

産業用ロボットの CP 制御における動力学モデルを用いない最適軌道生成

著者	立矢 宏, 伊谷 英之, 林 道大, 伊賀 裕和, 樋口 理宏
雑誌名	日本機械学会論文集. C編 = Transactions of the Japan Society of Mechanical Engineers. C
巻	79
号	805
ページ	3075-3087
発行年	2013-01-01
URL	http://hdl.handle.net/2297/36261

doi: 10.1299/kikaic.79.3075

産業用ロボットの CP 制御における動力学モデルを用いない最適軌道生成

立矢 宏^{*1}, 伊谷 英之^{*2}, 林 道大^{*3}
伊賀 裕和^{*2}, 樋口 理宏^{*1}

Generating Optimized Trajectories of Industrial Robots in CP motion without Kinetic Models

Hiroshi TACHIYA^{*1}, Hideyuki ITANI, Michihiro HAYASHI,
Hirokazu IGA and Masahiro HIGUCHI

^{*1}Kanazawa University, Institute of Science and Engineering, Faculty of Mechanical Engineering,
Kakuma-machi, Kanazawa city, Ishikawa, Japan

This study proposes a method of generating the optimized trajectory, which determines change of the displacement of a robot with respect to time, to reduce electrical energy or the peak current value occurred in motors. Previously, a lot of studies tried to optimize the trajectory of a robot, needing its kinetic model to obtain the input torque or energy as the evaluation value. However, the construction of the kinetic model is considerably difficult, because it needs to identify or to measure a lot of specifications of a robot. Thus, the present study proposes the method that obtains the evaluation value by measuring the currents, powers and so on of an actual robot in operation. Therefore, it does not need the analysis of the motion of a robot with the kinetic model. The proposed method optimizes the trajectories with a heuristic procedure by evaluating the measured values. In this study, we addressed the optimization method for continuous path controlled robots and applied it to a commercially available industrial manipulator, confirming that the generated trajectory can reduce electrical energy and the peak current value in a short time.

Key Words : Robot, Motion Control, Trajectory, Kinetic Model, Heuristic Algorithm

1. 緒 言

産業用ロボットには生産性向上のため動作の高速化がしばしば望まれる。しかし、動作の高速化はロボットの消費電力の増大や、駆動力の増加によるモータの大型化を招きやすい。そこで、作業内容や所要時間を変えずにロボットの消費電力や駆動力を低減させる手法を確立すれば、生産効率を保ったまま、省エネルギー化の実現、モータの小型化によるロボットの軽量化などが可能となる。

ロボットの駆動力は、ロボットの位置・姿勢の時間に対する変化に大きく依存する⁽¹⁾。そこで、ロボットの位置・姿勢の時間に対する変化、すなわち、軌道を最適化して、駆動力、消費エネルギーなどを低減することが検討されている^{(2)~(6)}。その多くは、図1の赤枠内に示すように、対象とするロボットの動力学モデルを構築して数値解析を行い、駆動力などを求め評価値として、最適化手法により軌道を決定する。しかし、多自由度機構であるロボットの数値解析は複雑で、駆動力と位置・姿勢との関係は強い非線形性を有するため最適化は容易でない。例えば、加藤は最小原理に基づく最適化問題として産業用ロボットの消費エネルギーなどを最小化する厳密解を求める方法を検討している⁽²⁾。しかし、ロボットの運動を表す状態方程式の決定や、その偏導関数の計算は複雑である。また、ロボットの軌道の決定は、多峰性を有する非線形最適化問題となるため、長時間の繰返し計算と

*原稿受付 2013年03月10日

*1 正員, 金沢大学理工研究域 機械工学系 (〒920-1192 石川県金沢市角間町)

*2 金沢大学 大学院 自然科学研究科 (〃)

*3 正員, 金沢工業高等専門学校 (〒921-8601 石川県金沢市久安 2-270)

E-mail: tachiya@se.kanazawa-u.ac.jp

初期値選定の試行錯誤を要することが多い。そこで、ロボットの状態方程式と偏導関数の簡便な計算法⁽³⁾の提案や、多項式近似⁽⁴⁾、3次スプライン曲線⁽⁵⁾、正弦関数⁽⁶⁾などにより軌道を関数近似し、解の形を制限して最適化を容易にし、駆動トルクや消費エネルギーを最小化する方法が提案されている。しかし、これらの手法も産業用ロボットの自由度や動作範囲が拡大すれば、決定すべき変数が増え、最適化は困難となる。そこで、著者らは先に発見的手法を用いた軌道の最適化を検討し⁽⁴⁾、大型産業用ロボットの駆動力の低減を実現した。同方法では、初期値を選定することなく、比較的短時間で駆動力を低減する軌道の探索が行えた。

しかし、著者らの手法を含め以上の方法はすべて、産業用ロボットの精確な動力学モデルを必要とする。ロボットの動力学モデルを構築するためには、機構の寸法、質量、慣性モーメント、さらに、可動部の摩擦係数など多数の諸元と特性値が必要である。市販されている産業用ロボットでは、これらが未知であることが多く、その同定も困難なため⁽⁸⁾、これまで述べた手法を適用することは容易でない。

そこで、本研究ではロボットの動力学モデルを用いた解析を行わず、図1の青枠内に示すように、ロボットの実機そのものを稼働させてモータの電流値などを測定し評価値として最適な軌道の生成を行う手法を提案する。同手法は測定値から評価値を得るため、ロボットの動力学モデルを必要としない。よって、稼働中の産業用ロボットへも適用が行え、通常はモデル化が困難な摩擦や経年変化などの影響も考慮可能である。

本論文では同手法の産業用ロボットへの適用を容易とする軌道の表現方法や、ロボットに要求される動作が複雑な場合においても短時間で軌道生成を行う方法を提案する。また、実際に市販の産業用ロボットを対象として、モータの最大発生電流値、消費電力を低減する軌道の生成を行う。なお、本論文では、ロボットのハンド先端の軌跡が指定される、いわゆる CP (Continuous Path) 制御を行うロボットを対象とする。

