

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

フォーメーション制御モジュール群

京都大学 松野研究室

2010/10/8

--	--	--

目次

1. はじめに.....	5
1.1 本書の適用範囲.....	5
1.2 関連文書.....	5
1.3 本文書を読むに当たって.....	5
2. 機能仕様.....	6
2.1 機能概要.....	6
2.2 モジュール構成.....	6
3. RTC 仕様.....	8
3.1 FormationCenter モジュール.....	8
3.1.1 機能概要.....	8
3.1.2 動作環境.....	8
3.1.3 入出力ポート情報.....	8
3.1.4 設定ファイル.....	9
3.2 FormationController モジュール.....	10
3.2.1 機能概要.....	10
3.2.2 動作環境.....	10
3.2.3 ポート情報.....	10
3.2.4 設定ファイル.....	11
3.3 MultiRobotMux モジュール.....	12
3.3.1 機能概要.....	12
3.3.2 動作環境.....	12
3.3.3 ポート情報.....	12
3.4 SimpleFormationGUI モジュール.....	13
3.4.1 機能概要.....	13
3.4.2 動作環境.....	13
3.4.3 ポート情報.....	14
3.5 SimpleJoyPad モジュール.....	14
3.5.1 機能概要.....	14
3.5.2 動作環境.....	15

3.5.3	ポート情報	15
4.	モジュール使用方法	17
4.1	環境準備	17
4.2	モジュールの起動	17
4.3	使用方法	19

1. はじめに

1.1 本書の適用範囲

本書は、複数台の移動ロボットを協調走行(フォーメーション走行)させるためモジュール群の構成及びモジュール群の使用手順を記述したものである。

1.2 関連文書

本書では本知能モジュール群に含まれない知能モジュールも使用し説明を行っている。本知能モジュール群に含まれないモジュールの詳細については以下を参照されたい。

モジュール名	文章名
beego 制御用モジュール	「 beego 制御用モジュール仕様書.pdf 」

1.3 本文書を読むに当たって

本書はRTミドルウェア, RTコンポーネント(以下, RTC)に関する基本知識を備えた利用者を対象としている。RTミドルウェア, RTCについては下記を参照されたい。

- ・ OpenRTM-aist Official Website:
<<http://www.openrtm.org/OpenRTM-aist/html/>>

また、本書で記述しているモジュールは C++、Java 言語でコーディングされており、C++、Java 言語についての基礎知識があることが望ましい。

2. 機能仕様

2.1 機能概要

本知能モジュール群は、複数の移動体がフォーメーションを組み、集団で走行することを実現するものである。「フォーメーションを組んで走行する」とは、一定の幾何学的形状を維持したまま走行することを意味する。その一例を図1に示す。図1は3台の移動体が三角形の形状を維持しながら走行する様子を示したものである。ここでは、Leaderとは、操作者が操作する移動体であり、FollowerはLeaderに対して指定した位置関係で自律的に追従する移動体のことである。

本知能モジュール群は、このような Leader-Follower 型のフォーメーション制御手法に基づき、複数台のロボット編隊走行を実現するための計5つのモジュールを提供するものである。なお、本知能モジュールが適用可能な移動体は並進・角速度で制御可能な(2輪型の)機構を持つものである。また、使用には移動体を制御するための知能モジュールも別途必要となる。

