

次世代ロボット知能化技術開発プロジェクト
ロボット知能ソフトウェア再利用性向上技術の開発

機能仕様書
ロボットアーム分解運動速度制御モジュール
(ACT 共通 I/F 対応版)

Ver. 1.0

2011 年 6 月 30 日

RTC再利用技術研究センター

改版履歷

[illegible]

目次

改版履歴	i
目次	ii
1. はじめに.....	1
1. 1. 本書の適用範囲	1
1. 2. 関連文書	1
1. 3. 本書を読むにあたって	1
2. 機能仕様	2
2. 1. 機能概要	2
2. 2. ターゲットハードウェア	2
2. 3. モジュール構成	2
2. 3. 1. PA10 シミュレータ環境	2
2. 3. 2. PA10 制御	3
2. 3. 3. RH アームシミュレータ環境	4
2. 3. 4. RH アーム制御	5
3. RTC 仕様	6
3. 1. ACT_PA10_RH(中レベル PA10、リファレンスハード制御コンポーネント)	6
3. 1. 1. 機能概要	6
3. 1. 2. 動作環境	8
3. 1. 3. 動作条件	8
3. 1. 4. ポート情報	9
3. 1. 5. サービスインターフェース仕様	10
3. 2. RMRC(分解運動速度制御コンポーネント)	15
3. 2. 1. 機能概要	15
3. 2. 2. 動作環境	15
3. 2. 3. 動作条件	15
3. 2. 4. ポート情報	16
3. 2. 5. コンフィグレーション	17
3. 2. 6. 入出力データフォーマット	18
3. 2. 7. サービスインターフェース仕様	19
3. 2. 8. 依存モジュール	23
3. 3. HandCtrl(RH707 制御モジュール)	24
3. 4. pa10vel(PA10 制御モジュール)	24
3. 5. vel_7dof(7 軸アームシミュレータコンポーネント)	24
3. 5. 1. 機能概要	24
3. 5. 2. 動作環境	24
3. 5. 3. 動作条件	24
3. 5. 4. ポート情報	25

3. 5. 5. 入出力データフォーマット	25
3. 6. pa10disp(PA10 幾何モデル描画コンポーネント)	26
3. 6. 1. 機能概要	26
3. 6. 2. 動作環境	26
3. 6. 3. 動作条件	26
3. 6. 4. ポート情報	26
3. 6. 5. 入出力データフォーマット	26
3. 7. hand_disp(ハンド付き PA10 幾何モデル描画コンポーネント)	27
3. 7. 1. 機能概要	27
3. 7. 2. 動作環境	27
3. 7. 3. 動作条件	27
3. 7. 4. ポート情報	27
3. 7. 5. 入出力データフォーマット	27
3. 8. RHHandWrapper(リファレンスハードグリッパー開閉速度制御コンポーネント)	29
3. 8. 1. 機能概要	29
3. 8. 2. 動作環境	29
3. 8. 3. 動作条件	29
3. 8. 4. ポート情報	30
3. 8. 5. コンフィグレーション情報	31
3. 8. 6. 入出力データフォーマット	31
3. 9. vel_6dof(6 軸アームシミュレータコンポーネント)	33
3. 9. 1. 機能概要	33
3. 9. 2. 動作環境	33
3. 9. 3. 動作条件	33
3. 9. 4. ポート情報	33
3. 9. 5. 入出力データフォーマット	34
3. 10. RH1disp(RH1 幾何モデル描画コンポーネント)	35
3. 10. 1. 機能概要	35
3. 10. 2. 動作環境	35
3. 10. 3. 動作条件	35
3. 10. 4. ポート情報	35
3. 10. 5. 入出力データフォーマット	36
4. 特記事項	37

1. はじめに

1. 1. 本書の適用範囲

本書はロボット向けミドルウェア OpenRTM 上で多軸アームロボットの分解運動速度制御を行う知能モジュールの仕様について記述した文書である。

1. 2. 関連文書

本書の関連文書は 下表の通り。

表 1-1 関連文書

No.	文書名	備考
1	ACT(低レベル)共通インターフェース仕様書	-
2	ACT(中レベル)共通インターフェース仕様書	-
3	機能仕様書 ロボットハンド(RH707)制御モジュール	-
4	機能仕様書 ロボットアーム(PA10)制御モジュール	-
5	RefHard2 アーム制御 RTC	-

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:

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

2. 機能仕様

2. 1. 機能概要

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の作業サブ・ワーキンググループにおいて定められた中レベル ACT RTC の共通インターフェース仕様に基づき、ロボットアームの分解運動速度制御およびハンド制御を実現するモジュールである。

2. 2. ターゲットハードウェア

本知能モジュールがターゲットとするハードウェア構成は以下の通りである。

構成	アーム	ハンド	構成詳細
1	三菱重工業製汎用ロボット PA10 (以下 PA10)	シュンク・ジャパン社製 電動ハンド RH707 (以下 RH707)	関連文書 3、4 に記載
2	知能化モジュール検証用 リファレンスハードウェア ¹ のアーム (以下 RH アーム)	リファレンスハードウェアの アーム先端グリッパ	関連文書 5 に記載

