技術論文

産業用多関節形ロボットアームの相互位置同期制御を図った高精度輪郭制御法

前田 貴信^{*1} 中村 政俊^{*1} 久良 修 郭^{*2}

A Method for Contour Control of Industrial Multi Link Robot Arm using Synchronous Position of Mutual Axes

Takanobu Maeda^{*1}, Masatoshi Nakamura^{*1} and Nobuhiro Kyura^{*2}

A method for synchronous position control of mechatronic servo system of multiple axes was required to attain a high accuracy performance of contour control in sealing process, laser cutting and so on. Accurate contour control method for multi link robot arm was proposed by using synchronous position of mutual axes. The proposed method was evaluated by experiment of actual multi link robot arm and its simulation study under disturbed torque circumstance, and then showed satisfactory performance.

Key Words: Robot Arm, Contour Control, Synchronous Position Control, Error Feedback

1. ま え が き

シーリング作業やレーザカッティング等の例が示すように、産業界において、多関節形ロボットアームによる輪郭制御は生産技術の重要な位置を占めている。輪郭制御は、作業座標上の目標軌跡に沿って制御対象であるロボットアームの手先の位置を追従させる制御である。この場合、ロボットの姿勢による動特性の変化や各関節間の相互干渉等のサーボ機構に対する外乱が生じるので、ロボットの動作速度、位置誤差などに対する要求が厳しくなるにつれて、現在産業界でとられている制御法の各軸独立な制御方法では、ロボットアームの先端を目標軌跡に正しく追従させることは難しくなる。したがって、今後の産業用ロボットの制御方法としては、各関節間の相互干渉や動特性変化を考慮した制御系設計が必要となる[1]。その手段として、外乱オブザーバを用いて各関節間を非干渉化して線形制御する方法[2]や、各関節の角度を関節座標系で表して逆ダイナミクスモデルを介在してオンラインに目標軌道を生成する手法[3][4]、最適制御を用いた学習によってモデルのパラメータを求め、制御性能の向上を図る手法[5]、各関節の角度からオブザーバによって制御系の構成を可変して、位置誤差と速度誤差を減少させる手法[6]などがある。しかしながらこれらの方法では、非干渉化を行う手法は複雑な計算を要するので計算処理能力の高い計算機を必要とし高コストとなるために実用化は困難であり、また、モデル化誤差やロボットの手先位置が動作する作業座標系での評価を行っていないなどの問題が残る。

筆者らは、直交座標系の産業用ロボットアームの輪郭制御法として、二次元空間の作業座標上のX軸、Y軸方向の位置出力の同期を図る方法を提案した[7]。本研究では、産業用多関節形ロボットにおける高精度輪郭制御の実現を目指して、先に示した二次元直交空間での輪郭制御法を三次元多関節空間に拡張した。本提案法は作業座標各軸間の相互誤差をフィードバックするだけの非常に簡単なアルゴリズムを有するため、既存のロボットのサーボ部分の計算機的能力で十分に実行可能で、しかも得られる輪郭制御性能は著しく高いものとなる。この効用を実際の多関節形ロボットアーム(Performer MK2)に適用して輪郭制御実験を行い、その結果から本提案法の有効性を確認した。したがって、産業用ロボットのサーボが各軸独立に制御されている現状を覆して、輪郭制御性能向上のためには本手法によって各軸相互に位置同期を図ることを推奨するものである。

2. 産業用ロボットアームを用いた輪郭制御の問題点

輪郭制御は、制御対象であるロボットアームの手先の位置を、指定された目標速度を保ちながら、X、Y、Z軸で表される作業座標空間に示された目標軌跡に沿って最終目標点まで移動させるものである。直線や曲線の組み合わせからなる目標軌跡は、サーボの指令信号としては一般に $(r_{x0}, r_{y0}, r_{z0}) \rightarrow (r_{x1}, r_{y1}, r_{z1}) \rightarrow (r_{x2}, r_{y2}, r_{z2}) \rightarrow \dots \rightarrow (r_{xn}, r_{yn}, r_{zn}) \rightarrow (r_{xn+1}, r_{yn+1}, r_{zn+1}) \rightarrow \dots$ という具合に、位置情報に関する多角形近似された直線の組み合わせで与えられる。

現在産業界で使われている多関節形ロボットアームにおいては、ロボットを構成する関節部分のモータの位置制御は、各関節のモータごとに独立に行われている。しかし、各モータを独立に制御すると、ロボットアームの動作中に機構部の姿勢に

原稿受付 1996年4月1日

^{*1}佐賀大学理工学部

^{*2}近畿大学九州工学部

^{*1}Faculty of Science and Engineering, Saga Univ.

^{*2}Faculty of Engineering, Saga Univ.

