HIRO-AGV 統合制御モジュールコンポーネント群

平成 24 年 2 月 24 日

豊橋技術科学大学　行動知能システム学研究室

目次

1　このコンポーネント群について

このコンポーネント群は，状態機械コンポーネントを中心とした，物体のピック&プレース作業を行う双腕ロボット HIRO(川田工業株式会社，ゼネラルロボティックス株式会社)と，物体を自動運搬する，Patrafour(関東自動車工業株式会社)を使用した AGV を統合制御するコンポーネント群である．

本コンポーネント群の統合制御は，事前に定義した状態遷移シナリオに従って動作する状態機械に基づいて行われる(例えば，初期状態→AGV 往路移動→物体位置姿勢推定→ピック&プレース→…→AGV 復路移動→終了状態(詳しくは後述))． なお，物体位置姿勢推定は本研究室作成の「物体位置姿勢推定 RTC」(http://openrtm.org/openrtm/ja/project/NEDO_Intelligent_PRJ_ID383)を，双腕ロボット HIRO の動作には共通インタフェースである HiroNXInterface (http://www.openrtm.org/openrtm/ja/project/NEDO_Intelligent_PRJ_HiroAccPrj_1003)を利用する．

2　開発・動作環境

本コンポーネント群は以下の環境で開発し，動作確認をしている．

- Windows 7 Professional 64bit，一部 Windows XP Professional，Ubuntu 10.04
- Open-rtm-aist 1.0.0(C++版)
- rtshell-3.0.0，及び，rtctree-3.0.0
- Microsoft Visual Studio 2008
- OpenCV 2.1
- Google Chrome，Firefox，Android 標準ブラウザ

3　コンポーネント構成

本コンポーネント群に含まれる RTC は以下の物である.

3.1　状態機械－各部制御に関連する RTC（AGV，物体位置姿勢推定，HIRO）

RTC 名	説明
StateMachine	状態機械.
AGVController	AGV 制御.
ObjectPoseEstimationGate	物体位置姿勢推定 RTC 制御.
HiroPickerAndPlacer	HIRO ピック&プレース動作計画・実行制御.

3.2　AGV の動作に関連する RTC

RTC 名	説明
PathPlannerV2	AGV の壁沿い走行経路計画.

3.3　HIRO の動作に関連する RTC

RTC 名	説明
HiroGate	HIRO の RTC への接続. ここでは，逆運動学を解くためのみに使用.

4　各コンポーネントについて

4.1　StateMachine

本コンポーネントは，状態機械の状態，入力，出力を記述した XML ファイルに従って，接続される各 RTC の動作を制御するためのコンポーネントである．HTTP サーバが組み込まれており，ユーザは Web ブラウザを通して状態機械の状態の確認，遷移の決定などを行うことができる．

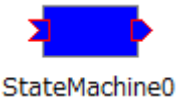


図 4.1-1 StateMachine

AGVController, ObjectPoseEstimationGate, HiroPickerAndPlacer の各コンポーネントは，本コンポーネントの状態遷移時の出力（＝各コンポーネントへの入力）を受け取り，その出力に応じた処理を行う．さらに各コンポーネントは処理結果に応じて，本コンポーネントの状態機械へ入力（＝各コンポーネントからの出力）を行い，それをトリガとして状態遷移が生じて，処理が進行する．

4.1.1 データポート

ポート名	データ型	入出力	備考
input	RTC::TimedStringSeq	Inport	状態機械への入力.
output	RTC::TimedStringSeq	Outport	状態機械からの出力.

4.1.2 サービスポート

なし.

4.1.3 各ポートのデータ型・インタフェースについて

- input（RTC::TimedStringSeq）

メンバ名	データ型	備考
tm	RTC::Time	未使用.
data	StringSeq	data[0]のみを状態機械への入力に利用.

- output（RTC::TimedStringSeq）

メンバ名	データ型	備考
tm	RTC::Time	未使用.
data	StringSeq	data[0]のみを状態機械からの出力に利用.

4.1.4 コンフィギュレーションパラメータ

パラメータ名	データ型	備考
scenario_path	string	状態機械を記述したシナリオファイルへのパス.
tcp_port	unsigned short	HTTP サーバの TCP ポート番号.

4.1.5 状態機械の記述ファイルについて

状態機械は XML ファイルとして記述する．詳しくは，同梱の scenario.xml を参照．

4.1.6 ビルド方法

現在，準備中．

4.1.7 ライセンスについて

本コンポーネントでは，XML ファイルの読み込みライブラリとして，TinyXML（http://www.grinninglizard.com/tinyxml/）を利用している．このライブラリに関しては，zlib/libpng License で提供されている．

4.2 AGVController

本コンポーネントは，前述の状態機械コンポーネント StateMachine からの出力を受けて AGV の経路計画を行うコンポーネント PathPlannerV2（後述，4.5）を動作させるためのものである．

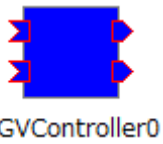


図 4.2-1 AGVController

4.2.1 データポート

ポート名	データ型	入出力	備考
input	RTC::TimedStringSeq	Inport	状態機械からの本 RTC 入力（状態機械からの出力 output と接続）.
output	RTC::TimedStringSeq	Outport	状態機械への本コンポーネントからの出力（状態機械への入力 input と接続）.
fromPlanner	RTC::TimedString	Inport	AGV 経路計画 RTC からの往路・復路の終了（到着）メッセージを受け取る.
toPlanner	RTC::TimedString	Outport	AGV 経路計画 RTC への往路・復路の開始（出発）メッセージを送る.

4.2.2 各ポートのデータ型・インタフェースについて

- input（RTC::TimedStringSeq）

メンバ名	データ型	備考
tm	RTC::Time	未使用.
data	StringSeq	data[0]のみを本 RTC への入力に利用.

- output（RTC::TimedStringSeq）

メンバ名	データ型	備考
tm	RTC::Time	未使用.
data	StringSeq	data[0]のみを本 RTC からの出力に利用.

- fromPlanner （RTC::TimedString）

メンバ名	データ型	備考
tm	RTC::Time	未使用.
data	String	AGV 経路計画 RTC からの本 RTC へのメッセージ.

