

ROS ロボットと連動する VOD 型 e-Learning システムの実現

Implementing an e-Learning VOD system linked with actual ROS robot

○学 瀧本 恒平 (はこだて未来大) 松原 克弥 (はこだて未来大)
正 鈴木 昭二 (はこだて未来大)

Kouhei TAKIMOTO, Future University Hakodate, g2121032@fun.ac.jp

Katsuya MATSUBARA, Future University Hakodate

Sho'ji SUZUKI, Future University Hakodate

ROS, an open source robot control software platform, is widely used for robot development. Accordingly, robot development engineers are required to have skills in robot control programming using ROS, but in most of the existing ROS-learning contents, it is not possible to confirm that the program and the robot are running in synchronization in the real world at hand. Therefore, we may lose the significance of checking at hand, such as understanding the difference between the virtual world and the physical world. In this study, we propose an e-Learning system for ROS programming that links a lecture-style video with an actual robot at hand. By measuring the latency from the time the video is played to the time the actual robot at hand starts to move, we demonstrated the practicality of the system and summarized the problems and future prospects.

Key Words: Robot Operating System (ROS), e-Learning, actual robot

1 はじめに

OSS ロボット制御ミドルウェアである Robot Operating System (ROS) は、開発をサポートする豊富なツール、再利用性の高いパッケージ群やそれらを柔軟に組み合わせるためのノードグラフ・モデルの採用により、多くのロボットシステムで採用が進んでいる。ROS の普及にともなって、ROS を用いたロボット制御プログラミングのスキルを持つエンジニアの需要が高まっている。実際、これに対応するように、ROS を用いたロボット制御プログラミングの学習教材や学習環境が充実してきている。コンピュータ上で動作させる通常のプログラミングと比較して、ロボット制御プログラミングでは、プログラミング言語や API 等の理解だけでなく、ロボットの物理的な装置の挙動について意識する必要がある。ロボットの物理的な装置の挙動で意識すべき点とは、ロボットが移動する際の加速度や、ロボットが移動した際にバランスを保ったまま移動できているかどうか等を指す。しかし、ロボット制御プログラミングに関する既存の学習教材の多くは、プログラムコードやその解説が記述されたテキスト形式のコンテンツが主体となっているため、教材が対象としているロボットの物理的な装置の挙動を伝えることや、その挙動を意識しながら学習することが難しいという課題がある。

また、コロナ禍における対面教育の制限にともなって、プログラミング学習に対する e-Learning の需要が拡大している。e-Learning による学習は、教材の質が均一であり、講師毎の違いに左右されず、時間と場所を選ばずに繰り返し学習できることがメリットである。一方、e-Learning による学習では、学習を継続するためのモチベーションの維持が難しいことや、本研究が対象とするロボット制御のような、実習・実技をとまなう学習を十分にサポートすることが難しいという課題が存在する。ロボット制御プログラミング向けの既存 e-Learning 教材では、ロボットを動かしている様子を撮影した動画や CG で再現した仮想ロボットを使ってロボットの挙動を解説しているものが多く、前述の学習モチベーションの維持や、実世界におけるロボットの挙動を意識するという点において、動画や CG だけではロボットの挙動を解説する方法として十分とは言えない。

本研究では、ロボット制御プログラミングを対象とした e-Learning において、学習モチベーションの向上と実勢会でのロボット挙動を意識した学習の実現を目的として、講義形式の動画の解説に合わせて手元のロボットが連動し、ロボット内で動作しているプログラムの状態理解を支援するシステムを提案する。提案システムの概要を、図 1 に示す。学習者は、ロボットの挙動とプログラムコードを解説した動画を Web ブラウザ内で視聴する



Fig.1 Overview of the proposed system

ことで学習を進めると同時に、視聴している動画が指定されたタイミングまで進むと、Web ブラウザ上で対応するロボット制御プログラムが実行され、手元にある実機ロボットが動画と連動して動作する。これにより、プログラムとロボットが実世界で同期して動いていることを手元で確認しつつ、e-Learning の持つメリットである、時間と場所を問わずに繰り返し学習できることと教材の質が均一であることを活かした学習ができるようになる。

2 既存の e-Learning システムの課題

ロボット制御プログラミングを対象とした既存の e-Learning システムを紹介し、それらが抱える課題について述べる。

Udemy[1] は、オンラインの学習プラットフォームであり、専門的な知識を解説したスライドや、ソースコードを記述しながら口頭で解説している様子に加え、その実行結果を記録した動画を用いて学習できる。

