

SEED-Noid における双腕作業のための RTC 群

名城大学メカトロニクス工学科
ロボットシステムデザイン研究室

2018 年 12 月 12 日

目次

| | |
|--|------------------------|
| 1. はじめに..... | 4 |
| 1.1. コンポーネントの概要..... | 4 |
| 1.2. 関連文書..... | 4 |
| 1.3. 開発環境..... | 4 |
| 2. RTC 仕様..... | 5 |
| 2.1. SeedUpperBody..... | 5 |
| 2.2. DualArmController..... | 6 |
| 2.3. DualArmSimulation..... | 7 |
| 2.4. SEED-Noid_Bridge_command_r/ SEED-Noid_Bridge_command_l..... | 8 |
| 2.5. SEED-Noid_Bridge_state_r/ SEED-Noid_Bridge_state_l..... | 9 |
| 3. 実機制御方法..... | 10 |
| 3.1. 動作確認環境..... | 10 |
| 3.2. 環境構築..... | 10 |
| 3.2.1. Boost ライブラリのインストール..... | 10 |
| 3.2.2. SEED-Noid_Dual-Arm_pkg のダウンロード..... | 10 |
| 3.2.3. SEED-Noid_Dual-Arm_pkg のビルド..... | 10 |
| 3.3. 操作方法..... | 11 |
| 3.3.1. RTC 群の起動およびアクティベート..... | 11 |
| 3.3.2. DualArmController の操作..... | 11 |
| 4. シミュレーション制御方法..... | 11 |
| 4.1. 動作確認環境..... | 11 |
| 4.2. 環境構築..... | 11 |
| 4.2.1. ROS の環境構築..... | 11 |
| 4.2.2. aero-ros-pkg のダウンロード..... | 12 |
| 4.2.3. eSEAT のインストール..... | 12 |
| 4.2.4. SEED-Noid_Dual-Arm_pkg のダウンロード..... | 12 |
| 4.2.5. SEED-Noid_Dual-Arm_pkg のビルド..... | 12 |
| 4.3. 操作方法..... | 12 |
| 4.3.1. RTC 群の起動及びアクティベート..... | 12 |
| 4.3.2. DualArmController の操作..... | 13 |
| 4.3.3. Gazebo 実行画面..... | エラー! ブックマークが定義されていません。 |
| 5. DualArmController について..... | 13 |
| 5.1. DualArmController 実行画面..... | 13 |
| 5.2. DualArmController が有する機能..... | 14 |

| | |
|----------------|----|
| 5.3. 座標系 | 15 |
|----------------|----|

1. はじめに

1.1. コンポーネントの概要

本 RTC 群は，THK（株）が開発した双腕ロボット SEED-Noid における双腕作業のための RTC 群であり，サンプルコントローラから，実機およびシミュレーション上の SEED-Noid を動かすことが可能である．

1.2. 関連文書

関連する文書を以下に示す．

| No, | 文書名 |
|-----|--|
| 1 | ロボットアーム制御機能共通インタフェース仕様書（第 1.0 版） https://www.sec.co.jp/robot/_downloads/interface_arm_1.0.pdf |
| 2 | 双腕ロボット制御機能共通インタフェース仕様書（第 1.0 版） https://www.sec.co.jp/robot/_downloads/interface_doublearm_1.0.pdf |
| 3 | 双腕ロボット制御機能共通インタフェース 2.0 仕様書 |

1.3. 開発環境


開発環境を以下に示す．

| | |
|--------|----------------------------|
| OS | Ubuntu16.04 |
| RTM | OpenRTM-aist-1.1.2-RELEASE |
| ROS | ROS kinetic |
| Gazebo | Gazebo7 |
| eSEAT | eSEAT2.5 |

2. RTC 仕様

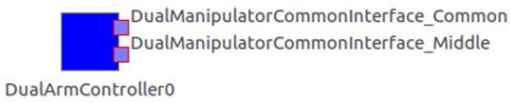
2.1. SeedUpperBody

SeedUpperBody は実機 SEED-Noid の双腕制御用コンポーネントである.

| RTC の名称 | | | |
|---------------------------------------|---|--|--------------|
| SeedUpperBody |  SeedUpperBody0 | | |
| サービスポート（コンシューマ） | | | |
| 名称 | インタフェース型 | 説明 | |
| DualManipulatorCommonInterface_Common | DualManipulatorCommonInterface_Common | 双腕共通コマンドインタフェース | |
| DualManipulatorCommonInterface_Middle | DualManipulatorCommonInterface_Middle | 双腕モーションコマンドインタフェース | |
| コンフィグレーションパラメータ | | | |
| 名称 | データ型 | デフォルト値 | 説明 |
| port_name | string | /dev/serial/by-id/usb-FTDI_TTL232R-3V3_FT98HKZC-if00-port0 | シリアル通信ポートの指定 |


