

# 汎用モーション RTC 操作マニュアル

本書は、著作権法により保護されています。

本書の内容を全部あるいは一部に関わらず、弊社の許諾を得ずに、いかなる方法においても無断で第3者へ開示、複製することを禁じています。  
本書の内容は予告なく変更されることがあります。



資料番号：RB1400295

改版：第8版

版	年 月	変 更 内 容
初版	2011/09/22	
2 版	2011/10/25	受入確認チェックシート 382 に関する対応。
3 版	2011/10/27	追加指摘事項対応。
4 版	2011/12/06	冗長性利用モジュールとの接続図（図 4）を修正。
5 版	2011/12/16	動作環境セットアップ手順を修正。冗長性利用モジュールとのリンク、OpenHRP3 のバージョンなど。
6 版	2011/12/19	使用するモジュール名を“冗長性利用モジュール”から“汎用モーション RTC コア”に修正。
7 版	2012/01/05	3.1.1 Java ダウンロード URL の変更。3.1.7 手順 の修正（外部 JAR の追加）。 4.2.2 start.sh のファイルパス修正。
8 版	2012/02/14	2. RTC モジュールの図を修正。RTSystemEditor 上の表示とあわせる。 4.4 モーション動作後の 3DView を追記。

## はじめに

本マニュアルは、「次世代ロボット知能化技術開発プロジェクト」の「施設内生活支援ロボット知能研究開発」において開発した汎用モーション RT コンポーネント(以下、汎用モーション RTC)を、OpenHRP3 環境を利用して操作する手順を説明するものです。

本マニュアルは RT ミドルウェア、RTC、OpenHRP3(GrxUI)に関する基本知識を備えた利用者を対象として記述されており、RT ミドルウェア、RTC や関連ツールに関する一般的な知識を持つことを前提とします。

RT ミドルウェア、RTC については下記を参照下さい。

OpenRTM-aist Official Website

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

### 関連文書

- 汎用モーション RTC インターフェース仕様書(資料番号 : RB1400065)

### 登録商標など

- Java は、サン・マイクロシステムズの登録商標です。
- Ethernet は、米国 Xerox Corporation の登録商標です。
- 上記の他、本マニュアルに記載した製品名・会社名などの固有名詞は各社の商標・登録商標・商品名です。本文中の各社の商標・登録商標には TM・®マークは表示しておりません。

### ソフトウェアについて

- 本ソフトウェアの逆コンパイル・逆アセンブルなどを行うことは固くお断りいたします。
- 本ソフトウェアの一部または全部を当社の事前の承認なしに譲渡・交換・転貸などによって第三者に使用させることは固くお断りいたします。

# 目次

1. システム概要.....	6
1.1. システムの構成 .....	6
1.2. 動作環境 .....	7
2. RTC モジュール.....	8
2.1. 汎用モーション RTC 群 .....	8
2.1.1. 汎用モーション RTC.....	8
2.1.2. 結合 RTC.....	12
2.1.3. 分配 RTC.....	13
2.1.4. 汎用モーション RTC コア .....	13
2.2. シミュレーション用 RTC 群 .....	13
2.2.1. アーム RTC .....	13
2.2.2. 腰 RTC .....	14
2.2.3. SmartPalController .....	16
2.2.4. SmartPal .....	16
2.3. クライアント RTC .....	17
2.3.1. 汎用モーションクライアント RTC .....	17
3. 動作環境セットアップ .....	18
3.1. ソフトウェアインストール .....	18
3.1.1. Java のインストール .....	18
3.1.2. OpenHRP3 のインストール .....	18
3.1.3. Eclipse のインストール .....	19
3.1.4. OpenRTM-aist のインストール.....	19
3.1.5. 汎用モーション RTC コアのインストール.....	19
3.1.6. 汎用モーション RTC モジュール群のインストール .....	19
3.1.7. 汎用モーション RTC モジュール群のビルド .....	20
3.2. 環境設定 .....	20
3.2.1. プロジェクトファイル、モデルファイルの設定 .....	20
4. シミュレーションの実行手順.....	21
4.1. OpenHRP3 の起動.....	21
4.1.1. OpenHRP3 の起動.....	21
4.1.2. プロジェクトファイルの読み込み .....	21
4.2. コンポーネントの起動.....	22
4.2.1. 汎用モーション RTC コアの起動.....	22
4.2.2. 汎用モーション RTC、その他 RTC の起動 .....	22
4.2.3. コントローラの起動.....	22
4.3. コンポーネントの接続.....	23
4.4. 汎用モーションクライアント RTC の操作.....	24
4.4.1. 汎用モーションクライアント RTC の説明.....	24

4.5.	シミュレーションの実行手順.....	25
------	--------------------	----

# 1. システム概要

## 1.1. システムの構成

本システムの構成を図 1 に示します。本システムは、汎用モーション RTC 群、シミュレーション用 RTC 群、汎用モーションクライアント RTC から構成されます。汎用モーション RTC 群は、RT ミドルウェアに対応したソフトウェアモジュールであり、他の RT コンポーネントと接続して使用することができます。

汎用モーション RTC は、SmartPal のような 7 軸アーム、腰軸、全方向移動台車といった冗長自由度を備えたロボットにおいて、アーム先端の制御点の目標位置指令を入力として、関節座標系における補間動作（PTP 動作）や直交座標における直線補間（CP 動作）を計算し、キネマティクス変換を行った上で、各関節への低レベル位置指令を生成するモジュールです。

東北大学殿の汎用モーション RTC コアは、キネマティクス変換処理を行い各軸の目標関節角度を算出します。

結合 RTC は、各関節角度を結合して 1 つのデータに結合しています。

分配 RTC は、1 つのデータを 2 つのデータに分配しています。

シミュレーション用 RTC 群は、アーム RTC、腰 RTC、移動台車 RTC（将来機能）、SmartPalController、OpenHRP3(GrxUI)から構成されます。シミュレーション用 RTC 群を実機用の RTC に置換えることで、実際にロボットを動作させることができます。

汎用モーションクライアント RTC は、汎用モーション RTC に動作指令を行う、画面を持ったクライアントです。汎用モーション RTC とのインターフェースについては「2.1.1.2：サービスポートインターフェース」を参照ください。

本ソフトウェアの PC へのインストール手順を 3 章で説明します。

本システムを使用して、シミュレーション上のロボットを操作する手順を 4 章で説明します。

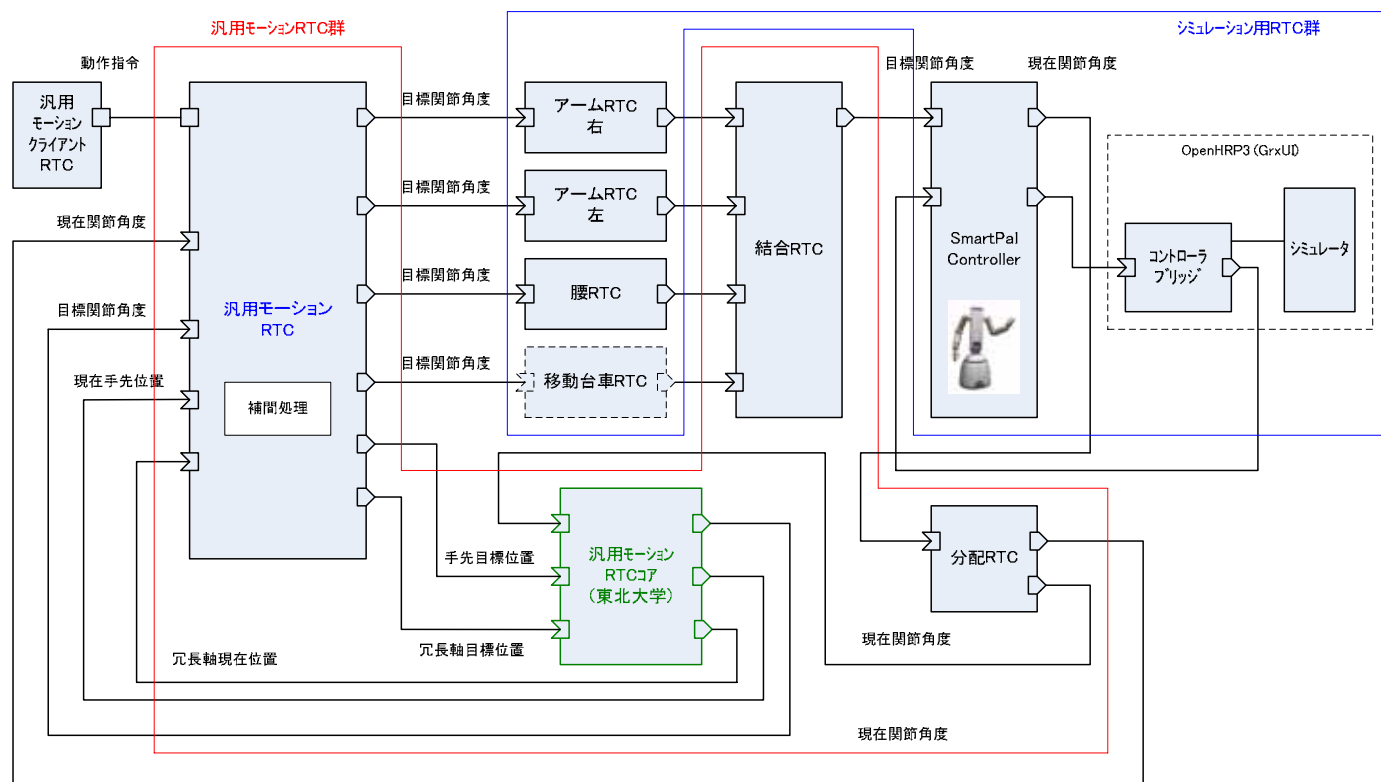


図 1 システム構成

## 1.2.動作環境

汎用モーション RTC を動作させるにあたって、下記のものを準備していただく必要があります。

- パーソナルコンピュータ (以下、「PC」と記述)
- 汎用モーション RTC モジュール群
- シミュレーション用 RTC 群
- 汎用モーションクライアント RTC
- Java
- OpenHRP3
- Eclipse
- OpenRTM-aist(Java 版)
- OmniORB

