

ロボットのリアルタイム学習を可能にする リザーバーコンピューティングソフトウェアの開発 —高速オンライン学習による実時間適応制御の実現—

1. 背景

実世界において頑健に振る舞うロボット制御技術として、昨今では深層強化学習や大規模トランスフォーマーといった機械学習を用いた手法が盛んに研究されている。しかし、これらのモデルは膨大なパラメータを有し、学習に多大なデータ量と計算時間を要するだけでなく、推論時にも高い計算資源とエネルギーを消費する。特に実時間制御において、現場で自律的な運動学習をリアルタイムに実行することは極めて困難である。

リザーバーコンピューティング (Reservoir Computing: RC) は、計算コストを大幅に抑えつつ、出力層のみを線形最適化することで極めて高速な学習が可能な手法として注目されている。特に時系列データの処理に優れ、ロボット制御への応用が期待されているが、その実用性は実装の効率に大きく依存する。機械学習分野で現在主流の Python は、プログラムを実行する際の処理時間にわずかな変動が生じやすく、計算負荷の高い行列演算を常に一定の短い時間内で完結させることが技術的に難しいため、実装の仕方によっては RC であっても期待したほどの実行速度が得られない場合がある。例えば、数百ニューロン規模のエコステートネットワーク (Echo State Network: ESN) に対し、逐次最小二乗法 (Recursive Least Squares: RLS) によるオンライン学習を適用した場合、Python/NumPy ベースの素朴な実装ではステップ更新に数十から数百ミリ秒を要し、50Hz から 100Hz 程度の高速な制御周期が求められるロボットのリアルタイム制御への適用は困難であった。本プロジェクトでは、この実装上のボトルネックを解消し、ロボットが現場で即時的な学習を可能にする高速な RC ソフトウェア基盤の構築を目指した。

2. 目的

本プロジェクトの目的は、ロボットの運動学習をリアルタイムかつオンラインで実行可能にする、高速な RC ソフトウェアライブラリを開発することである。採択者はこれまで Python での実装経験を通じてその限界を認識してきた。この知見を基に、計算性能の抜本的な向上を目指し、計算コア部分を C++ で最適化した実装を行い、Python インターフェイスを提供する構成を採用した。

具体的には、以下の達成を目標とした。

1. **高速リザーバー計算ライブラリの開発:** 大規模な RC モデルにおいても 50Hz 以上のロボット制御周期内で動作可能な高速演算基盤の構築。
2. **利便性の高い OSS としての提供:** 機械学習コミュニティで標準的な Python からシームレスに利用できる、オープンソースソフトウェアとしての公開。
3. **ロボット実機での有効性実証:** 7 自由度ロボットアームを用い、教示動作の再現タスクや未知の負荷変動を通じたリアルタイム適応制御の達成。

3. ソフトウェア開発内容

本プロジェクトでは、高速リザーバー計算ライブラリ `rcLib` を開発した。本ライブラリは、行列演算ライブラリ Eigen をベースに C++17 で実装され、`pybind11` を用いて Python バインディングを提供している。主な技術的特徴と最適化内容は以下の通りである。

- **リザーバー計算の高速化:** リザーバー内部の結合の大部分がゼロであるという疎行列の性質を活かした最適化計算ループを実装した。これにより、大規模なネットワークでも無駄な計算を排除し、リザーバー状態の更新速度を大幅に向上させた。また、リザーバーの動特性を決定する指標であるスペクトル半径の算出において、反復計算によって必要な値のみを効率的に推定する手法を採用し、初期化時間を短縮した。

- オンライン学習 (RLS) の最適化:** オンライン学習のアルゴリズムである RLS において、扱う行列が対称であることを利用し、計算範囲を半分に絞ることで演算量を約 50%削減した。また、ステップ更新のための計算式を工夫して変換することで、行列演算ライブラリ Eigen が備える高度に最適化された専用関数を利用可能にし、CPU の演算能力を最大限に引き出した。
- 堅牢な制御基盤の構築:** 7 自由度マニピュレータ CRANE-X7 (図 1(a)) を対象に、ROS 2 を可視化ツールとして利用しつつ (図 1(b))、ハードウェア通信を C++ で直接制御する低レイテンシな基盤を構築した。各ステップの計算処理に要した時間から次の実行までの残り待機時間を算出することで、一定の制御周期を厳密に維持するタイミング管理機能や、関節角の急変を検知するセーフティ機能により、安定した運動制御を実現した。

