知覚情報を用いた動的軌道生成*

竹内 郁雄^{*1}, 亀島 鉦二^{*2}
浜田 朋之^{*2}, 土屋 雅弘^{*2}

Perception Based Dynamic Trajectory Generation

Ikuo TAKEUCHI, Kohji KAMEJIMA,
Tomoyuki HAMADA and Masahiro TSUCHIYA

A dynamic motion generation method is presented for interactive robot operation. This method is aimed to make a robot execute the task of touching an object accurately with abstract instruction. The class of motion, unified regular motion and its adaptive motion, is presented within the framework of online update procedure for an a priori geometric trajectory, essentially specified in terms of task knowledge. In this presentation, planned trajectory is converted into basic input for actuators of robot with a real-time kinetics simulator and adapted through tactile and visual feedback to the object to be manipulated. The dynamic trajectory generation method is verified through experimental studies; for example, handling the object.

Key Words: Perception, Dynamics, Robot Manipulator, Trajectory Generation, Real-Time Simulator

1. 緒 言

フランジの組立や分解のような作業をロボットを用いて実行する場合には、ロボットと対象物との接触およびオペレータとロボットとのインタラクションという問題が生じる。つまり、ロボットに的確に作業を実行させるためには、接触時の位置と力の両方の情報を精密に管理しながらロボットの軌道を生成しなければならない。また、ロボットへの動作指示は作業の進捗状況に応じて、例えば、工具の装着状態を確認しながらオペレータが与えるので、ロボットはランダムに動作のスタートとストップを繰り返すことになる。このために、この繰り返しを安定に実行することが要求される。これらの問題を解決するためにはロボットの軌道生成過程に力学知識、すなわち運動方程式を導入すればよい。

しかし、単に力学知識を導入するだけでは不十分である。つまり、エンドエフェクタと対象物の接触時の相対的な位置関係や力関係を示す知覚情報に応じてロボットの軌道をダイナミックに適応させるわけである

が、この接触に関する情報はロボットが現実動作して初めて獲得可能なものであるのであらかじめ計画することができない。また、かなりの質量を有するロボットが動作することを考えると軌道の適応は力学的拘束下でリアルタイムに実行されなければならない。このために、単に力学知識を導入しただけでは動作軌道つまり、作業を実行可能な軌道の生成過程が複雑になるのでオペレータの作業指示を簡単に実現することができない。そこで、ロボットの動作軌道をインタラクティブに生成するためには、定形動作つまり、形状が既知の物体をハンドリングするように動作を一定の手順で表現できる動作と、これを確実に実行するために位置や姿勢を微妙に修正する適応動作をまとめて類似動作として単一の手続きで表現することが必要になる⁽¹⁾。

本報では、リアルタイムシミュレータを用いて力学知識を表現することにより知覚情報をロボットの軌道処理に直接フィードバックする仕組と、この仕組のもとで類似動作を単一の表現で実行する方法を示す。さらに、フランジの組立および分解作業を例に、本手法を用いて具体的な類似動作を実行した結果について述べる。

* 原稿受付 平成2年6月27日。

^{*1} 正員、(株)日立製作所機械研究所 (〒300 土浦市神立町502)。^{*2} (株)日立製作所機械研究所。

2. 軌道生成アルゴリズム

三次元物理空間でオペレータの指示どおりの作業を実行させるためにはさまざまなセンサ情報を用いて環境の変化を知り、この変化に適応するように作業知識を用いて計画した軌道を修正する必要がある。ところで、物理空間での物体の動きは運動方程式により支配されており、軌道修正は力学的拘束を満たすことが要求される。したがって、ロボットの動作軌道を生成するためにはオンラインで運動方程式を解き、常に力学的条件を満足するように計画軌道を修正することが必要である。そこで、運動方程式、つまり力学知識を用いて作業知識とセンサ情報を融合処理することによりロボットの動作軌道を生成する方式を検討する。

まず、通常利用できる作業知識および力学知識とセンサ情報の分類を表1に示す。作業知識およびセンサ情報には力に関するものと位置に関するものがある。一方、力学知識は位置から力を求める Inverse Dynamics と力から位置を求める Direct Dynamics の2通りの手続きとして表現される。Inverse Dynamics は位置に関する作業知識と力に関するセンサ情報を結び付けるのに利用可能であり、Direct Dynamics は力に関する作業知識と位置に関するセンサ情報を結び付けるのに利用可能である。しかし、ロボットの動作開始以前に確定しているという意味で静的な知識とロボットが動作後初めて確定するという意味で動的なセンサ情報を融合することは簡単ではない。

そこで、知識とセンサ情報を融合するために図1に示す処理方式を開発した。本方式では、計画レベルでは、作業知識を用いてオペレータからの作業指示を理解するとともに、作業環境を認識することにより静的な軌道表現としての幾何学的な点列を求める⁽²⁾。ロボット動作時には、まず、この点列をローパスフィルタ(LPF)でスムージングすることにより波形データである計画軌道(θ)を順次生成する。ついで、運動制御レベルでは、シミュレータを用いてロボットの動きを支