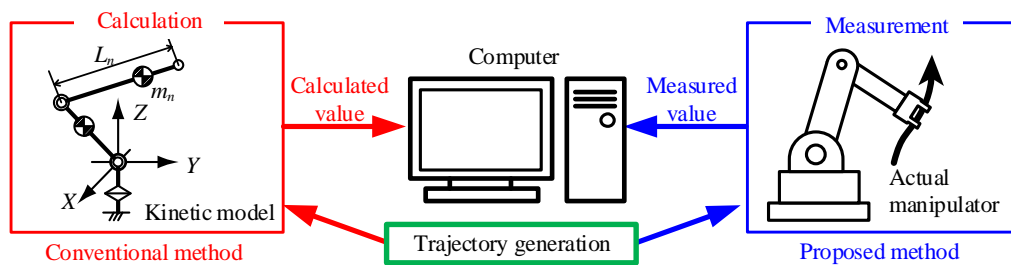


Fig. 1 Outline of generating trajectories

本論文で使用する主な記号は以下のとおりである。

- | | |
|-------------------------------|-------------------------------|
| C_{t_i} : 区間 i の通過に要する時間 | C_{v_q} : 分割した区間どうしの接続速度 |
| D_t : 動作指令領域の時間分割数 | D_v : 動作指令領域の速度分割数 |
| l_c : 速度を指定する区間の軌跡長の和 | l_i : 所要時間を変化させる区間 i の長さ |
| l_r : 動作開始から終了までの総移動量 | m_i : 無次元時間を決定するための探索値 |
| n_i : 無次元速度を決定するための探索値 | R_i : 区間 i の所要時間増減比 |
| s : 速度を選択する接続部分の数 | s_i : 区間 i の移動量 |
| S : 無次元総移動量 | S_i : 動作指令領域上の格子点 |
| St_i : 等速で移動時の区間 i の所要時間 | t : 時間の有次元量 |
| t_c : 速度指定区間通過時間と停止時間の総和 | t_i : i 番目の動作指令点が示す時間 |
| t_r : 動作開始から終了までに要する時間 | t_{ri} : 区間 i の所要時間 |
| T : 時間の無次元量 | T_b : 動作指令領域の時間幅の無次元量 |
| T_f : 終点と隣接する動作指令領域との無次元時間幅 | T_i : i 番目の動作指令点の表す無次元時間 |
| T_m : 動作指令領域間の時間幅の無次元量 | T_s : 始点と隣接する動作指令領域との無次元時間幅 |
| u : 所要時間を変化させる区間の数 | v : 速度の有次元量 |
| v_i : i 番目の動作指令点が表す速度 | V : 無次元速度 |
| V_i : i 番目の動作指令点が表す無次元速度 | V_m : 動作指令点の示す無次元最小速度 |
| w : 動作指令領域および動作指令点の数 | α_i : 速度指令値の補正係数 |

2. 軌道の生成法

2・1 生成法の概略

提案する手法は、ロボットを実際に動作させ、発生した電流値などを評価値として軌道を更新する操作を繰り返す。したがって、数値解析を用いる場合と異なり、ロボットを動作させるための時間とエネルギーを要する。よって、初期値を変化させながら多数回の繰り返し計算を行う非線形最適化法を用いることは適切でない。そこで、最適化には、S. Lin らが提案し⁽⁹⁾、著者らがマニピュレータの軌道生成⁽⁴⁾や機構設計⁽¹⁰⁾に有用であることを示した発見的手法 SHA (S. Lin's Heuristic Algorithm)を用いる。

SHA は探索する変数を図 2 に示すような行列で表す。行列の 1 から n 列までの各列は変数の種類を表し、各列の 1 から k 行までのそれぞれの要素は変数が取り得る離散値を示す。同行列を変数行列と呼ぶ。変数行列の各列から要素を 1 つずつ選択すれば、それらの変数の組合せが 1 組の解の候補となる。これを経路と呼ぶ。SHA では図 2 に示すように○で示す LBP(Look-ahead-base-point)と呼ぶ要素のある列においてランダムに選択して複数の経路を生成し、それぞれに対する評価値を求め、他の経路の評価値と比較する操作を繰り返すことでより良好な変数の組合せを探索する⁽⁴⁾⁽¹⁰⁾。

以上のように SHA は乱数を用いて解を探索するため、得られる結果は真の最適値であることは保証されない。しかし、非線形性の強い最適化問題に対し、比較的少数の繰り返し操作で最適値に近い解を探索し得る⁽⁴⁾⁽⁹⁾⁽¹⁰⁾ことから、本論文では同手法を用い、最適値に近い解が得られることを特定の例で確認した上で、探索結果を最適軌道とする。

2・2 軌道の表現方法

SHA を用いるためには軌道を数値表現する必要がある。本論文では作業中にロボットのハンド先端がたどる軌跡および作業開始から終了までに要する時間 t_r は指定する。したがって、ハンド先端の開始位置および動作中の加減速が探索の対象となる。開始位置は座標値を変数として表す。加減速に関しては、従来、高次多項式による運動曲線で表していた⁽⁴⁾。しかし、複雑な加減速を表そうとすると次数が増加し、決定すべき変数も増え取り扱いが困難となる。また、産業用ロボットの制御指令値を数式に厳密に一致するように生成することも容易でない。

そこで、本論文では産業用ロボットの動作指令プログラムに注目し軌道を表す変数を決定する。産業用ロボットの多くは、SLIM (Standard Language for Industrial Manipulators)⁽¹¹⁾などのロボット言語で記述された動作指令プログラムにより制御されている。SLIM による動作指令プログラムの例を図 3 に示す。①は速度、②では①の速度でハンド先端を移動させる軌跡の形状を LMOVE で、また、目標位置を TRANS に付記した座標で指定している。なお、LMOVE は直線形状を表す。これらの指令を適切な時間間隔でコントローラへ与えロボットを動作させる。本論文ではハンド先端がたどる軌跡はコントローラで指令可能な形状とする。よって、時間に対するロボットハンド先端の速度と座標を指定すれば、動作指令プログラムを生成できる。

以上より、ロボットの動作開始時からの時間 t と、その時のハンド先端速度 v の組合せで加減速を表現し、これらを表す変数を SHA の探索対象とする。ハンド先端の軌跡はあらかじめ与えるため、動作開始時のハンド先端の座標、時間 t でのハンド先端速度 v を指定すれば、簡単な演算でハンド先端の座標を求めることができ、動作指令プログラムを生成できる。その結果、時間に対するロボットの加減速が調整される。

t および v の探索範囲は、ハンド先端の時間に対する速度変化を図 4 に示すようにモデル化して設定する。横軸は無次元時間 T 、縦軸は無次元速度 V である。図 4 中の○印が t および v を決定する無次元量の組合せを表している。以後、本論文ではこれらの組合せ (T, V) を動作指令点と呼ぶ。図 4 中の●印は対象動作の開始および終了点であり、時間、速度はともに与えられる。具体的な設定方法を以下に示す。

