

次世代ロボット知能化技術開発プロジェクト
移動知能(社会・生活分野)の研究開発

モジュール仕様書

人追従機能モジュール群

セグウェイジャパン株式会社

2011/5/12

改版履歷

[illegible]

人追従機能モジュール群 システム・動作 仕様

システム概要

対象人物の色ヒストグラム情報を記憶して、その人を追従する移動体向けの速度指示値を生成する。

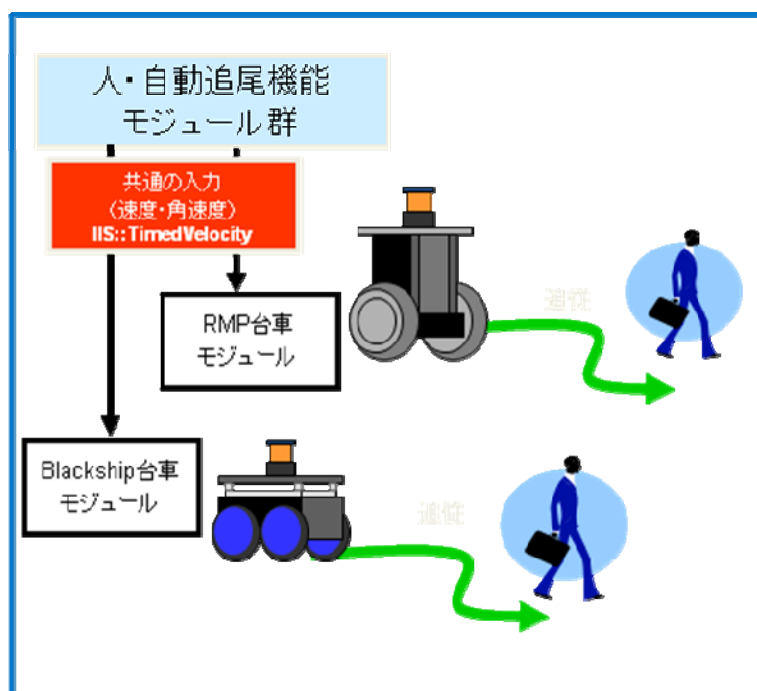


図 機能概要図



図 システム動作概要図

システム構成

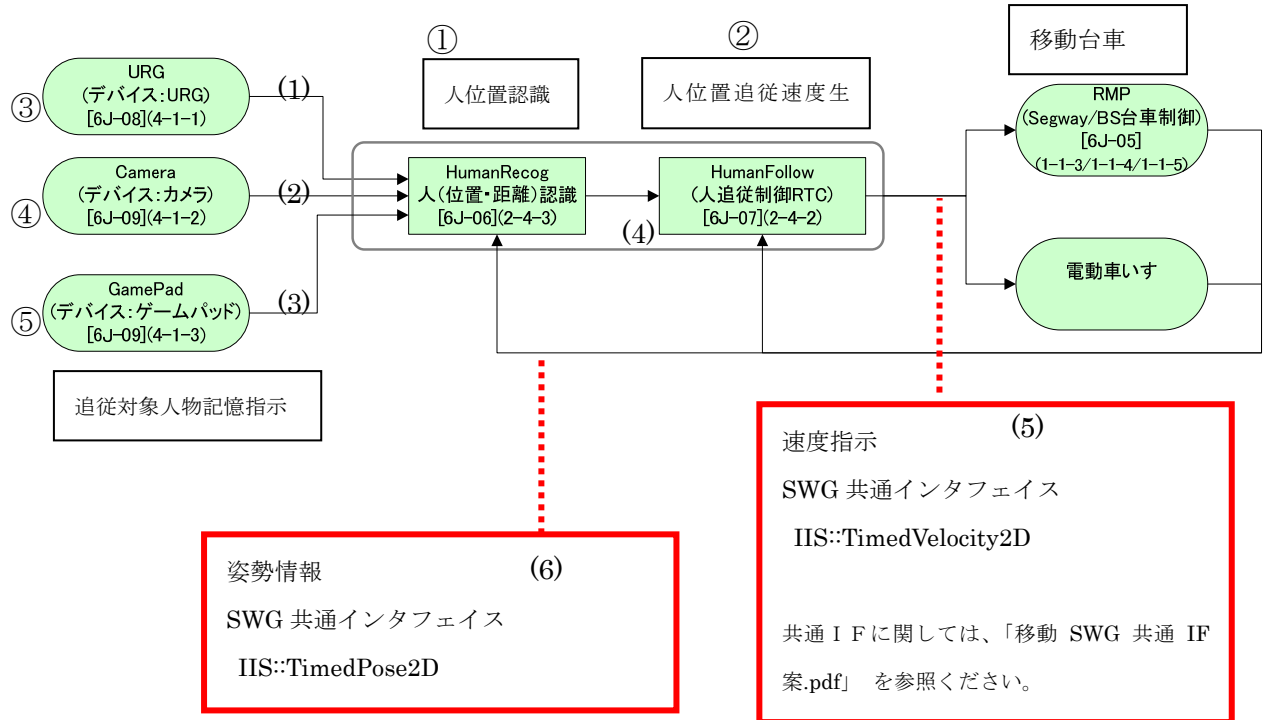


図 システム構成 RTC 接続図

表 人追従機能 モジュール一覧

NO	名称	種別	備考
①	人(位置・距離)認識 RTC (HumanRecog)	RTC	カメラ・レーザーレンジから人位置を認識
②	人追従制御 RTC (HumanFollow)	RTC	人位置に対して追従する速度成分を生成
③	レーザーレンジ RTC (URG)	RTC	デバイス IF モジュール
④	カメラ RTC (Camera)	RTC	デバイス IF モジュール
⑤	ゲームパッド RTC (GamePad)	RTC	デバイス IF モジュール
⑥	台車 RTC (RMP/ 電動車いす) 基本台車モジュールの仕様書を参照ください	RTC	台車駆動モジュール。SWG 共通 IF IIS::TimedVelocity2D を入力とする台車であれば置き換え可能

表 コネクタ情報

NO	データタイプ	説明
(1)	TimedLongSeq	測域センサの距離データ
(2)	TimedUShortSeq	YUYV 形式の画像データ
(3)	TimedULong	ゲームパッドのボタン入力。 対象人物の登録時に使用
(4)	IIS::TimedPose2D	人位置情報
(5)	IIS::TimedVelocity2D	速度指示。 v_x, w のみ有効。 SWG 共通 IF です
(6)	IIS::TimedPose2D	姿勢情報。

① 人（位置・距離）認識 RTC (HumanRecog)

概要

最初に登録した人の位置を認識してその位置を出力します。

動作説明

