

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(PA10)分解運動速度制御モジュール
(Linux)

V e r . 1 . 0

2010年3月23日

R T C 再利用技術研究センター

目次

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. モジュール構成	3
2. 3. ターゲットハードウェア	8
3. RTC 仕様	11
3. 1. frm_ctrl (軌跡制御コンポーネント)	11
3. 2. tr_jacob (ヤコビ行列変換コンポーネント)	16
3. 3. mixer (重み付けコンポーネント)	19
3. 4. jinv (逆ヤコビ行列計算コンポーネント)	23
3. 5. vel_7dof (7 軸アームシミュレータコンポーネント)	25
3. 6. pa10fk (PA10 順運動学計算コンポーネント)	27
3. 7. coord_trans (座標系変換コンポーネント)	30
3. 8. move (操作制御コンポーネント)	32
3. 9. pa10disp (PA10 幾何モデル描画コンポーネント)	34
3. 10. slider(データ送出コンポーネント)	36
3. 11. j_jacob(ヤコビシーケンス生成コンポーネント)	38
3. 12. pa10vel (PA10 実機制御コンポーネント)	40
4. 特記事項	42

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 上で三菱重工業製汎用ロボット PA10 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	ロボットアーム(PA10)分解運動速度制御モジュール (Linux) 操作手順書	ロボットアーム(PA10)分解運動速度制御モジュールのシミュレータ及び実機環境の操作手順について記載。

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.is.aist.go.jp/rt/OpenRTM-aist/>

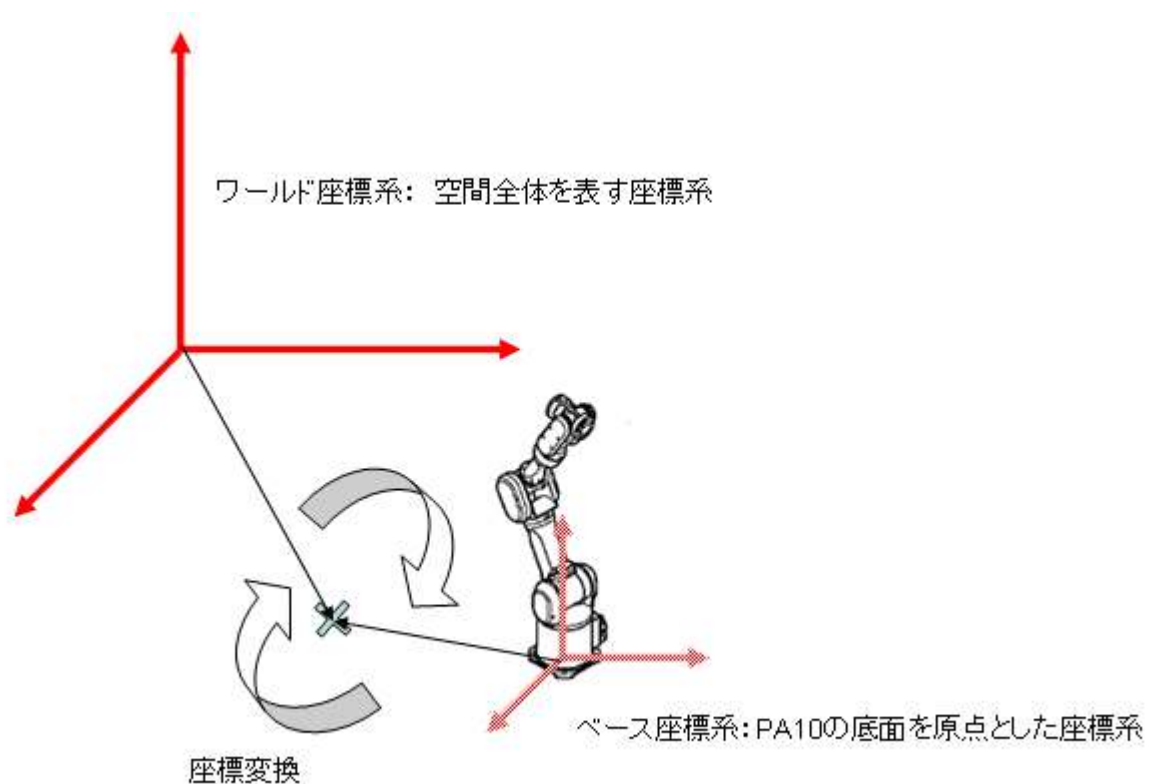
2. 機能仕様

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。また本モジュールでは逆運動学計算によって決めた分解運動速度へのセンサーフィードバックを行うための機能が用意されている。

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。また目標値の入力に関して、逆運動学計算に用いられる座標系（ベース座標系）とは異なる座標系¹（例えばワールド座標系）からの入力が可能となっている。

尚、今回はターゲットハードウェアとして三菱重工業製汎用ロボット PA10（以下 PA10）を使用している。



¹ 通常、ロボットアームはある座標系（ワールド座標系）上のどこかに位置しており、ロボットアームへ与える目標値はその座標系での位置・姿勢である場合が多い。しかし、順（逆）運動学計算を行う際はロボットアームを中心に置いたベース座標系で考えた方が便利であるため、座標系の変換を行う。

2. 2. モジュール構成

2. 2. 1. シミュレータ(VPython)動作環境

シミュレータ動作時の本知能モジュール（分解運動速度制御モード）は図 2-1 に表されるように、

- ① 軌跡制御コンポーネント (frm_ctrl)
- ② ヤコビ行列変換コンポーネント (tr_jacob)
- ③ 重み付けコンポーネント (mixer)
- ④ 逆ヤコビ行列計算コンポーネント (jinv)
- ⑤ 7 軸アームシミュレータコンポーネント (vel_7dof)
- ⑥ PA10 順運動学計算コンポーネント (pa10fk)
- ⑦ 座標変換コンポーネント (coord_trans)
- ⑧ 操作制御コンポーネント (move)
- ⑨ PA10 幾何モデル描画コンポーネント (pa10disp)
- ⑩ 軸データ送出コンポーネント (slider)
- ⑪ ヤコビシーケンス生成コンポーネント (j_jacob)

で構成される。手先の目標位置・姿勢の設定は操作制御コンポーネントで、動作速度の上限設定は軌跡制御コンポーネントで、手先ツール長の設定は PA10 順運動学計算コンポーネントでサービスとして提供されている。図 2-1 シミュレータ接続時モジュール構成図中の点線で囲んだ箇所は動作環境に合わせて付け替える部分であり、ここでは VPython によるシミュレータ環境を接続したものである。図 2-2 が VPython シミュレータ表示画面になる。