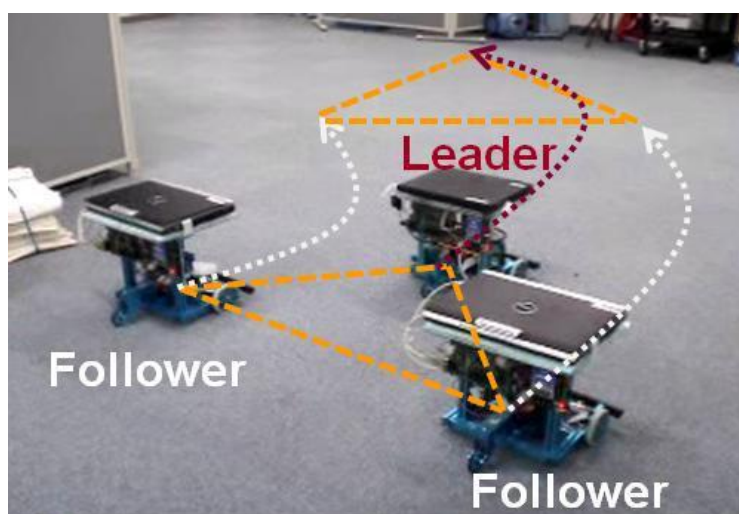


図1 フォーメーション走行（三角形形状の例）

2.2 モジュール構成

本知能モジュール群の構成例を図2に示す。図2は移動体として3台の beego (テクノクラフト社製)を用いた例である。図2に示すように、本知能モジュール群には以下の5つの知能モジュールが含まれる。

- ①SimpleJoyPad, ②SimpleFormationGUI, ③FormationCenter,

④FormationController, ⑤(beego), ⑥MultiRobotMux

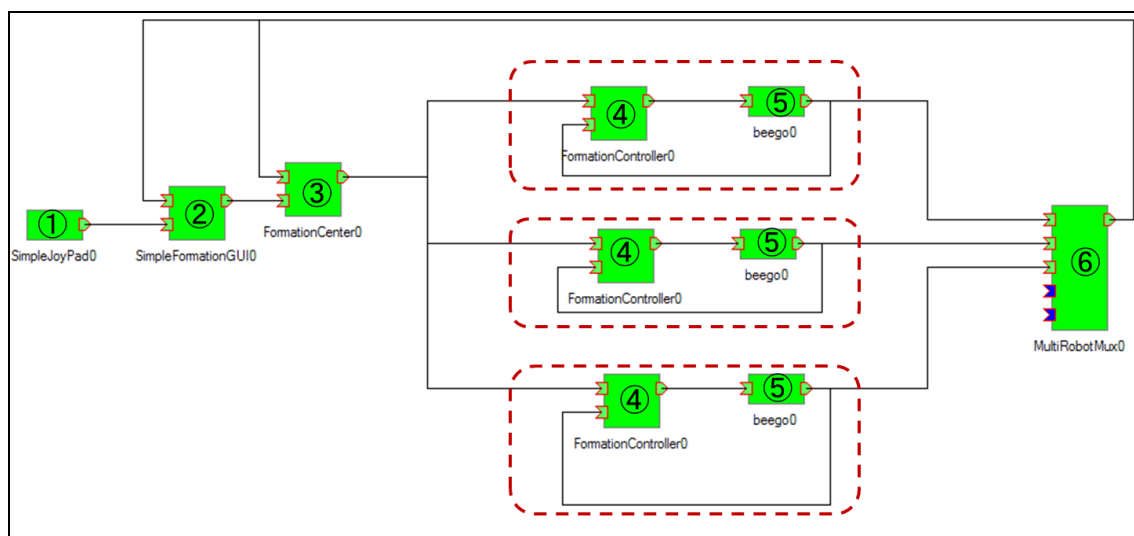


図 2 本知能モジュール群の構成例(beego3台の場合)

なお、ロボットの台数の増減は赤線部分に含まれる④FormationController と⑤beegoの計2モジュールをロボットの台数に応じて増減させるのみで対応が可能である。本知能モジュール群では最大5台の移動体まで対応している。

次章では①～⑥(⑤除く)の各知能モジュールの詳細について説明する。

3. RTC 仕様

本章では、本知能モジュール群に含まれる 5 つの知能モジュール群の詳細について順に説明をしていく。

3.1 FormationCenter モジュール

3.1.1 機能概要

本知能モジュールは、SimpleFormationGUIによって決定されたメンバ情報、形状情報およびLeader/Follower情報に基づき、実際に移動体(Follower)の目標位置の割り当てを行うものである。各移動体(Follower)の目標位置は、衝突の可能性・移動距離を考慮し自動的に決定される。