- toPlanner （RTC::TimedString）

メンバ名	データ型	備考
tm	RTC::Time	未使用.
data	String	本 RTC から AGV 経路計画 RTC へのメッセージ.

4.2.3 サービスポート

なし.

4.2.4 コンフィギュレーションパラメータ

なし.

4.2.5 ビルド方法

現在，準備中.

4.3 ObjectPoseEstimationGate

本コンポーネントは，前述の状態機械コンポーネント StateMachine からの出力を受けて，物体位置姿勢推定コンポーネントである ObjectPoseEstimation（http://openrtm.org/openrtm/ja/project/NEDO_Intelligent_PRJ_ID383）を動作させるためのコンポーネントである．推定に関するデータ・サービスポートは，共通インタフェースに準拠している．また，双腕ロボット HIRO のピック&プレース計画・実行コンポーネントである HiroPickerAndPlacer へ，推定結果から計算したパラメータを送るサービスを提供する．

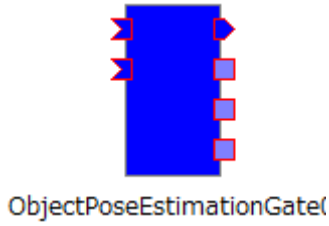


図 4.3-1 ObjectPose EstimationGate

4.3.1 データポート

ポート名	データ型	入出力	備考
input	RTC::TimedStringSeq	Inport	状態機械からの本 RTC への入力（状態機械からの出力 output と接続）.
output	RTC::TimedStringSeq	Outport	状態機械への本 RTC からの出力（状態機械への入力 input と接続）.
RecognitionResult	RTC::TimedDoubleSeq	Inport	物体の位置姿勢推定結果を受け取る.

4.3.2 サービスポート

ポート名	インタフェース型	Provider/ Consumer	備考
RecognitionService	RecognitionService	Consumer	物体の位置姿勢機能を呼び出す.
MDService	ModelDefinitionRTC::idl::MDService	Consumer	物体のモデル情報を参照する機能を呼び出す.
ObjectPoseEstimationGateService	ObjectPoseEstimationGateRTC::ObjectPoseEstimationGateService	Provider	双腕ロボット HIRO のピック&プレース計画・実行コンポーネントへ必要なパラメータ計算・送出するサービスを提供する.

4.3.3 各ポートのデータ型・インタフェースについて

- input（RTC::TimedStringSeq）

メンバ名	データ型	備考
tm	RTC::Time	未使用.
data	StringSeq	data[0]のみを本 RTC への入力に利用.

- output（RTC::TimedStringSeq）

メンバ名	データ型	備考
tm	RTC::Time	未使用.
data	StringSeq	data[0]のみを本 RTC からの出力に利用.

- RecognitionResult（RTC::TimedDoubleSeq）
- RecognitionService（RecognitionService）
- MDService（ModelDefinitionRTC::idl::MDService）

これらデータ型の詳細は，物体位置姿勢推定コンポーネント ObjectPoseEtimation（http://openrtm.org/openrtm/ja/project/NEDO_Intelligent_PRJ_ID383）を参照.

- ObjectPoseEstimationGateService
  （ObjectPoseEstimationGateRTC::ObjectPoseEstimationGateService）

サービス名	引数/返戻値	備考
long getLastEstimated Result	in RTC::Pose3D W2R_m_rad	ロボット座標系での，物体位置姿勢推定 RTC の 世界座標系 W の位置姿勢. 位置は[m]単位，角度は[radian]単位. 回転は水平面上のみと仮定しており， W2R_m_rad.orientaion.y のみを利用する.
	out long id	推定結果のモデル ID.
	out RTC::Size3D size_m	推定結果のモデルの高さと幅. 単位は[m].
	out RTC::Pose3D poseR_m_rad	ロボット座標系 R での物体の位置姿勢. 位置は[m]単位，角度は[radian]単位.
	返戻値	正常時：物体 ID，異常時：負数
	推定結果の物体情報を 1 つずつ取得する.	

4.3.4 コンフィギュレーションパラメータ

なし.

4.3.5 ビルド方法

現在，準備中.

4.4 HiroPickerAndPlacer

本コンポーネントは，前述の状態機械コンポーネント StateMachine からの出力を受けて，双腕ロボット HIRO に物体のピック&プレースを行わせるために，その計画・実行を行うコンポーネントである．

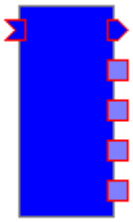


図 4.4-1 HiroPickerAndPlacer

4.4.1 データポート

ポート名	データ型	入出力	備考
input	RTC::TimedStringSeq	Inport	状態機械からの本 RTC への入力（状態機械からの出力 output と接続）.
output	RTC::TimedStringSeq	Outport	状態機械への本 RTC からの出力（状態機械への入力 input と接続）.

4.4.2 サービスポート

ポート名	インタフェース型	Provider/ Consumer	備考
HiroGateService	HiroGateRTC:: HiroGateService	Consumer	逆運動学の解の有無を調べるサービスを呼び出す.
ObjectPose EstimationGate Service	ObjectPoseEstimation GateRTC::ObjectPose EstimationGateService	Consumer	ピック＆プレース計画に必要なパラメータ計算・送出するサービスを呼び出す.
HiroNX	HiroNX	Consumer	HiroNXInterface により提供されるサービスを呼び出す. HiroNXInterface を構成する RTC, HiroNXProvider の同名のポートへ接続.
HIRO	CommonCommands, MotionCommands	Consumer	同上.

4.4.3 各ポートのデータ型・インタフェースについて

- input（RTC::TimedStringSeq）

メンバ名	データ型	備考
tm	RTC::Time	未使用.
data	StringSeq	data[0]のみを本 RTC への入力に利用.

- output（RTC::TimedStringSeq）

メンバ名	データ型	備考
Tm	RTC::Time	未使用.
Data	StringSeq	data[0]のみを本 RTC からの出力に利用.

- HiroGateService（HiroGateRTC::HiroGateService）
  後述する 4.6.3 を参照.

