学術・技術論文

軌道計画と軌道更新に基づく STS 制御の実現手法の提案と オープンアーキテクチャ型ロボットを用いた その有効性の実験的検証

青 柳 誠 司^{*1} 桑 原 一 義^{*1} 神 野 崇 治^{*2} 高 野 政 晴^{*1}

Proposal of Realizing Method of STS Control Based on Trajectory Planning and Trajectory Update and Experimental Verification of Its Effectiveness by Employing an Open Architecture Robot

Seiji Aoyagi^{*1}, Kazuyoshi Kuwahara^{*1}, Takaharu Jinno^{*2} and Masaharu Takano^{*1}

For the speedup of robot's task, authors have already proposed STS (State to State) control, which deals with a state (a pair of position and velocity) as a controlled variable and transfers a robot arm's tip from an initial state to a final state in a specified time. In order to realize STS control, the method based on the combination of the trajectory planning and trajectory update is newly proposed in this paper. Since the initial reference trajectory can be planned mathematically and the influence of the robot dynamics is eliminated by the on-line trajectory update, repeated trials for obtaining the series of command positions prior to the task are not needed. Moreover, this method can cope with the change of the final state during the task. A practical STS control system is constructed by employing an open architecture articulated robot and a pin insertion experiment to a moving hole on a belt conveyer is carried out. The accuracy of the final state in the case of applying the proposed method is fairly good compared with that applying the method without updating the trajectory. A pin insertion task under the clearance of ± 0.25 [mm] is succeeded with almost 100% probability by applying the proposed method.

Key Words: STS Control, Trajectory Planning, Trajectory Update, Open Architecture Articulated Robot, Pin Insertion Task

1. は じ め に

現在産業用ロボットは一般にティーチングプレイバック方式で使用されており, PTP (pose-to-pose) 制御や CP (continuous path) 制御によりあらかじめ教示された動作が実現されている. これら PTP, CP 制御は, 基本的にロボット先端の位置またはその順序集合である経路 (path) を実現目標とする制御方式であるため, 高速移動対象物を取り扱いにくい. 高速移動対象物を取り扱うにはロボット先端の位置だけでなく速度も考慮し, 位置・速度を状態 (state) として扱い, 制御開始時刻における初期状態から指定時刻における目標状態に移す STS (state-to-state) 制御方式 [1] が有効である (Table 1 参照). この方式を例えば工場におけるベルトコンベア上を順次流れるワークに対する作業に適用した場合, ワークとロボット先端の速度を指定時刻に

おいて一致させることが可能なため, ワークを停止させることなく, またロボットも加減速を必要とする停止を伴わずにワークを逐次処理することが可能となり, サイクルタイム, ロボットの消費エネルギーの観点から生産性の向上が期待できる. この作業に限らず現在多数の作業が PTP 制御で行われているが, 原理的にはこれらのすべてを対象物およびロボットを静止させなくてすむ STS 制御で代行することが可能であり, この意味からも STS 制御は生産分野で将来有望であると考えられる. またアミューズメントロボット, スポーツロボットなどの開発の分野

Table 1 Comparison of characteristics of PTP, CP, STS control concept

	Dealing with moving object	Possibility of speedup	Path accuracy	Controlled variable
PTP control	×	○	Ignored	Position
CP control	○	×	Important	Position
STS control	○	○	Ignored	State (position and velocity)

○:good ×:not good

原稿受付 2000 年 3 月 10 日

^{*1}関西大学工学部

^{*2}三菱重工業 (株)

^{*1}Faculty of Engineering, Kansai University

^{*2}Mitsubishi Heavy Industries, LTD.

においても、人間を模擬して高速な動作をロボットに行わせる場合にSTS制御の適用が有効である。例えばジャグリング動作(お手玉動作)[1]、野球のピッチング動作[2]などを模擬する場合、お手玉やボールのリリース時刻におけるロボット先端の位置と速度両者の正確な制御が要求される。

筆者らのうち高野[1]はすでに、STS制御を学習制御およびBang-Bang制御の手法を用いて実現し、スカラロボットによるジャグリングタスクに応用した。これはロボットの各関節に与えるトルクを、高速の運動制御に適しているBang-Bang制御の手法を参考にして n 個の方形波状の時間関数で与え(n はロボットの自由度)、学習制御の手法を適用してその切り替え時間と方形波の高さを繰り返し試行により獲得するものである。この方法では、切り替え時刻の増分と目標状態の増分との関係、および方形波の高さの増分と目標状態の増分との関係を表す二つのヤコビ行列をあらかじめ実験的に獲得して、これらを指定時刻に目標状態を実現するような初期軌道の学習と、作業中の目標状態の軽微な変化に対応するための軌道のオンライン修正に用いている。しかしながらこれらヤコビ行列の獲得には時間と労力が必要とされ、また作業の仕様が変更されて指定時刻や目標状態が大きく変更されるたびにこの獲得を行わなければならないという問題点があった。

