垂直2関節ロボットの負荷推定と補償制御*

小林 義武^{*1}, 山藤 和男^{*2}

Estimation and Compensation Control of a Load of a Vertical Two-Link Robot

Yoshitake KOBAYASHI and Kazuo YAMAFUJI

It is widely recognized that the industrial robots used in the production lines or in other engineering fields are installed comparatively higher rated actuators and have higher rigidity than those required, whereas they have too small payload capacity. To achieve high speed drive and accurate positioning under a high payload is indispensable for an advanced industrial robot. In order to increase payload/deadweight ratio without losing high speed driving and accurate control of robots, the nonlinear terms in the equations of motion relating to their load and attitude must be well compensated. The authors have developed and examined two kinds of the load estimation and compensation control method for a vertical-type manipulating robot, which are based on the gravity estimation-compensation and the Fuzzy-set theory. It is confirmed experimentally that although both compensating methods are useful, the Fuzzy theory is much better than the gravity compensation method.

Key Words: Industrial Robot, High Load/Deadweight Ratio, Load Estimation, Compensation Control, Fuzzy-set Theory

1. 緒 言

今日、産業用ロボットは生産自動化のための有力な手段として部品加工や組立工程でもっとも多く用いられている。これらの産業用ロボットを制御という面から見るとまだ多くの問題をかかえている。そのいくつかを挙げれば、高い剛性と減速比をもち、必要な定格よりも大きなパワーのモータで駆動される一方、残留振動の問題のために最高速度に余力を残しており、負荷自重比が非常に小さいという欠点がある。すでに産業用ロボットについて、負荷自重比を大きくするとともに、高速駆動と高精度制御、制御の知能化を実現するための研究開発が行なわれている⁽¹⁾⁻⁽³⁾。

本研究はロボット先端に把握した物体重量と姿勢の変化にともなうシステムパラメータ変動を推定しながら動作することにより、負荷/自重比が大きい場合でも制御性の悪化しない高速・高精度ロボット制御システムを開発することを目的とする。そのため、運動方程式中の非線形要素が無視できないほど大きな垂直2

関節型ロボットを製作し、非線形要素を無視することなく、これらのパラメータ変動の影響を吸収することのできる制御系を開発する。これらの目的を達成するために、具体的には次の方法により研究を行なう。

- (1) マニピュレータ先端に重量物負荷を把握させた場合でも、ロボットの高速駆動と無衝撃停止を実現するための制御法と各軸同時駆動により連続経路制御を行うことのできる軌道発生法を検討する。
- (2) 実機について、いくつかの代表的な駆動条件に対して運動方程式における各非線形要素の影響を比較する。
- (3) 姿勢と負荷の変化にもとづく非線形要素の変化を補償するために提案した2つの制御法について計算機シミュレーションと実機の制御実験を行ない、結果を比較する。

2. 垂直2関節ロボットと運動方程式

製作した垂直2関節ロボットの概要を図1に示す。

[記号] $i = 1, 2$ として、

J_i : 第 i 軸アームの慣性モーメント [kg m^2]

M_i : 第 i 軸の質量 [kg]

M_3 : 手先に取り付けた負荷の質量 [kg]

θ_i : 第 i 軸の角度 [rad]

* 昭和63年10月29日 山梨地方講演会において講演、原稿受付 平成元年4月3日。

^{*1} セイコーエプソン(株) (〒399-02 長野県諏訪郡富士見町)。

^{*2} 正員、電気通信大学 (〒182 調布市調布ヶ丘1-5-1)。

l_i : 第 i 軸アームの長さ [m]
 e_i : 第 i 軸の重心位置 [m]
 e_3 : 手先に取り付けた負荷の重心位置 [m]
 τ_i : 第 i 軸の入力トルク [Nm]
 g : 重力加速度 [m/s²]
 D_i : 第 i 軸回転部の粘性摩擦トルク係数 [Nms]
 F_i : 第 i 軸回転部の動摩擦トルク [Nm]
 運動方程式はマニピュレータの第 1, 2 軸に関して次のようになる。