The Construct[2] は、ソースコードや専門的な知識を解説したテキストを閲覧しつつ、ブラウザ上に用意された仮想開発環境でコマンドやソースコードの記述を行い、シミュレータ上でロボットを動作させることができるオンライン学習サービスである。また、実機ロボットの操作権限を事前に予約しておくことで、The Construct が所有している実機ロボットをカメラ越しに動作させることもできる。

Gustavo らのロボット制御プログラミングの学習支援システム [3] では、Web ブラウザ上で ROS のソースコードを記述し、プログラムを実行することで、Web ブラウザ上にシミュレータの動作や出力結果を表示するシステムを提案している。提案の実現により、学習者は開発環境の構築や、コンパイル等の手間を掛けずにロボット制御プログラミングを学習できる。

これらの既存のロボット制御プログラミングを学習可能な e-Learning システムでは、動画やシミュレータを用いてロボットの

物理的な装置の挙動を確認するようにしているが、動画やシミュレータによるロボットの動作確認は、プログラムとロボットが実世界で同期して動いていることを手元で確認できないという課題がある。ロボットを手元で確認する意義には、

- 仮想的な世界と物理的な世界の違いを理解すること
- 現実のものが動くことによる面白さ
- ハードウェアの仕組みに関する関心が深まること
- 学習してきた物理・数学が実世界とどう関わっているか知ること
- ロボット本体と周辺環境に触れて試行錯誤すること

等がある。これらのロボットを手元で確認する意義は、実機があるからこそ学ぶことができるものであり、仮想的な世界のみでは得ることができない。

3 提案

動画やスライドといった順序型コンテンツと実機デバイスの連動によりプログラミング学習の支援を行っている研究として、川谷らの研究 [4] がある。この研究では、Web 上に存在する IoT プログラミング学習が可能な順序型コンテンツに対して、コンテンツの順序に応じたプログラムを埋め込むことで、特定のタイミングでソースコードの表示や IoT デバイスの制御が可能なシステムを提案している。提案の実現により、順序型コンテンツ内の IoT デバイスと手元の IoT デバイスが連動することで、学習者の IoT デバイスに対する理解度と学習意欲の向上を図っている。

本研究では、川谷らの研究から着想を得て、動画と実機ロボットが手元で連動することで、ロボット制御プログラミング学習の支援を行うシステムを提案する。提案システムでは、動画の再生位置にもとづいて、手元の実機ロボットの制御を行い、制御に用いられた制御コマンドをブラウザ上で確認できるようになっている。ユーザは、動画内のロボットやソースコードの解説と連動した実機ロボットの動作を見ることができると、動画やシミュレータのみで学習するよりもロボットの物理的な装置の挙動について理解しやすくなる。また、手元の実機ロボットが動作した際の制御コマンドを確認することができると、ユーザ自身で同様のプログラムを再現しやすくなる。想定されるシステムの利用ケースとして、ユーザが PC でロボット制御プログラミング学習動画を視聴している際、動画内のロボットが前進すると、ユーザの手元にある実機ロボットも連動して前進する。その際、動画を視聴しているブラウザ上にリアルタイムで制御コマンドが表示される。

提案システムに必要な要件として、以下の 2 つがある。

1. 実機ロボットがユーザの手元で動作すること
2. ロボットの動作に用いた制御コマンドを表示すること

要件 1 は、動画内でロボットの動作を見るだけでなく、実際に手元の実機ロボットが動作している様子を確認することで、ロボットの物理的な装置の挙動に関する理解度の向上や、学習に対するモチベーションの向上に繋がるため設定した。要件 2 は、ユーザが実際にロボット開発を進める時に、動画の再生位置のどの部分でロボットにどのような制御コマンドが送信されていたのかを知ること、ロボットの動作に要するセンサの値について理解しやすくなるため設定した。本研究では、これらの要件を満たすシステムを構築する。

4 実装

本章では、本提案システムの主要部分である、動画と手元の実機ロボットを連動させる機能の実装について述べる。実装するシステム構成図を図 2 に示す。本実装では、動画コンテンツ、動画に埋め込むスクリプトプログラム、動画の視聴とスクリプトの実行を行う Web ブラウザ、ロボットと Web ブラウザ内のスクリプトが連動するためのローカルサーバからシステムを構成する。

