

次世代ロボット知能化技術開発プロジェクト

## 機能仕様書

相対位置決め制御モジュール

V e r . 1 . 1

2012 年 1 月 24 日

株式会社 東芝

国立大学法人 東北大学

## 改版履歷

[illegible]

# 目次

1. はじめに .....	4
1. 1. 本書の適用範囲 .....	4
1. 2. 関連文書 .....	4
1. 3. 本書を読むにあたって .....	4
2. 機能仕様 .....	5
2. 1. 機能概要 .....	5
2. 2. モジュール構成 .....	5
2. 3. ターゲットハードウェア .....	6
3. RTC仕様 .....	7
3. 1. RelativePositionControlComp .....	7
3. 1. 1. 機能概要 .....	7
3. 1. 2. 動作環境 .....	7
3. 1. 3. ポート情報 .....	8
3. 2. RHController_HGComp .....	10
3. 2. 1. 機能概要 .....	10
3. 2. 2. 動作環境 .....	10
3. 2. 3. ポート情報 .....	11
3. 3. dq_combineComp .....	12
3. 3. 1. 機能概要 .....	12
3. 3. 2. 動作環境 .....	12
3. 3. 3. ポート情報 .....	13
3. 4. DummyPlanner Comp (コンポーネント名) .....	14
3. 4. 1. 機能概要 .....	14
3. 4. 2. 動作環境 .....	14
3. 4. 3. ポート情報 .....	15
4. 特記事項 .....	17

# 1. はじめに

## 1. 1. 本書の適用範囲

本書は、相対位置決め制御モジュールの機能仕様について記述した文書である。本モジュールは、ハンド・アイ・カメラを搭載したロボットアームに、ビジュアルフィードバック動作を行わせるために開発したものである。ハンド・アイ・カメラで撮影したマーカ<sup>1</sup>の画像を基に、マーカの位置姿勢を算出するRTC「単眼位置姿勢計測・表示モジュール<sup>2</sup>」がある。本モジュールは、このモジュールと接続することにより、マーカの位置・姿勢情報を受け、マーカに近づくためのアーム関節角（速度）を算出するものである。

## 1. 2. 関連文書

表 1-1 関連文書

No。	文書名	備考
1	単眼位置姿勢計測・表示モジュール機能仕様書	
2	単眼位置姿勢計測・表示モジュール操作手順書	
3	相対位置決め制御モジュール操作手順書	

## 1. 3. 本書を読むにあたって

本書はRT ミドルウェア、RT コンポーネント（以下、RTC）に関する基本知識を備えた利用者を対象としている。RT ミドルウェア、RTC については下記を参照のこと。

OpenRTM-aist Official Website

URL : <http://www.openrtm.org/>

ロボットの動作シミュレータ（OpenHRP3）を用いる場合、下記を参照のこと。

URL : <http://www.openrtp.jp/openhrp3/jp/>

<sup>1</sup> 予め形状が登録されたマーカ

<sup>2</sup> 別途、「次世代ロボット知能化技術開発プロジェクト」の指定のサーバからダウンロードする

## 2. 機能仕様

### 2. 1. 機能概要

OpenHRP 3 でシュミレーションを行う場合、以下のコンポーネントを用いる。

- ①MarkerRecognitionComp (単眼位置姿勢計測・表示コンポーネント)
- ②RelativePositionControlComp (相対位置決めコンポーネント)
- ③RHController\_HGComp (リファレンスハード制御コンポーネント)
- ④dq\_combineComp (dq コンバインコンポーネント)
- ⑤DummyPlanner Comp (ダミープランナコンポーネント)

下記 Fig. 1 のモジュール構成に示すように、まず、⑤のダミープランナから指示を出し、①で目標マーカの位置・姿勢を算出し、②でアーム関節角速度を算出する。④でデータ形式を整え、③でリファレンスハードを制御する。

