タスク指向のロボット言語を有する 力制御仕上げロボットの開発

神 野 誠^{*1} 尾 崎 文 夫^{*1} 吉 見 卓^{*1} 辰 野 恭 市^{*1}
高 橋 幹 男^{*2} 神 田 眞 三^{*2} 玉 田 保 彦^{*2} 永 滝 真太郎^{*2}

Development of a Force-Controlled Finishing Robot System with a Task-Directed Robot Language

Makoto Jinno^{*1}, Fumio Ozaki^{*1}, Takashi Yoshimi^{*1}, Kyoichi Tatsuno^{*1}
Mikio Takahashi^{*2}, Masakazu Kanda^{*2}, Yasuhiko Tamada^{*2} and Shintaro Nagataki^{*2}

We have developed a force-controlled robot system for finishing tasks such as grinding, chamfering and polishing tasks in machining plants. Some force-controlled robot systems have been developed already. However, these robot systems are not used widely because these robot systems are difficult to use. In particular, it is too difficult for robot users to select or decide the many force control parameters, for example desired force, force control direction, force detecting direction, force control gains and so on, in the conventional robot languages. To solve the problem, we have analyzed finishing tasks and have found that the force control parameters closely depend on the tasks. Thus, we have developed a task-directed robot language. The force control parameters are assigned by designating only three key parameters—a task name (TASK), a tool code (TCODE) and a tool center point (TCP) — and one force control motion command (GMOVE/CMOVE) in the task-directed robot language. This robot system has been applied to chamfering tasks for the edges between the cast surface and the machined surface of machine parts. In this way the effectiveness of the task-directed robot language has been proved.

Key Words: Force Control, Task-Directed Robot Language, Force/Torque Sensor, Finishing, Chamfering

1. は じ め に

現在、産業用ロボットは、スポット溶接、アーク溶接、塗装、電子部品の組立など、様々な分野で幅広く使われている。これらの作業では、ロボットとワーク間に作用する力を積極的に制御する必要がないため、位置制御のロボットでも十分作業を行うことができる。しかし、最近では、グライнда作業、面取り作業、磨き作業などの仕上げ作業への自動化の要求が非常に増えてきている。このような作業では、ロボットが保持したツールをワークに対して適切な力で押し付ける必要があるため、ロボットに力制御機能を備えることが重要である。これまで、力制御に関する一般的な研究[1]~[5]や具体的にグライнда作業、バリ取り作業などを対象とした研究[6]~[11]は数多く行われてきている。また、力制御機能を有するロボットを試作または商品化した例も少なくない[12]~[15]。

しかしながら、いまだにこれらのロボットが広くユーザに利用されていないのが現状である。この原因の一つとして、力制御ロボットのロボット言語を使うことの難しさが上げられる。従来のロボット言語では、力制御に必要なパラメータを一般のユーザが設定しなければならず、大変面倒で使いづらいものであった。また、仕上げ作業と言っても、様々な作業があり、使用する工具や加工時の反力の状態や作業方法などはそれぞれ異なっているため、個々の仕上げ作業に適した力制御方法やそれを支えるロボット言語を構成することが必要である。そこで、筆者らは、これらの問題を解決するために、グライнда作業、面取り作業、磨き作業などの仕上げ作業を対象とした新たなロボット言語を有するロボットシステムを開発した。特に、ロボット言語については、仕上げ作業の場合には、作業によって、押し付け方向や検出すべき反力の方向などの力制御パラメータを概ね決定できることに注目し、作業内容、使用する工具、その作用点、力制御移動命令を指定することで、力制御パラメータを自動的に割り出すタスク指向のロボット言語を開発した。

本論文では、新たに開発した仕上げロボットシステムの構成

原稿受付 1995年9月14日

^{*1}株式会社東芝

^{*2}東芝機械株式会社

^{*1} Toshiba Corporation

^{*2} Toshiba Machine Co., Ltd.