よって重力が変化して手先の負荷が変わり、また動作速度により遠心力や慣性力等が生じ、それらがすべてサーボ系の特性変化になるので、目標軌跡に手先が正しく追従不可能となる。また、PUMA 形ロボットアームのようなシリアルリンクの場合には、一つのアームで生じた誤差がそのアームに接続されているアームに影響し、自由度が高くなるほどにロボットアームの手先での誤差が増幅される。作業内容の面では、グライディング加工やバリ取りのように工具が被加工物に接触するときにモータ側へ加わる反力等の外乱が生じる場合にも、ロボットの手先位置は目標軌跡からずれる。また、制御の側面から見ると、ロボットアームの手先の位置は作業座標系上であるが、実際に制御されるのは関節座標系で表される各関節のサーボ系である。よって、手先位置を制御するために関節座標系から作業座標系への非線形変換を行うが、そのときの計算誤差によって目標軌跡から各サーボ系へ与える位置指令に誤差が生じ、その結果、追従軌跡と目標軌跡が一致しなくなる。以上のように、ロボットアームの輪郭制御を行う場合には解決すべき問題が多くあり、これらが製品の品質に大きく影響を及ぼす。よって、現在産業界でとられている各軸独立の輪郭制御方法には、それがどんなに高度な制御法であろうとも限界がある。

産業用ロボットの制御の将来的な方向性としては、作業座標系上の X, Y, Z 軸方向への位置出力を各方向ごとに独立に制御するのではなく、互いに干渉しあうことを積極的に受けとめ、それぞれの軸の位置出力が他の軸の位置出力と綿密に連繋して同期を図るようにモータの制御を行うことが必要となる。その際に重要なことは、現存のロボットアーム等実際に適用できるかである。新しい制御法を考案しても、それを実際の作業現場で用いることができないならば、その価値は低く、また適用できても制御精度が低くは意味がない。すなわち、現存の機器をそのまま用いて、その変更は少ない方がよい。本研究では、このことを踏まえて輪郭制御に位置同期制御の概念を積極的に取り入れて輪郭制御の制御性能の向上を図った。

3. 多関節形ロボットの位置同期を図った高精度輪郭制御法

3.1 三次元空間での位置同期を図った高精度輪郭制御法

多関節形ロボットアームの輪郭制御を、三次元空間の位置同期によって図る。前章で述べたように、輪郭制御性能を向上させるには、作業座標系上の X, Y, Z 軸の位置出力の同期を図ることが必要である。

提案する相互位置同期を図った輪郭制御法は、比例関係にある二つの軸の位置出力の差に適当なゲインを乗じてそれぞれの位置制御の入力部にフィードバックすることで、その差を減じさせる方法である。

Fig. 1 に提案する相互位置同期を図った輪郭制御法の流れ図を示し、以下図に従って、提案法の実行手順を示す。

3.1.1 位置入力の実算

輪郭制御のロボットの位置入力は、与えられた目標軌跡とその移動する目標速度から、制御時間間隔ごとの目標位置として与える。

Fig. 1 Block diagram of contour control for a robot arm of three degrees of freedom

Fig. 2 A straight line of robot of objective in three dimensional space

輪郭制御の目標軌跡は、一般的に直線と曲線の組み合わせからなるが、産業用ロボットでは通常それらの目標軌跡を細かな区間の直線によって近似して、それをロボットに目標位置として指令を与える方法がとられている。**Fig. 2** に 1 区分の直線の指令信号の構成のされ方を示す。ロボットアームの手先位置が、この直線の始点から目標一定速度 v [m/s] を保ちながら 1 区分の直線区間の終点までを移動することを目標とする。よってこの直線は、ランプ入力 $u(t) = vt$ で表現される。これを X, Y, Z 軸に分解すると

$$\begin{aligned} u_{xo}(t) &= \sin \phi_{z-xy} \cos \phi_{x-y} u(t) \\ &= c_x u(t) \end{aligned} \quad (1)$$

$$\begin{aligned} u_{yo}(t) &= \sin \phi_{z-xy} \sin \phi_{x-y} u(t) \\ &= c_y u(t) \end{aligned} \quad (2)$$

$$\begin{aligned} u_{zo}(t) &= \cos \phi_{z-xy} u(t) \\ &= c_z u(t) \end{aligned} \quad (3)$$

で表され、 $u(t)$ に比例係数 c_x , c_y , c_z をかければよい。ただし、 ϕ_{z-xy} は目標軌跡と Z 軸がなす角度、 ϕ_{x-y} は目標軌跡を XY 平面に投影した直線が X 軸となす角度である。

この $u_{xo}(t)$, $u_{yo}(t)$, $u_{zo}(t)$ をそのまま用いる方法を従来法とし、後の章で提案法と比較する。

3.1.2 三次元空間から二次元空間への変換

位置同期を図るにはまず、この近似直線軌跡を Fig. 2 に一点鎖線の矢印で示すように、目標軌跡 $u(t)$ から XY, YZ, ZX の三つの二次元平面上にそれぞれ垂線を引き、投影する。これは、適用する相互位置同期制御が 2 軸の位置出力を対象としているため、三つの二次元空間の座標でそれぞれ位置同期を図る。図中の $u_{xyo}(t)$, $u_{yzo}(t)$, $u_{zxo}(t)$ は、それぞれ XY, YZ, ZX 座標に投影した位置入力である。

3.1.3 位置同期のための座標変換

相互位置同期制御法は、比例関係にある 2 軸の位置出力の差を求め、適当なゲインを乗じて位置制御系の位置入力へフィードバックすることで実現される。そのためには、二つの軸の位置出力は絶えず比例関係が保たれている必要があるが、2 軸のうちどちらか一方の方向にしか移動しないような目標軌跡の場合は適用できない。そこで、どのような目標軌跡であっても同期を図れるように、Fig. 3 に示すような相互位置同期を図りやすい新しい座標系を設け、その座標上に位置出力を変換をして、位置同期を図る。新たな座標は、二次元空間に変換したそれぞれの座標系について存在するが、Fig. 3 にはそのなかの XY 座標とその変換後の新しい座標系 X'Y' 座標について説明する。

