計測自動制御学会産業論文集
Vol.2, No.4, 27/33 (2003)

Nelder-Mead 法条件緩和アルゴリズムによるマニピュレータ制御パラメータのチューニング法

太田　順*・金子慎一郎*・新井民夫*・前田雄介*・杉　正夫*・千葉龍介*

Parameter Tuning Methodology for Manipulator Control by using Nelder-Mead Method with Constraint-Relaxation Algorithm

Jun OTA*・Shin'ichiro KANEKO*・Tamio ARAI*・Yusuke MAEDA*・Masao SUGI* and Ryousuke CHIBA*

Abstract: In this paper we propose a methodology of control parameter setting to move articulated manipulators quickly and precisely. This kind of operation has been done by humans, and it has taken a few months to set a large number of control parameters. We solve the problem by combining (a) computer simulations of manipulator's motion, and (b) non-linear optimization problem solver. We utilize a simulation-based optimization method as a whole. We adopt Nelder-Mead method as a specific optimization algorithm. We propose a constraints-relaxation method for fast calculation. Simulation results indicate the effectiveness of the proposed architecture. We can get the solution with 66 variables in about 3 hours.

Keywords: articulated manipulator, motion control, non-linear optimization, simulation-based optimization

1. 序論

高速・高精度に位置決め可能な多関節型マニピュレータの開発は工場における生産性の向上の観点から非常に重要である．

この問題に対して，現状でのロボット制御系調整現場では，汎用性の高い台形速度曲線を利用した制御則の枠内で，立ち上がり時間等の多数個の制御パラメータを設定するアプローチが採られている．ある特定の手先軌道（以後テスト動作パタンと呼ぶ）における手先振動とモータへのトルクの指令値を限界値以内に抑えつつ，評価関数である位置決め時間を最小とする制御パラメータを導出するものである．現状では，現場の人間が制御パラメータと個別の要求項目との関係を経験的に把握し，実機の挙動をみながら状況に応じて“勘”を頼りに制御パラメータ設定を行っていることが多い．そのため，調整者の熟練度に応じて調整結果にバリエーションが存在し，かつ設定に数ヶ月程度かかる問題を抱えている．

マニピュレータの高速制御という一般的な問題に対しては，理論的側面から多くの研究がなされている．マニピュレータを分布定数モデル（例えば文献 1））または集中定数モデル（例えば文献 2），3））の弾性体と考え，その制御手法を考案するのが一つのアプローチである．更にはそれらのモデル化誤差による影響を回避する手法として，システム同定を行いながら制御系を構成する適応的な手法（例えば文献 4）,5））や，物理パラメータの変動幅を予め規定し，その範囲内の変動に対応する制御系を構成するロバスト制御的な手法 [6)7)]が提案されている．しかしながらこれらの方法論は概して要求仕様と設計パラメータの関係を直観的に把握しにくく，設計時のパラメータ調整が難しい短所を有する．また，現場においては，従来用いられてきた制御則の代替として全く新しい制御則を採用することへの抵抗が根強く，なかなか実マニピュレータに適用されていないのが現状である．このようにマニピュレータの高速・高精度制御法には改善の余地が残されている．

*東京大学大学院工学系研究科
* School of Engineering, the University of Tokyo
(Received January 24, 2003)

本研究では，「現場においても適用が容易なマニピュレータの高速運動制御法の構築」を目的とする．ここでは，人間が行っている制御パラメータ調整則を非線形最適化問題として定式化し，計算機パワーにより適切な制御パラメータを高速に導出するというアプローチを採用する．本課題のチャレンジングな箇所は以下の三点である．

(a) 制御対象となるマニピュレータの機構モデル，制御手法のモデルが大変複雑であり解析的に解くことが困難である．
(b) 解導出の対象となる問題が「制約条件付き非線形最適化問題」という最適化アルゴリズムの中で難しいクラスに属するものであり，適用アルゴリズムの選定が困難である．
(c) 制御パラメータやテスト動作パタンの個数が非常に多いため通常の最適化アルゴリズムを適用したのみでは解導出に時間がかかる．

本論文では(a)に関しては，マニピュレータ及び制御則のモデルを計算機上に実装し，あるテスト動作パタンに対する制御パラメータを繰り返し順問題として解く．(b)に関しては，評価関数の勾配情報を用いない Nelder-Mead 法を採用する．(c)に関しては，本研究で扱う問題の性質を利用して，全制御パラメータ中の一部と制約条件のうちの一部を考慮して繰り返し演算を行う条件緩和アルゴリズムを提案する．

本論文の構成を以下に示す．２章前半で調整対象となる制御パラメータ並びに設計時の制約条件，評価関数について述べる．後半で提案するアーキテクチャについて述べる．３章では Nelder-Mead 法の選定過程について述べ，４章では条件緩和アルゴリズムについて説明する．５章でシミュレーション結果の評価を行い，提案手法の有効性を示す．６章で結論を述べる．