2.2. DualArmController

DualArmController は双腕共通 I/F を用いたサンプルコントローラである.

| RTC の名称 | | |
|---------------------------------------|---------------------------------------|--|
| DualArmController | |  |
| サービスポート (プロバイダ) | | |
| 名称 | インタフェース型 | 説明 |
| DualManipulatorCommonInterface_Common | DualManipulatorCommonInterface_Common | 双腕共通コマンドインタフェース |
| DualManipulatorCommonInterface_Middle | DualManipulatorCommonInterface_Middle | 双腕モーションコマンドインタフェース |


2.3. DualArmSimulation

DualArmSimulation はシミュレーション上の SEED-Noid の双腕を制御するためのメインのコンポーネントである.

| RTC の名称 | | |
|---------------------------------------|--|---------------------|
| DualArmSimulation |  | |
| サービスポート（プロバイダ） | | |
| 名称 | インタフェース型 | 説明 |
| DualManipulatorCommonInterface_Common | DualManipulatorCommonInterface_Common | 双腕共通コマンドインタフェース |
| DualManipulatorCommonInterface_Middle | DualManipulatorCommonInterface_Middle | 双腕モーションコマンドインターフェース |
| データポート（入力） | | |
| 名称 | データ型 | 説明 |
| r_JointAngle_in | TimedFloatSeq | 右腕関節角度の現在値 入力 |
| l_JointAngle_in | TimedFloatSeq | 左腕関節角度の現在値 入力 |
| データポート（出力） | | |
| 名称 | データ型 | 説明 |
| r_JointAngle_out | TimedFloatSeq | 右腕目標関節角度の出力 |
| l_JointAngle_out | TimedFloatSeq | 左腕目標関節角度の出力 |

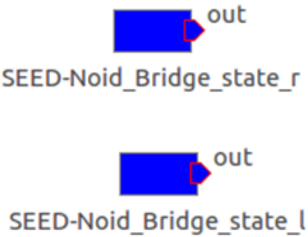
2.4. SEED-Noid_Bridge_command_r/ SEED-Noid_Bridge_command_l

SEED-Noid_Bridge_command_r および SEED-Noid_Bridge_command_l は，ROS と連携した Gazebo 上のモデルを動かすために，ROS 側に目標関節角度を送信するための ROS と RTM のブリッジである．なお，この RTC は SEATML ファイルで記述されており，eSEAT を用いて動作するようになっている．

| RTC の名称 | | |
|--|---------------------------------|--|
| SEED-Noid_Bridge_command_r SEED-Noid_Bridge_command_l | |  |
| データポート（入力） | | |
| 名称 | インタフェース型 | 説明 |
| in | TimedFloatSeq | 目標関節角度の入力 |
| ROS トピック通信（パブリッシャ） | | |
| 名称 | メッセージ | 説明 |
| /rarm_controller/command /larm_controller/command | Trajectory_msgs/JointTrajectory | 目標関節角度の出力 |

2.5. SEED-Noid_Bridge_state_r/ SEED-Noid_Bridge_state_l

SEED-Noid_Bridge_state_r および SEED-Noid_Bridge_state_l は，ROS と連携した Gazebo 上のモデルの関節角度現在値を，RTM 側に送信するための ROS と RTM のブリッジである．なお，この RTC は SEATML ファイルで記述されており，eSEAT を用いて動作するようになっている．

| RTC の名称 | | |
|--|---|--|
| SEED-Noid_Bridge_state_r SEED-Noid_Bridge_state_l | |  |
| データポート（出力） | | |
| 名称 | インタフェース型 | 説明 |
| out | TimedFloatSeq | 関節角度現在値の出力 |
| ROS トピック通信（サブスクリイバ） | | |
| 名称 | メッセージ | 説明 |
| /rarm_controller/state /larm_controller/state | control_msgs/JointTrajectory ControllerState | 関節角度現在値の入力 |

3. 実機制御方法

3.1. 動作確認環境

本 RTC 群の動作確認環境を以下に示す.

| | |
|-----|----------------------------|
| OS | Ubuntu16.04 |
| RTM | OpenRTM-aist-1.1.2-RELEASE |

3.2. 環境構築

OpenRTM-aist および rtshell はインストールされているものとする.