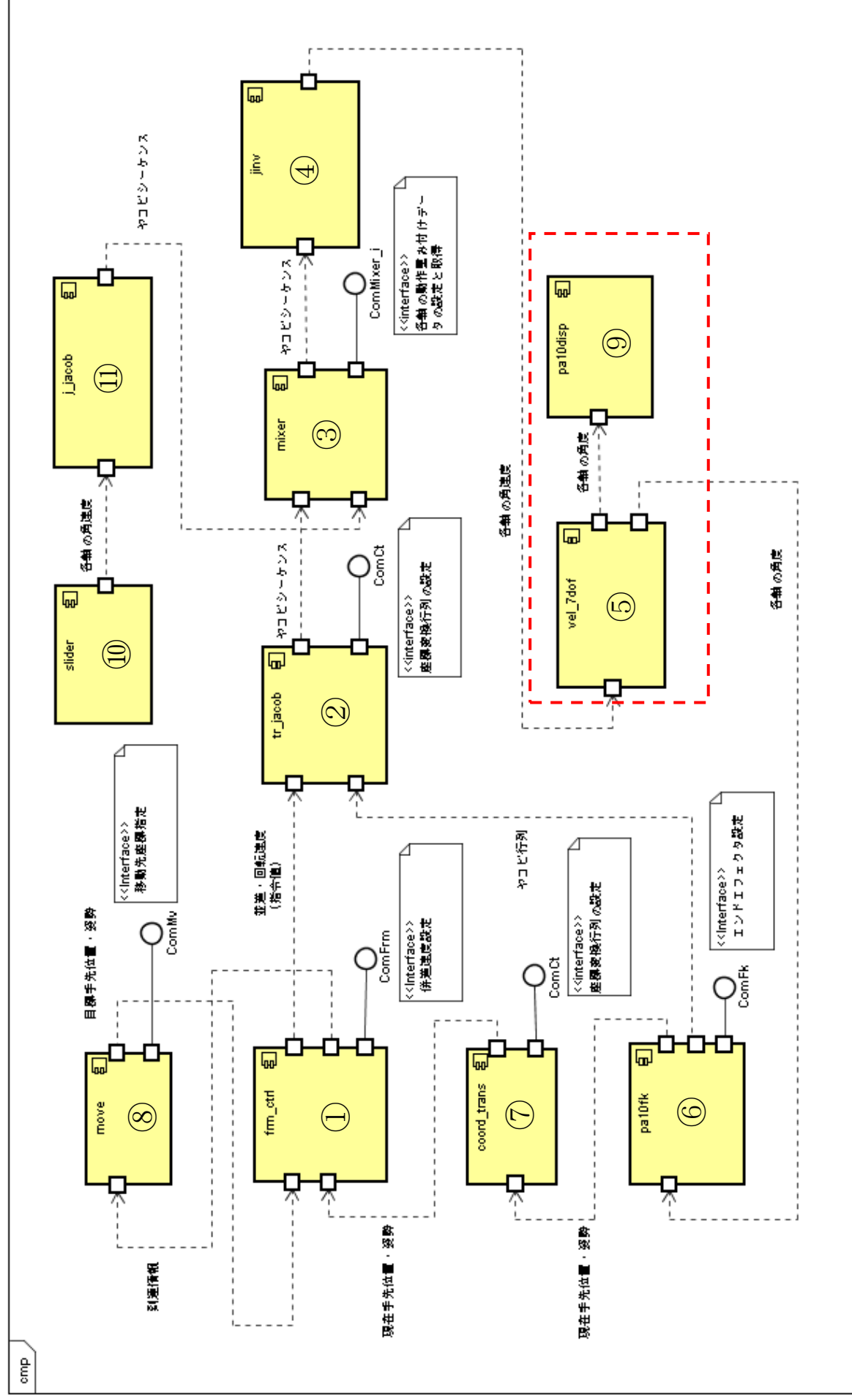


図 2-1 シミュレータ接続時モジュール構成図 (分解運動速度制御モード)

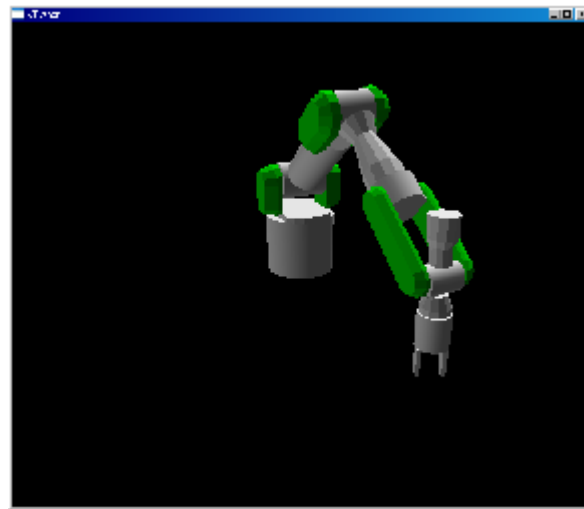


図 2-2 VPython シミュレータ表示画面

2. 2. 2. 実機動作環境

実機動作時のモジュール構成（分解運動速度制御モード）は、前節のシミュレータ関連コンポーネント（⑤、⑨）を実機制御コンポーネント（⑫）に置き換えたものになる。図 2-3 に実機動作時モジュール構成図を示す。図 2-3 実機動作時モジュール構成図と図 2-1 シミュレータ接続時モジュール構成図の違いは図中の点線で囲んだ部分のみである。本モジュールに接続した実機の写真を図 2-4 接続した実機に示す。