XY 座標上で、投影した目標軌跡の直線の 1 区分が 45° の角度で二等分線となる新たな座標 X' 軸、Y' 軸を設ける。X'Y' 座標上での手先の位置 $(x'(t), y'(t))$ は、Fig. 3 のように、XY 座標上の位置 $(x(t), y(t))$ から X' 軸、Y' 軸に垂線をおろすだけで変換できる。座標変換後の X' 軸、Y' 軸の位置入力、位置出力は

Fig. 3 Coordinate transformation for synchronous position control

$$x'(t) = \{(\tilde{x}(t) - \tilde{y}(t))c_{x'} + (\tilde{x}(t) + \tilde{y}(t))c_{y'}\} / \sqrt{2} \quad (4)$$

$$y'(t) = \{(\tilde{x}(t) + \tilde{y}(t))c_{x'} - (\tilde{x}(t) - \tilde{y}(t))c_{y'}\} / \sqrt{2} \quad (5)$$

$$u_{x'o}(t) = u_{y'o}(t) = u_{xyo}(t) / \sqrt{2} \quad (6)$$

となる。ここで、 $c_{x'} = \cos \phi_{x-y}$, $c_{y'} = \sin \phi_{x-y}$ であり、 ϕ_{x-y} は、目標直線と X 軸とがなす角度である。このとき、 $u_{x'o}(t)$ と $u_{y'o}(t)$ は 1:1 の比例関係を持つことになる。以下、YZ, ZX 座標上でも上記 XY 座標上の変換と同様の作業を行う。

3.1.4 相互位置同期制御法

前節で設けた変換後の座標上で、相互位置同期制御法を用いて二つの軸ごとの位置同期を図るよう位置入力を修正する。そのためにまず、位置出力の差を求める。X'Y' 座標上で、位置出力 $x'(t)$, $y'(t)$ の差をとり、その値に適当なフィードバックゲイン α を乗じたものが、位置制御信号となる。このフィードバックゲイン α の決定法については付録で述べる。求めた位置同期信号を変換座標系上の各軸の位置入力へフィードバックする。その結果、X'Y' 座標での X' 軸、Y' 軸の位置入力は

$$u_{x'}(t) = u_{x'o}(t) - \alpha (x'(t) - y'(t)) \quad (7)$$

$$u_{y'}(t) = u_{y'o}(t) - \alpha (y'(t) - x'(t)) \quad (8)$$

となる。式 (7) (8) は、位置の誤差をフィードバックするだけで、モデルの同定等の面倒な計算を必要とせず、どのようなシステムにおいても簡単に用いることができることを意味する。この位置同期精度に関する理論的解析は文献 [7] にて行っている。

3.1.5 作業座標系への逆変換

3.1.4 項で求めた位置入力を、もとの作業座標系に変換する。それには今までの逆の手順を追って、変換座標上の位置入力を、もともとの XY, YZ, ZX 座標上の値に逆変換する。次に、三つの二次元座標での位置入力を合成し、三次元空間の XYZ 座標上での X 軸、Y 軸、Z 軸の位置入力を求める。位置入力の合成は、X 軸に関して述べると、二次元空間の XY 座標と ZX 座標の位置入力をたし合わせることで、もとの三次元空間の X 軸の位置入力が求められる。Y 軸、Z 軸に関しても同様にする。その結果、X, Y, Z 軸の位置入力は最終的に

$$u_x(t) = u_{xo}(t) - \alpha (e_{xy}(t) - e_{zx}(t)) \quad (9)$$

$$u_y(t) = u_{yo}(t) - \alpha (e_{yz}(t) - e_{xy}(t)) \quad (10)$$

$$u_z(t) = u_{zo}(t) - \alpha (e_{zx}(t) - e_{yz}(t)) \quad (11)$$

となる。ここで、 $e_{xy}(t)$, $e_{yz}(t)$, $e_{zx}(t)$ は、XY, YZ, ZX の各座標での二つの軸の誤差である。

以上のようにして求めた位置同期を図った位置入力は、計算時間を短縮して現実的なものにするために、以上の座標変換や位置同期信号の計算を一括して行うようにプログラムを作成して用いる。

3.1.6 目標軌跡の角の接点部分の処理

目標軌跡の直線部分は、以上の方法で位置同期を図ることで制御性能を向上させることができるが、近似した直線成分の角

Fig. 4 Synchronous position control on the curve

の接点付近では、Fig. 4 に破線で示すようにサーボ系の追従遅れによって追従軌跡に劣化が生じる。この問題の最も簡単な解決策は、各目標直線軌跡を追従遅れの分だけ延長することである。すなわち、通常、位置入力为目标直線軌跡の終端に達した時点で次の目標直線軌跡に移り、係数 c_x , c_y , c_z が変わるところを、 c_x , c_y , c_z をそのままにして X, Y, Z 軸方向の位置入力を与え続けると、位置入力を延長することができる。位置入力の延長は、サンプリング間隔 Δt [s] ごとに観測した手先位置が、目標直線軌跡の終端を中心とした半径 $v\Delta t$ [m] の円内に入るまでとする。このようにすることで、サーボ系の追従遅れによる軌跡劣化を減少できる。

