

微分可能シミュレーションを用いた線状柔軟物操作における接触回避動作計画

○応 長健 (信州大) 山崎 公俊 (信州大)

本研究では、ロボットによる線状柔軟物体の操作において、環境との接触回避を考慮しつつ、目標形状に到達するための動作を生成できるフレームワークを提案した。線状柔軟物体の微分可能シミュレーションを構築し、そこから行動方策を探索することを基本方針とする。探索手法として、現在と目標の状態に基づいて、環境との試行錯誤を通じてニューラルネットワークを end-to-end 学習させることにより、適切な行動方策を決定する。提案手法の有効性はシミュレーションの結果によって確認した。

1. はじめに

日常生活や製造現場では、紐やケーブル、ホース、手術用縫合糸など、様々な線状柔軟物体が使用される。これらの物体は一般に一次的で柔軟であり、外力が作用すると伸びたり曲がったり、ねじれたりといった弾性変形を起こす。このような特性は、操作中の物体の形状を正確に模することを困難にし、作業の自動化を妨げている。さらに、作業環境の中に障害物が存在する場合、ロボットによる障害物との接触を回避しながら作業の自動化を実現することは挑戦的な研究テーマになっている[1]。

本研究の目的は、線状柔軟物体を目標の形状に移行させるための行動方策を決定することである。しかしながら、作業空間の制約により、物体の操作中には環境構造物などの障害物との接触を考慮しなければ、操作後の最終的な形状が目標と大きく異なってしまう可能性がある。そのため、目標の形状を達成するためには、適切な接触回避方策を探ることが重要である。それに関して、従来研究では様々な方法が検討されている。Nozakiら[2]は、質量バネモデルに基づくシミュレーションを使用して、移動するケーブルの形状を模し、粒子と面の衝突を検出することで、ケーブルが構造物に接触しない動作計画を策定した。しかし、計画方策を構造物の違いによって変更する必要があった。McConachieら[3]は、ニューラルネットワークの分類器を用いて、ロープの状態と動作のペアリングし、近くの障害物を確実に回避するための遷移動作を予測する方法を提案した。しかし、適切な衝突回避動作を予測する能力を分類器に付与するためには、大量のデータセットを用意し、手作業によるラベリングが必要であった。一方で本提案手法では、事前に教師データを集める必要がなく、異なる環境構造物に対しても、環境との試行錯誤を通じて適切な行動方策を獲得することができる。本研究の貢献は以下のとおりである：

- Position Based Dynamics (PBD) に基づいて、環境との衝突も考慮した線状柔軟物体のシミュレーションを、end-to-endで微分可能にした。
- シミュレーションにニューラルネットワークを組み込み、勾配に基づく最適化手法によって、目標の形状に到達するための接触回避軌道を生成可能にした。

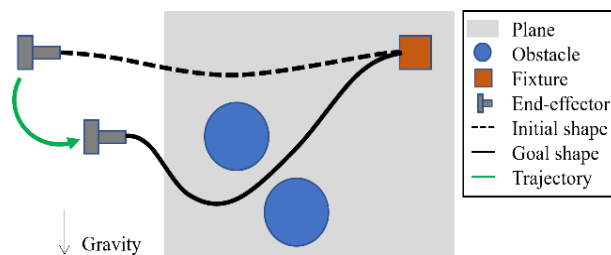


図1 想定した3次元の作業空間

2. 手法の概要

2.1 問題設定

図1に示されるように、3次元の作業空間において、鉛直面に環境構造物が障害物として存在する。ここでは、線状柔軟物体の形状はエンドエフェクタの動作によってのみ決定され、慣性効果の影響を受けないと仮定する。したがって、物体の形状に与える重力の影響を緩和するために、ロボットのエンドエフェクタは3自由度しか動かない、つまり並進運動のみが許容され、回転は行われぬとする。

上記の問題設定における課題は、1つは、エンドエフェクタの移動だけでなく、障害物との接触によっても線状柔軟物体が変形しうることである。その結果、操作後の最終形状が目標形状から大きく異なってしまう可能性がある。これが操作目的の達成を最も妨げる要因となる。もう1つは、障害物配置の変化に応じて、適切な行動方策を自律的に決定することが求められる。

2.2 アプローチ

前述の課題を解決するために、図2に示す手法を提案する。まず、環境に接触した線状柔軟物体の変形を模すために、シミュレーションに衝突判定および反応機能を追加する。その後、シミュレーション全体を自動微分付きフレームワークに移行させ、シミュレーション全体を end-to-end で微分可能にする。そして、線状柔軟物体シミュレーションに組み込まれたニューラルネットワークが、行動方策を近似する。これにより、異なる環境の構造や目標形状が与えられても、環境との試行錯誤を通じてネットワークを end-to-end 学習させることにより、適切な動作を獲得することが可能になる。

