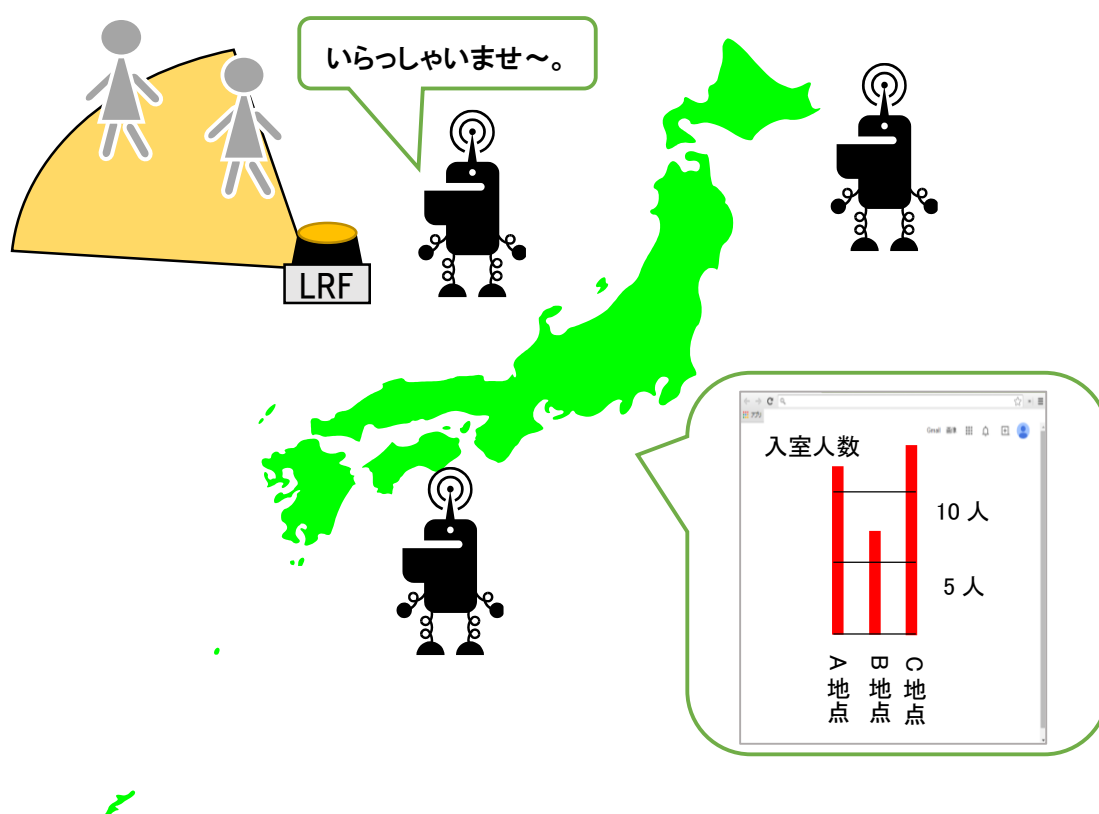


# RTM-RSNP による人数管理システム ユーザーマニュアル



芝浦工業大学 工学部 知能機械システム研究室

野見山 大基, 石田 真一, 生田目 祥吾, 松日楽 信人

# 目次

1. RTM-RSNP による人数管理システムの概要.....	3
1.1 はじめに .....	3
1.2 開発環境・動作確認.....	3
2. 複数人追跡 RTC.....	4
2.1 複数人追跡方法について(重要).....	5
2.1.1 物体検知方法 .....	5
2.1.2 人認識条件.....	5
2.1.3 人追跡条件.....	5
2.1.4 人削除条件.....	5
2.2 人数カウント機能について.....	7
2.2.1 人数カウント方法.....	7
2.2.2 入退室方向の入れ替え .....	7
2.3 OutPort について.....	8
2.3.1 human_dataX .....	8
2.3.2 human_dataY.....	8
2.3.3 human_dataV .....	8
2.3.4 human_dataT.....	8
2.3.5 count_data .....	8
3. 人数管理 RSNP RTC.....	10
3.1 セットアップ .....	11
3.2 人数管理 RSNP RTC コンフィギュレーションについて .....	11
3.3 サーバの設定について.....	11
4. 使用ソフトウェア及び入手方法, 参考資料 .....	12