2. 問題設定と提案アーキテクチャ

2.1 問題設定

2.1.1 制御パラメータと運動パタン

本論文では，ファナック(株)製 6 自由度垂直多関節型マニピュレータ R-2000iA/165F を制御対象とする．動作領域と座標系を **Fig. 1** に示す [8)]．当該マニピュレータは，質量が 1,210[kg]，繰り返し位置決め精度が±0.3[mm]，手首部最大可搬質量が 165[kg]である．各関節の最大速度と質量を **Table**

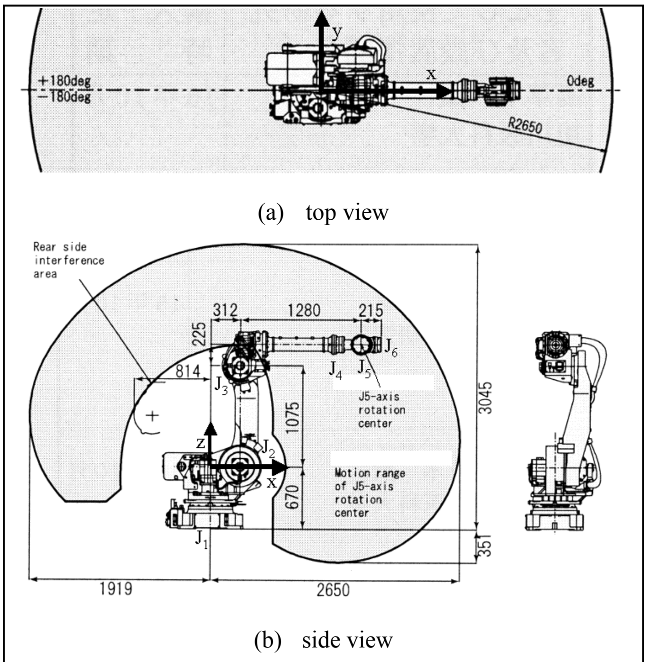


Fig. 1 Operational Space of the Robot

1 に示す．モータには AC サーボモータが用いられている．

手先の目標位置が与えられると各軸に関して **Fig. 2** に示すような台形速度曲線に基づいて運動生成する．ここで，目標位置到達までのおおまかな時間を調整する R_1 の部分と，位置決め時の振動を抑えるため曲線の「なまり」を調整する R_2 の部分が存在する．まず R_1 については，各関節の立ち上がり，立下り時それぞれにおいて，**Fig. 3** に示す３本の線分で近似される関節部モータの速度―最大出力可能トルク曲線の範囲内で動作可能な速度指令を出力する．また，R_2 の部分については，各関節において，5 種類の慣性モーメントに対する共振周波数値を推定し，その間を線形補間することで，慣性モーメントと共振周波数との関数関係を近似している．マニピュレータの軌道制御時のマニピュレータの姿勢から，慣性モーメントを求めることができ，上記の関数関係からその時の共振周波数が得られるため，その共振値を回避する加減速を行う．本来これらのパラメータはモータの電気特性やマニピュレータの機構特性を表現しており，通常は制御時の設計対象ではないが，本論文ではこれらの物理パラメータ値を推定し，それらの値に基づき制御入力を決定して，制御特性を評価している．すなわち，制御入力は上記の推定値に基づいて決定されるため，上記の推定値がマニピュレータ制御のための設計パラメータとなる．