とタスク指向のロボット言語の構成とその特徴、そして、本システムを鋳鉄部品の機械加工後の面取り作業に適用した結果について報告する。

2. ロボットシステム

2.1 システム構成

本システムは、基本的には、ロボット本体、コントローラ、ティーチングペンダントから構成されている。また、必要に応じて、周辺装置としてワーク搬入、搬出装置、ワークテーブル、工具自動交換装置などを構成する。

ロボット本体の外観を Fig. 1 に示す。先端部に 6 軸力覚センサを備えた 6 軸円筒座標型ロボットである。また、1 軸を直動軸とすることで直角座標型ロボットとすることも可能である。各ジョイントは、AC サーボモータ駆動である。空間での先端の最高速度は約 500 [mm/sec]、押し付け力は 0.5~3.0 [kgf] (4.9~29.4 [N]) である。

Fig. 2 にコントローラの構成を示す。コントローラは、マシンインターフェースボード (MMI)、プログラマブルコン

トローラボード (PC)、DSP ボード (DSP) から構成されている。MMI は、ロボット言語の編集やティーチペンダントのファンクションキーの処理、ロボット言語プログラムの翻訳などを行う。PC は、システム管理、各ボード間のデータ転送、周辺装置の制御などを行う。DSP は、軌道生成、AC サーボモータのコントロール、6 軸力覚センサの処理、力制御演算などを行っている。

2.2 制御構成

本システムの制御構成を Fig. 3 に示す。基本的には、位置制御系が構成されており、教示点 (各ステップにおけるツール先端部の最終目標位置、姿勢) に対して目標軌道 (各サンプリングにおけるツール先端部の目標位置、姿勢) が生成される。この目標軌道から各関節の目標位置が逆変換により算出され、各関節はこの目標位置に追従するように位置制御されている。一方、力制御の場合には、6 軸力覚センサのデータがフィードバックされ、力制御演算により、所定の押し付け力となるように修正量が算出され、目標軌道が修正される。

3. ロボット言語

一般に力制御では、力制御ゲイン、押し付け力、押し付け方向やその座標系など多くの力制御パラメータを設定しなければならない。従来の力制御ロボットのロボット言語では、力制御パラメータをロボット言語のプログラム中で記述する方法 [12] [13] [16] と、力制御パラメータをあらかじめテーブル化し、使用するテーブルをロボット言語中で指定する方法 [14] [15] の大きく分けて 2 通りがある。例えば、前者の例で Stanford University で開発された AL 言語 [12] では、以下のような力制御の移動命令になっている。

MOVE P0 TO P1 WITH FORCE=<strength of contact force> ALONG<axis of force direction> OF<force frame>

このような方法で力制御パラメータをロボット言語のプログラム中に詳細に記述するのは大変面倒であり、ユーザがそれらの内容を十分に理解して記述するのは非常に困難である。一方、後者の場合、力制御パラメータを詳細に記述する必要はないが、テーブルの数が不用意に増大する恐れがあり、どのテー

Fig. 1 Appearance of the manipulator

Fig. 2 Controller architecture

Fig. 3 Force control block diagram

ブルが何の仕上げ作業を行うものかを管理し、プログラム中で選択するのも容易でない。また、ロボット言語のプログラムからはその力制御パラメータの内容を推測することは困難である。

さらに、従来のロボット言語では、いろいろな工具を用いて行う様々な仕上げ作業に対応できるだけの力制御の移動命令とその制御方法、その制御のための力制御パラメータを十分に備えているとは言えない。

以上のような従来の問題点を考慮して、仕上げ作業を対象とした力制御ロボットのロボット言語への要求事項をまとめると以下になる。

- (1) 仕上げ作業を行う上で必要な力制御の移動命令を備えていること。
- (2) 作業条件、つまり力制御パラメータの設定が容易で理解しやすいこと。
- (3) それぞれの仕上げ作業に応じて詳細な力制御パラメータが設定できること。