- ObjectPoseEstimationGateService
  （ObjectPoseEstimationGateRTC::ObjectPoseEstimationGateService）
  前述した 4.3.3 の ObjectPoseEstimationGateService の項目を参照.

- HiroNX（HiroNX）
- HIRO（CommonCommands, MotionCommands）
  これらのポートのインタフェースについては，HiroNXInterface（http://www.openrtm.org/openrtm/ja/project/NEDO_Intelligent_PRJ_HiroAccPrj_1003）を参照.

4.4.4 コンフィギュレーションパラメータ

パラメータ名	データ型	備考
xWorldR_m	double	ロボット座標系 R での世界座標系 W（物体位置姿勢推定 RTC の世界座標）の x 座標[m]．
yWorldR_m	double	同 y 座標[m]．
zWorldR_m	double	同 z 座標[m]．
yWorldR_deg	double	同ヨー方向(ロボット系 z 軸周り)回転角度[degree]．
xPalletPlacedB_m	double	胸部関節の上に固定した胴体座標系 B（胸部と共に回転）でのパレットが置かれた位置の x 座標[m]．
yPalletPlacedB_m	double	同 y 座標[m]．通常 0（HIRO の胴体正面に置く）.
zPalletPlacedB_m	double	同 z 座標[m]．
yPalletPlacedB_deg	double	上記位置に置かれたパレットを掴むときの手の姿勢のヨー方向角度[degree]（y→p→r の順番のオイラー角指定）. 通常 0°．
pPalletPlacedB_deg	double	同ピッチ方向角度[degree]. 通常－90°．
rPalletPlacedB_deg	double	同ロール方向角度[degree]. 通常 0°．
directionPalletPlacedR_deg	double	ロボット座標系 R での パレットが置かれた方向[degree].
liftDistancePalletR_m	double	パレットを持ち上げる距離[m]．

xPalletToBePlacedB_m	double	胴体座標系 B でのパレットを置く位置の x 座標[m] .
yPalletToBePlacedB_m	double	同 y 座標[m]. 通常 0（HIRO の胴体正面に置く）.
zPalletToBePlacedB_m	double	同 z 座標[m] .
directionPalletToBePlacedR_deg	double	ロボット座標系 R でのパレットを置く方向[degree].
holdDistance_m	double	物体を掴むときの手の狭め幅[m]．掴む強さ・指の平行度の調整用．大きくすると強く掴む.
releaseDistance_m	double	物体を離すときの手の広げ幅[m].
fingerBacklash_deg	double	指の関節の遊び角[degree]．掴む強さ・指の平行度の調整用．大きくすると強く掴む.
xOPEOffsetR_m	double	ロボット座標系 R での水平面のずれの x 方向の微調整用[m].
yOPEOffsetR_m	double	ロボット座標系 R での水平面のずれの y 方向の微調整用[m].
speedBase	double	動作の基底速度．各動作はこの速度の 0.5 倍，1.2 倍…といった速度で動く．単位は%. HIRO の ArmControlService::setTargetAngular と同じ指定方法.
serialNo	double	HIRO のシリアル番号．内部での処理の分岐（主に追加のセンサによる位置オフセット調整）のために使用.

4.4.5 ビルド方法

現在，準備中.

4.5 PathPlannerV2

本コンポーネントは，前述の AGV 制御コンポーネント AGVController（4.2）から送られたメッセージ（往路・復路の開始）を受けて，AGV の壁沿い走行の経路計画・指示を行うコンポーネントである．本コンポーネントは，本研究室で開発した経路計画 RTC（http://www.openrtm.org/openrtm/ja/project/NEDO_Intelligent_PRJ_ID298）を壁沿い走行のために修正したものである．

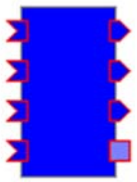


図 4.5-1 PathPlannerV2

4.5.1 データポート

ポート名	データ型	入出力	備考
robot_pose	IIS::TimedPose2D	Inport	ロボットの現在位置.
robot_velocity	IIS::TimedVelocity2D	Inport	ロボットの現在速度.
subgoal_waypoint	IIS::PoseVel2DSeq	Inport	目標地点の系列.
in_message	RTC::TimedString	Inport	AGVController からの往路・復路開始メッセージ.
robot_control	IIS::TimedVelocity2D	Outport	制御出力（速度）.
planning_img	TUT::TimedImageData	Outport	処理経過画像出力.
message	RTC::TimedString	Outport	AGVController への往路・復路終了メッセージ.

4.5.2 サービスポート

ポート名	インタフェース型	Provider/ Consumer	備考
RelativeMapService	MRFC:: RelativeMapService	Consumer	環境情報の取得．局所地図生成・更新 RTC が提供するサービスを呼び出す.

4.5.3 各ポートのデータ型・インタフェースについて

- robot_pose（IIS::TimedPose2D）
- robot_velocity（IIS::TimedVelocity2D）
- subgoal_waypoint（IIS::PoseVel2DSeq）
- robot_control（IIS::TimedVelocity2D）
- planning_img（TUT::TimedImageData）

これらデータ型の詳細は， 経路計画コンポーネント PathPlannerV2（http://www.openrtm.org/openrtm/ja/project/NEDO_Intelligent_PRJ_ID298）を参照.

- in_message（RTC::TimedString）

メンバ名	データ型	備考
tm	RTC::Time	未使用.
data	String	AGVController から本 RTC へのメッセージ. 往路開始時："ROBOT_START". 復路開始時："ROBOT_GO_BACK".

- message（RTC::TimedString）

メンバ名	データ型	備考
tm	RTC::Time	未使用.
data	String	本 RTC から AGVController へのメッセージ. 往路完了時：ROBOT_ARRIVED. 復路完了時：ROBOT_FINISH.