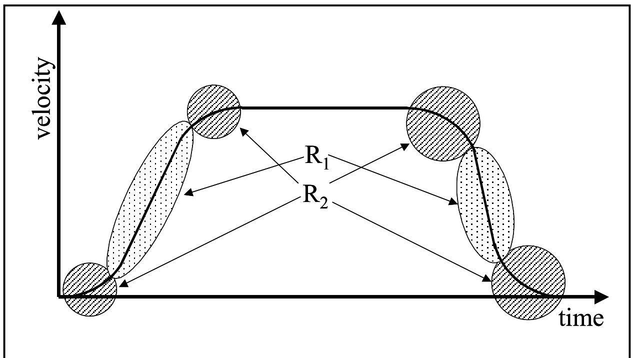


Fig. 2 Time sequence of angular velocity

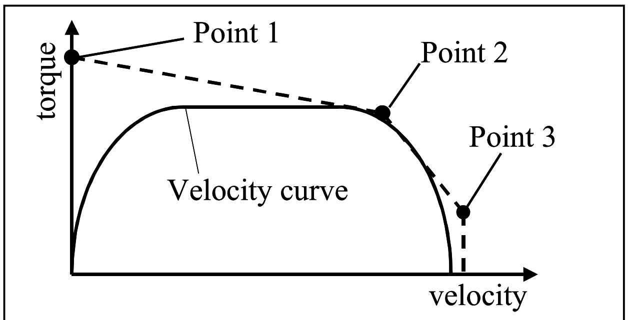


Fig. 3 Approximation of torque-velocity curve

設計パラメータの総数は，各関節あたり，最大トルクの推定値が立ち上がり立下りそれぞれ 3 つずつ存在し，共振周波数の推定値が計５つ存在するため, 合計 $6\times(3\times2+5)=66$ 個の設計パラメータが存在する．このようなパラメータ数・種類の設定は従来からのロボット制御において経験的に得られたものであり，本論文ではこれをそのまま採用する．これらを次項の評価関数を最適化するべく調整する．

Table 1 Maximal Angular Velocity of Axes

Axis	J_1, J_2, J_3	J_4, J_5	J_6
Maximal velocity	1.83 [rad/sec.]	2.27 [rad/sec.]	3.67 [rad/sec.]
Weight of the Motor	22[kg]	7[kg]	7[kg]

2.1.2 非線形最適化問題としての定式化

評価項目を明らかにするために，まず．評価項目の導出時に利用するテスト動作パタンについて述べる. ここでは, **Fig. 4** に示すある一定長の立方体の辺に沿った軌道をある一定刻み幅の間欠運動によりマニピュレータの手先が動作するものとする．具体的には 12 種類（100, 150, 200, 250, 300, 350, 400, 450, 500, 600, 700, 800[mm]）の刻み幅を与えることで，それぞれに対応するテスト動作パタンを考えている．このパタンによりマニピュレータを動作させ，以下の項目を評価する．

- サイクルタイム：各テスト動作パタンを生成するのにかかる時間をサイクルタイムとする．作業達成時間を評価している．
- 最大オーバーシュート量：マニピュレータの間欠動作の度にオーバーシュートが生じる．マニピュレータ基準座標系のＸ軸，Ｙ軸，Ｚ軸の各軸方向へこのオーバーシュート量の射影を取ったときの最大値を最大オーバーシュート量と呼ぶ．マニピュレー

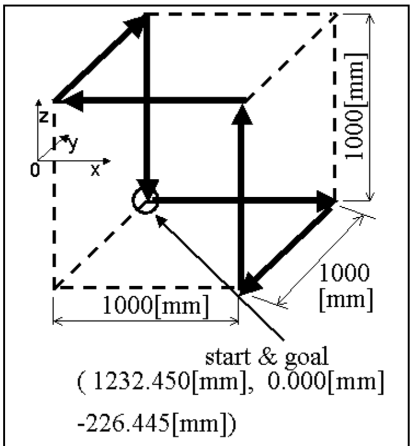


Fig. 4 Test Motion Pattern

タの精度を評価している.

- 最大トルク値：ハードウエアの制約によるモータのトルク限界値が存在する.

本論文では，最大オーバーシュート量と最大トルク値を制約条件とし，サイクルタイムを評価関数として最小化を目指す制約条件つき最適化問題を考える．外点ペナルティ関数法[9]を利用して制約条件なし最適化問題に変換して解いている．これはペナルティ関数を用いて，制約条件を満たしていない領域においては，評価関数値が極端に大きくなるように評価関数を修正する方法である．12 個全てのテスト動作パタン各々に対して評価を行い，各テスト動作パタンの評価値 $f_i(x)$ の単純和を総合評価値 $F(x)$とする.

$$F(\boldsymbol{x}) = \sum_{i=1}^{12} f_i(\boldsymbol{x}) \tag{1}$$

2.2 提案アルゴリズムの概要

「制御特性から適切な制御パラメータ調整則を推測する」という逆問題解法を適用するのは，モデルの複雑さの観点から困難である．そこで，「制御パラメータを調整すると制御特性が変化する」という順問題を繰り返し解くという手法が有効であると考え，Fig. 5 に示すアーキテクチャを提案する．ここで最適化モジュール（Optimization）において，前述した最適化問題を解く．シミュレータ（Simulator）としては，ファナック(株)製多関節型マニピュレータ用運動シミュレータを使用した．マニピュレータのシミュレーションモデル（Manipulator simulation model）としては，J_1,J_2,J_3 軸の減速器まわりのばね性を考慮した弾性に加えて J_2 部におけるアームのねじれ（z 軸周りの回転）と倒れ（y 軸周りの回転）の弾性を考慮している．本手法は simulation-based optimization[10]の一手法と位置付けられる.

本法のシミュレーションに用いた計算機環境は PentiumIII CPU1GHz, メモリ 256MB であり，C 言語を用いて最適化アルゴリズムをインプリメントした．一つの総合評価値を導出するのに要する時間は約 80 秒であり, 計算負荷がかなり重いことがわかる.

