

多関節ロボットのパスのエンコードとデコード

○西田 健 (北九州市立大学)

Encoding and decoding of paths for articulated robots

○NISHIDA Takeshi (The University of Kitakyushu)

Abstract: I propose a method for encoding and decoding paths of an articulated robot. The proposed method encodes and stores paths in the robot configuration space in a format that is independent of the robot's joint lengths and start/goal positions. It also enables fast decoding of the stored path information and reconstructing the paths depending on the situation. This method reduces the cost of reconstructing paths as the situation changes, and at the same time, it allows the integration of multiple paths and the sharing of paths between different robots.

1. 緒言

多関節ロボットを用いた多品種変量生産の自動化には、頻繁に生ずるタスク変更に対する迅速なパス生成の機能が重要である[1,2]。一方で、従来手法には調整パラメータが多く計算コストが高いなどの課題がある[2]。

一般のロボットの動作フロー (Fig.1) では、多関節ロボットのパスは、各関節角度で張られる高次のロボットコンフィギュレーション空間 (C 空間) において、所与のスタートとゴールを結ぶ複数の中間ノード (ウェイポイント) の系列として生成される。C 空間でのパスの運用は高速かつ滑らかさに特長がある一方で、障害物との衝突や自己干渉のチェックはタスク空間 (T 空間) で実施する必要がある。特に、衝突判定は計算コストが高く、パスプランの計算コストの 90% 近くを占めることもある。したがって、タスクが頻繁に変化する状況では、T 空間での衝突判定の回数を減ずる工夫が重要である。

従来手法には、ランダムサンプリングによる手法[3]、パスの評価関数の最小化を勾配探索する手法[4]などがある。しかし、これらの手法は状況に応じてパスを適応的に発見する手法であり、タスクの変化の度にパスを初期状態から再度探索する必要がある。多数の環境とパスの組み合わせを事前に学習する手法[5,6]が提案されているが、学習結果にロボットの運動学パラメータが畳み込まれるため、環境とロボットの物理パラメータの変化に脆弱である。

そこで本研究では、パスのスタートとゴールおよび運動学パラメータに依存しない形式で、C 空間におけるパスをエンコードする手法を提案する。また、エンコードされたパスを、与えられた任意のパスのスタートとゴールに対してデコードする手法を提案する。本手法によって、T 空間での衝突判定の回数を大幅に減じ、パス生成の計算コストの低減が可能になる。

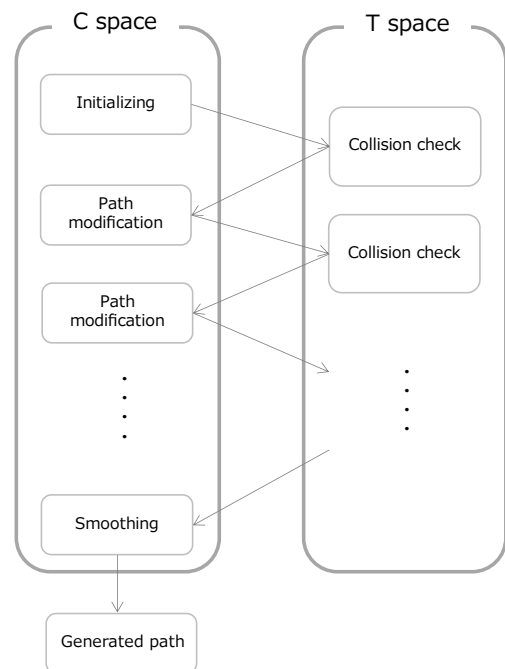


Fig.1 A typical path planning flow for a conventional articulated robot. Paths are heuristically generated by repeatedly mapping between C-space and T-space to avoid colliding with obstacles.

2. パスのエンコードとデコード

2.1 定義

C 空間の座標系を $\Sigma_C \subset \mathbb{R}^N$ と表す。 N はロボットの関節数を表す。所与のパスのスタート、ゴールおよびパスのウェイポイントを以下のように表す。

$$\mathbf{s} \triangleq [s_1 \ \dots \ s_N]^T \in \mathbb{R}^N \quad (1)$$

$$\mathbf{g} \triangleq [g_1 \ \dots \ g_N]^T \in \mathbb{R}^N \quad (2)$$

$$\mathbf{w}^{(m)} \triangleq [w_1^{(m)} \ \dots \ w_N^{(m)}]^T \in \mathbb{R}^N \quad (3)$$

ここで、 $m = 1, \dots, M$ はウェイポイントの番号、 M はウェイポイントのインデックスを表す。さらに、スタート

からゴールへのベクトルを以下で表す.

$$\mathbf{v} \triangleq \mathbf{g} - \mathbf{s} \quad (4)$$

また, \mathbf{v} の正規化ベクトルを以下で表す.

$$\mathbf{e} \triangleq |\mathbf{v}| \in \mathbb{R}^N \quad (5)$$

ここで, $|\cdot|$ はベクトルの正規化を表す. さらに, スタートとゴールを結ぶ線分を L とする.