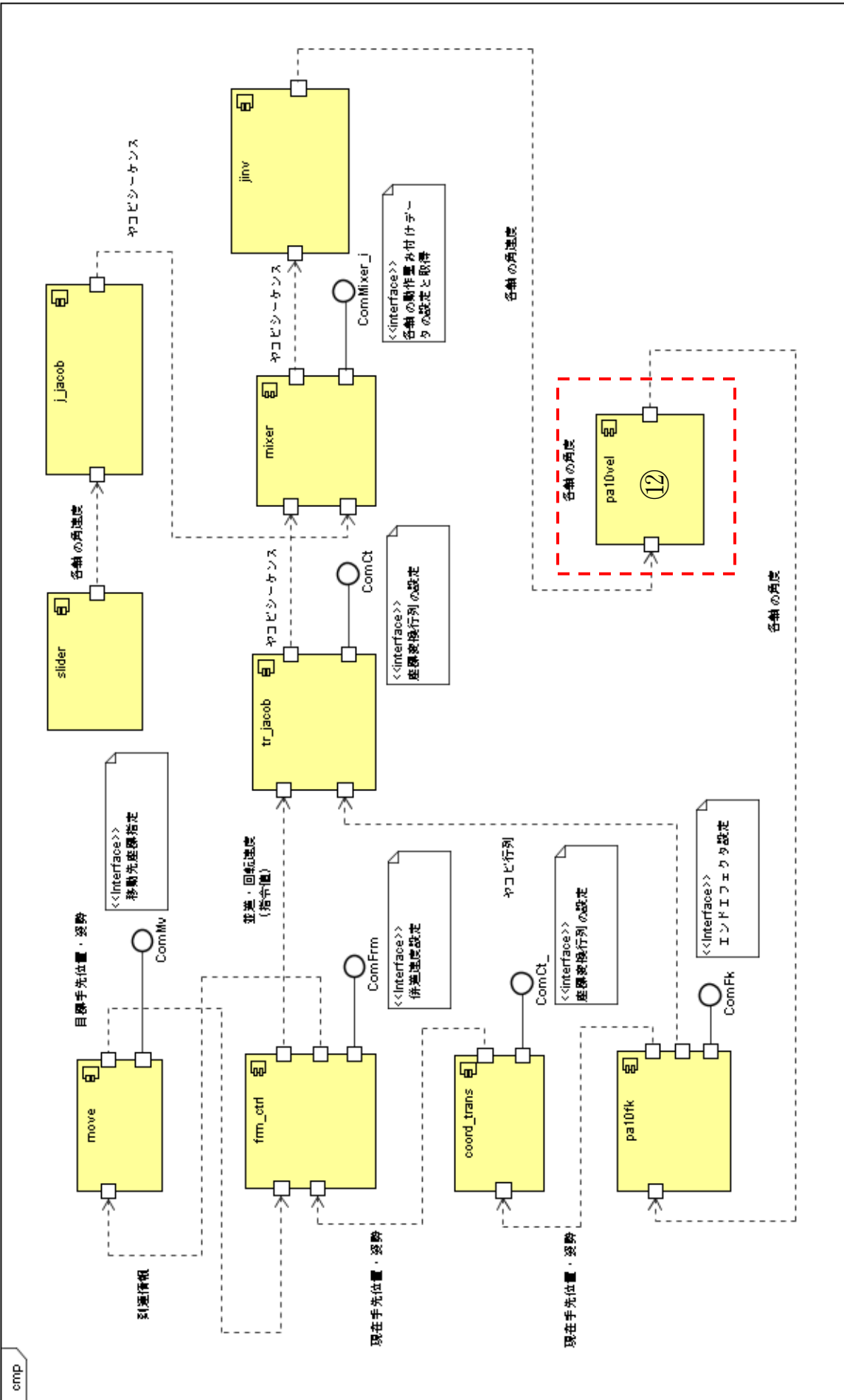


図 2-3 実機動作時モジュール構成図 (分解運動速度制御モード)



図 2-4 接続した実機

2. 3. ターゲットハードウェア

本知能モジュールの対象となるハードウェアは、三菱重工業製汎用ロボット PA10 である。ハードウェア構成及びその仕様は以下の通りである。

2. 3. 1. ハードウェア構成

表記： ——— 接続機器 ハードウェア

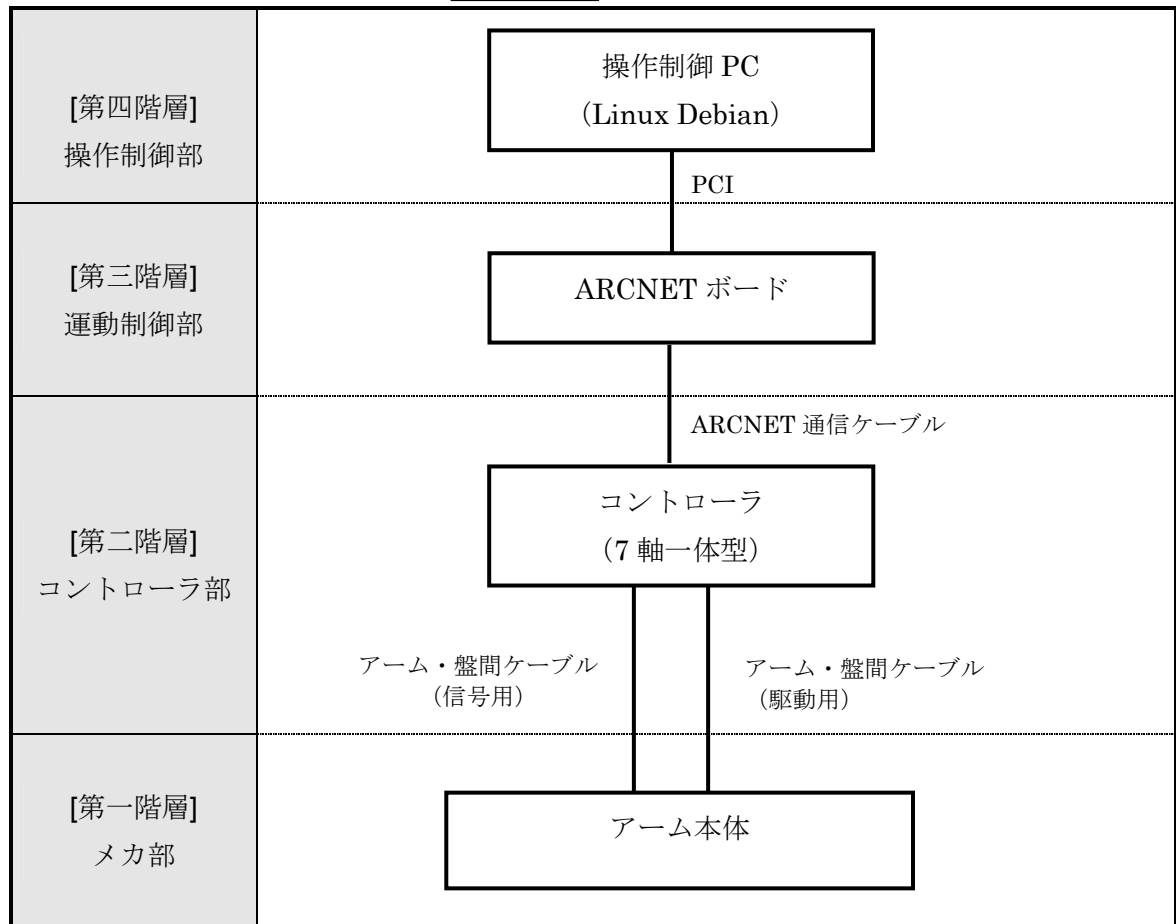


図 2-5 ハードウェア構成



図 2-6 アーム本体

2. 3. 2. ハードウェア仕様

2. 3. 2. 1. アーム本体仕様

表 2-1 アーム仕様

項目	内容				
名称	三菱重工汎用ロボット アーム				
型 名	PA10-7C-ARM				
関節数	7				
関節構成	ロボット取り付け面より R-P-R-P-R-P-R (R は回転関節, P は旋回関節を示す)				
関節名称	ロボット取り付け面より S1-S2-S3-E1-E2-W1-W2 (Sは肩関節, Eは肘関節, Wは手首関節を表す)				
アーム長	肩リーチ : 317mm (ベース面～S2 間) 上腕 : 450mm (S2～E1 軸間) 下腕 : 480mm (E1～W1 軸間) 手首リーチ : 80mm (W1～メカニカルインタフェース面間)				
関節動作範囲 及び 最高動作速度	軸名称	リミット (度)			
		メカリミット	サーボ リミット	ソフトウェア リミット	最高動作速度 (rad/sec)
	S1 (回転)	±180	±178	±177	±1
	S2 (旋回)	± 97	± 95	±94	±1
	S3 (回転)	±180	±175	±174	±1
	E1 (旋回)	±143	±138	±137	±2
	E2 (回転)	±270	±256	±255	±2π
	W1 (旋回)	±180	±166	±165	±2π
	W2 (回転)	±270	±256	±255	±2π

(詳細は「三菱重工 汎用ロボット PA10 シリーズ 総合取扱説明書 P2-2」を参照)