本研究では以上を考慮し、STS制御の実現手法として、①デカルト空間において指定時刻における目標状態を実現するロボット先端位置の軌道(Trajectory. 時間をパラメータにとって表現した経路)を数学的に計画し、これを目標位置曲線とする、②減速機の固体摩擦、アームイナーシャ、トルクリミットなどの影響に起因する目標軌道からの実際のロボット先端の追従誤差を考慮し、一定時間間隔ごとにロボット先端の状態を観測し、この状態を初期状態とみなして目標位置曲線を再度計算して更新する、という手法を提唱する。本手法では、①により数学的に軌道が得られ、②により試行中に軌道を修正することによりロボットの追従誤差の影響を除去できるので、作業前に繰り返し試行を行うことなくSTS制御を実現することができる。また②において一定時間間隔ごとに目標状態の推定も行えば、外乱などによる作業中の目標状態の変化にも対応することができる。

上記のような軌道更新を行うためには、リアルタイムにロボット先端の状態の観測が行え、それに基づいてリアルタイムに先端位置の指令値の変更が行えるオープンアーキテクチャ型のロボットの採用が不可欠である。本論文ではこの型のロボット[3]を採用し、ベルトコンベア上を移動する穴へのピン挿入作業を行うシステムを構築し、提案する軌道計画と軌道更新に基づくSTS制御の実現法の基本的な有効性を実験的に検証した。

2. STS制御とPTP制御、CP制御との比較

本章では、PTP制御、CP制御、STS制御の三者を、定義、目的、実現手法の観点から比較することで、STS制御の位置づけを明確にするとともに、軌道更新の重要性について述べる。

2.1 定義および目的の観点からの比較

JIS[4]の定義によれば、PTP制御は「飛び飛びの点のポーズだけを指定した命令によって行われる制御」であり、CP制御

は「全軌道または全経路を指定した命令によって行われる制御」である。これらに対してSTS制御の定義は、「制御対象を初期状態から指定時刻における目標状態に移す制御」である。STS制御で目的とされるのは如何にして指定時刻における目標状態を精度良く実現するかであり、途中の経路精度は重要でなく無視してよい。この途中の経路は問わないという観点では、STS制御はPTP制御を拡張したものといえる。このようにPTP制御は静止点への位置決めを目的とする制御、CP制御は経路の実現を目的とする制御、STS制御は指定時刻での位置・速度の実現を目的とする制御というように、各制御は目的が異なっており、概念の異なる制御であるといえる(Table 1 参照)。

2.2 実現方法の観点からの比較

CP制御では、一般的には以下の①～③の方法でデカルト空間における目標位置の点列を得て、その各点に次々にロボット先端を位置決めするという手法が採られている。

- ①微小空間間隔ごとに経路上の位置を多数指示する。
- ②PTP制御と同様に経路上の主要な飛び飛びの点の位置(補間点)を指示し、各点の間を直線、円弧などで補間し、これを適当な空間間隔で点列に分解する[5]。
- ③いくつかの指定した補間点を通過する軌道(時間をパラメータとする経路)を多項式等で数学的に計画し、これをサンプリングタイムで離散化する[6][7]。

一方STS制御は指定時刻に目標位置・速度を実現する制御であり、その目的が果たせるならばどのような実現手法を採用しても良い。したがって1章で述べたようにトルク指令型のロボットを採用して学習制御およびBang-Bang制御の手法を用いて実現することも考えられるが、学習に時間と労力がかかるという問題点がある。一方CP制御の実現方法のうち上記③の手法では、指定時間において所望の位置と速度を実現するような数学的な軌道を多項式等で計画することが可能であり、学習プロセスを省略できる可能性がある。したがって本論文で提案するSTS制御の実現手法でもこれを軌道計画部分に採用する。

2.3 軌道レベルのオープンループ制御の適用限界

前節で述べたCP制御の実現方法のうち③の方法をSTS制御の実現方法として適用する場合、各関節の減速機や対偶の固体・流体摩擦、アームイナーシャ、各関節のアクチュエータのトルクリミット、制御系の遅れ時間などの影響を受けて、移動速度が比較的高速な場合にロボット先端が正確に目標位置点列に追従しないという問題が生じる。この問題に対して軌道を時間に対する3～5次曲線[7]、3次スプライン曲線[7][8]、クロソイド曲線[9]などで計画し、補間点を滑らかに結ぶ軌道を計画することによりロボットの軌道に対する追従誤差を低減することを意図した方法も提案されているものの、これのみで完全に追従誤差を除去することは困難である。またこの方法はいったん軌道がオフラインで計画された後は、作業遂行中はその軌道に従ってコントローラに指令値を与えるという軌道レベル[†]でのオープンループ制御に基づいている。産業用ロボットのコント

[†]各関節の角度(または角速度、トルク)を被制御量とする一般のロボットの制御に対して、本論文で提案する軌道を直接の被制御量とする制御を区別するため、前者を関節レベルでの制御、後者を軌道レベルでの制御と呼ぶことにする。