4.5.4 コンフィギュレーションパラメータ

パラメータ名	データ型	備考
GOAL_AREA	double	ゴール到達判定距離[m].
LOOP_TIME	double	再計画周期[s].
ROBOT_ACCELERATION	double	AGV の加速度[m/s²]
ROBOT_MAX_SPEED	double	AGV の最大速度[m/s]
SETPOINT_1	double	壁までの距離 S_1（図 4.5-2）
SETPOINT_2	double	壁までの距離 S_2（図 4.5-2）
CHECKPOINT_1	double	壁までの距離 C_1（図 4.5-2）
CHECKPOINT_2	double	壁までの距離 C_2（図 4.5-2）
CHECKPOINT_3	double	壁までの距離 C_3（図 4.5-2）
CHECKPOINT_4	double	壁までの距離 C_4（図 4.5-2）

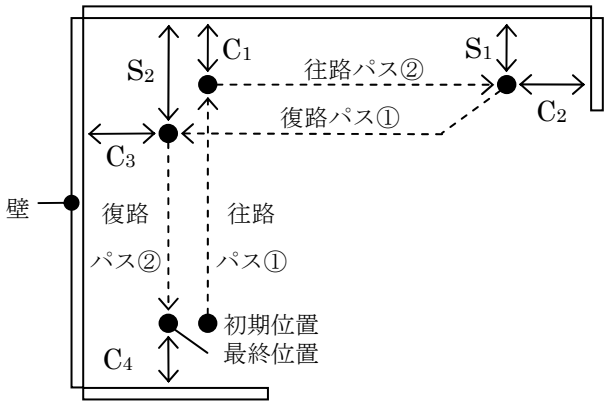


図 4.5-2 PathPlannerV2 のコンフィギュレーションパラメータ

4.5.5 ビルド方法

現在，準備中

4.6 HiroGate

本コンポーネントは, Windows から双腕ロボット HIRO を動作させるために, HIRO の QNX 上で動作している RT コンポーネントのサービスを直接呼び出すためのものである．加えて OpenRAVE（http://openrave.org/en/main/index.html）の逆運動学を解くライブラリである IKFast を利用して，アームが目標位置姿勢に到達可能か否かを判定するサービスも提供する．

本コンポーネント群では，この IKFast のサービスのみを利用しており，HIRO の動作自体には，前述の通り，共通インタフェースである HiroNXInterface を利用している．

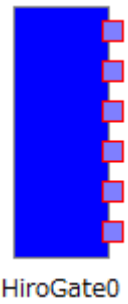


図 4.6-1 HiroGate

4.6.1 データポート

なし．

4.6.2 サービスポート

ポート名	インタフェース型	Provider/ Consumer	備考
RobotHWSvc	RobotHardwareService	Consumer	HIRO の QNX 上で動作する RTC へ接続（本コンポーネント群では未使用）.
SeqPlayerSvc	SequencePlayerService	Consumer	同上.
ArmCtrlSvcL	ArmControlService	Consumer	同上.
ArmCtrlSvcR	ArmControlService	Consumer	同上.
GrasperSvc	GrasperService	Consumer	同上.
HiroGateService	HiroGateRTC:: HiroGateService	Provider	上記各ポートを通して HIRO の QNX 上で動作する RTC の機能を呼び出し，関節角指定や位置姿勢指定によって HIRO の動作させるサービスを提供する.

4.6.3 各ポートのデータ型・インタフェースについて

ここでは，本コンポーネントの利用されるサービスのみを説明する．この他の HIRO の動作に関するサービスについては，HiroGateRTC.idl を参照.

サービス名	引数/返戻値	備考
boolean isIKSolutionFoundL	in RTC::Pose3D poseR_l_m_rad	ロボット座標系 R での左腕の目標位置姿勢. 位置は[m]単位, 姿勢は[radian]でオイラー角表現（orientation.y, p, r を利用し，y→p→r の回転順序で姿勢を表現する）.
	in double torso_angle_rad	胸部関節の角度[radian]. 上記の poseR_l_m_rad は，この胸部関節角度の姿勢から左腕が届くか否かが評価される.
	返戻値	解が見つかった場合：true 解が見つからなかった場合，：false
	IKFast を利用して，左腕の逆運動学の解があるかを調べる.	
boolean isIKSolutionFoundR	in RTC::Pose3D poseR_r_m_rad	ロボット座標系 R での右腕の目標位置姿勢. 位置は[m]単位, 姿勢は[radian]でオイラー角表現（orientation.y, p, r を利用し，y→p→r の回転順序で姿勢を表現する）.
	in double torso_angle_rad	胸部関節の角度[radian]. 上記の poseR_r_m_rad は，この胸部関節角度の姿勢から右腕が届くか否かが評価される.
	返戻値	解が見つかった場合：true 解が見つからなかった場合，：false
	IKFast を利用して，右腕の逆運動学の解があるかを調べる.	

4.6.4 コンフィギュレーションパラメータ

なし.

4.6.5 ビルド方法

現在，準備中.

5　使用方法

5.1　準備

5.1.1 使用 PC について

ここでは，本コンポーネント群を動作させる PC として，

1　StateMachine，AGVController，ObjectPoseEstimationGate 及び，物体位置姿勢推定コンポーネント群，HiroPickerAndPlacer，HiroGate を動作させる Windows PC（親機，ホスト名：localhost）．物体位置姿勢推定のセンサとして用いる SwissRanger（スイス MESA 社）を接続する．

2　PathPlannerV2 をはじめ, AGV を動作させるためのコンポーネント群を動作させる Windows PC（ホスト名：agv）．レーザレンジファインダ（LRF）TopURG(北陽電機株式会社)と Patrafour（関東自動車工業株式会社）を接続する．

3　HiroNX 共通インタフェースである HiroNXInterface のコンポーネント群を動作させる Ubuntu PC（HIRO に付属の PC，ホスト名：vision）

の 3 台を想定している．

localhost 機と agv 機は無線 LAN で，localhost 機と vision 機は有線 LAN（あるいは無線 LAN）で通信可能である必要がある．

5.1.2 実行時に必要なコンポーネントとライブラリについて

上記の各 PC について，本コンポーネント群から利用される RTC と外部ライブラリの準備について説明する．

- localhost 機

(1) ダウンロードした本コンポーネント群の圧縮ファイルを展開し，そのディレクトリ内の StateMachine，AGVController，ObjectPoseEstimationGate，HiroPickerAndPlacer，HiroGate を適当なディレクトリに置く．