$$\tau_1 = J_{11} \ddot{\theta}_1 + J_{12} \ddot{\theta}_2 + f_{12} \dot{\theta}_2^2 - g_1 + D_1 \dot{\theta}_1 + F_1 \text{sign}(\dot{\theta}_1) \quad (1)$$

$$\tau_2 = J_{12} \ddot{\theta}_1 + J_{22} \ddot{\theta}_2 + f_{12} \dot{\theta}_1^2 - g_2 + D_2 \dot{\theta}_2 + F_2 \text{sign}(\dot{\theta}_2) \quad (2)$$

ここに、 $J_{11} = J_1 + (M_2 + M_3) l_1^2$
 $J_{12} = 0.5 (M_2 + M_3) l_1 (l_2 + 2e_3) \times \cos(\theta_2 - \theta_1)$
 $J_{22} = J_2 + M_3 (l_2 + e_3)^2 + 0.5 M_3 \times (0.05 + e_3)^2$

$$f_{12} = 0.5 (M_2 + M_3) l_1 (l_2 + 2e_3) \times \cos(\theta_2 - \theta_1)$$

$$g_1 = [M_1 e_1 + (M_2 + M_3) l_1] g \sin \theta_1$$

$$g_2 = [M_2 e_2 + M_3 (l_2 + e_3)] g \sin \theta_2$$

製作した垂直2関節ロボットについて測定したシステムパラメータの値を表1に示す。

3. 制御システムとソフトウェア

制御系の構成を図2に示す。制御用コンピュータとして、16ビット・パーソナルコンピュータ(CPU 80286, 8MHz)を用い、OSにはMS-DOS, 開発用言語にはMS-Cおよびアセンブラ言語を使用した。制御周期は400Hz以上(サンプリング時間2.5ms以下)が要求されるため、制御部分はアセンブラを用い、その他の部分はC言語で開発を行った。これはC言語を用いれば比較的効率のよいコードの生成を行うことができ、かつ構造化プログラミング可能で保守性にすぐれているためである。

制御演算および軌道発生演算には倍精度浮動小数点演算を高速で行う必要があり、これをソフトウェアのみで実現することは困難であるため、高速数値演算用コプロセッサ(i80287)の搭載により実現した。

本制御プログラムでは最大制御周期は556Hz(サンプリング時間1.8ms)となった。ロボットの

図1 垂直2関節ロボットの概要

表1 システムパラメータの測定値

Symbol	1st Arm	2nd Arm
J_i [kgm ²]	1.67×10^{-1}	5.29×10^{-2}
M_i [kg]	3.50	2.15
l_i [m]	0.30	0.30
e_i [m]	0.125	0.142
D_i [Nms]	1.3×10^{-4}	5.2×10^{-3}
F_i [Nm]	2.64	4.9×10^{-2}

図2 制御システムの概要

制御は正確な時間間隔で行われなければならないので、パソコンに内蔵されたカウンタ用ICを利用して正確な割り込み信号を発生させ、各周期ごとに制御プログラムを割り込み信号により起動している。

4. 軌道発生法

一般に、作業空間内でアーム先端が動く曲線が指定されると、その曲線上で先端が滑らかな加減速を行なうように時間の関数としてロボットの先端位置、姿勢、速度および加速度ベクトルを指定する必要がある。これによりロボット先端が衝撃的に動くことはなくなり、曲線上で滑らかな加減速を行なうことができる。また、任意の軌道を指定することにより、連続経路制御(CP制御)を行なうことができる。

本研究では、図3に示す軌道発生用フローチャートに従って、指定された軌道上のロボット手先の位置、速度および加速度から各関節アクチュエータの変位、速度および加速度を逆運動学により計算した。これはロボット手先の状態量から各関節における状態量への逆変換を意味し、手先の状態量は加減速曲線を利用して、軌道上各点で時系列的に記述される。

