

# 移動作業ロボットのセンシング情報補完技術の開発

鈴木 慎一, 浦池 隆文, 今岡 広一

## Development of Method to Achieve Work by Supplementing Sensing Results for Mobile Work Robots

Shinichi SUZUKI, Takafumi URAIKE, Kouichi IMAOKA

キーワード：労働力不足，移動作業ロボット，ROS，人協働

### 1. はじめに

近年，ロボット技術やICTを活用したスマート農業を推進して労働力不足への対応や精密農業，作業の均一化に基づく農産物の品質向上などの研究が行われ，農作業ロボットの研究では，完全自動作業を目指した開発が進められている<sup>1)</sup>。しかし，屋外環境においては，日中の時間・天候による日射の状態，茎葉の繁茂や作業路の状態変化などによってセンシング環境が大きく異なるため，計測および情報処理後の結果から得られる情報（センシング情報）において検出漏れや誤検出を無くすることが難しい。そこで，ロボットのセンシング情報の誤りを人（作業管理者）の高度な認識能力を用いて補完することにより，ロボット作業の確実性向上が望まれている。

本研究では，汎用ロボット用ミドルウェアとしてROS（Robot Operating System）を搭載した小型移動作業ロボットシステムを用いて，ロボットがセンシング情報の補完情報を受け取り，選択ピックアップ作業を達成することを確認した。

### 2. センシング情報の補完

様々な分野において，人手作業を代替する自律型ロボットの活用範囲は広く，多数の研究が進められているが，作業対象の誤検出や検出漏れがあり実用化が進められていない。これは，屋外環境における計測条件の多様性に基づく計測精度の低下を起因とするところも少なくない。そこで，ロボットのセンシング情報を作業管理者に提示し，管理者がセンシング情報を補完することで作業の確実性向上が図られることが有効と考えた。図1にセンシング情報の補完作業イメージを

示す。ここで，カメラを搭載したロボットが，球形や四角形の物体が複数ある中を移動しながら，画像センシングを行い球形の物体を認識し，ピックアップ作業を行う。

このとき，ロボットと人が協調して作業する流れを以下に示す。



図1 センシング情報補完作業のイメージ

1. ロボットはセンシング情報を作業管理者の携帯端末に提示する。
2. 作業管理者は，センシング情報から誤検出や未検出を判断して収集作業対象の追加，削除を行うことでセンシング情報を補完し，その補完情報をロボットに送信する。図1では，模様をついた球体を作業する対象から削除（キャンセル）し，円柱の陰にある球体を追加している。
3. ロボットは補完されたセンシング情報に従ってピックアップ作業を行う。

### 3. 移動作業ロボットの製作と機能検証試験

#### 3.1 ROS（Robot Operating System）の概要

移動作業ロボットの試作にあたり，制御プログラムの開発にROS<sup>2)</sup>を用いた。ROSは，ロボット用アプリケーションプログラム作成を支援するライブラリとツールを提供している。

事業名：経常研究

課題名：移動作業ロボットのセンシング情報補完技術の開発

具体的には、a) ハードウェア抽象化, b) デバイスドライバ, c) ライブラリ, d) 視覚化ツール, e) メッセージ通信, f) パッケージ管理などの機能が提供される。

ROSの特徴<sup>3)</sup>を以下に示す。

- ①分散処理：ノードと呼ばれる最小単位のプロセス（実行単位）が独立して並列的に動作し、双方向のデータ送受信を行う。
- ②パッケージ管理：各ノードは、同一の目的で作成された複数のノードをパッケージとしてまとめて管理するため、パッケージの共有、修正、再配布が容易である。
- ③公開リポジトリ：開発されたパッケージはGitHub（プログラムバージョン管理サービス）等の開発者のリポジトリを通して公開される。
- ④APIの整備：ROSとは無関係に開発されたプログラムもROSで提供されるAPIを用いることにより簡単にROSから利用可能となる。
- ⑤多数のプログラミング言語への対応：Python, C++, Lisp, Java, C# 等に対応。

このような特徴を持つため、国内外のロボット研究現場ではROSを用いた研究・開発が盛んである。また、ROSはOSと呼ばれているが一般的OSのLinux上でロボットの機能を提供するためのミドルウェアにあたる。ROSはオープンソースのBSDライセンスによってライセンス化されているため、著作権の表示・免責条項を明記することで再利用・再配布が自由となる。今回使用したROS、OSのバージョンとソフトウェア開発言語を表1に示す。また、ROSのスタック構成<sup>4)</sup>を図2に示す。