動画と手元の実機ロボットの連動の実現を行うために、動画にスクリプトプログラムを埋め込み、動画の再生位置に合わせてプログラムを実行する機能の実装と Web ブラウザからロボットに制御コマンドを送信する機能の実装に加え、提案システムに対応

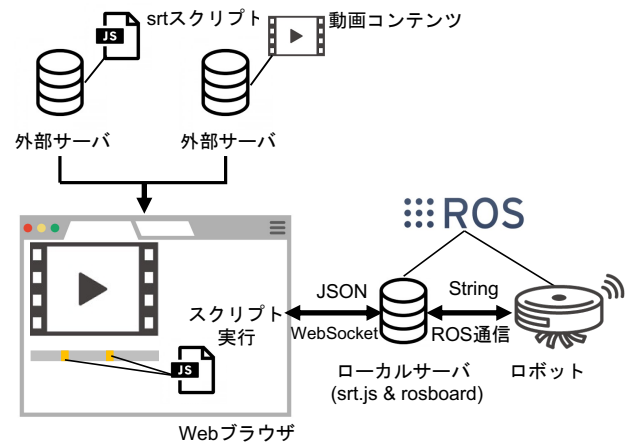


Fig.2 System configuration

```
1 1
2 00:00:01,000 --> 00:00:05,999
3 const turtle = [0.000,0.09];
4 controller(turtle);
5
6 2
7 00:00:06,000 --> 00:00:10,999
8 const turtle = [0.000,-0.09];
9 controller(turtle);
```

Fig.3 Example of srt script description

した ROS 学習動画に埋め込むプログラムの記述に関する制約を設定する必要がある。

動画にスクリプトプログラムを埋め込み、動画の再生位置に合わせて実行する機能の実装には、srt.js[5] を用いる。srt.js は、YouTube 等の映像コンテンツの字幕形式として公式にサポートされている srt ファイルの形で JavaScript を埋め込むことができ、動画上の再生位置に応じてプログラムを実行できるフレームワークである。図 3 に srt スクリプトの記述例を示す。srt スクリプトには、動画上の再生位置を記述し、そのタイミングで実行したい JavaScript プログラムを記述する。これにより、動画の再生位置に合わせてプログラムを実行する機能が実現される。

ただし、動画を一時停止してもプログラムの実行が停止されないことや、動画をスキップした際に srt スクリプトで指定した動画の再生位置を超えてしまうとそのプログラムを実行できないことがある。そのため、手元の実機ロボットが動画内で想定されている動作以上の範囲で動作してしまうことや、ユーザが動画に対して行うスキップ等の操作を制限してしまうことになる。この課題を解決するために、ユーザが動画に対して行う操作で考えられるケースについて検討し、それぞれに応じた処理を追加した。ユーザが動画に対して行う操作として、以下の 3 つが考えられる。

1. 動画が最初から最後まで通して再生された
2. 動画が一時停止された
3. 動画が前後どちらかにスキップされた

1 つ目のケースに関して、動画の再生状態が終了に変化した時、実機ロボットの動作を停止させる制御コマンドを送信する処理を追加することで、srt スクリプトにロボットを停止させるスクリプトが含まれていなかった場合であっても、ロボットを停止させることができるようにした。

2 つ目のケースに関して、動画の再生状態が一時停止に変化した時、実機ロボットの動作を停止させる制御コマンドを送信する

処理を追加することで、動画を一時停止してもプログラムの実行が停止されず、ロボットが動き続けてしまうという課題を解決した。また、動画の再生状態が一時停止から再生中に戻った時、一時停止前に実行していたスクリプトを再度読み込んで実行する処理を追加することで、一時停止状態から復帰した際に再びロボットを動作させることができるようにした。

3つ目のケースに関して、動画の再生中に再生位置がスキップされる際、必ずバッファリングが発生することから、バッファリングが発生した時に実機ロボットの動作を停止させる制御コマンドを送信する処理を追加することで、動画のスキップ中にロボットが動作することを防ぎつつ、スキップ先の再生位置で指定されているスクリプトの実行ができるようにした。