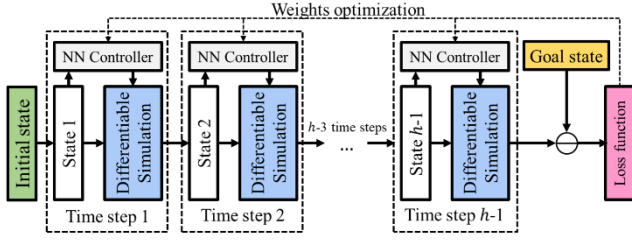


図2 フレームワークの全体

3. 微分可能シミュレーションを用いた動作計画

3.1 Position-based モデリング

線状柔軟物体の動的状態を模するために、PBD に基づいた Cosserat ロッドモデル[4]を利用する。従来のカベースのモデルとは異なり、PBD は力や速度の更新を省略し、直ちに一連の制約によって位置を修正する。このように、外部からの操作は位置で直接表現されるため、オーバーシュートの問題を回避することができる。さらに、衝突制約を追加することで、物体と環境との相互作用を簡単に扱える。さらに、このモデルの解析式は微分可能な関数であるため、自動微分付きフレームワークを使用することで、線状柔軟物体の微分可能シミュレーションを容易に実装することができる。

線状柔軟物体は、位置と方向によって表される n 個の粒子と $n-1$ 個フレームで離散化される (図 3)。位置は直交座標系のベクトル $\mathbf{x} = [x_1, x_2, \dots, x_n] \in \mathbb{R}^{3n}$ として定義され、方向は四元数 $\mathbf{q} = [q_1, q_2, \dots, q_{n-1}] \in \mathbb{R}^{4(n-1)}$ として定義される。以下では表記を簡潔にするため、 $\mathbf{p} = [x_1, x_2, \dots, x_n, q_1, q_2, \dots, q_{n-1}]$ とする。PBD では各時間ステップでの位置と方向の修正量 $\Delta \mathbf{p}$ を、テイラー級数展開で線形化し、以下の方程式を得る。

$$\mathbf{C}(\mathbf{p} + \Delta \mathbf{p}) \approx \mathbf{C}(\mathbf{p}) + \nabla_{\mathbf{p}} \mathbf{C} \Delta \mathbf{p} = \mathbf{0} \quad (1)$$

ここで、 $\nabla_{\mathbf{p}} \mathbf{C}$ は \mathbf{p} に対する \mathbf{C} のヤコビアンである。この方程式の解 $\Delta \mathbf{p}$ が、 $\nabla_{\mathbf{p}} \mathbf{C}$ の方向に制限されるようになる。即ち、ラグランジュ乗数の変化 $\Delta \lambda$ を決めることになる。

$$\Delta \mathbf{p} = \mathbf{M}^{-1} (\nabla_{\mathbf{p}} \mathbf{C})^T \Delta \lambda \quad (2)$$

ここで、 \mathbf{M} は質量・慣性行列である。その後、Gauss-Seidel 解法によって各制約 j に対して個別に計算される。

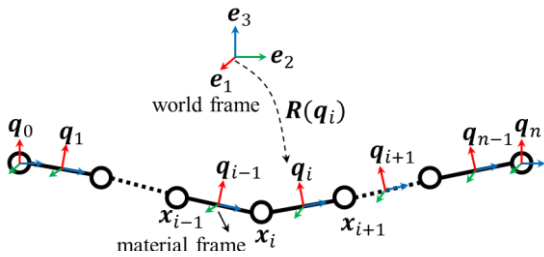


図3 Cosserat ロッドモデルの離散化

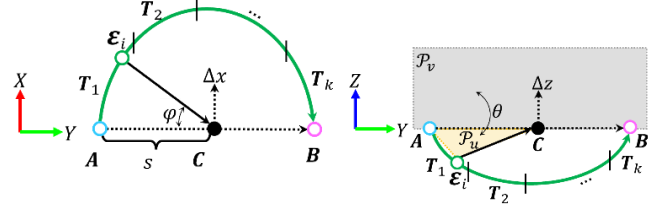


図4 操作軌道のパラメータ化

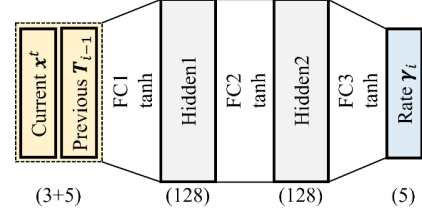


図5 ニューラルネットワークの構造

$$\Delta \lambda_j = -(\nabla_{\mathbf{p}} C_j \mathbf{M}^{-1} \nabla_{\mathbf{p}} C_j^T + \tilde{\alpha}_j)^{-1} (C_j(\mathbf{p}_i) + \tilde{\alpha}_j \lambda_{ij}) \quad (3)$$
ここで、 $\tilde{\alpha} = \alpha / \Delta t^2$ 、 α と Δt はそれぞれ剛性係数と時間ステップを表す。