2. 3. モジュール構成

本節では PA10 および RH アームを制御する際のモジュール構成について記載する。

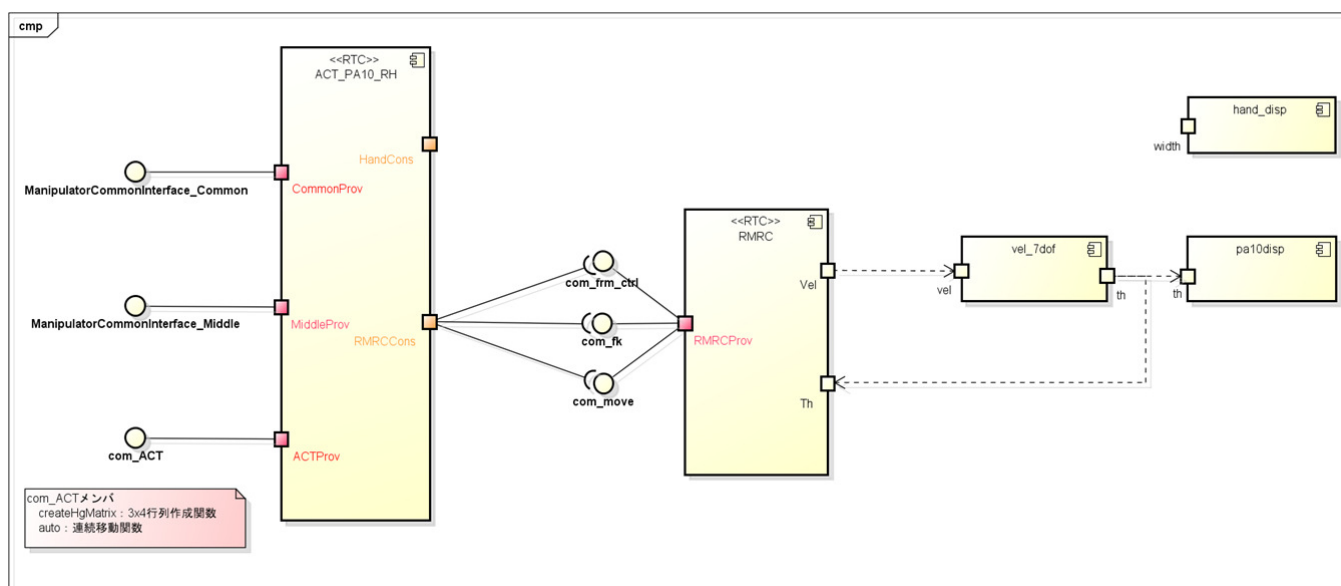
2. 3. 1. PA10 シミュレータ環境

表 2-1 構成 RTC 一覧

No.	構成 RTC	概要
①	ACT_PA10_RH	ACT(中レベル) 共通インターフェース提供 RTC
②	RMRC	分解運動速度制御 RTC
③	vel_7dof	各軸の角度を算出
④	pa10disp	PA10 アームモデルの描画
⑤	hand_disp	PA10 ハンドモデルの描画

¹山下, 柏原, 坂本, 熊沢: "RT コンポーネント検証用としてのリファレンスハードウェアロボットについて", 第10 回計測自動制御学会 システムインテグレーション部門講演会論文集, pp1461-1463, 2009

図 2-1 モジュール構成図

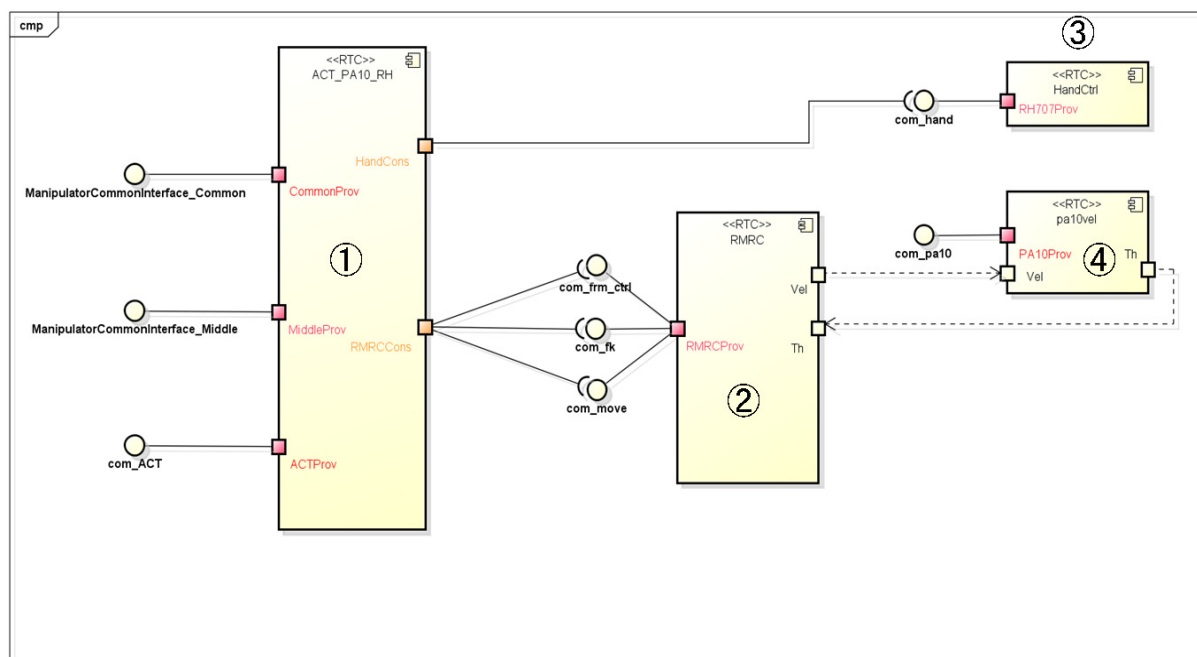


2. 3. 2. PA10 制御

表 2-2 構成 RTC 一覧

No.	構成 RTC	概要
①	ACT_PA10_RH	ACT(中レベル) 共通インターフェース提供 RTC
②	RMRC	分解運動速度制御 RTC
③	HandCtrl	RH707 制御 RTC
④	pa10vel	PA10 制御 RTC

図 2-2 モジュール構成図

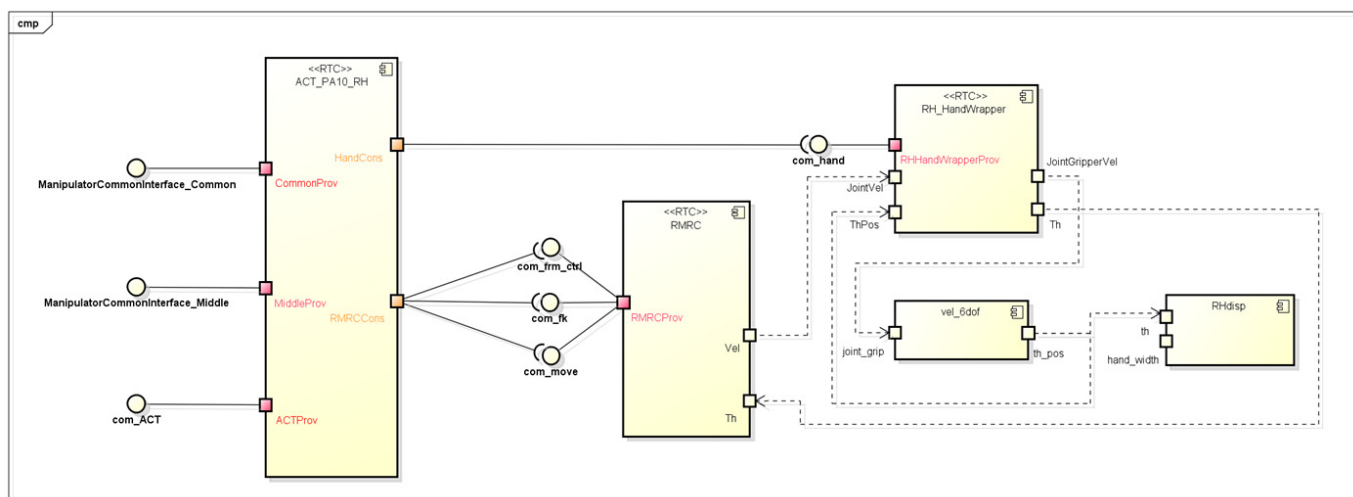


2. 3. 3. RH アームシミュレータ環境

表 2-3 構成 RTC 一覧

No.	構成 RTC	概要
①	ACT_PA10_RH	ACT(中レベル)共通インターフェース提供 RTC
②	RMRC	分解運動速度制御 RTC
③	RH_HandWrapper	リファレンスハードグリッパー開閉速度制御 RTC
④	vel_6dof	各軸の角度を算出
⑤	RHdisp	RH アームモデルの描画

