

# 機械学習による産業用ロボットの自律的な動作生成に関する試験研究事業

平間 毅\* 前島 崇弘\*\* 岡田 真\*\* 若生 進一\* 青木 邦知\*\*

## 1. はじめに

政府の「ロボット新戦略」策定に伴い、茨城県においても平成26～27年度次世代自動化システムフォーラム、平成28年度中小企業IoT等自動化技術導入促進事業、平成29～30年度IoT・ロボット研究会等を通じて、県内中小企業のロボット導入を促進するための取り組みを行ってきた。

また、平成28年経済センサス活動調査産業別集計（製造業・概要版）（経済産業省）によると、茨城県は製造業における従業者数が26万1,726人で全国8位、製造品出荷額等が12兆376億円で全国8位に位置する工業県であり、ロボットを活用した自動化をニーズに持つ企業も多い。

しかし、ロボットを生産現場へ導入するためには人が工程ごとにロボットへの動作教示作業や様々なパラメータの調整を行う必要があり、短納期多品種変量生産が求められる中小企業にとっては、その技術的な難しさやシステムインテグレート（システムの設計、設置、運用）に係る費用の増加、時間の制約等からロボット導入を妨げる一つの要因となっている。

このような背景のもと、当センターでは自動化システムを構築する際の教示者負担を軽減するため、機械学習を活用した産業用ロボットの自律的な動作生成に関する研究開発に取り組んでいる。

## 2. 目的

本研究開発では、機械学習による画像認識結果をもとに、ワークや環境の状況に応じた適切なロボット動作生成を行う手法の開発を目的としており、今年度は機械学習に向けたデータ拡張やロボットの動作計画を行うためのベースとなるロボットシミュレータ上に、実環境を模擬した環境の構築を行った。特に、RGBD カメラで取得したデータをもとに、垂直多関節ロボット、ロボット用作業机及び RGBD カメラをロボットシミュレータ上にモデリングする手法について検討を行った。

## 3. 研究内容

### 3.1 実験環境及び仮想環境構築の流れ

実験環境として、垂直多関節ロボット (Torobo Arm, 東京ロボティクス株式会社)、RGBD カメラ (RealSense D415<sup>2,3)</sup>, Intel), ロボット用作業机から構成されるロボット動作環境を構築した (図 1)。Torobo Arm は全軸にトルクセンサが搭載された 7 軸型の垂直多関節ロボットであり、ロボットアプリケーションフレームワークの一つである ROS<sup>1)</sup> (Robot Operating System) に対応しているため、外部システムと柔軟に連携することができる。RealSense D415 は深度計測が可能なステレオビジョンの深度カメラであり、2 個の深度センサ、カラー画像を取得するための RGB センサ、深度データをより正確に取得するために測定対象を照らすことので

きるアクティブ IR 投射器が搭載されている。また、USB 給電で動作が可能であり、測定可能な深度距離は 30cm ～10m と幅広いレンジに対応している。ロボットシミュレータには ROS と連携可能な Gazebo<sup>4)</sup> を使用し、Gazebo 上に垂直多関節ロボット、作業机、RGBD カメラから構成される仮想環境を構築した。

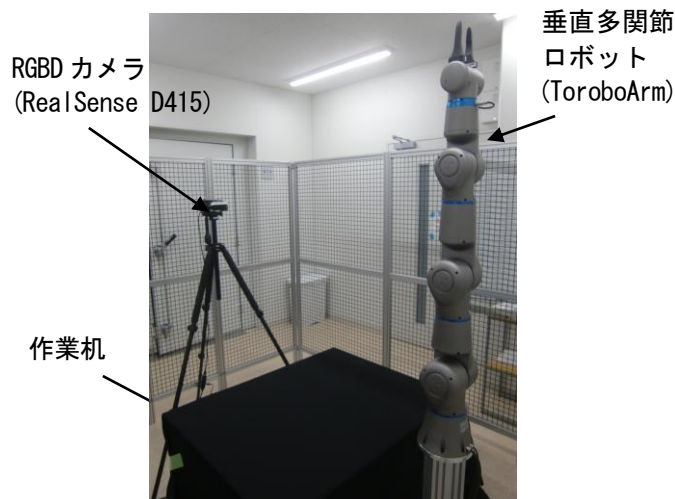


図 1 実験環境

仮想環境構築の流れを図 2 のとおり。

#### Gazebo上にロボットモデルを配置

1. ロボットハンド部にARマーカーク取付け
2. ARマーカークが撮影できる位置へロボット姿勢を制御
3. RGBDカメラで得られるカラー画像からARマーカークを認識し、ARマーカーク座標系とRGBDカメラ座標系の関係を算出
4. ロボットベース座標系とARマーカーク座標系の関係 (既知) をもとにロボットベース座標系とRGBDカメラ座標系の関係を算出