3.1.2 動作環境

FormationCenter モジュールは以下の環境で動作確認を行っている。

動作確認済み OS	Windows (VC2008) / Linux (Ubuntu 10.04)
開発言語	C++
RT ミドルウェアバージョン	OpenRTM-aist 1.0.0-RELEASE
依存ライブラリ	なし

3.1.3 入出力ポート情報

本モジュール(図 3 参照)の入出力及び Configuration を以下に示す。

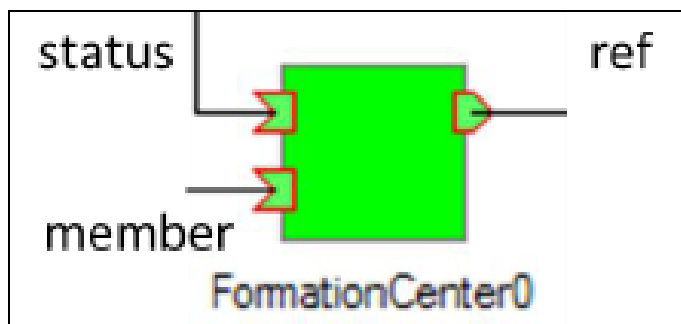


図 3 FormationCenter モジュール

A) データポート(Inport)

名前	型	説明
----	---	----

status	RobotDataSeq	全移動ロボットの位置姿勢情報
member	FormationMemberSeq	SimpleFormationGUI で決定したグループ情報

B) データポート(OutPort)

名前	型	説明
ref	FormationReferenceSeq	各移動ロボットの目標位置情報

C) Configuration

名前	型	説明
patternfile	std::string	目標形状を定義したファイル名(次節参照)

3.1.4 設定ファイル

ここでは、FormationCenterモジュールに用いる設定ファイルpattern.confの詳細について説明する。「pattern.conf」は形状情報を定義したファイルであり、このファイルに形状情報を追加することで、任意の形状を使用することが可能となる。具体的には以下に示すようなフォーマットで記述する。

patternname,ref1_x,ref1_y,ref2_x,ref2_y,

上記の各要素は以下の意味を持つものである。

- patternname：形状の名前(唯一なもの)。この名前でパターンを特定。
- ref1_x：1つめの目標位置のx座標値(図4参照)
- ref1_y：1つめの目標位置のy座標値
- ref2_x：2つめの目標位置のx座標値
- ref2_y：2つめの目標位置のy座標値

このように、Leaderに固定した移動座標系で必要な台数のFollowerの目標位置を記述していくものである。

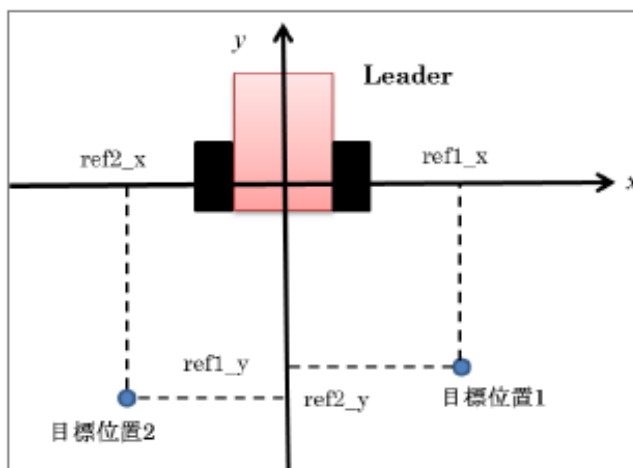


図 4 目標形状指定時の座標系の定義

3.2 FormationController モジュール

3.2.1 機能概要

本知能モジュールはFormationCenterモジュールによって決定された目標位置に追従するための制御入力を計算するものである。Leaderと選択された場合にはSimpleFormationGUIに入力された制御入力を出力し、Followerと選択された場合にはフォーメーション制御則に基づき、目標位置に追従する制御入力を計算・出力する。