図 2-3 モジュール構成図

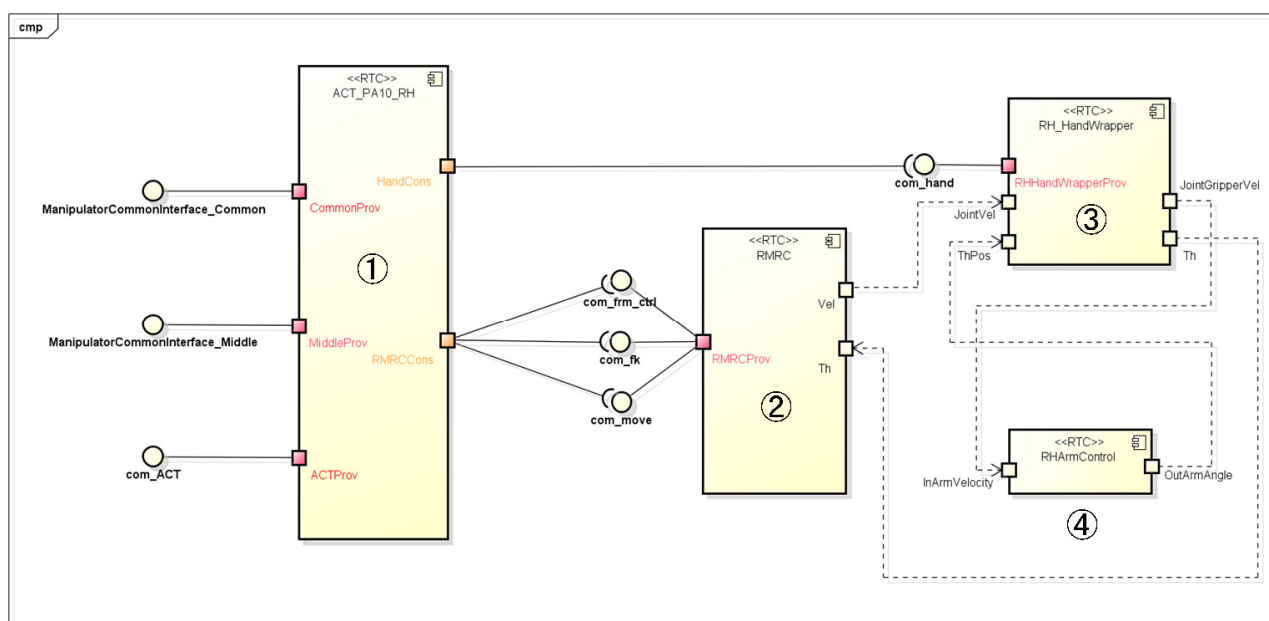


2. 3. 4. RH アーム制御

表 2-4 構成 RTC 一覧

No.	構成 RTC	概要
①	ACT_PA10_RH	ACT(中レベル)共通インターフェース提供 RTC
②	RMRC	分解運動速度制御 RTC
③	RH_HandWrapper	リファレンスハードグリッパー開閉速度制御 RTC
④	RHArmControl	RefHard2 アーム制御 RTC

図 2-4 モジュール構成図



3. RTC 仕様

3. 1. ACT_PA10_RH(中レベル PA10、リファレンスハード制御コンポーネント)

3. 1. 1. 機能概要

本 RTC は NEDO 知能化プロジェクト作業の作業サブ WG において定められている中レベル ACT RTC の共通インターフェースを提供する。インターフェース対応状況は以下の通りとする。

表 3-1 共通コマンドサービスポートのオペレーション

No	オペレーション名	概 要	対応
1	clearAlarms	アラームクリア	×
2	getActiveAlarm	アラーム情報の取得	×
3	getFeedbackPosJoint	関節座標系の位置フィードバック情報の取得	×
4	getManipInfo	マニピュレータ情報の取得	×
5	getSoftLimitJoint	関節座標系のソフトリミット値を取得	×
6	getState	ユニットの状態取得	×
7	servoOFF	全軸サーボ OFF	×
8	servoON	全軸サーボ ON	×
9	setSoftLimitJoint	関節座標系のソフトリミット値設定	×

※共通コマンドについては現在未対応であるが、今後対応することを想定しサービスポートを用意してある。

表 3-2 中レベル・モーションコマンド用サービスポートのオペレーション

No	オペレーション名	概 要	対応
1	closeGripper	グリッパを閉じる 閉口しきっていない場合も一定時間後に動作を完了する	○
2	getBaseOffset	マニピュレータの設置位置を取得	○
3	getFeedbackPosCartesian	直交座標系の位置フィードバック情報の取得	○
4	getMaxSpeedCartesian	直交動作時の最大動作速度を取得	×
5	getMaxSpeedJoint	関節動作時の最大動作速度を取得	×
6	getMinAccelTimeCartesian	直交動作時の最小動作加速時間を取得	×
7	getMinAccelTimeJoint	関節動作時の最小動作加速時間を取得	×
8	getSoftLimitCartesian	直交座標系のソフトリミット値を取得	○
9	moveGripper	グリッパの開閉動作	×
10	moveLinearCartesianAbs	直交座標の直線補間(絶対指令)	○

11	moveLinearCartesianRel	直交座標の直線補間(相対指令)	×
12	movePTPCartesianAbs	関節座標の直線補間(直交・絶対指令)	○
13	movePTPCartesianRel	関節座標の直線補間(直交・相対指令)	×
14	movePTPJointAbs	関節座標の直線補間(関節・絶対指令)	×
15	movePTPJointRel	関節座標の直線補間(関節・相対指令)	×
16	openGripper	グリッパを開く	○
17	pause	動作の一時停止	×
18	resume	動作の再開	×
19	stop	動作の停止	×
20	setAccelTimeCartesian	直交動作時の加速時間を設定	×
21	setAccelTimeJoint	関節動作時の加速時間を設定	×
22	setBaseOffset	マニピュレータの設置位置を設定	×
23	setControlPointOffset	制御点のフランジ面からのオフセット量を設定	○
24	setMaxSpeedCartesian	直交動作時の最大動作速度を設定	×
25	setMaxSpeedJoint	関節動作時の最大動作速度を設定	○
26	setMinAccelTimeCartesian	直交動作時の最小動作加速時間を設定	×
27	setMinAccelTimeJoint	関節動作時の最小動作加速時間を設定	×
28	setSoftLimitCartesian	直交座標系のソフトリミット値を設定	○
29	setSpeedCartesian	直交動作時の速度を設定	×
30	SetSpeedJoint	関節動作時の速度を設定	×