3.2.1. Boost ライブラリのインストール

本 RTC 群には, Boost ライブラリを使用しているためインストールする.

```
$ sudo apt-get install libboost-all-dev
```

3.2.2. SEED-Noid_Dual-Arm_pkg のダウンロード

Github から SEED-Noid_Dual-Arm_pkg をダウンロードする.

```
$ mkdir ~/workspace
```

```
$ cd ~/workspace
```

```
$ git clone https://github.com/Mayuka-Shii/SEED-Noid\_Dual-Arm\_pkg
```

3.2.3. SEED-Noid_Dual-Arm_pkg のビルド

RTC 群をビルドするためのシェルスクリプトを実行します.

```
$ cd /workspace/ SEED-Noid_Dual-Arm_pkg/script/
```

```
$ sh CompBuild.sh
```

3.3. 操作方法

3.3.1. RTC 群の起動およびアクティベート

実機 SEED-Noid を動かす準備は整っているものとする。RTC 群を起動およびアクティベートをするためのシェルスクリプトを実行する。

```
$ cd /workspace/ SEED-Noid_Dual-Arm_pkg/script/  
$ sh Exe_Connect.sh  
$sh Act.sh
```

3.3.2. DualArmController の操作

DualArmController で、実機の SEED-Noid を制御することができる。DualArmController の詳細については第 5 項参照。

4. シミュレーション制御方法

4.1. 動作確認環境

本 RTC 群の動作確認環境を以下に示す。

| | |
|--------|----------------------------|
| OS | Ubuntu16.04 |
| RTM | OpenRTM-aist-1.1.2-RELEASE |
| ROS | ROS kinetic |
| Gazebo | Gazebo7 |
| eSEAT | eSEAT2.5 |

4.2. 環境構築

OpenRTM-aist および rtshell はインストールされているものとする。

4.2.1. ROS の環境構築

以下の URL に従って、ROS の環境構築を行う。

URL : http://www2.meijo-u.ac.jp/~kohara/cms/technicalreport/ubuntu16-04_ros_install

4.2.2. aero-ros-pkg のダウンロード

aero-ros-pkg は, SEED-Noid を ROS で制御するための ROS パッケージである. 以下のリポジトリの URL にはドキュメントが用意されているため, ドキュメントに従ってインストールを行う.

URL : <https://github.com/seed-solutions/aero-ros-pkg>

4.2.3. eSEAT のインストール

ROS と RTM のブリッジを動作させるために eSEAT をインストールする. 以下の URL に従って eSEAT のインストールを行う.

URL : http://hara.jpn.com/_default/ja/Software/eSEAT%E3%81%AE%E6%A6%82%E8%A6%81%E3%81%A8%E3%82%A4%E3%83%B3%E3%82%B9%E3%83%88%E3%83%BC%E3%83%AB.html

4.2.4. SEED-Noid_Dual-Arm_pkg のダウンロード

Github から SEED-Noid_Dual-Arm_pkg をダウンロードする.

```
$ mkdir ~/workspace  
$ git clone https://github.com/Mayuka-Shii/SEED-Noid_Dual-Arm_pkg
```

4.2.5. SEED-Noid_Dual-Arm_pkg のビルド

RTC 群をビルドするためのシェルスクリプトを実行します.

```
$ cd /workspace/ SEED-Noid_Dual-Arm_pkg/script/  
$ sh CompBuild_sim.sh
```

4.3. 操作方法

4.3.1. RTC 群の起動及びアクティベート

RTC 群を起動およびアクティベートをするためのシェルスクリプトを実行する.

```
$ cd /workspace/ SEED-Noid_Dual-Arm_pkg/script/  
$ sh Exe_Connect_sim.sh  
$sh Act_sim.sh
```

4.3.2. ROS 側で launch ファイルの起動

```
$roslaunch aero_gazebo aero_gazebo.launch
```

4.3.3. DualArmController の操作

DualArmController で、シミュレーション上の SEED-Noid を制御することができる。
DualArmController の詳細については第 5 項参照。

5. DualArmController について

5.1. DualArmController 実行画面

DualArmControllerComp の実行ターミナルは以下のような画面になる。モード選択を行った後、値を打ち込むなどして操作が可能である。

```
-----  
Please Select Mode :)  
-----  
1 : Arm Servo ON/OFF  
2 : Gripper Servo ON/OFF  
3 : movePTPJointAbs  
4 : movePTPJointRel  
5 : moveLinearCartesianAbs  
6 : movePTPCartesianAbs  
7 : Gripper Open/Close  
8 : Gripper Opening Controll  
-----  
>> 
```