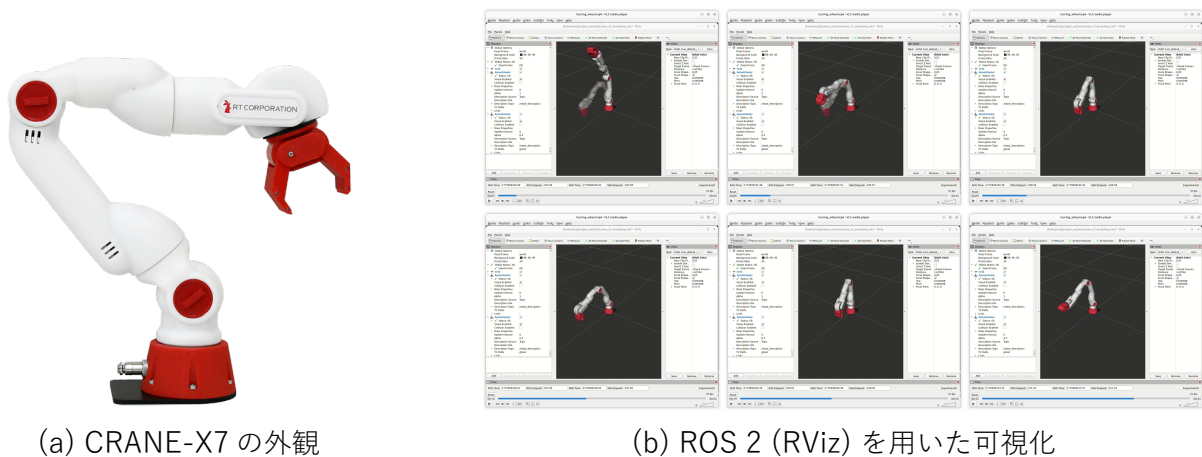


図 1: 実験に使用したロボットシステムと可視化環境

4. 新規性・優位性

4.1 既存ライブラリとの計算性能比較

本ライブラリはオフライン学習 (リッジ回帰) において、既存の主要な RC ライブラリである ReservoirPy と比較して高い優位性を持つ。Mackey-Glass 方程式で定義されるカオス時系列の 1 ステップ先予測を採用したベンチマークテストにおいて、20,000 ニューロン規模における推論速度の比較では、ReservoirPy と比較して約 2.6 倍の高速化を達成した (表 1)。

表 1: オフライン学習における計算時間 (Fit: 学習時間, Pred: 推論時間) の比較

Neurons	rclib Fit (s)	reservoirpy Fit (s)	rclib Pred (s)	reservoirpy Pred (s)	改善率 (Pred)
1,000	0.480	0.300	0.051	0.076	1.5x
4,000	4.521	6.575	0.253	0.600	2.4x
10,000	71.871	69.752	2.003	6.139	3.1x
20,000	453.502	439.342	9.885	25.612	2.6x

また逐次的なステップ更新が必要なオンライン学習において、表 2 に示すように最大で約 23.3 倍、10,000 ニューロン規模で約 7.7 倍の RLS 更新速度の高速化を達成した。50Hz の実時間制御を想定した場合、従来の Python 実装では 2,000 ニューロン程度でリアルタイム性が失われるのに対し、rclib では 4,000 ニューロン規模でも十分な時間的余裕を持って制御ループ内で動作可能であることを実証した。

表 2: オンライン学習 (RLS) における 1 ステップあたりの平均更新時間の比較 (s/step)