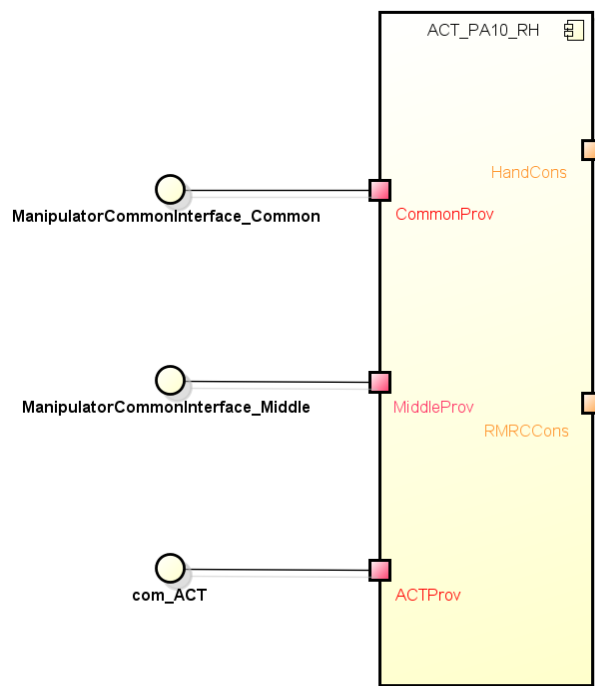
3. 1. 2. 動作環境

動作 OS	Ubuntu10.04
開発言語	C++
コンパイラ	g++4.4.3-1
RT ミドルウェア／バージョン	OpenRTM-aist-1.0.0-RELEASE (C++版)
依存パッケージ(OpenRTM)	OmniORB-4.1.2-1
依存パッケージ(その他)	行列演算ライブラリ

3. 1. 3. 動作条件

実行型	PeriodicExecutionContext
実行周期	デフォルト(1000Hz)

3. 1. 4. ポート情報



A) サービスポート(Provider)

サービスポート名	インターフェース名	説明
CommonProv	ManipulatorCommonInterface_Common	共通コマンド用サービスポート
MiddleProv	ManipulatorCommonInterface_Middle	モーションコマンド用サービスポート
ACTProv	com_ACT	目標手先位置・姿勢データを作成する。

各サービスインターフェースの詳細は3. 1. 5. に記載する。

B) サービスポート(Consumer)

参照するサービスポート(Provider)について記述する。

サービスポート名	インターフェース名	説明
HandCons	com_hand	ハンドの各種制御
RMRCCons	com_frm_ctrl	手先位置・姿勢の軌跡制御
RMRCCons	com_fk	順運動学計算
RMRCCons	com_move	算出した目標手先位置・姿勢を指定

3. 1. 5. サービスインターフェース仕様

3. 1. 5. 1. ManipulatorCommonInterface_Common

A) 基本情報

インスタンス名	m_ManipulatorCommonInterface_Common
変数名	ManipulatorCommonInterface_Common
IDL ファイル	ManipulatorCommonInterface_Common.idl
	ManipulatorCommonInterface_DataTypes.idl
インターフェース型	ManipulatorCommonInterface_Common

本 RTC では低レベルインターフェースに関しては後の拡張用にサービスポートのみを用意しておき、サービス関数の実装は行わないものとする。

3. 1. 5. 2. ManipulatorCommonInterface_Middle

A) 基本情報

インスタンス名	m_ManipulatorCommonInterface_Middle
変数名	ManipulatorCommonInterface_Middle
IDL ファイル	ManipulatorCommonInterface_Middle.idl
	ManipulatorCommonInterface_DataTypes.idl
インターフェース型	ManipulatorCommonInterface_Middle

B) サービス関数一覧

中レベル ACT_RTC の共通インターフェース対応状況は以下の通りとする。

No.	関数名	説明
1	closeGripper	グリッパを閉じる
2	openGripper	グリッパを開く
3	getBaseOffset	マニピュレータの設置位置を取得する
4	getFeedbackPosCartesian	直交座標系の位置フィードバック情報を取得する
5	getSoftLimitCartesian	直交座標系のソフトリミット値を取得する
6	moveLinearCartesianAbs	直交座標の直線補間(絶対指令)をする
7	movePTPCartesianAbs	関節座標の直線補間(直交・絶対指令)をする
8	setControlPointOffset	制御点のフランジ面からのオフセット量を設定する
9	setMaxSpeedJoint	関節動作時の最大動作速度を設定する
10	setSoftLimitCartesian	直交座標系のソフトリミット値を設定する

C) サービス関数詳細

関数名	closeGripper			
引数	名称	型	I/O	説明
	なし	-	-	-
戻り値	型			説明
	RTC::RETURN_ID			関連文書1の 3.1.4 参照
説明	グリッパを閉じる。 閉口しきっていない場合も一定時間後に動作を完了する。			

関数名	openGripper			
引数	名称	型	I/O	説明
	なし	-	-	-
戻り値	型			説明
	RTC::RETURN_ID			関連文書1の 3.1.4 参照
説明	グリッパを開く。			

関数名	getBaseOffset			
引数	名称	型	I/O	説明
	offset	RTC::HgMatrix	出力	関連文書 2 の 5.3.1 参照
戻り値	型			説明
	RTC::RETURN_ID			関連文書1の 3.1.4 参照
説明	マニピュレータの設置位置を取得する。			

関数名	getFeedbackPosCartesian			
引数	名称	型	I/O	説明
	pos	RTC::CarPosWithElbow	出力	関連文書 2 の 5.3.2 参照
戻り値	型			説明
	RTC::RETURN_ID			関連文書1の 3.1.4 参照
説明	直交座標系の位置フィードバック情報を取得する。			

関数名	getSoftLimitCartesian			
引数	名称	型	I/O	説明
	xLimit	RTC::LimitValue	出力	関連文書1の 3.1.3 参照
	yLimit	RTC::LimitValue	出力	関連文書1の 3.1.3 参照
	zLimit	RTC::LimitValue	出力	関連文書1の 3.1.3 参照
戻り値	型			説明
	RTC::RETURN_ID			関連文書1の 3.1.4 参照
説明	直交座標系のソフトリミット値を取得する。			