### 2. 2. モジュール構成

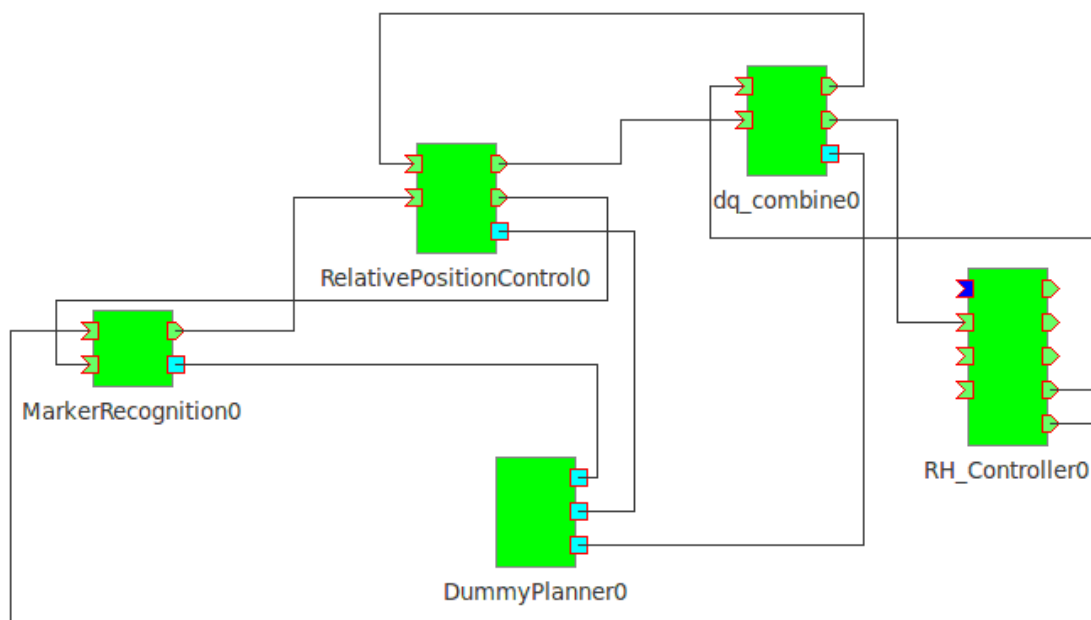


Fig.1 Module configuration

## 2. 3. ターゲットハードウェア

本知能モジュールは、「次世代ロボット知能化技術開発プロジェクト」で開発された Fig.2 に示すリファレンスハード（前川製作所製）を用い、ビジュアルフィードバック実験を行うことを想定して開発された。カメラをアーム手先に取り付ける必要があり、例えばシキノハイテック製VGAカメラ（KBCR-M05VU）などが適している。リファレンスハードの実機がない場合、Fig.3 に示すように、ロボット動作シミュレータ（OpenHRP3）を用いて、動作の確認を行えるようになっている。