3.2.2 動作環境

FormationController モジュールの動作環境は以下に示す通りである。

動作確認済み OS	Windows(VC2008)/Linux(Ubuntu 10.04)
開発言語	C++
RT ミドルウェアバージョン	OpenRTM-aist 1.0.0-RELEASE
依存ライブラリ	なし

3.2.3 ポート情報

本モジュール(図 5 参照)の入出力及び Configuration について以下に示す。

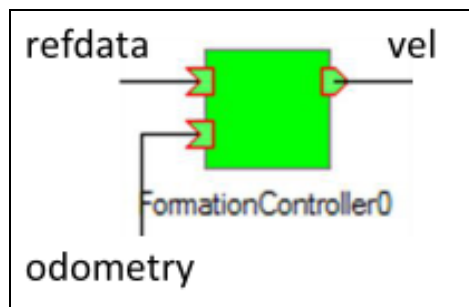


図 5 FormationController

A) データポート(Inport)

名前	型	説明
refdata	FormationReferenceSeq	目標位置情報
odometry	IIS::TimedPose2D	制御対象の位置情報(vel が入力される移動体の位置情報)

B) データポート(OutPort)

名前	型	説明
----	---	----

vel	IIS::TimedVelocity2D	制御対象への速度指令値
-----	----------------------	-------------

C) Configuration

名前	型	説明
id	int	目標形状を定義したファイル名(次節参照)
gain	double	フィードバックゲイン(0<gain<1)
max_vel	double	最大の並進速度[m/sec]
max_avel	double	最大の回転速度[rad/sec]
offset	double	制御点と車軸中心とのオフセット[m]

3.2.4 設定ファイル

FormationControllerのConfigurationはこの設定ファイルを適当に変更することで、変更可能である。下記がFormationController.confに記入している内容であり、対応する値(赤字部分)を変更すればよい。

```

conf.default.id:0
conf.default.gain:0.3
conf.default.max_vel:0.5
conf.default.max_avel:1.0
conf.default.offset:0.15

```

なお、FormationControllerの「id」は以下に示すようにMultiRobotMuxに接続するポートに対応した値に設定する(上から0~4と設定)。

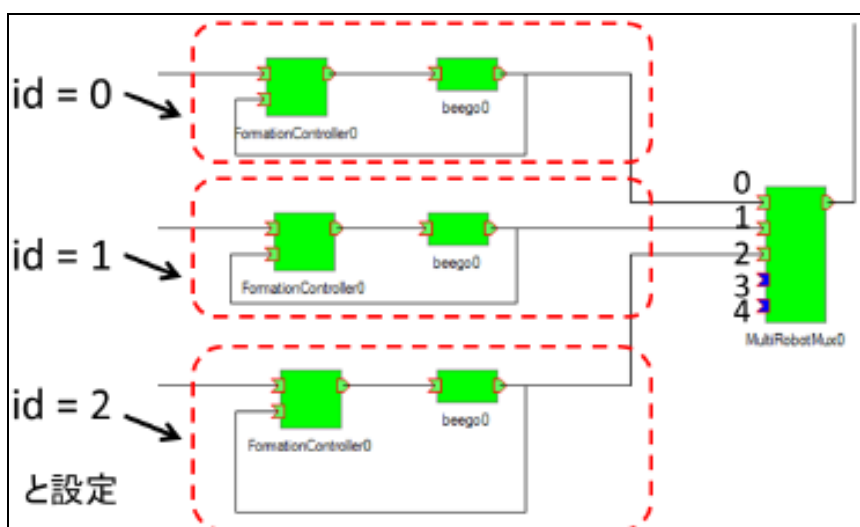


図 6 id の設定ルール

3.3 MultiRobotMux モジュール

3.3.1 機能概要

本知能モジュールは複数台のロボットの位置情報 (IIS::TimedPose2D) を集約し、1つの出力として出力するものである。

3.3.2 動作環境

MultiRobotMux モジュールの動作環境は以下に示す通りである。

動作確認済み OS	Windows (VC2008) / Linux (Ubuntu 10.04)
開発言語	C++
RT ミドルウェアバージョン	OpenRTM-aist 1.0.0-RELEASE
依存ライブラリ	なし