加減速曲線として変形正弦曲線^{(4)・(6)}を用いた。これは高速インデックスマシンや専用機駆動用カム曲線として開発されたもので、各特性曲線の連続性を保ちつつ加速度の最大値を低く、かつ速度の最大値も低く抑えるという条件を満足する加減速曲線である。これをソフトウェア化してパソコンのメモリに格納した。ロボットの駆動時にはこれをメモリから呼び出してサーボモータ・ドライバに出力し、各関節アクチュエータを駆動することになる。

図4には変形正弦曲線で各軸を駆動する場合の軌道発生の一例として、3次Bezier曲線为目标軌道とした場合の結果を示す。図では等時間間隔の供試ロボットアームの姿勢変化として軌道が示されている。

5. マニピュレータの非線形要素の評価

供試ロボットマニピュレータを駆動する場合に関係する非線形要素の大きさの評価を行うため、計算機で制御シミュレーションを行なった。この場合、マニピュレータを任意軌道上で加減速させるときに必要なトルクに占める非線形要素の割合を各非線形要素成分の自乗の積分値により評価する。前章で述べた軌道発生法と変形正弦曲線を用い、マニピュレータが2点間直線にそって動くときの計算機シミュレーションを行なう。数値計算にはルンゲ・クッタ法を用いた。摩擦に関する項は省略した。

図5(a)、(b)はそれぞれ、ロボット先端を最

図3 軌道発生用フローチャート

図4 3次Bezier曲線上を変形正弦速度曲線で駆動する場合の等時間間隔の軌道発生

(a) 第1軸の入力トルクの時間的变化

(b) 第2軸の入力トルクの時間的变化

図5 ロボット先端を最大速度 1500 mm/sで駆動する場合の第1, 2軸の入力トルク各成分の変化

大速度 1500 mm/s で動かした時の第1, 第2軸アクチュエータの入力トルクの各成分の時間的变化を示している。最高速度が 200 mm/s と 1500 mm/s について、入力トルクの各非線形成分の積分値によって非線形要素の大きさを評価した場合の入力トルクに占める各非線形要素の割合を表2に示す。

最大速度 200 mm/s では入力トルクの 98.2% は重力項で占められ、1500 mm/s では入力トルクの 38.3% は重力項、34.4% は各軸自身の慣性項となっている。これより低速域では非線形項と

して重力項のみを扱い、最大速度が1500 mm/s程度の領域では、非線形項としては重力項と慣性項の双方を扱えばよいことが明らかである。

6. 制御系の設計

非線形運動方程式(1)、(2)は一般の多関節マニピュレータのそれと同じく、次式で表わされる⁽⁶⁾。

$$\tau = M(\theta) \ddot{\theta} + f(\dot{\theta}, \theta) + V\dot{\theta} + g(\theta) \quad (3)$$

ただし、 τ : 各関節アクチュエータによる入力トルクからなるn次元ベクトル

θ : 各関節の変位を表わすn次元ベクトル

$M(\theta)$: $n \times n$ 慣性行列

$f(\dot{\theta}, \theta)$: 遠心力、コリオリ力のn次元ベクトル

V : 各関節の粘性摩擦トルク係数を対角要素とするn次元行列

$g(\theta)$: 重力項を表わすn次元ベクトル

ここで、新たに入力トルクとして τ' を用いて、

$$\tau = \alpha \tau' + \beta \quad (4)$$

ただし、 $\alpha = M(\theta)$ 、 $\beta = f(\dot{\theta}, \theta) + V\dot{\theta} + g(\theta)$ とする。

式(4)を(3)に入れて整理すると、

$$\tau' = \ddot{\theta} \quad (5)$$

となり、非線形方程式は単位質量あたりの運動方程式に線形化された。 K_p 、 K_v を角度および角速度偏差に対するフィードバック係数として、

$$\tau' = \ddot{\theta} r + K_v(\dot{\theta} r - \dot{\theta}) + K_p(\theta r - \theta) \quad (6)$$