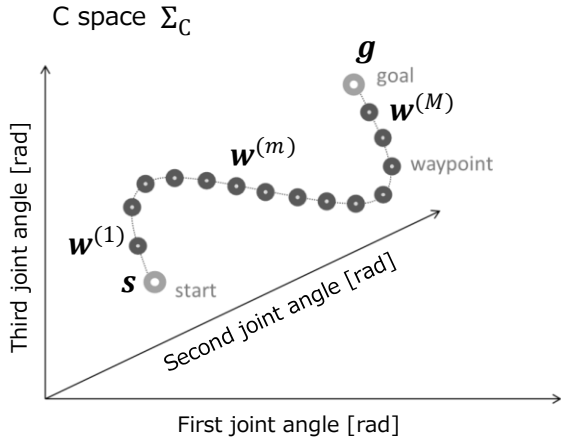
2.2 パスのエンコード

まず所与の \mathbf{s} と \mathbf{g} に対して, 適切なパスプランニング手法の適用により, 以下のように複数のウェイポイントの系列から構成されるパス P が与えられるとする.

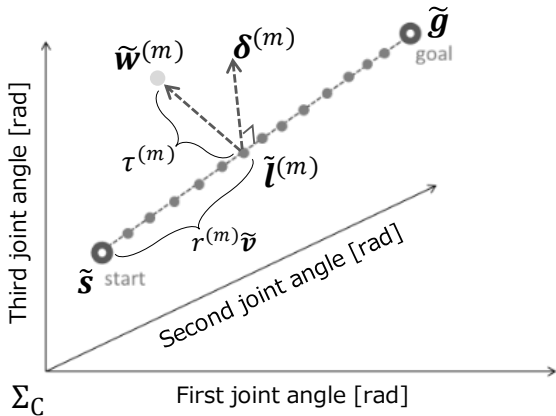
$$P \triangleq (\mathbf{s}, \mathbf{w}^{(1)}, \dots, \mathbf{w}^{(m)}, \dots, \mathbf{w}^{(M)}, \mathbf{g}) \quad (6)$$

次に, $\mathbf{w}^{(m)}$ と L の相対的な関係からパスの概形の情報を抽出する. まず $\mathbf{w}^{(m)}$ から L への垂線の足を以下で表す.

$$\mathbf{l}^{(m)} \triangleq [l_1^{(m)} \dots l_N^{(m)}]^T \in \mathbb{R}^N \quad (7)$$



(a) Approximate shape of a path in C-space (represented up to three axes).



(b) Path information extraction.

Fig.2 Overview of the proposed path encoding process.

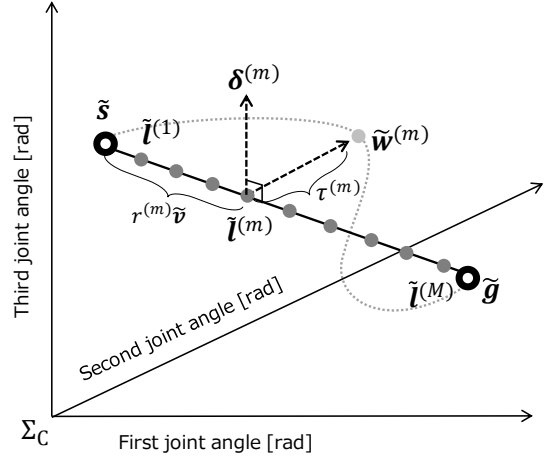


Fig.3 Overview of path seed decoding.

これは次のように求めることができる.

$$\mathbf{l}^{(m)} = \mathbf{s} + \{(\mathbf{w}^{(m)} - \mathbf{s})^T \mathbf{e}\} \mathbf{e} \quad (8)$$

次に, \mathbf{s} を基準とした $\mathbf{l}^{(m)}$ の位置の情報として $r^{(m)}$ を以下のように求める.

$$r^{(m)} = \|\mathbf{l}^{(m)} - \mathbf{s}\| / \|\mathbf{v}\| \quad (9)$$

ここで, $\|\cdot\|$ はベクトルのユークリッドノルムを表す. また, $\mathbf{l}^{(m)}$ と $\mathbf{w}^{(m)}$ の距離を求める.

