

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)

NEDO 次世代ロボット知能化技術開発プロジェクト

2012 年 2 月 24 日



【改版履歴】

日付	版番号	改版ページ	改版内容
2012. 2. 24	1. 0	全ページ	新規作成

【本書の利用にあたって】

本書は、クリエイティブ・コモンズ 表示 2.1 ライセンスの下に提供される。

(<http://creativecommons.org/licenses/by-sa/2.1/jp/>)



【本書の策定メンバー】

(敬称略、五十音順)

小笠原哲也 (東京大学大学院 情報理工学系研究科 知能機械情報学専攻)

河井良浩 (独立行政法人産業技術総合研究所)

中本啓之 (株式会社セック 開発本部 第四開発部)

二宮恒樹 (富士ソフト株式会社 ロボット事業グループ 商品開発ユニット)

原田研介 (独立行政法人産業技術総合研究所)

(所属は 2012 年 2 月 24 日現在)

目次

1	はじめに.....	1
1.1	対象機能の概要.....	1
1.2	標準システム構成.....	2
2	本書を読む上での注意.....	3
2.1	基本方針.....	3
2.2	フォーマットと表現方法.....	3
2.2.1	型定義.....	3
2.2.2	インタフェース定義.....	3
2.3	本仕様書における前提条件.....	3
2.3.1	座標系定義について.....	3
3	名前空間定義.....	4
4	データ型定義.....	4
4.1	標準型.....	4
4.2	型宣言.....	4
4.2.1	RETURN_ID.....	4
4.2.2	HgMatrix.....	5
4.2.1	DoubleSeq.....	5
4.2.2	JointPos.....	5
4.2.3	JointPosSeq.....	5
4.2.4	ULONG.....	5
4.2.5	CarPosWithElbow.....	5
5	共通インタフェース定義.....	6
5.1	データポート.....	6
5.2	サービスポート.....	6
5.2.1	CommonCommands.....	6
5.2.2	MotionCommands.....	7
6	共通インタフェースを利用したシステム構成例.....	9
6.1	頭部ステレオカメラを用いた双腕ロボットによるマニピュレーション作業.....	9
7	CORBA IDL.....	11
7.1	HIROController.idl.....	11
8	参考文献.....	13

表目次

表 4.1 RETURN_ID.....	4
表 4.2 戻り値一覧	4
表 4.3 CarPosWithElbow	5
表 5.1 CommonCommands.....	6
表 5.2 MotionCommands.....	7

図目次

図 1.1 双腕ロボット制御機能共通インタフェースの使用シーン例	1
図 1.2 双腕ロボット制御機能共通インタフェースの使用シーン例	2
図 1.3 双腕ロボット制御機能共通インターフェースを利用したシステム例	2
図 5.1 CommonCommands インタフェース	6
図 5.2 MotionCommands インタフェース	7
図 6.1 双腕ロボット制御コンポーネント適用例.....	9
図 6.2 頭部ステレオカメラを用いた双腕ロボットによるマニピュレーション作業モジュール群	10

1 はじめに

近年、ロボットの開発を効率化するためにコンポーネントベースのミドルウェア開発が盛んになっている。コンポーネントベースのミドルウェア開発において、インタフェースの共通化は、コンポーネントの相互接続性や相互運用性を確保するうえで非常に重要である。このような背景に基づき、本書では、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の共通仕様を定義する。

本共通インタフェースを規定することにより、使用するロボットの機種が異なっても、ロボットアームに指示を出す上位モジュールは同一命令で制御することができるため、ハードウェアを差し替えた場合に、ソフトウェアを再開発する必要がなくなるといったメリットが期待できる。