ただし、 θr : 各関節の指令角のn次元ベクトル

$\dot{\theta} r$: 指令角速度を表わすn次元ベクトル

$\ddot{\theta} r$: 指令角加速度を表わすn次元ベクトルとし、式(5)を(6)に入れ、 $e = \theta r - \theta$ とおくと、

$$\ddot{e} + K_v \dot{e} + K_p e = 0 \quad (7)$$

式(5)で τ' を定義するとこのシステムは閉ループシステムとなり、 K_p 、 K_v を適当に選ぶことにより、容易に好ましい軌道追従制御を行なうことができる。

式(7)の特性方程式は、次式となる。

$$s^2 + K_v s + K_p = 0$$

この系が臨界減衰となるように、 K_v 、 K_p を選ぶと、

$$K_v = 2\sqrt{K_p}$$

以上の結果を垂直2関節ロボットに適用すると、

$$\tau_1' = \ddot{\theta}_1 r_1 + K_v(\dot{\theta}_1 r_1 - \dot{\theta}_1) + K_p(\theta_1 r_1 - \theta_1)$$

$$\tau_2' = \ddot{\theta}_2 r_2 + K_v(\dot{\theta}_2 r_2 - \dot{\theta}_2) + K_p(\theta_2 r_2 - \theta_2)$$

これより各アクチュエータへの入力は、

表2 入力トルク各成分の2乗積分値に占める非線形要素の割合
(a) ロボット先端の最大速度: 200 mm/s

i	j	unit	Sii	Tij	Uj	Vi	Total
1	2	N ² m ² s %	0.24 0.03	0.01 0.00	0.01 0.00	728.8 97.6	746.6 100.0
2	1	N ² m ² s %	0.16 0.02	0.01 0.00	0.01 0.00	663.3 98.9	670.9 100.0

(b) ロボット先端の最大速度: 1500 mm/s

i	j	unit	Sii	Tij	Uj	Vi	Total
1	2	N ² m ² s %	102.5 33.0	5.0 1.6	4.5 1.5	97.8 31.5	310.7 100.0
2	1	N ² m ² s %	71.0 35.7	4.4 2.2	4.0 2.0	89.5 45.0	199.0 100.0

$$S_{ii} = \int_0^{t_e} (J_{ii} \tau_i)^2 dt, \quad T_{ij} = \int_0^{t_e} (J_{ij} \tau_j)^2 dt$$

$$U_j = \int_0^{t_e} (f_{12} \dot{\theta}_j)^2 dt, \quad V_i = \int_0^{t_e} (g_i)^2 dt \quad (\text{Gravitational term})$$

$$t_e = \text{Controlling time.}$$

$$\tau_1 = J_{11} \tau_1' + J_{12} \tau_2' + f_{12} \cdot \dot{\theta}_2^2 - g_1 + D_1 \cdot \dot{\theta}_1 + F_1 \cdot \text{sign}(\dot{\theta}_1) \quad (8)$$

$$\tau_2 = J_{12} \tau_1' + J_{22} \tau_2' + f_{12} \cdot \dot{\theta}_1^2 - g_2 + D_2 \cdot \dot{\theta}_2 + F_2 \cdot \text{sign}(\dot{\theta}_2) \quad (9)$$

ただし、 τ_1' 、 τ_2' : 第1、2軸がそれぞれ線形であると仮定した場合の入力トルク[Nm]、 τ_1 、 τ_2 : 第1、2軸に対する実際の入力トルク[Nm]。

7. 重力トルクによる負荷の推定

第5章の結果より、マニピュレータの先端に取り付けた負荷の影響を無視することはできないことがわかった。また、フィードバックゲインを大きくするだけでは負荷の影響を除去することはできず、何らかの方法で負荷質量を推定して、その補償を行うことが必要である。ロボットの姿勢がわかれば、その姿勢における重力トルクは簡単な式で計算できる。計算により得られた重力トルクをマニピュレータに与えたときの腕の動きから負荷質量を推定する。