$$\tau^{(m)} \triangleq \|\mathbf{w}^{(m)} - \mathbf{l}^{(m)}\| \quad (10)$$

さらに, 後述のデコードでウェイポイントの位置を復元するために, $\mathbf{w}^{(m)} - \mathbf{l}^{(m)}$ と L の単位直交ベクトルを以下のように求める.

$$\delta^{(m)} \triangleq |\mathbf{e} \times (\mathbf{w}^{(m)} - \mathbf{l}^{(m)})| \quad (11)$$

この $\delta^{(m)}$ によって, L に対する $\mathbf{w}^{(m)}$ の存在する方向を特定することが可能になる. これらの情報の関係性の概要を三次元に限定して Fig.2 に示す. 以上で求めた情報を次のように行列形式で表現する.

$$R \triangleq [r^{(1)} \dots r^{(m)} \dots r^{(M)}] \in \mathbb{R}^M \quad (12)$$

$$T \triangleq [\tau^{(1)} \dots \tau^{(m)} \dots \tau^{(M)}] \in \mathbb{R}^M \quad (13)$$

$$\Delta \triangleq [\delta^{(1)} \dots \delta^{(M)}] \in \mathbb{R}^{N \times M} \quad (14)$$

これらのパスの情報を組み合わせて管理する.

$$S = \{R, T, \Delta\} \quad (15)$$

以降では, この組み合わせ S をパスシードと呼び, タスクごとに分類された複数のパスシードの集合をパスアーカイブと呼ぶ.

2.3 パスのデコード

ここでは, 前述のパスシードからパスをデコードする手順を示す. まず, 新しいスタート位置 $\tilde{\mathbf{s}}$ と新しいゴ

ール位置 $\tilde{\mathbf{g}}$ が与えられると、それらを結ぶ線分 \tilde{L} が定まる。また、スタートとゴールを結ぶベクトル $\tilde{\mathbf{v}}$ とその正規化ベクトル $\tilde{\mathbf{e}}$ が定まる。

$$\tilde{\mathbf{v}} \triangleq \tilde{\mathbf{g}} - \tilde{\mathbf{s}} \quad (28)$$

$$\tilde{\mathbf{e}} \triangleq |\tilde{\mathbf{v}}| \quad (29)$$

次に、パスシードを別に定めるアルゴリズムによって選定する。その後、 $r^{(m)}$ を利用して \tilde{L} 上に $\tilde{\mathbf{l}}^{(m)}$ を配置する。

$$\tilde{\mathbf{l}}^{(m)} = \tilde{\mathbf{s}} + r^{(m)}\tilde{\mathbf{v}} \quad (30)$$

これは、エンコードでウェイポイントから L への垂線の足を求めた操作の逆の手順であり、デコードでは $\tilde{\mathbf{l}}^{(m)}$ を足として \tilde{L} に直交するベクトルを求める。すなわち、 $\delta^{(m)}$ と $\tilde{\mathbf{e}}$ の外積によって、 $\tilde{\mathbf{l}}^{(m)}$ を足として \tilde{L} に直交するベクトルを求め、ノルム $\tau^{(m)}$ を用いて $\tilde{\mathbf{w}}^{(m)}$ の位置を以下のように算出する。

$$\tilde{\mathbf{w}}^{(m)} = \tilde{\mathbf{l}}^{(m)} + \tau^{(m)}(\delta^{(m)} \times \tilde{\mathbf{e}}) \quad (\forall m) \quad (31)$$

これらの手順によって、新しいウェイポイントが求まり、パス \tilde{P} が生成される。

$$\tilde{P} \triangleq (\tilde{\mathbf{s}}, \tilde{\mathbf{w}}^{(1)}, \dots, \tilde{\mathbf{w}}^{(m)}, \dots, \tilde{\mathbf{w}}^{(M)}, \tilde{\mathbf{g}}) \quad (32)$$

以上のデコードの手順に関する各ベクトルの関係を Fig.3 に示す。パス \tilde{P} の実行の前には、例えば、STOMP[4] などの最適化手法によって、環境との衝突や自己干渉が生じないように一度だけ調整する。Fig.4 にパス生成の流れを示す。

2.3 パスのシードのアーカイブ

ロボットのタスクに関連付けて複数のパスシードをまとめてパスアーカイブを構成する。例えば、ピックアップタスクや組付けタスクなどのパスアーカイブが考えられる。前者についてさらに、ワークを平面上に置く場合のパスシードや、カゴから取り出す場

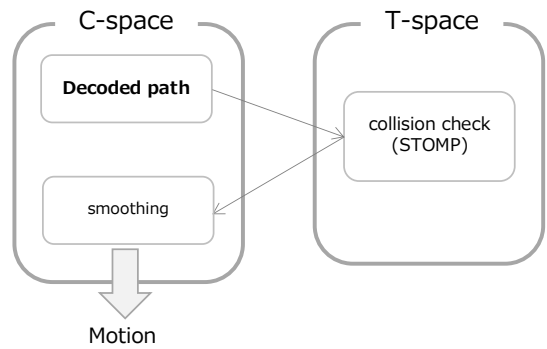


Fig.4 Flow of path planning for an articulated robot using the proposed method. Decoded paths are used after fine-tuning by checking collision with obstacles only once.

