

# ジェスチャーを用いたスカラーロボットの制御に関する研究

A study on control of scalar robot using hand gestures

何 沐キン<sup>1)</sup>

指導教員 林 誠治<sup>1)</sup>

1) 拓殖大学工学部 機械・電子システム工学専攻 林研究室

キーワード：スカラーロボット，ジェスチャー，遠隔制御，Kinect V2

## 1. はじめに

最近コロナの影響で新しい生活様式が求められ、一方で感染防止のために人との接触を 8 割減らすことが重要であるとの認識さえ出された。しかしながら経済活動を回すため、職場や受付業務においては人と人が密接にならざるを得ず感染リスクが高まる傾向にある。そこで本研究では人間同士の接触の機会を少しでも減らし、人間の手のジェスチャーを用いてロボットアームを遠隔制御する一つの可能性を模索することを目的とする。ハンドジェスチャーをロボットと連動させる先行研究としては、手指に取り付けた複数のモーションセンサーの値を取得することにより、それらの変動成分をロボットアームの動作に反映させるものがある[1]。しかしながらコロナ禍の状況において、操作する側でもセンサー類の取り付け取り外し時に感染リスクを高める危険性も否定できない。このことから非接触でハンドジェスチャーを取得できる Kinect は非常に有効的なデバイスであるといえる。

## 2 Kinect V2 とスカラーロボット VS-ASR

### 2.1 Kinect V2

Microsoft 社 Kinect V2 は RGB カメラのほか対象物体との距離を測定するための 3D 深度センサー、赤外線センサーを備えており、(図 1)に示すような人間の骨格情報(スケルトンデータ)を取得することが可能である。

### 2.2 Kinect V2 の各種座標系と関節の位置

Kinect V2 は RGB カラーカメラ、深度センサー、赤外線センサーを備えており、座標系はカラー座標系、Depth 座標系、カメラ座標系の 3 種類に分けられる。カラー画像はカラー座標系を、深度画像は Depth 座標系を使用し、2 つの座標系はどちらも 2 次元データである。また、画面の左上を原点とし、+X を右、+Y を下にして、単位はピクセルで表す。カメラ座標系はセンサーを原点とする 3 次元空間座標系(単位はメートル)で、本研究での人間の骨格追跡に必要な座標系

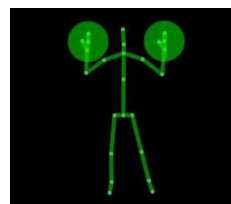


図 1: 骨格情報取得例

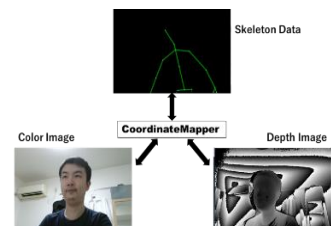


図 2: Kinect における 3 つの座標系の関係図

である。この 3 つの座標系の関係を表した具体的な例を(図 2)に示す。カラー画像と深度画像の解像度が異なるため、どちらかの座標系に正しく位置合わせを行う必要がある。そのための変換ユーティリティが CoordinateMapper である。

### 2.3 スカラーロボット VS-ASR

本研究で使用するロボットは、V-stone 株式会社より発売されているスカラーロボット VS-ASR のペンタイプである(図 3)。スカラーロボットとは、水平方向にアームが動作するロボットであり、水平方向の動きに特化したロボットとなっているが、先端には Z 軸のモータを持っており上下への押し込みなどの動作が可能である。VS-ASR は PC と USB 接続され、2通りの制御方法に対応している。1 つはシミュレーター機能が搭載されたモーションエディタソフトウェア「SCARA Programmer」(図 4)であり、ロボットの座標や関節角度の入力、ポーズ読み取り機能などを利用する教示を行うことができる。もう 1 つは C 言語に

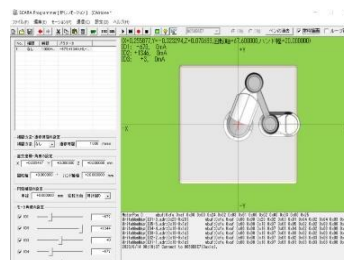


図 3 VS-ASR 本体



図 4 Scara Programmer

よるロボットプログラミング環境であり、専用の SDK を使用することでロボットを制御することができる。

### 3. 本研究におけるプログラム開発

表 1 プログラム開発環境

|               |                            |
|---------------|----------------------------|
| PC および OS     | Windows10                  |
| Kinect SDK    | Kinect for Windows SDK 2.0 |
| Comm. Library | CP2110-USB to URAT         |
| モーションエディタ     | SCARA Programmer           |

