており、この手法は試行回数が増えるほど探索精度が増し、従来法に比べ局所解に強い手法であると言える。

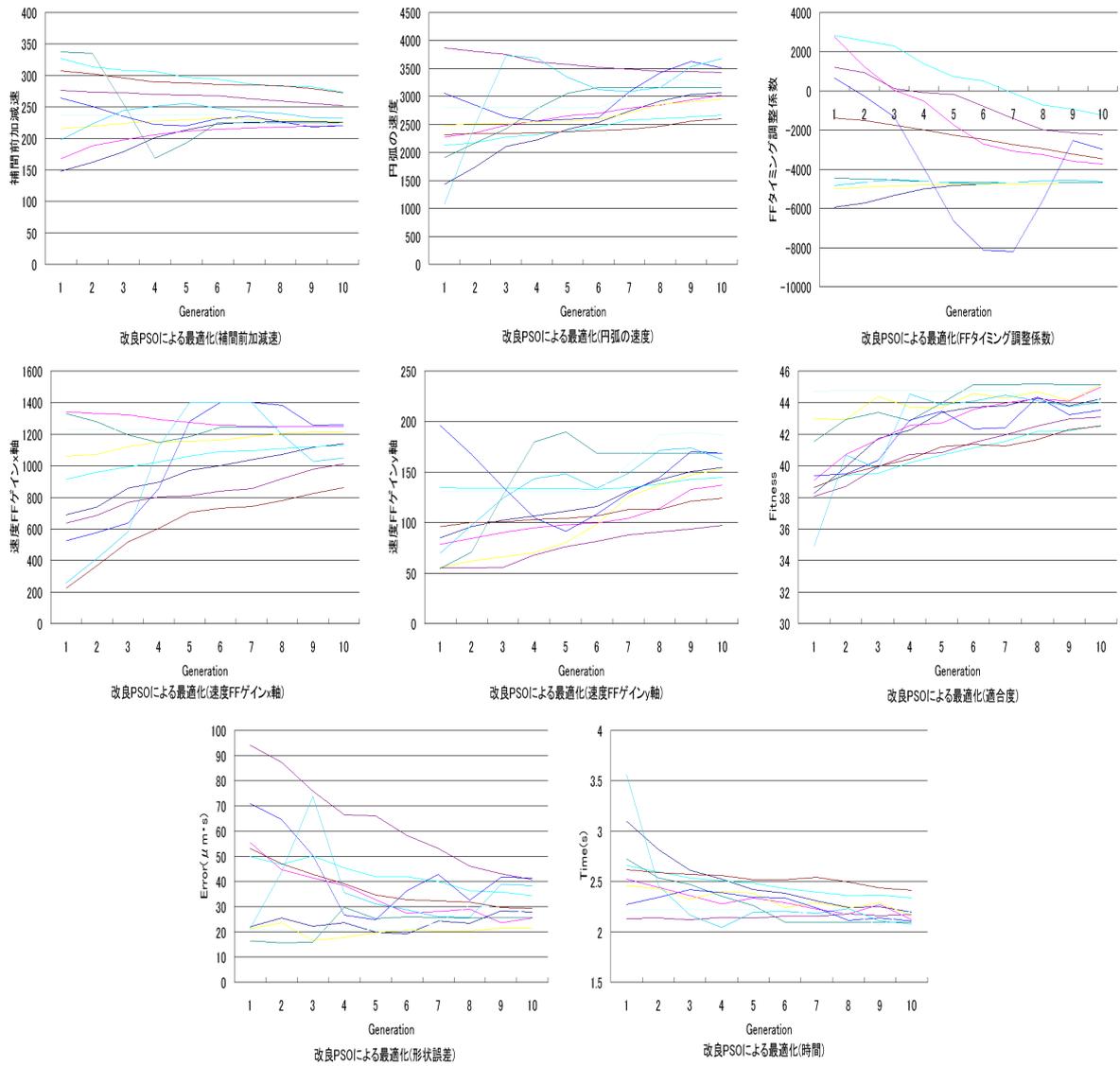


図 5.1: 改良 PSO によるパラメータ最適化実験結果

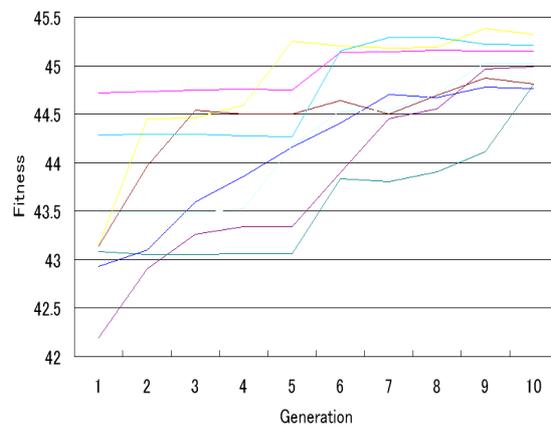


図 5.2: 改良 PSO による各世代の適合度最大値の推移

表 5.1: 改良 PSO による調整結果 (集団数 10, 世代数 10)

	P1	P2	P3	P4	P5	Error(μ m · s)	Time(sec)	Fitness
Improved PSO 1st	226	2947	-4706	1215	153	21.5469	2.154	45.1519
Improved PSO 2nd	291	3619	-4907	1226	77	28.8095	2.023	45.3801
Improved PSO 3rd	300	3021	-3836	1257	62	22.0996	2.164	45.0238
Improved PSO 4th	277	3048	-5137	1191	105	23.177	2.146	45.0472
Improved PSO 5th	431	3252	-5133	1160	50	24.5839	2.159	44.8118
Improved PSO 6th	236	2846	-5149	1198	145	21.4198	2.217	44.7056
Improved PSO 7th	268	3277	-4829	1222	114	24.9744	2.087	45.2973
Improved PSO 8th	294	3663	-4870	1272	54	27.7398	2.087	45.0207

表 5.2: GA, PSO, 改良 PSO による調整結果 (集団数 24, 世代数 16)

	P1	P2	P3	P4	P5	Error(μ m · 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	193	3136	-4565	1243	86	23.528	2.11	45.2743
Improved PSO 1st	240	3337	-4572	1172	139	30.2018	2.007	45.3574
Improved PSO 2nd	318	3596	-4032	1131	91	32.5374	2.041	44.9941
Improved PSO 3rd	276	2836	-4519	1151	70	17.6551	2.168	45.442
Improved PSO 4th	312	3241	-4176	1126	75	23.6685	2.079	45.4861

5.2 NN(ニューラルネットワーク)を利用したNCパラメータ最適化

5.2.1 NNの学習システム

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。[7][8]

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、 $y_i^k = f(\sum_i w_{ij}^{k-1,k} y_i^{k-1})$ に従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 $\Delta w_{ij}^{k-1,k} = -\eta \delta_j^k y_i^{k-1}$, $\delta_j^k = (\sum_l \delta_l^{k+1} w_{jl}^{k,k+1}) y_j^k (1 - y_j^k)$ 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。[9][10][11]

この特徴を活かすための手法を次節で説明する。

5.2.2 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

4.4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、それ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できるものと思われる。フローチャートで表すと図 5.3 のようになる。

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、

$$\begin{aligned}\mathbf{F} &= f(\mathbf{P}) \\ \mathbf{F} &= (F_1, F_2, \dots) \\ \mathbf{P} &= (P_1, P_2, \dots)\end{aligned}$$