ローラでは目標関節角軌道の微分値等の既知情報を用いてフィードフォワード補償を行うことで、追従誤差を低減している場合も多いが[10]、この手法も目標位置点列を作業前に修正していることと等価であり、軌道レベルではオープンループ制御である。このような軌道レベルでのオープンループ制御の追従精度を改良するものとして、学習制御[11]を適用して、初期に計画された軌道を試行のたびに逐次修正していき最終的に理想的な軌道(指令点列)を獲得する手法も考えられる。しかしながら、この場合作業の仕様変更に伴い目標とする軌道が変更されるたびに、新たに軌道獲得のための試行を繰り返さなければならず、トータルな生産効率を考えた場合必ずしも有利ではないと考える。さらに移動対象物を扱う場合は、外乱などによりその速度が変化することが多く、オンラインで軌道を修正しなければならないが、学習制御も含めた軌道レベルでのオープンループ制御ではこれに対応することが困難である。

2.4 軌道レベルのフィードバックによる STS 制御の実現

本論文では STS 制御の実現手法として、作業遂行前に CP 制御の実現方法③に従って目標軌道を計算し、作業遂行中に一定時間間隔ごとに状態を観測し、この状態を新しい初期状態とみなして目標軌道を再度計算して更新する手法を提案する(詳細については次章で述べる)。これは定性的には作業遂行中に軌道の誤差(ある時刻における位置、速度の誤差)を検出し、それを軌道レベルで間欠的にフィードバックすることで指定時刻で目標状態を実現するように軌道を修正していることになる。本 STS 制御の実現手法の特長をまとめると以下ようになる。

- ①軌道更新により、高速動作時(指定時刻 t_f が短く、対象物の移動速度 v_f が大きい場合)においても、ロボットの追従誤差が目標状態の精度に与える影響を低減でき、指定時刻 t_f において状態(位置・速度)が高精度に実現できる。
- ②初期に計画された目標軌道と更新された目標軌道は大きく異なる可能性もあるが、STS 制御では途中の経路の精度を問わないため問題ない。
- ③作業中に外乱などにより目標状態が変化した場合にも、それを t_f で実現するような新しい軌道をオンラインで計算することで対応が可能である。
- ④アームイナーシャ、減速機の固体摩擦、トルクリミットなどの影響はすべて外乱とみなされフィードバック(軌道更新)により除去されることが期待できるので、正確なロボットのパラメータ推定やトルクリミットの考慮は不要となる。
- ⑤作業遂行前の目標軌道の計画は数学的に行えるので、学習制御のように目標軌道獲得のために試行を繰り返し行う必要がない。

軌道更新を行うためには、オープンアーキテクチャ型のロボットの採用が不可欠である。この型のロボットを用いて軌道更新を頻繁に行うことにより、CP 制御の目的である途中の経路精度の向上が高速動作時においても期待できるかもしれないが、本論文ではあくまで STS 制御の目的である指定時刻において所望の位置・速度を実現することのみに焦点を絞るので、途中の経路精度は議論しないことにする。すなわち STS 制御と CP 制御は目的が異なる制御であり、本論文は STS 制御(の目的)を、従来 CP 制御の実現方法として採用されている軌道計画と、

オープンアーキテクチャ型ロボットにより可能となる軌道更新とを組み合わせることで実現する方法を提案するものである。

3. 軌道計画と軌道更新に基づく STS 制御の基本アルゴリズム

3.1 5 次多項式による目標位置曲線の計算

ロボットベース座標系を基準としたロボット先端の座標を (x, y, z) とし、本論文では以下一貫して x 座標に関して議論を行う(y, z 座標についても同様な議論が成立する)。ロボット先端を時刻 $t = 0$ における初期状態 (x_0, v_0) から指定時刻 $t = t_f$ において目標状態 (x_f, v_f) に移す STS 制御を行うとする。これら初期状態と目標状態を満たす軌道の計画法としては、2.2 節で述べた CP 制御の実現方法のうち③の方法を採用する。初期状態と目標状態を満たす多項式のうち、最も次数の低いものは 3 次多項式である。

現実の産業界でのアプリケーションにおいてはエンドエフェクタとワークとの間の接触を伴う組立作業など、 t_f から短い期間ロボット先端と移動する対象物との相対位置を一致させなければいけない場合が多い。例えばベルトコンベア上を移動するワークに設けられた穴に対し、 t_f でピンを挿入する作業をロボットで行う際、ピンを上方から放して落下させる場合は t_f の瞬間のみロボット先端とワークの位置と速度を一致させればよいが、確実な挿入を目的としてピンを穴に挿入してから放す場合、 t_f 以降ピンと穴が勘合している短い期間ロボット先端とワークの位置と速度を一致させなければいけない。この場合、ベルトコンベアの搬送速度が一定であると仮定して、 t_f において x_f, v_f のみでなく、加速度 α_f を 0 に指定して t_f において速度を滑らかに接続する(加速度を連続にする)必要がある。なお t_f 以降、対象物が直線軌道以外の三次元的な軌道上を移動する場合は、 α_f をこの軌道の t_f における加速度に指定すればよい[12]。