カメラによる対象人物の色ヒストグラム成分とのマッチングにより、対象人物の方向を認識します。その認識された方向に対して、レーザーレンジより距離を算出し、対象人物の位置を特定してその位置を出力します。



Input への入力がある間、ターゲットとなる人の色情報を更新します。入力が 0 になると確定となり、その時点での色情報が人の位置認識に使用されます。色成分はカメラ画像の右側に表示されます。認識できない場合、位置はロボット位置と同じ座標を出力します。

基本情報

種別	RTC
動作 OS	Linux(Ubuntu10.04) / Windows
RT ミドルウェア	OpenRTM-aist-1.0.0-RELEASE
開発言語	C++
依存ライブラリ	なし

ポート情報

DataIn Port

ポート名	型	説明
Input	TimedULong	対象人物の登録（1 入力時）

ImangeYUYV	TimedUShortSeq	YUYV 形式画像 raw データ
Scan	TimedLongSeq	レーザーレンジ距離
RobotPose2DIIS	IIS::TimedPose2D	ロボットの位置

DataOut Port

ポート名	型	説明
HumanPose2DIIS	IIS::TimedPose2D	人の位置

Configuration 情報

名前	型	説明
imagew / imageh	int	画像サイズ (ピクセル)
grayS	double	グレースケール判定の敷居値(割合 0~1)
grayK	double	グレースケール要素の係数
matchvalue	double	マッチング判定の敷居値
matchblur	double	マッチング対象をぼかす広さ
matchw / matchh	int	色情報範囲 (ピクセル)
search / search	int	マッチング範囲 (ピクセル)
addx / addy	int	マッチング確認間隔 (ピクセル)
maxjump	double	ターゲット認識移動距離範囲
angleofview0	int	カメラ画像の画角
angleofscan	int	LRF 距離判定時の判定角度

※Windows 版ではウィンドウは表示されません。

② 人追従制御 RTC (HumanFollow)