以下の節では、筆者らが開発した仕上げ作業を対象としたロボット言語の内容について、上記の要求事項を各節に対応させて述べる。

3.1 仕上げ作業における力制御方法と移動命令

仕上げ作業の場合、作業の種類は大きく分けて、表面をなめらかに仕上げる作業と表面を所定の形状に仕上げる作業の2種類に分類することができる。具体的な作業としては、前者の場合、ワークの表面の研磨、研削作業や錆落とし、あるいは、ワーク稜線に対して行う面取りや機械加工バリ取りなどの作業がある。後者の場合、切断作業や突起の除去、大きなバリの除去、ビート研削などの作業がある。従って、仕上げ作業時の移動命令も作業目的にあわせて基本的には2種類に対応できることになる。以下にそれぞれの移動命令の制御方法を示す。

まず、表面をなめらかに仕上げる作業では、Fig. 4(a)に示すように、ツールを概ね一定速度で送り、ツールの進行方向に対して概ね直交した方向に力制御による目標軌道の修正を行う。従って、ツール先端の目標軌道 r の基本式は次式のように表せる。

$$r = r_p + \int K_f (F_d - F) F_a dt \quad (1)$$

(a) GMOVE

(b) CMOVE

Fig. 4 Force control method

Fig. 5 Reaction force and modification

r_p は、教示点に基づき算出される位置制御時の目標軌道、 K_f は力制御ゲイン、 F_d は指定された方向への目標押し付け力の大きさ、 F は指定された方向の検出力の大きさ、 F_a は修正方向単位ベクトルを表している。右辺第2項は修正量ベクトルを表している。なお、本論文では、修正方向を力制御方向と呼ぶことにする。

本システムでは、検出力 F の方向と修正方向ベクトル F_a を全く独立に扱っている点の特徴である。例えば、Fig. 5 に示すように鋳物の機械加工面と鋳肌面の稜線を円筒形のツールで面取りするような場合を考える。この場合、鋳肌面に沿った方向や 45 度方向に修正を行うより、寸法精度の出ている機械加工面に沿った方向に修正する方が作業が確実にできる。一方、反力の方向は、機械加工面に沿った方向が必ずしも支配的ではなく、概ね 45 度方向になる。面取り作業をしている状態では、切削による主分力が作用するため、ツール回転面内の方向で、アップカットやダウンカットや稜線の凹凸の状態によって変化する。従って、仕上げ作業の場合、反力の方向（力の検出方向）と力制御方向（修正方向）を独立に考えた方がよりスムーズに力制御が行える。

次に、表面を指定の形状に仕上げる作業では、Fig. 4(b) に示すように、ツールの進行方向に対して力制御を行う。この時、ツールは所定の軌跡を通るが、押し付け力の状態でツールの送り速度は変化することになる。従って、ツール先端の目標軌道 r の基本式は直線移動の場合、次式のように表せる。

$$r = r_s + \int K_f (F_d - F) \frac{r_e - r_s}{|r_e - r_s|} dt \quad (2)$$

この場合、 r_s は初期位置（前ステップの教示点）を、 r_e は終点位置（教示点）を表している。

また、式(1)を適用した場合でも、右辺第 2 項の修正量を制限することで所定量以上削り込まないようにすることは可能である。従って、繰り返し作業を行うことで所定の形状に仕上げることが可能である。

このようなことから、本ロボット言語では基本的に 2 種類の力制御移動命令を備えている。前者を GMOVE、後者を CMOVE と呼んでいる。この 2 種類をベースに直線移動や円弧移動、絶対位置あるいは相対位置への移動などの命令を備えることで、基本的な仕上げ作業を行うことができる。

3.2 タスク指向のロボット言語

前述のように力制御パラメータをいかに容易に設定でき、理解しやすいかが力制御ロボットのロボット言語のポイントの一つである。筆者らは、特に仕上げ作業ということに注目し、作業内容、使用する工具、その作用点、力制御移動命令を指定することで力制御パラメータを自動的に割り出すタスク指向のロボット言語を開発した。以下にその根拠と内容について述べる。

