接触遷移グラフを用いたロボットアームの搬送作業の タスクスケジューリング

Task Scheduling for Carrying Task of Robotic Arm Using Contact Transition Graph

○非 石井 雅人, 正 田崎 勇一, 正 鈴木 達也 (名古屋大)

Masato ISHII, m_ishii@nuem.nagoya-u.ac.jp

Yuichi TAZAKI and Tatsuya SUZUKI (Nagoya University)

This paper presents a method for scheduling carrying tasks of a robotic arm using a contact transition graph. This method generates a task sequence to achieve the objective work-environmental state focused on the contact state of an environment. Objects in contact state are moved together as an object is carried. A task sequence of a robotic arm is determined by solving a shortest path problem of the state transition system of the contact states of work environment. The motion trajectory of a robotic arm is generated by planning a trajectory based on the task sequence.

Key Words : trajectory generation, state transition system, multi-joint robot

1. は じ め に

近年, 家庭やオフィスなどで人間を補佐するサービスロボットに対する期待が高まっている. しかし, このようなロボットが対象とする作業環境は非常に幅広く, 事前に動作を決定することが難しい. このため, ロボットが自律的に行動を決定するための様々な研究がなされている.

Nau らは, 目的とする作業環境を達成するためのロボットの作業手順を記号の手順により生成する方法を提案した^[1]が, 複雑な機構を持つ多自由度ロボットの動作を対象とする場合, これらの動作を記号的に記述することは難しい. 一方で, Keith らの研究^[2]では, ロボットの行う操作や操作間の開始, 終了時間の前後関係が与えられているもとで, 目的の動作を達成するための作業のスケジューリングを行った. しかし, 自律的な行動の実現のためには, 作業目的を達成するための一連の操作も計画されることが望ましい.

本研究では, 搬送作業において物体を同時に搬送できる状態に着目した作業環境モデルを利用することで, 目標とする作業環境を達成するためのロボットの作業手順を決定する手法を提案し, それを田崎らによる軌道計画手法^[3]と組み合わせることにより, ロボットの動作軌道を生成する方法について述べる.

2. 作業環境の接触遷移グラフによる表現

2.1 接触状態ツリーの定義 本研究では作業環境の状態を, 作業環境中の物体と物体間の接触状態により表現する. 接触状態とは, 物体 A の移動に伴い, 他の物体 B が同時に搬送される関係にある状態を指す. 物体をノード, 物体間の接触状態をリンクにより表した有向グラフを接触状態ツリーと呼ぶ.

本研究では接触状態として, 物体が積み重なっている

状態, またはロボットハンドが物体を把持している状態の二つの状態を想定する. Fig.1(a) に示される作業環境において, 物体 n_0 は食器棚, n_1 と n_2 は皿, n_3 はコップ, n_4 はロボットアーム, n_5 はテーブルを表している. 食器棚 n_0 には皿 n_1 , n_2 が積み重ねられて, またコップ n_3 が置かれており, n_0 は n_1 , n_3 と, n_1 は n_2 と接触状態にある. 物体の集合を $\mathcal{N}$ とする. また, 物体 $n_a \in \mathcal{N}$ が物体 $n_b \in \mathcal{N}$ と接触状態にあるとき, $(n_a, n_b) \in \mathcal{L}$ と書く. 以上より, 接触状態ツリー $\mathcal{T} = (\mathcal{N}, \mathcal{L})$ を定義する.

2.2 接触状態ツリーの状態遷移 ロボットが作業環境に対し操作を行うことにより, 作業環境の接触状態が変化する. とり得る接触状態ツリーの集合を $\mathcal{V}$, 後述する事前条件, 事後条件を満たす $T_i \in \mathcal{T}$, $T_j \in \mathcal{T}$ の対の集合 $\mathcal{E}$ により, 接触遷移グラフ $G = (\mathcal{V}, \mathcal{E})$ を定義する. その概要を Fig.2 に示す.