本研究では、ハンドジェスチャーとして右手の手のひらの追跡を考え、その移動距離に応じてロボットアームを動作させるプログラムを作成した。

本研究におけるプログラム開発環境を表 1 に示す。

#### 3.1 マルチスレッドプロセス

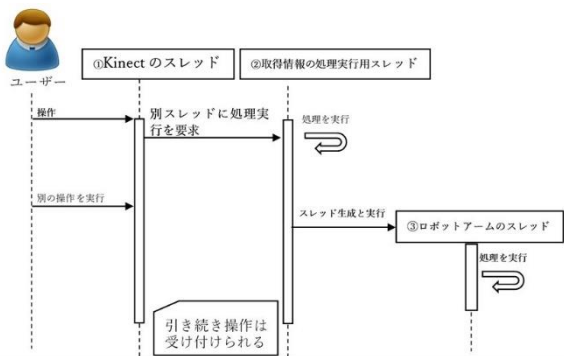


図 5 マルチスレッドプロセスの構成

本研究のプログラムはマルチスレッドのプロセスで動作させる。マルチスレッドとは、1つのコンピュータプログラムを実行する際に複数の処理の流れを並行して進められる構造を持つ。図 5 に実際のスレッド間の対応を示す。①のプロセスは主に Kinect を通じて骨格の座標を収集・加工することであり、動画フレーム毎に処理が行われる。①から収集された座標は②に送られ、CoordinateMapper を通じてロボット座標系への変換計算とロボットアームへ動作命令を発行する。③では②からの指示に従ってロボットアームを動作させる。以上のプロセスを非同期に行っている。

#### 3.2 フローチャート

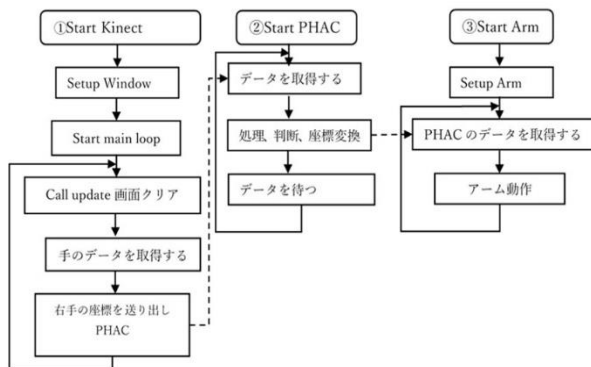


図 6 フローチャート

本研究のフローチャートを図 6 に示す。図 6 の①②③は、図 5 の①②③と一対一に対応している。ここで、

PHAC とは Process hand and coordinate の略である。

### 4. 実行結果と考察

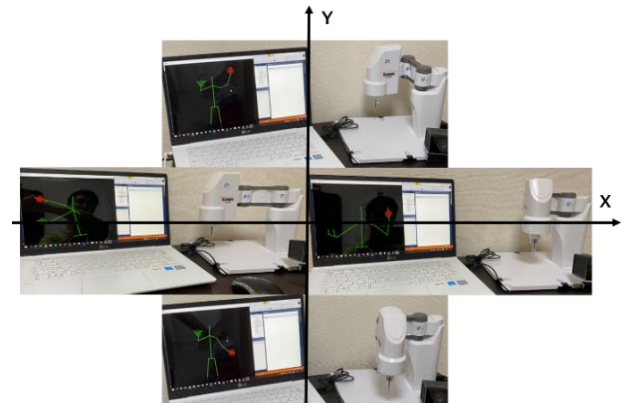


図 7 実行結果

右手の手のひらを追跡するハンドジェスチャー(左右上下への移動)に応じて、ロボットアームを 2 次元空間 x-y 座標系上で移動させることができた。図 7 に示すように、手のひらの右(左)の動きがロボットの x 軸の正方向(負方向)、上(下)の動きが y 軸の正方向(負方向)に対応している。しかしながら、Kinect の各種センサーから得られるフレームレートはロボットアームの移動速度よりも非常に高速であるため、図 6 の③でアーム操作命令のバッファオーバーフローが発生し、ロボットアームが正常に動作しないという問題がみられた。これはロボットアームの移動時間に応じて、次の移動座標を送信するタイミングを適切に調整することで解決することができると考えられる。

#### 5.まとめ

本研究では Kinect V2 を用いて右手の位置をハンドジェスチャーとして認識し、ロボット座標系へのマッピング計算と VS-ASR への動作命令を発行することで、手の左右上下の動きに応じたロボットアームの動作を実現できた。今後は、ロボット命令発行のタイミング調整とハンドジェスチャーとロボットのより自然な動作の確立、および人間の連続の動きを記録して再現する機能を追加する予定である。

#### 参考文献

- [1] 佐藤永梨花, 山崎千裕, 奥村康行, “筋電位で制御されるアームロボットの試作”, <http://www.st.nanzan-u.ac.jp/info/gr-thesis/2017/okumura/pdf/14sc064.pdf>
- [2] Vstone 株式会社, [http://www.vstone.co.jp/products/scara\\_robot](http://www.vstone.co.jp/products/scara_robot)
- [3] 中村薫, 杉浦司, 高田智宏, 上田智章, “KINECT for Windows SDK プログラミング”, 秀和システム, 2015 年