#### Gazebo上にRGBDカメラモデルを配置

1. RGBDカメラで取得した点群データから作業机を検出
2. 取得した点群データから作業机をモデリング
3. ロボットベース座標系とRGBDカメラ座標系の変換行列 (既知) をもとにロボットベース座標系と作業机座標系の関係を算出

#### Gazebo上に作業机モデルを配置

図 2 仮想環境構築の流れ

### 3.2 RGBD カメラ及び垂直多関節ロボット間の位置姿勢関係を算出

RGBD カメラのモデルを Gazebo 上に適切に配置するためには、実環境において RGBD カメラが基準となる座標系(今回はロボットベース座標系)に対してどのような位置姿勢で配置されているかを把握する必要がある。

本研究開発ではロボットハンドに取り付けた AR マーカーを RGBD カメラで撮影し(図 3)、取得したカラー画像に対して画像認識処理を行うことで AR マーカーの位置姿勢を算出した。ここで、AR マーカーの作成と認識には AR 向けライブラリ Alvar<sup>5)</sup>及びその ROS Wrapper<sup>6)</sup>で提供されている機能を利用した。

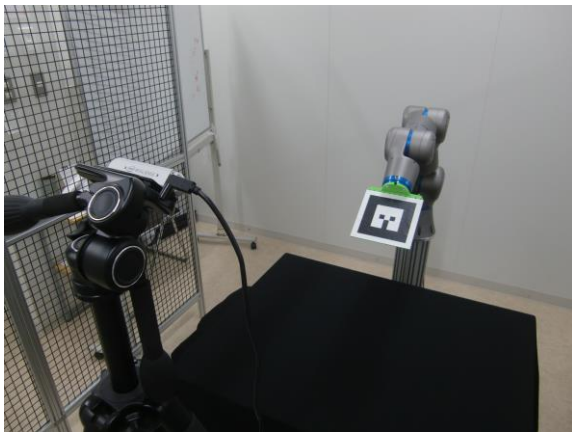


図 3 垂直多関節ロボット及び AR マーカーの配置

図 4 に AR マーカーの認識結果を示す。上段は RGBD カメラから取得した色付き点群データ、下段はカラー画像を ROS 可視化ツール Rviz で可視化した画面であり、AR マーカー座標系の認識結果が AR マーカー上に重ねて表示されている。なお、座標軸の赤、青、緑はそれぞれ x 軸、y 軸、z 軸に対応しており、上段左下の座標系は RGBD カメラ座標系を表す。

ROS では座標管理を行うパッケージ tf<sup>7)</sup>が提供されており、複数の座標系を tf 変換ツリーとして管理することができる。また、tf 変換ツリーに登録されている座標系に関しては同パッケージで提供されている Listener を用いた座標変換処理が可能になる。本研究開発で用いた垂直多関節ロボット等、ROS に対応したロボットであればロボットの各関節の座標系等、ロボットに直接関係する座標系は tf 変換ツリーとして管理されており、また AR マーカー座標系についても、今回独自に製作したロボットハンドの設計情報からロボットと AR マーカー間の位置姿勢関係が既知であるため、tf 変換ツリーへ登録することができる。更に、AR マーカーの認識結果より得られた AR マーカー座標系と RGBD カメラ座標系間の関係から RGBD カメラ座標系を tf 変換ツリーに登録することができるため、tf パッケージで提供されている Listener を用いて、ロボットベース座標系と RGBD カメラ座標系間の位置姿勢関係を求めることができる。