1.1 対象機能の概要

本仕様書では、双腕を有するロボットアームを制御するための共通インタフェースを規定している。

双腕ロボット制御機能共通インタフェースを実装した RT コンポーネントの使用シーンの一例を以下に示す。



図 1.1 双腕ロボット制御機能共通インタフェースの使用シーン例

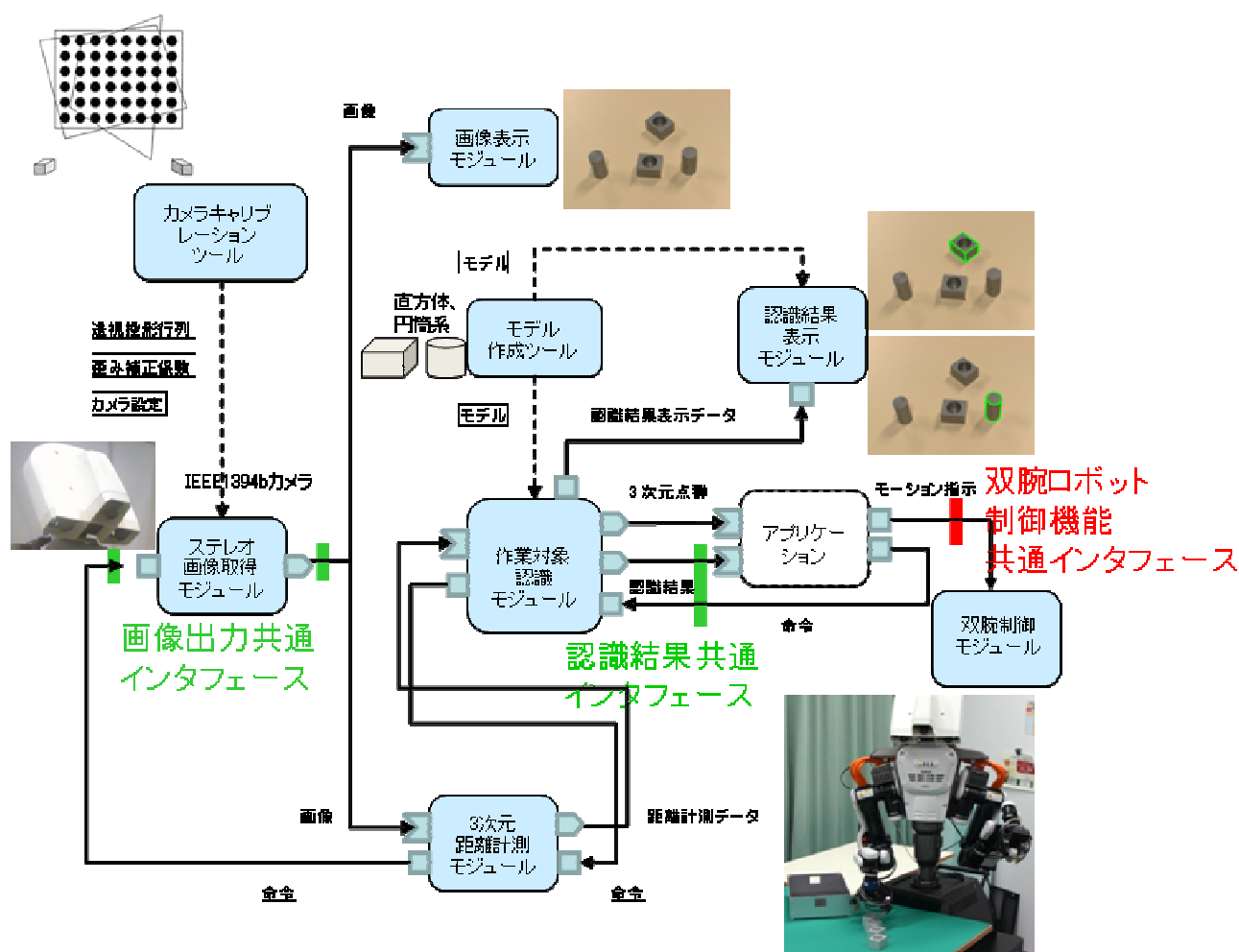


図 1.2 双腕ロボット制御機能共通インタフェースの使用シーン例

1.2 標準システム構成

双腕ロボット制御機能共通インタフェースを利用した標準的なシステム構成例を以下に示す。

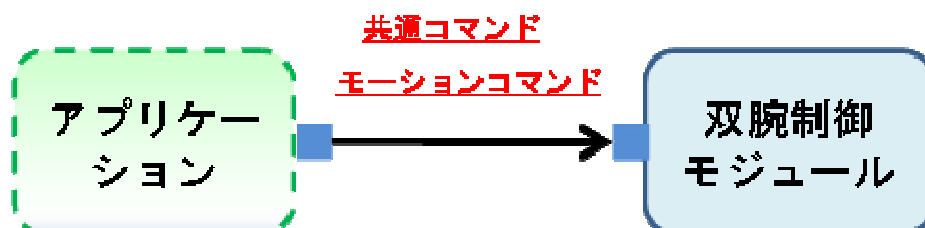


図 1.3 双腕ロボット制御機能共通インターフェースを利用したシステム例

本仕様書中では、上図中の「双腕制御モジュール」を対象としており、このモジュールの入出力情報を規定している。具体的には、外部コンポーネントから「作業対象認識モジュール」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と、双腕のモーションを指示するためのインタフェースを規定している。