さらに次節で述べるようにロボット先端を滑らかに動作させて追従誤差を低減させるため、本研究では各更新時刻 t_f で目標軌道の加速度が連続になるように加速度を指定して新しい軌道を計算する。これを考慮して $t = 0$ においても同様に加速度 α_f を 0 に指定することにする。結局 $t = 0$ および $t = t_f$ で加速度に関する条件が二つ付加され、 $x_0, v_0, \alpha_0, x_f, v_f, \alpha_f$ の条件を満たす多項式のうち最も次数の低い 5 次多項式を初期の目標位置曲線 $x = f(t)$ として採用する。この場合 $f(t)$ は以下の式により計画できる。

$$x = f(t) = at^5 + bt^4 + ct^3 + dt^2 + et + f \quad (1)$$

$$a = \frac{1}{t_f^5} \left\{ 6(x_f - x_0) - 3(2v_f + v_0)t_f + \frac{1}{2}(\alpha_f - \alpha_0)t_f^2 \right\} \quad (2)$$

$$b = \frac{1}{t_f^4} \left\{ -15(x_f - x_0) + (7v_f + 8v_0)t_f - \left(\alpha_f - \frac{3}{2}\alpha_0 \right) t_f^2 \right\} \quad (3)$$

$$c = \frac{1}{t_f^3} \left\{ 10(x_f - x_0) - (4v_f + 6v_0)t_f + \frac{1}{2}(\alpha_f - 3\alpha_0)t_f^2 \right\} \quad (4)$$

$$d = \alpha_0/2 \quad (5)$$

$$e = v_0 \quad (6)$$

Fig. 1 Concept of position reference curve with update

$$f = x_0 \quad (7)$$

3.2 目標位置曲線の更新と更新時間間隔に対する検討

本 STS 制御の実現手法では、目標位置曲線からの実際のロボット先端位置の追従誤差を考慮し、Fig. 1 に示すように一定時間間隔ごとに状態を観測し、この状態を新しい初期状態とみなし、これと目標状態とともに満足するような目標位置曲線を再度計算して更新する。その際前節で述べたように $\alpha_f = 0$ に指定し、更新時刻 t_r における加速度 α_r を更新前の目標位置曲線 t_r における 2 階微分値に指定する条件を付加する。後者に関しては本来なら t_r における実際のロボットの加速度を観察してその値を用いるべきであるが、観測ノイズの影響からこれが困難であるためこのような措置を講じた。 t_r で観測された状態を (x_r, v_r) とすると、新しい目標位置曲線は式 (1) において t を $(t - t_r)$ に置き換え、式 (2)~(7) において t_f を $(t_f - t_r)$ に、 (x_0, v_0, α_0) を (x_r, v_r, α_r) に置き換えたものとなる。

更新時間間隔が短いほど軌道レベルでの間欠的なフィードバックが頻繁に行われるので好ましいように思えるが、極端に更新時間間隔を短くすることは以下の観点から好ましくない。

- ① 頻繁な目標位置曲線の更新は計算量を増加させ、制御系の CPU の負担を増加させる。
- ② 指定時刻 t_f の直前の更新時刻 t_{f-1} において目標状態 (x_f, v_f) からの誤差が大きい場合、1 更新時間間隔でこれを 0 にするために目標位置曲線は大きな加減速を伴う。この度合いは更新時間間隔が短いほど大きくなり、ロボット先端の不要な振動を引き起こすおそれがある。

これらを踏まえて、最適な更新時間間隔は実験的に設定する必要がある、これについては 5 章で検討する。

3.3 指定時刻 t_f 以降の制御法についての検討

t_f から短い期間ロボット先端と移動する対象物との相対位置を一致させなければいけない場合、 t_f 以降は対象物の移動軌道を目標位置点列に分解して追従制御を行えばよい [12]。STS 制御を適用した場合、① t_f で位置と速度が正確に実現されているため t_f 以降の追従誤差が小さい、② このため追従期間を最小限に設定することができ作業の高速化が期待できる、という利点がある。 t_f 以降の目標位置点列は、空間点列の教示、軌道計画、学習による繰返し試行等で与えることになるが、上記①②の利点から t_f 以降に接続する軌道が三次元的な複雑なものであっても教示点数や学習回数は少なくすむことが期待され、