まず、動作開始から終了までに指定する動作指令点の数 w を設定する。図 4 は $w=3$ の場合である。次に、探索する (T, V) の取り得る範囲を w 個の動作指令点に対してそれぞれ設定する。この範囲を動作開始から順に i 番目 ($i=1\sim w$) の動作指令領域と呼ぶ。なお、時間経過無しに動作指令を連続することはできないので、図 4 に示すように動作開始および終了時を含む隣接する動作指令領域の間には動作指令点を指定しない間隔 (T_s, T_f, T_m) を設ける。この間隔が広すぎると加減速の変化が制限されるため、その合計が動作時間全体の 10% 以下となるように、また、各動作指令領域の無次元化された時間幅 T_b は一定値として設定する。無次元量である速度の大きさ

V は最大値を 1.0 とし、条件として与えられる軌跡長、動作時間などを満たすように以下に示す方法で実際の速度の指令値 v に変換する。ただし、本論文ではあらかじめ指定されている条件以外でロボットを停止させないとし、 V の範囲には最小値 V_m を設ける。

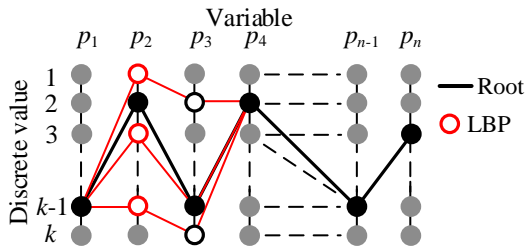


Fig. 2 Variable matrix

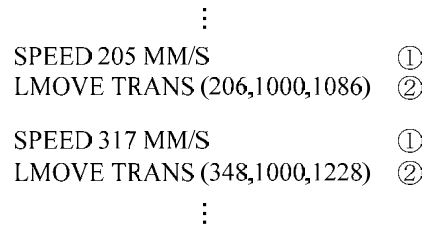


Fig. 3 Example of robot program by SLIM

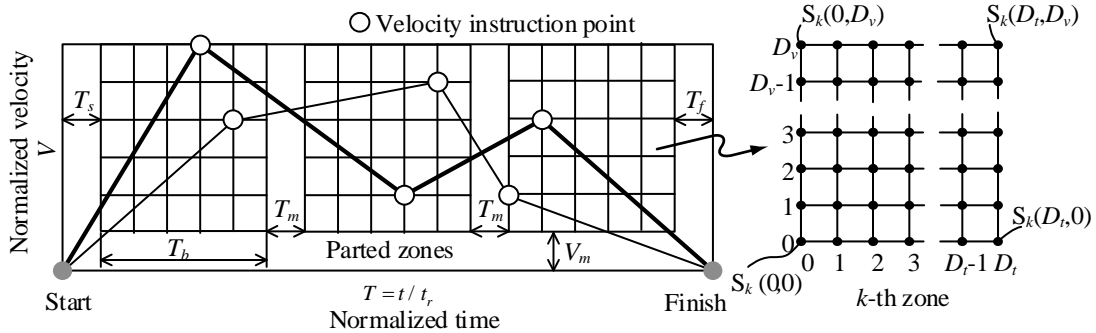


Fig. 4 Depiction of movements of a robot hand for SHA

さらに、以上で設定した動作指令領域を時間方向、速度方向にそれぞれ分割する。図 4 は時間および速度の分割数 D_t および D_v がともに 5 の場合を示している。図 4 の右に示すように、動作指令点は分割した領域内の格子点で指定する。各動作指令領域の左下の格子点を $S_i(0,0)$ 、右上の格子点を $S_i(D_t, D_v)$ とする。添え字は動作指令領域の番号を示す。1~ w 番目の動作指令領域で $S_i(m_i, n_i)$ をそれぞれ決定すれば、動作中のハンド先端の加減速の変化が決定される。すなわち、探索する変数 T, V の値は、 $m_1 \sim m_w, n_1 \sim n_w$ で表され、これらを取り得る値は、それぞれ $0 \sim D_t, 0 \sim D_v$ の範囲の離散値である。図 4 の太実線では 3 つの動作指令領域内で $S_1(3,5), S_2(3,1), S_3(1,3)$ が選択されている様子を示す。

以上で述べた動作指令点の探索に対して SHA で用いる変数行列は図 5 のようになる。同行列を用いて時間に対する速度の変化 $S_i(m_i, n_i) (i=1 \sim w)$ を決定する。得られた m_i, n_i より、図 4 に示す開始点からの時間 t と、その時のハンド先端速度 v の有次元量を算出する方法を以下に述べる。

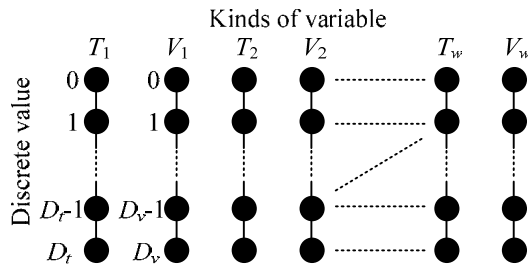


Fig. 5 Variable matrix for determining the operation variables of motion

まず、動作指令点の数が w であるときの各動作指令領域の時間幅の無次元量 T_b を上述の T_s, T_f, T_m より求める。なお、各動作指令領域の時間幅 T_b はすべて等しい。

$$T_b = [1 - T_s - T_f - T_m(w-1)] / w \quad (1)$$

次に、SHA で決定した i 番目の動作指令領域での時間に関する格子点位置 m_i から、 i 番目の動作指令点の表す無次元時間 T_i を次式で算出する。

$$T_i = T_s + T_b \cdot \frac{m_i}{D_t} + (T_b + T_m)(i-1) \quad (2)$$

式(2)で求めた i 番目の動作指令点の表す無次元時間 T_i に、開始点から終了点までの動作に要する時間の有次元量 t_r を次式のように乗じ、 T_i の有次元量 t_i を算出する。

$$t_i = T_i \cdot t_r \quad (3)$$

次に、SHA で決定した i 番目の動作指令領域での速度に関する格子点位置 n_i から、 i 番目の動作指令点の表す無次元速度 V_i を次式で算出する。

$$V_i = V_m + n_i(1 - V_m) / D_v \quad (4)$$

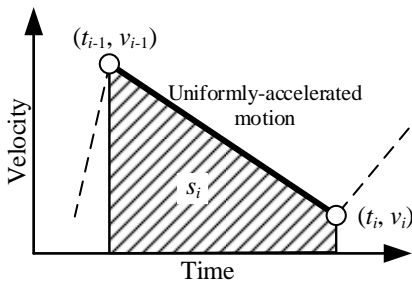
式(2)および式(4)で求めた T_i および $V_i (i=1 \sim w)$ より、開始点から終了点までの移動量の総和の無次元量 S を求める。なお、動作指令点間の速度は図 4 に例を示すとおり、加速度を一定として直線状に変化させることを想定するため、 S は次式で表される。