3.2 ニューラルネットワークによる軌道生成

図 4 に示すように、エンドエフェクタの作業空間は鉛直面の法線方向、すなわち X 軸の正方向に沿った領域とする。この領域内で適切な操作をおこなうための行動方策を獲得する。本研究では、少ないパラメータ数で適切な操作軌道を定義できる方法として、ある 2 点を通る円弧上の曲線を組み合わせて操作軌道を表現する。ただし、軌道の正確な寸法は目標形状や障害物の配置に依存する。そのため、軌道の寸法を定めるパラメータを可変とし、ニューラルネットワークを使用してそれらのパラメータを予測する。軌道上の離散点を直接予測する方法より、この方法では軌道が有効範囲内に制約されるため、シミュレーションの失敗を防ぐことができる。

始点 A から終点 B までの操作軌道は k 個のサブ軌道 $\mathbf{T} = (\mathbf{T}_1, \mathbf{T}_2, \dots, \mathbf{T}_k)$ に分割される (図 4)。各サブ軌道 \mathbf{T}_i は 5 つのパラメータ ($s, \phi, \theta, \Delta x, \Delta z$) によって定義される。これにより、物体の現在の状態に応じて軌道の寸法を調整し、次の操作行動 ϵ_i を予測する。

ニューラルネットワークは 3 つの全結合層で構成される (図 5)。入力層は物体の現在の状態 \mathbf{x}^t と、前のサブ軌道のパラメータ \mathbf{T}_{i-1} である。その後、 \tanh 活性化関数を持つ 2 つの隠れ層が続く。出力層は、現在のサブ軌道の比率ベクトル γ_i である。 \tanh 関数は -1 から 1 の範囲の数値を出力するため、最大ベクトル Λ を乗じた後、サブ軌道パラメータ $\mathbf{T}_i = \gamma_i \cdot \Lambda$ は有効な範囲に制限することができる。

微分可能シミュレーションの誤差逆伝播に基づいて、ニューラルネットワークの重みに対する誤差の勾配が得られる。そして、勾配ベースの最適化方法を用いてニューラルネットワークの重みを更新し、適切な軌道パラメータを予測する。そのためには、予め誤算関数を設計する必要がある。ここで、粒子と障害物との接触効果を考慮した誤差関数を構築する。

$$L_p = \omega_1 \sum_{\Gamma} L_c + \omega_2 MSE(\mathbf{x}^h, \mathbf{x}^*) \quad (19)$$

ただし,

$$L_c = \begin{cases} |\mathbf{x}_{i,x}^t - \mathbf{o}_{c,x}| & \text{if } \mathbf{x}_i^t \text{ c.w. surface} \\ |\mathbf{x}_{i,x}^t - \mathbf{x}_{i,x}^*| + |\mathbf{x}_{i,y}^t - \mathbf{x}_{i,y}^*| & \text{if } \mathbf{x}_i^t \text{ c.w. section} \end{cases} \quad (20)$$

ここで, ω_1 と ω_2 は異なる誤差項間のトレードオフ係数であり, Γ は全時間ステップにおける接触回数であり, h は最終の時間ステップであり, $MSE(\cdot)$ は平均二乗誤差を表す. また, \mathbf{o}_c は障害物断面の中心を指し, c.w.は「接触した」を意味する.

4. 実験

構築した微分可能シミュレーションにより, 操作目的を達成できるかどうかを確認した. 自動微分付きフレームワークでは Diff-Taichi [5] を利用した. 線状柔軟物体の長さは 445 mm, 直径は 8 mm であった. 一端は固定され, もう一端は移動可能にした. シミュレーションでは, 紐を 30 個の粒子で離散化し, 時間ステップは 0.01 sec とし, 総時間ステップ数は 300 とした. すなわち, $n = 30$, $\Delta t = 0.01$ sec であった. 紐と同じ鉛直面の上に 2 本の円柱 (直径と高さ両方とも 60 mm) を設置し, 2 つの円柱の中心間距離を 100 mm とした. ニューラルネットワークの学習に関しては, 操作軌道を 10 個のサブ軌道に分割し ($k = 30$), サブ軌道のパラメータの最大ベクトルは $\mathbf{\Lambda} = (0.008, 1.5, 0.5, 0.05, 0.05)$ とした. 他の学習関連のハイパーパラメータは, トレードオフ係数 $\omega_1 = 1.0, \omega_2 = 0.1$, 学習率は 0.001 とし, 最適化手法には最急降下法を使用した.