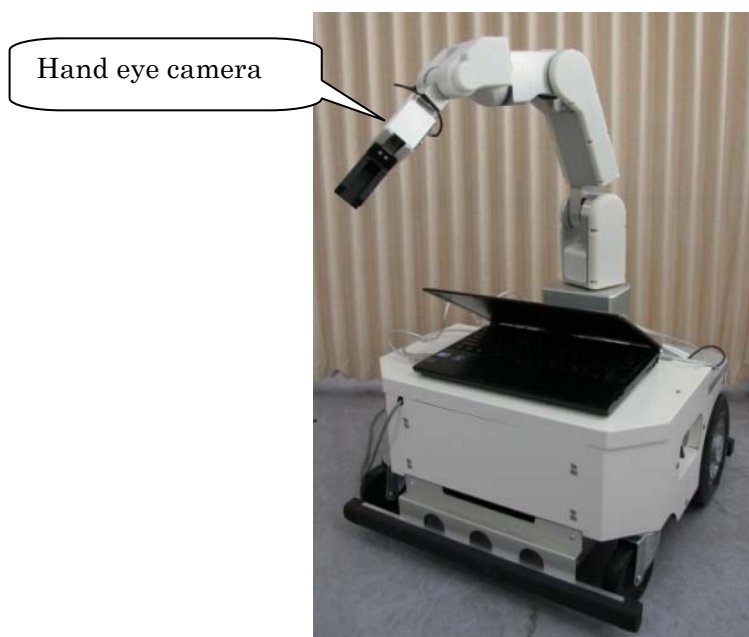


Fig. 2 Reference hardware

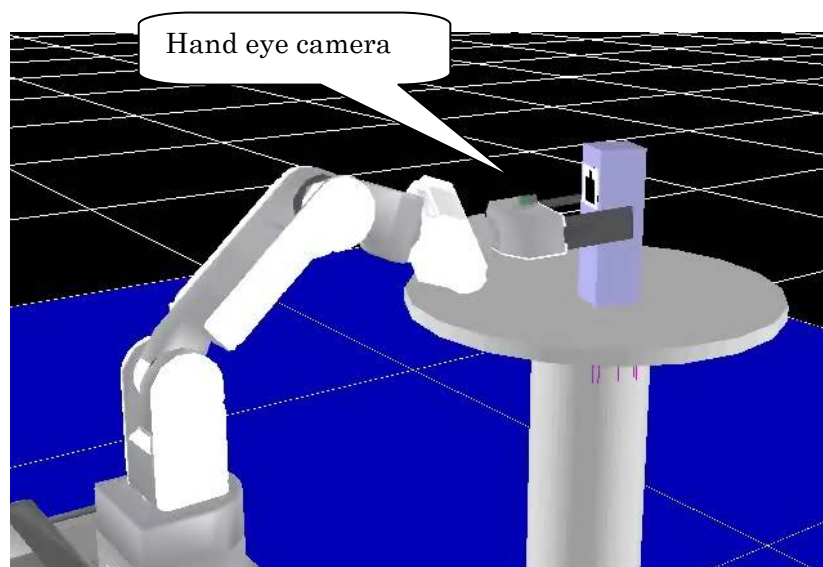


Fig. 3 Displayed results of OpenHRP3

## 3. RTC 仕様

### 3. 1. RelativePositionControlComp

#### 3. 1. 1. 機能概要

MarkerRecognition モジュールからのマーカ情報をもとにアームの各関節の角速度を出力するモジュールである。エンドエフェクタ座標系における対象物との相対位置・姿勢の設定はサービスポートにより行う。また、現在のロボット座標系におけるエンドエフェクタの位置・姿勢をMarkerRecognition モジュールに出力する。

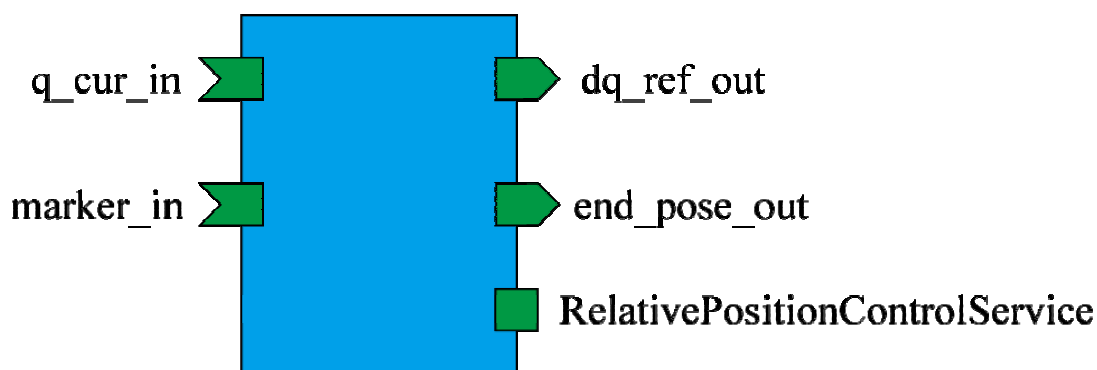


Fig.4 RelativePositionControl module

#### 3. 1. 2. 動作環境

表 3-1 動作環境

動作 OS	OS : Ubuntu 10.04 LTS
開発言語	C++
コンパイラ	g++
RT ミドルウェア／バージョン	OpenRTM-aist-1.0.0-RELEASE
依存パッケージ	(OpenHRP 3.1.1)