### (1) PC

動作 OS	Linux(Ubuntu Linux 10.04 にて動作を確認)
RT ミドルウェア	OpenRTM-aist-1.0.0-RELEASE(Java)

### (2) 汎用モーション RTC モジュール群

### (3) シミュレーション用 RTC 群

### (4) 汎用モーションクライアント RTC

シミュレーション上で SmartPal を動作させるために必要です。

### (5) Java

Java は、汎用モーション RTC および RT ミドルウェア (以下 RTM と記述) を動作させるために必要です。

バージョン 1.6 を推奨します。

### (6) OpenHRP3

OpenHRP3 は、シミュレーション(GrxUI)を動作させるために必要です。

バージョン 3.0.8 を推奨します。

### (7) Eclipse

Eclipse は、RTSystemEditor を動作させるために必要です。

バージョン 3.4 を推奨します。

### (8) OpenRTM-aist

OpenRTM-aist は、各 RTC を動作させるために必要です。

バージョン 1.0.0 を推奨します。

# 2.RTC モジュール

## 2.1.汎用モーション RTC 群

### 2.1.1.汎用モーション RTC

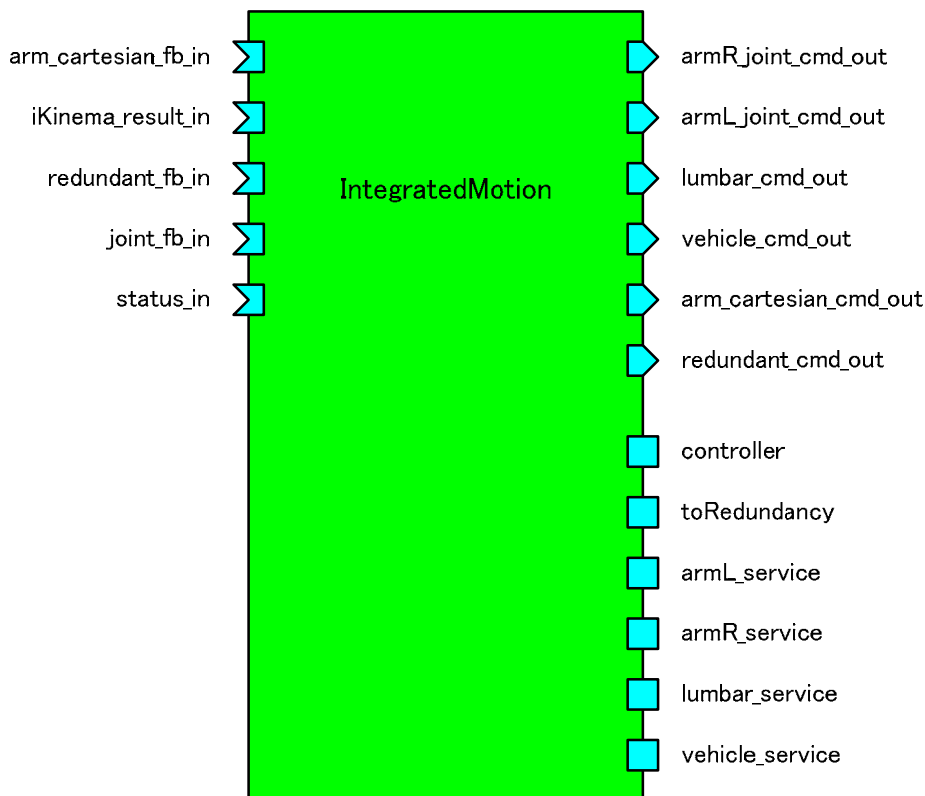


図 2 汎用モーション RTC

#### 2.1.1.1.ポート

表 1 にデータポートの詳細を示します。

表 2 にサービスポートの詳細を示します。

表 1 汎用モーション RTC データポート

ポート名	入出力	データ型	説明
arm_cartesian_fb_in	入力	TimedDoubleSeq	右手先・左手先の現在位置 (汎用モーション RTC コアのデータポート xyzR_out に接続)
			[0] 右アーム x [mm]
			[1] 右アーム y [mm]
			[2] 右アーム z [mm]
			[3] ~ [11] 右アーム回転行列(3×3)
			[12] 左アーム x [mm]
			[13] 左アーム y [mm]
			[14] 左アーム z [mm]



			[15] ~ [23]	左アーム回転行列( 3 × 3 )
iKinema_result_in	入力	TimedDoubleSeq	逆運動学演算結果 ( 汎用モーション RTC コアのデータポート qref_out に接続 )	
			[0]	移動 x [mm]
			[1]	移動 y [mm]
			[2]	移動 [degree]
			[3]	腰軸 rx ( 現状ダミー ) [degree]
			[4]	腰軸 ry [degree]
			[5]	腰軸 rz ( 現状ダミー ) [degree]
			[6] ~ [12]	右アーム J1 ~ J7 [degree]
			[13] ~ [19]	左アーム J1 ~ J7 [degree]
			[20] ~ [22]	頭 [degree]
redundant_fb_in	入力	TimedDoubleSeq	冗長軸の現在位置 ( 汎用モーション RTC コアのデータポート redun_cur_out に接続 )	
			[0]	移動 x [mm]
			[1]	移動 y [mm]
			[2]	移動 [deg]
			[3]	腰軸 rx [deg]
			[4]	腰軸 ry [deg]
			[5]	腰軸 rz [deg]
			[6]	右アーム肘角 [deg]
			[7]	左アーム肘角 [deg]
joint_fb_in	入力	TimedDoubleSeq	現在関節角度・位置 格納順：移動 x 移動 y 移動 腰軸 右アーム 左アーム。 ( 分配 RTC のデータポート data_out1 に接続 )	
			[0]	移動 x [mm]
			[1]	移動 y [mm]
			[2]	移動 [degree]
			[3]	腰軸 rx ( 現状ダミー ) [degree]
			[4]	腰軸 ry [degree]

			[5]	腰軸 rz ( 現状ダミー ) [degree]
			[6] ~ [12]	右アーム J1 ~ J7 [degree]
			[13]	グリップ
			[14] ~ [20]	左アーム J1 ~ J7 [degree]
			[21]	グリップ
status_in	入力	TimedDoubleSeq	汎用モーション RTC コアのステータス情報 ( 汎用モーション RTC コアのデータポート status_out に接続 )	
			[0]	ステータス
			[1]	コンディショニングナンバー
armR_joint_cmd_out	出力	TimedJointPos	右アーム・グリップ各軸の目標関節角度	
armL_joint_cmd_out	出力	TimedJointPos	左アーム・グリップ各軸の目標関節角度	
lumbar_cmd_out	出力	TimedJointPos	腰の各軸の目標関節角度	
vehicle_cmd_out	出力	TimedDoubleSeq	移動台車の目標位置・姿勢	
arm_cartesian_cmd_out	出力	TimedDoubleSeq	アーム手先目標位置・回転行列 格納順：右アーム 左アーム ( 汎用モーション RTC コアのデータポート xyzR_in に接続 )	
			[0]	右アーム x [mm]
			[1]	右アーム y [mm]
			[2]	右アーム z [mm]
			[3] ~ [11]	右アーム回転行列( 3×3 )
			[12]	左アーム x [mm]
			[13]	左アーム y [mm]
			[14]	左アーム z [mm]
			[15] ~ [23]	左アーム回転行列( 3×3 )
redundant_cmd_out	出力	TimedDoubleSeq	冗長軸の目標位置 ( 汎用モーション RTC コアのデータポート redun_ref_in に接続 )	
			[0]	移動 x [mm]
			[1]	移動 y [mm]
			[2]	移動 [degree]
			[3]	腰軸 rx ( 現状ダミー ) [degree]
			[4]	腰軸 ry [degree]
			[5]	腰軸 rz ( 現状ダミー ) [degree]

			[6]	右アーム肘角 [degree]
			[7]	左アーム肘角 [degree]

表 2 汎用モーション RTC サービスポート

ポート名	極性	型	説明
controller	提供	IntegratedMotionInterface	上位からの指令受付
armR_service	要求	ManipulatorCommonInterface_Common	アーム RTC への要求 (右アーム)
armL_service	要求	ManipulatorCommonInterface_Common	アーム RTC への動作要求 (左アーム)
lumbar_service	要求	LumbarInterface_Common	腰 RTC への要求
vehicle_service	要求		移動台車 RTC への要求
toRedundancy	要求	RedundancyConfig	汎用モーション RTC コアへの 要求