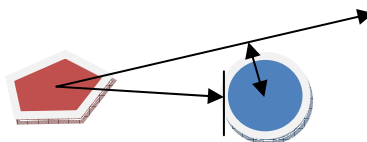
概要

人（位置・距離）認識 RTC によって認識された、人位置情報に対して追従する速度成分を生成する。

動作説明

(1) Follow モード ターゲットへの直線的な追従 (Mode:0)

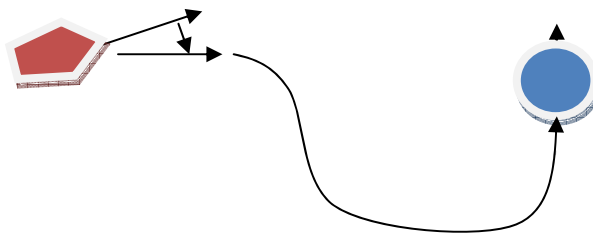
正面方向に対して横方向のズレの距離に比例して旋回成分を生成。また、対象までの指定間隔の距離との差に比例してその距離を詰める併進速度を生成します。



(2) Trace モード ターゲット位置をトレースする追従 (Mode:1)

ターゲット位置を順次記憶していき、前方方向の速度と、角速度によってその位置を辿るように追従します。

注意：このドキュメント中の人位置認識 RTC(HumanRecog)はカメラ正面画角範囲しかターゲットを捉えられないのでこのモードでは使用できません。別途人位置もしくはその他物体位置を入力する機能が必要になります。



基本情報

種別	RTC
動作 OS	Linux(Ubuntu10.04) / Windows
RT ミドルウェア	OpenRTM-aist-1.0.0-RELEASE

開発言語	C++
依存ライブラリ	なし

ポート情報

DataIn Port

ポート名	型	説明
HumanPose2DIIS	IIS::TimedPose2D	人の位置
RobotPose2DIIS	IIS::TimedPose2D	ロボットの位置

DataOut Port

ポート名	型	説明
RobotVel2DIIS	IIS::TimedVelocity2D	追従のための速度情報 vx, vw のみ有効

Configuration 情報

名前	型	説明
Kspeed	double	生成併進速度に乘じられる係数
Kturn	double	生成角速度に乘じられる係数
MAXspeed	double	Follow モード 最大速度(m/sec)
MAXturn	double	Follow モード 最大角速度(rad/sec)
Distance	double	Follow モード ターゲットとの距離(m)
PathVel	double	Trace モード 追従経路移動速度(m/sec) ※ PathTime が設定されているとき無効
PathTurn	double	Trace モード 追従経路移動角速度(m/sec)
PathTargetR	double	Trace モード 経路情報からの許容距離(m)
PathStopR	double	Trace モード ゴール位置からの許容距離(m)
PathTime	double	Trace モード 現在の経路の長さを移動する時間(sec)
IntervalTime	double	Trace モード ターゲット位置更新時間間隔 (sec)
imagew / imageh	int	Trace モード 経路表示画面のサイズ
Mode	int	動作モード 0: Follow モード 1: Trace モード

③ レーザーレンジ RTC (URG)

概要

HOKUYO UTM-30LX 向けのデバイスモジュール

動作説明

mm 単位の距離値をスキャン数(1081)出力します。

デバイス内部タイマー値から、PC 時間タイムスタンプを見積もる機能付き。

基本情報

種別	RTC
動作 OS	Linux(Ubuntu10.04)
RT ミドルウェア	OpenRTM-aist-1.0.0-RELEASE
開発言語	C++
依存ライブラリ	なし

ポート情報

DataOut Port

ポート名	型	説明
Scan	TimedLongSeq	タイムスタンプ付距離データ(mm) tm : PC 時間同期タイムスタンプ
ScanTS	TimedScan	(Scan と同じ内容のため将来削除予定) タイムスタンプ付き距離データ(mm)

Configuration 情報

名前	型	説明
str_port	string	デバイスポート名(/dev/ttyACM0)
start	Int	スキャン開始位置
end	Int	スキャン終了位置
skip	Int	スキップ数
win	int	スキャンデータの Windows 表示
timesynccount	Int	デバイス・PC 時間同期回数
timesyncperiod	Int	デバイス・PC 時間同期間隔