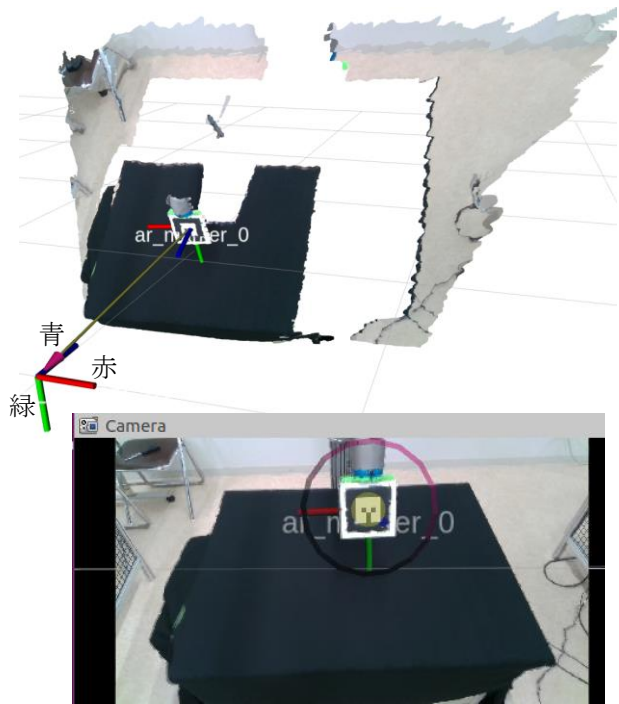


図 4 AR マーカーの認識結果  
(上段：色付き点群表示，下段：カラー画像表示)

### 3.3 作業機の検出とモデリング

作業機のモデルを Gazebo 上に配置するため、RGBD カメラで取得した色付き点群データから作業機を抽出し、モデル化を行った。

作業機の抽出にあたっては点群処理ライブラリ Point Cloud Library<sup>8)</sup>を用いて以下の処理を行った。

処理 1：

解析領域を限定するために深さ方向 2m までの点群データを抽出

処理 2：

RANSAC(random sample consensus)による平面検出

処理 3：

ユークリッド距離クラスタリングによるクラスタ化及び領域サイズによるフィルタリング(サイズ最大のクラスタを抽出)

それぞれの処理の結果を Rviz で可視化した様子を図 5 に示す。また、RGBD カメラから見た作業機の位置姿勢が既知であるため、3.2 の結果を用いてロボットベースと作業機の位置姿勢関係を求めることができる。

次に、Gazebo で読み込むことのできるモデルを作成するため、3D データ編集ソフト MeshLab<sup>9)</sup>を用い、抽出した色付き点群データから stl ファイル及びテクスチャファイルを作成した。この時、表面の再構成にはボールピボットアルゴリズムを適用した。MeshLab による色付き点群データのメッシュ化とテクスチャ生成の結果を図 6 に示す。

以上をもとに、Gazebo 向けの作業機モデルとして SDF ファイルを作成した。stl ファイル及びテクスチャファイルはそれぞれ SDF ファイル中の geometry タグ、material タグから読み込まれ、ロボットシミュレーション時の可視化や衝突検出のために用いられる。

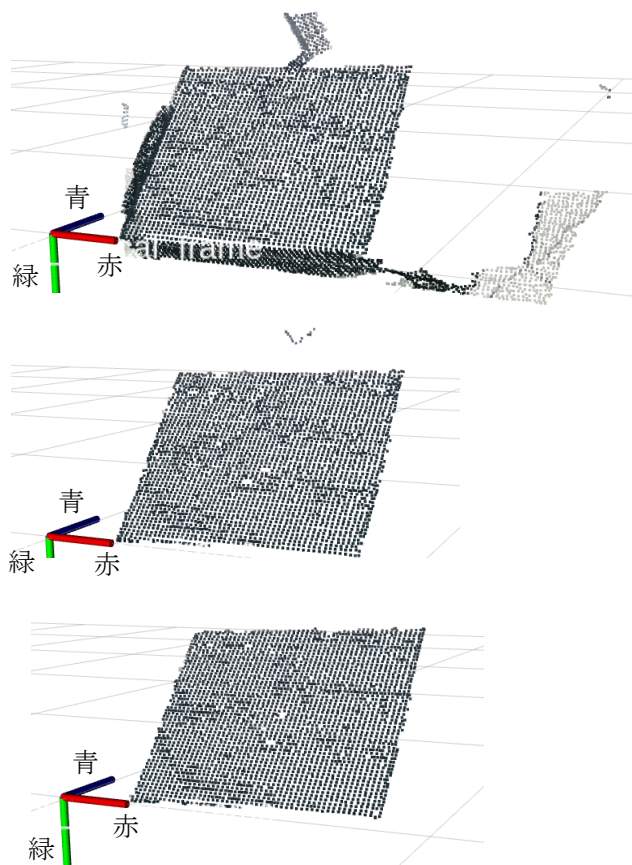


図 5 点群処理結果

(上段：処理 1，中段：処理 2，下段：処理 3)