ロボットの操作と接触グラフの遷移の因果関係は, 遷移するために接触グラフが満たさなければならない条件を規定する事前条件, 遷移による接触状態の変化を規定する事後条件の二つから構成される. 本研究ではロボットが行うことのできる操作として操作 PICK と操作 PLACE の二つを考える. PICK(n_a) は物体 $n_a \in \mathcal{N}$ をハンドで把持する操作を表す. また, PLACE(n_a, n_b) はハンドで保持している物体 $n_a \in \mathcal{N}$ を物体 $n_b \in \mathcal{N}$ の上に置く操作を表す. 以下に, それぞれの操作における事前条件と事後条件を示す. ただし, ロボットハンドを表すノードを $n_{hand} \in \mathcal{N}$ と表記する.

(a) 操作 PICK(n_a)

事前条件

ノード n_{hand} が子ノードをもたない.

事後条件

$(n_b, n_a) \in \mathcal{L}$ を削除し, $(n_{hand}, n_a) \in \mathcal{L}$ を作成する.

(b) 操作 PLACE(n_a, n_b)

事前条件

(a) Initial state

(b) PICK(n_1)

(c) PLACE(n_1, n_5)

Fig. 1 Work environment and contact state tree

Fig. 2 Contact transition Graph

- ノード n_{hand} がノード n_a を子ノードとして持つ.
- ノード n_b が子ノードを持たない.
- ノード n_a はノード n_b を子孫ノードとして持たない.

事後条件

$(n_{hand}, n_a) \in \mathcal{L}$ を削除し, $(n_b, n_a) \in \mathcal{L}$ を作成する.

Fig.1(a) の作業環境に対し, 操作 PICK(n_1) を実行した場合を Fig.1(b) に, さらに操作 PLACE(n_1, n_5) を実行した場合を Fig.1(c) に示す.

3. 接触遷移グラフの状態空間の探索

作業環境の初期状態から目標状態を達成するための操作の組の決定は, 接触遷移グラフにおける二頂点間の

Fig. 3 Positions of the objects in Fig.1(a)

Table 1 Variables and Sets of Dijkstra's Algorithm

Symbols	description
$d_v \in \mathbb{R}$	the shortest distance to vertex v
$p_v \in \mathcal{N}$	the previous vertex of vertex v
$\mathcal{H} \subseteq \mathcal{V}$	Set of visited vertexes
$\mathcal{A} \subseteq \mathcal{V}$	Set of neighbor vertexes of set H

経路を決定する問題として扱われる. 以降, ロボットの操作の組を操作シーケンスと呼ぶ. 操作シーケンスの評価として総作業時間を用い, この最小化を目指す.

ここで, 作業時間はロボットハンドと移動先位置間の距離に比例すると仮定し, これを遷移重みとする. 遷移重みの評価のために, 作業環境に対し物体の空間的位置を近似的に定める. すなわち, 接触関係をもつノード間について, 子ノードはその親ノードと同じ位置に存在すると仮定する. 物体 $n_a \in \mathcal{N}$ の位置を $p_{n_a} \in \mathbb{R}^3$ と表記すると, Fig.3 において, $p_{n_0} = p_{n_1} = p_{n_2} = p_{n_3}$ という関係が成り立つ.

以上により定められた物体の位置により, 接触グラフ T においてロボットを物体 n_{dest} に移動する操作の遷移重み $d_{n_{dest}} \in \mathbb{R}$ を

$$d_{n_{dest}} = \|p_{n_{dest}} - p_{hand}\| \quad (1)$$

により求める. ただし, 物体 n_{dest} は作業におけるロボットの移動先物体を表し, 作業 PICK(n_a) では p_{n_a} , 作業 PLACE(n_a, n_b) では p_{n_b} を意味する. 作業を行い接触グラフが遷移すると, 遷移後の接触グラフの接触状態に従い, 物体の位置が移動後の親ノードの位置に更新される.

