

経路計画コンポーネント

平成 23 年 7 月 21 日

豊橋技術科学大学 行動知能システム学研究室

1. このコンポーネントについて

このコンポーネントは静止・移動障害物を回避しながら、指定物体を追う経路を計画するコンポーネントである。

2. 開発・動作環境

このコンポーネントは以下の環境で開発し、動作確認を行った。

- Windows XP Pro SP3
- OpenRTM-aist 1.0.0(C++版)
- Visual Studio 2008
- OpenCV 2.1

3. 入出力データポート

ポート名	データ型	入出力	備考
robot_pose	IIS::TimedPose2D	入力	ロボットの現在位置
robot_velocity	IIS::TimedVelocity2D	入力	ロボットの現在速度
subgoal_waypoint	IIS::TimedPoseVel2DSeq	入力	目標地点の系列
localization_port	IIS::TimedPose2D	入力	確率的自己位置姿勢(不使用)
float_map	MRFC::TimedFloatRelativeOGMapData	入力	地図データ
tracking_people	MRFC::TimedPeopleTrackingData	入力	移動障害物・追跡対象のデータの取得
robot_control	IIS::TimedVelocity2D	Output	制御出力(速度)
planning_img	TUT::TimedImageData	Output	処理経過画像出力

4. サービスポート

MovingObstaclePort からは、移動物体(人物)の位置と速度が取得できる。また、そのうちの一つを追跡するようになっている。情報は MRFC::PeopleTrackingService の getTrackingData というサービスから入力される。このサービスから戻り値として MRFC::TimedPeopleTrackingData 型のデータを渡す。メンバ変数 data のメンバ変数

の中に `MRFC::PersonData` 型の配列がある。この配列の要素の一つ一つが、人物（移動障害物）の情報となる。この中の指定された一つを追跡する。指定しなかったものは、移動障害物として判断され、回避するような経路を作る。追跡対象はメンバ変数 `id` で指定する。人物（移動障害物）配列の要素番号が `id` と同じデータの人物（移動障害物）が追跡される。

ポート名	データ型	ポート型	備考
<code>PeopleTrackingService</code>	<code>MRFC::PeopleTrackingService</code>	<code>Service Consumer</code>	移動障害物・追跡対象のデータを取得
<code>RelativeMapService</code>	<code>MRFC::RelativeMapService</code>	<code>Service Consumer</code>	環境情報の取得

5. インターフェースおよびデータ型について

`MRFC::PeopleTrackingService`

サービス名	引数	戻り値	備考
<code>getTrackingData</code>	なし	<code>MRFC::TimedPeopleTrackingData</code>	対象の速度・絶対位置を取得

`MRFC::RelativeMapService`

サービス名	引数	戻り値	備考
<code>getFloatRelativeOGMap</code>	なし	<code>MRFC::TimedFloatRelativeOGMapData</code>	環境データの取得

`IIS::TimedPose2D`

メンバ名	データ型	備考
<code>tm</code>	<code>RTC::Time</code>	タイムスタンプ
<code>id</code>	<code>sequence<long></code>	(不使用)
<code>data</code>	<code>RTC::Pose2D</code>	ロボットの位置姿勢
<code>error</code>	<code>sequence<double></code>	(不使用)

IIS::TimedVelocity2D

メンバ名	データ型	備考
tm	RTC::Time	タイムスタンプ
id	sequence<long>	(不使用)
data	RTC::Velocity2D	並進速度(m/s),角速度(rad/s)
error	sequence<double>	(不使用)

IIS::TimedPoseVel2DSeq

メンバ名	データ型	備考
tm	RTC::Time	タイムスタンプ
id	sequence<long>	(不使用)
data	sequence<RTC::PoseVel2D>	目標地点の系列
error	sequence<double>	(不使用)

MRFC::TimedPeopleTrackingData 型

メンバ名	データ型	備考
tm	RTC::Time	タイムスタンプ
data	MRFC::PeopleTrackingData	対象の速度・相対距離

MRFC::PeopleTrackingData 型

メンバ名	データ型	備考
id	Long	追跡物体の id
person	sequence<MRFC::PersonData>	移動物体のパラメータ系列

MRFC::PersonData 型

メンバ名	データ型	備考
position	RTC::Point3D	移動物体までの相対位置
velocity	RTC::Velocity2D	移動物体の速度

MRFC::TimedFloatRelativeOGMapData

メンバ名	データ型	備考
mapconfig	RTC::OGMapConfig	地図の大きさやスケール
cells	RTC::OGMapFloatCells	各セルの値を float 型の系列に格納
pose	RTC::Pose2D	ロボット中心の絶対座標
tm	RTC::Time	タイムスタンプ