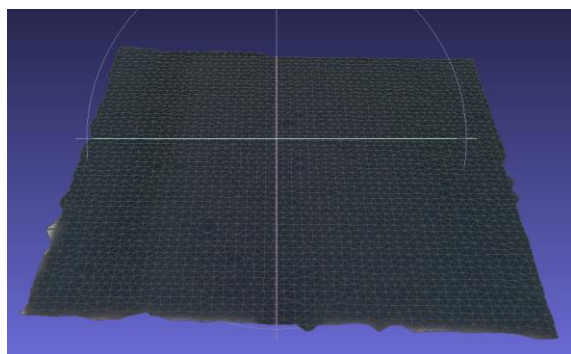


図 6 MeshLab によるメッシュ化及びテクチャ生成の結果

### 3.4 仮想環境の構築

本研究開発で使用した Gazebo はオープンソースで開発された 3D ロボットシミュレータであり，Gazebo 上に適切な環境を構築することによってロボットの動力学シミュレーションやカメラ，レーザーレンジファインダ等のセンサーシミュレーションを行うことができる。また，Gazebo そのものは ROS とは独立したソフトウェアであるが，`gazebo_ros_pkgs`<sup>10)</sup>と呼ばれるパッケージが Gazebo に ROS のインターフェースを提供しており，例えば `spawnurdfmodel` サービス，`spawnsdfmodel` サービス，`deletemodel` サービスにより，それぞれ URDF フォーマットで記述されたモデルの生成，SDF フォー

マットで記述されたモデルの生成，モデルの削除を行うことができる。

本研究開発では，URDF で記述された垂直多関節ロボットのモデルと SDF で記述された RGBD カメラ及び作業機のモデルを Gazebo 上に生成した。URDF には垂直多関節ロボットのリンク，関節，エンドエフェクタの情報が定義されており，ロボットベース座標系は Gazebo のワールド座標系に固定されている。このモデルを Gazebo 上に配置した結果を図 7 に示す。

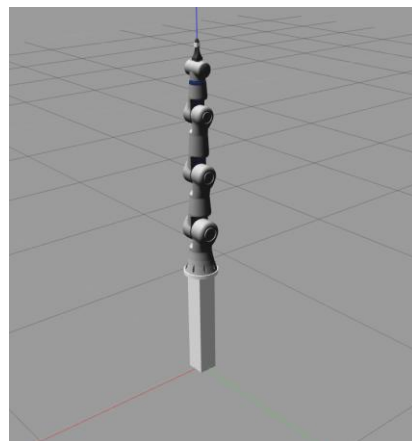


図 7 垂直多関節ロボットモデルの配置

RGBD カメラは SDF の sensor タグに `camera` 及び `depth` のタイプ指定をすることでモデル化した。ここで，`camera` は RGBD カメラのカラーカメラ部，`depth` は深度センサ部をそれぞれモデル化している。また，sensor タグ内に `noise` タグを加えることによって，このモデルから取得できる画像や深度データに実環境のようなノイズを付加することもできる。なお，この RGBD カメラモデルは画像データや深度データを ROS のトピックとして配信するため，他ノードからデータを購読することでデータに対する解析や処理を行うこともできる。RGBD カメラモデルを 3.2 で算出したロボットベースと RGBD カメラの位置姿勢関係をもとに Gazebo 上へ配置した結果を図 8 に示す。図中，左側の白い物体が RGBD カメラを表す。