手先位置が目標直線軌跡の終端に達した後、次の目標直線軌跡を位置入力として与えるが、到着直後に位置同期を図ると、観測精度による軌跡誤差が存在する場合に Fig. 4 の一点鎖線のように振動が生じる可能性がある。そこで、直線軌跡に追従を開始するときに、最初から位置同期を図るのではなく、誤差フィードバックゲイン α を

$$\alpha(t) = \alpha_0 (1 - e^{-at}) \quad (12)$$

と時間的な指数関数にして、徐々に位置同期を図るようにする。ここで、 α_0 は本来設計した誤差フィードバックゲインの値で、 t は各直線軌跡に対して追従し始めてからの時間、 a は $\alpha(t)$ が α_0 に収束する時定数である。このようにすると、目標軌跡の直線成分の接点での適切な処理が施される。

3.2 制御パラメータの設計

誤差をフィードバックするゲイン α_0 の値を大にするほど各軸の位置同期は強く働くが、あまりに大きな値をとると、位置制御部がサンプリング間隔 Δt [s] ごとに計算を行っているために量子化誤差が生じ、制御系のゲインが大きいほどにその誤差も増幅され、制御系が不安定になる。よって、安定に輪郭制御を行うためには、位置制御部の処理を行う計算機の性能によって設定できる α_0 の上限値をあらかじめ知る必要がある。

ここでは、 α_0 の簡単な決定法として、サンプリング間隔ごとの位置制御をオイラー法による数値計算と等しいとみなして、すでに知られているオイラー法で計算する場合の発散条件 [7] をもとに決定式を導く。そのために、まず各関節を制御するサーボ系を線形一次系のモデル [8] で表し、次に関節座標系での位

置を直交座標系の位置で近似した式をモデルにあてはめて、位置同期制御を行う場合の位置の特性方程式を求め、その極と、デジタル制御のサンプリング間隔 Δt [s] から、制御系が発散しないような関係を求めると、 α_0 の設計目安を得ることができる。3 自由度多関節形ロボットアームの例をあげると、関節軸 1, 2, 3 のサーボ系が等しく線形一次系に近似できる場合、その一次近似の比例ゲインを K [1/s] とすると、フィードバックゲイン α_0 は

$$\alpha_0 < \{(1/K\Delta t) - 1\} / 2 \quad (13)$$

となる (付録参照)。

式 (13) をみると、 α_0 は各関節軸の比例ゲイン K [1/s] とサンプリング間隔 Δt [s] のみで決定できることが分かる。これは、目標軌跡がどのような形状、位置であっても、フィードバックゲイン α を一意に決定でき、輪郭制御の制御性能は目標軌跡の形状や位置によらないことを示しているが、ここではその検証は行わない。

3.3 高精度輪郭制御のロボットアームへの適用

以上提案した輪郭制御法を、一般的な 3 自由度を持つ多関節形ロボットアームに適用するときの手順を示す。

3.3.1 位置入力生成部および輪郭制御理論部

前節にて示したように、輪郭制御理論部で生成された作業座標上での X, Y, Z 軸方向の位置入力 $u_x(t)$, $u_y(t)$, $u_z(t)$ は時間的に連続な値であるが、実際に計算機で位置制御を行うためには、位置制御のサンプリング間隔 Δt [s] ごとの離散的な値にしなければならないので、サンプリング間隔 Δt [s] ごとの、時刻 t [s] での位置入力 $u_x(t)$, $u_y(t)$, $u_z(t)$ を計算する。

3.3.2 逆キネマティクス

輪郭制御理論部で生成された作業座標上での X, Y, Z 軸方向の位置入力 $u_x(t)$, $u_y(t)$, $u_z(t)$ を、ロボットアームの逆キネマティクス (逆運動学) に従って、アームを駆動する各関節ごとの位置入力に変換する。Fig. 5 は一般的な 3 自由度を持つ多関節形ロボットアームのモデル図で、 l_1 , l_2 , l_3 はアーム長 [m], x , y , z はロボットアームの手先位置 [m], θ_1 , θ_2 , θ_3 は関節軸 1, 2, 3 の角度 [rad] である。 x , y , z は θ_1 , θ_2 , θ_3 で表すことができ、このモデル図からロボットアームのキネマティクス (運動学) を求め、さらにその逆変換である逆キネマティクスを求めると、各関節の位置入力 $u_{\theta 1}(t)$, $u_{\theta 2}(t)$, $u_{\theta 3}(t)$ は

$$\begin{aligned} u_{\theta 1}(t) &= \tan^{-1}(u_y(t)/u_x(t)) \\ u_{\theta 2}(t) &= \tan^{-1}\left(\frac{u_{xy}(t)}{u_z(t)}\right) \\ &\quad - \sin^{-1}\left(\frac{l_3 \sin \theta_3}{\sqrt{u_{xy}(t)^2 + u_z(t)^2}}\right) \\ u_{\theta 3}(t) &= \pm \cos^{-1}\left\{\frac{u_{xy}(t)^2 + u_z(t)^2 - l_2^2 - l_3^2}{2l_2l_3}\right\} \end{aligned} \quad (14)$$

となる。ただし、 $u_{xy}(t) = \sqrt{u_x(t)^2 + u_y(t)^2} - l_1$ である。 $u_{\theta 3}(t)$ は 2 値存在するので、ここでは作業対象 (切削等であれば被切削物) がアームの下に存在するときの $u_{\theta 3}(t) > 0$ を用いる。

ここで示した逆キネマティクスは、Fig. 5 のような 3 自由度の多関節形ロボットアームに関するものであるが、その他の種

Fig. 5 Diagram of multi link robot arm

類のロボットアームに適用する場合は、キネマティクスが異なるだけで、逆キネマティクスを用いて $u_{\theta 1}(t)$, $u_{\theta 2}(t)$, $u_{\theta 3}(t)$ を求める要領は同一である。また、XY テーブルのような直交座標系のロボットの場合は、キネマティクス、逆キネマティクスの部分を削除するとよく、この場合は先にあげた文献 [7] にて紹介している。