$$S = \sum_{i=1}^{w+1} \frac{(V_i + V_{i-1})}{2} (T_i - T_{i-1}) \quad (5)$$

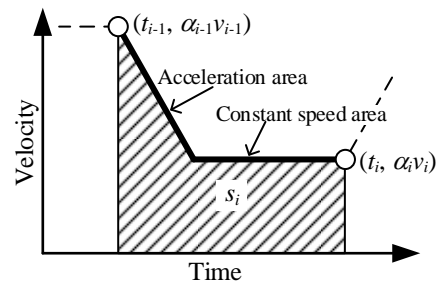
以上で求めた S の有次元量が開始点から終了点までの動作に要する時間 t_r の間の総移動量 l_r となるように、無次元速度 V_i の有次元量 v_i を次式で求める。

$$v_i = \frac{(l_r / S)}{t_r} V_i \quad (6)$$

なお、 V_0 , V_{w+1} , T_0 はあらかじめ与えられる値である。ハンド先端の座標は、以上で決定される T_i および V_i の実際の値 t_i および v_i を用いてあらかじめ指定されている軌跡に沿って求められる。さらに、これらの結果より動作指令プログラムを生成する。なお、以上で生成する (t_{i-1}, v_{i-1}) から (t_i, v_i) までの速度変化は図 4 のとおり図 6(a) に示す等加速度運動を想定している。しかし、本論文で用いるロボット言語による動作指令では、速度の大きさのみ指定可能で加速度は指定できない。実際に対象とするロボットを多数回動作させ測定した結果、加速度は常にほぼ一定であり、速度変化は図 6(b) のようになることを確認した。そこで、生成する動作指令プログラムにおいては、図 6 に示すとおり、時間 t_{i-1} から t_i にいたるまでの移動量 s_i が等しくなるように、速度 v_i を定数 α_i 倍し動作指令プログラムを補正する。なお、SHA による軌道探索時に図 6(b) のような速度波形を用いる方法も考えられるが、ここでは、将来的に様々なロボットのコントローラに対応するため、探索時には図 6(a) に示す単純な速度波形を想定し、動作指令値に変換する時点で補正する方法を用いることとした。



(a) Velocity variation on SHA



(b) Adjusting the velocity for robot program

Fig. 6 Adjusting the operation variable of velocity

2・3 軌跡の分割方法

ロボットの動作時の軌跡が長くなるにつれ、また、方向の転換、曲線経路などを含み複雑になるにつれ、ハンド先端の加減速を表す動作指令点の数が増え、変数行列の列数が増加する。SHA では変数行列の各列の値の組合せで解を探索するため、列が追加されると探索回数が指数的に増加する。提案する手法では探索のたびに実機を稼働させるため、その回数を減らす工夫が重要である。そこで、本論文では、ロボットに要求される動作を分割し、それぞれ独立した変数行列を用いて軌道探索を行う方法を提案する。これにより、軌道全体を1つの変数行列で表す場合に比べて、実機の動作回数が大幅に減り、探索時間を削減できる。

動作の分割はロボットの加減速が調整不可能または困難な、速度や軌跡の変化が指定される箇所で行う。具体的には以下のとおりである。

- (1) 停止箇所、速度が指定される区間
- (2) 軌跡の補間方法が変化する箇所

(1)の停止箇所、速度が指定される区間の開始箇所と終了箇所でのハンド先端の座標と速度は、動作開始位置が選択されれば決定されるため、これらの箇所の動作指令点(T, V)は一意に定められる。また、(2)の軌跡の補間方法が変化する箇所も、開始位置に対しハンド先端の座標は定まるため、その箇所での速度のみが調整可能である。すなわち、これらの箇所の動作指令点は図4中の開始および終了点と同じく横軸の T は固定値となり、停止箇所、速度指定区間の開始および終了箇所では V も固定され、軌跡補間方法の変更箇所では V のみが調整可能となる。

したがって、これらの箇所の前後の速度変化は、それぞれ図4に示すような動作指令領域を独立して設定し動作指令点を探索可能である。そこで、変数行列は、これらの各箇所で分割した軌道に対し設定する。ただし、分割した軌道の接続部は速度を一致させる。(2)の箇所のように接続部の速度が指定されない場合は同値をSHAであらかじめ選択する。さらに、隣接する区間の終点の時間を次の区間の開始時間とする。結果的に探索する速度変化の範囲は変数行列を分割しない場合と同じであり、決定され得る軌道の良好さは変わらない。

また、本論文ではロボットの動作開始から終了までの全体の所要時間を指定する。よって、各区間の所要時間の合計は一定とし、かつ、適切な加減速を決定するために、それぞれの区間の所要時間を変化させる。そこで、まず、指定される所要時間を満たすように、速度が指定される区間以外は等速で移動するとして各区間の移動に要する時間 St_i を算出する。すなわち、速度を指定する区間の軌跡長の和を l_c 、同区間の移動に要する時間および動作中に含まれる停止時間の和を t_c 、各区間の軌跡長を l_i とすれば、 St_i は次式で表される。

$$St_i = (t_r - t_c) \frac{l_i}{l_r - l_c} \quad (7)$$

次に、各区間の所要時間の増減比 R_i をSHAで探索し、式(8)で各区間の所要時間 Ct_i を求める。ただし、 Ct_i の和は必ずしも全体の所要時間に一致しないため式(9)で補正する。

$$Ct_i = St_i \cdot R_i \quad (8)$$

$$t_{ri} = Ct_i \cdot \left(\frac{\sum_{j=1}^u St_j}{\sum_{j=1}^u Ct_j} \right) \quad (9)$$

t_{ri} は各区間を移動するために必要な所要時間であり、 u は所要時間を変化させる区間の数である。

2・4 軌道探索の流れ

探索手順を図7に示す。Step1においてハンド先端の開始位置の座標(X, Y, Z)と、速度波形を探索する各区間の所要時間の増減比 R_i ($i=1 \sim u$)を選択する。 R_i より式(8)および(9)によって各区間の所要時間 t_{ri} ($i=1 \sim u$)を算出する。Step2で、以上の値を条件として各区間の接続部の速度 Cv_q ($q=1 \sim s$)、それぞれの区間の T_i, V_i ($i=1 \sim w$)を選択する。 w は図4で述べた動作指令領域の数、 s は速度を選択すべき接続部の数である。

次に、Step3において、選択した T_i, V_i を式(1)~(6)に代入し、動作開始時からの時間 t におけるハンド先端の速度 v と位置を算出する。これらを満たすように図3に示すような指令値を生成する。なお、 v の最大値がロボ

ットの最大出力速度を超える場合は、同探索結果を破棄して Step1 に戻り、再度、探索を行う。次に、Step4 において以上で生成した指令値でロボットを動作させ、評価値を得るための諸量を測定する。Step5 では評価値を算出して、これまでに得た最良評価値と比較し、より良好な値を得た場合は、その軌道を新たな最良軌道とする。