(2) 外部のコンポーネント群として，以下をダウンロード・インストールする．

・物体位置姿勢推定コンポーネント

http://www.openrtm.org/openrtm/ja/project/NEDO_Intelligent_PRJ_ID383

(3) 外部ライブラリとして，以下をダウンロード・インストールする．

・OpenCV2.1

http://sourceforge.net/projects/opencvlibrary/files/

OpenCV-2.1.0-win32-vs2008 をダウンロードし，インストールする．

環境変数の Path にインストールディレクトリ内の bin（cv210.dll, cxcore210.dll 等が存在するディレクトリ）を加える（例：C:¥OpenCV2.1¥bin）．

- agv 機

(1) ダウンロードした本コンポーネント群の圧縮ファイルを展開し，そのディレクトリ内の PathPlannerV2 を適当なディレクトリに置く．

(2) 外部のコンポーネント群として，以下をダウンロード・インストールする．

・「局所地図生成・更新 RTC」

http://www.openrtm.org/openrtm/ja/project/NEDO_Intelligent_PRJ_ID296

・「関東自動車工業株式会社「Patrafour」（パトラフォー）用制御コンポーネント」

http://www.openrtm.org/openrtm/ja/project/NEDO_Intelligent_PRJ_ID380

・「URGDataFlowComp」（LRF データ取得コンポーネント）

（株式会社セック様へお問い合わせください．）

以上を付属のマニュアルに従ってインストールし，動作することを確認する．

(3) 外部ライブラリとして，以下をダウンロード・インストールする．

・OpenCV2.1

http://sourceforge.net/projects/opencvlibrary/files/

OpenCV-2.1.0-win32-vs2008 をダウンロードし，インストールする．

環境変数の Path にインストールディレクトリ内の bin（cv210.dll，cxcore210.dll 等が存在するディレクトリ）を加える（例：C:¥OpenCV2.1¥bin）．

- vision 機

(1) 外部のコンポーネント群として，以下をダウンロード・インストールする．

・HiroNXInterface

http://www.openrtm.org/openrtm/ja/project/NEDO_Intelligent_PRJ_HiroAccPrj_1003

付属のマニュアルに従ってインストールし，動作することを確認する．

5.1.3 HiroPickerAndPlacer のコンフィギュレーションパラメータ

4.4.4 で示したように，HiroPickerAndPlacer には多くのパラメータがあり，適切に設定する必要がある．ここでは，既定値のまま使用できない，特に重要なパラメータの設定について説明する．

- 物体位置姿勢推定コンポーネントと HIRO の世界座標系キャリブレーション

HIRO は物体位置姿勢推定コンポーネントの推定結果を受けてピック&プレース動作を行うので，双方で同じ座標系を基準にする必要がある．

ここでは，物体位置姿勢推定コンポーネントで利用するキャリブレーションパターンの原点（=世界座標系の原点）と，HIRO のロボット座標系の原点の相対位置を測定して利用する．図 5.1-1 に測定するパラメータを示す．

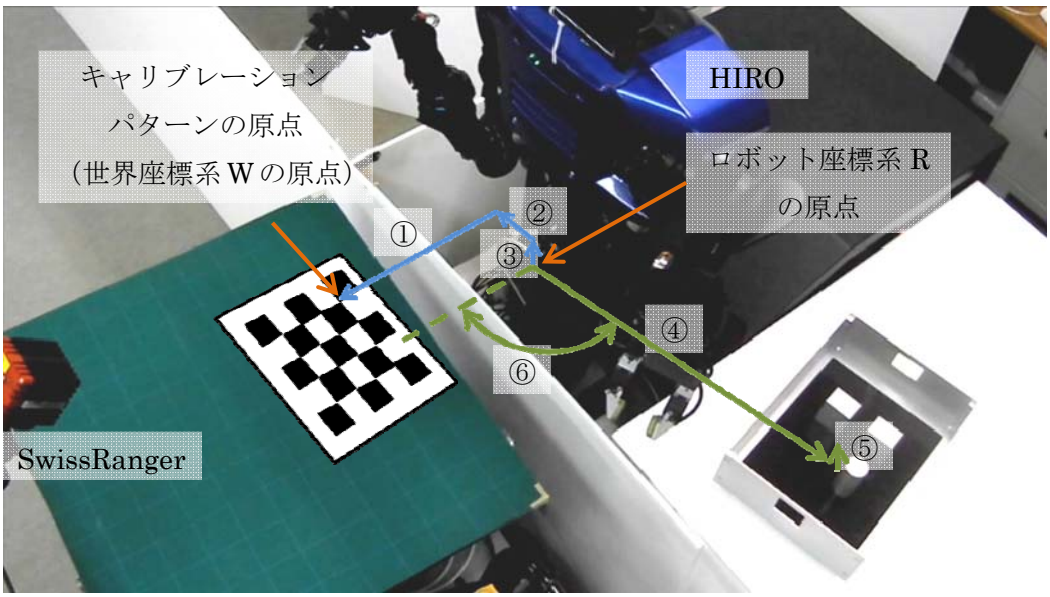


図 5.1-1 物体位置姿勢推定コンポーネントと HIRO の世界座標系のキャリブレーション

この図において，青色の矢印（①～③）を測定し用いる．値の正負はロボット座標系の軸の方向に従う．この矢印とコンフィギュレーションパラメータの対応は以下の通りである．

① ⇔ xWorldR_m
② ⇔ yWorldR_m
③ ⇔ zWorldR_m

- パレットが置かれた位置と置く位置のパラメータ

パレットが置かれた位置に関しては，図 5.1-1 の緑色の矢印（④～⑤）のパラメータを設定する必要がある．ここでは，胴体がパレットの方向を向いている時に，胴体の

正面にパレットがある（中央に置かれている）と仮定している．また，④については，胴体と共に z 軸回転する胴体座標系 B での値を，⑥については，ロボット座標系 R での胴体座標系 B の回転を表している．この矢印とコンフィギュレーションパラメータの対応は以下の通りである．