関数名	moveLinearCartesianAbs			
引数	名称	型	I/O	説明
	carPoint	const RTC::CarPosWithElbow&	入力	関連文書2の 5.3.2 参照
戻り値	型			説明
	RTC::RETURN_ID			関連文書1の 3.1.4 参照
説明	直交座標の直線補間(絶対指令)をする。			

関数名	movePTPCartesianAbs			
引数	名称	型	I/O	説明
	jointPoints	const RTC::JointPos&	入力	関連文書1の 3.1.2 参照
戻り値	型			説明
	RTC::RETURN_ID			関連文書1の 3.1.4 参照
説明	関節座標の直線補間(直交・絶対指令)をする。			

関数名	setControlPointOffset			
引数	名称	型	I/O	説明
	offset	const RTC::HgMatrix&	入力	関連文書2の 5.3.1 参照
戻り値	型			説明
	RTC::RETURN_ID			関連文書1の 3.1.4 参照
説明	制御点のフランジ面からのオフセット量を設定する。			

関数名	setMaxSpeedJoint			
引数	名称	型	I/O	説明
	speed	const RTC::DoubleSeq&	入力	関連文書1の 3.1.1 参照
戻り値	型			説明
	RTC::RETURN_ID			関連文書1の 3.1.4 参照
説明	関節動作時の最大動作速度を設定する。			

関数名	setSoftLimitCartesian			
引数	名称	型	I/O	説明
	xLimit	const RTC::LimitValue&	入力	関連文書1の 3.1.3 参照
	yLimit	const RTC::LimitValue&	入力	関連文書1の 3.1.3 参照
	zLimit	const RTC::LimitValue&	入力	関連文書1の 3.1.3 参照
戻り値	型			説明
	RTC::RETURN_ID			関連文書1の 3.1.4 参照
説明	直交座標系のソフトリミット値を設定する。			

3. 1. 5. 3. com_ACT

A) 基本情報

インスタンス名	m_com_ACT
変数名	com_ACT
IDL ファイル	com_ACT.idl
インターフェース型	ComACT

B) サービス関数一覧

No.	関数名	説明
1	createCarPosWithElbow	CarPosWithElbow 型のデータを作成する。

C) サービス関数詳細

関数名	createCarPosWithElbow			
引数	名称	型	I/O	説明
	xyz	Vector	入力	目標手先位置 [mm]
	abc	Vector	入力	目標手先回転角度 [degree]
戻り値	型			説明
	RTC::CarPosWithElbow			目標手先位置・姿勢を有した CarPosWithElbow 型のデータ
説明	CarPosWithElbow 型のデータを作成する。			

3. 2. RMRC(分解運動速度制御コンポーネント)

3. 2. 1. 機能概要

本知能モジュールはロボットアームの分解運動速度制御を実現するものである。分解運動速度制御とは、手先の並進運動および姿勢の更新を行うために、ロボットアームの各関節の運動速度を制御することである。

本知能モジュールでは、目標とする手先位置・姿勢の指定、動作速度の上限指定、手先に取り付けるツール長とその姿勢の設定を行う機能がサービスとして提供されている。

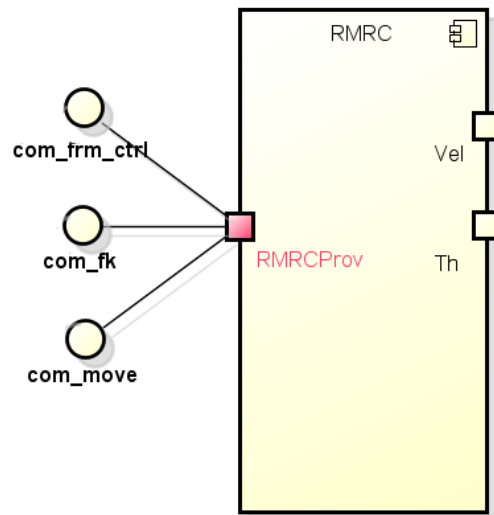
3. 2. 2. 動作環境

動作 OS	Ubuntu10.04
開発言語	C++
コンパイラ	g++4.4.3-1
RT ミドルウェア／バージョン	OpenRTM-aist-1.0.0-RELEASE (C++版)
依存パッケージ(OpenRTM)	OmniORB-4.1.2-1
依存パッケージ(その他)	行列演算ライブラリ
	順運動学計算モジュール

3. 2. 3. 動作条件

実行型	PeriodicExecutionContext
実行周期	デフォルト(1000Hz)

3. 2. 4. ポート情報



A) データポート(InPort)

名称	型	データ長	説明	備考
Th	TimedDoubleSeq	n_size	現在の各軸角度情報 (単位:rad)	コールバック実装※

※データポートコールバック

addConnectorDataListener(リスナタイプ:ON_BUFFER_WRITE)

を用いた実装を行っている。

B) データポート(OutPort)

名称	型	データ長	説明
Vel	TimedDoubleSeq	n_size	各軸の動作角速度 n_size:軸数(単位:rad/sec)

C) サービスポート(Provider)

サービスポート名	インターフェース名	説明
RMRCProv	com_frm_ctrl	手先位置・姿勢の軌跡制御
RMRCProv	com_fk	順運動学計算
RMRCProv	com_move	算出した目標手先位置・姿勢を指定

各サービスインターフェースの詳細は3. 2. 7. に記載する。

3. 2. 5. コンフィグレーション

名称	型	デフォルト値	説明
n_size	int	7	関節数(≥ 1) 本モジュールではアーム自由度のデフォルト値を7とする。
m_max	int	100	特異値分解行列の最大行数(≥ 1)
min_joint_vel	vector<double>	[0.0,0.0,0.0, 0.0,0.0,0.0,0.0]	各関節の最小動作角速度 [rad/sec] データフォーマット: [第1軸, 第2軸, … 第7軸]

RMRCParam.conf に記述されるコンフィグレーションセット

コンフィグレーションセット名	名称	値
PA10	n_size	7
	m_max	100
	min_joint_vel	[0.0,0.0,0.0, 0.0,0.0,0.0,0.0]
RefHardArm	n_size	6
	m_max	100
	min_joint_vel	[0.006,0.005,0.012,0.014,0.031, 0.001]

3. 2. 6. 入出力データフォーマット

3. 2. 6. 1. 入力: **Th**(関節角度)