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。

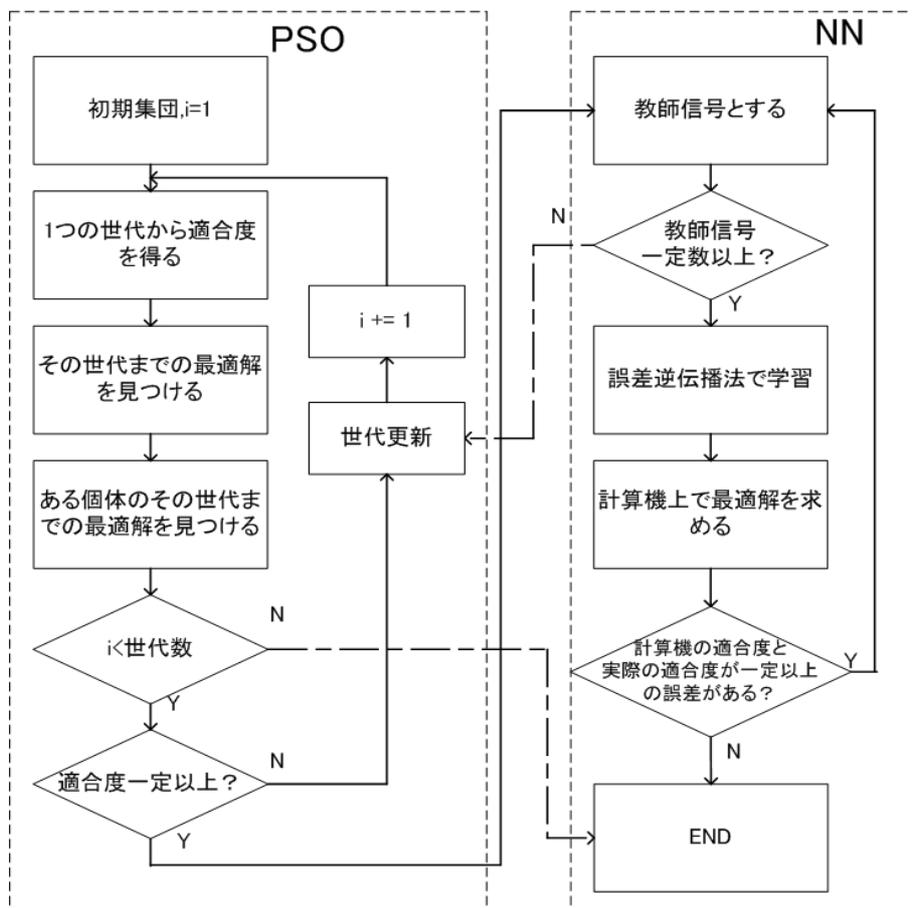


図 5.3: PSO と NN を組み合わせたアルゴリズム

5.2.3 PSO に NN の学習能力を組み合わせた手法による実験・考察

実際に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。この実験を 3 回行った。

その結果求められた最適解は表 5.3~5.5 のようになり、PSO 単体による結果が悪ければある程度の改善が見られたが、PSO 単体で最適値に近い値を導き出していると NN でそれより良い値を探索することはできなかった。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。

表 5.3: PSO、PSO+NN による調整結果 1 回目

	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

表 5.4: PSO、PSO+NN による調整結果 2 回目

	P1	P2	P3	P4	P5	Error(μ m · s)	Time(sec)	Fitness
PSO	363	2921	-4777	1008	50	23.1687	2.303	43.9041
PSO+NN	291	3205	-3408	1205	50	25.8689	2.154	44.7197

表 5.5: PSO、PSO+NN による調整結果 3 回目

	P1	P2	P3	P4	P5	Error(μ m · s)	Time(sec)	Fitness
PSO	172	4000	-4331	1282	50	35.3608	1.949	45.2641
PSO+NN	117	3257	-6312	1268	50	31.0739	2.034	45.0735

第6章 本論文のまとめと今後の課題

6.1 本論文のまとめ

本研究はNC工作機械の精密制御の為の最適パラメータ探索に適しているアルゴリズムを検証することを目的としたものである。3章において、実際に実験機を人の手で調整した時どのような調整法を取らなければならないのか、また自動チューニングする際の最適値の目安となるものをここで調べた。4章において、手動でのパラメータ調整が非常に面倒かつ難解な作業であるため、従来の最適化手法であるGA、PSOを用いてパラメータ探索の実験をすることでこれらの比較検証をした。その結果、GAは選択、交叉、突然変異を用いて世代更新をするが、それが確実に最適解に収束していくことは限らないことを示し、PSOは最適解近傍への収束は早いが局所解に陥りやすいことを示した。

これらの結果を踏まえ、最適パラメータ探索に適した新たなアルゴリズムの提案と検証を第5章で行った。具体的には、PSOの世代更新方法を集団全て同じものを使用するのではなく一部分は収束性に優れるよう、別の一部分は局所解に陥りづらくするようなPSOの改良案を最初に提案した。実際に実験をして検証してみると、従来のPSOと比べて最適値近傍への収束が遅くなってしまったものの、従来法よりも正確な最適値の探索ができるという結果が得られた。

