

双腕制御機能共通インタフェース 2.0 仕様書

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.....	3
1.1. 関連文書.....	3
1.2. 基本方針.....	3
2. 名前空間定義.....	3
3. データ型定義.....	4
3.1. 標準型.....	4
3.2. 型宣言.....	4
3.2.1. RETURN_ID.....	4
3.2.2. HgMatrix.....	4
3.2.3. DoubleSeq.....	4
3.2.4. JointPos.....	4
3.2.5. ULONG.....	4
3.2.6. CarPosWithElbow.....	5
4. 共通インタフェース定義.....	5
4.1. データポート.....	5
4.2. サービスポート.....	5
4.2.1. DualManipulatorCommonInterface_Common.....	5
4.2.2. DualManipulatorCommonInterface_Middle.....	6
5. 双腕制御機能共通インタフェース 2.0 を利用したシステム構成例.....	7
5.1. SEED-Noid における双腕作業のための RTC 群.....	7
6. CORBA IDL.....	8
6.1. DualManipulatorCommonInterface_DataTypes.idl.....	8
6.2. DualManipulatorCommonInterface_Common.idl.....	8
6.3. DualManipulatorCommonInterface_MiddleLevel.idl.....	9

1. はじめに

双腕ロボット制御機能共通インタフェースは，NEDO 次世代ロボット知能化技術開発プロジェクトによって作成された，双腕を持つロボットアームの制御に関わるインタフェースである．この双腕ロボット制御機能共通インタフェースに，関数の追加や変更などを行ったのが双腕ロボット制御機能共通インタフェース 2.0 である．

1.1. 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

1.2. 基本方針

双腕ロボット制御機能共通インタフェース 2.0 には，右腕用のロボットアーム制御機能共通インタフェースと，左腕用のロボットアーム制御機能共通インタフェースがインクルードされている．このため，双腕ロボット制御機能共通インタフェース 2.0 のみで，双腕制御に関する機能と，両腕のロボットアーム制御機能共通インタフェースに定義された機能が使えるようになっている．

2. 名前空間定義

双腕ロボット制御機能共通インタフェース 2.0 に属する双腕制御に関する機能には，固有の名前空間として「JARA_ARM_DUAL」を定義している．本仕様書において規定するデータ型およびインタフェース定義において，名前空間の記載がない場合は，名前空間「JARA_ARM_DUAL」に属しているものとする．また，インクルードしている右腕用のロボットアーム制御機能共通インタフェースでは，固有の名前空間として「JARA_ARM」，左腕用のロボットアーム制御機能共通インタフェースでは，固有の名前空間として「JARA_ARM_LEFT」を定義している．

3. データ型定義

双腕ロボット制御機能共通インタフェース 2.0 で使用するデータ型を以下に示す.

3.1. 標準型

本仕様書では OpenRTM-aist の標準型は使用していない.

3.2. 型宣言

3.2.1. RETURN_ID

リターン情報を保持するための型

属性		
id	long	リターンコード
comment	string	詳細コメント

3.2.2. HgMatrix

同次変換行列 4×4 の第 4 行を省略した 3×4 の行列

<code>typedef double HgMatrix[3][4];</code>

3.2.3. DoubleSeq

基本データ型 `double` のシーケンス型

<code>typedef sequence<double> DoubleSeq;</code>
--

3.2.4. JointPos

関節座標値を表現するための型であり `double` の配列として定義

<code>typedef sequence<double> JointPos;</code>

3.2.5. ULONG

基本データ型 `unsigned long` の短縮形

<code>typedef unsigned long ULONG;</code>

3.2.6. CarPosWithElbow

位置姿勢(同次変換行列)と肘角を格納するための型

属性		
carPos	HgMatrix	位置姿勢を表現する同次変換行列
elbow	double	肘の角度
structFlag	ULONG	付加情報を格納するためのフラグ

4. 共通インタフェース定義

以下に双腕ロボット制御機能共通インタフェース 2.0 で使用する共通インタフェースの定義を示す.

4.1. データポート

本仕様では固有のデータポートは定義していない.

4.2. サービスポート

4.2.1. DualManipulatorCommonInterface_Common

アプリケーションなどが,「双腕ロボット制御モジュール」にサーボ ON/OFF など一般的なロボットの動作を指示するためのインタフェースである.

メソッド		
servoONArm	RETURN_ID	腕部のサーボモータを ON にする.
servoOFFArm	RETURN_ID	腕部のサーボモータを OFF にする.
servoONHand	RETURN_ID	グリップ部のサーボモータを ON にする.
servoOFFHand	RETURN_ID	グリップ部のサーボモータを OFF にする.