IDL 情報

```
struct TimedScan {  
    Time tm;  
    TimedLongSeq scan;  
};
```

④ カメラ RTC

概要

カメラ向けのデバイスモジュール

動作説明

UVC (USB Video Class)に対応したカメラ向けデバイスモジュール。カメラ画像を raw データにて出力します。Video4Linux2 を使用しています。

(Logicool QCAM-200V で動作実績あり)

基本情報

種別	RTC
動作 OS	Linux(Ubuntu10.04)
RT ミドルウェア	OpenRTM-aist-1.0.0-RELEASE
開発言語	C++
依存ライブラリ	なし

ポート情報

DataOut Port

ポート名	型	説明
Image	TimedULongSeq	24bitRGB 値の画像(4byte/pixel)
ImageYUYV	TimedUShortSeq	YUYV 形式の画像 (2byte/pixel)

Configuration 情報

名前	型	説明
str_port	string	デバイスポート名(/dev/video0)
width	Int	画像幅
height	Int	画像高



⑤ ゲームパッド RTC

概要

PC 用ゲームパッド向けのデバイスモジュール

動作説明

PC 用ゲームパッド向けデバイスモジュール。ゲームパッドの入力値を出力します。

基本情報

種別	RTC
動作 OS	Linux(Ubuntu10.04) / Windows(VC2008)
RT ミドルウェア	OpenRTM-aist-1.0.0-RELEASE
開発言語	C++
依存ライブラリ	なし

ポート情報

DataOut Port

ポート名	型	説明
Button	TimedULong	ボタン状態 (各 bit)
StickLX	TimedFloat	左アナログスティックの x 方向値
StickLY	TimedFloat	左アナログスティックの y 方向値
StickRX	TimedFloat	右アナログスティックの x 方向値
StickRY	TimedFloat	右アナログスティックの y 方向値
StickLXd	TimedDouble	左アナログスティックの x 方向値
StickLYd	TimedDouble	左アナログスティックの y 方向値
StickRXd	TimedDouble	右アナログスティックの x 方向値
StickRYd	TimedDouble	右アナログスティックの y 方向値
Velocity2DIIS	IIS::TimedVelocity2D	左アナログスティックからの速度生成



追従機能モジュール群 使用手順

環境準備

- 1) OpenRTM-aist-1.0.0-RELEASE (C++版 UNIX/Windows) のインストール

【マニュアルの場所】 OpenRTM-aist Official Web Site ⇒ ドキュメント ⇒ インストール

<http://www.openrtm.org/openrtm/ja/node/999>

【インストール方法】 マニュアルに従い、インストールを行う。

- 2) RTSystemEditor のインストール

【マニュアルの場所】 OpenRTM-aist Official Web Site ⇒ ドキュメント ⇒ ツール ⇒ RT System Editor-1.0.0

<http://www.openrtm.org/openrtm/ja/content/rtsystemeditor-1>

【インストール方法】 マニュアルに従い、UNIX/Windows 用 RTSystemEditor のインストールを行う。

ハードウェア準備

各種ハードウェアの電源を入れ、準備可能にする。この際に、Linux に認識されたデバイスがそれぞれ存在するか確認をすると不具合が少なくなる。たとえば /dev/ttyACM* /dev/input/js* /dev/ttyUSB* /dev/video* 等。

起動手順

- 1) OpenRTM-aist RT / System Editor の起動

【マニュアルの場所】 OpenRTM-aist Official Web Site ⇒ ドキュメント ⇒ ツール ⇒ RT System Editor-1.0.0

<http://www.openrtm.org/openrtm/ja/content/rtsystemeditor-1>

【起動方法】 マニュアルに従い、RT System Editor の起動を行う。

- 2) Comp ファイルの実行

【Comp ファイルの場所】 各モジュールの src ディレクトリの中

【マニュアルの場所】 各モジュールの doc ディレクトリの中 (manual.txt)