アームベース面(取り付け面)から1番目の軸を第1軸とする。

データ位置	格納値
0	第1軸の角速度
1	第2軸の角速度
2	第3軸の角速度
:	:
n	第n軸の角速度 (n:n_size)

3. 2. 6. 2. 出力: **Vel**(関節角速度)

アームベース面(取り付け面)から1番目の軸を第1軸とする。

データ位置	格納値
0	第1軸の角速度
1	第2軸の角速度
2	第3軸の角速度
:	:
n	第n軸の角速度 (n:n_size)

3. 2. 7. サービスインターフェース仕様

3. 2. 7. 1. com_frm_ctrl

A) 基本情報

インスタンス名	m_com_frm_ctrl
変数名	com_frm_ctrl
IDL ファイル	com_RMRC.idl
インターフェース型	ComFrmCtrl

B) サービス関数一覧

No.	関数名	説明
1	set_param	並進速度、回転速度計算に使用するパラメータの設定を行う。

C) サービス関数詳細

関数名	set_param			
引数	名称	型	I/O	説明
	k_pv	double	入力	時定数(k_gain)として設定される
	v_m	double	入力	最大速度(v_max)として設定される (単位:mm/sec)
	k_pr	double	入力	-
戻り値	値			説明
	なし			-
説明	並進速度、回転速度計算に使用するパラメータの設定を行う。			
備考	デフォルトでは k_pv = 1.0, v_m = 200, k_pr = 200 であり、基本的に k_pv と k_pr は変更を行わない。必要に応じて最大速度の調整を行う場合には、十分注意して適切なパラメータ設定をすること。			

3. 2. 7. 2. com_fk

A) 基本情報

インスタンス名	m_com_fk
変数名	com_fk
IDL ファイル	com_RMRC.idl
インターフェース型	ComFk

B) サービス関数一覧

No.	関数名	説明
1	set_tool	ツール長と手先座標系から見たツールの姿勢を設定する。
2	get_cur_frm	現在手先位置・姿勢を取得する。
3	get_joint_num	軸数を取得する。

C) サービス関数詳細

関数名	set_tool			
引数	名称	型	I/O	説明
	frm	const Frame&	入力	ツールの位置・姿勢
戻り値	型			説明
	short			0:正常終了 その他:異常終了
説明	ツール長と手先座標系から見たツールの姿勢を設定する。			

Frame 型の IDL 定義 (com_RMRC.idl 内)

```

struct Frame{
    float mat[3][3];
    float pos[3];           //[mm]
}

```

関数名	get_cur_frm			
引数	名称	型	I/O	説明
	なし	-	-	-
戻り値	型			説明
	Frame			現在の手先の位置・姿勢
説明	現在の手先の位置・姿勢を取得する。			

関数名	get_joint_num			
引数	名称	型	I/O	説明
	なし	-	-	-
戻り値	型			説明
	short			軸数
説明	軸数を取得する。			

3. 2. 7. 3. com_move

A) 基本情報

インスタンス名	m_com_move
変数名	com_move
IDL ファイル	com_RMRC.idl
インターフェース型	ComMove

B) サービス関数一覧

No.	関数名	説明
1	move_to	分解運動速度制御のトリガーであり、目標手先位置・姿勢を指定する。
2	set_max_joint_vel	最大動作角速度を設定する。
3	move_joint	各軸の角度制御を行う。
4	get_softlimit	直行座標系におけるソフトリミット(可動範囲)を取得する。
5	set_softlimit	直行座標系におけるソフトリミット(可動範囲)の設定を行う。

C) サービス関数詳細

関数名	move_to			
引数	名称	型	I/O	説明
	frm	const Frame&	入力	手先の目標位置[mm]と姿勢
戻り値	型			説明
	short			0:正常終了 その他:異常終了
説明	分解運動速度制御のトリガーであり、目標手先位置・姿勢を指定する。			

関数名	set_max_joint_vel			
引数	名称	型	I/O	説明
	max_vel	const DoubleSeq&	入力	各軸最大動作角速度[rad/sec]
戻り値	型			説明
	short			0:正常終了 その他:異常終了
説明	最大動作角速度を設定する。			

DoubleSeq 型の IDL 定義 (com_RMRC.idl 内)

```
struct DoubleSeq {
    sequence<double> data // [rad/sec]
}
```

関数名	move_joint			
引数	名称	型	I/O	説明
	ref_angles	const DoubleSeq&	入力	各軸の目標角度[rad]
戻り値	型			説明
	short			0:正常終了 その他:異常終了
説明	角度制御により各軸を目標角度へ移行させる。			

関数名	get_softlimit			
引数	名称	型	I/O	説明
	x_max	double	出力	X 軸のソフト上限値 [mm]
	y_max	double	出力	Y 軸のソフト上限値 [mm]
	z_max	double	出力	Z 軸のソフト上限値 [mm]
	x_min	double	出力	X 軸のソフト下限値 [mm]
	y_min	double	出力	Y 軸のソフト下限値 [mm]
	z_min	double	出力	Z 軸のソフト下限値 [mm]
戻り値	型			説明
	short			0:正常終了 その他:異常終了
説明	直行座標系におけるソフトリミット(可動範囲)を取得する。			

関数名	set_softlimit			
引数	名称	型	I/O	説明
	x_max	double	入力	X 軸のソフト上限値 [mm]
	y_max	double	入力	Y 軸のソフト上限値 [mm]
	z_max	double	入力	Z 軸のソフト上限値 [mm]
	x_min	double	入力	X 軸のソフト下限値 [mm]
	y_min	double	入力	Y 軸のソフト下限値 [mm]
	z_min	double	入力	Z 軸のソフト下限値 [mm]
戻り値	型			説明
	short			0:正常終了 その他:異常終了
説明	直行座標系におけるソフトリミット(可動範囲)を設定する。			

3. 2. 8. 依存モジュール

3. 2. 8. 1. 行列演算ライブラリ

モジュール名	行列演算ライブラリ
ファイル/ヘッダ名	geo_std.o
機能概要	行列演算に関する関数群を定義している。

3. 2. 8. 2. 順運動学計算モジュール

モジュール名	順運動学計算モジュール
ファイル/ヘッダ名	fk.cpp / fk.h
機能概要	本ファイル内でロボットアームの関節数、リンク長を設定し、その情報に基づきアームの順運動学計算を行う。

現在は本モジュールにおいてロボットアーム固有のパラメータ設定や、順運動学計算の記述がされている。本モジュールはロボットアーム毎に各自でファイルを用意するものとする。制御するロボットアームを変更する際は、適宜対応するファイルに差し替え、リコンパイルし、RMRCComp オブジェクトを再度生成する。

3. 3. HandCtrl(RH707 制御モジュール)