RTC::OGMapFloatCells

sequence<float> に OGMapFloatCells という新しく名前を付けたもの。

RTC::OGMapConfig

メンバ名	データ型	備考
xScale	double	X 軸方向の地図のスケール[m/cell]
yScale	double	Y 軸方向の地図のスケール[m/cell]
width	double	X 軸方向の地図の大きさ[cell]
height	double	Y 軸方向の地図の大きさ[cell]
origin	RTC::Pose2D	cell(0,0) のロボット中心から見た絶対座標

TUT::TimedImageData 型

メンバ名	データ型	備考
Tm	RTC::Time	タイムスタンプ
Data	TUT:: ImageData	画像データ

TUT::ImageData 型

メンバ名	データ型	備考
nChannels	long	チャンネル数
depth	long	1 画素あたりのビット数
origin	long	画像データの原点 (基準) 0:左上原点 (デフォルト) 1:左下原点
width	long	画像の幅(画素数)
height	long	画像の高さ(画素数)
imageSize	long	画像データのサイズ (バイト数)
imageData	sequence<char>	各画素の値
widthStep	long	画像データの幅のバイト数

(*)TUT::ImageData の構造は OpenCV の IplImage 構造体に準拠している。

6. 準備

このコンポーネントを使用するためには、OpenCV ライブラリ（配布元：<http://sourceforge.net/projects/opencvlibrary/>）が必要である。インストール方法は以下の通りである。

イ) 配布サイトなどでOpenCV-2.1.0-win32-vs2008.exeをダウンロードし、実行する。

ロ) 環境変数Pathに”C:¥OpenCV2.1¥bin”を追加する。

また、PathPlanner コンポーネントは単体で動作させる事が出来ないため、robot_velocity, robot_control, MovingObstaclePort, StaticObstaclePort に接続するコンポーネントを最低限用意する必要がある。なお、planning_img, subgoal_waypoint への接続は無くても動作させる事が出来る。robot_pose, localization_port はそれぞれロボットの位置を取得するためのポートであるため、どちらか1つに接続するコンポーネントを用意すればよい。

