

## ROS を用いたアームロボット制御 2 モーションキャプチャを用いた動作教示

キーワード：ロボット、ROS、教示、軌道生成

### 1. はじめに

近年、人手不足がますます深刻化し、ものづくり現場での自動化への需要が高まっています。しかし、ロボットを用いた自動化は、ロボット専用言語によるプログラミングや専用機器による動作教示の難しさが障壁となり、多品種・少量生産が多い中小企業では普及が進んでいません。そこで、モーションキャプチャを用いて直感的に動作教示が行えるシステムを、汎用的プログラミング言語(C++, Python)で制御できるフレームワークである ROS<sup>1)</sup>(Robot Operating System)を用いて構築しました。

本稿では、動作教示に用いた軌道生成ライブラリについて紹介するとともに、ROS を用いて構築した動作教示システムを紹介します。

### 2. 軌道生成ライブラリ-Descartes<sup>2)</sup>

前回のテクニカルシート<sup>3)</sup>では、軌道生成ライブラリとして MoveIt<sup>4)</sup>を紹介しましたが、ここでは、軌道生成ライブラリの一つである「Descartes」を用います。Descartes は、溶接などの手先の経路や姿勢の制御が必要な作業を、アームロボットに行わせるために開発された軌道生成ライブラリです。アームロボットの手先を表す位置と姿勢の時系列データを与えることで、目標とする軌跡を実現する各関節の角度を計算し、軌道生成を行います。開発元の ROS-Industrial によるチュートリアル<sup>5)</sup>が充実しているため、容易に ROS のシミュレーション環境上で使用できます。

### 3. Descartes とモーションキャプチャを用いた文字書き取り動作の教示

本事例では、人が文字を書き取る際のペンの動きをモーションキャプチャで取得し、そのデータを基に Descartes により軌道生成することで、アームロボットに人が書いたものと同じ文字を書き取らせました。使用したアームロボットの仕様は表 1 の通りです。

アームロボットへの動作教示の手順を①～③に分けて説明します(図 1 参照)。

手順①：アームロボットの手先の位置、姿勢、時刻を示すデータを用意します。

手順②：アームロボットの手先の位置、姿勢を実現する各関節の角度を計算します。各関節が、与えられた時刻に計算した角度になるよう、関節角の時系列データを生成します。

手順③：制御用ライブラリを用いて、各関節角が時系列データに従うようにモータを制御します。

表 1 アームロボットの仕様<sup>6)</sup>

型式	株式会社アールティ製 Sciurus17
作業有効範囲	直径 1200 mm
可搬重量	約 0.5 kg
自由度	頭部 2、右腕 7、左腕 7、腰 1
サイズ	270 × 393 × 665 mm
重量	約 6 kg
深度カメラ	Intel 製 RealsenseD415

### 教示対象動作：「ペンによる文字の書き取り」

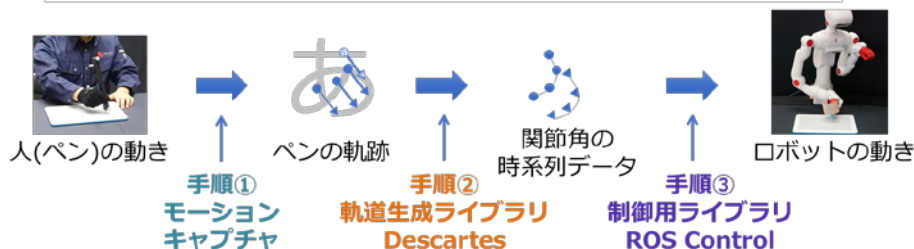


図 1 動作教示の手順および概要

表 2 モーションキャプチャの仕様<sup>7)</sup>

型式	NaturalPoint 製 Optitrack Prime 17W
解像度	1664×1088
フレームレート	30 FPS～360 FPS (調整可能)
FOV	水平 70°、垂直 49°
インタフェース	GigE/PoE+
レイテンシ	2.8 ms