Fig. 2 Composition of experimental system for realizing STS control

この観点からも STS 制御は有効であると考えられる。

4. オープンアーキテクチャ型の多関節型ロボットとベルトコンベアを用いた STS 制御用実験システムの構築

4.1 オープンアーキテクチャ型ロボットの採用

オープンアーキテクチャ型の多関節型ロボットとして (株) 三菱重工製 PA-10 [13] を採用して STS 制御系を構築し、ベルトコンベア上を移動するワークに設けられた穴にピンを挿入する実験を行った。 Fig. 2 に実験システムの構成を示す。 DSP により逆変換、サーボアンプへの指令電圧の計算等を 10 [ms] 周期で行う運動制御ボードがメーカーにより供給されており、本ロボットのコントローラはこのボードがユーザにより用意される市販の DOS/V 仕様パーソナルコンピュータの拡張スロットに挿入される構成をとる。 このボードに対して先端位置制御、各軸位置制御などを行わせるコマンド群の発行を行う C 言語の関数ライブラリがメーカーにより供給されている。

穴の直径は 9.5 [mm]、ピンの直径は 9 [mm] であり、 t_f におけるロボット先端の位置決めにおいては y 方向の誤差が 0 であると仮定して x 方向で ± 0.25 [mm] の誤差が許容される。

4.2 ベルトコンベアの速度検出と蛇行防止

Fig. 2 に示すようにベルトコンベアの駆動軸にロータリエンコーダを取り付けた。 ベルトの進行距離 (ただし次節で述べるロボットベース座標系における距離) とエンコーダパルスのカウント数との関係はあらかじめ実験的に求められているので、一定時間間隔ごとのカウンタ値の増分を検出すれば、それよりベルトコンベアの速度が算出できる。

ベルトコンベアのベルトには蛇行する性質があるため、 Fig. 3 に示すような蛇行防止機構を開発した。 ワークはプラットホー

Fig. 3 Mechanism of snaking prevention

ムに設置される。プラットフォームはベルトコンベアの側面に取り付けられたアルミ製のアングル材に沿って直動するため、ワークの横方向（進行方向と垂直方向）の位置は一定に保たれる。プラットフォームに取り付けられた4個の車輪がベルト表面との間に生じる横方向の摩擦力により回転するので、蛇行に伴うプラットフォーム本体に負荷される横方向の力は低減される。

4.3 ロボットベース座標系における直線軌道の教示

本実験では目標状態の獲得のため、ロボットベース座標系における穴の移動軌道（直線軌道）をあらかじめ把握しておく必要がある。このためワークをベルトコンベア上で準静的に移動させて、Fig. 2に示す通過センサが反応した時点でワークを静止させ、ティーチングボックスを用いてピンを穴に挿入し、そのときのロボット先端の位置を教示する。次に指定時刻 t_f とベルトコンベアの指定速度 v_b から計算されるピン挿入位置近傍にワークを静止させ、同様にしてロボット先端の位置を教示する。この教示された2点からコントローラが想定するロボットベース座標系における直線軌道が計算できる。

ロボットの機構の寸法誤差に基づき、コントローラが想定するロボットベース座標系でのこの2点間の距離 d_R は、実際の距離 d_M とは正確には一致しない。前節において述べたベルトの進行距離とエンコーダパルスのカウント数との関係を求める際には、ベルトコンベアの駆動軸の一回転に対するコンベアの移動距離を実測し、それに補正係数 d_R/d_M を乗じた値を一回転あたりのベルトの進行距離としている。

4.4 更新時間ごとの目標状態の修正と軌道更新

挿入実験の直前に5秒間エンコーダのパルスをカウントしてそれよりベルトコンベアの平均的な速度 v_b を求める。前節で得られた直線軌道と x 軸との関係から、 v_b の x 軸方向への成分を求めてこれを v_{bx} とする。通過センサをワークが通過した時刻を $t = 0$ とし、指定時刻 t_f における目標状態の初期値 (x_f, v_f) を $x_f = v_{bx} \times t_f$, $v_f = v_{bx}$ により求める。

本ベルトコンベアはオープンループで制御されるインダクションモータにより駆動されているため、負荷変動などに起因してベルトの速度は変動し、それに伴い t_f における目標状態も変動する。このため以下の方法で各更新時刻において目標状態の修正およびそれに基づいた軌道計算を行うことにする。

- ① k 回目の軌道の更新時刻を $t_{r(k)}$ とし、エンコーダの現在値からその時刻における穴の位置 $x_{b(k)}$ を求める。
- ② $t_{r(k-1)}$ から $t_{r(k)}$ までのエンコーダのカウント値の増分から計算されるベルトコンベアの速度の x 方向成分 $v_{bx(k)}$ を用いて、以下の式により目標状態を修正する。