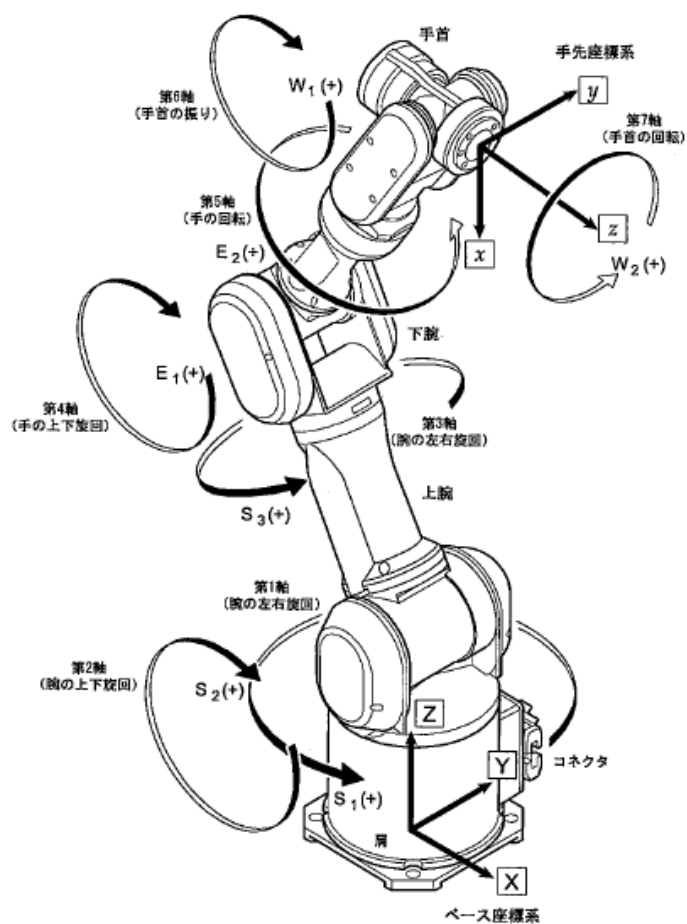


図 2-7 アームの動作軸と軸名称（総合取扱説明書より）

2. 3. 2. 2. コントローラ・運動制御ボード仕様

表 2-2 コントローラ・運動制御ボード仕様

項目	型番
コントローラ	PA10-7C-CNT
アーム I/F	ライフロボティクス社製 PA10 高速制御装置

（詳細は「三菱重工 汎用ロボット PA10 シリーズ 総合取扱説明書 P2-13,16(コントローラ)」、
「ライフロボティクス PA10 用制御ライブラリ 取扱説明書 (PA-10、1ms 制御用マニュアル ver.1.1)」を参照）

3. RTC 仕様

3. 1. frm_ctrl（軌跡制御コンポーネント）

3. 1. 1. 機能概要

本コンポーネントはロボットアームの手先位置・姿勢の軌跡制御を行う。手先の現在位置・姿勢及び手先の目標位置・姿勢から、ロボットアームの動作完了の判定、または、ロボットアームの手先並進速度、回転速度の算出を行う。

本コンポーネントでは速度計算に必要なパラメータの設定も行う。

3. 1. 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	Ubuntu8.04
開発言語	C++
コンパイラ	g++4.2.3-1
RT ミドルウェア／バージョン	OpenRTM-aist-0.4.2-RELEASE（C++版）
依存ライブラリ (OpenRTM)	OmniORB-4.1.1-2
	ACE-5.4.7-13
依存ライブラリ(その他)	行列演算ライブラリ（自作）

動作 OS	Debian4.0r4
開発言語	C++
コンパイラ	g++4.1.1-21
RT ミドルウェア／バージョン	OpenRTM-aist-0.4.2-RELEASE（C++版）
依存ライブラリ (OpenRTM)	OmniORB-4.0.6-2.1
	ACE-5.4.7-12
依存ライブラリ(その他)	行列演算ライブラリ（自作）

3. 1. 3. ポート情報

A) データポート (InPort)

名称	型	データ長	説明
ref_frm	TimedDoubleSeq	12	目標手先位置 (単位 : mm) ・ 姿勢
cur_frm	TimedDoubleSeq	12	現在手先位置 (単位 : mm) ・ 姿勢

B) データポート (OutPort)

名称	型	データ長	説明
vel	TimedDoubleSeq	6	並進速度 (単位 : mm/sec) 回転速度 (単位 : rad/sec)
state	TimedLong	1	到達情報 (1 : 到着 0 : 未到着)

C) サービスポート (Provider)

サービス名	インターフェース名	説明
ComFrmCtrl	com_frm_ctrl	手先位置・姿勢の軌跡制御

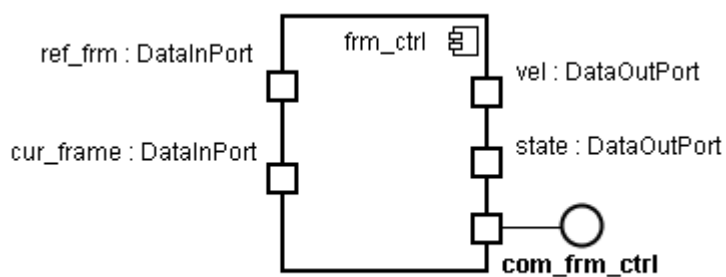


図 3-1 frm_ctrl コンポーネント

3. 1. 4. 入出力データフォーマット

3. 1. 4. 1. 入力 : **ref_frm**、**cur_frm**

データ位置	格納値
0	手先の姿勢を表す回転行列の 1 行 1 列目
1	手先の姿勢を表す回転行列の 1 行 2 列目
2	手先の姿勢を表す回転行列の 1 行 3 列目
3	手先の姿勢を表す回転行列の 2 行 1 列目
4	手先の姿勢を表す回転行列の 2 行 2 列目
5	手先の姿勢を表す回転行列の 2 行 3 列目
6	手先の姿勢を表す回転行列の 3 行 1 列目
7	手先の姿勢を表す回転行列の 3 行 2 列目
8	手先の姿勢を表す回転行列の 3 行 3 列目
9	手先の位置ベクトルの X 成分 (任意の座標系)
1 0	手先の位置ベクトルの Y 成分 (任意の座標系)
1 1	手先の位置ベクトルの Z 成分 (任意の座標系)

3. 1. 4. 2. 出力 : **vel** (指令値)

データ位置	格納値
0	手先の x 軸方向への並進速度
1	手先の y 軸方向への並進速度
2	手先の z 軸方向への並進速度
3	手先の x 軸回りの回転速度
4	手先の y 軸回りの回転速度
5	手先の z 軸回りの回転速度

3. 1. 4. 3. 出力 : **state**

データ位置	格納値
0	到達情報

3. 1. 5. サービスポート I/F 仕様

3. 1. 5. 1. ComFrmCtrl

(1) set_param

関数名	set_param			
引数	名称	型	I/O	説明
	k_pv	float	入力	時定数 (k_gain) として設定される
	v_m	float	入力	最大速度 (v_max) として設定される (単位 : mm/sec)
	k_pr	float	入力	並進・回転換算比率 (k_pos_rot) として設定される
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。			

パラメータ設定実装例

```

RTC::ReturnCode_t frm_ctrl::onActivated(RTC::UniqueId ec_id)
{
    // default
    // k_gain = 1.0, means time constant is 1 sec.
    // v_max = 200[mm/sec]
    // k_pos_rot = 200[mm], means that sizes of objects are about this size.
    m_com_frm_ctrl.set_param(1.0, 200, 200);

```

(2) calc_vel

関数名	calc_vel			
引数	名称	型	I/O	説明
	geo_ref	FRAME	入力	手先の目標位置と姿勢
	geo_cur	FRAME	入力	手先の現在位置と姿勢
	p_vel	VECTOR	出力	並進速度
	r_vel	VECTOR	出力	回転速度
戻り値	値			説明
	0			正常終了
	その他			異常終了
説明	手先の目標位置・姿勢、手先の現在位置・姿勢を入力として、目標到達判定及び並進速度、回転速度の計算を行う。			
備考 1	本関数は内部処理用として使用している。			
備考 2	FRAME 型は 3×3 行列と 3 次元ベクトルを並べた 3×4 行列で構成されている。			