次に提案した手法が、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である。実際に実験してみたが、PSOでの結果が悪ければNNを使うことで性能向上を実現することができたが、PSOが良い結果をだしていると、そこからの性能向上がほとんど行われなかった。

6.2 今後の課題

改良PSOは理想の結果を得たが、NNを利用したPSOはあまりうまくいかなかった。ただ、計算機の性能が日々向上している現代において工作機械の特性を計算機で推定して最適値を予想する手法は有力なものとなる可能性は秘めている。今回の実験では単純に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サポートベクトルマシンなど学習能力のある別の手法を試してみるのも悪くないと思われる。そして、改良PSOの方は速度更新式を世代毎に変化させる等してさらなる改善を行えればと考えている。

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] 伊庭 斉志, ”遺伝的アルゴリズムの基礎-GA の謎を解く-”, オーム社,1998 年.
- [3] 伊庭 斉志, ”遺伝的プログラミング入門”, 東京大学出版会,2001 年.
- [4] 張 炳勳, 堀 洋一, ”2 慣性システムのパラメータ自動チューニングに関する研究”, 電気学会産業計測制御研究会, IIC-05-47, 2005 年.
- [5] Sehoon Oh, Yoichi Hori, ”Control Parameter Optimization in the Hardware-in-the-loop System using Novel Search Algorithm”,IEEE-IECON 2006
- [6] 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.
- [7] 志水 清孝, ”ニューラルネットと制御”, コロナ社,2002 年.
- [8] 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.
- [9] 山崎 浩, ”ニューラルネットワークの近似理論と最適制御への適用”, 修士論文,1999 年.
- [10] 田口 功, 須貝 康雄, ”3 層ニューラルネットワークの関数学習の規格化による誤差変化”, 敬愛大学国際研究第 12 号,2003 年.
- [11] 田口 功, 須貝 康雄, ”3 層ニューラルネットワークにおける誤差特性を利用した関数学習の加速化”, 敬愛大学国際研究第 12 号,2004 年.

謝辞

本論文の執筆にあたり、私の所属する堀研究室のメンバーの皆様には大変にお世話になりました。

教授の堀洋一先生には、ご多忙な身であるにもかかわらず熱心なご指導をしていただき、研究を正しい方向に導いてくださったことに深く感謝いたします。技官の内田利之さんには、研究室にある実験機器を細かく、そして分かりやすく説明して頂き知識の幅を非常に大きなものとして頂きました。秘書の越智由里子さんには事務処理だけではなく研究をする上でさまざまなサポートをしていただきました。

研究員の呉世訓さんには自身の研究でお忙しいにも関わらず、本論文の作成のために実験機の仕組み・動かし方・注意点などを貴重な時間を割いてまで説明していただき、研究の方向性や実験結果を共に考察していただきました。

博士課程の賀鵬さん、河島清貴さん、藤井毅さん、耿聡さん、殷徳軍には常に親しく接していただきいろいろな話をしていただき大変参考になりました。特に河島清貴さんにはEVに使用される電気機器の作成の場実際に立ち合わせて頂き、回路の構造等を丁寧に教えていただきました。今後の研究にも、そして私生活においても非常に有用なものとなりました。

修士課程2年の吉田憲吾さん、中村則仁さん、趙莉さんには制御に関しての知識があまりない私に年齢の近い先輩として分かりやすく指導をして頂きました。特に吉田憲吾さんにはプログラムの書き方や、一般的な最適化手法の特徴を細かく説明して頂きました。そして中村則仁さんには報告資料の作成や発表の際の留意点などをアドバイスして頂き、そのスキルを大きなものとして頂きました。

修士課程1年の大西祐介さん、小池卓志さんには同年の学生がいない私の最も近い人間として接して頂きました。研究、生活面双方で何でも相談にのって頂き、研究室で過ごす時間を非常に楽しいものとして頂きました。

そして、実験機を提供して下さっている松原様、岩下様には実験方法、実験結果に対するアドバイスなど非常に懇切丁寧に対応して頂きました。深く御礼申し上げます。

最後に、この一年間は私にとってこれ以上ない有意義なものとなりました。堀先生をはじめ、研究室の皆さんに感謝したいと思います。本当にありがとうございました。