4.2.2. DualManipulatorCommonInterface_Middle

アプリケーションなどが、「双腕ロボット制御モジュール」に双腕のモーションを指示するためのインタフェースである。

メソッド				
closeGripper			RETURN_ID	グリップスを完全に閉じる。
	in	rArm	ULONG	1：右グリップスを閉じる
	in	lArm	ULONG	1：左グリップスを閉じる
getRelativePosition			RETURN_ID	双腕の相対位置姿勢関係を取得する。
	in	RelPos	HgMatrix	相対位置・姿勢[単位：mm, degree]
moveGripper			RETURN_ID	グリップスを指定した開閉角度とする。
	in	rArm	ULONG	1：右グリップスを閉じる
	in	lArm	ULONG	1：左グリップスを閉じる
moveLinearCartesianAbs			RETURN_ID	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。
	in	rArm	CarPosWithElbow	右腕絶対目標位置・姿勢[単位:mm、degree]
	in	lArm	CarPosWithElbow	左腕絶対目標位置・姿勢[単位:mm、degree]
moveLinearCartesianRel			RETURN_ID	ロボット座標系の相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。
	in	rArm	CarPosWithElbow	右腕相対目標位置・姿勢[単位:mm、degree]
	in	lArm	CarPosWithElbow	左腕相対目標位置・姿勢[単位:mm、degree]
movePTPCartesianAbs			RETURN_ID	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。
	in	rArm	CarPosWithElbow	右腕絶対目標位置・姿勢[単位:mm、degree]
	in	lArm	CarPosWithElbow	左腕絶対目標位置・姿勢[単位:mm、degree]
movePTPCartesianRel			RETURN_ID	ロボット座標系の相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。
	in	rArm	CarPosWithElbow	右腕相対目標位置・姿勢[単位:mm、degree]
	in	lArm	CarPosWithElbow	左腕相対目標位置・姿勢[単位:mm、degree]
movePTPJntAbs			RETURN_ID	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。
	in	rArm	JointPos	右腕絶対目標角度[単位:degree]
	in	lArm	JointPos	左腕絶対目標角度[単位:degree]
movePTPJointRel			RETURN_ID	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。
	in	rArm	JointPos	右腕相対目標角度[単位:degree]
	in	lArm	JointPos	左腕相対目標角度[単位:degree]

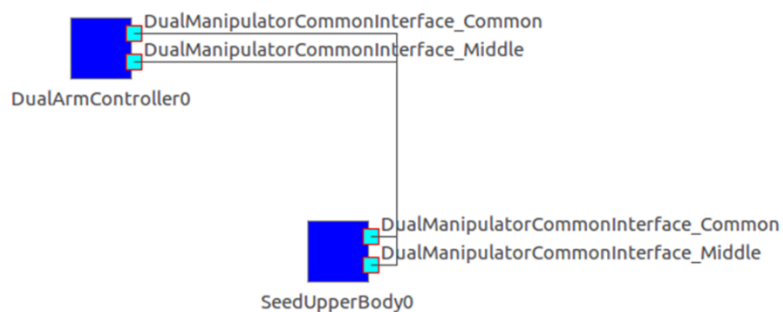
5. 双腕制御機能共通インタフェース 2.0 を利用したシステム構成例

5.1. SEED-Noid における双腕作業のための RTC 群

○詳細 URL : https://github.com/Mayuka-Shii/SEED-Noid_Dual-Arm_pkg

○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。インタフェースとして、双腕制御機能共通インタフェース 2.0 を使用している。



6. CORBA IDL

双腕制御機能共通インタフェース 2.0 の IDL 定義を以下に示す.

6.1. DualManipulatorCommonInterface_DataTypes.idl

```
#ifndef DUALMANIPULATORCOMMONINTERFACE_DATATYPES_IDL
#define DUALMANIPULATORCOMMONINTERFACE_DATATYPES_IDL

#include "BasicDataType.idl"

module JARA_ARM_DUAL{

    typedef double HgMatrix [3][4];
    typedef sequence<double> DoubleSeq;
    typedef sequence<double> JointPos;
    typedef sequence<JointPos> JointPosSeq;
    typedef unsigned long ULONG;

    struct RETURN_ID {
        long id;
        string comment;
    };

};

#endif // DUALMANIPULATORCOMMONINTERFACE_DATATYPES_IDL
```

6.2. DualManipulatorCommonInterface_Common.idl

```
#ifndef DUALMANIPULATORCOMMONINTERFACE_COMMON_IDL
#define DUALMANIPULATORCOMMONINTERFACE_COMMON_IDL

#include "ManipulatorCommonInterface_Common.idl"
#include "LeftManipulatorCommonInterface_Common.idl"
#include "DualManipulatorCommonInterface_DataTypes.idl"
```



```

module JARA_ARM_DUAL{

interface DualManipulatorCommonInterface_Common {

    RETURN_ID servoOFFArm();
    RETURN_ID servoOFFHand();
    RETURN_ID servoONArm();
    RETURN_ID servoONHand();

};

};

#endif // DUALMANIPULATORCOMMONINTERFACE_COMMON_IDL

```

6.3. DualManipulatorCommonInterface_MiddleLevel.idl

```

#ifndef DUALMANIPULATORCOMMONINTERFACE_MIDDLE_IDL
#define DUALMANIPULATORCOMMONINTERFACE_MIDDLE_IDL

#include "ManipulatorCommonInterface_MiddleLevel.idl"
#include "LeftManipulatorCommonInterface_MiddleLevel.idl"
#include "DualManipulatorCommonInterface_DataTypes.idl"

module JARA_ARM_DUAL{

struct CarPosWithElbow {
    HgMatrix carPos;
    double elbow;
    ULONG structFlag;
};

interface DualManipulatorCommonInterface_Middle {

    RETURN_ID closeGripper(in ULONG rArm, in ULONG lArm);
    RETURN_ID getRelativePosition(in HgMatrix RelPos);

```

```
    RETURN_ID moveGripper(in ULONG rArm, in ULONG lArm);
    RETURN_ID  moveLinearCartesianAbs(in  CarPosWithElbow  rArm,  in
CarPosWithElbow lArm);
    RETURN_ID  moveLinearCartesianRel(in  CarPosWithElbow  rArm,  in
CarPosWithElbow lArm);
    RETURN_ID movePTPCartesianAbs(in CarPosWithElbow rArm, in CarPosWithElbow
lArm);
    RETURN_ID movePTPCartesianRel(in CarPosWithElbow rArm, in CarPosWithElbow
lArm);
    RETURN_ID movePTPJointAbs(in JointPos rArm, in JointPos lArm);
    RETURN_ID movePTPJointRel(in JointPos rArm, in JointPos lArm);
    RETURN_ID openGripper(in ULONG rArm, in ULONG lArm);

};

};

#endif // DUALMANIPULATORCOMMONINTERFACE_MIDDLE_IDL
```