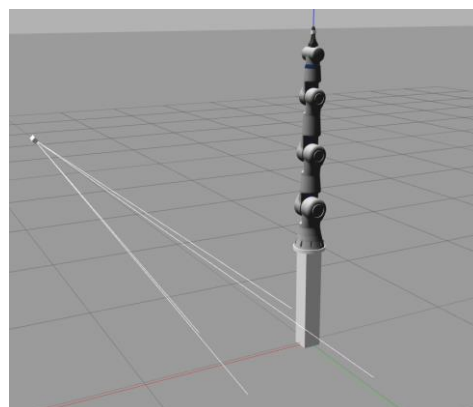


図 8 RGBD カメラモデルの配置

作業機には 3.3 で作成したモデルを用い，また 3.3 で算出したロボットベースと作業機の位置姿勢関係をもとに Gazebo 上へ配置した。作業機モデル配置後の結果を図 9 に示す。

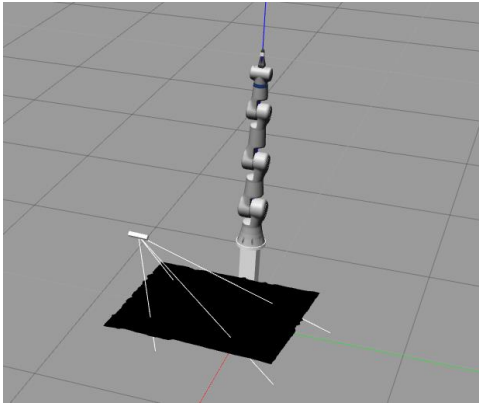


図9 作業機モデルの配置

#### 4. 研究結果と考察

- ARマーカー及び画像認識技術を用いることでRGBDカメラ及び垂直多関節ロボット間の位置姿勢推定を行うことができた。今回は、垂直多関節ロボットの位置姿勢推定を目的としたが、対象物にARマーカーを貼り付けることができ、またカメラで撮影することができれば対象物の位置姿勢推定が可能であるため、例えば作業ワークの認識等にも用いることができる。
- RGBDカメラで取得した色付き点群に対して点群処理を行うことで作業機の抽出とモデル化を行うことができた。今回は平面検出を行うことで作業機の抽出を行ったが、複雑な形状を持つ対象物には対応できない。その場合は、予め準備したモデルに対しての局所特徴量ベースのマッチングや細かな位置合わせのためのICP(Iterative Closest Point)アルゴリズム等を検討する必要があると考えられる。
- 本研究開発ではロボットシミュレータGazebo上にロボット動作環境をモデリングした。Gazebo上ではロボットの動力学シミュレーションだけでなく、仮想環境上で画像等のセンサデータが併せて取得できるため、これらを機械学習用データの一部として活用できると考えられる。

#### 5. まとめ

- RGBDカメラから取得した画像から垂直多関節ロボット先端に取り付けたARマーカーの位置姿勢を認識することで、ロボットシミュレータGazebo上に垂直多関節ロボット及びRGBDカメラのモデルを実環境に合わせて配置することができた。
- RGBDカメラから取得した点群データから点群処理により作業機を認識することで、作業機のモデリング及びロボットシミュレータGazebo上への実環境に合わせた配置を行うことができた。

#### 6. 今後の課題

- 機械学習手法によるワークの認識
- ワーク位置に応じた垂直多関節ロボットの動作計画
- はめあい作業を想定した動作生成及び垂直多関節ロボット実機による動作検証

#### 7. 謝辞

本研究開発を進めるにあたりご助言いただきました(国立研究開発法人)産業技術総合研究所知能システム研究部門マニピュレーション研究グループの喜多 泰代様及び山野辺 夏樹様に感謝いたします。

#### 8. 参考文献等

- 1) <http://www.ros.org/>
- 2) <https://click.intel.com/intelr-realsensetm-depth-camera-d415.html>
- 3) <https://github.com/intel-ros/realsense/releases>
- 4) <http://gazebo-sim.org/>
- 5) <http://virtual.vtt.fi/virtual/proj2/multimedia/alvar/>
- 6) [http://wiki.ros.org/ar\\_track\\_alvar](http://wiki.ros.org/ar_track_alvar)
- 7) <http://wiki.ros.org/tf>
- 8) <http://pointclouds.org/>
- 9) <http://www.meshlab.net/>
- 10) [http://wiki.ros.org/gazebo\\_ros\\_pkgs](http://wiki.ros.org/gazebo_ros_pkgs)