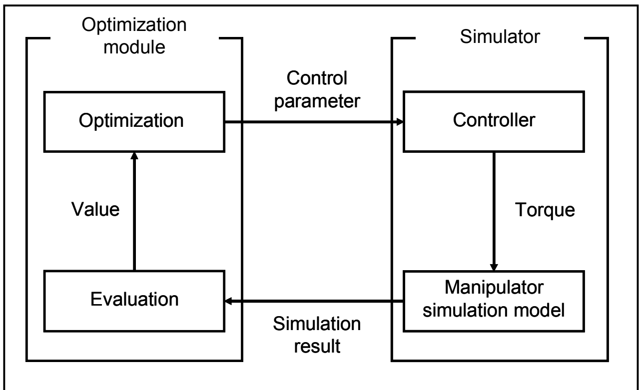


Fig. 5 Architecture of the Proposed System

3. 最適化アルゴリズムの選定

本手法では最適化計算にシミュレータを用いるため，以下に示す問題点が生じる.

1) 評価関数の微分値の利用が困難
2) シミュレータ･実機間にモデル化誤差が存在
3) 1 回のシミュレーションにかかる時間が大

そこで，微分値を利用せずに最適解を探索する手法として代表的である Nelder-Mead 法と Powell 法とを比較検討する.

3.1 Nelder-Mead 法と Powell 法

Nelder-Mead 法とは，n 次元空間においてまず n+1 個の解候補集合を与え，それらに対して，状況に応じて四種類（鏡像，拡張，収縮，縮小）の操作のいずれかを選び，解候補集合を徐々に最適解に収束させていく方法論である [9]．このうち，鏡像，拡張，収縮の三操作は，解候補集合中の一候補を移動させるものであるのに対して，最後の縮小操作は他の操作が不調に終わった時に用いるものであり，ほとんどすべての候補を移動させるものであることに注意されたい．一つの解候補を移動させるためには，総合評価値を一回計算する必要がある．本研究で扱っている問題のように評価に要する計算負荷が重い問題に対しては，縮小操作に多大な時間を要することがわかる．この縮小操作は解導出が停滞すると頻繁に呼び出される特徴がある．また，探索の終了条件は，評価値の最大値と最小値により表される関数（二つの値の差を正規化したもの）があるしきい値 T_h より小さくなる条件を満たした場合とする．なお，ここでの最大値，最小値とは，各操作後における解候補集合内での最大値，最小値を示す．また，以下の式において，評価値の最大値を maxPIvalue，最小値を minPIvalue としている.

$$2 \times \frac{\left|\max \mathrm{PIvalue} - \min \mathrm{PIvalue}\right|}{\left|\max \mathrm{PIvalue} + \min \mathrm{PIvalue}\right|} < T_h \tag{2}$$

Powell 法は，解候補を一点とり，そこから別の点を探索するために，探索次元数分の直線探索方向の調整とその方向への直線探索を繰り返す．その中の一番適切な解へ移動し収束するまで同様の手続きを繰り返す方法である [9].

3.2 Nelder-Mead 法と Powell 法の比較評価

本研究で扱う問題における両方法の評価を行う.

最初に Nelder-Mead 法の計算時間を見積もる. 収束に掛かる評価回数を N_n 回とし，縮小操作が m 回起きたとする．縮小の場合の計算時間は 66×80=5280 秒, その他の操作の場合は 80 秒であるため，収束するまでに要する時間は，式(3)のようになる.

$$80 \times (N_n - m) + 5280 \times m \tag{3}$$

次に Powell 法の計算時間について見積もる. 収束までに要する解候補の移動回数を N_p 回, 一回の直線探索に要する評価回数を M 回とすると，1 回の移動に際して各パラメータの軸方向毎に計 66 回の直線探索を行わなければならないので, 計算時間は式(4)に示すようになる.

$$5280 \times N_p \times M \tag{4}$$

具体的数値を代入することで，上記の二手法を比較する．制御パラメータ設定のための N_n を 80 とする時に後述するシミュレーション結果より m の値は 3～5 程度であった．これに対して，Powell 法では，例えば N_p が 5，M が 10 程度と仮定するのは不自然ではないが, その場合でも Powell 法の方が Nelder-Mead 法の 10 倍程度かかってしまう. このことから, Nelder-Mead 法が有効であるといえる．その理由としては, 以下の二つが考えられる. (1)前述したように 評価値を導出するために時間がかかる, (2) パラメータ導出時には，初期値設定が重要であるが，本問題の場合解候補のおおまかな推測が