### 3. 1. 3. ポート情報

#### InPort

ポート名	型	データ長	説明
q_cur_in	TimedDoubleSeq	6 (アームジョイント数)	アームの現在関節角度
marker_in	TimedDoubleSeq	20×n (可変)	カメラ座標系におけるマーカの位置・姿勢 (センス系共通 I/F)

#### OutPort

ポート名	型	データ長	説明
dq_ref_out	TimedDoubleSeq	6 (アームジョイント数)	アームの目標関節角速度
end_pose_out	TimedDoubleSeq	20	ロボット座標系におけるエンドエフェクタの位置・姿勢 (センス系共通 I/F)

#### ServicePort

サービス名	型	説明
RelativePositionControlService	RelativePositionControlService	動作の開始・停止, 及び対象物との目標相対位置・姿勢の設定を行う。



ここで, RelativePositionControl の処理内容は以下のようになっている。

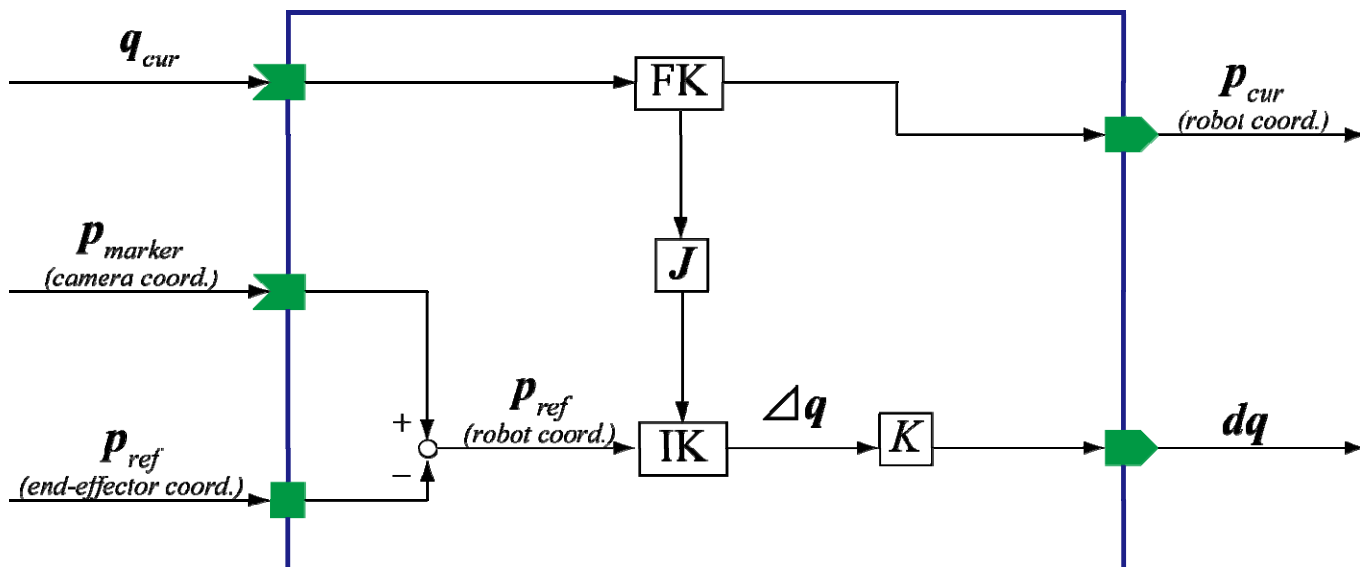


Fig.5 block diagram of RelativePositionControl module

- ・ 現在の関節角度から順運動学（FK）を計算し、ヤコビ行列とエンドエフェクタの現在位置・姿勢（ロボット座標系）を取得する。
- ・ 現在のマーカ位置・姿勢から対象物とエンドエフェクタとの目標相対位置・姿勢を差し引き、エンドエフェクタの最終的な目標位置・姿勢（ロボット座標系）を計算する。
- ・ 計算した目標位置・姿勢とヤコビ行列を用いて逆運動学（IK）を行い、 $\Delta q$  を求める。
- ・  $\Delta q$  に定数  $K$  を乗算し、関節角速度  $dq$  を計算する。