なお、本論文では、ロボットハンドの開始位置(X, Y, Z)をベースに固定した 3 次元直交座標で表す。また、動作指令点の探索を行う各区間の動作指令領域の数 w は等しく 3 であり、図 4 に示す T_s, T_m, T_f, V_m の値はいずれの区間においても 0.05, 0.025, 0.05, 0.3 とする。さらに、動作指令領域の分割 D_i, D_v も一定であり 20 とする。

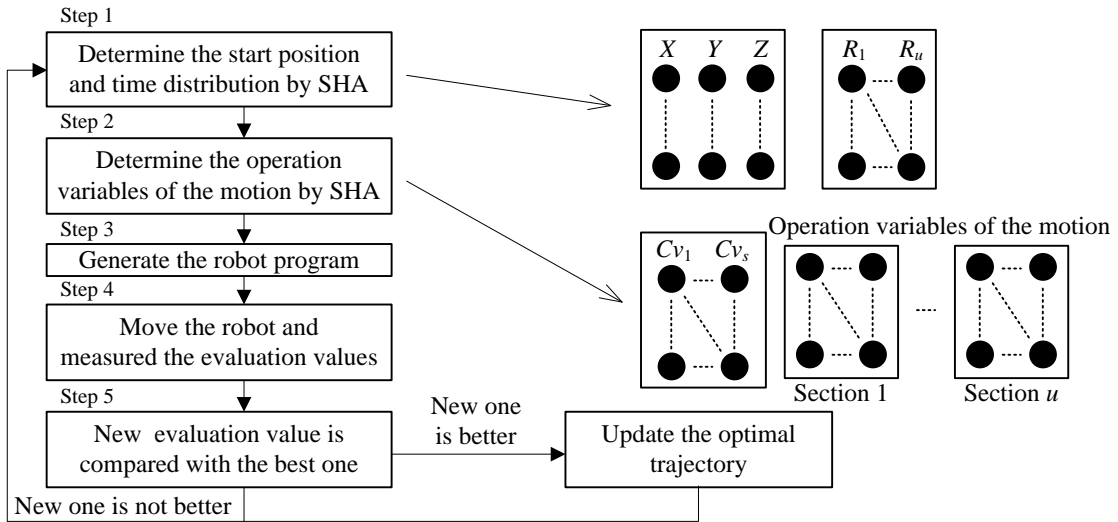


Fig. 7 Flow of searching the trajectories

2・5 SHA における処理

SHA の処理^{(4) (10)}の概略を以下に説明する。本手法では、軌道を探るために複数の変数行列を用いるが、ここでは図 8 に示す 1 つの変数行列を代表として用い処理の手順を説明する。まず、図 8 の左に示すように黒実線で示す経路を乱数で発生させ、基準経路とする。同経路によって決まる軌道により、ロボットを動作させ発生する電流などを測定し評価値を求める。次に、同図に○で示すように LBP を乱数で複数発生させ、各 LBP に連結する次列の変数の値を、乱数を用い選択する。次々列以降の変数の値は基準経路と同じである。次に、これら複数の経路で表される軌道を用いてロボットを動作させ、それぞれの軌道に対する評価値を求める。これらの評価値と基準経路の評価値を比較して、最も良好な評価値を得た軌道を新たに基準経路とする。

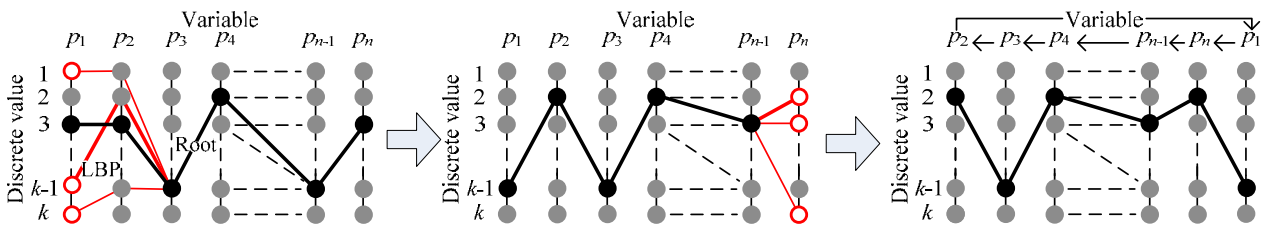


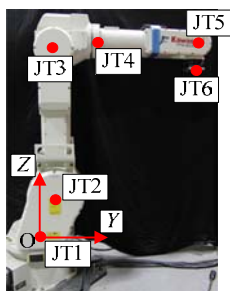
Fig. 8 Operation of the variable matrix

以上の処理を変数行列の左端から右端の列へ、1 列ずつ移動しながら順次実行する。なお、文献(4)では最終列において全ての要素を用いて軌道を生成し比較を行った。しかし、本手法ではロボットの動作回数を減らすため、図 8 中央に示すとおり最終列においても指定した LBP の個数で軌道を生成し比較する。また、これらの操作が図 8 中央のように最終列まで達した場合は、図 8 右に示すとおり変数行列の先頭列を最終列に移動させ、その他の列を前列へと移動させて新たな変数行列を構築し、その先頭列から以上の操作を繰り返す。

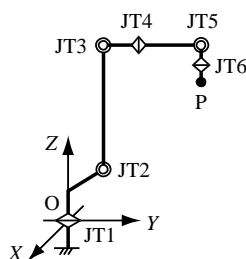
3. 軌道生成の対象

3・1 ロボットの諸元

対象とする産業用ロボットは、図 9(a)に写真、同図(b)に骨格図、表 1 に仕様を示す川崎重工業社製の垂直多関節形ロボット FS60L である。関節をベース側から順に JT1～JT6 と記す。ロボットの動作および軌道は JT1 の回転軸を Z 軸とする固定座標系 O-XYZ で表す。動作指令には SLIM 言語に準じている同社製の AS (Advanced Superior) 言語を用いる。



(a) Over view



(b) Skeleton Model

Fig. 9 Industrial manipulator FS60L

Table 1 Specifications of FS60L

Type	Vertically articulated
Degree of freedom	6
Maximum payload	60 [kg]
Repeatability	±0.15 [mm]
Mass	Approx. 585 [kg]

3・2 評価値

本論文では提案する手法を用い、モータの最大電流値、一次入力電源の仕事率をそれぞれ評価値として、これらを低減する軌道の生成を行う。それぞれの詳細は以下の通りである。