配している力学的拘束を満たすように実時間で動作軌道($\hat{\theta}$)を次の手順で生成する。まず、計画軌道(θ)に対して Inverse Dynamics の演算を行うことにより計画トルク(τ)を求め、次に、オンラインでこの計画トルクに力覚および視覚センサからの情報を融合して得た力の情報($\Delta\tau$)を加えて動作トルク($\hat{\tau}$)を求める。さらに、この動作トルクに対して Direct Dynamics の演算を行うことにより動作軌道($\hat{\theta}$)を生成する。最後に、実行レベルでは、この動作軌道に基づいてロボットメカニズムを制御するとともに、センサ融合を行うための力覚および視覚の各センサ情報を取り込んで運動制御レベルへフィードバックする。

以上のように、本方式ではオンラインでアダプテーション処理を行って実環境に適応しているため、予測不可能な摩擦成分や弾性項を含んだ完全なモデルを必要としない。そこで、Inverse Dynamics および Direct Dynamics の演算には、設計パラメータのみから成る Newton-Euler 法に基づく Luh, Walker, Paul の計算法をベースにしたアルゴリズムを用いる⁽³⁾。また、本

表 1 知識とセンサ情報の分類

Task Knowledge	Dynamics Knowledge	Sensor Information
Work Space Model	Inverse Dynamics	Vision (force) Touch Force Sense
Work Force	Direct Dynamics	Vision (position)

図 1 知識とセンサ情報の融合処理方式

アルゴリズムでは演算の高速化を図るために、添字 $i-1$ および $i+1$ が付いた変数は 1 サイクル前の $i-1$ 関節および $i+1$ 関節のデータを用いて、各関節ごとに並列に演算を行う⁽⁴⁾。この演算アルゴリズムを以下に示す。

Inverse Dynamics

$$\begin{aligned}\omega_i &= A_i^{i-1}(\omega_{i-1} + z_0 \dot{\theta}_i) \\ \dot{\omega}_i &= A_i^{i-1}\{\dot{\omega}_{i-1} + z_0 \ddot{\theta}_i + \omega_{i-1} \times (z_0 \dot{\theta}_i)\} \\ \dot{v}_i &= \dot{\omega}_i \times r_i + \omega_i \times (\omega_i \times r_i) + A_i^{i-1} \dot{v}_{i-1} \\ \ddot{v}_i &= \omega_i \times (\omega_i \times s_i) + \dot{\omega}_i \times s_i + \dot{v}_i \\ f_i &= m_i \ddot{v}_i + A_i^{i+1} f_{i+1} \\ n_i &= I_i \dot{\omega}_i + \omega_i \times (I_i \omega_i) + m_i(r_i + s_i) \times \ddot{v}_i \\ &\quad + r_i \times A_i^{i+1} f_{i+1} + A_i^{i+1} n_{i+1} \\ \tau_i &= (A_i^{i-1} z_0)^T n_i\end{aligned}$$

Direct Dynamics

$$\begin{aligned}\hat{\theta}_i &= [\hat{\tau}_i - (A_i^{i-1} z_0)^T \{I_i H + \omega_i \times (I_i \omega_i) \\ &\quad + M \times (\omega_i \times (\omega_i \times s_i) + H \times s_i + H \times r_i \\ &\quad + \omega_i \times (\omega_i \times r_i) + A_i^{i-1} \dot{v}_{i-1}) \\ &\quad + r_i \times A_i^{i+1} f_{i+1} + A_i^{i+1} n_{i+1}\}] \\ &\quad / [(A_i^{i-1} z_0)^T \{I_i A_i^{i-1} z_0 \\ &\quad + M \times (A_i^{i-1} z_0 \times (r_i + s_i))\}]\end{aligned}$$

ここに、

$$H = A_i^{i-1}\{\dot{\omega}_{i-1} + \omega_{i-1} \times (z_0 \dot{\theta}_i)\}$$

$$M = m_i(r_i + s_i)$$

ここで、 i は関節番号、 n は自由度数、 θ_i は各関節の回転角、 z_0 は単位ベクトル、 r_i は i 座標系による $i-1$ 座標系原点から i 座標系原点へのベクトル、 A_i^{i-1} は i 座標系から $i-1$ 座標系への回転に関する変換行列、 ω_i はリンク i の角速度ベクトル、 $\dot{\omega}_i$ はリンク i の角加速度ベクトル、 v_i はリンク i の直線加速度ベクトル、 $\ddot{v}_i$ はリンク i の重心における直線加速度ベクトル、 f_i はリンク $i-1$ からリンク i へ作用する力のベクトル、 n_i はリンク $i-1$ からリンク i へ作用するモーメントのベクトル、 s_i は i 座標系による座標原点からリンク i の重心に向かうベクトル、 m_i はリンク i の質量、 I_i はリンク i の重心に対する慣性行列である。