なされているため，縮小操作が呼ばれる回数が比較的少なくて済んでいる．

また，Nelder-Mead 法のように複数個の解候補の集合を求める方法論は，Powell 法のようにピンポイントの最適解を求める方法論と異なり，評価値を最適化する一定の範囲の領域を導出するという性質を有する．そのため，本章の冒頭で述べた性質の一つである，シミュレータ･実機間のモデル化誤差が存在する場合にも妥当な解を得る可能性が高い．

4. 条件緩和アルゴリズムの提案

Nelder-Mead 法を使った場合でも計算時間の問題はなお深刻である．これを解決するために条件緩和のアルゴリズムを提案する．

総合評価値は 12 個存在するテスト動作パタンそれぞれに対する評価値の線形和で表現されているため，総合評価値を計算するには多大な時間がかかる．ここで，総合評価値に悪影響を与えている要素に関してのみ評価する，すなわち，ボトルネックとなっている評価値を検出して，その評価値を集中的に改善するべくパラメータ調整をするという戦略を採用することで総合評価値を高速に改善できる可能性がある．これは「ある一つのテスト動作パタンに関する制約条件，評価関数のみを考慮し，他のテスト動作パタンに関するものは考慮せずに解探索をする」ため，条件緩和アルゴリズムと呼ぶ．

その手順を以下に示す．フローチャートを **Fig. 6** に示す．

ステップ 0： 総合評価値について最初の Nelder-Mead 法の変換操作を 1 回行う．

ステップ 1： 終了条件を判定する．終了条件を満たせば終了する．満たさなければ，ステップ 2 へ行く．

ステップ 2： 総合評価値に最も悪影響を与えている評価値をボトルネック評価値とする．

ステップ 3： ボトルネック評価値を評価関数として K 回 Nelder-Mead 法の変換操作を行う．

ステップ 4： 評価値の最大値を与えている制御パラメータと，総合評価値を最大にしている制御パラメータと交換する．

ステップ 5： 総合評価値について Nelder-Mead 法の変換操作を 1 回行う．ステップ 1 に行く．

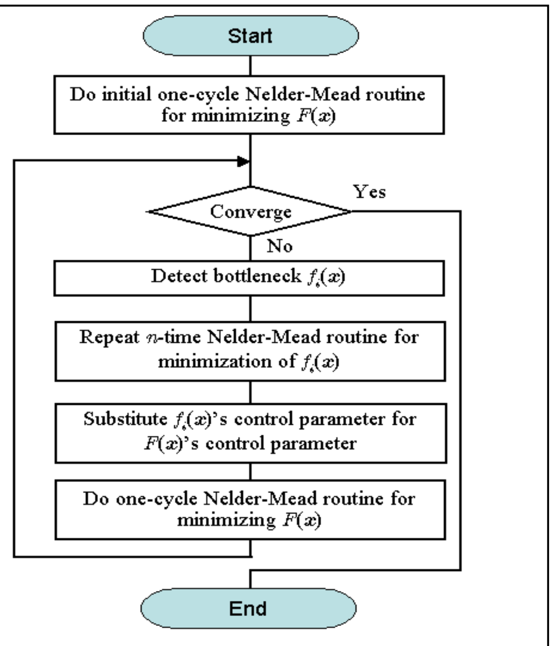


Fig. 6 Relaxation algorithm

ここで K の値を大きくすればする程，一回の移動に要する時間が短くなるため高速に最適解を得られる可能性がある．ただしあまり大きくし過ぎると導出される解候補が振動的になって結果的に収束が遅くなる危険性がある．また，ボトルネックである評価値の選択は，あらかじめ各テスト動作パタンに標準終了時間を設定し，各評価値をその標準値で正規化した後でもっとも大きい値をとるものとした．

5. シミュレーションによる検証

5.1 制約条件と条件緩和アルゴリズムの関係

まず最初に，最適化問題の性質と条件緩和アルゴリズムとの関係について基礎実験を行う．以下のシミュレーションにおいて解探索の終了条件を，式(2)の T_h =0.05 と設定した．条件緩和アルゴリズムを用いない手法と繰り返し回数 K を 5 回，10 回，20 回とした条件緩和アルゴリズムを用いた手法について比較した．両方法ともまず制約条件が満たされていなく，ペナルティが課せられた状態（評価値が著しく大きい）から始まり，あらゆる制約条件を満たすべく探索を進め，制約条件を満たす範囲内に到達する（ここまでを第一段階と呼ぶ）．その後，その範囲内でより適切な解を探索し，結果的に準最適解を導出する（ここまでを第二段階と呼ぶ）．このように最適値探索を二つの段階に分けることができるため，それぞれの段階における適用手法の評価を行う．