【実行方法】 各 Comp ファイルを実行の後、各マニュアルを参考にコンフィグレーションの設定を行う。

3) モジュールの接続

【接続方法】以下の図のようにモジュールを接続する。

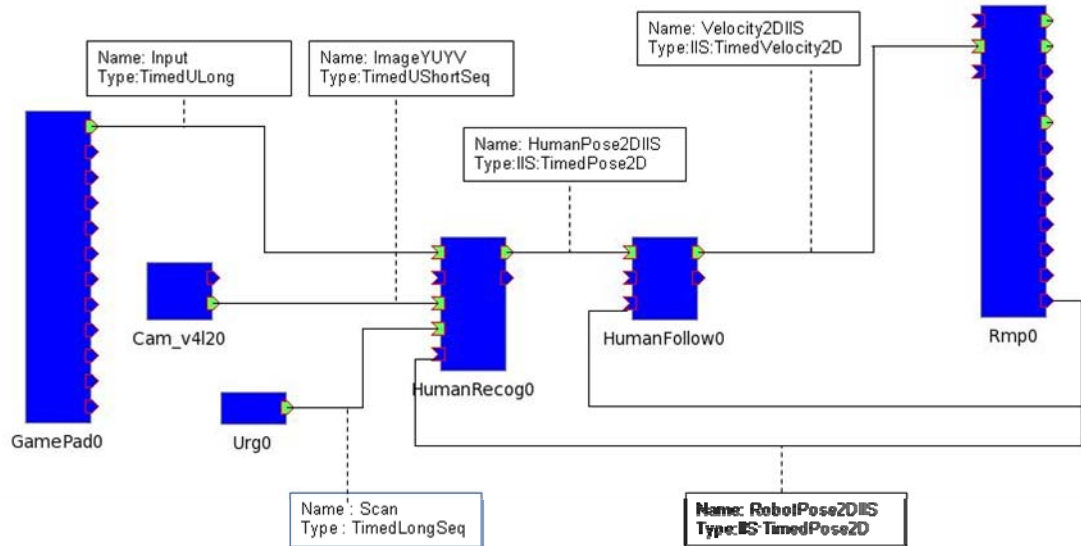


図 人追従機能モジュール群接続図

4) モジュールの起動

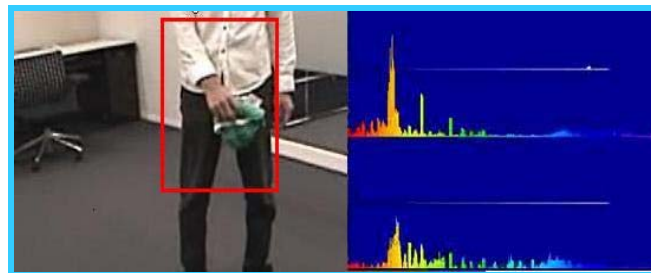
RT-SystemEditor にて、各モジュールを Activate します。

使用方法

1) ターゲットの記憶

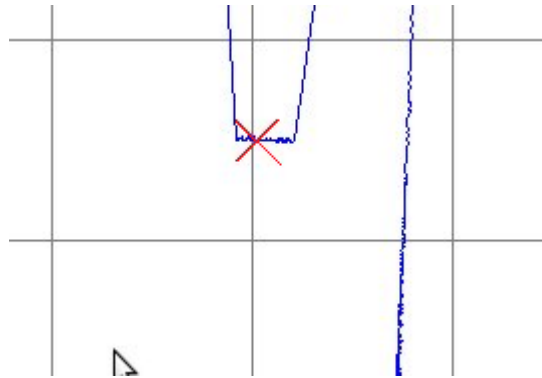
ターゲットとなる人が **HumanRecog** に表示されている画面真ん中に映るように立ちます。青色の四角の枠が表示されていますので、その枠をなるべく埋めるように位置します。ロボットと人の距離は、デフォルトで 1.5 メートルになります。

ゲームパッドのボタンを押すと、ターゲットの色情報を更新し、ボタンを離した時点の色情報が追従するターゲットの情報として確定されます。その際の色分布がウィンドウの右側に表示されます。(※ウィンドウは linux 版のみ表示されます。)



2) ターゲットへの追従

ゲームパッドのボタンを離すとロボットが追従を開始します。青色の枠が赤色に変わり、ターゲットの色情報として認識された位置に表示されます。そのターゲットの方向と、レーザーレンジセンサの距離情報からターゲットの距離が認識され、その位置が下の画面にレーザーレンジセンサの結果と共に赤の×印で表示されます。



ターゲットが移動をすると、それに追従するようにロボットが移動します。ターゲットを見失うとターゲット位置情報はロボット位置と同じ位置の情報が送られ、追従動作はストップします。

操作・使用方法

・各デバイスのロボットへのセッティング



次ページ以降のモジュール接続図のようにRTCを用意し、接続します。
各デバイス、各RTCの環境設定に関する情報は
各RTCのファイル内にあるmanual.txtやpdを参照してください。