7. 起動手順

イ) 圧縮ファイルを展開する。

ファイルの中身は図1のようにになっている。

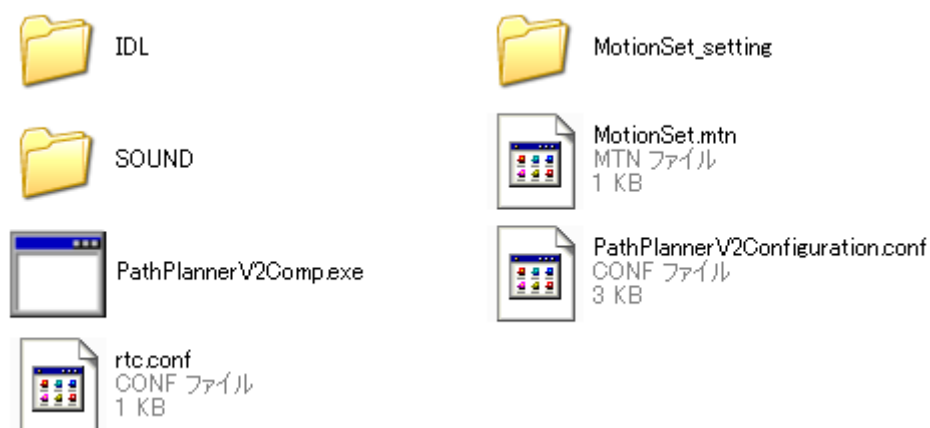


図1 解凍フォルダの中身

(a) “IDL”フォルダ

このコンポーネントの独自のデータ型・サービスを定義した IDL ファイルが格納されているフォルダ。

(b) “SOUND”フォルダ

デバッグ用ロボット音源を格納しているフォルダ。

(c) “MotionSet.mtn”ファイル

初期設定で指定されている、動作セットの定義ファイル。

(d) “MotionSet_setting”フォルダ

動作セットの定義ファイルを作成するアプリケーションと動作セットの定義フ

ファイルの内容を確認するアプリケーションが格納されている。

(e) rtc.conf

RT コンポーネントのさまざまな設定(ネーミングサービス・動作周波数・コンフィグファイルのパス等)を記述するための設定ファイル。

(f) PathPlannerV2Configuration.conf

あらかじめ用意された複数のコンフィギュレーションセットの値を指定したコンフィグファイル。

(g) “PathPlannerV2Comp.exe”ファイル

本モジュールの実行ファイル。

ロ) ネームサーバーを起動する。

スタート > すべてのプログラム > OpenRTM-aist > C++ > examples > Start Naming Service を選択する。

ハ) モジュールの起動

展開したフォルダ内にある『PathPlannerV2Comp.exe』を起動する。また、robot_pose, robot_velocity, getTrackingDataPort, LocalMapServicePort に接続するためのコンポーネントも起動しておく。

二) RTC System Editor での操作

eclipse を起動する。

(a) メニューバーより、「ウィンドウ>パースペクティブを開く>その他」と辿り、「パースペクティブを開く」ウィンドウを表示させる(図 2)。

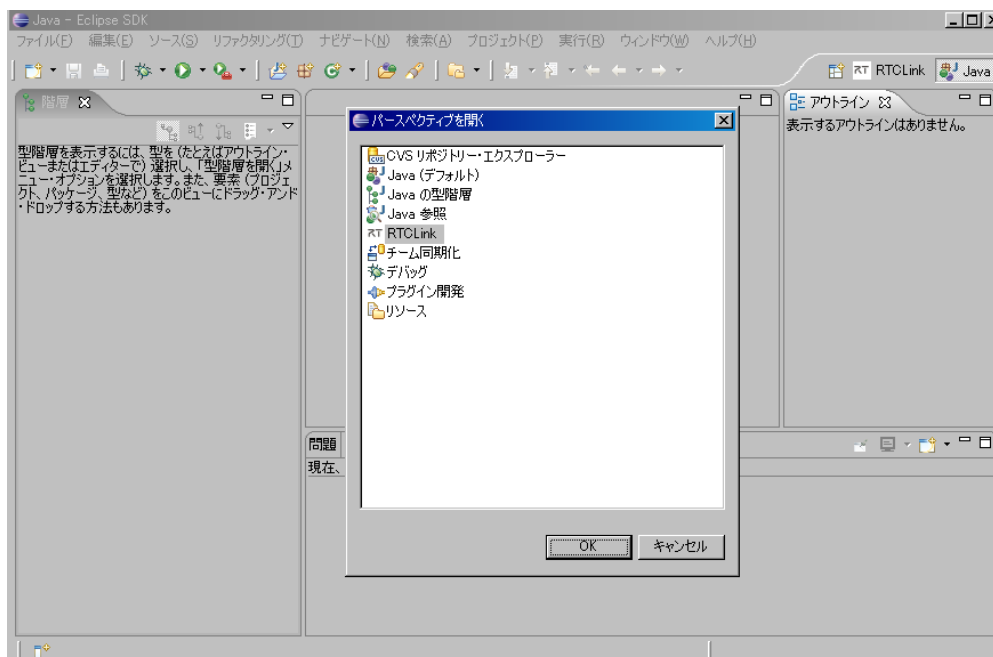


図 2 「パースペクティブを開く」ウィンドウを表示

(b) 「RTC System Editor」を選択し、OK ボタンを押す。

- (c) 図3の赤い丸で囲んだアイコンを選択する。(NameServiceViewが表示されていない場合は、メニューバー→ウィンドウ→ビューの表示→NameServiceViewを選択)
- (d) 図3のようにConnect Name ServerのAddress Portに『localhost』と入力してOKを選択する。