この方法により、負荷の推定を行なった結果を図6に示す。図から負荷は計算値に比べて推定値の方が一般に大きくなっていることがわかる。この結果を用いれば、ロボット先端負荷の値を実用に耐える程度の精度で推定することは容易である。しかし、重力による負荷の推定法はロボットが負荷を把握するたびに推定動作を行なわなければならないというわずらわしさがあり、さらに推定に時間がかかるので、マニピュレータの動作中に負荷の推定を行う次章の方法が望ましい。

8. ファジィ理論を用いたトルクエラー推定法

従来から行われている非線形要素のフィードバックによる線形化制御では完全な線形化を実現することはできず、制御時に各種の偏差を生じる。

本研究では偏差の原因となるトルクの過不足分をトルクエラーとし、ロボットが物体を把握することによる重量の変化と動摩擦トルクの変化から生じるトルクエラーを、ファジィ理論を応用してマニピュレータの動作中に推定する方法を考える。位置偏差 e と速度偏差 $\dot{e}$ からファジィ理論を用いてトルクエラー τ_e を推定することにしたが、これはロボットのセンサーの数を増やさないためにも有利である。

(1) 入力トルク: τ_{e1} , τ_{e2} をそれぞれ第1, 2軸のファジィ同定によるトルクエラーとして、各軸アクチュエータの入力トルクを次式のように与える。

$$\tau_1 = J_1 \tau_1' - (M_1 e_1 + M_2 l_1) g \sin \theta_1 + \tau_{e1} + D_1 \dot{\theta}_1 + F_1 \text{sign}(\dot{\theta}_1) \quad (10)$$

$$\tau_2 = J_2 \tau_2' - M_2 e_2 g \cdot \sin \theta_2 + \tau_{e2} + D_2 \dot{\theta}_2 + F_2 \text{sign}(\dot{\theta}_2) \quad (11)$$

ここに、 τ_1' , τ_2' は第6章のそれらと同じ。

(2) ファジィ同定のためのルール:

a. ファジィルール; "If e is A and $\dot{e}$ is B, then $\Delta \tau_e$ is C." (12)

e , $\dot{e}$ および $\Delta \tau_e$ の大きさを判定するためのファジィルール⁽⁷⁾として、A, B, Cのそれぞれについて、P(正), ZO(ゼロ), N(負), B(大きい), M(中程度)の5通りを設定した。さらに、A, B, Cの間に表3に示す関係をつけると、全体でファジィルールは25種類となる。

b. メンバシップ関数; 上の5通りの基本ルールの組合せからつくられるPB, PM, ZO, NM, NBに対応するメンバシップ(MS)関数を図7に示す。

c. 推論法; e と $\dot{e}$ に対して、25種のルールすべてを用いて、

$$W_i = A_i(e) \cdot B_i(\dot{e}) \quad i=1 \sim 25 \quad (13)$$

を求め、更新メンバシップ関数 $W_i \cdot C_i$ を計算し、合成メンバシップ関数 $C_{\#}$ を得る。

$$C_{\#} = U(W_i \cdot C_i), U: \text{or 演算} \quad (14)$$

さらに、この関数 $C_{\#}$ の重心を求めて、 $\Delta \tau_e$ とする。

$$\Delta \tau_e = \{ C_{\#}(t) t dt / \int C_{\#}(t) dt \quad (15)$$

以上の方法を適用して、計算機によりロボットアームについて2点間の動作シミュレーションを行った結果の一例を図8に示す。ファジィ・シミュレーションによれば低速でも高速でもトルクエラーの推定が良好で、負荷の影響をすべて補償して行ったシミュレーション結果と同程度の角度偏差が得られることがわかつ

図6 重力トルクを利用した負荷の推定

表3 ファジィルール

A	PB or NB					PM or NM					ZO				
B	PB	PM	ZO	NM	NB	PB	PM	ZO	NM	NB	PB	PM	ZO	NM	NB
C	PB	PB	PM	PM	NM	PB	PB	PM	ZO	NM	PB	PM	ZO	NM	NB

P: Positive, ZO: Zero, N: Negative, B: Big, M: Medium

図7 メンバシップ関数の図示

図8 ファジィ理論によるトルクエラーの推定(シミュレーション)

図9 ファジィ理論によるトルクエラーの推定(実験)