合のパスシードなどが考えられる。ロボットにタスクが与えられた場合に、適切なパスアーカイブを選択し、そこからさらに適切なパスシードを読み出して、提案手法によってデコードして利用することになる。その流れを Fig.5 に示す。

4. 結言

多関節ロボットの運動学パラメータやパスのスタートとゴールの情報を含まない形式で、ロボットのパスをエンコードする手法を提案し、エンコードされたパスの情報をパスシードと名付けた。また、タスク毎にパスシードを分類したものをパスアーカイブと呼び、タスクに応じてパスシードをデコードしてパスを生成する手法を提案した。また、本提案手法はパス生成の計算コストを大幅に低減でき、多品種変量生産における自動化に有効であることや、型式の異なる多関節ロボット間でのパスの共有を可能にすることを述べた。

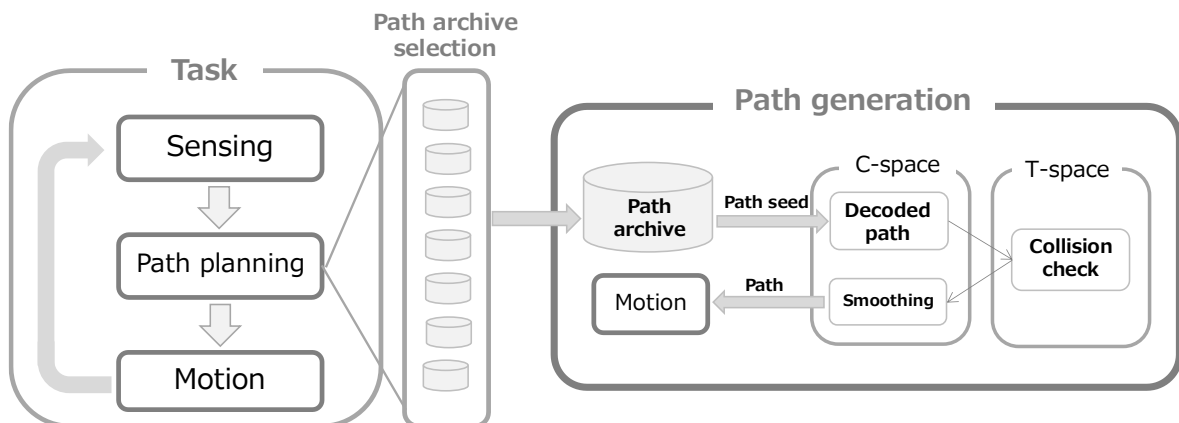


Fig.5 Overview of path archives and path usage. Multiple path seeds are associated with a task and managed together as a path archive. During the path generation procedure, an appropriate path seed is selected and decoded.

今後は、パスアーカイブから適切なパスを選択する手法の確立を進めていく予定である。

参考文献

- [1] 西田, 松永, 多田隈: “中小企業における産業用ロボット導入の障壁とその解決方法の提案”, 計測自動制御学会 *SI* 部門講演会 *SICE-SI* 予稿集 (2014)
- [2] 西田: “可逆な自動化”, *人工知能*, Vol. 37, No. 3, pp. 286-291 (2022)
- [3] R. Kabutan, T. Nishida: “Motion Planning by T-RRT with Potential Function for Vertical Articulated Robots”, *Electrical Engineering in Japan*, DOI10.1002/eej.23103 (2018)
- [4] M. Kalakrishnan, S. Chitta, E. Theodorou, P. Pastor, S. Schaal: “STOMP: Stochastic trajectory optimization for motion planning”, *IEEE Int. Conf. on Robotics and Automation*, pp. 4569-4574 (2011)
- [5] T. Barbié, R. Kabutan, R. Tanaka, T. Nishida: “Gaussian Mixture Spline Trajectory: Learning from a Dataset, Generating Trajectories Without One”, *Advanced Robotics*, Vol. 32, Issue 10, pp. 547-558 (2018)
- [6] M. Inoue, T. Yamashita, T. Nishida: “Robot Path Planning by LSTM Network Under Changing Environment”, *Proc. of The International Conference on Computational Sciences, Advanced Database and Computing*, CS38 (2017)