FRAME 型の定義

```

#define REAL float
typedef REAL VECTOR[3];
typedef REAL MATRIX[3][3];
typedef struct {
    MATRIX mtr;
    VECTOR vtr;
} FRAME;

```

3. 2. tr_jacob（ヤコビ行列変換コンポーネント）

3. 2. 1. 機能概要

本コンポーネントは、座標系の変換に相当するヤコビ行列の変換を行う。また速度とヤコビ行列の成分を一行に並べたシーケンスを作成する。

3. 2. 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	Ubuntu8.04
開発言語	C++
コンパイラ	g++4.2.3-1
RT ミドルウェア／バージョン	OpenRTM-aist-0.4.2-RELEASE（C++版）
依存ライブラリ (OpenRTM)	OmniORB-4.1.1-2
	ACE-5.4.7-13
依存ライブラリ (その他)	行列演算ライブラリ（自作）

動作 OS	Debian4.0r4
開発言語	C++
コンパイラ	g++4.1.1-21
RT ミドルウェア／バージョン	OpenRTM-aist-0.4.2-RELEASE（C++版）
依存ライブラリ (OpenRTM)	OmniORB-4.0.6-2.1
	ACE-5.4.7-12
依存ライブラリ (その他)	行列演算ライブラリ（自作）

3. 2. 3. ポート情報

A) データポート（InPort）

名称	型	データ長	説明
vel	TimedFloatSeq	6	並進速度（単位：mm/sec） 回転速度（単位：rad/sec）
jacob	TimedFloatSeq	row×col (=6×7)	ヤコビ行列 row：行数, col：列数

B) データポート (OutPort)

名称	型	データ長	説明
y_jacob	TimedFloatSeq	48	ヤコビシーケンス

C) サービスポート (Provider)

サービス名	インターフェース名	説明
ComCt	com_ct	ヤコビ行列の変換

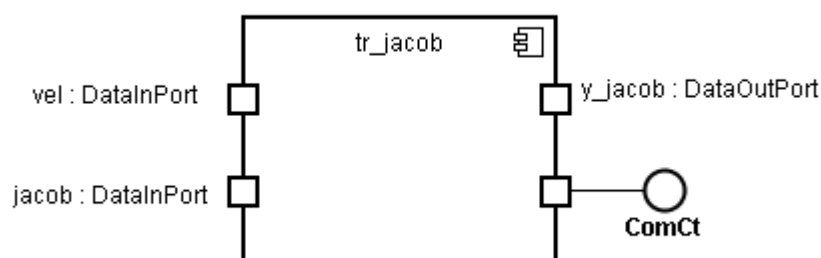


図 3-2 tr_jacob コンポーネント

3. 2. 4. 入出力データフォーマット

3. 2. 4. 1. 入力 : **vel** (3. 1. 4. 2. と同様)

3. 2. 4. 2. 入力 : **jacob**

ヤコビ行列は 6×7 の行列である。

データ位置	格納値
0	ヤコビ行列の 1 行 1 列目
1	ヤコビ行列の 1 行 2 列目
2	ヤコビ行列の 1 行 3 列目
3	ヤコビ行列の 1 行 4 列目
4	ヤコビ行列の 1 行 5 列目
5	ヤコビ行列の 1 行 6 列目
6	ヤコビ行列の 1 行 7 列目
7	ヤコビ行列の 2 行 1 列目
	...
40	ヤコビ行列の 6 行 6 列目
41	ヤコビ行列の 6 行 7 列目

3. 2. 4. 3. 出力：y_jacob

ヤコビシーケンスは指令値とヤコビ行列の成分で構成されたシーケンスである。

データ位置	格納値
0	手先の x 軸方向への並進速度
1	ヤコビ行列の 1 行 1 列目
2	ヤコビ行列の 1 行 2 列目
3	ヤコビ行列の 1 行 3 列目
4	ヤコビ行列の 1 行 4 列目
5	ヤコビ行列の 1 行 5 列目
6	ヤコビ行列の 1 行 6 列目
7	ヤコビ行列の 1 行 7 列目
8	手先の y 軸方向への並進速度
9	ヤコビ行列の 2 行 1 列目
	...
4 0	手先の z 軸回りの回転速度
4 1	ヤコビ行列の 6 行 1 列目
4 2	ヤコビ行列の 6 行 2 列目
4 3	ヤコビ行列の 6 行 3 列目
4 4	ヤコビ行列の 6 行 4 列目
4 5	ヤコビ行列の 6 行 5 列目
4 6	ヤコビ行列の 6 行 6 列目
4 7	ヤコビ行列の 6 行 7 列目

3. 2. 5. サービスポート I/F 仕様

3. 2. 5. 1. ComCt

(1) set_trans

関数名	set_trans			
引数	名称	型	I/O	説明
	trans	Frame&	入力	変換行列
戻り値	値			説明
	なし			-
説明	変換したい座標系への変換行列の設定			
備考				

3. 3. mixer（重み付けコンポーネント）

3. 3. 1. 機能概要

本コンポーネントは各軸の動作に対する重み付けを行う。

3. 3. 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	Ubuntu8.04
開発言語	Python
インタプリタ	Python2.5.2-0
RT ミドルウェア／バージョン	OpenRTM-aist-Python-0.4.1-RELEASE
依存ライブラリ (OpenRTM)	OmniORBpy-3.2.1
	ACE-5.4.7-13

動作 OS	Debian4.0r4
開発言語	Python
インタプリタ	Python2.4.4-2
RT ミドルウェア／バージョン	OpenRTM-aist-Python-0.4.1-RELEASE
依存ライブラリ (OpenRTM)	OmniORBpy-2.6-3.3
	ACE-5.4.7-12

3. 3. 3. ポート情報

A) データポート（InPort）

名称	型	データ長	説明
cartesian	TimedFloatSeq	row×(col+1) (=6×8)	ヤコビシーケンス（逆運動学計算用）
joint	TimedFloatSeq	row×(col+1) (=6×8)	ヤコビシーケンス（センサーフィードバック用）

B) データポート（OutPort）

名称	型	データ長	説明
out	TimedFloatSeq	row×(col+1) (=6×8)	ヤコビシーケンス

C) サービスポート (Provider)

サービス名	インターフェース名	説明
ComMixer_i	com_mixer	各軸の動作重み付けデータの設定と取得

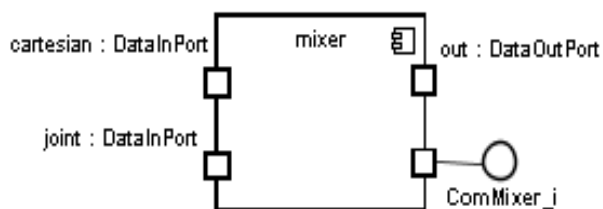


図 3-3 mixer コンポーネント

3. 3. 4. コンフィグレーション情報

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を 7 に固定するものとし、基本的に変更は行わない。

3. 3. 5. 入出力データフォーマット

3. 3. 5. 1. 入力 : **cartesian** (3. 2. 4. 3. と同様)