た、図9には、同じ条件で行った実験結果を示す。実験によれば、低速では必要なトルクと推定値から得られる入力トルクはほぼ一致しているが、高速では推定がやや不十分であった。その理由は関節アクチュエータの固体摩擦のため、ファジィ推定法で許容される以上のトルクエラーが生じたためと考えられる。

9. トルクエラーと重力による 負荷推定の比較

繰り返し位置精度の測定結果を図10に示す。負荷を補償した場合（負荷の質量も含めて、運動方程式中

の非線形要素を計算し、非線形フィードバックを行い線形化する方法）とファジィ理論を適用した場合には負荷補償がない場合に比べて定常偏差は小さい。

図11は偏差の絶対値を制御開始時から終了時まで積分した値（偏差積分）により動作中の位置偏差を評価したものである。動作中つねに0.1 mmの偏差があるとすれば、偏差積分値は0.875 mm sとなる。これは絶対値の積分となっているので正の偏差と負の偏差で打ち消し合うことはない。低速では負荷を補償した場合の偏差積分値と負荷を補償しない場合の偏差積分とのほぼ中間の値を得た。ファジィ理論を用いたトルクエラー推定法では初期条件は $\tau_e = 0$ であり、初期状態では負荷を補償できず、動作中にトルクエラーを推定し負荷の補償を行うことを考慮すれば、比較的良好な軌道追従性が得られた。

10. 結 論

マニピュレータ先端に負荷を把握させた場合でも、高速駆動、高精度位置決めを実現するための制御法と連続経路制御を行うことのできる軌道発生法を検討するとともに、先端負荷と姿勢の変化によるシステムパラメータ変動を重力とファジィ理論を用いて推定する二方法を提案した。主な結論は次の3点である。

- (1) 変形正弦曲線にもとづく速度曲線とここで提案した軌道発生法とを用いて、サンプリング時間ごとに軌道発生を行って、マニピュレータを高速駆動し、精密に位置決めすることができた。
- (2) 重力を利用する負荷推定法ではロボットの姿勢がわかれば重力トルクは簡単に計算でき、これを軸アクチュエータに与えたときのマニピュレータの動きから負荷質量を推定する。これにより、ロボット先端負荷の値を実用に耐える程度の精度で推定する

図10 繰り返し位置精度の比較

図11 偏差積分による動作中の位置偏差の評価

ことができ、定常位置偏差も小さくなる。しかし、この方法では負荷を把握するたびに推定動作を行わなければならないというわずらわしさがある。

- (3) ファジィ理論を用いる方法では、低速から高速までロボットの動作中に同定できるため、重力を利用する方法に比べて有利である。この方法は、実験によればマニピュレータ自重の約80%までの負荷変動に対しても有効で、制御性能を改善できることを確かめた。マニピュレータ先端に把握した負荷重量の変化にともなうパラメータ変動をファジィルールを用いて推定しながら動作することにより、負荷/自重比が大きい場合でも制御性の悪化しないロボット制御法を実現することができた。

参 考 文 献

- (1) 神谷好承, ロボットの高速化・高精度化, 精密機械, 51-9, (昭 60), 2014.
- (2) 小菅一弘, 古田勝久, ロボットアームのロバスト制御, コンピュータール, No.13 (昭 61), 103.
- (3) 大須賀公一, 小野敏郎, マニピュレータの作業座標空間上でのロバストモデル追従制御, 日本ロボット学会誌, 6-2, (昭 63), 117.
- (4) 牧野 洋, 自動機械機構学, 日刊工業(昭 51).
- (5) 山藤和男, 小嶺哲也, ソフトウェア・カム曲線によって駆動されるスライダ・振子型ローダアームの振動制御, 日本機械学会論文集, C編, 55-515 (平 1), 1684.
- (6) John J. Craig, Introduction to Robotics, Addison Wesley Publishing Co., (1986).
- (7) 寺野寿郎, 浅居喜代治, 菅野道夫, ファジィシステム入門, オーム社 (昭 62).