④ ⇔ xPalletPlacedB_m

yPalletPlacedB_m = 0

⑤ ⇔ zPalletPlacedB_m

⑥ ⇔ directionPalletPlacedR_deg

パレットを置く位置に関しては，ロボット座標正面の AGV の上にパレットを置くため，

directionPalletToBePlacedR_deg = 0

xPalletToBePlacedB_m = ④

yPalletToBePlacedB_m = 0

zPalletToBePlacedB_m = ③

と設定する．

この他のパラメータについては，4.4.4 の説明を参照し，適切に設定する．

5.2　コンポーネントの起動

vision，agv，localhost の順番に起動する．

- vision 機のコンポーネントの起動

(1) HiroNXInterface の HiroNXProvider，及び，HiroNXGUI を付属マニュアルに従って起動し，ポートを接続後(図 5.2-1)，アクティベートしておく．
また，HIRO の QNX も起動しておく．

(2) HiroNXGUI の，"Set up Robot"ボタンと"Calibrate Joints"ボタンを押し，加えて，Servo Hands の"ON"ボタンを押して HIRO を準備する．

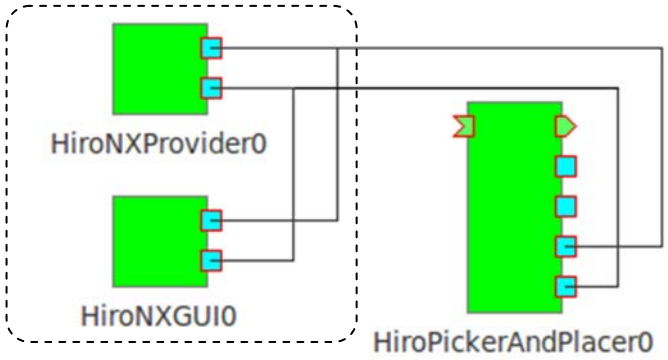


図 5.2-1 vision 機の RTC 構成・接続図（点線内）

- agv 機のコンポーネントの起動

(1) 以下の各コンポーネントを各インストールディレクトリから起動する．

・局所地図生成・更新 RTC：LocalMap
　パス：　LocalMapRTC¥LocalMapComp.exe
・Patrafour 用制御コンポーネント：PatrafourControllerModule
　パス：　PatrafourContorollerModule¥
　　PatrafourContorollerModuleComp.exe
・LRF データ取得コンポーネント：URGDataFlowComp
　パス：　Top-URG_VCxx_v1.1_exe¥rtm1.0.0¥
　　URGDataFlowCompComp.exe
　　起動に先立って電源を入れておく必要がある．
・経路計画：PathPlannerV2
　パス：　PathPlannerV2AGV¥components¥PathPlannerV2Comp.exe

(2) RTSystemEditor 等で図 5.2-2 のようにポートを接続する．

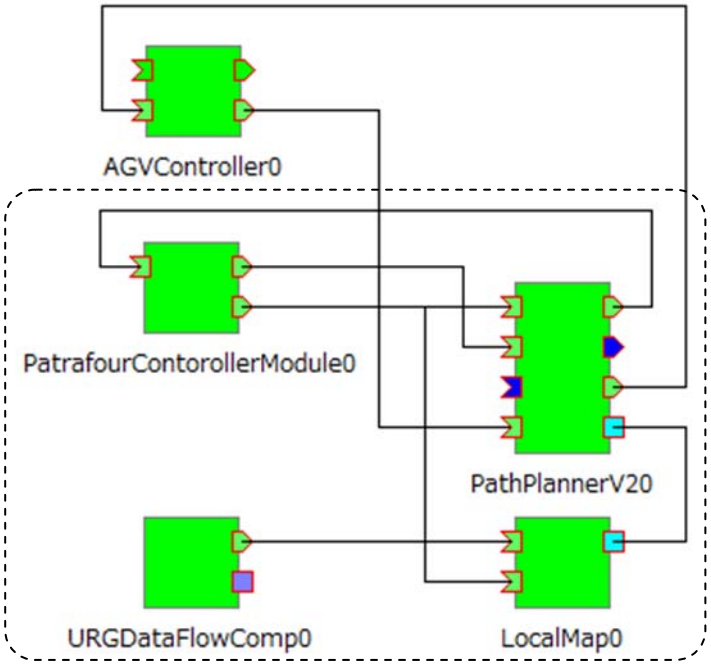


図 5.2-2 agv 機の RTC 構成･接続図（点線内）

(3) コンフィギュレーションパラメータを設定する.

・PatrafourControllerModule

Foward_Direction を BACK に設定（後ろ向きに進む）.

・PathPlannerV2

4.5.4 のパラメータを環境に合わせて，適切に設定する.

(4) 各コンポーネントをアクティベートする.

最初に URGDataFlowComp をアクティベートしておく.

次に PatrafourControllerModule をアクティベートする．この方法については，付属のマニュアルを参照.

その後，LocalMap，PathPlannerV2 をアクティベートする.

- localhost 機

(1) 以下の各コンポーネントを各インストールディレクトリから起動する.

・状態機械経路計画：StateMachine

パス： StateMachine¥components¥StateMachineComp.exe

・AGV 制御：AGVController

パス： AGVController¥components¥AGVControllerComp.exe

・物体位置姿勢推定制御：ObjectPoseEstimationGate

パス： ObjectPoseEstimationGateCIF¥components¥
ObjectPoseEstimationGateComp.exe

・ピック&プレース行動計画・実行：HiroPickerAndPlacer

パス： HiroPickerAndPlacerCIF¥components¥
HiroPickerAndPlacerComp.exe

・HIRO 接続コンポーネント：HiroGate

パス： HiroGate¥components¥HiroGateComp.exe

以下のコンポーネントは，物体位置姿勢推定コンポーネントに付属のマニュアルに従って起動する．

・SwissRanger データ取得コンポーネント：SwissRanger4K
・物体モデル定義コンポーネント：ModelDefinition
・物体位置姿勢推定コンポーネント：ObjectPoseEstimation
・物体位置姿勢推定結果表示コンポーネント：EstimationResultViewer

(2) RTSystemEditor 等で図 5.2-3 のようにポートを接続する．

(3) コンフィギュレーションパラメータを設定する．

・StateMachine

必要ならば tcp_port を変更する．

また，scenario_path には，適切な状態機械を記述した XML ファイルを指定する．既定の scenario.xml は，StateMachineComp.exe と同じディレクトリにある，HIRO－AGV 連携の状態機械を記述したファイルである．図 5.2-4 に，このファイルで定義された状態遷移図を示す．また，同梱 pnp.xml は，AGV が HIRO の前に居る状態から移動させずに，物体の位置姿勢推定とピック&プレース動作のみを行わせる状態機械 である．