4 つの異なる障害物配置のケースで実験を行った. 図 6 に示すように, 紐の目標形状 (ピンク) が与えられた後, 試行錯誤によって行動方策を学習させることで, 円弧状の操作軌道 (緑) を生成することができた. そして, 生成した軌道に沿って紐 (白) の可動側を動かし, 接触回避を成功させ, 目標形状を達成できた. これによって, シミュレーションとニューラルネットワークを組み合わせた提案手法により, 線状柔軟物体における接触回避動作を生成できることを確認した. また, 環境の構成 (青) が変化しても, 本手法によって自律的に行動方策を決定できたことを明らかにした.

さらに, ケース 1 に対して, 接触関連項を無視した誤差関数, すなわち単なる $MSE(\mathbf{x}^h, \mathbf{x}^*)$ 関数を用いて実験を行った. 図 7 に示すように, 誤差値は最終的に紐が障害物に接触した時点で収束し, これ以上減少は見られなかった. この原因は, MSE 関数のみでは, 障害物から離れた方向に動かすようニューラルネットワークを更新すると, むしろ誤差が増加するためである. この比較により, 接触の効果を考慮した誤差関数の優位性を示された.

5. まとめ

本稿では, 線状柔軟物体操作において, 微分可能シミュレーションとニューラルネットワークを組み合

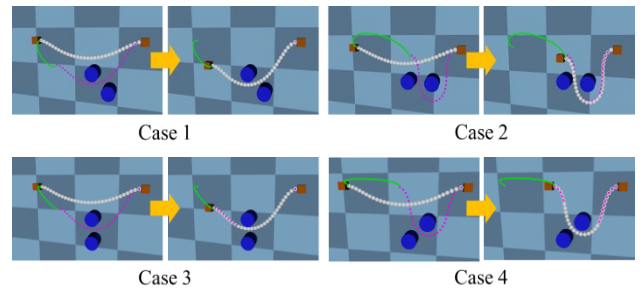


図 6 シミュレーションに回避軌道生成の結果

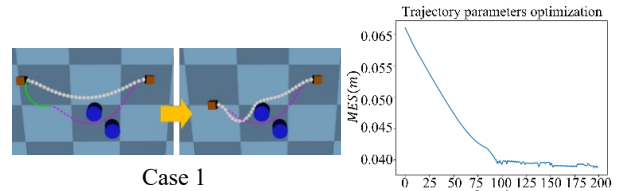


図 7 MSE は誤差関数として軌道生成の結果 (左) 誤差値と往復回数 (右)

わせ, 環境との接触を回避しながら目標形状に到達するための動作生成手法を提案した. シミュレーションの実験結果により, 目標形状や環境の構成を変更しても, 本提案手法によって操作目的を実現できていることが明らかになった. 今後は実機での検証や, より複雑な環境の構成や目標形状に対応できるように, 観測値に基づいたフィードバックを用いて操作行動のオンライン生成を行う予定である.

謝辞

本研究は JST 次世代研究者挑戦的研究プログラム JPMJSP2144 (信州大学), JSPS KAKENHI 23H00492 および JST [Moonshot R&D][Grant Number JPMJMS2034] の支援を受けたものです.

参考文献

- [1] H. Yin, A. Varava, D. Kragic, “Modeling, learning, perception, and control methods for deformable object manipulation,” *Sci. Robot.* 6, eabd8803, 2021.
- [2] K. Nozaki, C. Ying, Y. Matsuura, and K. Yamazaki, “Manipulation Planning for Wiring Connector-Attached Cables Considering Linear Object’s Deformability,” *Int. J. Automation Technol.*, vol.17 no.4, pp. 399-409, 2023.
- [3] D. McConachie, T. Power, P. Mitrano and D. Berenson, “Learning When to Trust a Dynamics Model for Planning in Reduced State Spaces,” in *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3540-3547, 2020.
- [4] T. Kugelstadt, and E. Schömer, “Position and orientation based Cosserat rods”, *Symposium on Computer Animation*, 2016.
- [5] Y. Hu, L. Anderson, T. M. Li, Q. Sun, N. Carr, J. Ragan-Kelley, and F. Durand, “DiffTaichi: Differentiable programming for physical simulation,” *The International Conference on Learning Representations (ICLR)*, 2020.