これらの処理を実装することで、手元の実機ロボットの想定外の動作を防ぎつつ、ユーザは自分のペースで動画を用いた学習ができるようになる。

Web ブラウザからロボットに制御コマンドを送信する機能の実装には、rosboard[6]を用いる。rosboard は、ROS ロボットが送受信するデータをブラウザ上で可視化でき、ブラウザと ROS ロボット間での通信もサポートできるフレームワークであり、Web サーバを実行する ROS ノードである。他のブラウザと ROS ロボット間の通信をサポートするフレームワークと比較して、通信をサポート可能な ROS のバージョンが幅広いことや、ROS ノード間で通信されるデータの可視化までサポートしていることから、本研究で扱う Web ブラウザと ROS ロボット間の通信をサポートするフレームワークとして選択した。rosboard は、WebSocket 通信により、ブラウザから ROS ノードに対して同期的にデータを送信する。このとき、ロボットを動作させるためにブラウザから送信されるデータの整形や処理を追加することで、srt スクリプトの作成者や提案システムのユーザがデータ形式の違いを意識することなく、Web ブラウザとロボット間で通信が可能となる。これにより、srt スクリプトに書き込まれた JavaScript プログラムと、Python や C++ で記述され、ロボット内で実行される ROS ノードとの通信を同期的に仲介することができるようになる。

また、提案システムに対応した ROS 学習動画に埋め込むプログラムの記述に関する制約として、以下の3点がある。

1. ロボットを動作させるスクリプトの後に、必ず停止させるスクリプトを記述させること
2. スクリプトを実行する時間を正しく記述すること
3. srt スクリプトの1つの字幕要素内に複数の連続した動作を記述しないこと

制約1については、動画が一時停止された、または終了した際にロボットの動作を停止させる機能については提案システムで実装しているが、動画を途中で停止せずに視聴し続けた場合にロボットの動作をシステム側で保証する機能を持っていないことから設定した。そのため、学習コンテンツの開発者はロボットの一連の動作について srt スクリプト内で完結するように記述する必要がある。これにより、手元の実機ロボットが動画内のロボット以上に動作することを防ぐことができる。

制約2については、動画の再生状態が一時停止から再生中に変化した、または再生位置がスキップされた際に、動画とロボットの連動が行われなくなるため設定した。これは、提案システム内で、一時停止時やスキップ時にロボットを停止する制御コマンドを送信するようにしているために起こるものである。この制約を満たすことで、srt スクリプト内で指定する動画の再生位置を、次のスクリプトを記述する動画の再生位置の直前までを指定することで、動画の再生状態が一時停止から再生中に変化した、または再生位置がスキップされた際でも再度 srt スクリプトの読み込みが可能となる。

制約3については、動画の再生状態が一時停止から再生中に変化した、または再生位置がスキップされた際に再度 srt スクリプトの読み込みを行うため設定した。再度 srt スクリプトを読み込むことで、動画内でロボットの動作に必要な時間として想定されているよりも短い時間で手元の実機ロボットが動作し、srt スクリプトの実行途中で別の srt スクリプトの実行に移ってしまう可能性がある。この制約を満たすことで、動画内でロボットが複数の連続した動作を行っても、srt スクリプトに動作を記述する

際は独立した動作として分割することで、srt スクリプトの再読み込みが発生しても動作が途中で切り替わることを防ぐことができる。

現在の実装状況について、ユーザが動画に対して操作を行った際にロボットの動作を停止する処理は、移動ロボットの加速度を0にすることで一時的に実装しているため、他の動作に対応することができない。そのため、今後は汎用的なロボットの動作を停止する機能を実装し、システムに組み込む必要がある。

5 実験

提案システムでは、動画と手元の実機ロボットが連動することで学習者の学習効率と意欲の向上を図っているが、学習するコンテンツによっては動画とのより厳密な同期が必要とされる場合、手元の実機ロボットが動作するまでのレイテンシが問題になる可能性がある。そのため、本実験では動画と手元の実機ロボットが連動する際のレイテンシと、動画が停止されてから手元の実機ロボットが停止するまでのレイテンシを計測する。実験によって得られたレイテンシの値から提案システムの実用性について検討する。

動画と手元の実機ロボットが連動する際のレイテンシは、ロボットが前進する制御コマンドを含んだ srt スクリプトを組み込んだ提案システム上でロボット制御プログラミングを学習できる YouTube 動画を再生し、動画が再生されてから1秒後に srt スクリプトが実行されるタイミングでの Unix 時間と、制御コマンドを受け取ったロボットが前進するタイミングの Unix 時間を10回ずつ出力し、その差を求めることで計測した。