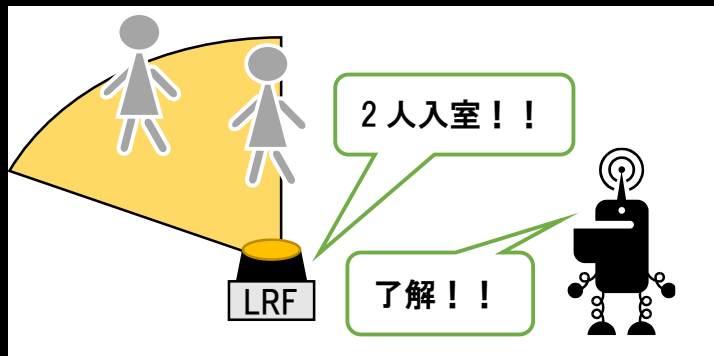
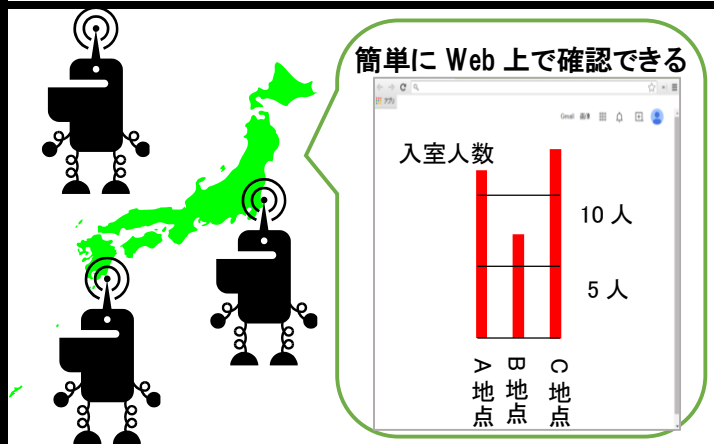
# 1. RTM-RSNP による人数管理システムの概要

## 1.1 はじめに

我々の研究室では、測域センサから取得した人間の位置情報を元に、ロボットが応答を行うシステムを構築している。このようなシステムはロボットの機能を拡張するものとしては基本的なものである。さらに、複数ロボット間でネットワークを構築することも機能拡張に繋がる。したがって、これらのシステムを RTC 化することで人の位置情報や歩行情報などを利用した様々なアプリケーションを容易に作成することが可能となる。

そこで今回、「複数人追跡 RTC」と「人数管理 RSNP RTC」の2つの RTC を開発した。この RTM-RSNP による人数管理システムは測域センサを用いることで簡単に複数人追跡、及び人数管理をインターネット上で行うことが出来るシステムである。RSNP(Robot Service Network Protocol)とは RSi(ロボットサービスイニチアティブ)という団体が作成した、ロボットを共通の通信プロトコルで動かすことができる仕組みである [1]。※このプロトコルを使用するためには、以下のサイトを参考に書面またはメールで申し込みが必要。

RSNP ライブラリ v2.3(株式会社富士通研究所)「作品使用ライブラリ」 [http://robotservices.org/rsi\\_spec.html](http://robotservices.org/rsi_spec.html)

人数管理システムの流れ	説明
	<p>LRF からの距離情報を元に周囲に居る複数の人を追跡する。その際、各人の歩行情報(位置, 速さ, 向き), 及び入退室数を算出する。</p> <p>機能を提供するのは…… 複数人追跡 RTC</p>
	<p>RSNPを利用し、人数カウント情報をサーバにアップロードする。取得したデータを元に、サーバ上で人数管理が可能になる。</p> <p>また、複数ロボットからのデータを比較しやすいように図にまとめる。</p> <p>機能を提供するのは…… 人数管理 RSNP RTC</p>