3.3.3 ポート情報

本モジュール(図 7 参照)の入出力及び Configuration を以下に示す。

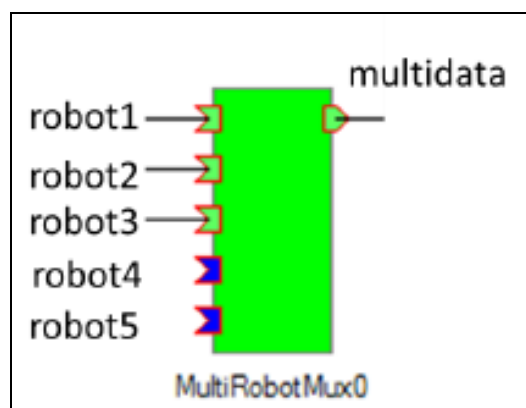


図 7 MultiRobotMux

A) データポート (Inport)

名前	型	説明
robot1	IIS::TimedPose2D	Robot1 の位置情報
robot2	IIS::TimedPose2D	Robot2 の位置情報
robot3	IIS::TimedPose2D	Robot3 の位置情報
robot4	IIS::TimedPose2D	Robot4 の位置情報
robot5	IIS::TimedPose2D	Robot5 の位置情報

B) データポート(OutPort)

名前	型	説明
multidata	RobotDataSeq	全ロボットの位置情報

C) Configuration

名前	型	説明
なし		

3.4 SimpleFormationGUI モジュール

3.4.1 機能概要

本知能モジュールはGUI上で、フォーメーションに含める移動体の設定(グルーピング)とLeader/Followerの役割決め、および形状を設定するためのものである。図8にそのGUI画面を示す。