3. 3. 5. 2. 入力 : **joint**

データ位置	格納値
0	S1 軸の角速度
1	ヤコビ行列の 1 行 1 列目
2	ヤコビ行列の 1 行 2 列目
3	ヤコビ行列の 1 行 3 列目
4	ヤコビ行列の 1 行 4 列目
5	ヤコビ行列の 1 行 5 列目
6	ヤコビ行列の 1 行 6 列目
7	ヤコビ行列の 1 行 7 列目
8	S2 軸の角速度
	...
4 8	W2 軸の角速度
	...
5 4	ヤコビ行列の 7 行 6 列目
5 5	ヤコビ行列の 7 行 7 列目

3. 3. 5. 3. 出力 : **out**

- ・ **cartesian** ポートに書き込みが行われた場合は、3. 2. 4. 3. が出力される
- ・ **joint** ポートに書き込みが行われた場合は、3. 3. 4. 2. が出力される

3. 3. 6. サービスポート I/F 仕様

3. 3. 6. 1. ComMixer_i

(1) set_weight

関数名	set_weight			
引数	名称	型	I/O	説明
	self	-	-	自身クラスのインスタンス
	name	文字列	入力	重みデータの名前
	weight	リスト	出力	重みデータ
戻り値	値			説明
	なし			-
説明	重みデータの格納			

(2) get_weight

関数名	get_weight			
引数	名称	型	I/O	説明
	self	-	-	自身クラスのインスタンス
	name	文字列	入力	重みデータの名前
戻り値	値			説明
	self.weight[name]			正常終了：重みデータ
	[]			異常終了：空リスト
説明	重みデータの取得			

(3) get_keys

関数名	get_keys			
引数	名称	型	I/O	説明
	self	-	-	自身クラスのインスタンス
戻り値	値			説明
	self.comp.out_buf.keys()			データキー（重みデータの名前）の取得
説明	データキーの取得			

3. 4. jinv（逆ヤコビ行列計算コンポーネント）

3. 4. 1. 機能概要

本コンポーネントはロボットアームの手先の並進速度、回転速度とヤコビ行列を用いて、各軸の角速度の算出を行う。尚、冗長自由度アームにも対応するため、ヤコビ行列の逆行列計算には特異値分解（SVD）を用いている。

3. 4. 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	Ubuntu8.04
開発言語	C++
コンパイラ	g++4.2.3-1
RT ミドルウェア／バージョン	OpenRTM-aist-0.4.2-RELEASE（C++版）
依存ライブラリ (OpenRTM)	OmniORB-4.1.1-2
	ACE-5.4.7-13
依存ライブラリ (その他)	行列演算ライブラリ（自作）

動作 OS	Debian4.0r4
開発言語	C++
コンパイラ	g++4.1.1-21
RT ミドルウェア／バージョン	OpenRTM-aist-0.4.2-RELEASE（C++版）
依存ライブラリ (OpenRTM)	OmniORB-4.0.6-2.1
	ACE-5.4.7-12
依存ライブラリ (その他)	行列演算ライブラリ（自作）

3. 4. 3. ポート情報

A) データポート（InPort）

名称	型	データ長	説明
y_jacob	TimedDoubleSeq	row×(col+1) (6×8)	ヤコビシーケンス

B) データポート (OutPort)

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

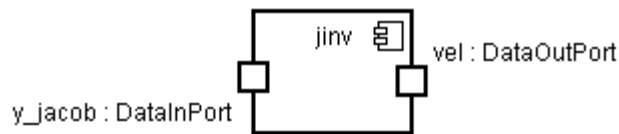


図 3-4 jinv コンポーネント

3. 4. 4. コンフィグレーション情報

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を7に固定するものとし、基本的に変更は行わない。
m_max	int	100	特異値分解行列の最大行数 (≥1)

3. 4. 5. 入出力データフォーマット

3. 4. 5. 1. 入力 : **y_jacob**

mixer のデータポート出力 (3. 3. 5. 3.) に依存するため、3. 2. 4. 3. もしくは、3. 3. 5. 2. が入力される。

3. 4. 5. 2. 出力 : **vel** (関節角速度)

軸名称はアーム本体使用参照 (図 2-7 参照)。

データ位置	格納値
0	S1 軸の角速度
1	S2 軸の角速度
2	S3 軸の角速度
3	E1 軸の角速度
4	E2 軸の角速度
5	W1 軸の角速度
6	W2 軸の角速度

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	Ubuntu8.04
開発言語	C++
コンパイラ	g++4.2.3-1
RT ミドルウェア／バージョン	OpenRTM-aist-0.4.2-RELEASE (C++版)
依存ライブラリ (OpenRTM)	OmniORB-4.1.1-2
	ACE-5.4.7-13

動作 OS	Debian4.0r4
開発言語	C++
コンパイラ	g++4.1.1-21
RT ミドルウェア／バージョン	OpenRTM-aist-0.4.2-RELEASE (C++版)
依存ライブラリ (OpenRTM)	OmniORB-4.0.6-2.1
	ACE-5.4.7-12

3. 5. 3. 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)
th2	TimedDoubleSeq	7	pa10fk へ出力する関節角度 (単位 : rad)

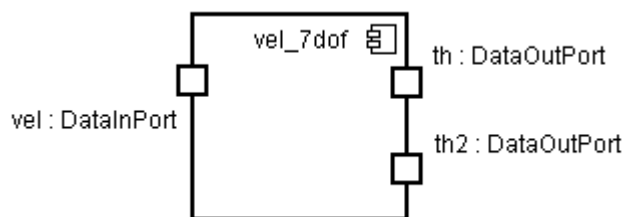


図 3-5 vel_7dof コンポーネント

3. 5. 4. 入出力データフォーマット

3. 5. 4. 1. 入力 : **vel** (関節角速度) (3. 4. 5. 2. と同様)

3. 5. 4. 2. 出力 : **th**

関節名称はアーム本体仕様に記載してある (図 2-7 参照)。

データ位置	格納値
0	S1 軸の角度
1	S2 軸の角度
2	S3 軸の角度
3	E1 軸の角度
4	E2 軸の角度
5	W1 軸の角度
6	W2 軸の角度

3. 5. 4. 3. 出力 : **th2** (3. 5. 4. 2. と同様)

3. 6. pa10fk（PA10 順運動学計算コンポーネント）

3. 6. 1. 機能概要

本コンポーネントは三菱重工業製汎用ロボット PA10 の順運動学計算を行う。
またアーム先端に取り付けるツールの設定も行う。

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	Ubuntu8.04
開発言語	C++
コンパイラ	g++4.2.3-1
RT ミドルウェア／バージョン	OpenRTM-aist-0.4.2-RELEASE（C++版）
依存ライブラリ (OpenRTM)	OmniORB-4.1.1-2
	ACE-5.4.7-13
依存ライブラリ (その他)	行列演算ライブラリ（自作）

動作 OS	Debian4.0r4
開発言語	C++
コンパイラ	g++4.1.1-21
RT ミドルウェア／バージョン	OpenRTM-aist-0.4.2-RELEASE（C++版）
依存ライブラリ (OpenRTM)	OmniORB-4.0.6-2.1
	ACE-5.4.7-12
依存ライブラリ (その他)	行列演算ライブラリ（自作）

3. 6. 3. 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	現在の各軸角度情報 (単位 : rad)

B) データポート (OutPort)

名称	型	データ長	説明
frame	TimedDoubleSeq	12	手先位置 (単位 : mm) ・ 姿勢
jacob	TimedDoubleSeq	42	ヤコビ行列