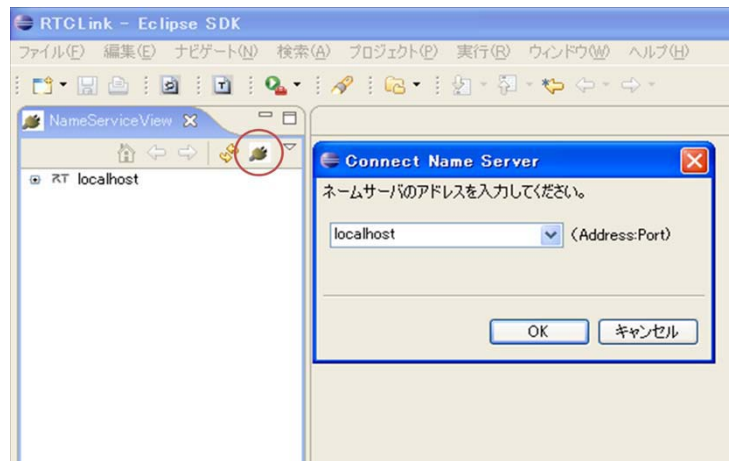


図3 NameServiceViewの操作

- (e) NameServiceViewのlocalhostのツリーを開き、③で起動したモジュールが表示されていることを確認する。
- (f) メニューバーより、ファイル>Open New System Editorを選択する。
- (g) NameServiceView上のモジュールを選択して、System Editor上にドラッグしてモジュールのアイコンを表示させる。
- (h) System Editor上でPathPlannerコンポーネントを左クリックし、ConfigurationView(図4)においてパラメータの設定を行う(ConfigurationViewが表示されていない場合は、メニューバー→ウィンドウ→ビューの表示→ConfigurationViewを選択)。設定可能なパラメータの値は表1に示す。値を変更後、Applyボタンを押す。
- (i) なお、本モジュールでは富士通サービスロボット「enon」と、Mobile Robots社ロボット「PeopleBot」を動かすために適切な初期パラメータを事前に用意している。適切なパラメータを利用するためには、「ENON」「PeopleBot」と書かれたラジオボタンを選択する。また、この場合もApplyボタンを押さなければならない事に注意する必要がある。

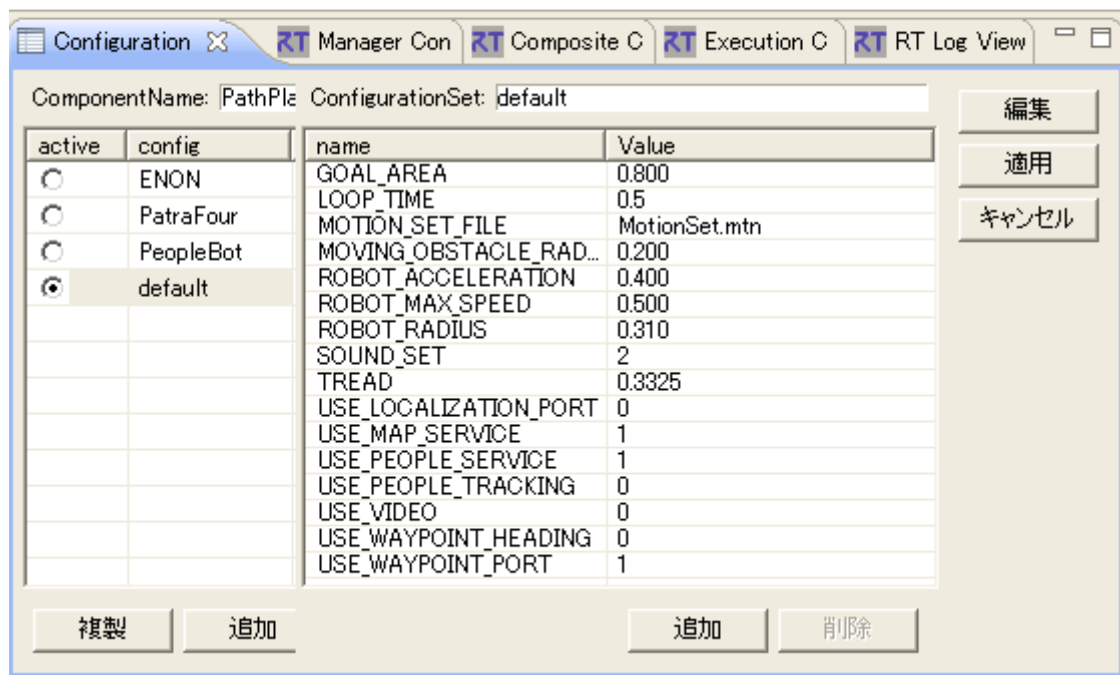
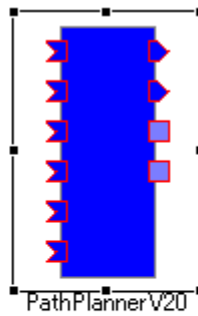