2 本書を読む上での注意

2.1 基本方針

インタフェース仕様の共通化は、仕様に合致しないコンポーネントを排除するため、時に開発内容を制限してしまうこともある。本仕様では、そのような制限を低減するために、以下のような方針で共通インタフェース仕様を定義する。

- 最低限のインタフェース仕様の定義:コンポーネントを相互接続・相互運用するために必要な最低限のインタフェース仕様のみを定義する。開発の制約となる仕様は最低限にとどめ、その他の部分は開発者が自由に拡張することができるようにする。
- 任意の機能の定義:いくつかの機能については実装を任意とする。実装された場合は、本書に書かれた仕様に準拠することを要求するが、実装をするかどうかは任意であり、それを実装していなかったからといって共通インタフェース仕様から外れるものとはしない。

2.2 フォーマットと表現方法

2.2.1 型定義

本仕様書では、型定義を次の表形式を用いて記述する。

表 XX:<型名>

属性		
<要素名>	<要素型>	<内容>
...

2.2.2 インタフェース定義

本仕様書では、インタフェース定義を次の表形式を用いて記述する。

表 XX:<インタフェース名>

メソッド				
<メソッド名>		<戻り値型>	<内容>	
	<方向>	<パラメータ名>	<パラメータ型>	<内容>

<備考>				

2.3 本仕様書における前提条件

2.3.1 座標系定義について

本仕様書で取り扱う双腕ロボットの座標系は、絶対座標系で座標が定義され、前方方向をX軸方向とした右手系を想定している。また、各要素の単位は、位置[m]、速度[m/s]、角度[radian]、角速度[radian/s]を想定している。

3 名前空間定義

作業系画像認識機能共通インタフェースでは、固有の名前空間は定義していない。

4 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で使用するデータ型を以下に示す。

4.1 標準型

本仕様書では OpenRTM-aist の標準型は使用していない。

4.2 型宣言

4.2.1 RETURN_ID

リターン情報を保持するための型。

表 4.1 RETURN_ID

属性		
id	long	リターンコード
comment	string	戻りを説明するための詳細コメント

※本仕様書では id に格納するリターンコードとして、使用頻度が高いと思われる以下の値を事前定義する。

表 4.2 戻り値一覧

値	戻り値名	概要
0	OK	オペレーションを正常に受け付け
-1	NG	オペレーションを拒否
-2	STATUS_ERR	オペレーションを受け付け可能な状態でない
-3	VALUE_ERR	引数が不正
-4	NOT_SV_ON_ERR	全ての軸のサーボが入っていない
-5	FULL_MOTION_QUEUE_ERR	バッファが一杯
-6～ -9999	システム予約領域	
-10000 ～	機種依存領域	

4.2.2 HgMatrix

同次変換行列 4×4 の第4行を省略した 3×4 の行列。座標系は右手系。

```
typedef double HgMatrix[3][4];
```

4.2.1 DoubleSeq

基本データ型 double のシーケンス型。

```
typedef sequence<double> DoubleSeq;
```

4.2.2 JointPos

関節座標値を表現するための型。double の配列として定義されている。

```
typedef sequence<double> JointPos;
```

4.2.3 JointPosSeq

関節座標値 JointPos のシーケンスを表現するための型。JointPos の配列として定義されている。

```
typedef sequence<JointPos> JointPosSeq;
```

4.2.4 ULONG

基本データ型 unsigned long の短縮形。

```
typedef unsigned long ULONG;
```

4.2.5 CarPosWithElbow

位置姿勢(同次変換行列)と肘角を格納するための型。

表 4.3 CarPosWithElbow

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

※structFlag に格納する情報は、機種依存データとなる。詳細は各マニピュレータのドキュメントを参照のこと。

5 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で使用する共通インタフェースの定義を示す。

5.1 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。

5.2 サービスポート

5.2.1 CommonCommands

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。



図 5.1 CommonCommands インタフェース

表 5.1 CommonCommands

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。
servoON	RETURN_ID	グリップ・腕部両方のサーボモータを ON にする。
servoOFF	RETURN_ID	グリップ・腕部両方のサーボモータを OFF にする。