以上により定義された状態遷移系における最短経路問題をダイクストラ法を用いて解決する. ダイクストラ法は, 辺の重みが非負数のグラフに対し, 単一始点からの最短距離を求めるアルゴリズムである. ダイクストラ法のアルゴリズムを Algorithm 1 に, またその変数, 集合を Table 1 示す. ただし, Algorithm 1 において, $outedges(v_j)$ は頂点 $v_j \in \mathcal{V}$ から出る辺の集合を返す関数, $head(e)$ は辺 $e \in \mathcal{E}$ の遷移先の頂点を返す関数, $weight(v_{j_1}, v_{j_2})$ は頂点 $v_{j_1} \in \mathcal{V}$ から頂点 $v_{j_2} \in \mathcal{V}$ の間の辺のもつ重みを返す関数である.

Algorithm 1 を実行することにより, 頂点 $v_h \in \mathcal{H}$ について, 始点 v_{ini} から頂点 v_h に達するための状態遷移の系列とそのときの最短距離 d_{v_h} が得られる. これにより, 状態遷移に対応するロボットの作業の組として, 作業シーケンスが得られる.

4. 軌道の決定

接触遷移グラフの探索により求められた作業シーケンスに対し, 田崎らによる軌道計画手法を適用し, ロボッ

Algorithm 1 Dijkstra's Algorithm**loop** $u := \arg \min_{v_a \in A} \text{distance}(v_a)$ add vertex u into set H remove vertex u from A **if** $u = g$ **then**

break

end if**for all** $e \in \text{outedges}(u)$ **do** $t := \text{head}(e)$ **if** $d_t > d_u + \text{weight}(u, t)$ **then** $d_t \leftarrow d_u + \text{weight}(u, t)$ $p_t := u$ **if** $t \notin A$ **then**add vertex t into set A **end if****end if****end for****end loop**

ト動作の軌道を得る。この軌道計画手法では、ロボットの動作仕様を規定するためにリーチングタスクを用いる。本節では、この軌道計画手法の概要と、操作シーケンスからリーチングタスクを生成する方法について述べる。軌道計画手法について詳しくは文献^[3]を参照されたい。

4.1 軌道計画手法の概要 この手法において、ロボットはマルチボディシステムとして表現されている。マルチボディシステム上でロボットは剛体と、それらを接続する関節より構成される。各剛体の運動に関しては並進・回転運動方程式にもとづく拘束が課され、関節による機構条件により相対運動が拘束される。ここで定義されている軌道は、離散的な制御点とその間を補完する区分的二次曲線によって表現され、それぞれの制御点において連続な軌道となるための拘束条件が課される。以上の制約はまとめて物理制約と呼ばれており、軌道計画を行う時間区間全域で有効となる。

ロボットアームが行うべき作業を記述するものとしてリーチングタスクが設定されている。リーチングタスクとは二つの剛体の位置を所定の時区間において一致させるタスクである。このタスクにより時刻 t_k での剛体 n の位置を $p_{n,k}$ 、タスク r の開始時刻を t_r 、終了時刻を $\bar{t}_r$ とすると、剛体 n_a , n_b の軌道に課す拘束は次のように書ける。

$$p_{n_a,k} = p_{n_b,k} \quad \forall k \text{ s.t. } t_k \in [t_r, \bar{t}_r] \quad (2)$$

これにより、ロボットが手先効果器 n_{hand} を物体 n_{dest} に一致させる動作を指定できる。この制約はタスク達成制約と呼ばれる。なお、軌道計画手法における位置は、第3節の場合とは異なり、接触状態に基づいて近似されるという仮定は設けられていないことに注意されたい。

以上の物理制約とタスク達成制約に優先度を設定しそれらに従って制約条件を分類する。これらの制約条件から目標関数を定義し、この目標関数を Lexicographical goal programming(LGP) に基づき優先度に従って最小化を行う。

Fig. 4 Simulation Condition

Table 2 Initial position of each node

p_0	(-0.5, 0.5, 0.5)
p_1	(-0.5, 0.5, 0.5)
p_2	(-0.5, 0.5, 0.5)
p_3	(0, 1.0, 0.0)
p_4	(0, 0.5, 0.5)
p_5	(0.5, 0.5, 0.5)

なお、本研究では、優先度 0 を持つ制約条件として物理制約を、優先度 1 を持つ制約条件としてタスク達成制約を設定している。以上の方法によりロボット動作の解軌道を得る。