各手法における総合評価値の最小値の時間遷移を **Fig. 7** に，最大値の時間遷移を **Fig. 8** に示す．また，最初から制約条件を満たしている初期値を与えたときの総合評価値の最大値の時間遷移を **Fig. 9** に示す．ここで各図の縦軸は 1 ステップを 8msec とした際の総合評価値（単位はステップ数）を示している．ここで，Fig. 7，Fig. 8，Fig. 9 は縦軸の目盛りが異なっていることに注意されたい．また，Fig. 7，Fig. 8 は第一段階と第二段階の両者を含み，Fig. 9 は第二段階のみを含んでいる．Fig. 7 からわかるように，総合評価値の最小値については各手法間で評価値はたかだか約 150 程度の範囲内に収まっており，相互に大きな違いが見られなかった．一方 Fig. 8, Fig. 9 から総合評価値の最大値については、以下のことがわかった．

- ペナルティ関数の影響を受けている段階，すなわち制約条件を満たしていない第一段階を含む探索においては，条件緩和アルゴリズムを利用しない手法がより速く改善されている．
- ペナルティ関数の影響を受けない段階，すなわち制約条件を満たしている第二段階においては，繰り返し回数が 5 回，10 回の緩和アルゴリズムを用いた手法のほうが速く改善されている．収束条件を満たすまでのステップ数（この場合評価値が約 32,500）を比較すると，繰り返し回数 5 回の場合は緩和アルゴリズムを用いない場合と比較して，約 60%程度の時間で到達している．

上記の現象の理由であるが，制約条件を満たしていない領域では，探索中の制御パラメータ周囲のボトルネックとなる評価値の勾配方向が総合評価値のそれと大きく異なることが

予想できる．そのため，条件緩和をして勾配方向を求めてもそれが必ずしも全体評価値の勾配方向と一致せず，無駄な探索を繰り返す危険性が高い．一方ひとたび制約条件が満たされれば，ボトルネックとなる評価値が総合評価値に大きな影響を及ぼすことが考えられるため，条件緩和アルゴリズムが有効に働くと理解できる．

このことから，制約条件を満たしていない領域では通常の探索手法を用い，制約条件を満たしたら条件緩和アルゴリズムを用いるハイブリッド型の探索手法が望ましいと推測できる．この妥当性について次節で検証する．

5.2 シミュレーション結果

制約条件を満たしていない場合には条件緩和アルゴリズムを利用せず，制約条件を満たした場合には条件緩和アルゴリズムを利用する（繰り返し回数5回もしくは10回）ハイブリッド型手法と，通常の（条件緩和アルゴリズムを利用しない）手法とを比較する．適当にとった5種類の初期値からの収束の度合いを比較評価した．以下の結果が得られた．

- 条件緩和アルゴリズムを用いた場合は，5回中すべての場合で解を得ることができた．計算に要する時間は約3時間程度であった．
- 条件緩和アルゴリズムを用いない手法の場合，全5回中2回しか解が導出できなかった．それ以外の場合は規定の評価回数80回を超えたため、処理を強制終了してしまっている．解が導出されたときの計算時間は約6.4時間であった．

ある初期値に対する，総合評価値の最大値の遷移状況を **Fig. 10** と **Fig. 11** に示す．ハイブリッド型でも制約条件を満たすまでの収束過程は通常の方法と同一であるため，Fig. 10においてほとんど違いが見られず，わずかに10,000ステップ周辺で微小にずれているのみである．そこでFig. 11はFig. 10において9,000ステップ以降を拡大表示している．条件緩和アルゴリズムを用いた方は，約11,000ステップまでには収束条件を満たして探索を終了しているのに対し，条件緩和アルゴリズムを用いない通常の方法では，収束に至る前に約11,000秒～約22,000秒で再度縮小操作を行い，多大な時間を費やしている．縮小操作がたまたま発生しない場合には，通常の手法でも比較的短時間で解が得られることもあるが，ひとたび縮小操作が必要となると多大な時間がかかっていることがわかる．

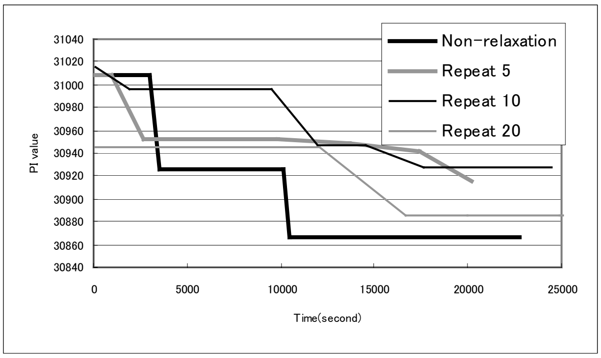


Fig. 7 Comparison of non-relaxation method and relaxation method (minimal value)