(1) モータの最大電流値

モータの発生電流値は駆動力の大きさと相関が高い。そこで、モータに流れる最大電流値を評価値とし、駆動力を抑制し得る軌道を生成する。本論文では、相対的に大きなトルクが発生する JT2, JT3 を対象とし、JT2 および JT3 に配置したモータに生じる電流値の和を各時刻で求め、その最大値を評価値とする。なお、各モータに流れる電流値はコントローラのモニタリング機能で測定する。

(2) 一次入力電源の仕事率

一次入力電源の入力電流値から求められる皮相電力を動作時間で除した仕事率を評価値とする。皮相電力と消費電力は比例関係に近いとして、この値を低減することで消費電力を抑制する軌道の生成を行う。しかし、3 相 3 線電源のモータは各線に流れる電流値が不均等であり、その比率も不定である。そのため、3 線全ての電流値から皮相電力を算出する。

4. 軌道生成例

4・1 軌跡および探索条件

図 10(a), 11(a)に探索を行う軌道 1, 2 でのハンド先端の軌跡 1, 2 をそれぞれ示す。黒実線が軌跡、赤矢印は移動方向である。いずれも動作開始から終了までの所要時間は、静止区間を含めて 10.5s とし、ハンド先端に 60kg の錘を取り付けてロボットを動作させる。図 10(b), 11(b)は 2・3 節で述べたように軌跡 1, 2 を分割した結果である。各区間の長さ、方向は表 2 に示す。なお、軌道 1 の区間 V および軌道 II の区間 IV の軌跡は円弧であり、表 2 には円弧が含まれる平面と半径を示した。

軌道 1 では区間 II の速度を 150mm/s とし、区間 III と IV の間に 0.5s の停止時間を設ける。区間 IV, V, VI では指令する軌跡の補間形状を変化させる。なお、区間 IV と V および区間 V と VI の接続部の速度の大きさは SHA で探索する。軌道 2 では区間 II の速度を 150mm/s とし、区間 II と III の間に 0.5s の静止時間を設ける。区間 III, IV, V で指令する軌跡の補間形状を変化させる。これらの区間の接続部の速度の大きさは SHA で探索する。さらに、区間 V と VI の間は速度を一旦、0 mm/s とし方向変換させる。

4・2 軌道生成結果

4・2・1 モータの最大電流値の低減

JT2 および JT3 のモータに生ずる電流値の絶対値の和の最大値を評価値として生成した軌道 1 および 2 を表 4 に示す。なお、表 4 に示す各区間の所要時間の増減比は、探索した最適軌道に基づき生成した動作指令プログラムにより実際にロボットが動作した時間をコントローラのモニタリング機能より求め算出した値である。軌跡 1 の区間 I の増減比が表 3 に示す SHA による探索時の最大値 1.2 を超えているが、これはロボットの加速度のばらつきに起因する図 6 に示した補正による誤差などが原因と考えられる。しかし、その差はわずかであり、ロボットの動作開始から終了までの所要時間も、条件として与えた 10.5s に対し、軌道 1 では 10.85s、軌道 2 では 10.35s とほぼ近い結果となっていることを確認しており、今回は許容範囲とする。

各軌道に関して生成結果を上段に、比較のための基準軌道の諸元をその下段に示す。また、図 12, 13 に軌道 1, 2 に対する SHA の探索での評価値の推移、基準軌道および生成された軌道での JT2 と JT3 のモータ電流値の和、ハンド先端速度の時間に対する変化を示す。なお、図 12(a)中の◇は各回の探索で得られた軌道による評価値を、赤実線はそれまでの探索で得られた評価値の最良値を示している。以下、評価値の探索回数に対する変化は同様を示す。

表 4 より生成した軌道 1, 2 の最大電流値は基準軌道と比較して、それぞれ約 26, 21%低減している。探索に要した時間はいずれの場合も約 30 分であった。ただし、図 12(a), 13(a)からわかるように、いずれの場合も、その半分程度の時間で最良値を示す軌道が得られている。

ここで、SHA による探索結果が最適値近傍であるかを、探索範囲を限定し確認する。すなわち、軌跡 1 を用い、速度の加減速は基準軌道と同一として、探索範囲全域において開始位置のみを表 3(a)の刻み幅で変化させ、各開始位置に対する評価値を測定した。その結果、評価値が最小となる開始位置(X, Y, Z)は(-30,660,1320)となり、表 4 に示す軌道 1 の結果とほぼ一致した。また、測定された最小評価値は 14.1A となり、軌道 1 に比べ大きな値となった。これは、軌道 1 では加減速も考慮しており、その効果の差であると考えられる。また、以上の測定には約 2000 分と長時間を要したが、提案手法では、その約 2%の時間で加減速も考慮した軌道の生成が行えており、その有用性が確認される。また、著者らが先に動力学モデルを用い SHA によって軌道の探索を行った質量 1600kg、ハンド先端の負荷が約 300N である 5 自由度マニピュレータでのモータの最大電流値の低減率は、同機種が標準装備していた軌道に対し約 25%であった⁽⁴⁾。機種、軌道の条件などが異なるため単純に比較はできないが、構築が困難な動力学モデルを用いない場合でも同程度の結果が得られていることから本手法の実用性が期待できる。

図 12(b)より、軌道 1 の探索結果では電流値が全体的に低下している。これは、主に軌道の開始位置を本体に近い JT2 近傍として、ハンド先端荷重およびロボットの自重による駆動トルクを抑えたためである。また、基準軌道で、電流値がピークを示す 0~1s および 6~7s 付近において、軌道 1 では図 12(c)に示すようにハンド先端の速度を階段状に変化させ加速度を軽減し、電流値をさらに抑制していると考えられる。また、これらの範囲の速度低下による動作時間の増加は、重力方向にアームを振り下げるため大きな力を必要としない、4~5 および 10~11s 付近の増速によって補っている。

Table 4 Results of generated trajectories

Trajectories		Start position [mm]			Ratio of change of time distribution						Evaluation value [A]	Reduction rate [%]
		X	Y	Z	I	II	III	IV	V	VI		
1	Generated	-60	600	1230	1.21	----	0.96	1.01	0.98	0.84	12.6	25.9
	Standard	-300	900	1080	1.00	----	1.00	1.00	1.00	1.00	17.0	----
2	Generated	90	600	840	1.07	----	0.91	0.89	0.97	1.16	14.3	20.6
	Standard	0	900	1080	1.00	----	1.00	1.00	1.00	1.00	18.0	----