動画が停止されてから手元の実機ロボットが停止するまでのレイテンシは、提案システムのブラウザに組み込まれた動画の再生状態が一時停止に変化し、実機ロボットを停止させるスクリプトが実行されたタイミングでの Unix 時間と、制御コマンドを受け取ったロボットが停止するタイミングの Unix 時間を10回ずつ出力し、その差を求めることで計測した。

実験環境を表1に、動画と手元の実機ロボットが連動する際のレイテンシ計測結果を表2に、動画が停止されてから手元の実機ロボットが停止するまでのレイテンシ計測結果を表3に示す。表から、動画と手元の実機ロボットが連動する際のレイテンシの最大値は0.200秒であり、最小値は0.022秒、平均値は0.131秒であった。また、動画が停止されてから手元の実機ロボットが停止するまでのレイテンシの最大値は0.500秒であり、最小値は0.126秒、平均値は0.193秒であった。

実験結果より、提案システムにおける動画とロボットが連動する際のレイテンシと動画が停止されてから手元の実機ロボットが停止するまでのレイテンシは、実機ロボットに対して厳密に動画と同期する動作を要求する際には考慮するべき点にはなるが、一般的な学習用ロボットを用いて学習を行う上では大きな問題にならないと考察できる。

6 おわりに

本稿では、講義形式の動画にプログラムを埋め込み、動画と手元の実機ロボットが連動することでロボット制御プログラミング学習を支援する e-Learning システムの提案・実装について述べ、動画と手元の実機ロボットが連動する際のレイテンシと動画が停止されてから手元の実機ロボットが停止するまでのレイテンシを計測することで、システムの実用性を評価した。

今後の展望として、実装で述べた汎用的なロボットの動作を停止させるスクリプトを生成する機能と、学習者が試行錯誤を繰り返して学びを得る機会を作るために Web ブラウザ上でプログラムの編集・記述を行う機能の追加を行った上で、システムを利用した際の学習効率と意欲の変化について評価を行う。

Table 1 Experimental environment

ブラウザ	Firefox Browser 96.0(64-bit)
サーバ OS	Ubuntu 20.04.3 LTS
ロボット	Turtlebot3 Burger
ロボット搭載 OS	Ubuntu 20.04.2 LTS
ROS 2	Foxy Fitzroy

Table 2 Latency when the video is linked to the actual robot at hand

計測回数	レイテンシ (秒)
1 回目	0.112
2 回目	0.187
3 回目	0.022
4 回目	0.060
5 回目	0.111
6 回目	0.200
7 回目	0.189
8 回目	0.153
9 回目	0.122
10 回目	0.155

Table 3 Latency from when the video is stopped to when the actual robot at hand stops

計測回数	レイテンシ (秒)
1 回目	0.146
2 回目	0.126
3 回目	0.149
4 回目	0.235
5 回目	0.173
6 回目	0.149
7 回目	0.133
8 回目	0.166
9 回目	0.500
10 回目	0.149

参考文献

- [1] Benesse, “オンラインコース – いろんなことを, あなたのペースで [Udemy], Udemy, <https://www.udemy.com/> (参照日 2021 年 7 月 23 日).
- [2] The Construct, “The Construct: A Platform to Learn ROS-based Advanced Robotics Online,” <https://www.theconstructsim.com/> (参照日 2021 年 7 月 23 日).
- [3] Gustavo A. Casan, Enric Cervera, Amine A. Moughlbay, Jaime Alemany and Philippe Martinet., “ROS-based online robot programming for remote education and training,” Proc. ICRA 2015, pp.6101–6106, 2015.
- [4] 川谷知寛, 塚田浩二, 栗原一貴, “IoTeach: 実世界と順序型コンテンツを連携した IoT プログラミング学習支援システム,” 日本ソフトウェア科学会 WISS 2021, 2021.
- [5] 栗原一貴, 橋本美香, “srt.js:映像コンテンツへの IoT 指向拡張プログラム埋め込みフレームワーク,” 日本ソフトウェア科学会 WISS 2016 論文集, pp.24–29, 2016.
- [6] dheera, “rosboard,” <https://github.com/dheera/rosboard> (参照日 2022 年 2 月 9 日).