C) サービスポート (Provider)

サービス名	インターフェース名	説明
ComFk	com_fk	PA10 の順運動学計算

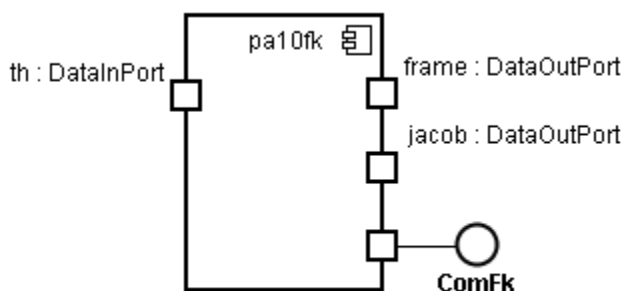


図 3-6 pa10fk コンポーネント

3. 6. 4. 入出力データフォーマット

3. 6. 4. 1. 入力 : **th** (3. 5. 4. 2. と同様)

3. 6. 4. 2. 出力 : **frame** (3. 1. 4. 1. と同様)

3. 6. 4. 3. 出力 : **jacob** (3. 2. 4. 2. と同様)

3. 6. 5. サービスポート I/F 仕様

3. 6. 5. 1. ComFk

(1) set_tool

関数名	set_tool			
引数	名称	型	I/O	説明
	frm	const Frame&	入力	手先の位置とツールの姿勢
戻り値	値			説明
	なし			-
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 (myRTC/pa10fk/com_fk.idl 内)

```
struct Frame{
    float mat[3][3];
    float pos[3];
}
```

手先の位置と姿勢の設定実装例

```
RTC::ReturnCode_t pa10fk::onActivated(RTC::UniqueId ec_id)
{
    FRAME tmp;
    tmp.vtr[0] = 0.0;
    tmp.vtr[1] = 0.0;
    tmp.vtr[2] = 150.0
    MATRIX m_b, m_c;
    mb(PI,m_b);
    mc(PI/4.0,m_c);
    m_mul(m_b,m_c,tmp.mtr);

    m_com_fk.fk_kinema.set_tool(&tmp);
}
```


3. 7. coord_trans（座標系変換コンポーネント）

3. 7. 1. 機能概要

本コンポーネントは、座標系の変換を行う。

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	Ubuntu8.04
開発言語	C++
コンパイラ	g++4.2.3-1
RT ミドルウェア／バージョン	OpenRTM-aist-0.4.2-RELEASE（C++版）
依存ライブラリ (OpenRTM)	OmniORB-4.1.1-2
	ACE-5.4.7-13
依存ライブラリ (その他)	行列演算ライブラリ（自作）

動作 OS	Debian4.0r4
開発言語	C++
コンパイラ	g++4.1.1-21
RT ミドルウェア／バージョン	OpenRTM-aist-0.4.2-RELEASE（C++版）
依存ライブラリ (OpenRTM)	OmniORB-4.0.6-2.1
	ACE-5.4.7-12
依存ライブラリ (その他)	行列演算ライブラリ（自作）

3. 7. 3. ポート情報

A) データポート（InPort）

名称	型	データ長	説明
in	TimedDoubleSeq	12	現在手先位置（単位：mm）・姿勢

B) データポート（OutPort）

名称	型	データ長	説明
out	TimedDoubleSeq	12	現在手先位置（単位：mm）・姿勢

C) サービスポート (Provider)

サービス名	インターフェース名	説明
ComCt	com_ct	座標系の変換

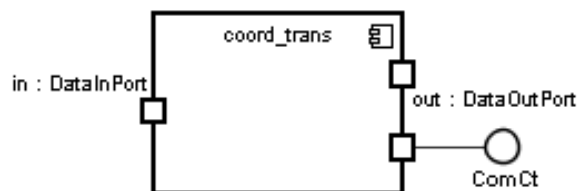


図 3-7 coord_trans コンポーネント

3. 7. 4. 入出力データフォーマット

3. 7. 4. 1. 入力： **in** (現在位置・姿勢) (3. 1. 4. 1. と同様)

3. 7. 4. 2. 出力： **out** (現在位置・姿勢) (3. 1. 4. 1. と同様)

3. 7. 5. サービスポート I/F 仕様

3. 7. 5. 1. **ComCt** (3. 2. 5. 1. と同様)

3. 8. move（操作制御コンポーネント）

3. 8. 1. 機能概要

本コンポーネントは目標手先位置・姿勢の入力を行う。

3. 8. 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	Ubuntu8.04
開発言語	Python
インタプリタ	Python2.5.2-0
RT ミドルウェア／バージョン	OpenRTM-aist-Python-0.4.1-RELEASE
依存ライブラリ (OpenRTM)	OmniORBpy-3.2.1
	ACE-5.4.7-13
依存ライブラリ (その他)	行列演算ライブラリ（自作）

動作 OS	Debian4.0r4
開発言語	Python
インタプリタ	Python2.4.4-2
RT ミドルウェア／バージョン	OpenRTM-aist-Python-0.4.1-RELEASE
依存ライブラリ (OpenRTM)	OmniORBpy-2.6-3.3
	ACE-5.4.7-12
依存ライブラリ (その他)	行列演算ライブラリ（自作）

3. 8. 3. ポート情報（Python 版）

A) データポート（InPort）

名称	型	データ長	説明
state	TimedLong	1	到達情報（1：到着 0：未到着）

B) データポート（OutPort）

名称	型	データ長	説明
frame	TimedDoubleSeq	12	目標手先位置（単位：mm）・姿勢

C) サービスポート (Provider)

サービス名	インターフェース名	説明
ComMv	com_move	算出した目標手先位置・姿勢をデータポート (OutPort)へ書き込む

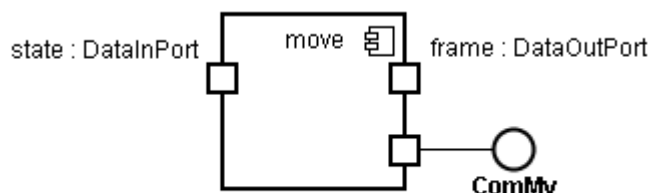


図 3-8 move コンポーネント

3. 8. 4. 入出力データフォーマット

3. 8. 4. 1. 入力 : **state** (3. 1. 4. 3と同様)3. 8. 4. 2. 出力 : **frame** (3. 1. 4. 1と同様)

3. 8. 5. サービスポート I/F 仕様

3. 8. 5. 1. **ComMv**(1) **com_move**

関数名	com_move			
引数	名称	型	I/O	説明
	self	-	-	自身クラスのインスタンス
	frm	FRAME	入力	目標手先位置・姿勢
戻り値	値			説明
	なし			-
説明	引数として渡された frm をデータポート(OutPort)に書き込む。			

3. 9. pa10disp (PA10 幾何モデル描画コンポーネント)

3. 9. 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、幾何モデルの表示を行う。

3. 9. 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	Ubuntu8.04
開発言語	Python
インタプリタ	Python2.5.2-0
RT ミドルウェア／バージョン	OpenRTM-aist-Python-0.4.1-RELEASE
依存ライブラリ (OpenRTM)	OmniORB-4.1.1-2
	OmniORBpy-3.2.1
	ACE-5.4.7-13
依存ライブラリ (その他)	VPython3.2.9-3