### 2.1.1.2. サービスポートインターフェース

サービスポート controller のオペレーション一覧を表 3 に示します。

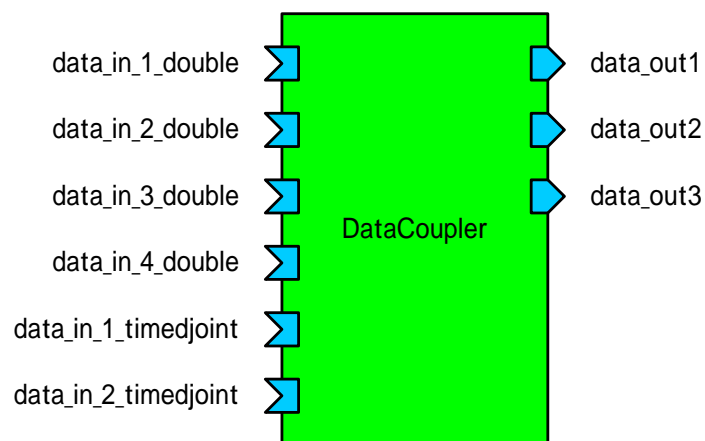
各オペレーションの詳細については、関連文書：汎用モーション RTC インターフェース仕様書を参照ください。

表 3 controller インターフェース

オペレーション	説明
abort	現在の動作を中止し、即時停止する
clearAlarms	全てのアラームを解除する
closeGripper	グリッパを完全に閉じる (将来機能)
getActiveAlarm	発生中のアラームを取得する
getFeedbackCartesian	制御点の直交座標のフィードバック値を取得する
getFeedbackPosJoint	指定ユニットの各軸の現在位置を取得する
getGripperPos	グリッパの現在位置を取得する (将来機能)
getState	モジュールの状態を返す
getVersion	バージョン情報を取得する
isMoving	全軸の動作状態を返す
isServoOn	サーボ制御の入切状態を返す
moveCPHoldAbs	制御点を維持したまま冗長軸を絶対指令で動作する
moveCPHoldRel	制御点を維持したまま冗長軸を相対指令で動作する
moveGripper	グリッパを指定された角度へ移動する (将来機能)
moveLinearCartesianAbs	制御点を原点フレームに基づく絶対位置指令で直線補間動作する
moveLinearCartesianRel	制御点を原点フレームに基づく相対位置指令で直線補間動作する

movePTPCartesianAbs	関節空間において、目標位置を絶対直交座標指定により直線補間を動作する
movePTPCartesianRel	関節空間において、目標位置を相対直交座標指定により直線補間を動作する
moveUnitAbs	ユニットの各軸を指定位置（絶対位置）に移動する
moveUnitRel	ユニットの各軸を指定位置（相対位置）に移動する
openGripper	グリッパを完全に開く（将来機能）
pause	全軸の動作を一時停止する
resetOriginalFrame	原点フレームをロボットの現在姿勢でリセットする（将来機能）
resume	一時停止した動作を再開する
selectRedundantAxes	動作に使用可能な冗長軸を設定する
servoOff	全モーションモジュール全軸のサーボ制御を切る
servoOn	全モーションモジュール全軸のサーボ制御を入れる
setControlPointOffset	制御点のフランジ面からのオフセット量を設定する
stop	現在の動作を中止し、減速停止する

## 2.1.2. 結合 RTC



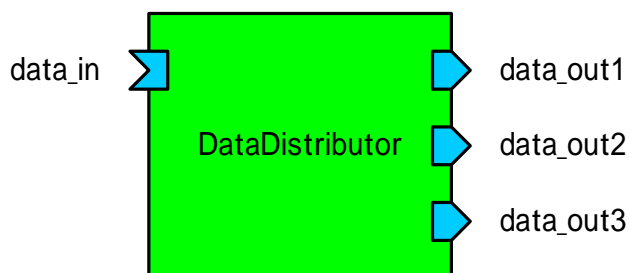
### 2.1.2.1. ポート

表 4 にデータポートの詳細を示します。

表 4 データポート

ポート名	入出力	データ型	説明
data_in_1_double	入力	TimedDoubleSeq	補間周期毎の右アームの目標関節角度
data_in_2_double	入力	TimedDoubleSeq	補間周期毎の左アームの目標関節角度
data_in_3_double	入力	TimedDoubleSeq	補間周期毎の腰の目標関節角度
data_in_4_double	入力	TimedDoubleSeq	未使用
data_in_1_timedjoint	入力	TimedJointPos	未使用
data_in_2_timedjoint	入力	TimedJointPos	未使用
data_out1	出力	TimedDoubleSeq	補間周期毎の全軸の目標関節角度
data_out2	出力	TimedDoubleSeq	補間周期毎の全軸の目標関節角度
data_out3	出力	TimedDoubleSeq	補間周期毎の全軸の目標関節角度

### 2.1.3.分配 RTC



#### 2.1.3.1.ポート

表 5 にデータポートの詳細を示します。

表 5 データポート

ポート名	入出力	データ型	説明
data_in	入力	TimedDoubleSeq	全軸の現在関節角度
data_out1	出力	TimedDoubleSeq	全軸の現在関節角度 (汎用モーション RTC のデータポート joint_fb_in に接続)
data_out2	出力	TimedDoubleSeq	全軸の現在関節角度 (汎用モーション RTC コアのデータポート q_cur_in に接続)
data_out3	出力	TimedDoubleSeq	未使用

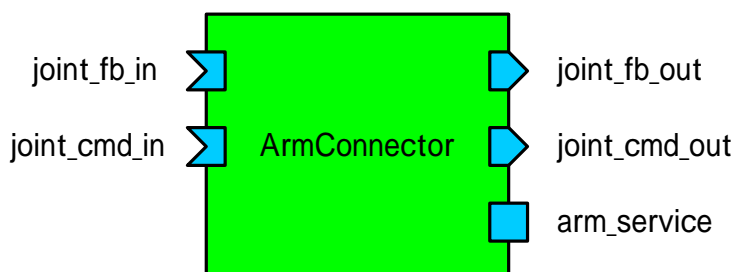
#### 2.1.4.汎用モーション RTC コア

汎用モーション RTC コアの詳細については、東北大学殿が提供するドキュメントを参照ください。

## 2.2.シミュレーション用 RTC 群

本システムをシミュレーションで動作させる場合に、必要なコンポーネントを以下に示します。

#### 2.2.1.アーム RTC



#### 2.2.1.1.ポート

表 6 にデータポートの詳細を示します。

表 7 にサービスポートの詳細を示します。

表 6 データポート

ポート名	入出力	データ型	説明
joint_fb_in	入力	TimedDoubleSeq	アーム・グリッパの現在関節角度
joint_cmd_in	入力	TimedJointPos	補間周期毎のアーム・グリッパの目標関節角度
joint_fb_out	出力	TimedJointPos	アーム・グリッパの現在関節角度
joint_cmd_out	出力	TimedDoubleSeq	アーム・グリッパの目標関節角度

表 7 サービスポート

ポート名	極性	型	説明
arm_service	提供	ManipulatorCommonInterface_Common	上位からの指令受付

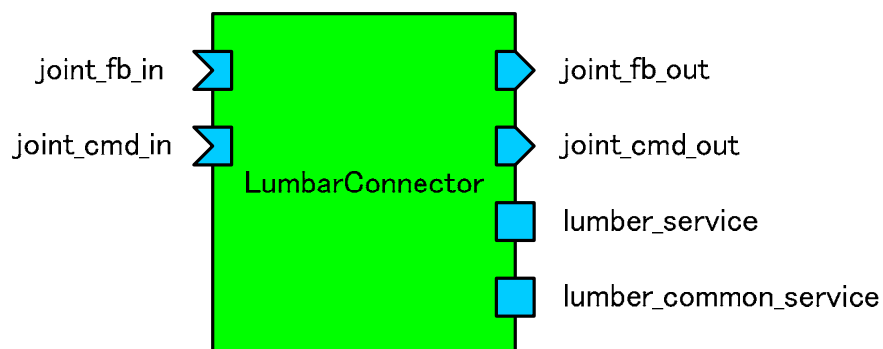
### 2.2.1.2. サービスポートインターフェース

サービスポート armService のオペレーション一覧を表 8 に示します。

表 8 armService インターフェース

オペレーション名	説明
clearAlarms	アラームクリア
getActiveAlarm	アラーム情報の取得
getFeedbackPosJoint	関節座標系の位置フィードバック情報の取得
getManipInfo	マニピュレータ情報の取得
getSoftLimitJoint	関節座標系のソフトリミット値を取得
getState	ユニットの状態取得
servoOFF	全軸サーボ OFF
servoON	全軸サーボ ON
setSoftLimitJoint	関節座標系のソフトリミット値を設定

### 2.2.2. 腰 RTC



#### 2.2.2.1. ポート

表 9 にデータポートの詳細を示します。

表 10 にサービスポートの詳細を示します。

表 9 データポート

ポート名	入出力	データ型	説明
joint_fb_in	入力	TimedDoubleSeq	腰の現在関節角度
joint_cmd_in	入力	TimedJointPos	補間周期毎の腰の目標関節角度
joint_fb_out	出力	TimedJointPos	腰の現在関節角度
joint_cmd_out	出力	TimedDoubleSeq	腰の目標関節角度

表 10 サービスポート

ポート名	極性	型	説明
lumbar_service	提供	LumbarUnit	上位からの指令受付
lumbar_common_service	提供	ExtAxes	未使用