Fig. 4 Compensation for maintaining continuity of series of positions at update time

$$x_f = x_{b(k)} + v_{bx(k)} \times (t_f - t_{r(k)}) \quad (8)$$

$$v_f = v_{bx(k)} \quad (9)$$

- ③ この目標状態と、ロボット先端の現在の状態から3章で述べた方法により目標位置曲線を再度計算して更新する。

4.5 ロボット先端位置制御系の遅れ時間の補償

本研究で提唱する STS 制御の実現手法では軌道レベルでの間欠的なフィードバックが行われるものの、目標状態を高精度に実現するためにはロボット先端の位置制御における速応性になるべく高めておくことが望まれる。そこで本研究では運動制御ボードが実現する先端位置制御系の伝達関数（入力：指令値、出力：実現値）を実験的に時定数 $T_m = 0.15$ [s] の1次遅れ系として求め、この伝達関数から遅れ時間を補償する入力を求めてフィードフォワード補償を行っている。具体的には、目標位置曲線 $f(t)$ から以下の式によりコントローラに与える指令値 $h(t)$ を計算する。

$$h(t) = f(t) + T_m f'(t) \quad (10)$$

4.6 更新時刻における指令値の連続性を保つための補償

軌道更新において、新しい目標位置曲線は観測された状態を新たな初期状態として作成されるので、Fig. 4に示すように更新時刻において目標位置曲線とロボット先端位置との偏差は突然0にリセットされ、式(10)で示したコントローラへの位置指令値 $h(t)$ が不連続になる。これによりロボット先端の動作が不安定になり、大きな振動が生じるおそれがある（紙面の都合上データは省略するが、実際にロボットの動作が不安定になることを実験的に確認した）。このため本研究ではFig. 4に示すように、更新時刻において指令値およびその1, 2階微分値が連続になるように補償関数 $g(t)$ を計算し、これを式(10)の $h(t)$ に足し込んでいる。結局 $p(t) = h(t) + g(t)$ をサンプリングタイムで離散化してコントローラに与えることになる。以上

(a) Position error transition when $\Delta t_r = 50$ [ms] (b) Velocity data when $\Delta t_r = 50$ [ms] (c) Position error transition when $\Delta t_r = 200$ [ms] (d) Velocity data when $\Delta t_r = 200$ [ms] (e) Position error transition when $\Delta t_r = 400$ [ms] (f) Velocity data when $\Delta t_r = 400$ [ms]**Fig. 7** Experimental results by changing update interval Δt_r ($t_f = 4$ [s], $v_b = 115$ [mm/s])

Fig. 6 (e) より軌道更新を行わない場合 4 [mm] 程度の誤差で推移するのに対して, Fig. 6 (f) の拡大図より軌道更新を行う場合 0.2 [mm] 以内に誤差が保たれている. 速度に関しては, Fig. 6 (c) (d) の拡大図に示すように軌道更新を行わない場合 4 [mm/s] 程度の誤差が持続するのに対して, 軌道更新を行う場合は最大 2 [mm/s] 程度のスパイク状の誤差が生じるものの, それ以外では 1 [mm/s] 以内に誤差が保たれている. Fig. 6 (d) の拡大図に示すように, STS 制御では速度誤差を 1 更新時間間隔で 0 にするために t_{f-1} から t_f の間の目標速度の増減変化が大きく, それに伴い目標加速度の変化も大きい. さらに本実験では t_f において速度一定の軌道に接続するため, t_f 前後での目標加速度の変化も大きい. このように軌道更新を行う場合は t_{f-1} 以降大きな加減速が要求されるため, t_f 以降にロボット先端が振動してスパイク状の速度誤差が生じるものと考えられる. しかしながらこの速度誤差が位置精度に与える影響は Fig. 6 (f) の拡大図に示すようにピン挿入のための許容誤差の範囲内である.

5.2 目標位置曲線の更新時間間隔に対する実験的検討

本節では軌道の更新時間間隔 Δt_r を変化させた場合に目標状態がどの程度の精度で達成できるかを比較した. Fig. 7 に Δt_r が 50, 200, 400 [ms] の場合の t_f 付近における位置, 速度に関する結果の一例をそれぞれ示す. なお Δt_r が 100 [ms] の場合のデータは Fig. 6 (d) (f) の拡大図を参照することにする. また各 Δt_r について 5 回ずつ試行を行い, t_f における実現状態の誤差について, その絶対値の平均値および最大値を求めた結果を Table 3 に示す.

Fig. 7 (a) (b) より Δt_r が 50 [ms] と短い場合, Fig. 6 (d) (f) の拡大図で示す Δt_r が 100 [ms] の場合と同程度の位置, 速度誤差が t_f 以降生じていることが分かる. これは前節で述べた t_{f-1} 以降の加減速に起因していると考えられる. このことと Δt_r が短いほど軌道更新に伴う計算量が増加することを考慮すれば, Δt_r は 50 [ms] よりも 100 [ms] に設定する方が本実験条件下では好ましい.

一方 Δt_r が 200 [ms], 400 [ms] と長くなると, t_{f-1} 以降の加速度の増減変化が小さくなるため, Table 3, Fig. 7 (d) (f) に示すように Δt_r が 100 [ms] の場合に比較して t_f および t_f 以降での速度誤差が減少する. しかしながら軌道レベルでの間欠的なフィードバックの頻度が減少するため, Table 3, Fig. 7 (c)

(e) に示すように Δt_r が 100 [ms] の場合に比較して t_f および t_f 以降での位置誤差が増大し, 場合によっては 0.3 [mm] 以上となる. t_f 以降の速度誤差は前節で述べたように位置精度にそれほど大きな影響を与えないのに対して, t_f 以降の位置誤差は直接ピン挿入作業の成否に影響すること, またピン挿入が可能となるためのロボット先端位置の許容誤差が 0.25 [mm] であることを考慮すると, Δt_r を 200 [ms] 以上に設定することは本実験条件下では好ましくない. 以上のことより結局, 本実験条件下で選択した Δt_r の中でも 100 [ms] が最適であることが分かる. またこれより一般的に, 更新時間間隔を先端位置制御系のサンプリングタイム (本実験システムでは 10 [ms]) の 10 倍程度に設定することが好ましいことが実験的に推察される.