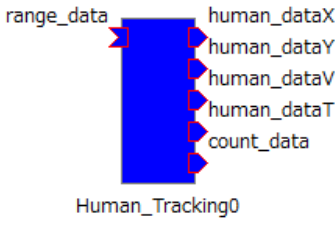
## 1.2 開発環境・動作確認

動作確認は OpenRTM-aist1.1.0 をインストールした PC(OS:64bit 版 Windows8)で行った。開発環境は同 PC 上の Visual C++ 2010 Professional を用いた。

## 2. 複数人追跡 RTC

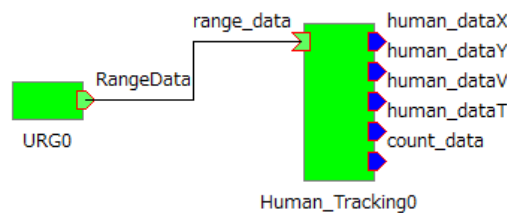
この RTC は InPort から取得した LRF の距離情報を元に、複数人を追跡し、人数カウント情報や各人の歩行情報(位置, 速さ, 向き)を提供することが出来る。また、コンフィギュレーションパラメーターを多く設定しているため、様々な場所に合わせて利用が可能である。

※今回は北陽電機株式会社製の URG-04LX を使用

RTC の名称		
Human_Tracking		
InPort		
名称	データ型	機能
range_data	RangeData	LRF から取得した距離データを入力
OutPort		
名称		機能
human_dataX	TimedLongSeq	各人の位置データ(X軸[mm])を出力
human_dataY	TimedLongSeq	各人の位置データ(Y軸[mm])を出力
human_dataV	TimedLongSeq	各人の歩行データ(速さ[mm/s])を出力
human_dataT	TimedLongSeq	各人の歩行データ(向き[deg])を出力
count_data	TimedLongSeq	入退室及び追跡中的人数データを出力
コンフィギュレーションパラメーター		
名称	意味	
Human_NUM	検出する人数(最大 10 人まで)	
Environmental_maxdata	最大環境データ	
LRFdata_min	LRF で使用するレーザの最小ステップ番号	
LRFdata_max	LRF で使用するレーザの最大ステップ番号	
LRFdata_mid	LRF で使用するレーザの中間ステップ番号	
LRF_Total_steps	LRF のトータルステップ数	
LRFdata_theta	走査角度(角度分解能算出に使われている角度)	
Human_width	物体検出の最小幅	
Count_mode	人数カウントを行う際の入退室の向き変更	
Count_area_distance	人数カウントを行う際のエリア距離	
Human_setdistance	人認識を行う最小距離	
Human_TrackingData	人追跡処理を行う範囲	

## 補足:複数人追跡 RTC の使い方

- ① range\_data(InPort)に LRF から取得した距離データを入力する。  
LRF のコンポーネントは、各自で用意したもの、または 4 章の参考資料[2]から「URG 計測 RTC」をダウンロードしたものを使用する。
- ② LRF を大体**腰の位置**になるようにセットし、使用する LRF に合わせてコンフィギュレーションパラメーターを変更する。
- ③ LRF の RTC 及び複数人追跡 RTC を Activate し、複数人検知及び追跡を開始する。
- ④ OutPort では上記の表にまとめたように人の歩行情報(位置, 速さ, 向き), 人数カウント数(入室, 退室, 追跡中の人数)が出力される。



## 2.1 複数人追跡方法について(重要)

### 2.1.1 物体検知方法

物体検知には、背景差分を使用している。一度目のスキャンで環境データを取得し、その後はスキャン結果によって物体を検知する。そのため、一度目のスキャン時(コンポーネントを Activate した時)には周りに物体(人)が居ないことが望ましい。また、環境データは一度目のスキャン以降のデータに応じて更新される。