## 3. 2. RHController\_HGComp

### 3. 2. 1. 機能概要

リファレンスハードのハイゲインコントローラである。入力は関節角度と関節角速度の二種類の内どちらかを選択して利用可能であるが、現在は関節角速度を利用している。

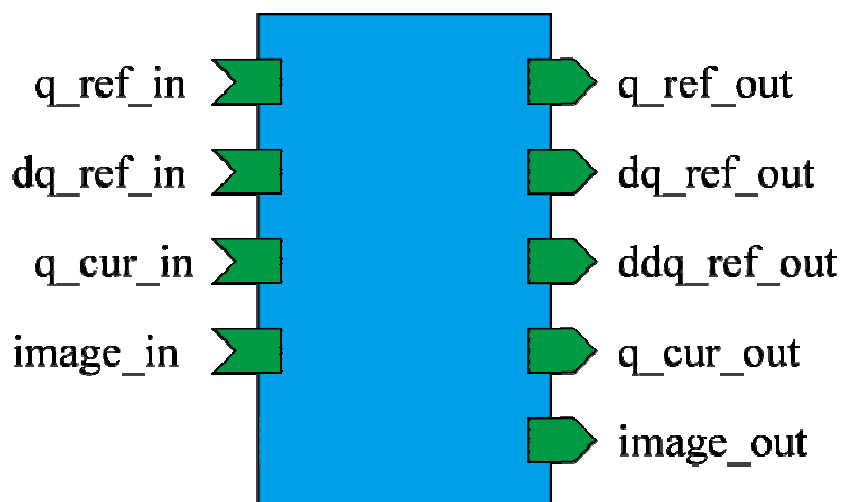


Fig.6 RHController\_HG module

### 3. 2. 2. 動作環境

表 3-2 動作環境

動作 OS	OS : Ubuntu 10.04 LTS
開発言語	C++
コンパイラ	g++
RT ミドルウェア／バージョン	OpenRTM-aist-1.0.0-RELEASE
依存パッケージ	(OpenHRP 3.1.1)

### 3. 2. 3. ポート情報

#### InPort

ポート名	型	データ長	説明
q_ref_in	TimedDoubleSeq	9 (RH 全ジョイント数)	目標関節角度の入力
dq_ref_in	TimedDoubleSeq	9 (RH 全ジョイント数)	目標関節角速度の入力
q_cur_in	TimedDoubleSeq	9 (RH 全ジョイント数)	OpenHRP から現在の関節角度を取得する
image_in	TimedLongSeq	640×480	OpenHRP の ViewSimulation からのカメラ画像データを取得する

#### OutPort

ポート名	型	データ長	説明
q_ref_out	TimedDoubleSeq	9 (RH 全ジョイント数)	OpenHRP へ目標関節角度を出力
dq_ref_out	TimedDoubleSeq	9 (RH 全ジョイント数)	OpenHRP へ目標関節角速度を出力
ddq_ref_out	TimedDoubleSeq	9 (RH 全ジョイント数)	OpenHRP へ目標関節角加速度を出力
q_cur_out	TimedDoubleSeq	9 (RH 全ジョイント数)	現在の関節角度
image_out	TimedLongSeq	640×480	OpenHRP の ViewSimulation からのカメラ画像データ