力制御ゲインや力センサのカットオフ周波数などの力制御パラメータは、通常、ロボット言語のプログラム中で設定することは少なく、固定された値となっている場合が多い。しかし、仕上げ作業では決められた工具を常に保持している訳ではなく、いろいろな工具を適宜交換して作業を行うため、使う工具の質量や回転数、工具先端の作用点の機械的コンプライアンスなどは、使う工具や行う作業によって様々である。そのようなすべての工具や作業に対して適切な力制御ゲインや力センサのカットオフ周波数などを設定するのは非常に困難である。このようなことから力制御ゲインや力センサのカットオフ周波数などの力制御パラメータは、使用する工具や行う作業によって設定することが必要である。

一方、反力方向や力制御方向などの力制御パラメータは、使用する工具や作業内容などによって概ね決まってしまう場合が多い。例えば、ディスクグラインダで比較的なめらかな表面の研削作業を行うような場合は、力制御方向はワークの法線方向、前述の Fig. 5 に示すような面取り作業では、力制御方向は機械加工面に沿った方向など概ね決まってしまう。逆に、作業内容や使用する工具が力制御パラメータを概ね決定するとも言える。

このように、仕上げ作業の場合、作業内容や使用する工具などによって力制御パラメータが設定できることが必要であったり、また、作業内容や使用する工具などによって、力制御パラメータが決まってしまう場合が多い。そこで、これらのことに注目し、タスクを指定することで力制御パラメータが設定されるタスク指向のロボット言語とした。

タスクを指定する要素は、作業内容 (TASK)、使用する工具 (TCODE) の 2 項目の他に、工具のどの部分で作業するかを設定する作用点 (TCP: TOOL CENTER POINT)、および力制御移動命令 (GMOVE/CMOVE) の 2 項目を加え、合計 4 項目とした。この 4 項目から力制御パラメータを自動的に割り出す構成となっている。なお、本論文では、この 4 項目をあわせてタスクと呼んでいる。つまり、タスクとは、何の作業をどの工具のどの部分でどのように作業するかということを意味している。

作用点 (TCP) は、工具がワークへ接触する位置や方向によっても加工条件や機械的コンプライアンスなどが変化する場合があるためタスクを指定する項目に加えている。また、本ロボット言語では、基本的に 2 種類の力制御移動命令(力制御方法)を備えているため、力制御移動命令によっても力制御のパラメータを設定する必要がある。このような理由で作用点 (TCP) と力制御移動命令 (GMOVE/CMOVE) をタスクを指定する項目に加えている。

タスク指向のロボット言語のプログラムの一部を Fig. 6 に示す。この場合、まず、TASK, TCODE, TCP によりグラインダの先端部を使ってグラインダ作業を行うということを宣言している。そして、教示点 P1 に向かって、目標押し付け力 1[kgf] (9.8[N]) で概ね進行方向に直交する方向に力制御しながら、速度 10[mm/sec] で直進することを意味している。

このプログラムを実行する時の力制御パラメータを割り出す仕組みは以下になっている。Fig. 7 にロボット言語プログラム(Program)と作業定義テーブル(Skill Pointer Table)、動作テーブル (Skill Table)、工具テーブル (Tool Table) の関係を示す。作業定義テーブルは、TASK, TCODE, TCP, G/CMOVE の組合せとその組合せに対応する動作テーブルポ