・HiroPickerAndPlacer

0 のパラメータを設定する．

(4) 各コンポーネントをアクティベートする．

最初に，HiroGate を，物体位置姿勢推定コンポーネントに付属のマニュアルに従ってアクティベートし，その後，StateMachine，AGVContorller，ObjectPoseEstimationGate， HiroPickerAndPlacer の順にアクティベートする．

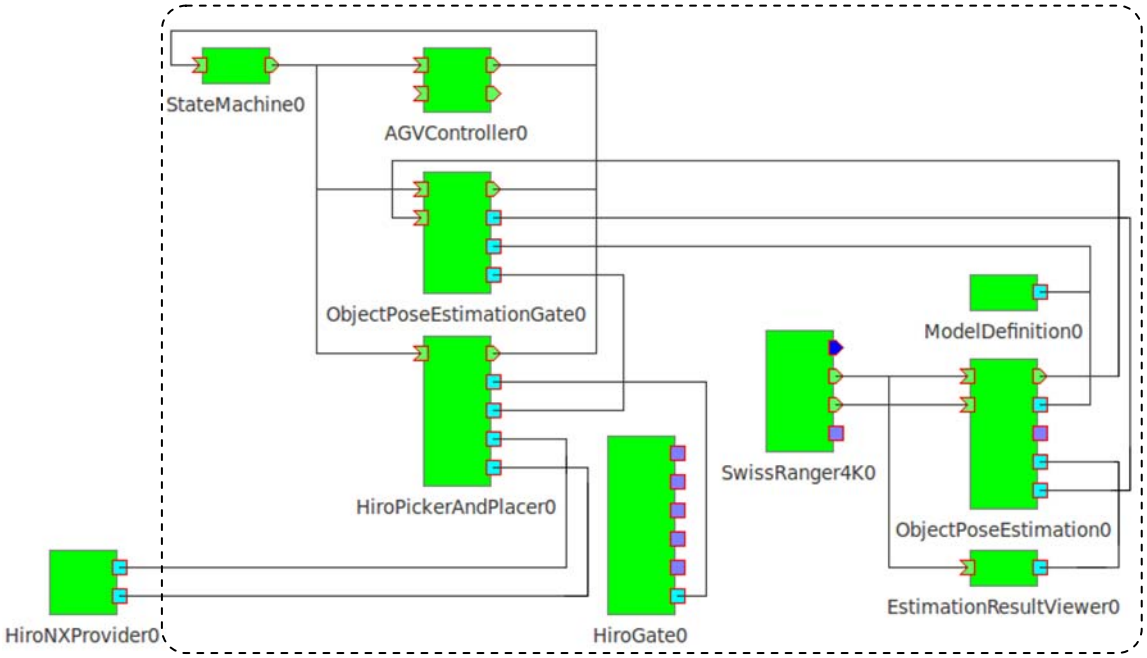


図 5.2-3 localhost 機の RTC 構成・接続図（点線内）

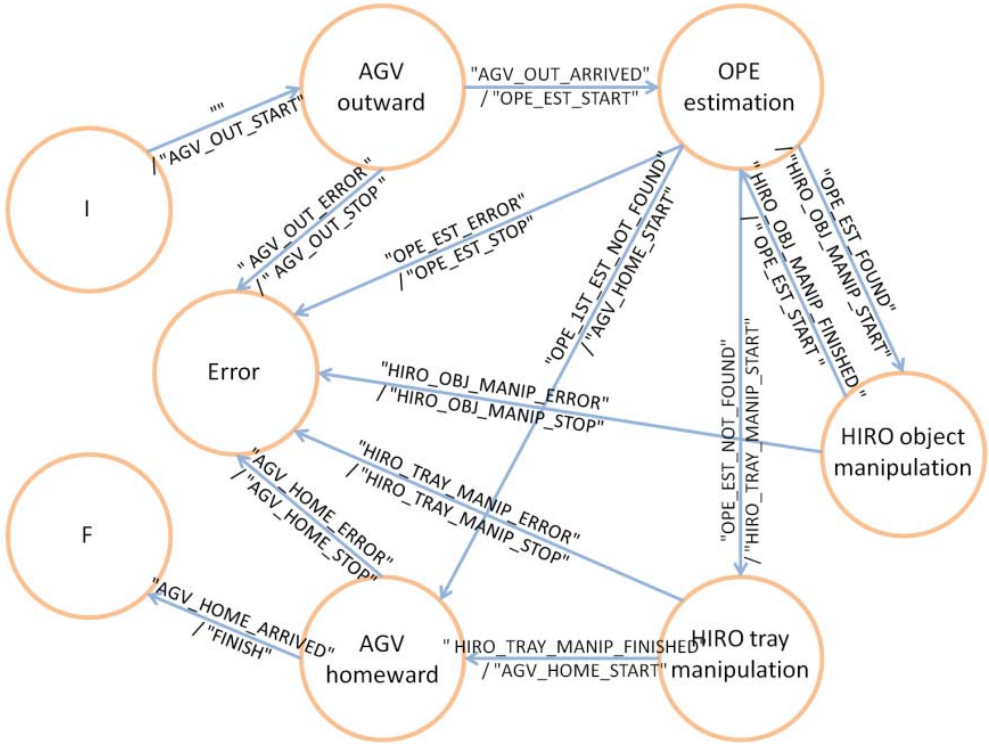


図 5.2-4 scenario.xml で定義された状態遷移図

5.3　コンポーネントの動作

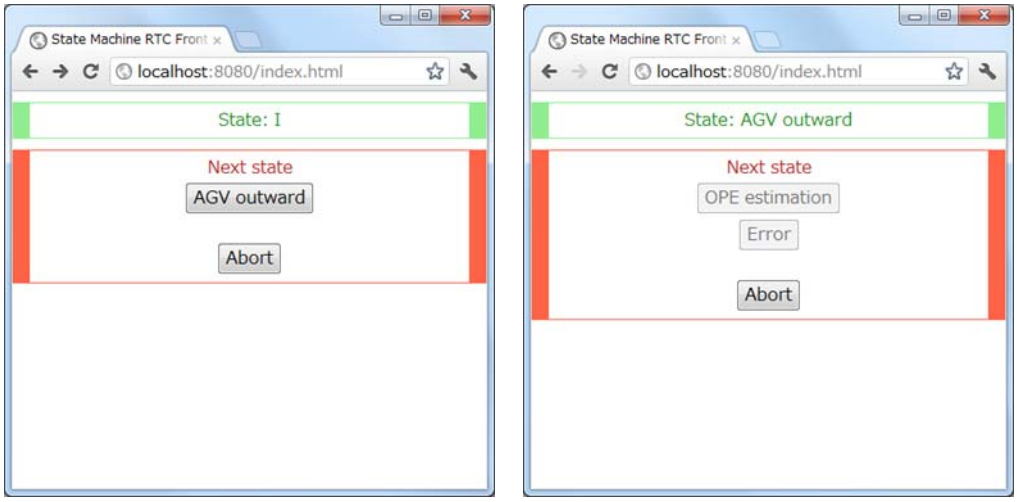
全コンポーネントが起動すると，ユーザは，StateMachine に組み込まれた HTTP サーバから配信される状態確認用の Web ページから，コンポーネントの動作の確認ができるようになる．

localhost 機からであれば，Google Chrome などの Web ブラウザのアドレスバーに http://localhost:8080/と入力すると，図 5.3.-1 のような，現状態と遷移可能な状態名が記されたボタンから構成される Web ページが表示される．

ボタンは，記された状態への遷移のタイミング指定が手動で，かつ，遷移可能になった時点で有効化される．例えば，下図の(b)のように，AGV が往路移動中には物体位置姿勢推定を行ってはならないので，初期表示ではボタンは無効化されている．AGV が往路の走行を完了し，AGVController から StateMachine へ入力（=状態遷移のトリガ）がなされてから，遷移ボタンが有効化される．そして，有効化されたボタンを押すと，状態の遷移と遷移時の出力が生じる．なお，Abort ボタンは常に有効であり，各状態に対して定義されたエラー状態へ強制的に遷移することができる．

同梱している状態機械の XML ファイルでは，遷移タイミングを手動としている（ただし，エラー状態への遷移は自動）．