5.3 ベルトコンベア速度 v_b , 指定時刻 t_f を変化した場合のピン挿入確率

本節では, ベルトコンベアの速度 v_b を 115, 175 [mm/s] の二通りに設定し, それぞれに対して指定時刻 t_f を 2, 3, 4 [s] の三通りに変化させ, 各条件において 10 回試行を行った場合のピン挿入確率を求めた. Fig. 8 にロボットの経路の上面図を示す. z 方向の軌道については, ピンの底面が t_f において穴の上方 2 [mm] に位置決めされてから, 0.1 [s] で 10 [mm] 下降し, その変位を 0.1 [s] の間保ち, その後 0.1 [s] で元に戻るよう計画した.

Table 4 に結果を示す. これよりベルトコンベアの速度が速く指定時刻も短いというロボットにとっては最も厳しい $v_b = 175$ [mm/s], $t_f = 2$ [s] の条件において 10 回の試行のうち 1 回ピン挿入に失敗した. この失敗の理由として Fig. 8 (b)

Fig. 8 Path of robot arm's tip in pin insertion experiments (top view)

Table 3 Results of final state at t_f by changing update interval Δt_r

Update Interval Δt_r (ms)	Absolute position error (mm)		Absolute velocity error (mm/s)	
	Average	Max.	Average	Max.
50	0.1	0.1	1.0	1.7
100	0.0	0.1	1.8	2.4
200	0.3	0.4	1.0	1.5
400	0.2	0.3	0.4	0.9

- ※ 5 trials are carried out for each update interval.
- ※ Experimental conditions are same as Fig. 7.
- ※ Values of 100ms are same as Table 2.

Table 4 Probability of successful pin insertion under various conditions

$v_b \backslash t_f$	2.0s	3.0s	4.0s
115mm/s	10/10	10/10	10/10
175mm/s	9/10	10/10	10/10

Unit: (Successful trial number)/(All trial number)

 t_f : Specified time v_b : Belt's velocity **Fig. 9** Consecutive photographs of pin insertion task

に示すようにこの条件下では経路の x 座標の増減が他の条件に比べて大きく、ロボット先端に大きな加速度の変化が要求され追従誤差が生じやすいこと、指定時刻が短いため軌道更新により軌道を修正する回数が他の条件に比べて少ないこと、などが考えられる。しかしながらこの他の条件ではすべての試行においてピン挿入に成功し、挿入確率は 100%であった。なおピン挿入の様子の一例を **Fig. 9** の連続写真に示す。この写真の例では挿入状態を明示するためハンドを開いておらず、ピンは最終的に穴の上方まで引き上げられている。

6. 結 言

高速移動対象物を取り扱う作業に対して有効な制御法として、ロボット先端の位置・速度を状態として扱い、制御開始時刻における初期状態から指定時刻における目標状態に移す STS 制御がすでに提案されている。本論文では STS 制御の実現手法として、①デカルト空間において指定時刻における目標状態を実現するロボット先端位置の軌道を数学的に計画し、これを目標位置曲線とする、②一定時間間隔ごとにロボット先端の状態の観測および目標状態の推定を行い、目標位置曲線を再度計算して更新する、という手法を提唱した。本手法では、①により数学的に軌道が得られ、②により試行中に軌道を修正することによりロボットの追従誤差の影響が除去できるので、作業前に繰り返し試行を必要としない。また②により外乱による目標状態の変化にも対応できる。既存の CP 制御、フィードバック制御と

フィードフォワード制御との組み合わせ、学習制御などは軌道レベルでオープンループの制御であるのに対して、本手法は軌道レベルで間欠的なフィードバックを行い、これによる情報をもとに目標軌道をオンラインで修正するという特徴を有する。

リアルタイムに先端位置を指令できるオープンアーキテクチャ型の多関節型ロボットを用いて STS 制御系を構築し、ベルトコンベア上を移動するワークに設けられた穴にクリアランス ± 0.25 [mm] の条件でピンを挿入する実験を行った。これより軌道更新を行わない場合に比べてそれを行う場合には、指定時刻およびそれ以降のピン挿入に必要な短い期間において高い精度で状態（位置・速度）を実現できることが判明した。またベルトの速度 175 [mm/s]、指定時刻 2 [s] の条件で 10 回中 1 回ピンの挿入に失敗したものの、その他の条件下では 10 回のすべての試行においてピンの挿入に成功し、高い挿入確率を得た。