Step	Command
1	TASK=GRIND
2	TCODE=GRINDER
3	TCP=TIP
4	GMOVE P1,S10,F1

Fig. 6 Example program of grinding task

Fig. 7 Program/table relation

インタからなっている。従って、実行時にロボット言語のプログラム中で指定された TASK, TCODE, TCP, G/CMOVE から、それらの組合せと一致する作業定義テーブルがまず選択される。次に、選択された動作定義テーブル中の動作テーブルポインタから動作テーブルが選択される。動作テーブルには仕上げ作業に必要な力制御パラメータが記述されており、その力制御パラメータに基づき力制御が実行される。一方、TCODE, TCP の組合せからは、工具テーブルが選択され、工具重量、重心位置、工具座標系などが設定される。作業定義テーブル、動作テーブル、工具テーブルは、ロボットメカ側の経験やノウハウに基づきあらかじめデータベース化されたものがユーザ側へ提供される仕組みになっている。もちろんユーザ側の経験やノウハウを盛り込むことも可能である。

このようにタスク指向のロボット言語では、ユーザがロボット言語のプログラム中に TASK, TCODE, TCP, G/CMOVE を適宜記述することでその組合せに適した動作テーブル、つまり力制御パラメータが自動的に設定されることになる。このようにタスクを指定することで容易に力制御パラメータの設定が行える。また、力制御パラメータの内容もタスクから容易に推測することができる。

3.3 力制御パラメータ（動作テーブル）

力制御ロボットのロボット言語では、力制御パラメータが容易に設定できるだけでなく、それぞれの仕上げ作業に応じて詳細に設定できなければならない。本タスク指向のロボット言語では、動作テーブルには仕上げ作業に必要な力制御パラメータが記述されており、この動作テーブルを編集することで力制御の条件を詳細に設定することができる。

動作テーブル内の主な力制御パラメータは、押し付け方向パラメータ、検出力パラメータ、力パターンパラメータ、力制御ゲインパラメータ、スキップパラメータなどである。なお、目標押し付け力と工具の送り速度は、研削量や面取り量などを調整するパラメータなので、動作テーブル内には割り当てず、ロ

Fig. 8 Force control direction

Fig. 9 Force pattern

ット言語内で直接記述することにした。以下に力制御パラメータの詳細について示す。

(1) 押し付け方向パラメータ

押し付け方向パラメータは、基準方向とその定義座標系および押し付け方法から構成されている。押し付け方法は、Fig. 8 (a)に示すように、基準方向に直接押し付ける方法と、Fig. 8 (b)に示すように、ツールの進行方向と基準方向との外積方向に押し付ける方法の2種類がある。

基準方向をその定義座標系を含めて D と表すと、押し付け方向 F_a は、前者の場合

$$F_a = D \quad (3)$$

と表せ、後者の場合

$$F_a = V \times D \quad (4)$$

と表せる。ここで、 V はツールの送り方向ベクトルである。

一般的にグラインダ作業や磨き作業など平面的、曲面的な作業については前者の方法で、面取りやバリ取りなどの直線的、曲線的な作業は後者の方法を用いる。

基準方向としては、例えば Fig. 8 に示すようなワークの場合、前者の場合 $-Z_w$ 、後者の場合 Z_w が基準方向となる。

(2) 検出力パラメータ

検出力パラメータは、検出方向とその定義座標系から構成されている。検出方向には、通常の並進力 F_x, F_y, F_z の他、ツール重心回りのモーメント M_x, M_y, M_z 、平面内合成力 F_{xy}, F_{yz} , F_{zx} 、全合成力 F_{xyz} などから選択できる。

(3) 力パターンパラメータ

力パターンパラメータは、Fig. 9 に示すように、押し付け力のパターンを決めるパラメータである。ツールの送りの加減速

Fig. 10 Skip function

時には、ワークを削り込む場合があるため、目標押し付け力を減少させることで、削り始めと削り終わりの削り込みを防ぐことができる。

(4) 力制御ゲインパラメータ

力制御ゲインパラメータは、式(1)(2)に示す力制御ゲインである。同式だけでなく、タスクによって他の制御則に切り替えることも本構成なら容易である。

(5) スキップパラメータ

スキップパラメータは、進行方向（経路長）誤差を吸収するためのスキップ機能を設定するパラメータである。式(1)で示す力制御の場合、ワークの押し付け方向の形状誤差は吸収することはできるが、進行方向の誤差を吸収することはできない。例えば、Fig. 10に示すようなワークの面取り作業をする場合、教示点P1を経由してP2に向かう時、たとえ力制御を行っていても、P1付近ではツールがワークから離れてしまい、スムーズな面取り作業を行うことはできない。本来なら、P1'付近でその方向への移動を終え、次のP2に向かって作業を行うべきである。