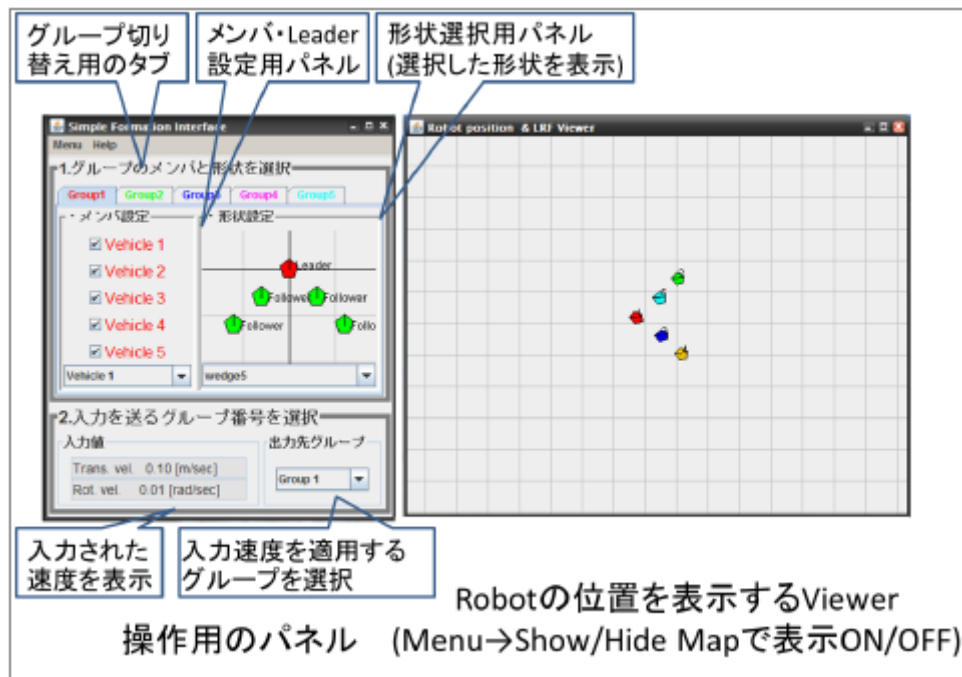


図8 SimpleFormationGUI の UI 概要

3.4.2 動作環境

SimpleFormationGUI モジュールの動作環境は以下に示す通りである。

動作確認済み OS	Windows / Linux (Ubuntu 10.04)
-----------	--------------------------------

開発言語	JAVA (JDK 1.6.0.0220)
RT ミドルウェアバージョン	OpenRTM-aist 1.0.0-RELEASE
依存ライブラリ	なし

3.4.3 ポート情報

本モジュール(図 9 参照)の入出力及び Configuration について以下に示す。

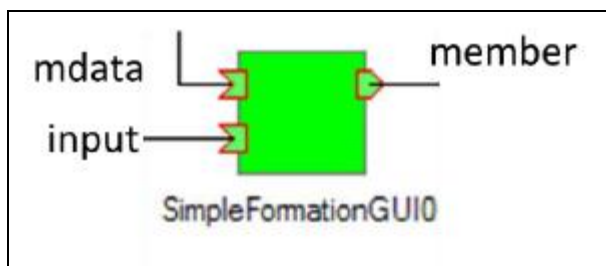


図 9 SimpleFormationGUI

A) データポート(Inport)

名前	型	説明
mdata	RobotDataSeq	全ロボットの位置情報
input	IIS::TimedVelocity2D	Leader への速度指令値

B) データポート(OutPort)

名前	型	説明
member	FormationMemberSeq	操作するグループの情報

C) Configuration

名前	型	説明
patternfile	std::string	形状を定義するファイル名。FormationCenter と同じ内容のこと。3.1.5 節参照)

3.5 SimpleJoyPad モジュール

3.5.1 機能概要

本知能モジュールはIIS::TimedVelocity2D型の出力をGUI画面からの入力によって生成する簡易GUIである(図10参照)。本知能モジュールはLeaderに対する制御入力を生成するために用いられており、これを他モジュールと置き換えることで容易に機能を拡張することができる(例えば経路追従を実現するモジュールなど)。