5.2.2 MotionCommands

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。



図 5.2 MotionCommands インタフェース

表 5.2 MotionCommands

メソッド				
closeGripper			RETURN_ID	グリッパを完全に閉じる。
グリッパを閉じた際の姿勢については、機種依存。				
moveGripper			RETURN_ID	グリッパを指定した開閉角度とする。
	in	r_angle	DoubleSeq	右グリッパの開閉角度割合[%] 0%:完全に閉じた状態 100%:完全に開いた状態
	in	l_angle	DoubleSeq	左グリッパの開閉角度割合[%] 0%:完全に閉じた状態 100%:完全に開いた状態
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]
「直交空間における直線補間」とは、直交空間中の各方向の並進および回転の動作が、同時に開始・終了するとともに、全ての加速時間と減速時間が同じになるように軌跡生成する動作のことである。 6軸マニピュレータの場合は、引数 carPoint のメンバ変数 elbow は無視される。				
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]
「直交空間における直線補間」とは、直交空間中の各方向の並進および回転の動作が、同時に開始・終了するとともに、全ての加速時間と減速時間が同じになるように軌跡生成する動作のことである。 6軸マニピュレータの場合は、引数 carPoint のメンバ変数 elbow は無視される。				

メソッド				
movePTPJointAbs			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	jointPoints	JointPos	絶対目標位置[単位 degree]
「関節空間における直線補間」とは、全軸の動作が、同時に開始・終了するとともに、全ての加速時間と減速時間が同じになるように軌跡生成する動作のことである。 引数 jointPoints 配列の値の順番は、J1、J2、J3、・・・とする。				
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	jointPoints	JointPos	相対目標位置[単位 degree]
「関節空間における直線補間」とは、全軸の動作が、同時に開始・終了するとともに、全ての加速時間と減速時間が同じになるように軌跡生成する動作のことである。 引数 jointPoints 配列の値の順番は、J1、J2、J3、・・・とする。				
movePTPJointAbsSeq			RETURN_ID	両方の腕に対して、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	jointPointsSeq	JointPosSeq	両腕部のための絶対目標位置[単位 degree]
「関節空間における直線補間」とは、全軸の動作が、同時に開始・終了するとともに、全ての加速時間と減速時間が同じになるように軌跡生成する動作のことである。 引数 jointPoints 配列の値の順番は、J1、J2、J3、・・・とする。				
openGripper			RETURN_ID	グリップを完全に開く。
グリップを開いた際の姿勢については、機種依存。				
setSpeedCartesian			RETURN_ID	直交空間における動作時の速度を%指定する。
	in	spdRation	ULONG	最大速度に対する割合指定[単位:%]
上限は 100%。初期値は 0%				
setSpeedJoint			RETURN_ID	関節空間における動作時の速度を%指定する。
	in	spdRation	ULONG	最大速度に対する割合指定[単位:%]
上限は 100%。初期値は 0%				

6 共通インタフェースを利用したシステム構成例

6.1 頭部ステレオカメラを用いた双腕ロボットによるマニピュレーション作業

○開発者:独立行政法人 産業技術総合研究所 知能システム研究部門 タスクビジョン研究グループ

○詳細 URL:http://www.openrtm.org/openrtm/ja/project/NEDO_Intelligent_PRJ_HiroAccPri_1001

○概要

双腕ロボットを用いて工場での部品整理をイメージしたサービスを行うための RT コンポーネント群。作業台に置かれた部品を状況に応じて再配置し、個々の部品を検出して双腕を活かして箱に整理して入れ、双腕で別の場所に運ぶ Pick&Place 作業を行うものである。