スキップ機能では、教示点P1近傍（スキップ判定領域内）において、反力の状態を監視している。そして、指定された反力の大きさの範囲より増大または減少した時に、コーナー部にさしかかったとして（P1'を検出）、P1に向かう移動を終え、次のステップを実行、すなわちP2に向かって移動する。これにより、鋳物など形状寸法に誤差のあるような場合でも、力制御方向ならびに進行方向の誤差を吸収し、仕上げ作業を行うことができる。

以上のように動作テーブルにより力制御パラメータを詳細に設定することができる。このような構成ならば、動作テーブル内にパラメータを追加することで、対象作業の適用範囲を広げるための制御則の改良なども比較的容易に行える。

本章で示したように本タスク指向のロボット言語は、仕上げ作業を行う上で必要な基本的な力制御の移動命令を備えており、また、力制御パラメータの設定が容易で理解しやすく、かつ、それぞれの仕上げ作業に応じて詳細な力制御パラメータが設定できるという特徴を備えている。

4. 適用作業

本ロボットシステムの適用作業の一例として、鋳鉄部品の機械加工後の面取り（機械加工バリ取り）作業を実施した。Fig.

Fig. 11 Chamfering application

Fig. 12 Part of workpiece

PROGRAM	COMMENT
TASK=CHAMFER	Assign a task
TCODE=ROUTER	Assign router as a tool
TCP=TIP	Assign tool center point
DFORCE=0.5	Assign desired force
SPEED=20	Assign arm tip speed
MOVE P0	Move to P0
FRESET	Force sensor reset
TOUCH P1,S5.0	Move straight to direction P1 with 5.0 mm/sec speed until the force exceeds an assigned value
MOVERS P2	Move straight to P2 (Relational position)
TOOL_ON	Make the tool run
SKIP=ON	Skip function is valid
GMOVES P3	Move to P3 with force = DFORCE
GMOVES P4	
GMOVES P5	
GMOVES P6	
GMOVES P7	
SKIP=OFF	Skip Function is invalid
MOVERS P8	
TOOL_OFF	Make the tool stop
END	

Fig. 13 Robot language program

11に面取り作業の様子を示す。

Fig. 12に示すような形状の機械加工面と鋳肌面の稜線部を面取りする場合の加工プログラムをFig. 13に示す。同プログラムでは、まず、TASK、TCODE、TCPおよび加工条件を

Fig. 14 Chamfering force

(押し付け力と送り速度)宣言した後に、ロボットの動作が記述されている。ワーク形状にバラツキが大きい場合には、同図にあるようにワークに接触して加工スタート位置を認識してから作業を開始する。なお、適用作業で対象とした鋳物ワークは、同図のような単純な形状だけでなく、様々な形状の稜線が数十カ所あるワークである。

動作テーブルでは、押し付け方向を Fig. 8(b)と同様に、検出力方向を全合成力に、スキップ判定の数値値を目標押し付け力の20%以下などに設定している。そして、押し付け力: 0.5[kgf] (4.9[N]), ツール送り速度: 20[mm/sec]の加工条件で行うことにより、概ねC0.5からC1.0で面取りすることができた。同プログラムを実行した時の、反力の様子を Fig. 14に示す。定常状態では安定に力制御できていることが確認できる。また、コーナー部では、スキップ機能によりコーナー部を認識し次の動作命令を実行している。この時、若干反力が変動しているが、面取り状態としては問題なかった。