本 RTC 詳細に関しては関連文書 3 を参照のこと。

3. 4. pa10vel(PA10 制御モジュール)

本 RTC 詳細に関しては関連文書 4 を参照のこと。

3. 5. vel_7dof(7 軸アームシミュレータコンポーネント)

3. 5. 1. 機能概要

本コンポーネントは逆運動学計算に基づいた7自由度アームの動作シミュレーションを行う。尚、幾何モデル表示機能は別コンポーネント(pa10_disp)で用意してある。

3. 5. 2. 動作環境

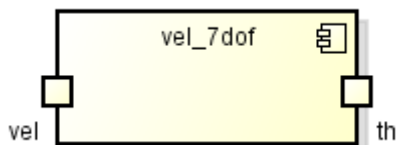
コンポーネント動作環境を以下に示す。

動作 OS	Ubuntu10.04
開発言語	C++
コンパイラ	g++4.4.3-1
RT ミドルウェア／バージョン	OpenRTM-aist-1.0.0-RELEASE(C++版)
依存ライブラリ(OpenRTM)	OmniORB-4.1.2-1

3. 5. 3. 動作条件

実行型	PeriodicExecutionContext
実行周期	デフォルト(1000Hz)

3. 5. 4. ポート情報



A) データポート(InPort)

名称	型	データ長	説明
vel	TimedDoubleSeq	7	各軸の動作角速度 (単位:rad/sec)

B) データポート(OutPort)

名称	型	データ長	説明
th	TimedDoubleSeq	7	pa10_disp へ出力する関節角度(単位:rad)

3. 5. 5. 入出力データフォーマット

3. 5. 5. 1. 入力:**vel**(関節角速度)(3. 2. 6. 2. と同様)

3. 5. 5. 2. 出力:**th**(関節角度)(3. 2. 6. 1. と同様)

3. 6. pa10disp (PA10 幾何モデル描画コンポーネント)

3. 6. 1. 機能概要

本コンポーネントは PA10 シミュレータの一部であり、PA10 幾何モデルの表示を行う。

3. 6. 2. 動作環境

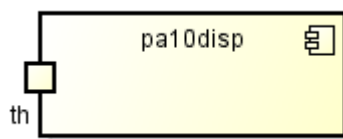
コンポーネント動作環境を以下に示す。

動作 OS	Ubuntu10.04
開発言語	Python
インタプリタ	Python2.6
RT ミドルウェア／バージョン	OpenRTM-aist-Python-1.0.0-RELEASE
依存ライブラリ (OpenRTM)	OmniORBpy-3.2-1
依存ライブラリ (その他)	python visual-5.12-1.1

3. 6. 3. 動作条件

実行型	PeriodicExecutionContext
実行周期	デフォルト (1000Hz)

3. 6. 4. ポート情報



A) データポート (InPort)

名称	型	データ長	説明
th	TimedDoubleSeq	7	関節角度 (単位: rad)

3. 6. 5. 入出力データフォーマット

3. 6. 5. 1. 入力: th (3. 2. 6. 1. と同様)

3. 7. hand_disp(ハンド付き PA10 幾何モデル描画コンポーネント)

3. 7. 1. 機能概要

本コンポーネントは PA10 シミュレータの一部であり、PA10 ツールとして二指のハンド幾何モデルを描画する。

3. 7. 2. 動作環境

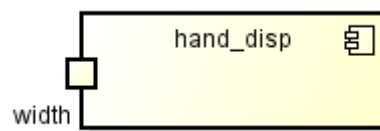
コンポーネント動作環境を以下に示す。

動作 OS	Ubuntu10.04
開発言語	Python
インタプリタ	Python2.6
RT ミドルウェア／バージョン	OpenRTM-aist-Python-1.0.0-RELEASE
依存ライブラリ(OpenRTM)	OmniORBpy-3.2-1
依存ライブラリ(その他)	python visual-5.12-1.1

3. 7. 3. 動作条件

実行型	PeriodicExecutionContext
実行周期	デフォルト(1000Hz)

3. 7. 4. ポート情報



A) データポート(InPort)

名称	型	データ長	説明
width	TimedDouble	1	開口幅(単位:m)

3. 7. 5. 入出力データフォーマット

3. 7. 5. 1. 入力:th

データ位置	格納値
0	ハンド開口幅

3. 8. RHHandWrapper (リファレンスハードグリッパー開閉速度制御コンポーネント)

3. 8. 1. 機能概要

本知能モジュールは、多軸ロボットアーム分解運動速度制御モジュールと共に用いて、RH のアーム先端に取り付けられたハンドのグリッパー開閉速度制御を行うための、ラッパーモジュールである。

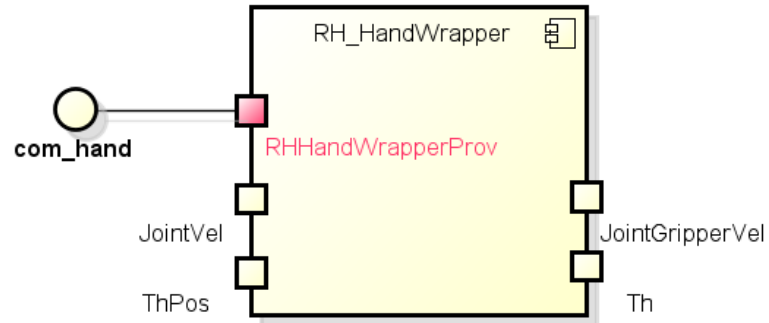
3. 8. 2. 動作環境

動作 OS	Ubuntu10.04
開発言語	C++
コンパイラ	g++4.4.3-1
RT ミドルウェア / バージョン	OpenRTM-aist-1.0.0-RELEASE (C++版)
依存パッケージ (OpenRTM)	OmniORB-4.1.2-1

3. 8. 3. 動作条件

実行型	PeriodicExecutionContext
実行周期	デフォルト (1000Hz)

3. 8. 4. ポート情報



A) データポート(InPort)

名称	型	データ長	説明	備考
JointVel	TimedDoubleSeq	jointnum	アーム関節角速度 (単位:rad/sec)	コールバック実装※
ThPos	TimedDoubleSeq	jointnum+1	関節角度(単位:rad) ツールの開口幅(単位:m)	コールバック実装※

※データポートコールバック

addConnectorDataListener(リスナタイプ:ON_BUFFER_WRITE)

を用いた実装を行っている。

B) データポート(OutPort)

名称	型	データ長	説明
Th	TimedDoubleSeq	jointnum	現在のアーム関節角度(単位:rad)
JointGripVel	TimedDoubleSeq	jointnum+1	アーム関節角速度(単位:rad/sec) ツールの開閉速度(単位:m/sec)