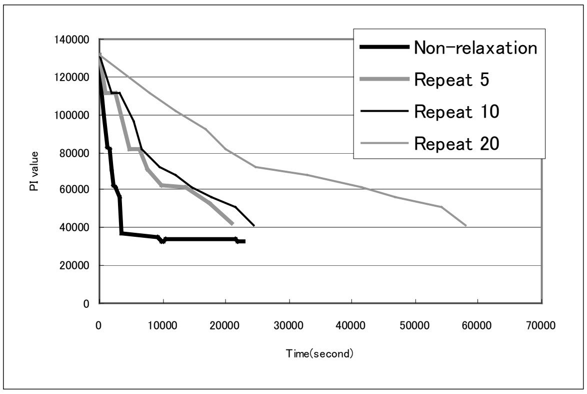


Fig. 8 Comparison of non-relaxation method and relaxation method (maximal value)

このシミュレーション結果より，ハイブリッド型の妥当性が示された．この傾向は，テスト動作パタン数が増大する程，更に増大すると考えられる．また，しきい値 T_h が小さな値をとるほど，すなわち探索終了条件が厳しくなるほど，緩和法を用いない場合に収縮操作が起動される可能性が高まり、提案手法の有効性が高くなると考えられる．なお，提案手法は大域的最適解を保証するものではない．本研究の枠内においては先に述べたようにパラメータ値に関する或る程度の見積もり値が存在するため，特にハイブリッド型を用いた場合，初期値の変動に対して解やそのクオリティが大きく異なることはなかった．初期値によらず大域的最適解の導出の保証が必要な場合は，シミュレーテッドアニーリング的手法をNelder-Mead 法に組み入れる [11]ことで展開可能であると考えられる．

本論文で示した手法は，シミュレータ上に実装でき制御パラメータを抽出できる他のあらゆる制御系に対して適用可能であり，一般性が高い．これは従来用いられている制御則アーキテクチャ自体を改変せずにその制御特性を最大限生かすという方法論となっており，現場への適用が困難ではないと考えられる．

6. 結論

本論文では，多関節型マニピュレータの高速制御のための制御パラメータ設定法を提案した．提案手法は以下の三つの特徴を有する．(a) 複雑な振る舞いをモデル化しつつ最適解を導出するためにシミュレータと最適化法を組み合わせた手法を適用している．(b)最適化法として評価関数値の微分情報を用いない Nelder-Mead 法を利用している．(c)計算の高速性を

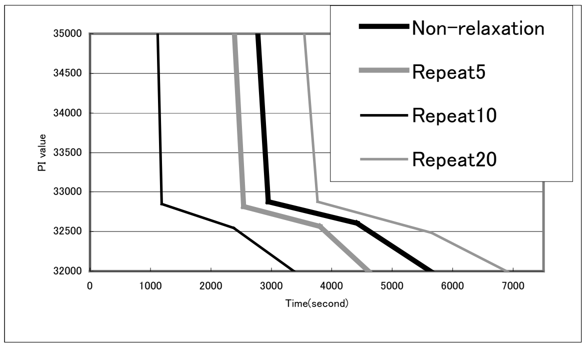


Fig. 9 Comparison of non-relaxation method and relaxation method (maximal value) after all the constraints are satisfied

得るため，条件緩和アルゴリズムを組み入れたハイブリッド型手法を採用している．シミュレーションによる実装の結果，66 種類のパラメータ数, 12 個のテスト動作パタンに対して約 3 時間弱の計算時間で解を得ることができた．このアルゴリズムは実際の制御パラメータ設定を支援するツールになると考えられる．問題領域に適した適切な繰り返し回数の導出等が今後の課題である．

謝辞

本研究遂行にあたり，多大な御支援をいただいたファナック(株) 渡邉 淳氏, 加藤 哲朗氏, 一之瀬 雅一氏に謝意を表す．

参　考　文　献

1) 嘉納秀明: フレキシブルアームの分布定数系モデル，日本ロボット学会誌, **6**-5，430/435(1988)
2) 山浦，渡邊，小野: 多リンクフレキシブルマニピュレータの制振駆動，日本機械学会論文集 C 編，**66**-645，1574/1581(2000)
3) 吉川，田村: フレキシブルアームに対する仮想受動関節モデルの有効性の検討，日本ロボット学会誌，**12**-2，250/259(1999)
4) 宮里，大島: ロボットマニピュレータの非線形適応制御，計測自動制御学会論文集, **24**-1，63/68(1988)
5) 川崎，望月，神崎: マニピュレータのモデルベースド適応制御における効率的な計算法と軌道制御実験，計測自動制御学会論文集, **30**-4，435/442(1994)
6) 尹，早川: ロボットマニピュレータのロバスト制御および適応ロバスト制御，日本機械学会論文集 C 編，**67**-657，1507/1514(2001)
7) 尹，尾形，早川: ロボットマニピュレータの非線形 $H\infty$ 適応ロバスト制御，計測自動制御学会論文集，**37**-7，647/656(2001)
8) http://www.fanuc.co.jp/ja/product/robot/lineup/r2000ia.htm
9) 今野，山下: 非線形計画法，日科技連 (1978)
10) M.A.Ahmed, T.M.Alkhamis: Simulation-based optimization using simulated annealing with ranking and selection, Computers & Operations Research, 29, 387/402(2002)
11) 大林茂: CFD 利用の新段階－数値最適化，日本機械学会誌, **105**-999，64/69(2002)