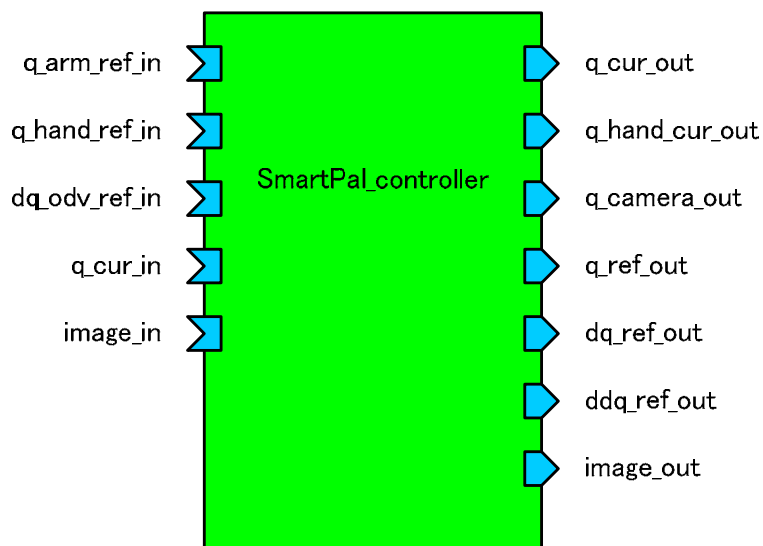
#### 2.2.2.2. サービスポートインターフェース

サービスポート lumbarService のオペレーション一覧を表 11 に示します。

表 11 lumbarService インターフェース

オペレーション名	説明
clearAlarms	アラームクリア
getActiveAlarm	アラーム情報の取得
getFeedbackPosJoint	関節座標系の位置フィードバック情報の取得
getSoftLimitJoint	関節座標系のソフトリミット値を取得
getState	ユニットの状態取得
servoOFF	全軸サーボ OFF
servoON	全軸サーボ ON
setSoftLimitJoint	関節座標系のソフトリミット値を設定

### 2.2.3.SmartPalController



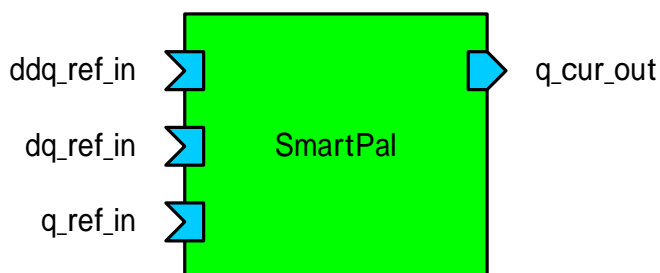
#### 2.2.3.1.ポート

表 12 にデータポートの詳細を示します。

表 12 データポート

ポート名	入出力	データ型	説明
q_arm_ref_in	入力	TimedDoubleSeq	補間周期毎の右アームの目標関節角度
q_hand_ref_in	入力	TimedDoubleSeq	未使用
dq_odv_ref_in	入力	TimedDoubleSeq	未使用
q_cur_in	入力	TimedDoubleSeq	全軸の現在関節角度
image_in	入力	TimedLongSeq	未使用
q_cur_out	出力	TimedDoubleSeq	全軸の現在関節角度
q_hand_cur_out	出力	TimedDoubleSeq	未使用
q_camera_out	出力	TimedDoubleSeq	未使用
q_ref_out	出力	TimedDoubleSeq	補間周期毎の全軸の目標関節角度
dq_ref_out	出力	TimedDoubleSeq	補間周期毎の全軸の速度
ddq_ref_out	出力	TimedDoubleSeq	補間周期毎の全軸の加速度
image_out	出力	TimedDoubleSeq	未使用

### 2.2.4.SmartPal





#### 2.2.4.1.ポート

表 13 データポート表 13 にデータポートの詳細を示します。

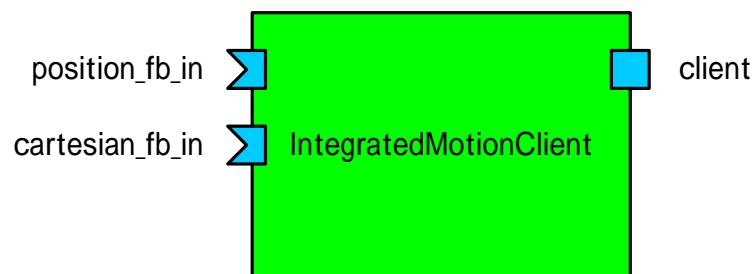
表 13 データポート

ポート名	入出力	データ型	説明
ddq_ref_in	入力	TimedDoubleSeq	未使用
dq_ref_in	入力	TimedDoubleSeq	未使用
q_ref_in	入力	TimedDoubleSeq	全軸の目標関節角度
q_cur_out	出力	TimedDoubleSeq	全軸の現在関節角度

### 2.3. クライアント RTC

#### 2.3.1. 汎用モーションクライアント RTC

汎用モーション RTC に動作指令を行う、画面を持ったクライアントです。



##### 2.3.1.1.ポート

表 14 にデータポートの詳細を示します。

表 15 にサービスポートの詳細を示します。

表 14 データポート

ポート名	入出力	データ型	説明
position_fb_in	入力	TimedDoubleSeq	未使用
caretsian_cmd_in	入力	TimedDoubleSeq	未使用






表 15 サービスポート

ポート名	極性	型	説明
client	要求	IntegratedMotionInterface	汎用モーション RTC への要求

## 3.動作環境セットアップ

### 3.1.ソフトウェアインストール

汎用モーション RTC を動作させるために、以下の手順に従って PC をセットアップしてください。それぞれの手順に関する詳しい説明は、右欄に記載した各節をご覧ください。

1. Java のインストール	「3.1.1 Java のインストール」
	
2. OpenHRP3 のインストール	「3.1.2 OpenHRP3 のインストール」
	
3. Eclipse のインストール	「3.1.3 Eclipse のインストール」
	
4. OpenRTM のインストール	「3.1.4 OpenRTM-aist のインストール」
	
5. 汎用モーション RTC コアのインストール	「3.1.5 汎用モーション RTC コアのインストール」
	
6. 汎用モーション RTC のインストール	「3.1.6 汎用モーション RTC モジュール群のインストール」 「3.1.7 汎用モーション RTC モジュール群のビルド」

#### 3.1.1.Java のインストール

下記 URL を開き、JDK(バージョン 1.6)ファイルをダウンロードし、インストールを行ってください。

URL: <http://www.oracle.com/technetwork/jp/java/javase/downloads/index.html?ssSourceSiteId=ocomjp>

#### 3.1.2.OpenHRP3 のインストール

下記 URL を開き、OpenHRP3 ( ver 3.0.8 ) をダウンロード後、手順にしたがってインストールを行ってください。

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

URL: [http://www.openrtp.jp/openhrp3/jp/install\\_3\\_0.html](http://www.openrtp.jp/openhrp3/jp/install_3_0.html)

### 3.1.3. Eclipse のインストール

下記 URL の「TOP>ダウンロード>GUI ツール関係」の、「Eclipse3.4.2+RTSE+RTCB Linux 用全部入り」をダウンロードしてください。

ダウンロードしたファイルを任意のフォルダへ解凍してください。

URL: <http://www.openrtm.org/openrtm/ja/content/openrtm-eclipse-tools-10-release>

### 3.1.4. OpenRTM-aist のインストール

下記 URL を開き、OpenRTM-aist-Java-1.0.0-RELEASE ( OpenRTM-aist-Java-1.0.0-jar.zip ) をダウンロードし、インストールを行ってください。

URL: <http://www.openrtm.org/openrtm/ja/content/openrtm-aist-java-100-release>

### 3.1.5. 汎用モーション RTC コアのインストール

オープンソース版 ( 東北大学版 ) をダウンロードし、手順にしたがってインストールを行ってください。本マニュアルでは、ダウンロードしたファイルをホームディレクトリに展開します。

( ホームディレクトリ ) /redundancy\_solution\_package

### 3.1.6. 汎用モーション RTC モジュール群のインストール

オープンソース版の汎用モーション RTC をダウンロードします。

ダウンロードしたファイルをホームディレクトリに解凍します。

3.1.5、3.1.6 完了時のディレクトリ構成は以下のようになります。

/	ホームディレクトリ
- /workspace	汎用モーション RTC モジュール群
- ArmConnector	アーム RTC
- DataCoupler	結合 RTC
- DataDistributor	分配 RTC
- IntegratedMotion	汎用モーション RTC
- IntegratedMotionClient	汎用モーションクライアント RTC
- LumbarConnector	腰 RTC
- project	GrxUI のプロジェクトファイル、モデルファイル
- run	起動用シェル
- /redundancy_solution_package	汎用モーション RTC コア

### 3.1.7.汎用モーション RTC モジュール群のビルド

Eclipse を起動し、「ファイル」メニューで「インポート」を選択してください。

「インポート」画面上の「一般」->「既存プロジェクトをワークスペースへ」をクリックし「次へ」ボタンをクリックしてください。

「インポート」画面上の「ルート・ディレクトリーの選択」にチェックを行い、「参照」ボタンをクリックしてください。

3.1.6 で解凍したフォルダから「IntegratedMotion」ディレクトリを選択し、「OK」ボタンをクリックしてください。

インポート可能なプロジェクトの一覧が表示されますので、「IntegratedMotion」にチェックを行った後で、「完了」ボタンをクリックしてください。

パッケージエクスプローラ上で「IntegratedMotion」のプロジェクトを右クリックし、「プロパティ」を選択します。

「Java のビルド・パス」を選択します。

「Java のビルド・パス」画面上の「ライブラリ」タブを選択します。

OpenRTM-aist のライブラリを追加します。

「外部 JAR の追加」ボタンをクリックし、「JAR の選択」画面から OpenRTM-aist のインストール先の OpenRTM-aist/1.0/jar 以下の jar ファイルを追加してください。

vecmath.jar を追加します。

OpenHRP3 のインストール先の OpenHRP-3.0.8/client/gui/Java3D/1.4.1/unix/lib/ext 以下を参照してください。