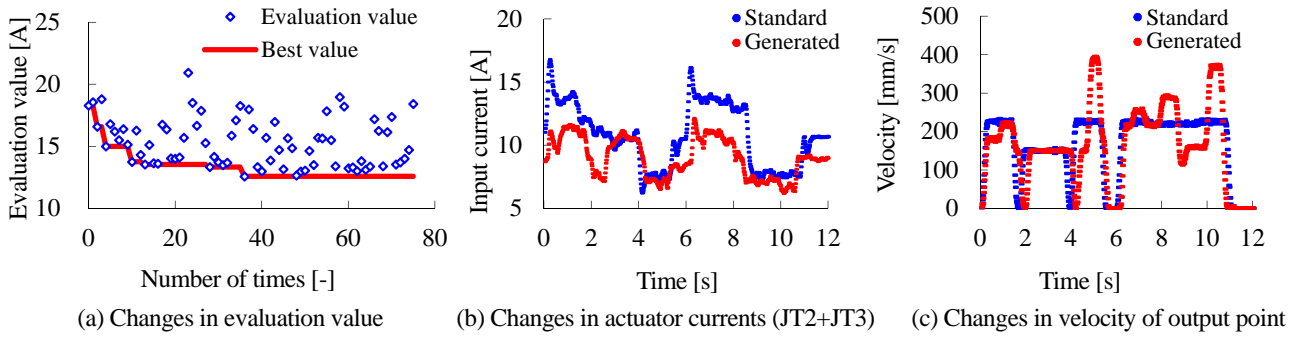


Fig. 12 Trajectory 1, Evaluation value: peak current value

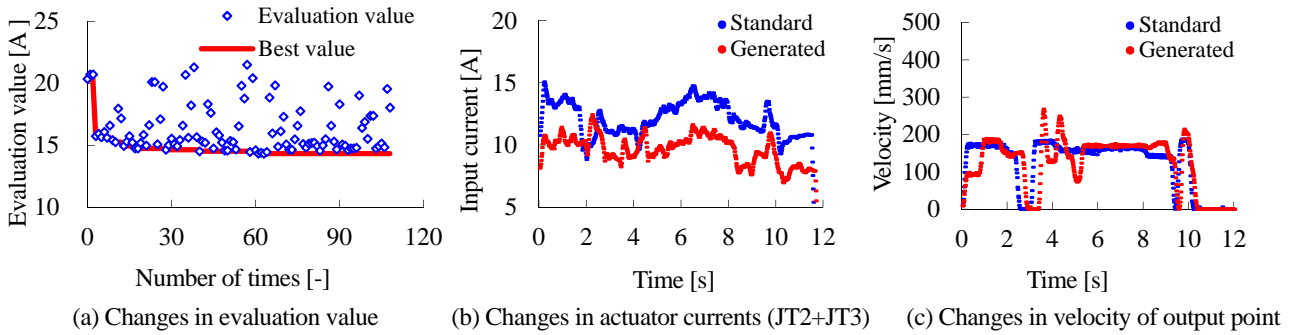


Fig. 13 Trajectory 2, Evaluation value: peak current value

図 13(b)より、軌道 2 の探索結果においても電流値が全体的に低下している。これは軌道 1 と同様に、軌道の開始位置を JT2 近傍として、駆動関節のトルクを抑えたためである。ただし、図 13(c)に示すように速度波形は、4s 前後で急上昇が見られるが、全体的に台形状となっており、基準軌道に類似した結果となっている。軌道 2 では水平方向の動作区間が多く、重力の影響が少ないため、加減速の少ない台形状の速度波形を選択し、慣性力の影響を軽減することで電流値の増加を抑制したと考えられる。

4・2・2 一次入力電源の仕事率の低減

評価値を一次入力電源の仕事率として生成した軌道 1 および 2 の結果を表 5 に示す。なお、探索された表 5 の軌道 1, 2 とともに区間 I, II~VI の所要時間の増減比の和の実測値は前項で述べた理由により 5.0 となっていないが、差はわずかであり、実際のロボットの動作開始から終了までの時間も指定した 10.5s に対し軌道 1 は 10.35s, 軌道 2 は 10.13s と近かったことから今回は許容範囲とみなす。また、図 14, 15 に軌道 1, 2 の探索での評価値の推移、基準軌道および生成された軌道での一次電源の電流値、ハンド先端速度の時間に対する変化を示す。なお、一次電源の電流値は、各時間での 3 線の電流値の和を $\sqrt{3}$ で除した値である。表 5 より軌道 1, 2 は、基準軌道と比較して仕事率がそれぞれ約 5, 6% 低減している。探索に要した時間はいずれの軌道においても約 30 分であったが、図 14(a), 15(a)に示すとおり、最良値は半分以下の時間で得られている。

なお、前項と同じく SHA による探索結果が最適値近傍であることを確認するため、軌道 1 に関して探索範囲全域において開始位置のみを変化させて評価値の測定を行った結果、その値が最小となる開始位置は (-300,960,1320) となり、前項と同様に生成結果とほぼ一致し、測定値は 0.80kW と SHA による探索結果に比べわずかであるが大きな値となった。これは前項で述べたように、SHA による探索では加減速も考慮されているためと考えられる。

図 14(b), 15(b)より、軌道 1, 2 とともに、基準軌道に比べ若干ではあるが全体的に電流値が減少している。図 14(c)に示すように軌道 1 では基準軌道と比較して 8s 付近でハンド先端速度が小さくなっている。このとき、ハンド先端は区間 V の円弧頂点付近を通過し、移動方向が上昇方向から下降方向へ変化する。ハンド先端の負荷荷重が大きいため、速度を低下することで方向変換時の慣性力を低減したことが、消費電力量の低減に効果があったと考

えられる。また、図 15(c)より、軌道 2 では、重力の影響が大きい 0 から 2s の区間で基準軌道より若干低速に動作し、水平区間である 5 から 8s を高速に動作することで消費電力を抑制していると考えられる。

しかし、前項で示したモータの最大電流値を評価値とした場合に比べ、仕事率の低減率は低い結果となった。これは、図 14, 15 からわかるように、軌道を変更しても一次電源の入力電流値にほぼ変化が見られないことが原因である。実際に開始位置を変化させ評価値を測定した先述の結果においても、一次電源側の電流値の変化は少ないことを確認した。しかし、4・2・1 項で示したように、モータの発生電流値は軌道により大きく変化する。これはロボットの動作に関係なく、コントローラが安定した制御を実現するため大きな電流値を必要としていることなどが原因と考えられる。これらの検証とコントローラを含めた最適化に関しては今後の検討課題としたい。

Table 5 Results of generated trajectories

Trajectories		Start position [mm]			Ratio of change of time distribution						Evaluation value [kW]	Reduction rate [%]
		X	Y	Z	I	II	III	IV	V	VI		
1	Generated	-270	990	1290	0.97	----	1.06	0.95	1.06	0.97	0.78	4.9
	Standard	-300	900	1080	1.00	----	1.00	1.00	1.00	1.00	0.82	----
2	Generated	-30	660	930	1.00	----	0.95	0.89	1.04	1.10	0.83	5.7
	Standard	0	900	1080	1.00	----	1.00	1.00	1.00	1.00	0.88	----