図 5.3-2 に，本コンポーネント群を用いた HIRO と AGV の連携実験の様子を示す．



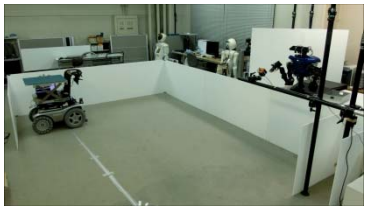
(a) 初期状態　　(b) AGV 往路走行状態

I　　AGV outward

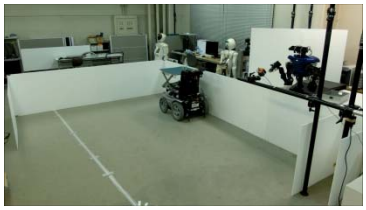
図 5.3.-1 状態確認 Web ページ

6　連絡先について

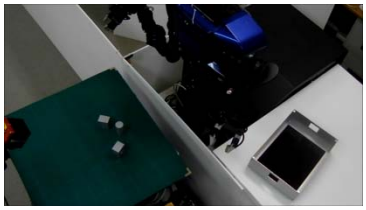
不明な点がある場合は rtc@aisl.cs.tut.ac.jp まで連絡をお願いします．



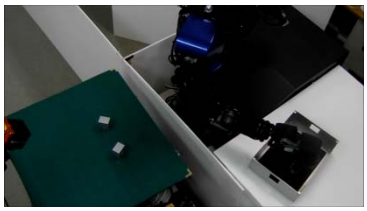
(a) 0[s],
I



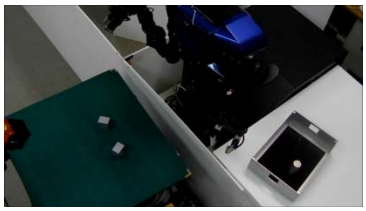
(b) 17[s],
AGV outward



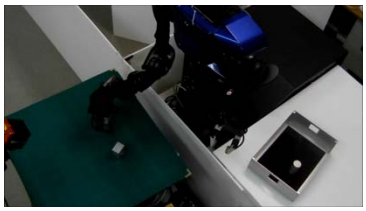
(c) 32[s],
OPE estimation



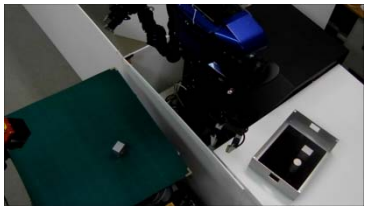
(d) 42[s],
HIRO object manipulation



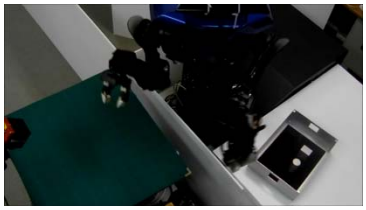
(e) 46[s],
OPE estimation



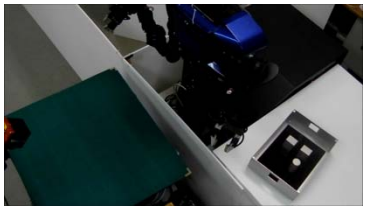
(f) 51[s],
HIRO object manipulation



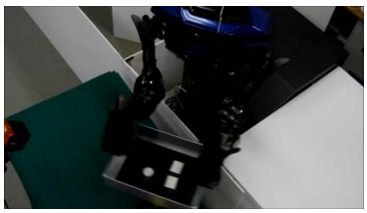
(g) 59[s],
OPE estimation



(h) 65[s],
HIRO object manipulation



(i)72[s],
OPE estimation



(j) 81[s],
HIRO tray manipulation

(k) 106[s],
AGV homeward

(l) 135[s],
F

図 5.3-2 HIRO－AGV 連携実験の様子