3.3.3 位置制御部

位置制御部はフィードバック制御方式で、 $u_{\theta 1}(t)$, $u_{\theta 2}(t)$, $u_{\theta 3}(t)$ と各関節角度 $\theta_1(t)$, $\theta_2(t)$, $\theta_3(t)$ との差に位置制御ゲイン $K_{p\theta 1}$, $K_{p\theta 2}$, $K_{p\theta 3}$ [1/s] を乗じてサーボコントローラに速度指令として与える。関節角度 $\theta_1(t)$, $\theta_2(t)$, $\theta_3(t)$ はモータに取り付けられているパルスジェネレータにて発生される回転角度に比例した出力パルス数をパルスカウンタで計測し、計算機内に取り込まれる。回転角度の取得周期は、位置入力を与えるサンプリング間隔と同じとする。

3.3.4 速度制御部、電流制御部、機構部

速度制御部、電流制御部は既存のロボットアームに使われているコントローラそのもので、機構部はロボットアーム本体である。提案の輪郭制御法では、この部分の変更はまったく必要ない。

3.3.5 キネマティクス

フィードバック信号として作業座標上でのロボットアームの手先位置 $x(t)$, $y(t)$, $z(t)$ を必要とするが、位置制御部で取得した関節角度 $\theta_1(t)$, $\theta_2(t)$, $\theta_3(t)$ に対してキネマティクス計算を施して $x(t)$, $y(t)$, $z(t)$ を求め、それを 3.1.5 項の座標変換へ送り、位置同期を図る。この誤差をフィードバックする周期は、位置制御のサンプリング間隔と同じものとする。

4. 位置同期による輪郭制御の性能評価

4.1 多関節形ロボットアーム実験装置と実験方法

提案した輪郭制御法を、Fig. 5 のモデルで表される多関節形アームロボット (Performer MK2) に適用した実験を行い、その結果から輪郭制御性能の評価をする。Performer MK2 はサーボ系の研究用として作られた 5 自由度の多関節形ロボットアームで、5 軸のうち先端の 2 軸を用いずに、軸 1, 2, 3 の三つの軸の輪郭制御実験を行った。各関節のサーボモータは速度、電流制御を行うサーボコントローラに接続され、サーボコントローラはプログラムにて位置制御を行う計算機に接続され

ている。各関節軸 1, 2, 3 には DC サーボモータ (定格速度 3,000 [rev/min]) があり、減速器を介してアームを駆動する。アームの長さは $l_1 = 0.10$ [m], $l_2 = 0.27$ [m], $l_3 = 0.33$ [m]、各関節軸の減速比は $n_1 = 120$, $n_2 = 160$, $n_3 = 144$ である。実験条件は、サンプリング間隔 $\Delta t = 0.004$ [s]、目標接線速度 $v = 0.2$ [m/s]、各関節のサーボ軸の位置制御部のゲイン $K_{p\theta 1} = K_{p\theta 2} = K_{p\theta 3} = 15$ [1/s] とする。この条件より式 (13) から $\alpha_0 = 7.8$ となったが、実際の装置に適用するためにある程度余裕をもって $\alpha_0 = 7$ と設定し、 $\alpha(t)$ の時定数 a は位置制御の比例ゲインの逆数の 1/2 とした。作業座標上での目標軌跡は図中に一点鎖線で示し、(0) 点 (0.3, 0.3, 0.0) から出発して、(1) 点 (0.4, 0.5, 0.2), (2) 点 (0.2, 0.5, 0.1) を通過し、(3) 点 ((0) 点) へ向かうものとする (座標の単位はすべて [m])。

実験を行う際に、各関節軸のサーボ系に故意に外乱を加えることで、サーボ系の特性変化による影響を観察した。外乱は一般的なトルク外乱と等価な影響を与えることが可能な速度外乱 [9] を用いた。ロボットアームの手先に、作業座標上の任意方向の速度外乱 d [m/s] が加わったとき、関節軸 1, 2, 3 に加わる角速度外乱 d_1 [rad/s], d_2 [rad/s], d_3 [rad/s] は、 d [m/s] を X, Y, Z 各軸方向に分解した d_x [m/s], d_y [m/s], d_z [m/s] と、ロボットアームの逆キネマティクスをもとに計算できる (付録参照)。この速度外乱を、目標軌跡 (0)–(1) の区間において、Z 軸方向の移動速度と同じ速度で逆方向のステップ入力を実験開始後 0.4~0.8 [s] の間加えた。

4.2 位置同期による輪郭制御性能の実験的検証

実験の結果を Fig. 6 に示し、上段は XY, YZ, ZX 各平面上での位置出力の軌跡、下段が追従軌跡と目標軌跡との誤差の時間推移である。輪郭制御性能評価は、X, Y, Z 軸方向の手先位置を実験中の各時刻で測定した関節角度をキネマティクス変換を施した値から求め、目標軌跡との差をもって行った。