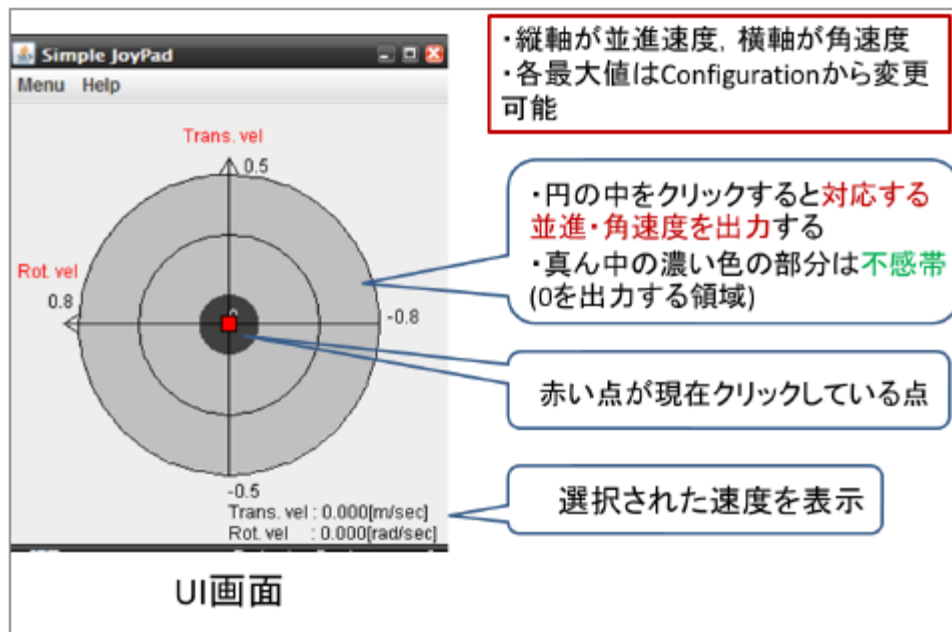


図 10 SimpleJoyPad の UI 概要

3.5.2 動作環境

SimpleJoyPad モジュールの動作環境は以下に示す通りである.

動作確認済み OS	Windows / Linux (Ubuntu 10.04)
開発言語	JAVA (JDK 1.6.0.0220)
RT ミドルウェアバージョン	OpenRTM-aist 1.0.0-RELEASE
依存ライブラリ	なし

3.5.3 ポート情報

本モジュール(図 11 参照)の入出力

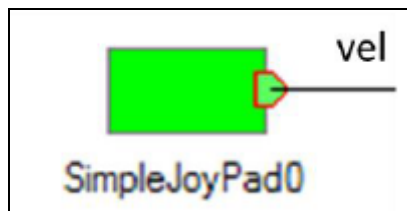


図 11 SimpleJoyPad

A) データポート(Inport)

名前	型	説明
なし		

B) データポート(OutPort)

名前	型	説明
vel	IIS::TimedVelocity2D	GUI上で選択した並進速度[m/sec]と回転速度[rad/sec]

C) Configuration

名前	型	説明
double_maxvel	double	最大の並進速度[m/sec]
double_maxavel	double	最大の回転速度[rad/sec]

3.5.4 設定ファイル

SimpleJoyPadのConfigurationはこの「SimpleJoyPad.conf」を適当に変更することで、変更可能である。下記がSimpleJoyPad.confに記入している内容であり、対応する値(赤字部分)を変更すればよい。

```
conf.default.double_maxvel:0.5  
conf.default.double_maxavel:0.8
```


4. モジュール使用方法

ここでは本モジュールを実行する上で必要となる環境の構築方法とモジュールの実行方法について説明する。なお、本パッケージに含めた「構成例」を用いて説明を行う。構成例には beego 制御用モジュールも含めているが、他ロボットへ適用する場合は beego 制御モジュールを置き換えて実行することで適用可能である。また、構成例では、Windows 7 32bit と Ubuntu 10.04 32bit でコンパイルしたモジュールを格納している。それ以外の環境で実行する場合には、ソースからコンパイルし、置き換えて使用すればよい。