Eclipse の「プロジェクト」メニューから「プロジェクトのビルド」を実行して下さい。

## 3.2.環境設定

### 3.2.1.プロジェクトファイル、モデルファイルの設定

OpenHRP3(GrxUI)で使用する、プロジェクトファイル、モデルファイルをコピーします。

(ホームディレクトリ)/workspace/project 内のファイルを全て、(OpenHRP3(GrxUI)のホームディレクトリ)/client/gui/project にコピーしてください。

# 4. シミュレーションの実行手順

本システムを使用して、シミュレーションを実行する手順を示します。

## 4.1. OpenHRP3 の起動

### 4.1.1. OpenHRP3 の起動

OpenHRP3(GrxUI)の起動用シェルを実行します。ホームディレクトリから `workspace` フォルダに移動し、`hrp3.sh` を実行してください。

```
$ cd workspace
```

```
$ ./hrp3.sh
```

なお、起動用シェル `hrp3.sh` で記載の環境変数“`OPENHRP3_HOME`”は、お使いの環境に合わせて設定してください。

### 4.1.2. プロジェクトファイルの読み込み

プロジェクトファイルの読み込みを行います。(OpenHRP3(GrxUI)のホームディレクトリ)/`client/gui/project` 以下の“`SmartPal5.xml`”を選択してください。図 3 のように 3D ビューにモデルが表示されます。

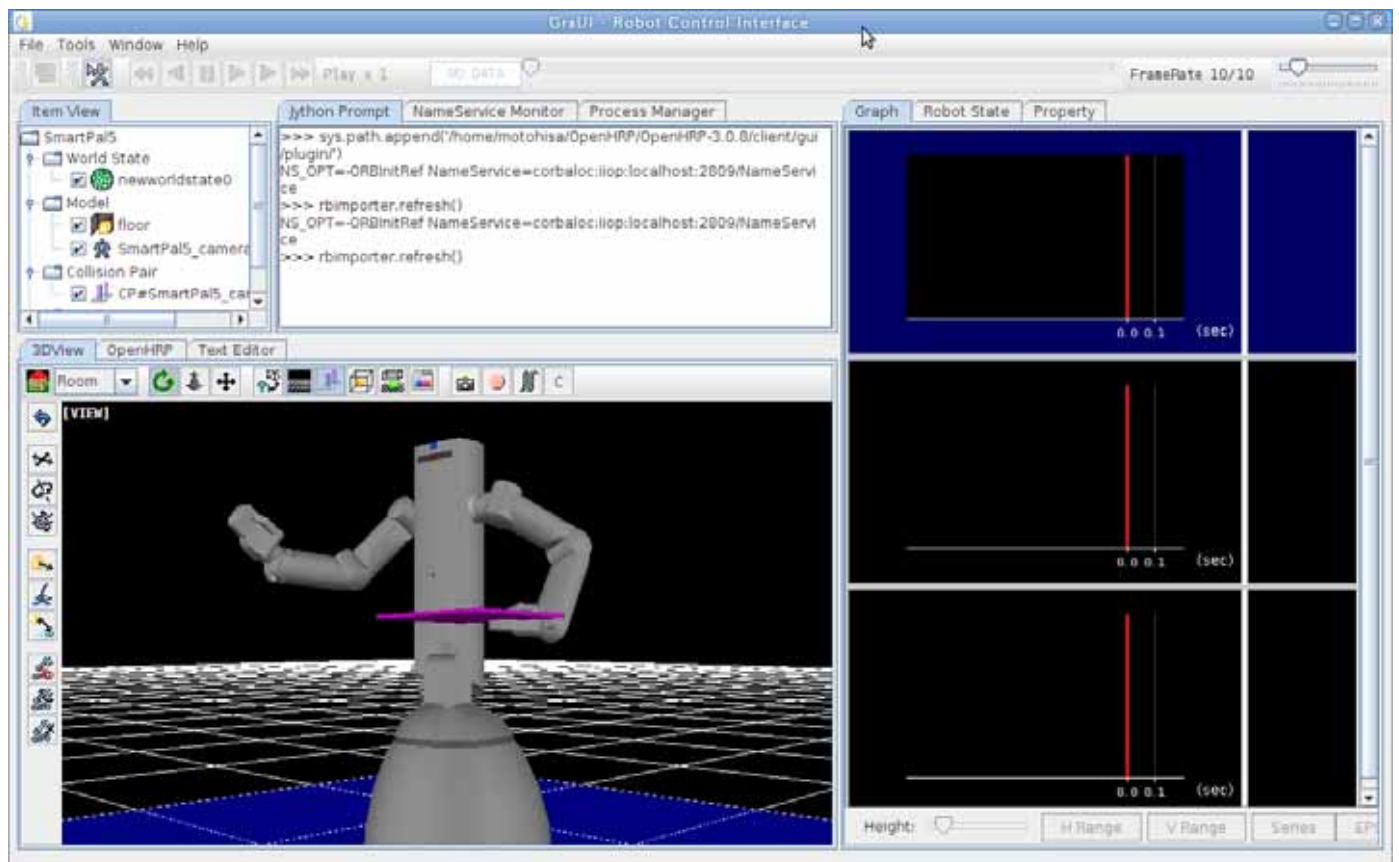


図 3 OpenHRP3(GrxUI)

## 4.2. コンポーネントの起動

以下の手順で各モジュールの起動を行ってください。

### 4.2.1. 汎用モーション RTC コアの起動

汎用モーション RTC コアの起動用シェルを実行します。ホームディレクトリから `workspace` フォルダに移動し、`redundancy_core.sh` を実行してください。

```
$ cd workspace
$ ./redundancy_core.sh
```

なお、起動用シェル `redundancy_core.sh` に記載のコマンドは、汎用モーション RTC コアが 3.1.5 に記載のインストール先に展開されているものとしています。インストール先が異なる場合はお使いの環境に合わせて相対パスを変更してください。

### 4.2.2. 汎用モーション RTC、その他 RTC の起動

汎用モーション RTC、汎用モーションクライアント RTC、アーム RTC、腰 RTC、結合 RTC、分配 RTC の起動用シェルを実行します。ホームディレクトリから `workspace/run` フォルダに移動し、`start.sh` を実行してください。

```
$ cd workspace/run
$ ./start.sh
```

### 4.2.3. コントローラの起動

SmartPalController の起動用シェルを実行します。ホームディレクトリから `workspace` フォルダに移動し、`smartpal.sh` を実行してください。

```
$ cd workspace
$ ./smartpal.sh
```

### 4.3. コンポーネントの接続

Eclipse を起動し、RT System Editor で、図 4 を参照し、各 RTC を接続して下さい。  
接続が完了しましたら、全てのコンポーネントを Activate してください。