### 3. 3. dq\_combineComp

#### 3. 3. 1. 機能概要

RHController\_HG と RelativePositionControl の仲立ちをするモジュールである。RHController\_HG から取得した現在関節角度の中からアーム部のみ抽出し RelativePositionControl へ出力する。また、RelativePositionControl からの目標関節角速度に台車部とグリッパ部の目標関節角速度を加え、RHController\_HG へ出力する。

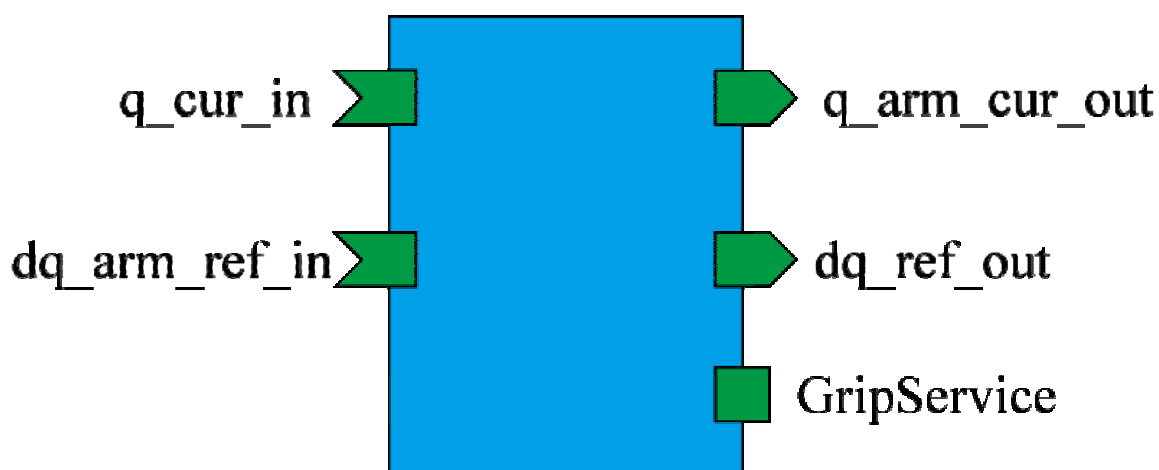


Fig.7 dq\_combine module

#### 3. 3. 2. 動作環境

表 3-3 動作環境

動作 OS	OS : Ubuntu 10.04 LTS
開発言語	C++
コンパイラ	g++
RT ミドルウェア／バージョン	OpenRTM-aist-1.0.0-RELEASE
依存パッケージ	(OpenHRP 3.1.1)

### 3. 3. 3. ポート情報

#### InPort

ポート名	型	データ長	説明
q_cur_in	TimedDoubleSeq	9 (RH 全ジョイント数)	RH の現在関節角度
dq_arm_ref_in	TimedDoubleSeq	6 (RH アームジョイント数)	RH アーム部の目標関節角速度

#### OutPort

ポート名	型	データ長	説明
q_arm_cur_out	TimedDoubleSeq	6 (RH アームジョイント数)	RH アーム部の現在関節角度
dq_ref_out	TimedDoubleSeq	9 (RH 全ジョイント数)	RH の目標関節角速度

#### ServicePort

サービス名	型	説明
GripService	GripService	グリップの開閉を行う。現在はグリップを閉める事しかできない。

#### GripService.idl

```
interface GripService
{
    boolean Open(); //未実装
    boolean Close();
    boolean SetDistance(in double dist); //未実装
};
```