※構成例には Operator フォルダ、robot1~5 フォルダが含まれている。

※Operator フォルダは操作者側で実行するモジュールをまとめたもの

※robot1~3 は各移動体上で実行するモジュールをまとめたもの

4.1 環境準備

モジュールを使用するには OpenRTM-aist 1.0.0 が動作する環境を整える必要がある。以下の手順で使用する環境に応じてインストールを行う。

(1) OpenRTM-aist-1.0.0 Release(C++, JAVA)をインストール

・ C++

【ダウンロード】 <http://www.openrtm.org/openrtm/ja/node/849>

【マニュアル】 <http://www.openrtm.org/openrtm/ja/node/999>

・ JAVA 版

【ダウンロード】 <http://www.openrtm.org/openrtm/ja/node/933>

【マニュアル】 <http://www.openrtm.org/openrtm/ja/node/657>

(2) RT SystemEditor をインストール

【ダウンロード】 <http://www.openrtm.org/openrtm/ja/node/941>

【マニュアル】 <http://www.openrtm.org/openrtm/ja/node/676>

4.2 モジュールの起動

ここでは RTSystem Editor による GUI 環境上でモジュールの起動・操作する方法について説明する。

(1) operator フォルダ、robot1~5 フォルダをコピー

構成例に含まれる `operator` フォルダは操作者側で実行するモジュールをまとめたもの、`robot1~3` は各移動体上で実行するモジュールをまとめたものである。したがって、これらのフォルダをそれぞれ実行する PC 上にコピーする必要がある。`Operator` フォルダは操作者側の PC に、`robot1~3` は各移動体上に搭載した PC にコピーする。なお、使用する移動体の台数に応じて使用する `robot` フォルダ数を変更する必要がある(例えば移動体が 3 台の場合には `robot1~3` を使用)。

(2) `rtc.conf`の調整

環境に応じて`rtc.conf`の一行目にある

```
corba.nameservers: localhost:2809
```

のホスト名とポート番号を変更する。

なお、`rtc.conf`は`operator`フォルダと`robot1~5`の各フォルダにあるので、そのすべてを実行環境に応じて変更する必要がある。

(3) 各種`conf`ファイルの調整

必要に応じて各種の設定ファイルを調整する。詳細は3章参照のこと。

(4) 移動体を起動

*詳細は用いる移動体のマニュアルを参照。

(5) 各モジュールの起動

【Windowsの場合】

`Operator`フォルダに含まれる、

```
SimpleFormationGUI.bat, SimpleJoyPad.bat,
```

```
FormationCenterComp.exe, MultiRobotMuxComp.exe
```

の計4 つを実行する。また、各`robot`フォルダに含まれる

```
FormationControllerComp.exe
```

を実行する。

【Ubuntu の場合】

`Operator`フォルダに含まれる、

```
SimpleFormationGUI.sh, SimpleJoyPad.sh,
```

```
FormationCenterComp, MultiRobotMuxComp
```

の計4 つを実行する。また、各`robot`フォルダに含まれる

```
FormationControllerComp.
```

も実行する。

(6) モジュールを接続する

図2のように接続を行い、アクティベートする。

4.3 使用方法

モジュールを接続・Activateすると以下に示すようなGUIが表示される。図12の手順に従って操縦するグループを決定し、SimpleJoyPadでLeaderへの入力を与えると移動が開始される。

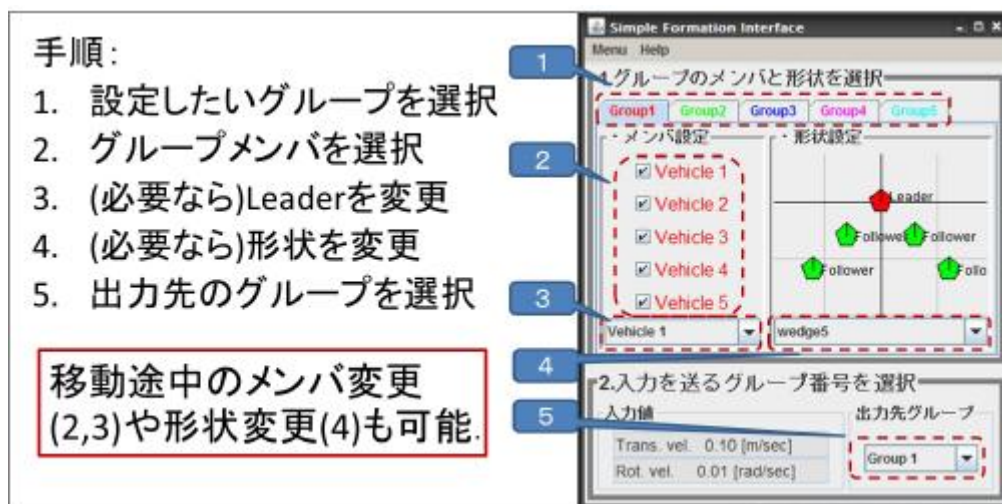


図12 SimpleFormationGUIでの設定手順