5.2. DualArmController が有する機能

DualArmController は以下のような機能を有している。尚，シミュレーション環境で使える機能は，ハイライトした項目のみである。

| 表示名 | 機能 |
|------------------------------|---|
| 1 : Arm Servo ON/OFF | アームのみサーボ ON/OFF |
| 2 : Gripper Servo ON/OFF | グリップのみサーボ ON/OFF |
| 3 : movePTPJointAbs | 絶対関節座標で指定された目標位置に対し，関節空間における直線補間で動作する。 |
| 4 : movePTPJointRel | 相対関節座標で指定された目標位置に対し，関節空間における直線補間で動作する。 |
| 5 : moveLinearCartesianAbs | ロボット座標系の絶対値で指定された目標位置に対し，直交空間における直線補間で動作する。 |
| 6 : movePTPCartesianAbs | ロボット座標系の絶対値で指定された目標位置に対し，関節空間における直線補間で動作する。 |
| 7 : Gripper Open/Close | グリップの開閉 |
| 8 : Gripper Opening Controll | グリップ開度[%]の指定 |

5.3. 座標系

右腕の制御のための座標系を図 2, 左腕の制御のための座標系を図 3 に示す.

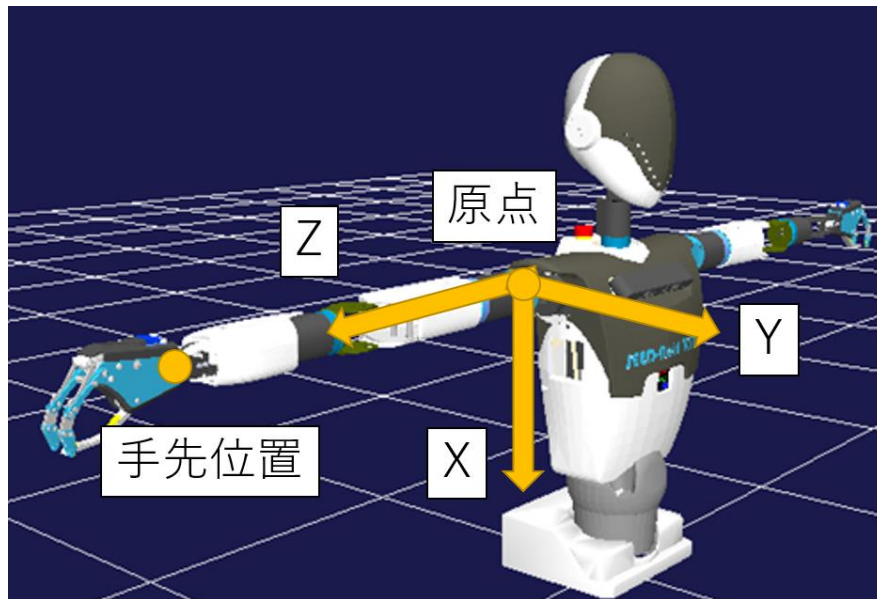


図 1 右腕の座標系

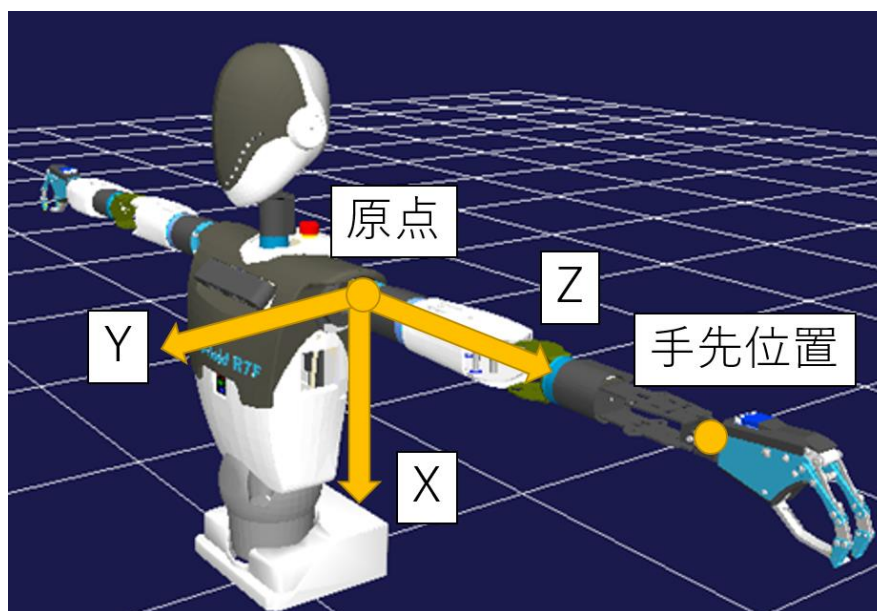


図 2 左腕の座標系