### 2.1.2 人認識条件

このコンポーネントでは、離れた場所から人が現れることを前提としているため、人の認識は図 2.1.1 にある Human\_setdistance の値以上の場所で行う。そのため、LRF の前に突然現れるなどの動作には対応していない。人と認識された場合は、図 2.1.2 のように赤枠の丸とIDが描写される。

### 2.1.3 人追跡条件

ある一定以上の速度(180[mm/s]固定値)で歩行している、あるいは図 2.1.1 にある Human\_TrackingData 以内の場所に存在している場合は追跡を行う。

### 2.1.4 人削除条件

Human\_TrackingData 以上の場所で、ある一定以下の速度(180[mm/s]固定値)で存在する。最大環境データ(図 2.1.1 Environmental\_maxdata)以上で存在する。ある一定以下の速度(180[mm/s]固定値)で、Y軸がマイナス(図 2.1.1 の座標系)の場合に存在する。これらの場合には赤枠の丸とIDを削除する。

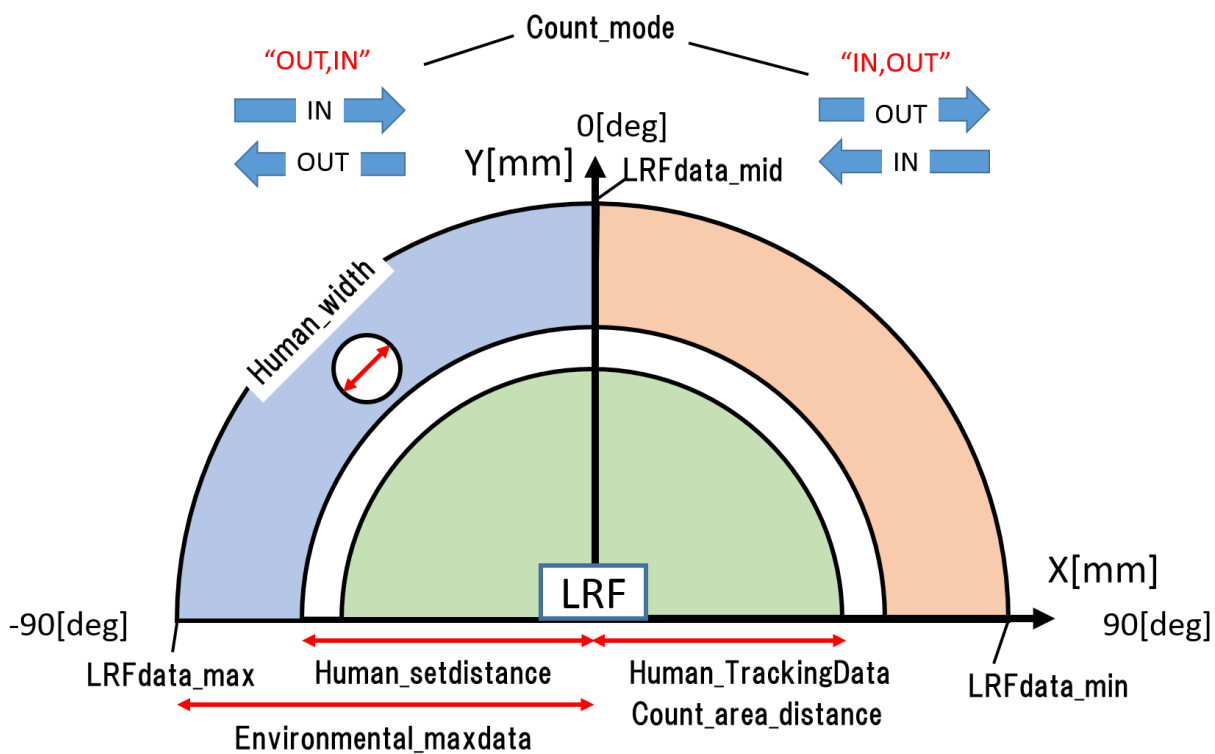


図 2.1.1 コンフィギュレーションパラメーター詳細図

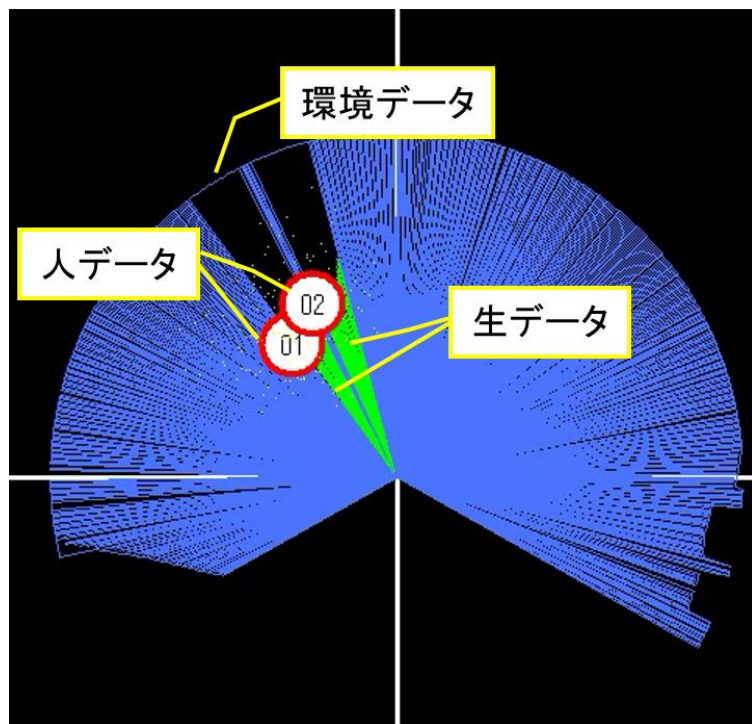


図 2.1.2 複数人追跡の様子

## 2.2 人数カウント機能について

### 2.2.1 人数カウント方法

図 2.1.1 で定めた `Count_area_distance` の値以上のエリアを図 2.2.1 のようにそれぞれを“IN area”，“OUT area”とする。“IN area”から“OUT area”を通過した時，入室にカウント。その逆を通過すると退室にカウントを行う。また，“IN area”を通り“OUT area”を通過せずに“IN area”に戻った場合はカウントされない。