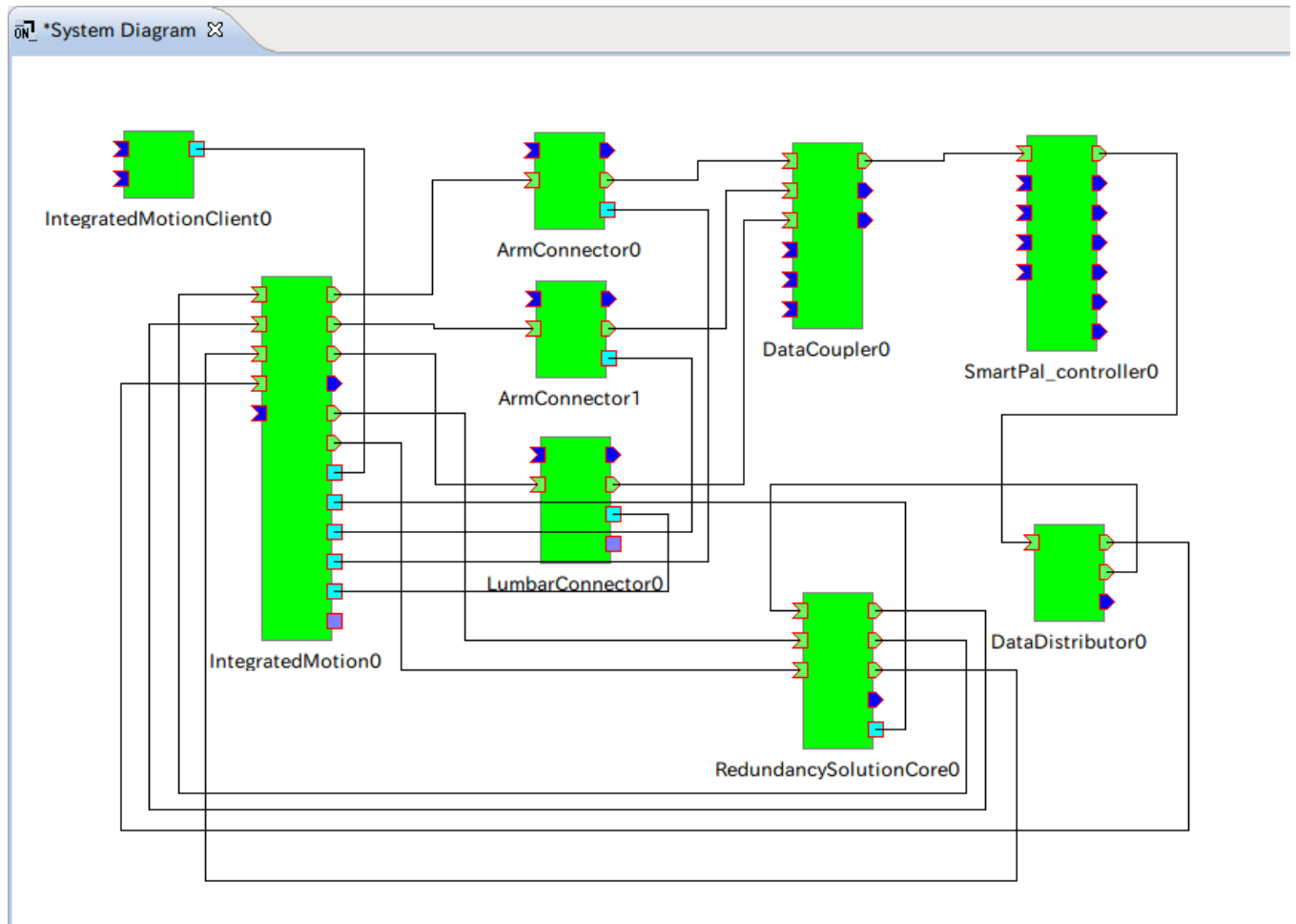


図 4 接続図

## 4.4.汎用モーションクライアント RTC の操作

### 4.4.1.汎用モーションクライアント RTC の説明

汎用モーションクライアント RTC の画面は図 5 のように、サーボコマンドエリア・モニターエリア・オペレーションコマンドエリアから構成されます。サーボエリアは、サーボの ON、OFF が行えます。モニターエリアは、各関節の角度、左右の制御点の直交座標が表示されます。オペレーションコマンドエリアは、機能ごとにタブで分かれています。

**Servo**

サーボコマンドエリア

ON OFF

movJ\_RArmAbs movJ\_RA

☒ right ☐ left ☐ isHold

MoveUnitR\_ArmAbs

J1 90

J2 60

J3 -90

J4 オペレーション  
コマンドエリア

J5 0

J6 -90

J7 0

vel[deg/s] 10.0

accel[deg/s<sup>2</sup>] 10.0

send

**Monitor**

Status

servo motion

モニターエリア

Feedback

R_Joint		L_Joint		R_Cartesian		L_Cartesian	
J1	57.29747	J1	153.55722	x	386.89496	x	-69.93889
J2	57.29747	J2	-70.47589	y	-316.49500	y	338.69411
J3	-68.75696	J3	-102.56247	z	957.18796	z	777.04665
J4	68.75696	J4	-97.97867	r11	0.79720	r11	0.99687
J5	-28.64873	J5	-107.14627	r12	-0.59873	r12	-0.07036
J6	-57.29747	J6	100.27057	r13	0.07735	r13	0.03615
J7	0.00000	J7	54.43260	r21	0.16238	r21	-0.07127
ry	0.00000	--	0.00000	r22	0.08925	r22	-0.99716
x	0.00000	--	0.00000	r23	-0.98268	r23	0.02452
y	0.00000	--	0.00000	r31	0.58146	r31	0.03432
th	0.00000	--	0.00000	r32	0.79596	r32	-0.02702
				r33	0.16838	r33	-0.99905
				elb	138.22542	elb	-62.01283

[servoOn] id: 0 comment: OK

Monitor Stop

Pause Resume abort

getState getVersion clearAlarm getAlarm

図 5 汎用モーションクライアント RTC



## 4.5. シミュレーションの実行手順

本システムの簡単な操作手順を示します。

シミュレーションの実行準備

「4 章シミュレーションの実行」に従い、各コンポーネントの起動、RTC の接続を行ってください。

シミュレーションの開始

OpenHRP3(GrxUI)の”start simulation”ボタンを押下してください。

“ Restart the Controller ” のウィンドウが表示されますので “ No ” を選択してください。



モニターを開始します。

汎用モーションクライアント RTC 画面 ( 図 5 ) のモニターエリアの”Monitor Start”ボタンを押下してください。

サーボ制御を入れます。

図 6 の画面の左上にあるサーボコマンドエリアの”ON”ボタンを押下してください。

モニターエリアの “ servo ” アイコンが点灯します。

右アームの各関節軸の直線補間を動作させます。

図 6 に示すように”movJ\_RArmAbs”タブに切替えてください。

四角 で囲まれた部分に値を入力後、”send”ボタンを押下してください。

( 本タブにおけるチェックボックスの設定は動作に影響しません。 )

モーション動作中はモニターエリアの “ motion ” アイコンが点灯します。

OpenHRP3 の 3D ビュー上でモデルの右アームが目標位置へ動作することを確認してください。( 図 7 )



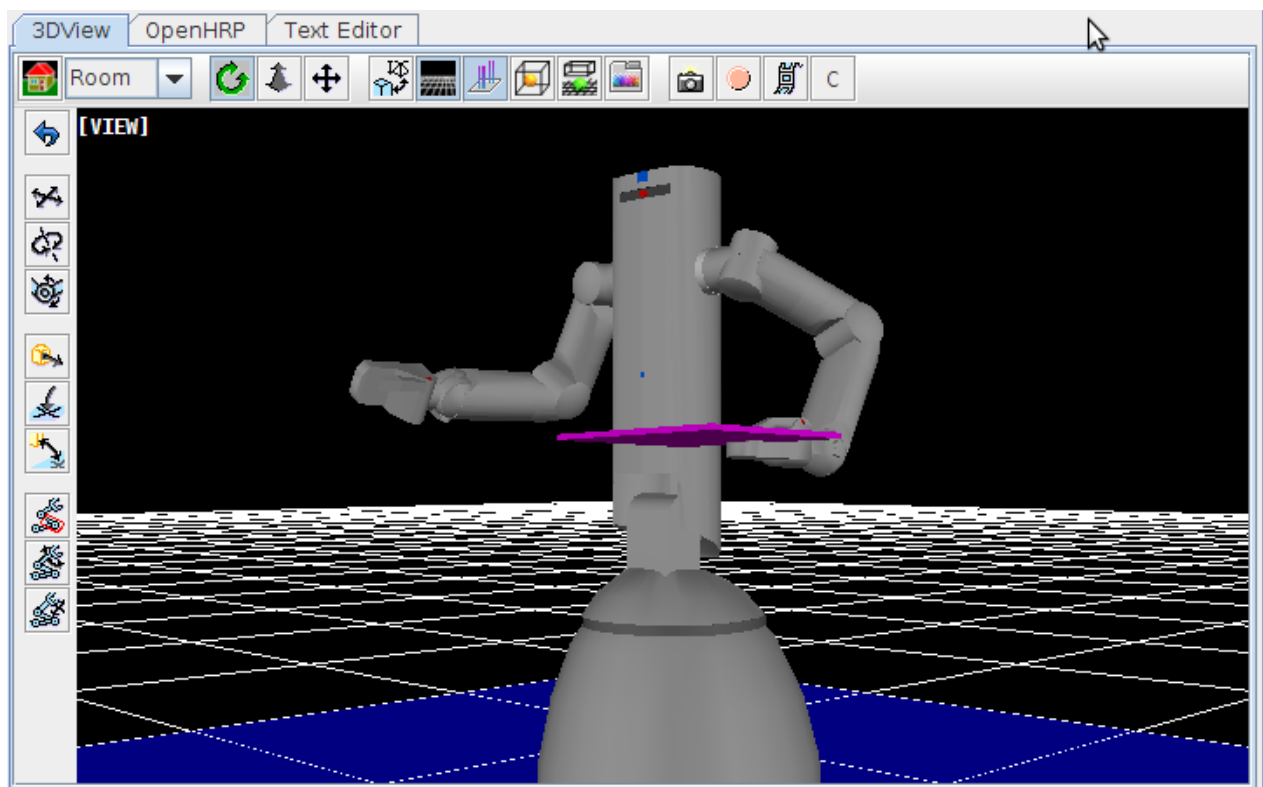


図 7 右アームの各関節軸の直線補間後の姿勢

左アームの各関節軸の直線補間を動作させます。

図 8 に示すように”movJ\_LArmAbs”タブに切替えてください。

四角 で囲まれた部分に値を入力後、”send”ボタンを押下してください。

(本タブにおけるチェックボックスの設定は動作に影響しません。)

OpenHRP3 の 3D ビュー上でモデルの左アームが目標位置へ動作することを確認してください。(図 9)

Servo

ON OFF

\_RArmRel **movJ\_LArmAbs**

☒ right ☐ left ☐ isHold

MoveUnitL\_ArmAbs

J1	90
J2	-30
J3	-60
J4	-30
J5	-60
J6	30
J7	20
vel[deg/s]	10.0
accel[deg/s2]	10.0

send

Monitor

Status

☒ servo ☐ motion

Feedback

R_Joint		L_Joint		R_Cartesian		L_Cartesian	
J1	90.00000	J1	90.00000	x	275.00000	x	98.56406
J2	60.00000	J2	-30.00000	y	-335.00000	y	599.49674
J3	-90.00000	J3	-60.00000	z	892.67314	z	813.47441
J4	90.00000	J4	-30.00000	r11	1.00000	r11	0.99765
J5	0.00000	J5	-60.00000	r12	-0.00000	r12	-0.02901
J6	-90.00000	J6	30.00000	r13	0.00000	r13	-0.06204
J7	0.00000	J7	20.00000	r21	0.00000	r21	0.06810
ry	0.00000	--	0.00000	r22	0.50000	r22	0.51615
x	0.00000	--	0.00000	r23	-0.86603	r23	0.85379
y	0.00000	--	0.00000	r31	0.00000	r31	0.00726
th	0.00000	--	0.00000	r32	0.86603	r32	-0.85601
				r33	0.50000	r33	0.51691
				elb	142.16189	elb	-69.52962

