



ROBOT SYSTEM DESIGN
LABORATORY

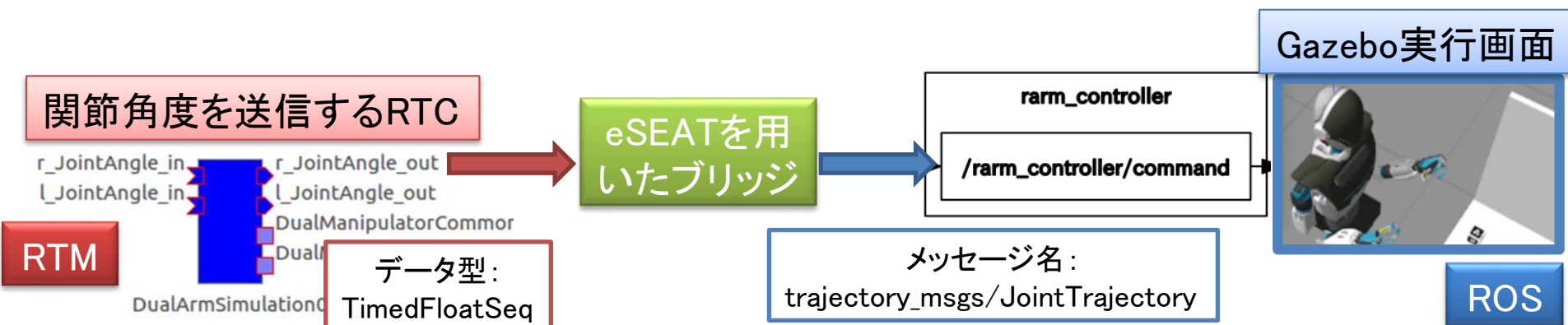
RTMとROSのブリッジ作成方法

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20日

はじめに

今回作るサンプル

<https://github.com/Mayuka-Shii/SEED-Noid Dual-Arm pkg/blob/master/SEATML/SEED-Noid Bridge command r.seatml>



RTMからROSに目標関節角度を送り、シミュレータ上のモデルを動かすためのブリッジ

RTMとROSの対応関係

RTM	データポート		データ型名
ROS	トピック通信	トピック名	メッセージ名

ROSは、RTMのように明示的にポート同士をつなぐことがないため、相手を指定するためにトピック名が必要となる

eSEATは産総研の原氏によって開発されたソフトウェアである。詳しい解説などはこちらのページを参照されたい。

http://hara.jpn.com/default/ja/Software/eSEAT_v2.5.html

本マニュアルでは、私の開発したRTCに対応するようなSEATMLファイルをサンプルとして記述方法の解説を行う。



アダプターの実装

アダプターは, RTMやROSの入出力ポートのようなものを定義できる

```
<adaptor name="" type="" datatype="" />
```

アダプターの実装方法

	adaptor name	type	datatype
RTM (データポート)	任意の名前	入力:rtcin 出力:rtcout	データ型名
ROS (トピック通信)	トピック名	入力:ros_sub 出力:ros_pub	メッセージ名

今回作るサンプルの場合

	adaptor name	type	datatype
RTM (データポート)	in	rtcin	TimedFloatSeq
ROS (トピック通信)	/rarm_controller/command	ros_pub	trajectory_msgs/JointTrajectory

オブジェクトの扱い方

```
data = seat.get_in_data()
```

dataの詳細

TimedFloatSeq

名前	型
tm	RTC::Time
data	sequence<float>

関節角度の値を取り出すとき、

data.data[0]

data.data[1]

という風に指定する

```
Rval = seat.adaptors['/rarm_controller/command'].newMessage()
```

```
Rp = JointTrajectoryPoint()
```

Rvalの詳細

trajectory_msgs/JointTrajectory

std_msgs/Header header

string[] joint_names

trajectory_msgs/JointTrajectoryPoint[] points

Rpの詳細

trajectory_msgs/JointTrajectoryPoint

float64[] positions

float64[] velocities

float64[] accelerations

float64[] effort

duration time_from_start

関節角度の値を入れるときには、

Rp.positions.append(値)

Rval.points.append(Rp)

という風に2段階踏むことになる



参考ページ

Nobu19800様のGithubで, RTMの基本データ型のマニュアルがまとめられており, 参考になります.

<https://github.com/Nobu19800/DataTypeManual/wiki/%E5%9F%BA%E6%9C%AC%E3%83%87%E3%83%BC%E3%82%BF%E5%9E%8B>

ROSの情報がまとめられたサイトです. メッセージの構成の確認などに有用です.

<http://docs.ros.org/>

例) JointTrajectory

http://docs.ros.org/api/trajectory_msgs/html/msg/JointTrajectory.html