このように、リアルタイムシミュレータを用いて力学知識を表現することにより、作業知識という静的な情報とセンサ情報という動的な情報を融合することができるようになった。これは知覚情報を用いて軌道をダイナミックに生成できることを意味しており、これによって、オペレータの意図に従って作業を的確に実行できる軌道を実時間で生成することが可能となる。

3. 類似動作の処理

本章では、ハンドリングを例にして、類似動作を具体的に表現してみる。2章に示した概念に従って類似動作をアルゴリズム表現した例を図2に示す。定形動作として幾何学的に計算した軌道を用いて的確なハンドリングを実現するためには視覚、力覚による位置や力に関する知覚情報を獲得しながら、グリッパと対象物の位置ずれを修正する必要がある。そこでまず、視覚情報を用いてロボットをグリッピング位置まで誘導する。次に、誘導点で、横方向の一定力を検出するまでフィンガを初期位置から内方向へ移動して両方のフィンガの移動量の差からグリッパの中心と対象物の中心のずれを認識する。このとき、点対称の物体を2本のフィンガで把持するので、グリッパを90度回転することにより二次元の位置ずれ dx および dy を求める。このずれ量に実験より求めた弾性係数 k_x および k_y を乗ずることにより、アーム先端への外力ベクトル $f_f = (k_x dx, k_y dy, 0)$ を演算する。この処理フローを図3に示す。さらに、この外力に対する各関節トルク $\Delta \tau$ を演算してリアルタイムシミュレータに入力することにより、ずれを修正するための軌道をダイナミックに生成してグリッパ位置を修正する。

以上の一連の動作は実行レベルで自律的に生成される。つまり、図2の処理アルゴリズムにおいて、グリッ

図2 類似動作処理アルゴリズム

パ閉の指示に対して二次元の位置ずれが閾値以下になれば軌道修正が終了し、グリップングも終了する。また、このアルゴリズムには対象物およびグリップに固有の幾何学的なパラメータは含まれていない。このように、本方式により類似動作を定形的な動作として単一のアルゴリズムで表現することができた。

4. ハードウェア構成と実行結果

前章に示した類似動作を実行するためのハードウェア構成を図4に示す。図1に示した計画レベルの処理をグラフィックWSが行い、運動制御レベルの処理をネットワークサーバおよびリアルタイムシミュレータが行っており、実行レベルの処理をその他のブロックが実行している。リアルタイムシミュレータ、ネットワークサーバおよびロボットコントローラは黒板型I/Fで接続している。また、本システムで用いているグリップおよび視覚カメラを装着したロボットの手先

構成を図5に示す。グリップシステムはLORD製のインスツルメントドグリップシステムで、独立に駆動できる2本のフィンガを持ち、各フィンガは六軸力覚センサおよび位置センサを備えている。この二つのセンサ情報により図3に示したずれ量の認識を行う。また、視覚カメラはCCD小形カメラを用いており、この情報を処理して環境地図の作成および修正を行う。さらに、手先にはボルト着脱用のインパクトレンチを取付けてある。

本システムでフランジの組立および分解作業を実行した結果、対象物のハンドリング時のグリップと対象物の位置のずれ量を0.15 mm以下にすることができた。また、オペレータからのインタラクティブな指示で作業を安定に実行することができた。このことは、視覚および力覚情報を融合することにより視覚のみに比べてハンドリング時のアームの相対的な位置精度が一桁向上したことを示している。これにより、オペレ

図3 ずれ量認識処理フロー

図5 ロボットの手先構成

図4 ハードウェア構成

ータはグリッパと対象物の接触の際の細々とした指示から解放されることになる。このように、ロボットの軌道生成過程に力学知識を導入することにより、簡単にロボットを操作できることを明らかにした。

5. 結 言

リアルタイムシミュレータで力学知識を表現することにより作業知識とセンサ情報を融合処理する方式を示した。本方式を用いれば類似動作を普遍的な手続きで表現することができ、ロボットをインタラクティブに操ることが可能となった。また、プロトタイプを用いた実験により類似動作の安定な実行を確認した。

本研究において、ロボットコントローラ I/F に関し

てご教示いただいた日立京葉エンジニアリング(株)、寺本和良氏並びに椎名司氏に感謝の意を表する次第である。

文 献

- (1) 竹内・ほか2名, 知覚情報を用いた軌道生成による類似動作の表現, 第5回 SICE ヒューマンインタフェースシンポジウム予稿集, (1989-10), 433-436.
- (2) 浜田・亀島, イメージ・ベースド・オペレーション, 第3回 SICE ヒューマンインタフェースシンポジウム予稿集, (1987-10), 149-152.
- (3) 川崎, ロボットアームの動力学計算法, 計測と制御, 25-1(1986), 23-29.
- (4) 竹内・亀島, マニピュレーショントルクの動的推定法, 文献(2)の389-392ページ.