

## ROS を用いたアームロボット制御 1 ツール・ライブラリとその活用事例

キーワード：ロボット、制御、ROS

### 1. はじめに

近年、ロボット制御の分野において、オープンソースによる標準化が進められています。ロボット制御のために開発されているオープンソースのプラットフォームの一つとしてROS<sup>1)</sup> (Robot Operating System) があります。ROS は Operating System と名がついていますが、Windows、macOS、Linux のような OS ではなく、Linux 系の OS 上で動くソフトウェアフレームワークです。ROS では、ロボット制御プログラムの開発に活用できる豊富なツール、ライブラリ、およびロボットシステムを構築するための通信ライブラリが提供されています。また、ROS 以外のライブラリとの連携も容易に行えます。

本稿では、ROS でアームロボットの制御を行う場合に活用できるツールやライブラリについて解説するとともに、アームロボットを ROS で制御した事例を紹介します。

### 2. アームロボットの制御に利用する

#### ROS ツール・ライブラリ

##### 2.1 ツール

###### ・rviz

ROS が提供している可視化ツールです(図 1)。ロボットの姿勢やセンサから得られた情報、更にはプログラム内で計算されたデータなど、非常に多くのデータを可視化することが可能です。例えば、アームロボットでは、把持動作を行うまでのアームの軌跡のほか、カメラから得られた情報を基に計算した把持対象の位置や姿勢なども可視化できます。また、複数のロボットやユニットを同じ座標系の仮想空間上に可視化することも可能であり、例えば、生産ライン全体の状況なども視覚的に把握できます。

###### ・Gazebo<sup>2)</sup>

ROS で標準的に使用されているシミュレータです(図 2)。ロボット外部の環境をシミュレータ上に構築することができます。これにより、様々な環境におけるカメラやセンサの働き、ロボットの挙動などを確認することが可能となります。また、後述するライブラリ ROS Control と連携することで、ロボット実機を動かすプログラムと同じものを用いてロボットの制御シミュレーションを行うことができます。

##### 2.2 ライブラリ

###### ・MoveIt<sup>3)</sup>

ROS で標準的に使用されている軌道計算ライブラリです。アーム先端の目標座標と目標姿勢を指定することで、現在の位置、姿勢から目標座標、および目標姿勢までのアーム先端の軌道を計算し、その軌道を実現する各関節角の時系列データを計算できます。また、rviz を使うことでグラフィカルに目標座標と目標姿勢を指定することも可能です。

###### ・ROS Control

アームロボットの関節の角度を制御するライブラリです。アクチュエータの状態を受け取り、PID 制御などのフィードバック制御を行うことができます。本稿で扱っているアームロボットでは、「joint trajectory controller」を用いて制御を行っています。このコントローラは、各関節角の時系列データとモーターからのフィードバック情報に基づき、複数関節のフィードバック制御を行うことができます。

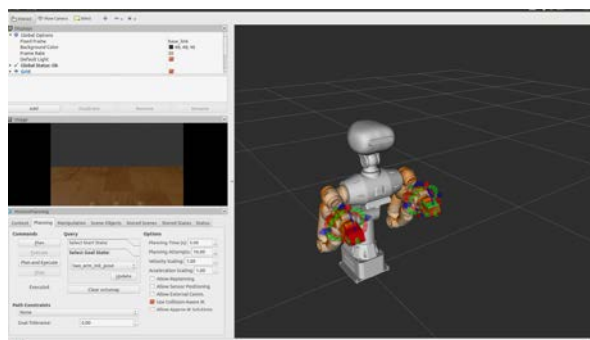


図 1 可視化ツール rviz



図 2 シミュレータ Gazebo

### 3. ROS を使ったアームロボット制御事例

本制御事例では、テーブル上のペットボトルをロボット頭部の深度カメラで認識し、ペットボトルに近い方の腕で拾い、バケツに入れる動作をアームロボットにさせました。表1にアームロボットの仕様、図3に本制御事例の概要を示します。

表1 アームロボットの仕様<sup>4)</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

最初に、頭部の深度カメラによってペットボトルの情報を点群データとして取得します。この点群データは、rviz を用いることで可視化できます(図4)。

次に、この点群データを対象とし、点群処理ライブラリ「PointCloudLibrary(PCL)<sup>5)</sup>」を用いて、高さが最も低いペットボトルをターゲットとして認識し、その座標と姿勢を計算します。

さらに、この座標と姿勢のデータを MoveIt に入力し、ロボットがターゲットを拾う姿勢になるまでの各関節角の時系列データを計算させます(図5)。

最後に、MoveIt によって計算された関節角の時系列データに基づいて、ROS Control で各関節を制御することで、実際にロボットがペットボトルを拾い、バケツに入れます(図6)。

### 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することが可能となります。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。

#### 参考文献

- 1) <https://www.ros.org/>
- 2) <http://gazebosim.org/>
- 3) <https://moveit.ros.org/>
- 4) <https://rt-net.jp/products/sciurus17/>
- 5) <https://pointclouds.org/>



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



図3 本事例の概要

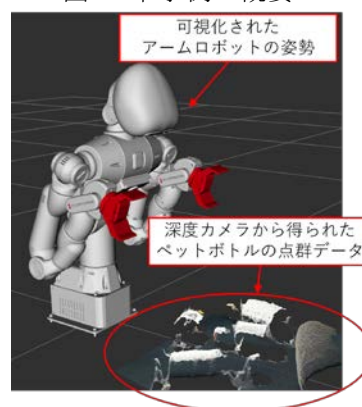


図4 rviz を用いた点群データの可視化

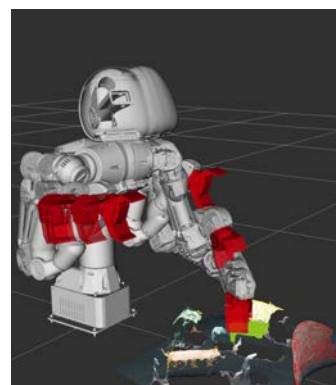


図5 PCL と MoveIt を用いた関節角の計算



図6 実機での動作の実行

発行日 2020年12月23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](mailto:akairyo@tri-osaka.jp)、[miyajima@tri-osaka.jp](mailto:miyajima@tri-osaka.jp)