本研究における実験は、ロボットは特異姿勢をとらずロボット先端の姿勢も常に一定値を保ち、ワークも直線軌道上を移動するという緩和された条件の下で行われている。しかしながらスカラ型ロボットなどの多関節型に比べて軸数が少ないロボットでは精度、速応性の観点から先端位置の制御が比較的容易であることなどを考慮すれば、本実験で示された軌道計画と軌道更新に基づく STS 制御の実現手法の基本的な有効性は、他のオープンアーキテクチャ型のロボットを用いて高速移動対象物を扱う作業を行う場合においても、普遍性を有するものと考えられる。

参 考 文 献

- [1] 朴 勇進, 高野政晴, 佐々木健: “STS 学習制御法とお手玉ロボットへの応用”, 精密工学会誌, vol.60, no.5, pp.667-672, 1994.
- [2] 山川博司, 高野政晴, 佐々木健, 関 啓明: “人間型ピッチングロボットに関する研究”, 1996 年度精密工学会秋季大会学術講演会講演論文集, pp.27-28, 1996.
- [3] 大西典司, 大西 誠: “可搬式汎用知能アーム登場! —オープンロボットの提案—”, 日本ロボット学会誌, vol.12, no.8, pp.1137-1142, 1994.
- [4] JIS B 0134 産業用ロボット—用語, 日本規格協会, pp.1-28, 1994.
- [5] 有本 卓: ロボットの力学と制御. pp.100-102, 朝倉書店, 1990.
- [6] J.J. Craig: Introduction to Robotics: Mechanics and Control (三浦宏文・下山 勲訳). pp.189-216, 共立出版, 1991.
- [7] K.S. Fu, R.C. Gonzalez and C.S.G. Lee, Robotics—Control, Sensing, Vision, and Intelligence—, (本多庸悟, 遠山茂樹, 古屋信幸, 金子俊一訳). pp.157-213, 日刊工業新聞社, 1989.
- [8] H. Makino, N. Furuya and M. Iwatsuki: “Performance Tests for CP Motion with the SCARA Robot,” Proc. of 15th Int. Symp. on Industrial Robots, pp.1011-1020, 1985.
- [9] 明 愛国, 伊藤忠司, 牧野 洋: “高速高精度平面軌跡制御用小型ロボットの開発”, 精密工学会誌, vol.59, no.11, pp.1865-1870, 1993.
- [10] 石川 均, 野瀬松男, 鯨島泰郎: “ロボットコントローラの運動制御技術”, 日本ロボット学会誌, vol.14, no.6, pp.776-779, 1996.
- [11] 有本 卓: “学習制御の理論”, 日本機械学会誌, vol.93, no.856, pp.180-186, 1990.
- [12] S. Aoyagi, T. Jinno and M. Takano: “Speedup of Insertion Task of a SCARA Robot by STS Control Using a Jig,” Proc. of 30th Int. Symp. on Robotics, pp.517-524, 1999.
- [13] 下山公宏, 大西 誠: “PC コントローラを用いたマニピュレーションシステム”, 日本ロボット学会誌, vol.16, no.8, pp.1050-1052, 1994.

青柳誠司 (Seiji Aoyagi)

1962 年 6 月 14 日生。1988 年 3 月東京大学大学院工学系研究科精密機械工学専攻修士課程修了。同年 4 月金沢大学工学部機械システム工学科助手。1995 年 11 月同助教授。1996 年 4 月関西大学工学部管理工学科助教授。現在に至る。博士(工学)。ロボット用センサ・アクチュエータ、ロボット制御、マイクログマシン等の研究に従事。精密工学会、日本機械学会、計測自動制御学会、電気学会等の会員。
(日本ロボット学会正会員)

神野崇治 (Takaharu Jinno)

1974 年 2 月 8 日生。1999 年 3 月関西大学大学院工学研究科管理工学専攻修士課程修了。同年 4 月三菱重工業(株)入社。在学中 STS 制御によるロボット運動の高速化に関する研究に従事。

桑原一義 (Kazuyoshi Kuwahara)

1974 年 10 月 30 日生。1998 年 3 月関西大学工学部管理工学科卒業。同年 4 月同大学大学院修士課程入学。現在に至る。STS 制御によるロボット運動の高速化に関する研究に従事。

高野政晴 (Masaharu Takano)

1936 年 11 月 25 日生。1963 年 3 月東京大学大学院工学系研究科精密機械工学専攻修士課程修了。同年 4 月同大学工学部精密機械工学科助手。1965 年 10 月同講師。1968 年 4 月同助教授。1980 年 4 月同教授。1997 年 4 月関西大学工学部管理工学科教授。工学博士。機械運動の高速化、ロボットの機構・運動制御、移動ロボットのナビゲーション、ロボットハンド、福祉ロボットなどの研究に従事。精密工学会、日本機械学会、日本騒音制御工学会等の会員。
(日本ロボット学会正会員)