比較のために、位置同期を図らない場合 (従来法) の実験を行い、提案法に比べ外乱等の影響が顕著に出ている Y-Z 座標での軌跡、および軌跡誤差を Fig. 7 (a-4), (b-4) にそれぞれ示す。Fig. 7 (a-4) および (b-4) の従来法では、定常的な誤差は最大値で 0.28 [mm] 程度であるのに対し、Fig. 6 (a-2) および (b-2) の提案法では最大で 0.019 [mm] 程度と、誤差は約 1/14 に減少した。目標直線軌跡 (0)–(1) 間に外乱が加えられているために、従来法では Z 軸方向の位置出力が目標軌跡に対して大きくずれている (最大 6.4 [mm]) が、提案法ではその誤差が十分に減少している (最大 0.36 [mm]) のが分かる。以上のことから、本提案法が輪郭制御における定常的な軌跡誤差や外乱による誤差に対して有効であることが実験的に確認できた。これから、従来法では取り除くことができなかったキネマティクスによる非線形変換や、外乱による追従軌跡と目標軌跡との誤差が、位置同期を図ることによって低減でき、輪郭制御性能が大きく向上していることが分かる。

4.3 計算機シミュレーション結果との比較評価

前節では実験的検証によって、輪郭制御法の有効性を示したが、ここでは計算機シミュレーションを行い、理屈の上から提案法の有効性を検証する。計算機シミュレーションでは、各関節のサーボ軸には線形一次系を用い [7]、各関節のサーボ軸の

Fig. 6 Experimental result of contour control with synchronous position control

Fig. 7 Experimental result of contour control without synchronous position control (a-4) (b-4), simulation result of contour control with synchronous position control (a-5) (b-5) and without synchronous control (a-6) (b-6)

比例ゲイン K_1, K_2, K_3 は、実際のロボットアーム実験装置の各軸の位置ループゲイン $K_{p\theta 1}, K_{p\theta 2}, K_{p\theta 3}$ と近似的に等価とみなし、 $K_1 = K_2 = K_3 = 15 [1/s]$ とした。アームの長さ、目標接線速度、サンプリング間隔等の条件は前節で示したロボットアームのパラメータを用いた。シミュレーション結果は、提案法を Fig. 7(a-5) および (b-5)、従来法を Fig. 7(a-5) および (b-5) に示し、各グラフの見方は実験と同じである。提案法、従来法ともにシミュレーション結果は実験結果とほとんど完全に一致しする結果を得た。したがって、本提案法の有効性を理屈の上からも確認できた。

5. ま と め

現在の産業用多関節形ロボットアームの輪郭制御の精度を向上する手段として、各軸の相互位置同期制御を図った輪郭制御法を提案した。提案法によって、ロボットアームの輪郭制御実験を行い、その有用性を示した。産業界のロボットアームによる輪郭制御の現状は、ロボットの各関節ごとのサーボ系を各軸独立に制御しているのがほとんどであるが、本方法では単に各軸の位置出力誤差信号をフィードバックするという簡単な手法で、諸々の動作条件下で非常に高精度な輪郭制御が達成できるので、今後の産業界におけるロボットアームの制御方式に改革をもたらすものと期待している。

参 考 文 献

- [1] 吉川：ロボット制御基礎論。pp.1-10, コロナ社, 1988.
- [2] 林, 黒江, 丸端：“外乱オブザーバによる非線形システムの非干渉化制御法—DD ロボットの作業空間における非干渉化—”, 日本ロボット学会誌, vol.13, no.3, pp.420-428, 1995.
- [3] 松下, 土谷：“オンライン目標値計画—ロボットマニピュレータの適用—”, 日本ロボット学会誌, vol.13, no.8, pp.1160-1168, 1995.
- [4] B. Fardanesh and J. Rastinger: “A New Model-Based Tracking Controller for Robot Manipulators using Trajectory Pattern Inverse Dynamics,” IEEE Trans. Robotics and Automations, vol.8, no.2, pp.279-285, 1992.
- [5] N. Sadegh and K. Guglielmo: “Design and Implementation of Adaptive and Repective Controllers for Mechanical Manipulators,” IEEE Trans. Robotics and Automations, vol.8, no.3, pp.395-400, 1992.
- [6] W.H. Zhu, H.T. Chen and Z.J. Zhang: “A Variable Structure Robot Control Algorithm with an observer,” IEEE Trans. Robotics and Automations, vol.8, no.2, pp.279-285, 1992.
- [7] 中村, 前田, 久良：誤差フィードバックによるロボットアームの輪郭制御法, 第13回日本ロボット学会学術講演会予稿集, 1995.
- [8] 堀, 久良, 川端：アドバンスト・モーションコントロール (運動の高性能制御) の理論と応用, 電気学会論文誌 D, vol.110, no.11, pp.1110-1118, 1990.
- [9] 中村, 前田, 後藤, 久良：“メカトロサーボ系におけるトルク外乱と等価な速度信号の決定と利用法”, 日本ロボット学会誌, vol.13, no.6, pp.808-812, 1995.