表1 ROS、OSのバージョン

ROS	Kinetic Kame
OS (Linux)	Ubuntu (16.04 LTS)
開発言語	C++

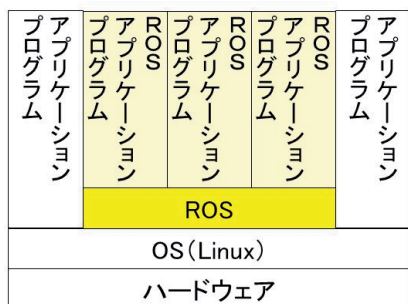


図2 ROSのスタック構成

### 3.2 試作した移動作業ロボットの概要

本研究で試作した小型移動作業ロボットを図3に示す。このロボットは、デブスカメラ2台と移動ロボット、アームロボット、ロボット制御パソコン各1台で構成した。各ロボット、カメラの主な仕様<sup>5)</sup>を表2に示す。

ロボットの移動部として使用した移動ロボット (YujinRobot

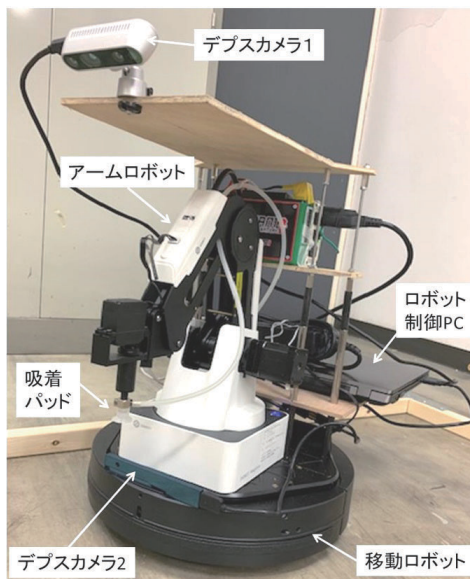


図3 試作した移動作業ロボット

表2 ロボット、カメラの仕様

移動ロボット (Kobuki)	
最大速度	70cm/s
最大回転速度	180deg/s
稼働時間	3時間
寸法(直径:D, 高さ:H)	D:351.5mm×H:124.8mm
アームロボット (DobotMagician)	
ペイロード	500g
マックスリーチ	320mm
位置再現性	0.2mm
ベース寸法	158mm×158mm
デブスカメラ (R200)	
動作範囲(最小 - 最大)	0.5m - 3.5m
深度センサ視野角	H: 59°, V: 46°, D: 70°
寸法	101.6mm × 9.6mm × 3.8mm
デブスカメラ (D435i)	
動作範囲(最小 - 最大)	0.2m - 10m
深度センサ視野角	H: 87°, V: 58°, D: 94°
寸法	90mm × 25mm × 25mm

製Kobuki)は、対向2輪型(2輪駆動1キャスト)の移動ロボットであり、開発のプラットフォームとして国内外の大学のロボット研究室などで広く利用され、豊富なROSアプリケーションプログラムが公開されている。

また、作業用に使用したアームロボット (Shenzhen Yuejiang Technology製DobotMagician)は、垂直多関節型(4軸)で、搬送や組み立て作業等で使用される産業用ロボットとして広く普及しているタイプのロボットである。センサとして使用したデブスカメラ (Intel製R200, D435i)は、画像と距離情報を出力する。これらロボットとカメラは、

ROSに対応している。

### 3.3 移動作業ロボットの制御

移動作業ロボットの構成要素であるロボット、カメラはロボット制御用のパソコン（CPU：Intel製corei5）を中心にシリアル通信（USB）で接続した（図4）。また、作業管理者がセンサのデータ確認、補完指示を行う端末機器としてノートパソコン（CPU：Intel製corei5）を用い、ロボット制御用と同じOS（Linux）を搭載した。ロボットとはSSH（SecureShell）というTCP/IPネットワークを利用する通信プロトコルを用いて無線通信を行った。そのため、通信の応答性は早くなく、用途に応じては改善が必要である。