図 4 ConfigurationView によるパラメータの設定画面

表 1 設定可能なパラメーター一覧

Parameter Name	Default	Description
GOAL_AREA	0.8	ロボットとゴール間の距離が指定値より小さくなったらロボットが停止する。(単位:メートル)
LOOP_TIME	0.5	経路計画のリプランニング周期の設定。(単位:秒)
MOTION_SET_FILE	MotionSet.mtn	ロボットが使用するモーションセットの定義

MOVING_OBSTACLE_RADIUS	0.2	移動障害物の半径(単位：メートル)
ROBOT_ACCELERATION	0.4	ロボットの加速度(単位：m/s ²)
ROBOT_MAX_SPEED	0.5	許容するロボットの最大速度(単位：m/s ²)
ROBOT_RADIUS	0.310	ロボットの半径(単位：メートル) ENON, PeopleBot, PatraFourの3種類
SOUND_SET	2	デバッグ音声の設定(0: 音声不使用 1: 音声セット1使用 2: 音声セット2使用)
TREAD	0.3325	車輪トレッドの指定 (単位：メートル) ENON, PeopleBot, PatraFourの3種類
USE_WAYPOINT_PORT	2	目標地点に対して経路点を使うかどうか。(0: 経路点は使わず, 人物追跡データを使用する. 1: 経路点を使用. 局所地図中で最も遠い経路点を目指す. 2: 経路点を使用. 各経路点の一つ一つを追従.)
USE_LOCALIZATION_PORT	0	robot_poseデータポートからロボットの現在位置を得る場合は0を指定. localization_portデータポートからロボットの現在位置を取得する場合は1を指定.(現在は使用していない)
USE_WAYPOINT_HEADING	0	経路点にロボットの方向を使うかどうか(0:使わず, 位置のみ; 1: 使う)
USE_PEOPLE_TRACKING	0	人物追跡を使うかどうか (0: 使わない; 1: 使う)
USE_MAP_SERVICE	1	map service port を使うかどうか (0:サービスポートは使わず,データポートを使う; 1: サービスポートを使う)

USE_PEOPLE_SERVICE	0	people tracking service port を使うかどうか (0: サービスポートは使わず, データポートを使う; 1: サービスポートを使う)
USE_VIDEO	0	local map の結果をビデオで撮るかどうか (0: ビデオを撮らない; 1: ビデオを撮る) (※未実装)

(j) 図 5 のようにモジュールを接続する。

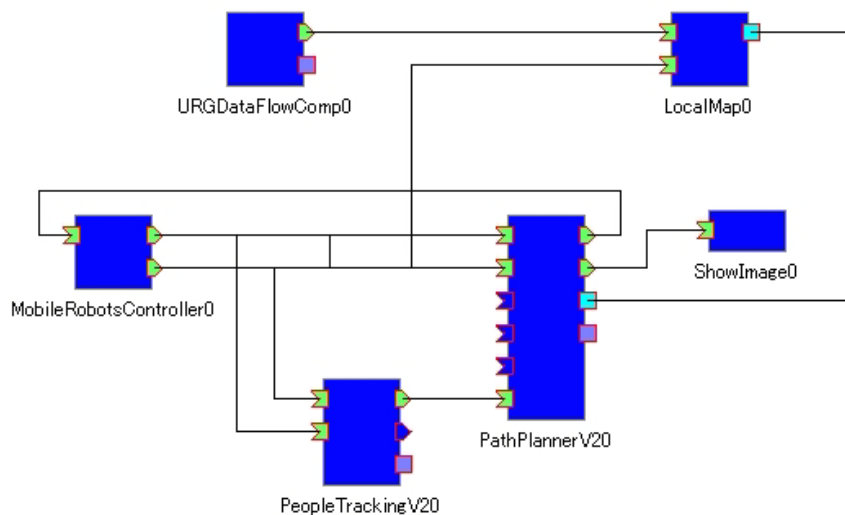


図 5 コンポーネントの接続例

ホ) モジュールのアクティベート

ツールバーの「All Activate」ボタンを押して全モジュールをアクティベートする。他のコンポーネントに接続順序についての説明があるときはそれに従う。

へ) 動作確認

SOUND_SET の値を 1 に設定していた場合、音声によるナビゲーション/デバッグが実行される。

また、planning_img ポートに対応する画像表示 RTC を接続している場合、図 6 のようなロボットを中心とした地図上において、ロボットが経路を探索している様子が表示される。ここで、画像中央の橙色の物体がロボットであり、ゴール地点は赤丸で示している。ロボットが移動可能な領域は青で示され、黒い部分は移動不可能な領域である。緑色で示しているのは静止障害物あるいは止まっている