[servoOn] id: 0 comment: OK

Monitor Stop

Pause Resume abort

getState getVersion clearAlarm getAlarm

図 8 左アームの各関節軸の直線補間例

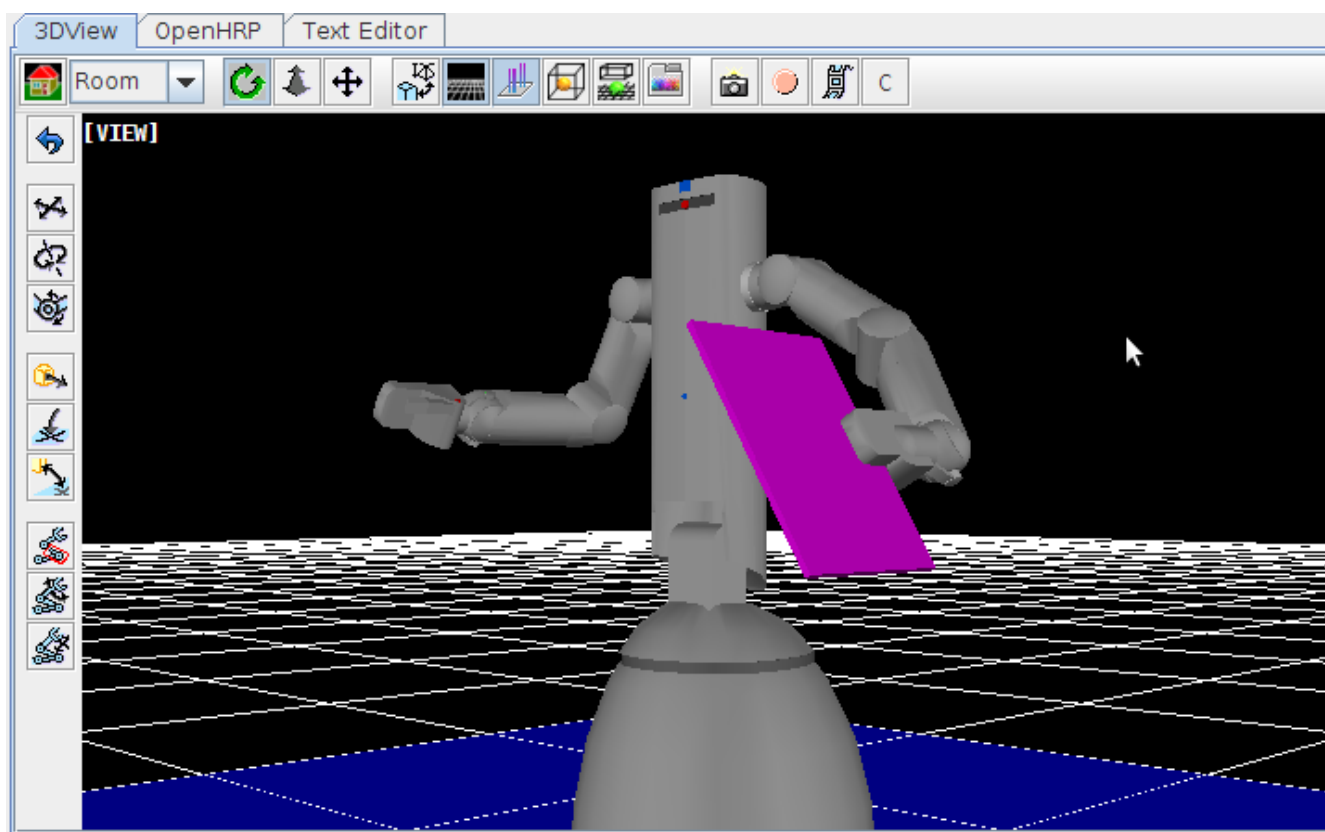


図 9 左アームの各関節軸の直線補間後の姿勢

右アームの制御点の直線補間を動作させます。

図 10 に示すように”movLinearRel”タブに切替えてください。

四角 で囲まれた部分の”right”にチェックを入れ、値を入力後、”send”ボタンを押下してください。

OpenHRP3 の 3D ビュー上でモデルの右アームが目標位置（直交座標）へ動作することを確認してください。（図 11）

”isHold”にチェックを入れた場合は、左アームの制御点の位置を保持します。

The screenshot displays the OpenHRP3 control interface. On the left, the 'Servo' section has 'ON' and 'OFF' buttons. Below it, the 'movLinearAbs' and 'movLinearRel' tabs are visible, with 'movLinearRel' selected. A blue rectangle highlights the 'right' checkbox (checked), 'left' checkbox (unchecked), 'isHold' checkbox (unchecked), and a list of input fields for 'MoveLinearRel' (x, y, z, r11, r12, r13, r21, r22, r23, r31, r32, r33, t vel, r vel, t accel, r accel). The 'send' button is at the bottom of this section. On the right, the 'Monitor' section shows 'Status' with 'servo' (green) and 'motion' (grey) indicators. Below is a 'Feedback' table with columns for 'R\_Joint', 'L\_Joint', 'R\_Cartesian', and 'L\_Cartesian'. The table contains numerical values for various joints and Cartesian coordinates. At the bottom of the monitor section is a 'Monitor Stop' button. The very bottom of the interface has a row of buttons: 'Pause', 'Resume', 'abort', 'getState', 'getVersion', 'clearAlarm', and 'getAlarm'.

R_Joint		L_Joint		R_Cartesian		L_Cartesian	
J1	90.00000	J1	90.00000	x	275.00000	x	98.56406
J2	60.00000	J2	-30.00000	y	-335.00000	y	599.49674
J3	-90.00000	J3	-60.00000	z	892.67314	z	813.47441
J4	90.00000	J4	-30.00000	r11	1.00000	r11	0.99765
J5	0.00000	J5	-60.00000	r12	-0.00000	r12	-0.02901
J6	-90.00000	J6	30.00000	r13	0.00000	r13	-0.06204
J7	0.00000	J7	20.00000	r21	0.00000	r21	0.06810
ry	0.00000	--	0.00000	r22	0.50000	r22	0.51615
x	0.00000	--	0.00000	r23	-0.86603	r23	0.85379
y	0.00000	--	0.00000	r31	0.00000	r31	0.00726
th	0.00000	--	0.00000	r32	0.86603	r32	-0.85601
				r33	0.50000	r33	0.51691
				elb	142.16189	elb	-69.52962

[servoOn] id: 0 comment: OK

Monitor Stop

Pause Resume abort getState getVersion clearAlarm getAlarm

図 10 右アームの制御点の直線補間例

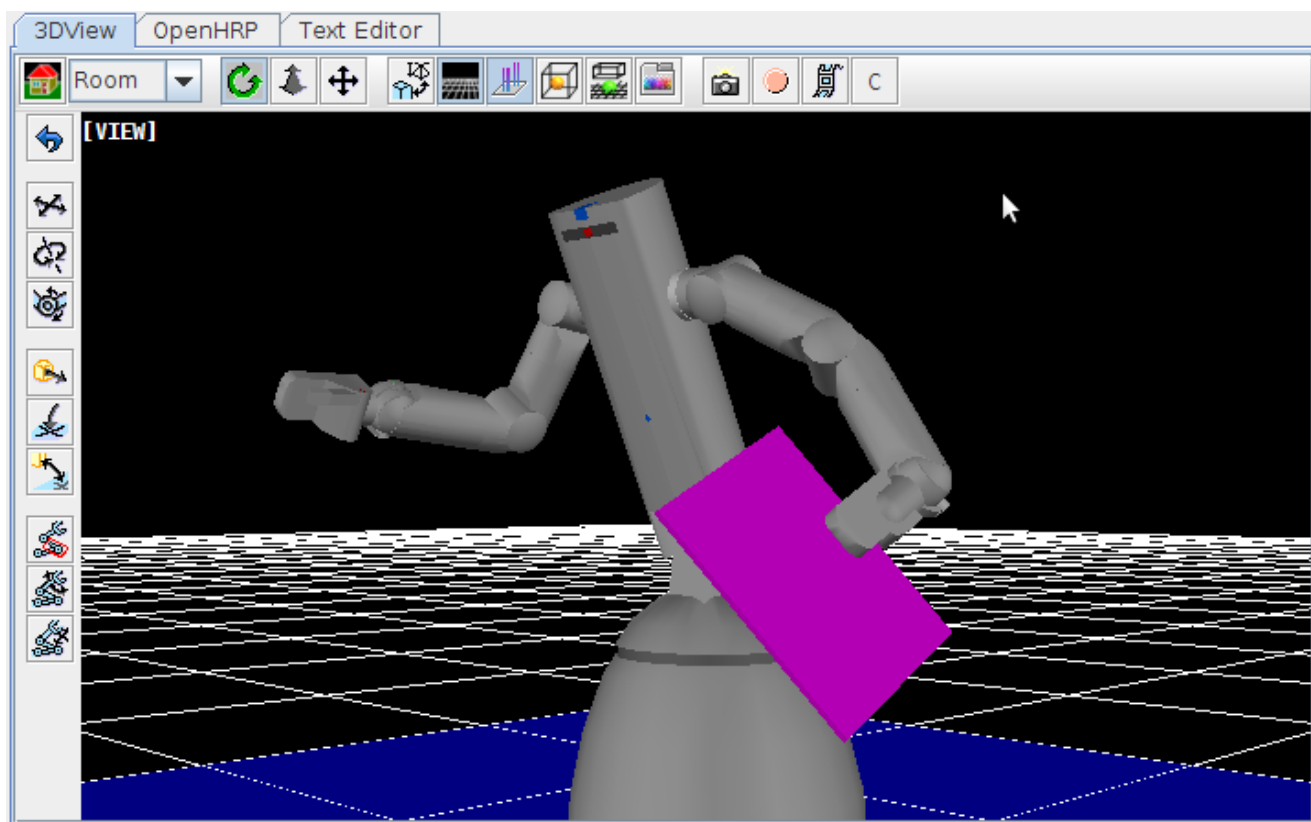


図 11 右アーム制御点の直線補間後の姿勢

使用可能な冗長軸の指定をします。

図 12 に示すように”selectRedun”タブに切替えてください。

動作可能な冗長軸を指定します。チェックを外した冗長軸は、直線補間動作時（手順 ）に固定された状態になります。初期設定は肘角、腰軸がいずれも動作可能になっています。