ROS上でアームロボット、移動ロボット、デプスカメラ、画像処理の制御プログラムをパッケージとして作成し、各々が独立して動作する（図5）。また、デプスカメラからの画像・距離情報は、OpenCVという画像処理用フリーウェアを用いて描画や特徴点抽出などの処理によって対象物の認識を行った。

次にROSプログラムによる通信動作を、デプスカメラのセンシング情報を基にしたアームロボットの動作を例に示す（図6）。デプスカメラ1のパッケージには画像取得、距離情報送信などのノードがあり、ノード間では画像・距離情報がメッセージとして通信される。画像処理パッケージには、距離画像描画や特徴点抽出、座標送信などのノードがある。アーム

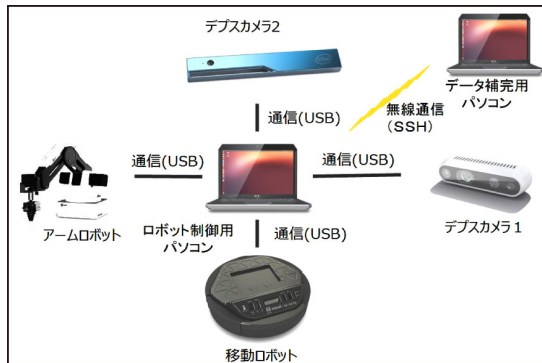


図4 移動作業ロボットの構成要素の接続

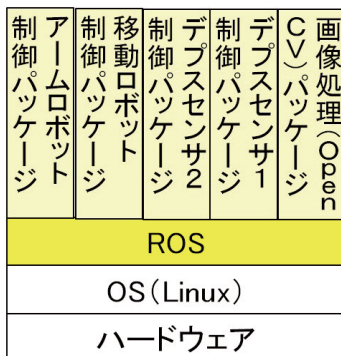


図5 ロボット制御プログラムの構成

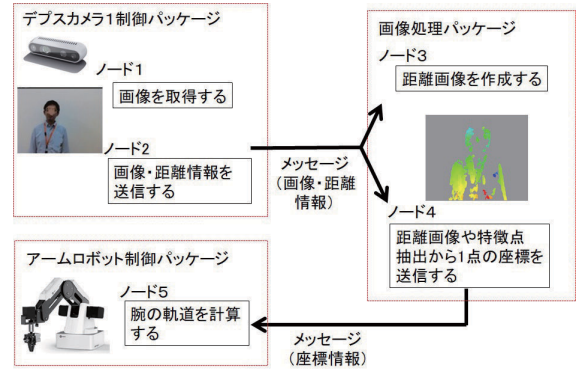


図6 移動作業ロボットの通信動作

ムロボット制御パッケージでは、指定された座標値へアーム先端を動かすノードがあり、画像処理ノードからのメッセージによって受け取ったデプスカメラの計測位置へアームロボットの動作を行う。

### 3.4 機能検証試験

この試作ロボットを用いてセンシング情報補完の機能検証試験を行った。検証用の目的作業としてボールのピックアップ作業を行った。ピックアップ対象のボールは卓球のボール（直径30mm）を用いて、そのボールが高さ165mmのスタンド上に1つ配置される。この組み合わせが複数存在する設定とした（図7）。

作業管理者とロボットは、無線通信を行って情報の送受信を行う。また、ロボットへは、事前情報としてピックアップの最低回数と作業終了位置（ゴール位置）を与えることとした。

機能検証試験の動作は以下とした（図8）。

- ① 作業管理者からロボットへ動作開始指令を発信。
- ② ロボットは、カメラ画像からピックアップ対象のボールをセンシングしてその結果を作業管理者へ発信。
- ③ 作業管理者は、センシング結果を確認して、ピックアップ対象の追加・削除等の補完情報をロボットへ送信する。図9は、3つあるボールの一番奥のボール検出を失敗していたため、管理者がその場所をマウスカーソルで指定している。
- ④ ロボットは受信した補完情報を基に再センシングを行う。再センシングでは、検出の基準値の変更や検出手法の変更などを行い、改めてピックアップ対象を認識するよう試みる。今回は、この再センシングの回数を1回とした。再センシングでも認識ができなかった場合、その結果を作業管理者へ報知するとともに作業管理者から指定された指示位置から動作座標を計算して追加した。
- ⑤ ロボットの初期センシング結果、再センシング結果、補正情報から経路計画を行い、ロボットは移動・ピックアップ作業を実施する（図10）。