適用作業を実施したシステムは、ワークの搬入装置、搬出装置、工具自動交換装置などを含んだシステムである。ワーク搬入から搬出までの一連の面取り作業に要する仕上げ時間は約30分で、現状の熟練者による作業時間30分とほぼ同一の時間で行うことができ、実用的な仕上げ時間であることを確認した。また、本タスク指向のロボット言語により一連の仕上げ作業の記述と力制御パラメータの設定が容易であることを確認した。

5. お わ り に

グラインダ作業、面取り作業、磨き作業などの仕上げ作業の自動化を対象とした力制御仕上げロボットを開発した。

本ロボットシステムでは、仕上げ作業を実現するための2種類の基本的な力制御移動命令を備えている。そして、仕上げ作業を対象とした場合、作業内容や使用する工具によって力制御ゲインや力センサのカットオフ周波数などの力制御パラメータが決められた方が便利であったり、また、逆に、反力の方向や力制御方向などのパラメータは、作業内容や使用する工具によって概ね決まってしまうことに注目し、タスクを指定する作業内容 (TASK), 工具 (TCODE), 作用点 (TCP), 移動命令 (GMOVE/CMOVE) の4項目から力制御パラメータを自動的に割り出すタスク指向のロボット言語を開発した。

さらに、本ロボットシステムの適用作業の一例として、鋳鉄部品の機械加工後の面取り (機械加工バリ取り) 作業を実施し

たところ、本タスク指向のロボット言語により、力制御パラメータの設定が容易にできることを確認したとともに、所望の面取り作業が可能であることを確認した。

今後、本ロボットシステムを様々な仕上げ作業に適用する予定である。

参 考 文 献

- [1] J. K. Salisbury: "Active Stiffness Control of a Manipulator in Cartesian Coordinates," IEEE Conf. on Decision and Control, pp. 95-100, 1980.
- [2] M. H. Raibert and J. J. Craig: "Hybrid Position/Force Control of Manipulators," Trans. ASME, J. of Dynamic Systems, Measurement, and Control, vol. 102, pp. 126-133, 1981.
- [3] J. P. Merlet: "C-surface applied to the design of a hybrid force-position robot controller," 1987 Int. Conf. on Robotics and Automation, pp. 1055-1059, 1987.
- [4] T. Yoshikawa and A. Sudou: "Dynamic Hybrid Position/ Force Control of Robot Manipulators: On-line Estimation of Unknown Constraint," 1990 Int. Conf. on Robotics and Automation, pp. 1231-1236, 1990.
- [5] L. Whitcomb, S. Arimoto, T. Naniwa and F. Ozaki: "Experiments in Adaptive Model-Based Force Control," 1995 IEEE Int. Conf. on Robotics and Automation, pp. 1846-1853, 1995.
- [6] G. Plank and G. Hirzinger: "Controlling a Robot's Motion Speed by a Force-Torque Sensor for Deburring Problems," IFAC Information Control Problems in Manufacturing Technology, pp. 97-102, 1982.
- [7] D. E. Whitney and M. L. Brown: "Metal Removal Models and Process Planning for Robot Grinding," SME Robots 11/17th ISIR, pp. 19-29, -44, 1987.
- [8] L. Liu, B. J. Ulrich and M. A. Elbestawi: "Robot Grinding Force Regulation: Design, Implementation and Benefits," 1990 IEEE Int. Conf. on Robotics and Automation, pp. 258-265, 1990.
- [9] 神野, 吉見, 阿部: "遠隔グラインダ作業ロボットの研究", 日本ロボット学会誌, vol. 10, no. 2, pp. 244-253, 1992.
- [10] S. Liu and H. Asada: "Task-Level Robot Adaptive Control Based on Human Teaching Data and Its Application to Deburring," 1993 American Control Conference, pp. 1414-1417, 1993.
- [11] Y. Nonaka, S. Sakaue, Y. Yanagihara and K. Yokoshima: "Development of an Impeller Grinding Robot System and a Gyromoment Compensated Compliance Control," 1995 IEEE Int. Conf. on Robotics and Automation, pp. 2084-2089, 1995.
- [12] S. Mujtaba and R. Goldman: AL Users' Manual. Stanford Artificial Intelligent Lab., Stanford Univ., 1982.
- [13] 渡辺 他: "力覚センサ搭載6軸垂直直多関節ロボット", 三菱電機技報, vol. 64, no. 5, pp. 11-14, 1990.
- [14] 樽沼, 柏木, 山田: "グラインダ作業用力制御ロボットの開発", 第9回日本ロボット学会学術講演会予稿集, pp. 47-48, 1991.
- [15] 久良, 田中: "インダストリアルロボットにおけるセンサ利用の現状", 日本ロボット学会誌, vol. 9, no. 3, pp. 342-348, 1991.