本システムは、頭部ステレオカメラによる部品認識・双腕ロボットでの作業計画・ロボット動作を行うためのモジュール群で構成される。

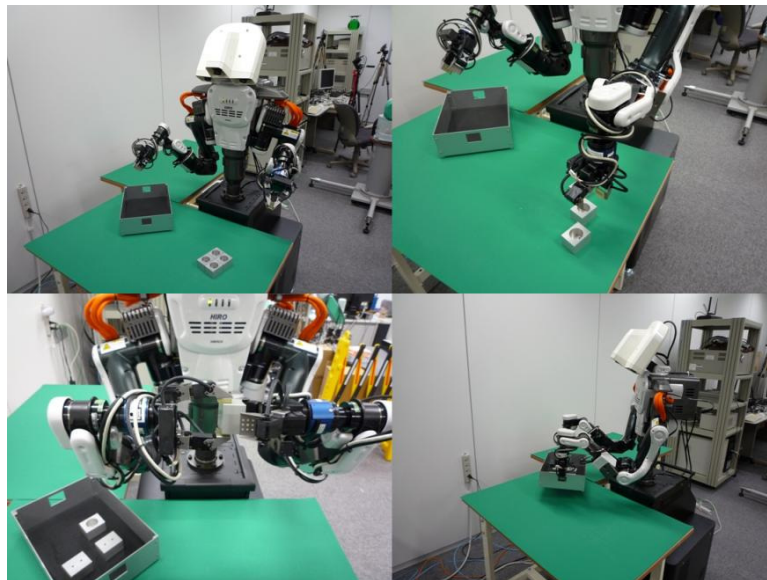


図 6.1 双腕ロボット制御コンポーネント適用例

本コンポーネントを利用したシステム構成例を以下に示す。図中の「ACT 共通 IF」の部分が、本仕様書で規定している共通インタフェースを使用している部分である。

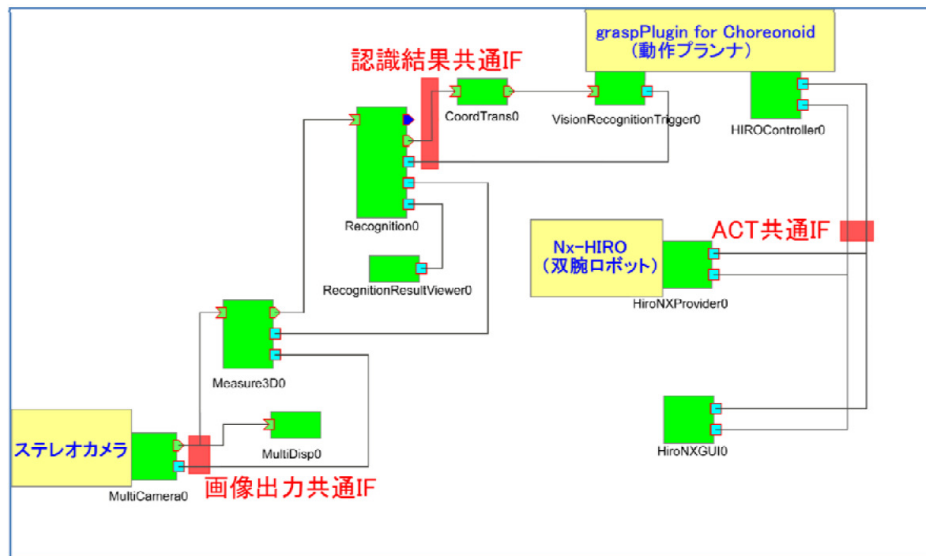


図 6.2 頭部ステレオカメラを用いた双腕ロボットによるマニピュレーション作業モジュール群

7 CORBA IDL

双腕ロボット制御機能共通インタフェースの IDL 定義を以下に示す。

7.1 HIROController.idl

```
#ifndef HIROController_IDL
#define HIROController_IDL

interface CommonCommands {

    struct RETURN_ID {
        long id;
        string comment;
    };

    RETURN_ID servoOFF();
    RETURN_ID servoON();

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

};
```

```

interface MotionCommands {

    typedef double HgMatrix [3][4]; // 3x3 の回転行列+1 列の座標
    typedef sequence<double> DoubleSeq;
    typedef sequence<double> JointPos;
    typedef sequence<JointPos> JointPosSeq;
    typedef unsigned long ULONG;

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

    struct RETURN_ID {
        long id;
        string comment;
    };

    RETURN_ID closeGripper();
    RETURN_ID moveGripper(in DoubleSeq r_angle, in DoubleSeq l_angle);
    RETURN_ID moveLinearCartesianAbs(in CarPosWithElbow rArm, in CarPosWithElbow lArm);
    RETURN_ID moveLinearCartesianRel(in CarPosWithElbow rArm, in CarPosWithElbow lArm);
    RETURN_ID movePTPJointAbs(in JointPos jointPoints);
    RETURN_ID movePTPJointRel(in JointPos jointPoints);
    RETURN_ID movePTPJointAbsSeq(in JointPosSeq jointPointsSeq);
    RETURN_ID openGripper();
    RETURN_ID setSpeedCartesian(in ULONG spdRatio);
    RETURN_ID setSpeedJoint(in ULONG spdRatio);

};

#endif /* HIROController_IDL */

```

8 参考文献

なし