Neurons	rclib (s/step)	reservoirpy (s/step)	改善率
500	0.00007	0.00056	7.7x
1,500	0.00066	0.01535	23.3x
2,000	0.00137	0.02269	16.6x
4,000	0.01094	0.10340	9.5x
10,000	0.07213	0.55306	7.7x

4.2 ロボット実機を用いた有効性実証

実機検証では図 2(a) に示す制御系のもとでティーチングプレイバック実験を行い、教示した関節角軌道をオフライン学習した ESN によって再現できることを確認した (図 2(b))。

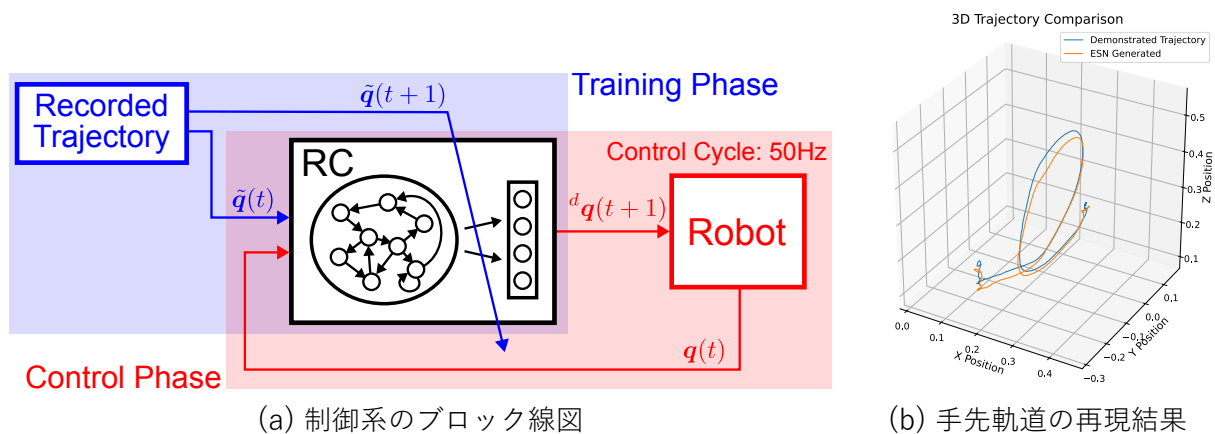


図 2: ティーチングプレイバック実験の構成と再現結果 (橙: 教示、青: ESN 出力)

また、人間の小脳における運動学習を模したフィードバック誤差学習に基づき、オンライン学習によって未知の荷重変化に対応する適応的重力補償制御を実現した (図 3)。この仕組みでは、重力の影響でロボット内部の制御器が解消できない運動の誤差に対して、リザーブ (ESN) が必要な修正量を予測するように学習することで目標とする軌道に収束させる。

実験では、200g の荷重を手先に取り付けた際、当初は荷重変化の影響で軌道が下方に沈み込むが、リザーブが必要な修正量を即座に学習し補正することで、数秒以内に目標軌道への追従運動を回復させることを確認した (図 4)。これは、リザーブが重さなどの影響によるロボットの動的な特性をリアルタイムに理解し、適切な補正信号を自律的に生成できるようになったことを示している。