4.2 リーチングタスクの生成 操作シーケンスを構成する個々の操作から、リーチングタスクを生成する。 $\mathcal{R}$ を 0 を起点とする、リーチングタスクのインデックス集合とする。リーチングタスク $r \in \mathcal{R}$ に対し、タスク開始時刻 t_r 、終了時刻 $\bar{t}_r$ の二つのパラメータを決定する。リーチングタスク r に対応する作業の移動先物体を n_{dest} とすると、第3節において操作に要する時間はロボットハンドの移動距離に比例すると仮定したことから、式(1)より個々の作業に対する作業時間を近似的に求める。操作に対応するリーチングタスク r の開始・終了時刻 t_r , $\bar{t}_r$ を、 k を比例定数として

$$t_r = \begin{cases} k \cdot d_{n_{\text{dest}}} & (r = 0) \\ \bar{t}_{r-1} + k \cdot d_{n_{\text{dest}}} & (r > 0) \end{cases} \quad (3)$$

$$\bar{t}_r = t_r \quad (4)$$

と算出する。この時区間において、 $p_{n_{\text{hand}}} = p_{n_{\text{dest}}}$ となる制約を課す。

以上のように決定されたリーチングタスクの集合 $\mathcal{R}$ を用いて軌道計画することにより、ロボットの動作軌道を生成する。

5. シミュレーション実験

Algorithm 1 を C++ により実装し、シミュレーションを行った。このシミュレーションにおいて、初期状態、目標状態における接触グラフは Fig.4 に示すように設定した。ただし、 n_3 がロボットアームを表すノードである。ここで、各ノードの位置は Table 2 のように初期化されている。

以上の設定に対し、Algorithm 1 を用いて探索を行った結果、Table 3 に示されるロボットの作業系列が得られ、

Table 3 The obtained task sequence

0:PICK(n_2)
1:PLACE(n_2, n_4)
2:PICK(n_1)
3:PLACE(n_1, n_2)
4:PICK(n_2)
5:PLACE(n_2, n_5)

Smooth Sequencing of Robotic Actions”, IROS2009, St Louis, USA, pp.1609-1614

- [3] 田崎勇一, 鈴木周一, 鈴木達也, “マルチボディシステムの拘束ベース多目的軌道計画”, 日本ロボット学会誌, Vol.31, No.5, pp.508-516

Fig. 5 Snapshots of planned trajectory

このときの総コスト 2.87 が得られた。ただし、式 (3), (4) において $k=2$ とした。なお、Fig.4 の作業環境において接触状態を利用した同時搬送を行わず、物体を一つずつ搬送する場合、総コストは 3.87 となった。このことから、接触状態を考慮することによって、効率的な搬送が行えることが確認できた。

Table 3 の操作シーケンスに対し軌道計画を行った結果、Fig.5 に示されるロボットアームの軌道が生成された。

ただし、Fig.5 の各スナップショット内における実線は、作業全体に対するロボットハンドの軌跡を表している。この軌道にリーチングタスクの達成制約は十分満たされており、所定の操作シーケンスを達成する軌道が生成できた。

6. ま と め

本手法により、作業環境の接触状態を利用したロボットの作業系列を決定することができた。

今回遷移コストとしてロボットの動作距離を式 1 により求めたが、ロボットの搬送作業において物体を持ち上げる、持ち上げている物体を置く動作に対しても作業時間を考慮すべきである。これらの動作の定式化は今後の課題としたい。また、今後の予定として、軌道計画の結果を利用した作業系列の再計画を行う手法についても取り組んでいくことを考えている。

文 献

- [1] D. Nau, Y. Cao, A. Lotem, and H. Munoz-Avila, “SHOP: Simple Hierarchical Ordered Planner” In IJCAI-99, pp. 968-973, 1999. IJCAI’99 Proceedings of the 16th international joint conference on Artificial intelligence
- [2] F. Keith, N. Mansard, S. Miossec, and A. Kheddar, “Optimization of Tasks Warping and Scheduling for