### 3. 4. DummyPlanner Comp（コンポーネント名）

#### 3. 4. 1. 機能概要

シナリオ管理モジュールのダミーとして各モジュールのサービスポートを操作するモジュールである。機能確認のために用意した。

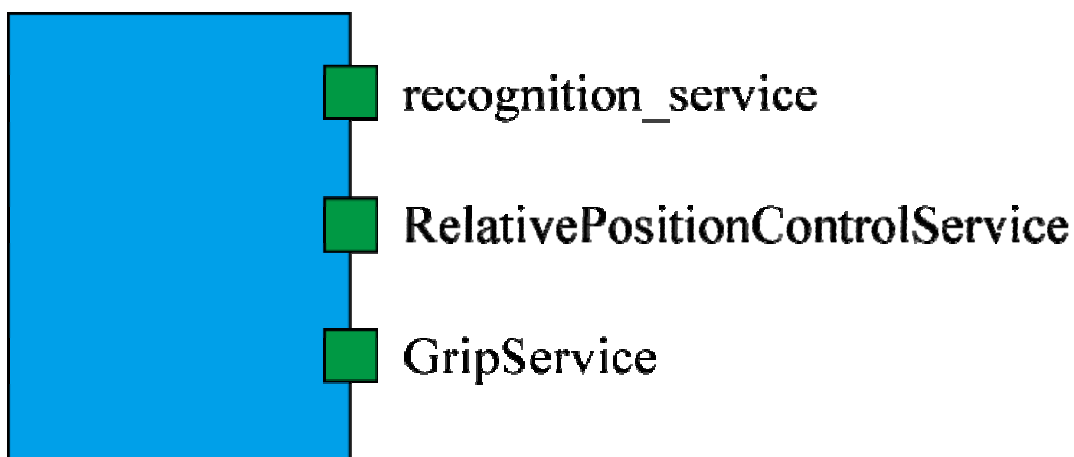


Fig.8 DummyPlanner module

#### 3. 4. 2. 動作環境

表 3-4 動作環境

動作 OS	OS : Ubuntu 10.04 LTS
開発言語	C++
コンパイラ	g++
RT ミドルウェア／バージョン	OpenRTM-aist-1.0.0-RELEASE
依存パッケージ	(OpenHRP 3.1.1)

### 3. 4. 3. ポート情報

#### ServicePort

サービス名	型	説明
recognition_service	RecognitionService	検出するマーカ ID の設定・取得を行う。
RelativePositionControlService	RelativePositionControlService	動作の開始・停止，及び対象物との目標相対位置・姿勢の設定を行う。
GripService	GripService	グリップの開閉を行う。現在はグリップを閉める事しかできない。

#### RecognitionService.idl

```
interface RecognitionService{
    long getModelID();
    void setModelID(in long ModelID);
};
```

#### RelativePositionControlService.idl

```
typedef sequence<double> doubleseq;

interface RelativePositionControlService{
    void start();
    void stop();
    boolean setReferencePose(in doubleseq Pose);
};
```

#### GripService.idl

```
interface GripService
{
    boolean Open(); //未実装
    boolean Close();
    boolean SetDistance(in double dist); //未実装
};
```

- ・ SetReferencePose について

目標相対位置・姿勢の設定を行う関数である。IDL 内で定義している doubleseq 型を用いる。設定例は以下のようになる。

```
doubleseq pose;  
pose.length(12);  
pose[0]=0.0; pose[1]=-1.0; pose[2]=0.0; pose[3]=-30.0; //r11, r12, r13, x[mm]  
pose[4]=-1.0; pose[5]=0.0; pose[6]=0.0; pose[7]=5.0; //r21, r22, r23, y[mm]  
pose[8]=0.0; pose[9]=0.0; pose[10]=-1.0; pose[11]=103.5; //r31, r32, r33, z[mm]
```



## 4. 特記事項

本モジュールは、別添の「エンドユーザー使用許諾契約書」の内容に、ご同意頂いた場合に限りご使用になれます。

※Windows® は、Microsoft Corporation の日本およびその他の国における商標または登録商標です。

※Linux®は、Linus Torvalds 氏の日本およびその他の国における商標または登録商標です。

※Ubuntu®は、Canonical Ltd. の日本およびその他の国における商標または登録商標です。