付録 誤差フィードバックゲイン α の決定法

誤差フィードバックゲイン α の決定手順は、まずロボットアームのキネマティクスを線形近似して、作業座標上で展開した制御理論を、関節座標上に近似的に変換する。次に、サーボ系のモデルを用いて解析を行い、輪郭制御を施した場合の特性方程式を求め、最後にデジタル制御の発散条件をもとに α の最大値を求める。ただし、本文で示した手法は三次元 3 自由度

ロボットアームであったが、関節 2 軸のロボットアームの制御特性を示す特性方程式を求め、それをもとに三次元の場合の結果を求めた。

(1) キネマティクスの線形近似

関節 2 軸のロボットアームは、

Fig. 5 において $\theta_1 = 0 [\text{rad}]$ とした場合の XY 平面上で動作を考えればよく、そのキネマティクス

$$\begin{aligned} x &= \{l_1 + l_2 \sin \theta_2 + l_3 \sin(\theta_2 + \theta_3)\} \cos \theta_1 \\ y &= \{l_1 + l_2 \sin \theta_2 + l_3 \sin(\theta_2 + \theta_3)\} \sin \theta_1 \quad (15) \\ z &= l_2 \sin \theta_2 + l_3 \sin(\theta_2 + \theta_3) \end{aligned}$$

に、式 (15) に $\theta_1 = 0 [\text{rad}]$ を代入するとよい。位置制御系のサンプリング間隔 $\Delta t [s]$ ごとにテイラー展開により θ_2^0, θ_3^0 まわりで一次近似すると、直交座標上の位置 z, x は

$$\begin{aligned} z &\simeq l_2 \cos \theta_2^0 + l_3 \cos(\theta_2^0 + \theta_3^0) \\ &\quad + l_2 \sin \theta_2^0 \cdot \theta_2^0 + l_3 \sin(\theta_2^0 + \theta_3^0) \cdot (\theta_2^0 + \theta_3^0) \\ &\quad - \{l_2 \sin \theta_2^0 + l_3 \sin(\theta_2^0 + \theta_3^0)\} \theta_2 - l_3 \sin(\theta_2^0 + \theta_3^0) \cdot \theta_3 \\ &= a_{z0} + a_{z1} \theta_2 + a_{z2} \theta_3 \quad (16) \end{aligned}$$

$$\begin{aligned} x &\simeq l_2 \sin \theta_2^0 + l_3 \sin(\theta_2^0 + \theta_3^0) \\ &\quad - l_2 \cos \theta_2^0 \cdot \theta_2^0 - l_3 \cos(\theta_2^0 + \theta_3^0) \cdot (\theta_2^0 + \theta_3^0) \\ &\quad \{l_2 \cos \theta_2^0 + l_3 \cos(\theta_2^0 + \theta_3^0)\} \theta_2 + l_3 \cos(\theta_2^0 + \theta_3^0) \cdot \theta_3 \\ &= a_{x0} + a_{x1} \theta_2 + a_{x2} \theta_3 \quad (17) \end{aligned}$$

となる。式 (16) (17) を行列で表すと

$$\begin{pmatrix} z \\ x \end{pmatrix} = \begin{pmatrix} a_{z0} \\ a_{x0} \end{pmatrix} + \begin{pmatrix} a_{z1} & a_{z2} \\ a_{x1} & a_{x2} \end{pmatrix} \begin{pmatrix} \theta_2 \\ \theta_3 \end{pmatrix} \quad (18)$$

となる。これから、 θ_2, θ_3 は

$$\begin{aligned} \begin{pmatrix} \theta_2 \\ \theta_3 \end{pmatrix} &= \begin{pmatrix} a_{z1} & a_{z2} \\ a_{x1} & a_{x2} \end{pmatrix}^{-1} \left\{ \begin{pmatrix} z \\ x \end{pmatrix} - \begin{pmatrix} a_{z0} \\ a_{x0} \end{pmatrix} \right\} \\ &= \frac{1}{2l_2 l_3 s_2} \begin{pmatrix} a_{x2} & -a_{z2} \\ -a_{x1} & a_{z1} \end{pmatrix} \left\{ \begin{pmatrix} z \\ x \end{pmatrix} - \begin{pmatrix} a_{z0} \\ a_{x0} \end{pmatrix} \right\} \\ &= \begin{pmatrix} b_{10} \\ b_{20} \end{pmatrix} + \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix} \begin{pmatrix} z \\ x \end{pmatrix} \quad (19) \end{aligned}$$

となる。

(2) 輪郭制御法の近似表記

相互位置同期制御法による Z, X 軸方向の位置入力 $u_z(t), u_x(t)$ は、式 (11) (9) で $e_{xy} = e_{yz} = 0$ とし e_{zx} のみを代入すると

$$u_z(t) = c_z u(t) - 2\alpha e(t) c_x \quad (20)$$

$$u_x(t) = c_x u(t) + 2\alpha e(t) c_z \quad (21)$$

となる。ただし

$$\begin{aligned} e(t) &= c_z x(t) - c_x z(t) \\ &= (c_x a_{z0} - c_z a_{x0}) \\ &\quad + (c_x a_{z1} - c_z a_{x1}) \theta_2(t) + (c_x a_{z2} - c_z a_{x2}) \theta_3(t) \\ &= e_0 + e_1 \theta_2(t) + e_2 \theta_3(t) \quad (22) \end{aligned}$$

である。これを $z = u_z(t)$, $x = u_x(t)$ とおいて式 (19) に代入すると、各関節への入力 $u_{\theta 2}(t)$, $u_{\theta 3}(t)$ が次式のように求められる。

$$\begin{aligned} u_{\theta 2}(t) &= b_{10} + b_{11}u_z(t) + b_{12}u_x(t) \\ &= b_{10} - 2\alpha(c_x b_{11} + c_z b_{12}) \\ &\quad + (c_x b_{11} + c_z b_{12})\{u(t) - 2\alpha(e_1\theta_2 + e_2\theta_3)\} \\ &= d_{10} + d_{11}\theta_2 + d_{12}\theta_3 + d_{13}u(t) \end{aligned} \quad (23)$$

$$\begin{aligned} u_{\theta 3}(t) &= b_{20} + b_{21}u_z(t) + (t)b_{22}u_x(t) \\ &= b_{20} - 2\alpha(c_x b_{21} + c_z b_{22}) \\ &\quad + (c_x b_{21} + c_z b_{22})\{u(t) - 2\alpha(e_1\theta_2 + e_2\theta_3)\} \\ &= d_{20} + d_{21}\theta_2 + d_{22}\theta_3 + d_{23}u(t) \end{aligned} \quad (24)$$