最初に動作教示手順①として、モーションキャプチャを使って人が文字を書く際のペンの位置と姿勢の時系列データ(軌跡)を取得します。なお、モーションキャプチャの仕様を表 2 に、設置環境を図 2 に示します。使用するモーションキャプチャは光学式であり、8 台あるカメラで座標取得用マーカーを捉え、その 3 次元座標を取得します。軌跡を取得するにあたり、ペンにはマーカーを 2 個取り付け、さらに参考のために手の甲にもマーカーを 1 個取り付け、それらの座標を取得しました(図 3)。ペンに付けた 2 個のマーカーの時系列座標から、ペンの軌跡を計算します。計算したペンの軌跡を、ROS で取り扱うためのデータ形式に変換し、ROS で読み込みます(図 4)。

次に、アームロボットへの動作教示手順②として、読み込んだペンの軌跡から、軌道生成ライブラリ Descartes を用いて、各関節角の時系列データを生成します。

最後に、アームロボットへの動作教示手順③として、関節角の時系列データに基づいて、制御用ライブラリ ROS Control で関節を制御します。これにより、アームロボットが人の書き取った文字と同じ文字を書き取ります(図 5)。

#### 4. おわりに

ROSとその豊富なライブラリを使うことで、ロボット制御プログラムを効率的に開発できます。当研究所では、ROSによるロボット制御プログラムの開発を支援しています。ご興味のある方はお気軽にお問い合わせください。本事例は、Youtube ORISTチャンネルにて動画でご覧いただけます。



Youtube ORIST チャンネル  
<https://www.youtube.com/channel/UCdPUmbfXn7AWLJ6i0At26GA>



図 2 モーションキャプチャの設置環境

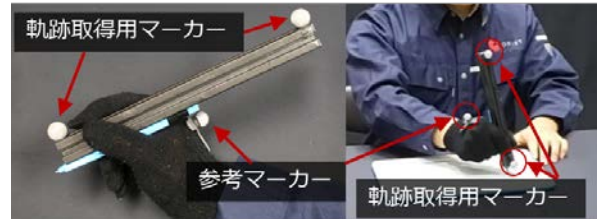


図 3 モーションキャプチャの座標取得用マーカー(左)と座標取得の様子(右)

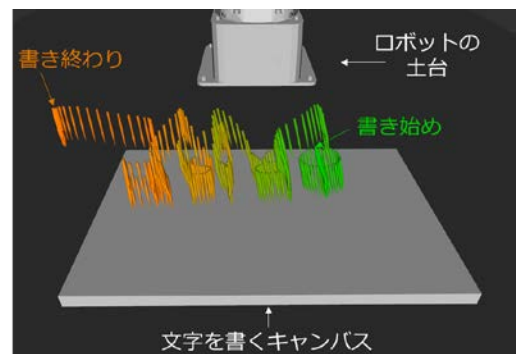


図 4 ROS のデータ形式で読み込み再現したペンの軌跡

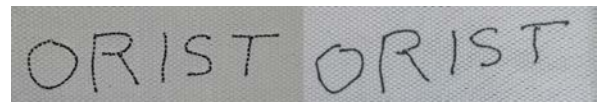


図 5 人が書き取った文字(左)とロボットが書き取った文字(右)

#### 参考文献

- 1) <https://www.ros.org/>
- 2) <https://wiki.ros.org/descartes/Tutorials/Getting%20Started%20with%20Descartes>
- 3) 大阪産業技術研究所テクニカルシート No.20-13 「ROS を用いたアームロボット制御ツール・ライブラリとその活用事例」
- 4) <https://moveit.ros.org/>
- 5) <https://industrial-training-jp.readthedocs.io/ja/latest/index.html>
- 6) <https://rt-net.jp/products/sciurus17/>
- 7) <https://optitrack.com/cameras/prime-17w/>

発行日 2021年8月16日  
 作成者 電子・機械システム研究部 知能機械研究室 赤井亮太、宮島健  
 Phone: 0725-51-2677、0725-51-2678  
 E-mail: akairyo@tri-osaka.jp、miyajima@tri-osaka.jp