[16] 新井 他：産業用ロボット言語 “SLIM”. pp. 177-185, 財団法人日本

規格協会, 1994.

神野 誠 (Makoto Jinno)

1960年7月29日生。1983年慶應義塾大学工学部機械工学科卒業。1985年同大学大学院修士課程(機械工学専攻)修了。同年(株)東芝入社。ロボットの機構および力制御機能を有するロボットシステムの研究開発に従事。現在、研究開発センター、機械・エネルギー研究所、研究主務。1994年度日本機械学会技術奨励賞受賞。日本機械学会会員。

(日本ロボット学会正会員)

吉見 卓 (Takashi Yoshimi)

1961年10月11日生。1985年大阪大学基礎工学部機械工学科卒業。1987年同大学大学院修士課程(機械工学分野)修了。同年(株)東芝入社。ロボットの力制御技術およびタスクプランニング技術の研究・開発に従事。現在、研究開発センター、機械・エネルギー研究所、研究主務。

(日本ロボット学会正会員)

高橋幹男 (Mikio Takahashi)

1940年1月5日生。1962年山形大学機械工学科卒業。同年東芝機械(株)入社。仕上げ加工ロボットの開発に従事。現在、東芝機械テクノ(株)主幹技師。日本ロボット工業会協賛会員。

玉田保彦 (Yasuhiko Tamada)

1947年6月22日生。1971年東京理科大学理学部応用数学科卒業。同年東芝機械(株)入社。FMS, CAD/CAM/CAE等の研究開発を経て、ロボットの制御ソフトウェアの研究開発に従事。現在、制御システム事業部、制御システム技術開発部、主査。

(日本ロボット学会正会員)

尾崎文夫 (Fumio Ozaki)

1958年2月9日生。1981年京都大学工学部電気工学科卒業。1983年同大学大学院修士課程(電気工学専攻)修了。同年(株)東芝入社。力制御機能を有するロボットシステムの研究開発に従事。ロボットの学習機能に興味を持つが、未だ研究に着手できずにいる。現在、研究開発センター、機械・エネルギー研究所、研究主務。

(日本ロボット学会正会員)

辰野恭市 (Kyoichi Tatsuno)

1950年1月1日生。1972年大阪大学電気工学科卒業。1973年(株)東芝入社。保守点検ロボットおよび光応用計測技術の研究・開発に従事。現在、研究開発センター、機械・エネルギー研究所、研究主幹。工学博士。1986年計測自動制御学会技術賞、注目発明賞受賞。日本機械学会、計測自動制御学会、IEEEなどの会員。

(日本ロボット学会正会員)

神田眞三 (Masakazu Kanda)

1943年3月30日生。1961年芝浦工機(東芝機械(株)前身)入社。1968年神奈川大学機械工学科卒業。仕上げロボットの開発に従事。現在、新規事業推進部、ロボット開発部、開発担当課長。

(日本ロボット学会正会員)

永滝真太郎 (Shintaro Nagataki)

1948年2月15日生。1971年早稲田大学理工学部機械工学科卒業。1973年同大学大学院修士課程(機械工学専攻)修了。同年東芝機械(株)入社。仕上げ加工ロボットの開発に従事。現在、新規事業推進部、ロボット開発部、主査。