(3) 特性方程式の解法

式 (23) (24) の位置入力を、線形一次系で表された各関節の角度を制御するサーボ系に与えたときの位置出力 θ_2 , θ_3 は、ラプラス変換すると

$$\Theta_2(s) = \frac{K}{s+K} U_{\theta 2}(s) \quad (25)$$

$$\Theta_3(s) = \frac{K}{s+K} U_{\theta 3}(s) \quad (26)$$

となる。ただし、 $K[1/s]$ は線形一次系のサーボ系の比例ゲインで、Z, X 軸共通とする。式 (25) (26) に式 (23) (24) をラプラス変換して代入し、 $\Theta_2(s)$, $\Theta_3(s)$ について解くと

$$\begin{aligned} \Theta_2(s) &= K[Kd_{11}d_{23} + \{s + (1 - d_{22})K\}d_{13}]U(s)/\Delta \\ &\quad + K[Kd_{12}d_{20} + \{s + (1 - d_{22})K\}d_{10}]/\Delta \end{aligned} \quad (27)$$

$$\begin{aligned} \Theta_3(s) &= K[Kd_{13}d_{21} + \{s + (1 - d_{11})K\}d_{23}]U(s)/\Delta \\ &\quad + K[Kd_{10}d_{21} + \{s + (1 - d_{11})K\}d_{20}]/\Delta \end{aligned} \quad (28)$$

が得られる。ただし

$$\begin{aligned} \Delta &= \{s + (1 - d_{11})K\}\{s + (1 - d_{22})K\} - K^2d_{12}d_{21} \\ &= s^2 + \{2 - (d_{11} + d_{12})\}Ks \end{aligned}$$

$$+ \{1 - (d_{11} + d_{12}) + (d_{11}d_{22} - d_{12}d_{21})\}K^2 \quad (29)$$

であり、 Δ は Θ_2 , Θ_3 の振る舞いを表す特性方程式である。ここで、式 (23) (24) に $b_{10} \sim b_{22}$, $e_{10} \sim e_{23}$ を代入して $d_{11} + d_{22}$, $d_{11}d_{22}$, $d_{12}d_{21}$ を求めると

$$d_{11} + d_{22} = -2\alpha \quad (30)$$

$$d_{11}d_{22} = d_{12}d_{21} \quad (31)$$

となることが分かり、式 (29) は

$$\begin{aligned} \Delta &= (s + K)\{s + K(1 + 2\alpha)\} \\ &= (s - p_1)(s - p_2) \end{aligned} \quad (32)$$

と簡単になる。 p_1 , p_2 は制御特性を示す極で、式 (32) から

$$p_1 = -K \quad (33)$$

$$p_2 = -K(1 + 2\alpha) \quad (34)$$

となる。

(4) α の決定法

極 p_1 , p_2 のうち、絶対値が大なる p_2 の逆数が制御系の時定 $T[s]$ と等しいとし、デジタル制御が発散しない条件式 [7]

$$\Delta t < 2T \quad (35)$$

の $T[s]$ に、式 (34) の p_2 の絶対値の逆数を代入すると、 α は

$$\alpha < 1/K\Delta t - 0.5 \quad (36)$$

となり、この範囲で α を設定するとよいことが分かる。Y 軸と Z 軸の場合で同様の結果が得られる。

Fig. 5 の三次元のロボットの場合、フィードバック信号が 2 個ある。X 軸と Y 軸の特性はほぼ等しいと考えることにより、三次元の場合のフィードバックする誤差の値は、二次元の場合の約 2 倍になっているとみなす。よって、三次元での α の値は、二次元の場合の約 1/2 でなければ制御系が発散するため、式 (36) を 1/2 倍した式 (13) が導かれる。

前田貴信 (Takanobu Maeda)

1969 年 8 月 6 日生。1990 年佐世保工業高等専門学校電気工学科卒業。1992 年佐賀大学理工学部電気工学科編入学。1994 年同大学院工学系研究科電気工学専攻博士前期課程入学。現在に至る。主としてメカニカルシステム制御の研究に従事。

(日本ロボット学会学生会員)

久良修郭 (Nobuhiro Kyura)

1942 年 2 月 12 日生。1966 年九州大学大学院工学研究科電気工学専攻修士課程修了。同年 (株) 安川電機入社。研究所、技術開発センター、基礎研究所部長を経て、現在近畿大学九州工学部電気工学科教授。工学博士。計測自動制御学会などの会員。

(日本ロボット学会正会員)

中村政俊 (Masatoshi Nakamura)

1943 年 8 月 20 日生。1973 年九州大学大学院工学研究科電気工学専攻博士課程修了。同年同大学助手。佐賀大学講師。同助教授。1988 年同教授になり現在に至る。工学博士。システム制御の理論と応用の研究に従事。電気学会、計測自動制御学会、システム制御情報学会、日本脳波・筋電図学会、日本 ME 学会、IEEE 学会などの会員。

(日本ロボット学会正会員)