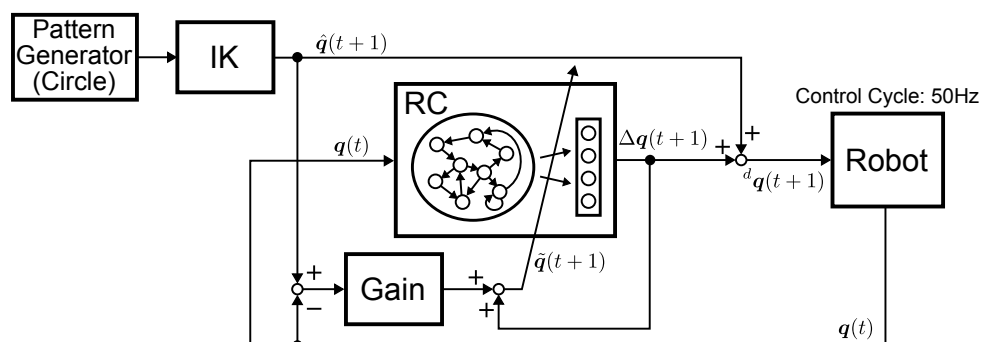


図 3: 適応的重力補償におけるフィードバック誤差学習のブロック線図

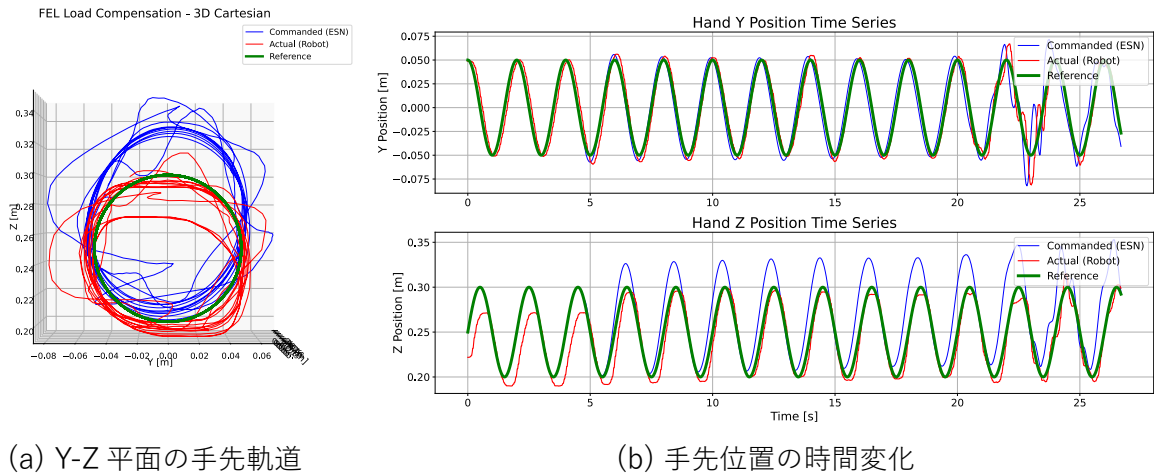


図 4: オンライン学習を適用した場合の適応動作

5. 期待されるユーザー価値と社会へのインパクト

本成果は、計算資源が限られた現場で即時に学習する必要がある状況、特にエッジデバイス上での「Physical AI」の実現を加速させるものである。

ユーザー価値の面では、ロボット開発者が複雑な C++ による最適化コードを自作することなく、Python 上で高速な適応学習を容易に実装可能となる。これにより、個体差や経年変化、未知の外部環境に即座に適応するロボットの開発工数が大幅に削減される。今後はより複雑な動特性を持つソフトロボットへの適用や人とロボットの協調タスクの実現も期待される。研究者にとっても、機械学習分野で標準的な Python からシームレスに利用できるため、RC の計算性能を多角的に検証できる。本ライブラリは高度にモジュール化されており、リザー内部の可塑性や自己組織化機能といった新規のモデルや学習則の導入も容易であり、今後の RC 技術の実用性を広げることに貢献する。

社会へのインパクトとしては、膨大なシミュレーションを経て実機へ適用する sim2real 転移技術を介さず、実機での直接的な高速学習を可能にすることで、動的に変化する現場や多品種少量生産の現場における自動化を促進する。また、RC の低計算コスト特性は、大規模なクラウド資源を必要としない省電力な AI 運用に寄与し、オンボードで学習・推論を完結させるエッジ AI の普及に向けた重要な一歩となる。本プロジェクトで構築した計算基盤は、データ効率に優れた持続可能な AI の新たな方向性を提示するものであり、実世界で実際に行動しながら賢くなる自律システムの実現に向けた基盤となる。

6. 氏名 (所属)

熱田 洋史 (大阪大学 先導的学際研究機構附属共生知能システム研究センター)

(参考) 関連 URL

- GitHub: <https://github.com/hrshtst/rclib>
- Document: <https://hrshtst.github.io/rclib/>
- PyPI: <https://pypi.org/project/rclib/>