⑥ ロボットはピックアップの回数とゴール位置の判定から動作の終了判定を行う。

この①～⑥を実行するノードを作成して統合化し、全体システムの動作が実現できることを確認した。



図7 機能試験の作業対象

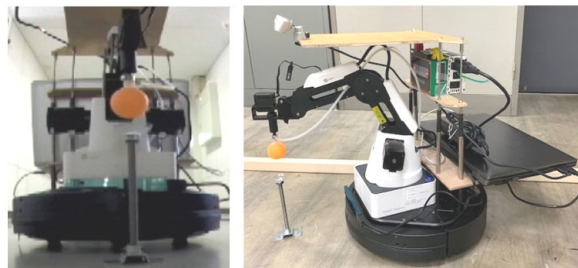


図10 ピックアップ作業の様子

#### 4. まとめ

ロボットのセンシング情報を作業管理者に提示して、センシングの検出漏れ・誤検出を補完することでロボットが作業を達成するセンシング情報補完技術をROS上で開発し、ロボットを用いた機能検証を行った。

試作した小型移動作業ロボットを用いてボールのピックアップ作業を行い、補正情報を与えて移動ロボットの作業が達成できることを確認した。

今後は、センシング情報補完技術の実用化に向けてロボットに搭載可能な組込み型制御システムの開発とユーザビリティの向上を図り、農場などにおいて早期の自動化が難しい果実の収穫作業や株間の除草作業等の省力化を行うロボットへの展開を図る。

#### 引用文献

- 1) 農林水産省 スマート農業の実現に向けた研究会  
[http://www.maff.go.jp/j/kanbo/kihyo03/gityo/g\\_smart\\_nougyo/pdf/b02\\_roadmap\\_an.pdf](http://www.maff.go.jp/j/kanbo/kihyo03/gityo/g_smart_nougyo/pdf/b02_roadmap_an.pdf)
- 2) ROS wiki <http://wiki.ros.org/ja>
- 3) 銭飛：ROSプログラミング，森北出版，3PP，(2016)
- 4) 西田健他：実用ロボット開発のためのROSプログラミング，2PP，(2018)
- 5) kobuki <http://kobuki.yujinrobot.com/>
- 6) Dobot <https://www.dobot.cc/dobot-magician/specification.html>
- 7) Intel製品仕様  
<https://ark.intel.com/content/www/jp/ja/ark/products/92256/intel-realsense-camera-r200.html>  
<https://ark.intel.com/content/www/jp/ja/ark/products/190004/intel-realsense-depth-camera-d435i.html>

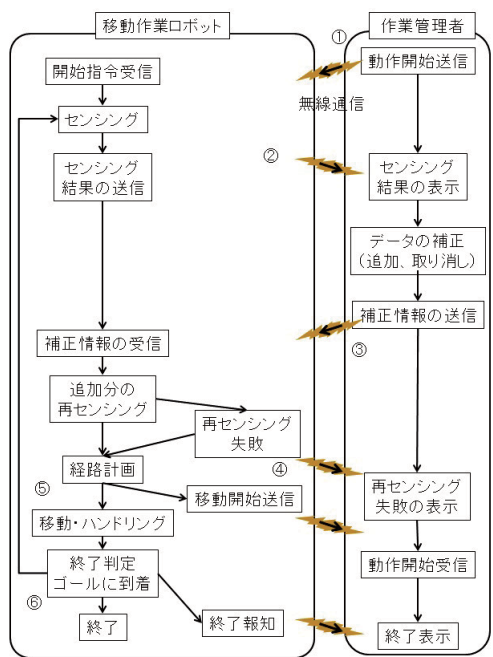


図8 機能検証試験の流れ

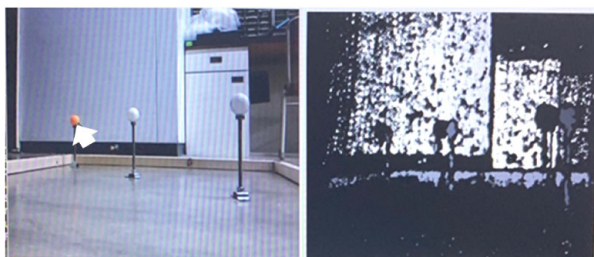


図9 センサ情報の補完指示  
 左：カーソルによる作業対象の指示，  
 右：センシング (R200) 結果 (距離画像)