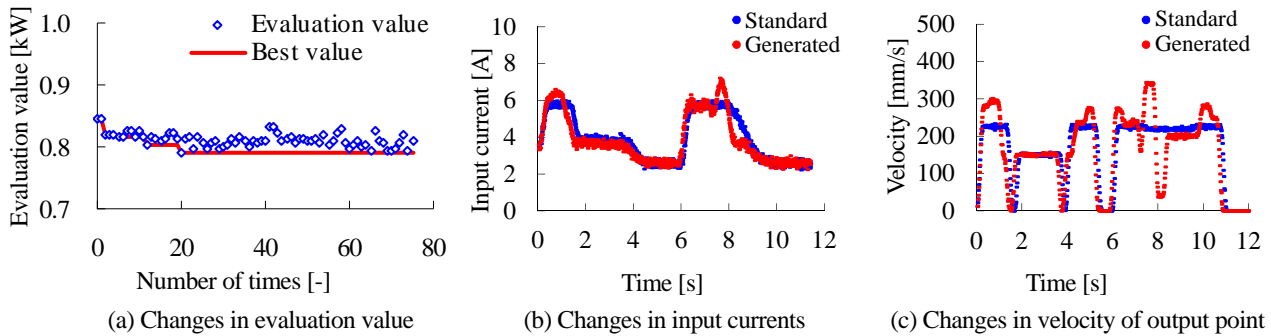


Fig. 14 Trajectory 1, Evaluation value: electrical energy

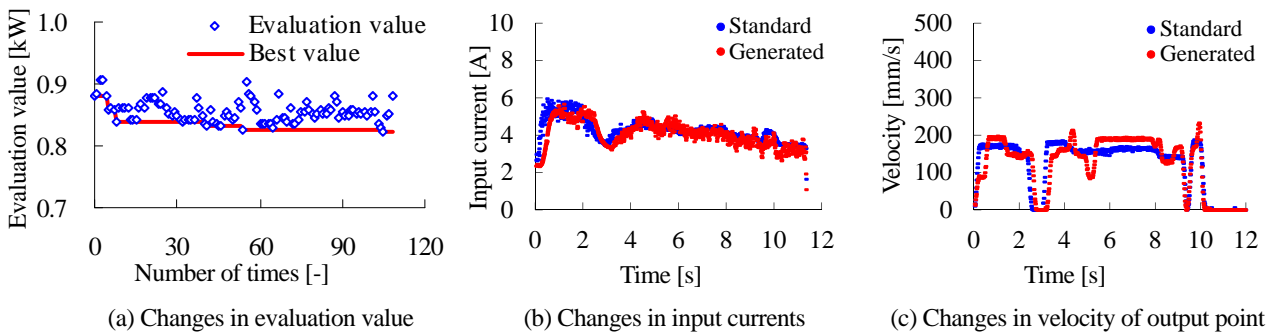


Fig. 15 Trajectory 2, Evaluation value: electrical energy

5. 結 言

本論文では、CP 制御で動作する産業用ロボットを対象として、モータに必要な電流や消費電力を抑制する軌道を、ロボットを実際に動作させて得る電流値などの測定結果を評価値として、発見的手法を用い生成する方法を提案した。同方法は、ロボットの動力学モデルを必要とせず、探索する変数は汎用的なロボット言語に対応しているため実用性が高い。実際に、市販の産業用ロボットを対象に同方法で軌道生成を行い、モータの電流値、仕事率を抑制する軌道をそれぞれ短時間で生成し、その有用性を確認した。

今後は、開始位置と終点位置の間の軌跡を任意とする PTP (Point To Point) 制御で動作させるロボットを対象に、より自由度の高い軌道の最適化を実現する。また、動作速度、位置決め精度など、多様な評価値を用いた軌道生成を試みる。

謝 辞

本研究は、財団法人スズキ財団のご支援を受けました。ここに記し、深く感謝の意を表します。

文 献

- (1) 立矢宏, 松井良雄, 村川俊昭, 秋野晋也, “動力学特性図によるロボット用開ループ機構の評価”, 日本機械学会論文集 C 編, Vol. 63, No. 606 (1997), pp. 613-619.
- (2) 加藤正, “マニピュレータの駆動方法とエネルギー消費特性”, 日本ロボット学会誌, Vol. 26, No. 3 (2008), pp. 271-276.
- (3) 永尾真史, 岩村誠人, 尾崎弘明, “剛体リンク系の汎用的な最適制御アルゴリズム (駆動冗長性を有する閉リンク系への拡張)”, 日本機械学会九州支部講演論文集, Vol. 62 (2009), pp. 129-130.
- (4) 林道大, 立矢宏, 浅川直紀, “発見的手法を用いた多自由度産業用ロボットの軌道決定 (鋼板搬送用マニピュレータの動的トルク抑制の実現)”, 日本機械学会論文集 C 編, Vol. 75, No. 750 (2009), pp. 262-269.
- (5) 阿部晶, 笹森和典, “フレキシブルマニピュレータの最適軌道計画—駆動エネルギーと残留振動を最小化するためのアプローチ—”, 計測自動制御学会論文集, Vol. 46, No. 2 (2010), pp. 130-132.
- (6) 佐藤浅次, 佐藤治, 河野通夫, 甲斐恵子, “重力下で PTP 制御される 2 自由度 D.D.マニピュレータの省エネルギー軌道”, 精密工学会誌, Vol. 69, No. 9 (2003), pp. 1281-1285.
- (7) 大谷幸司, 柿崎隆夫, 中村三男, “多関節マニピュレータの動力学パラメータ同定に関する実験的検討”, 日本ロボット学会誌, Vol. 11, No. 7 (1993), pp. 1083-1092.
- (8) 藤本真作, 小野敏郎, 逢坂一正, “挿入型 GA の提案とロボットマニピュレータの基底パラメータ推定への応用”, 日本機械学会論文集 C 編, Vol. 74, No. 739(2008), pp. 633-641.
- (9) Lin, S., Kernighan, B.W., “An effective heuristic algorithm for the Traveling-Salesman Problem”, *Operations Research*, Vol. 21, No. 2 (1973), pp. 498-516.
- (10) 立矢宏, 河合秀幸, “複数の機構特性の評価に基づく用途に応じたロボット動作機構の総合”, 日本機械学会論文集 C 編, Vol. 67, No. 655 (2001), pp. 880-887.
- (11) 新井民夫, “JIS 使い方シリーズ産業用ロボット言語“SLIM””, 日本規格協会, (1994), pp. 132-154