人物が存在する地点であり，三角形で示しているのは移動中の人物とその向きである．ロボットがこれまでに通った経路は橙色のラインで示され，黄色で探索した経路を示す．

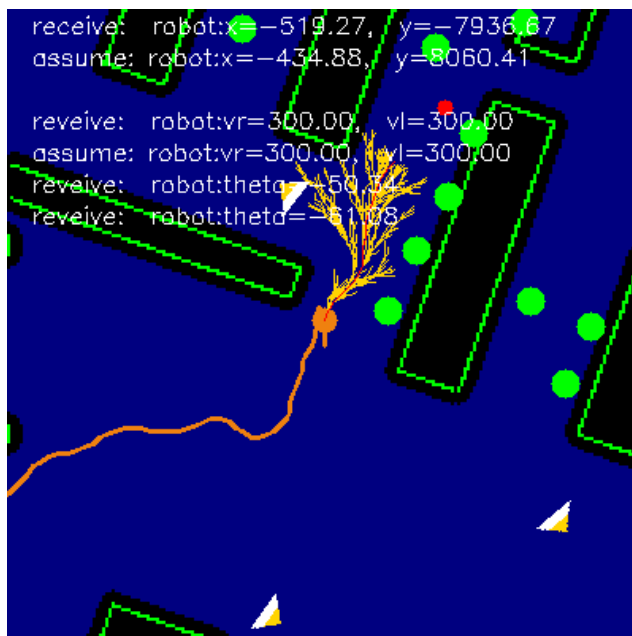


図 6 経路計画の様子(画像表示 RTC 利用時)

8. 動作セットについて

イ) 動作セットの定義

動作セットを定義し，その定義ファイルを作成する場合には，“MotionSet_setting”フォルダ内の”make_MotionSet.exe”を実行する．

以下のように表示されるので，出力する動作セットの定義ファイルのパスを入力する．

(注意：ここで指定したファイルが既に存在する場合は，内容が上書きされる.)

```
Please input file name to create a motion file.
```

```
_
```

以下は，ファイル名として”test.mtn”を入力した場合である．

```
test.mtn  
create file : test.mtn
```

次に，以下のように表示されるので，定義する動作の数を入力する．

```
motion num = _
```

すると，以下のような表示がされる．(下図は”29”を入力した場合)

```
motion num =29  
OK. motion num = 29.
```

次に、上で入力した回数分の動作を登録していく。

各動作の登録は、旋回半径を[mm]単位で入力する。ただし、直進の場合は旋回半径に0を指定する。

```
input 'R V'  
1:  
R=
```

次に、速度の入力を求められるので、[mm/sec]単位で指定する。

```
V=
```

ロ) 動作セット定義ファイルの内容確認

動作セット定義ファイル内に定義されている動作を確認するには、“MotionSet_setting”フォルダ内の”read_MotionSet.exe”を実行する。

以下のように表示されるので、動作セット定義ファイル名のパスを入力する。

```
Please input read file name.  
-
```

すると以下のように、そのファイルに定義されている動作の数と各動作の旋回半径[mm]と速度[mm/s]が表示される。

```
motion_num = 29  
1:r=-1600.00 v=-50.00  
2:r=-1600.00 v=50.00  
3:r=-1600.00 v=100.00  
4:r=-1600.00 v=150.00  
5:r=-1200.00 v=-50.00  
6:r=-1200.00 v=50.00  
7:r=-1200.00 v=100.00  
8:r=-1200.00 v=150.00  
9:r=-800.00 v=-50.00  
10:r=-800.00 v=50.00  
11:r=-800.00 v=100.00  
12:r=-800.00 v=150.00  
13:r=0.00 v=-50.00  
14:r=0.00 v=0.00  
15:r=0.00 v=50.00
```

9. 連絡先について

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