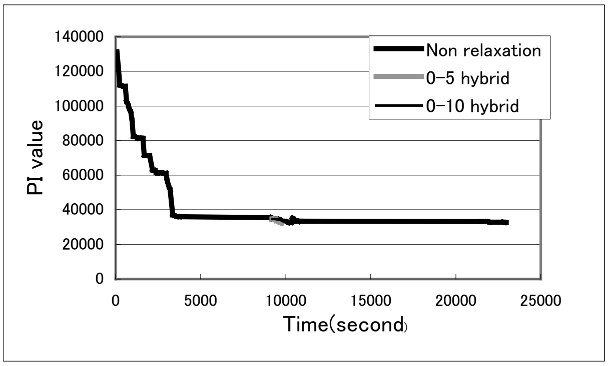


Fig. 10 Transition of the Maximal value

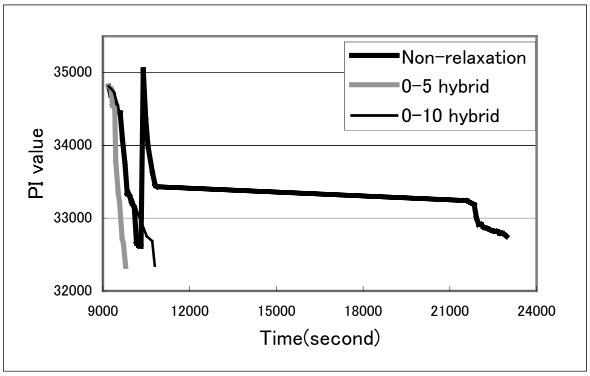


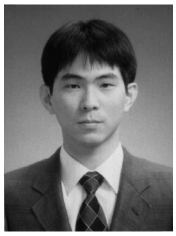
Fig. 11 Transition of the Maximal value (after 9000 step)

［著　者　紹　介］

太　田　　順（正会員）

1989 年東京大学大学院工学系研究科精密機械工学専攻修士課程修了．同年新日本製鐵(株)入社．91 年東京大学工学部助手．94 年同講師．96 年より東京大学大学院工学系研究科精密機械工学専攻助教授．この間 96～97 年 Stanford 大学 Center for Design Research 客員研究員，群知能ロボット，大規模搬送システム，ロボットの環境設計等の研究に従事．博士(工学)．

金　子　慎　一　郎



2000 年東京大学工学部精密機械工学科卒業．2002 年同大学大学院工学系研究科修士課程修了．現在ソニー(株)勤務．在学中は知能ロボットの研究に従事．

新　井　民　夫（正会員）

1947 年東京生まれ．70 年東京大学工学部精密機械工学科卒，77 年同博士課程修了．79 年英国エディンバラ大学人工知能学科研究員．87 年東京大学大学院工学系研究科精密機械工学専攻教授．00 年人工物工学研究センター長．産業用ロボット言語の標準化活動を推進．自動組立，クレーンとロボットとの協調制御，距離画像，移動ロボットの群制御，自律分散システム，ホロニック生産システムなどの研究に従事．RoboCup Legged Robot League 参加．精密工学会論文賞，IMS 賞など．JSPE,JSME,IEEE,IAS,RSJ,CIRP などの会員．

前　田　雄　介

1972 年 11 月 10 日生．97 年東京大学大学院工学系研究科精密機械工学専攻修士課程修了．同年大日本印刷(株)入社．99 年東京大学大学院工学系研究科助手．マニピュレーション計画，ロボットの技能教示，ホロニック組立システム等に関する研究に従事．IEEE，日本ロボット学会，精密工学会の会員．博士(工学)．

杉　　　正　夫（正会員）

1998 年東京大学工学部計数工学科卒業．2000 年同大学大学院工学系研究科修士課程修了（精密機械工学専攻）．2003 年同研究科博士課程修了（精密機械工学専攻），博士（工学）．現在は東京大学大学院情報理工学系研究科 COE 特任助手．自律分散型交通信号制御・自律分散型生産システムの研究に従事．IEEE，日本ロボット学会などの会員．

千　葉　龍　介

1999 年東京大学工学部精密機械工学科卒，2001 年同大学院修士課程修了，同年より東京大学大学院工学系研究科精密機械工学専攻博士課程に在籍．マルチロボットシステム設計などの研究に従事．2000 年 IMS 論文賞受賞．日本ロボット学会学生会員．