動作 OS	Debian4.0r4
開発言語	Python
インタプリタ	Python2.4.4-2
RT ミドルウェア／バージョン	OpenRTM-aist-Python-0.4.1-RELEASE
依存ライブラリ (OpenRTM)	OmniORBpy-2.6-3.3
	ACE-5.4.7-12
依存ライブラリ (その他)	VPython3.2.14+b

3. 9. 3. 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)

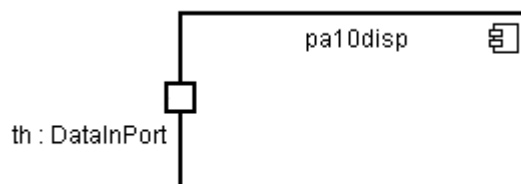


図 3-9 pa10disp コンポーネント

3. 9. 4. 入出力データフォーマット

3. 9. 4. 1. 入力 : **th** (3. 5. 4. 2. と同様)

3. 10. slider(データ送出コンポーネント)

3. 10. 1. 機能概要

本モジュールでは、各軸へ角速度または角度を送出し、逆運動学を考えず任意にアームを制御したりセンサーフィードバックを行いたい場合にこのコンポーネントを利用する。

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	Ubuntu8.04
開発言語	Python
インタプリタ	Python2.5.2-0
RT ミドルウェア／バージョン	OpenRTM-aist-Python-0.4.1-RELEASE
依存ライブラリ (OpenRTM)	OmniORBpy-3.2.1
	ACE-5.4.7-13

動作 OS	Debian4.0r4
開発言語	Python
インタプリタ	Python2.4.4-2
RT ミドルウェア／バージョン	OpenRTM-aist-Python-0.4.1-RELEASE
依存ライブラリ (OpenRTM)	OmniORBpy-2.6-3.3
	ACE-5.4.7-12

3. 10. 3. ポート情報

A) データポート (OutPort)

名称	型	データ長	説明
out	TimedDoubleSeq	7	各軸の角速度

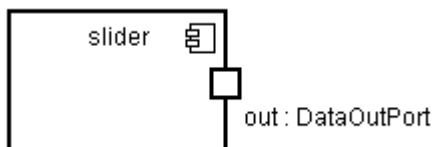


図 3-10 slider コンポーネント

3. 10. 4. コンフィグレーション情報

名称	型	デフォルト値		説明
channels	リスト	('data', -10, 10, 0.1, 100)		送出データパネルの設定
		data	データラベル	
		-10	最小送出値	
		10	最大送出値	
		0.1	送出値解像度	
		100	操作ウィンドウサイズ	

3. 10. 5. 入出力データフォーマット

3. 10. 5. 1. 出力: **out** (関節角速度) (3. 4. 5. 2. と同様)

3. 1 1. j_jacob(ヤコビシーケンス生成コンポーネント)

3. 1 1. 1. 機能概要

本コンポーネントは、各軸の角速度とヤコビ行列を一行に並べたヤコビシーケンスを作成する。

3. 1 1. 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	Ubuntu8.04
開発言語	C++
コンパイラ	g++4.2.3-1
RT ミドルウェア／バージョン	OpenRTM-aist-0.4.2-RELEASE (C++版)
依存ライブラリ (OpenRTM)	OmniORB-4.1.1-2
	ACE-5.4.7-13
依存ライブラリ(その他)	行列演算ライブラリ (自作)

動作 OS	Debian4.0r4
開発言語	C++
コンパイラ	g++4.1.1-21
RT ミドルウェア／バージョン	OpenRTM-aist-0.4.2-RELEASE (C++版)
依存ライブラリ (OpenRTM)	OmniORB-4.0.6-2.1
	ACE-5.4.7-12
依存ライブラリ(その他)	行列演算ライブラリ (自作)

3. 1 1. 3. 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	各軸の角速度

B) データポート (OutPort)

名称	型	データ長	説明
y_jacob	TimedDoubleSeq	56	ヤコビシーケンス

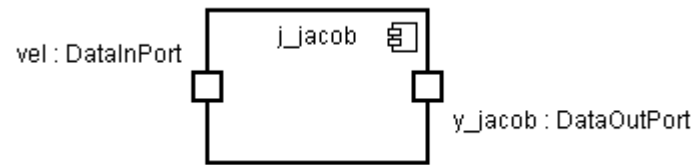


図 3-11 j_jacob コンポーネント

3. 1 1. 4. 入出力データフォーマット

3. 1 1. 4. 1. 入力 : **vel** (関節角速度) (3. 4. 5. 2. と同様)

3. 1 1. 4. 2. 出力 : **y_jacob** (3. 3. 4. 2. と同様)

3. 1 2. pa10vel (PA10 実機制御コンポーネント)

3. 1 2. 1. 機能概要

本コンポーネントは三菱重工業製汎用ロボット PA10 の関節角速度制御を行っている。

3. 1 2. 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	Debian4.0r4
開発言語	C++
コンパイラ	g++4.1.1-21
RT ミドルウェア／バージョン	OpenRTM-aist-0.4.2-RELEASE (C++版)
依存ライブラリ (OpenRTM)	OmniORB-4.0.6-2.1
	ACE-5.4.7-12
依存ライブラリ (その他)	行列演算ライブラリ (自作)
	ライフロボティクス社製 PA-10 高速制御用ソフトウェア (商用)

3. 1 2. 3. ポート情報

A) データポート (InPort)

名称	型	値の範囲	説明
vel	TimedDoubleSeq	[※1]	各軸 (7 軸) の動作角速度 (単位 : rad/sec)

[※1] PA10 の制御パラメータ情報より速度リミットを取得し、判断を行っている。リミット値のデフォルトの設定はアーム本体仕様を参照のこと。

B) データポート (OutPort)

名称	型	値の範囲	説明
th	TimedDoubleSeq	[※2]	現在の各軸（7 軸）角度（単位：rad）

[※2] PA10 の制御パラメータ情報より各軸の角度情報を取得し、出力パラメータとしている。リミット値のデフォルトの設定はアーム本体仕様を参照のこと。

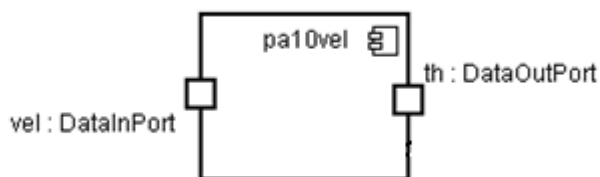


図 3-12 pa10vel コンポーネント

3. 1 2. 4. 入出力データフォーマット

3. 1 2. 4. 1. 入力：**vel**（関節角速度）（3. 3. 4. 2 と同様）
関節名称はアーム本体仕様に記載してある（図 2-7 参照）。

データ位置	格納値
0	S1 軸の角速度
1	S2 軸の角速度
2	S3 軸の角速度
3	E1 軸の角速度
4	E2 軸の角速度
5	W1 軸の角速度
6	W2 軸の角速度

3. 1 2. 4. 1. 出力：**th**（3. 4. 4. 1 と同様）

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。

- 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の「次世代ロボット知能化技術開発プロジェクト」内実施者向けに評価を目的として提供するものであり、商用利用など他の目的で使用することを禁じます。
- 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。
- PA-10 高速制御用ソフトウェアは、ライフロボティクス株式会社の製品であり、権利はライフロボティクス株式会社に帰属します。

【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