### 2.2.2 入退室方向の入れ替え

入退室方向の入れ替えは図 2.1.1 に示したように，コンフィギュレーションパラメータを変更することで可能。

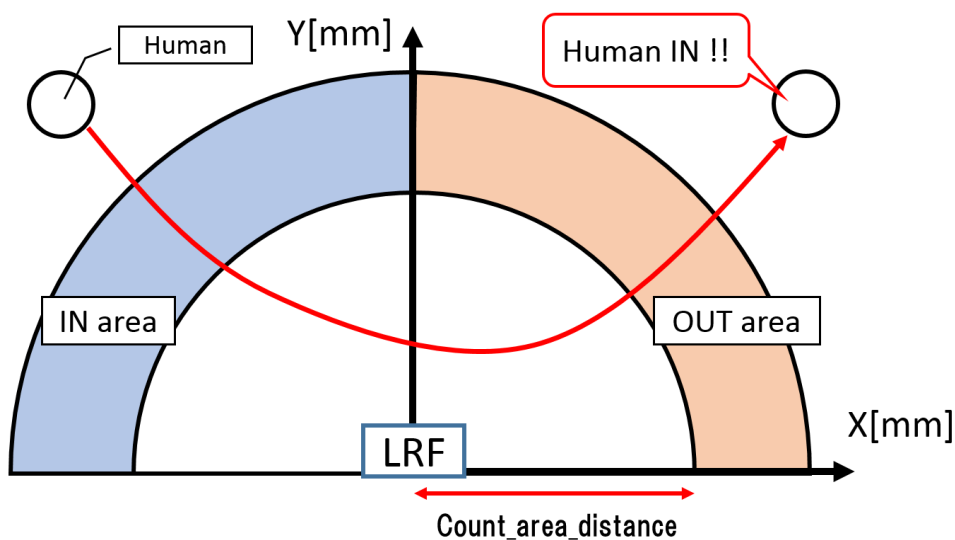


図 2.2.1 人数カウント方法

## 2.3 OutPort について

### 2.3.1 human\_dataX

図 2.3.1 及び図 2.3.2 に示す, LRF を原点とした各人の X 軸座標を TimedLongSeq 型で出力する. その際, 出力する配列の [0] を ID "01" の人のデータとして番号順 ([1]="02", [2]="03...") に出力する. 認識していない人のデータは 100000[mm] と出力.

### 2.3.2 human\_dataY

図 2.3.1 及び図 2.3.2 に示す, LRF を原点とした各人の Y 軸座標を TimedLongSeq 型で出力する. その際, 出力する配列の [0] を ID "01" の人のデータとして番号順 ([1]="02", [2]="03...") に出力する. 認識していない人のデータは 100000[mm] と出力.

### 2.3.3 human\_dataV

図 2.3.2 に示す, 各人の移動速度の絶対値を TimedLongSeq 型で出力する. その際, 出力する配列の [0] を ID "01" の人のデータとして番号順 ([1]="02", [2]="03...") に出力する. 認識していない人のデータは 0[mm/s] と出力.

### 2.3.4 human\_dataT

図 2.3.2 に示す, 各人の移動向きを TimedLongSeq 型で出力する. その際, 出力する配列の [0] を ID "01" の人のデータとして番号順 ([1]="02", [2]="03...") に出力する.

### 2.3.5 count\_data

人数カウントによって記録された入室数, 退室数, 追跡中の人数を TimedLongSeq 型で出力する. その際, 出力する配列の [0] を入室数, [1] を退室数, [2] を追跡中の人数としている.



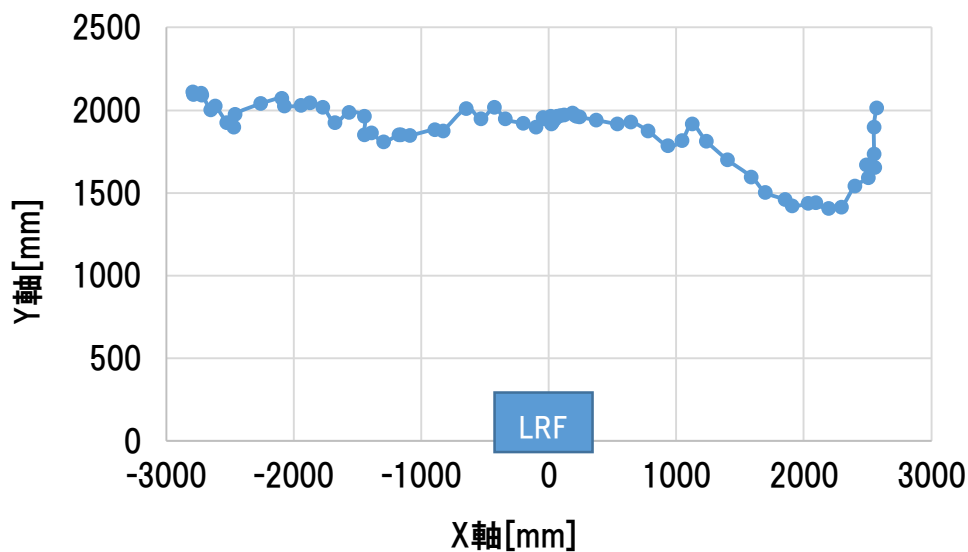


図 2.3.1 出力結果例(X 軸, Y 軸)