腰軸 “ lumbarRy ” のチェックを外した後、“send”ボタンを押下してください。

なお、設定が有効な冗長軸は”elbowR”と”lumbarRy”のみです。それ以外の項目については将来機能です。

Servo

ON OFF

selectRedun movLinearAl

☒ elbowR

☐ elbowL

☐ lumbarRy

☐ vehicleX

☐ vehicleY

☐ vehicleTh

send

Monitor

Status

☒ servo ☐ motion

Feedback

R_Joint		L_Joint		R_Cartesian		L_Cartesian	
J1	58.11143	J1	90.00000	x	475.00000	x	129.28041
J2	59.31026	J2	-30.00000	y	-335.00000	y	599.49674
J3	-89.05561	J3	-60.00000	z	892.67314	z	754.70642
J4	89.42085	J4	-30.00000	r11	1.00000	r11	0.90907
J5	-0.59772	J5	-60.00000	r12	0.00000	r12	-0.38468
J6	-95.59230	J6	30.00000	r13	-0.00000	r13	0.16004
J7	2.64074	J7	20.00000	r21	-0.00000	r21	0.06810
ry	24.74737	--	0.00000	r22	0.50000	r22	0.51615
x	0.00000	--	0.00000	r23	-0.86603	r23	0.85379
y	0.00000	--	0.00000	r31	0.00000	r31	-0.41104
th	0.00000	--	0.00000	r32	0.86603	r32	-0.76525
				r33	0.50000	r33	0.49541
				elb	144.36034	elb	-38.04080

[servoOn] id: 0 comment: OK

Monitor Stop

Pause Resume abort

getState getVersion clearAlarm getAlarm

図 12 冗長軸指定の例



再度、右アームの制御点の直線補間を動作させます。

図 13 に示すように”movLinearRel”タブに切替えてください。

四角 で囲まれた腰軸の現在位置を確認してください。

値を入力後、”send”ボタンを押下してください。

OpenHRP3 の 3D ビュー上でモデルの右アームが腰軸の位置を固定したまま目標位置へ動作することを確認してください。(図 14、図 15)

The screenshot shows the OpenHRP3 control interface. The 'Servo' section has 'ON' and 'OFF' buttons. The 'Monitor' section has 'Status' (servo, motion) and 'Feedback' data. The 'Feedback' data is organized into four columns: R\_Joint, L\_Joint, R\_Cartesian, and L\_Cartesian. The 'ry' joint value is highlighted with a blue box. The 'Send' button is at the bottom left, and the 'Monitor Stop' button is at the bottom right.

R_Joint		L_Joint		R_Cartesian		L_Cartesian	
J1	58.11143	J1	90.00000	x	475.00000	x	129.28041
J2	59.31026	J2	-30.00000	y	-335.00000	y	599.49674
J3	-89.05561	J3	-60.00000	z	892.67314	z	754.70642
J4	89.42085	J4	-30.00000	r11	1.00000	r11	0.90907
J5	-0.59772	J5	-60.00000	r12	0.00000	r12	-0.38468
J6	-95.59230	J6	30.00000	r13	-0.00000	r13	0.16004
J7	2.64074	J7	20.00000	r21	-0.00000	r21	0.06810
ry	24.74737	--	0.00000	r22	0.50000	r22	0.51615
x	0.00000	--	0.00000	r23	-0.86603	r23	0.85379
y	0.00000	--	0.00000	r31	0.00000	r31	-0.41104
th	0.00000	--	0.00000	r32	0.86603	r32	-0.76525
				r33	0.50000	r33	0.49541
				elb	144.36034	elb	-38.04080

[servoOn] id: 0 comment: OK

Monitor Stop

Pause Resume abort getState getVersion clearAlarm getAlarm

図 13 右アームの制御点の直線補間

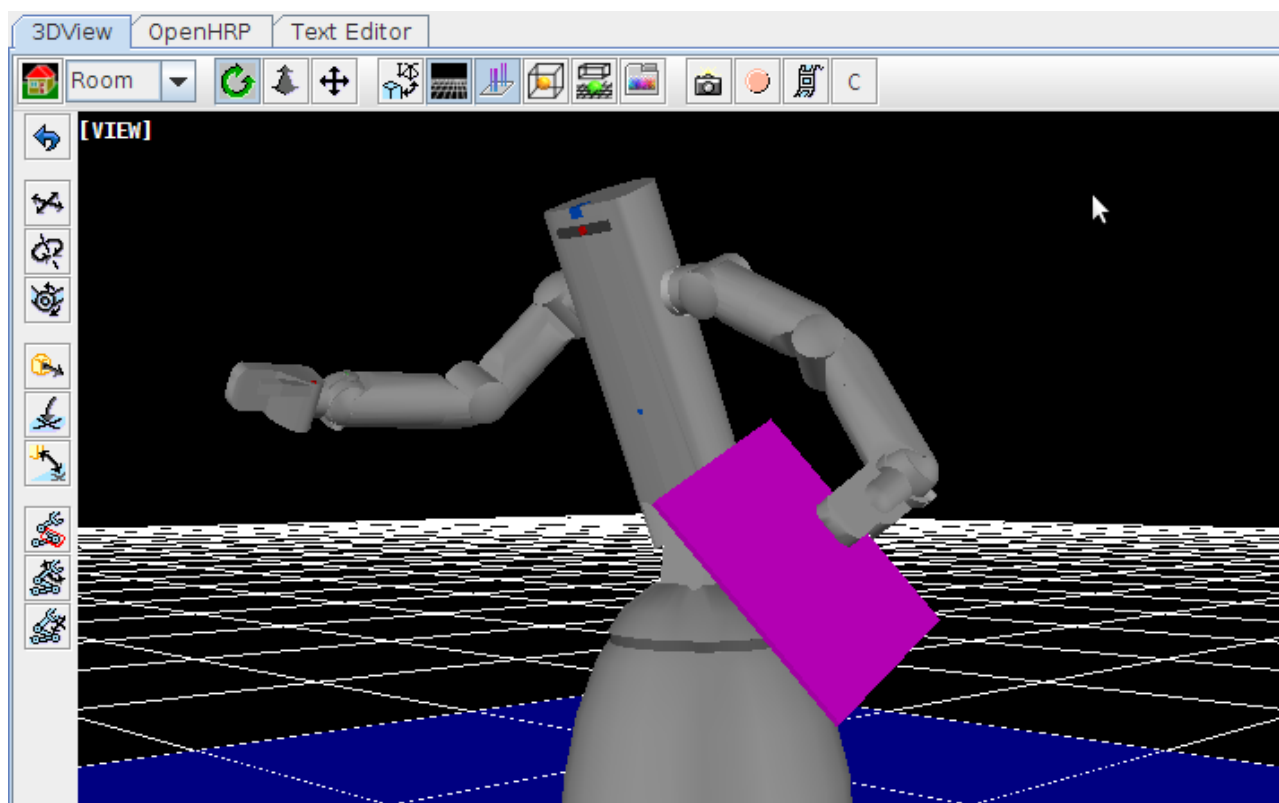


図 14 右アーム制御点の直線補間後の姿勢（腰軸固定）

Feedback							
R_Joint		L_Joint		R_Cartesian		L_Cartesian	
J1	29.75927	J1	90.00000	x	575.00000	x	129.28041
J2	66.01131	J2	-30.00000	y	-335.00000	y	599.49674
J3	-79.52996	J3	-60.00000	z	892.67314	z	754.70642
J4	54.56102	J4	-30.00000	r11	1.00000	r11	0.90907
J5	4.36416	J5	-60.00000	r12	-0.00000	r12	-0.38468
J6	-86.36719	J6	30.00000	r13	0.00000	r13	0.16004
J7	4.56499	J7	20.00000	r21	0.00000	r21	0.06810
ry	24.74737	--	0.00000	r22	0.50000	r22	0.51615
x	0.00000	--	0.00000	r23	-0.86603	r23	0.85379
y	0.00000	--	0.00000	r31	-0.00000	r31	-0.41104
th	0.00000	--	0.00000	r32	0.86603	r32	-0.76525
				r33	0.50000	r33	0.49541
				elb	151.88209	elb	-38.04080

図 15 直線補間後の腰軸位置

サーボ制御を切ります。

サーボコマンドエリアの OFF ボタンを押下してください。