C) サービスポート(Provider)

サービスポート名	インターフェース名	説明
ComHand	com_hand	ハンドの各種制御

各サービスインターフェースは HandCtrl と同一になっている。

詳細は関連文書 3 を参照のこと。

3. 8. 5. コンフィグレーション情報

名称	型	デフォルト値	説明
max_width	double	0.07	最大開口幅(単位:m)
grip_vel	double	0.01	開閉速度(単位:m/sec)
jointnum	int	6	アーム関節数(≥ 1) 本モジュールではアーム自由度のデフォルト値を6とする。

3. 8. 6. 入出力データフォーマット

3. 8. 6. 1. 入力: **JointVel**(アーム関節角速度)

アームベース面(取り付け面)から1番目の軸をJ1軸とする。

データ位置	格納値
0	J1 軸の角度
1	J2 軸の角度
2	J3 軸の角度
3	J4 軸の角度
4	J5 軸の角度
5	J6 軸の角度

3. 8. 6. 2. 入力: **ThPos**(関節角度+ハンド開閉幅)

アームベース面(取り付け面)から1番目の軸をJ1軸とする。

データ位置	格納値
0	J1 軸の角度
1	J2 軸の角度
2	J3 軸の角度
3	J4 軸の角度
4	J5 軸の角度
5	J6 軸の角度
6	ツールの開口幅

3. 8. 6. 3. 出力: **JointGripVel**(アーム関節角速度+ハンド開閉速度)

アームベース面(取り付け面)から 1 番目の軸を J1 軸とする。

データ位置	格納値
0	J1 軸の角速度
1	J2 軸の角速度
2	J3 軸の角速度
3	J4 軸の角速度
4	J5 軸の角速度
5	J6 軸の角速度
6	ツールの開口幅

3. 8. 6. 4. 出力: **Th**(アーム関節角度)

アームベース面(取り付け面)から 1 番目の軸を J1 軸とする。

データ位置	格納値
0	J1 軸の角度
1	J2 軸の角度
2	J3 軸の角度
3	J4 軸の角度
4	J5 軸の角度
5	J6 軸の角度

3. 9. vel_6dof(6 軸アームシミュレータコンポーネント)

3. 9. 1. 機能概要

本コンポーネントは、順運動学計算に基づいた6自由度アームのシミュレーションを行う。尚、幾何モデル表示機能は別コンポーネント(RH1disp)で用意してある。

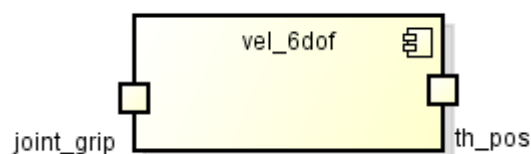
3. 9. 2. 動作環境

動作 OS	Ubuntu10.04
開発言語	C++
コンパイラ	g++4.4.3-1
RT ミドルウェア / バージョン	OpenRTM-aist-1.0.0-RELEASE (C++版)
依存パッケージ (OpenRTM)	OmniORB-4.1.2-1

3. 9. 3. 動作条件

実行型	PeriodicExecutionContext
実行周期	デフォルト (1000Hz)

3. 9. 4. ポート情報



A) データポート (InPort)

名称	型	データ長	説明
joint_grip	TimedDoubleSeq	指定なし	各軸の動作角速度 (単位: rad/sec) ツールの開閉速度 (単位: mm/sec)

B) データポート(OutPort)

名称	型	データ長	説明
th_pos	TimedDoubleSeq	7	関節角度(単位:rad) ツールの開口幅(単位:mm)

3. 9. 5. 入出力データフォーマット

3. 9. 5. 1. 入力:**joint_grip**(3. 8. 6. 3. と同様)3. 9. 5. 2. 出力:**th_pos**(3. 8. 6. 2. と同様)

3. 10. RH1disp(RH1 幾何モデル描画コンポーネント)

3. 10. 1. 機能概要

本コンポーネントは、シミュレータの一部であり、RH1 幾何モデルの表示を行う。

3. 10. 2. 動作環境

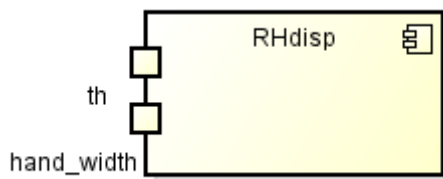
コンポーネント動作環境を以下に示す。

動作 OS	Ubuntu10.04
開発言語	C++
コンパイラ	g++4.4.3-1
RT ミドルウェア／バージョン	OpenRTM-aist-1.0.0-RELEASE (C++版)
依存パッケージ(OpenRTM)	OmniORB-4.1.2-1

3. 10. 3. 動作条件

実行型	PeriodicExecutionContext
実行周期	デフォルト(1000Hz)

3. 10. 4. ポート情報



A) データポート(InPort)

名称	型	データ長	説明
th	TimedDoubleSeq	指定なし	関節角度(単位:rad) ツールの開口幅(単位:mm)
hand_width	TimedDouble	1	ツールの開口幅(単位:mm)

3. 10. 5. 入出力データフォーマット

3. 10. 5. 1. 入力: **th** (3. 8. 6. 2. と同様)

3. 10. 5. 2. 入力: **hand_width**

データ位置	格納値
0	ハンドの開閉幅

4. 特記事項

本モジュールをご利用される場合には、以下の記載事項・条件にご同意いただいたものとします。

- ドキュメントに情報を掲載する際には万全を期していますが、それらの情報の正確性またはお客様にとっての有用性等については一切保証いたしません。
- 利用者が本モジュールを利用することにより生じたいかなる損害についても一切責任を負いません。
- 本モジュールの変更、削除等は、原則として利用者への予告なしに行います。また、止むを得ない事由により公開を中断あるいは中止させていただくことがあります。
- 本モジュールの情報の変更、削除、公開の中断、中止により、利用者に生じたいかなる損害についても一切責任を負いません。
- 本モジュールは独立行政法人 新エネルギー・産業技術総合開発機構(NEDO 技術開発機構)の「次世代ロボット知能化技術開発プロジェクト」(平成 19 年～平成 23 年度)において、評価を目的として開発されたものであり、商用以外の利用の場合、BSD ライセンスが適用されます。詳しくは同封の LICENSE-BSD.TXT を参照ください。
- 商用利用の際には連絡を要し、使用条件は個別に検討するものとします。

【連絡先】

RTC 再利用技術研究センター

〒101-0021 東京都千代田区外神田 1-18-13 秋葉原ダイビル 1303 号室

Tel/Fax: 03-3256-6353 E-Mail: contact@rtc-center.jp