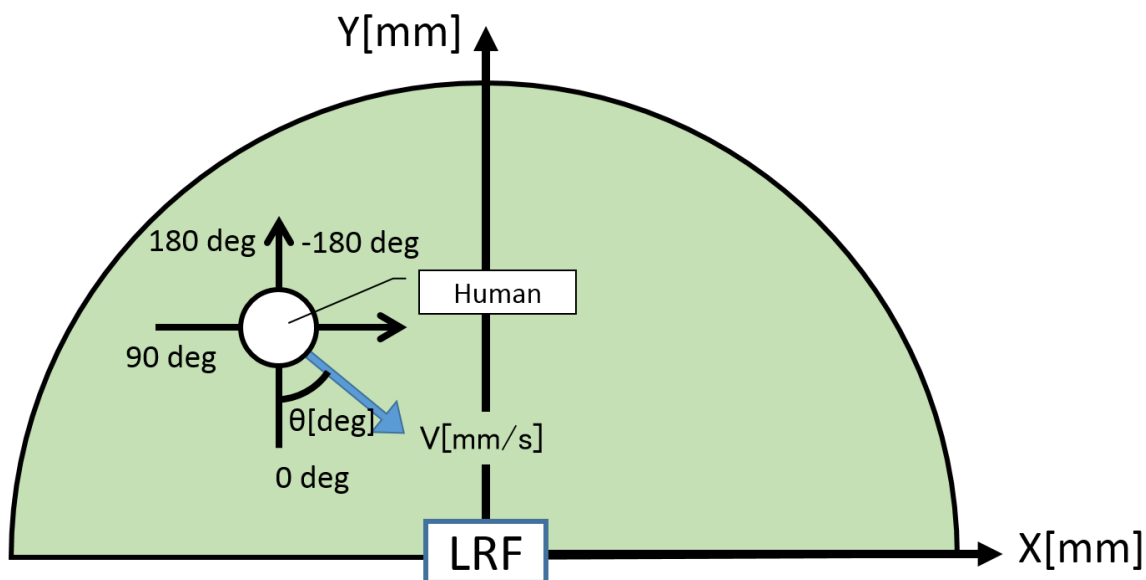
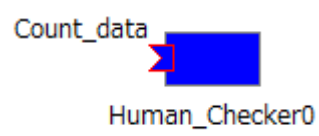


図 2.3.2 出力値の座標系

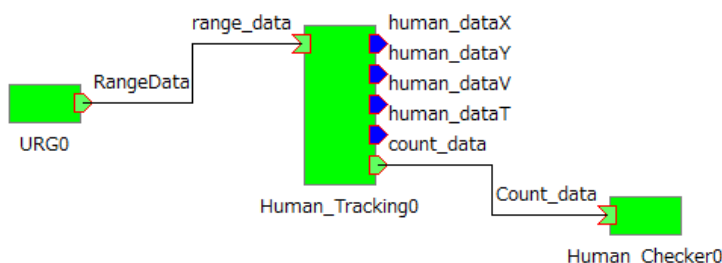
### 3. 人数管理 RSNP RTC

この RTC は InPort から取得した人数カウント情報をサーバに送信する。通信には RSNP という通信プロトコルを使用している。同一サーバ上にデータをアップすることで複数の情報をサーバ上で一括して管理が可能。今回のサーバでは受信されたデータを元に、Web 上に比較しやすい GUI を公開する。

RTC の名称		
Human_Checker		
InPort		
名称	データ型	機能
Count_data	TimedLongSeq	人数カウントデータを入力
コンフィギュレーションパラメーター		
名称	意味	
end_point	アクセスするURLを指定(サーバのグローバルIPを入力する)	
password	パスワードを指定	
robot_id	ロボットIDを指定	

#### 補足:人数管理 RSNP RTC の使い方

- ① Count\_data(InPort)と複数人追跡 RTC の count\_data(OutPart)を接続する。
- ② PocoServer のサーバを立ち上げる。
- ③ 設定した URL(ここでは [http://\"IP アドレス\":8080/PocoServer/Pocodata](http://\))にアクセスする。
- ④ 人数管理 RSNP RTC を Activate し、ブラウザを更新する。



### 3.1 セットアップ

Human\_CheckerRTC(人数管理 RSNP RTC)フォルダ直下の lib フォルダに jar ファイルを加える。

- apache-mime4j-0.6.jar
- commons-codec-1.3.jar
- commons-logging-1.1.1.jar
- httpclient-4.0.jar
- httpcore-4.0.1.jar
- httpmime-4.0.jar
- rsnp-service-api-2.3.0\_r49.jar
- rsnp-service-fjlib-2.3.0\_r49.jar

Human\_CheckerRTC フォルダ直下の Human\_Checker.bat を編集する。編集では上記の jar の CLACCPATH を記述する。

### 3.2 人数管理 RSNP RTC コンフィギュレーションについて

この RTC は以下のコンフィギュレーションパラメーターを設定できる。

- end\_point 接続先のサーバアドレス(ここでは http://“IP アドレス”:8080/PocoServer/Pocodata)  
IP アドレスの部分を変更することで接続できる。例えば, localhost
- password : 8073 に統合しているので今回は使用しない。
- robot\_id : デフォルトでは RobotID:1

### 3.3 サーバの設定について

サーバは, 参考資料[3]のサイトにアクセスし, 「自律・遠隔操作可能な追尾カメラ」 API 仕様書を参考に設定する。 ※WebContent->js フォルダに jquery-2.1.1.min.js を入れる。

今回サーバの設定に使用したソフトウェア及び入手方法は 4 章に記載。

## 4. 使用ソフトウェア及び入手方法, 参考資料

- RSNP ライブラリ v2.3(株式会社富士通研究所)  
「作品使用ライブラリ」  
・rsnp-robot-api-2.3.0\_r49.jar  
・rsnp-robot-fjlib-2.3.0\_r49.jar  
[http://robotservices.org/rsi\\_spec.html](http://robotservices.org/rsi_spec.html)  
(上記サイトを参考に書面またはメールで申し込みが必要)
- Apache Tomcat v7.0.55  
<http://tomcat.apache.org/>
- Axis2 v1.4.1  
<http://axis.apache.org/axis2/java/core/>
- Java SE Development Kit v1.7.0\_65 (JDK)  
<http://www.oracle.com/technetwork/opensource/index.html>
- Eclipse IDE for Java EE Developers v4.3  
<http://www.eclipse.org/downloads/packages/eclipse-ide-java-ee-developers/junosr1>
- apache-mime4j-0.6.jar  
<http://james.apache.org/mime4j/>
- commons-codec-1.3.jar  
<http://commons.apache.org/proper/commons-codec/>
- commons-logging-1.1.1.jar  
<http://commons.apache.org/proper/commons-logging/>
- httpclient-4.0.jar, httpcore-4.0.1.jar, httpmime-4.0.jar  
<http://hc.apache.org/downloads.cgi>
- jquery-2.1.1.min.js  
<http://jquery.com/download/>

[1] Robot Service Network Protocol2,2 仕様書

[2] 「RTM で動作する人追尾可能なカメラモジュールの開発」

<http://www.meo.shibaura-it.ac.jp/matsuhira/RTM.html>

[3] 知能機械システム研究室「自律・遠隔操作可能な追尾カメラ」

[http://www.sic.shibaura-it.ac.jp/~md13008/RTMC2013\\_rsnp\\_camera.html](http://www.sic.shibaura-it.ac.jp/~md13008/RTMC2013_rsnp_camera.html)