

Choreonoid での
MobileRobotNavigationFramework
操作マニュアル

名城大学メカトロニクス工学科
ロボットシステムデザイン研究室

2016 年 12 月 7 日

内容

1.	はじめに	2
1.1	目的.....	2
1.2	本書を読むにあたって.....	2
1.3	動作環境	3
1.4	開発環境	3
2.	シミュレーション, 実機操作の準備	3
2.1	MobileRobotNavigationFramework のダウンロード	3
2.2	dialout にユーザーの追加	3
2.3	VelocityConverterRTC のコンパイル.....	4
2.4	PoseConverterRTC のコンパイル	4
3.	シミュレータを用いた移動ロボットの操作方法.....	5
3.1	eclipse の起動	5
3.2	Choreonoid の起動.....	7
3.3	シミュレータ操作コンポーネントの実行.....	10
3.4	コンポーネント接続.....	11
3.5	シミュレーションの実行.....	12

1. はじめに

1.1 目的

本マニュアルは，早稲田大学尾形研究室で開発された移動ロボット用フレームワークである `MobileRobotNavigationFramework` を動力学シミュレータである `Choreonoid` でのシミュレーション動作を行うための動作手順を示したものである．

1.2 本書を読むにあたって

`OpenRTM` や `Choreonoid` をインストールしておくこと．

`OpenRTM` のインストールに関しては `OpenRTM` の公式 HP からインストールしてください．もしくは，今回の動作環境をインストールする環境のマニュアルをして公開しておりますので，どちらかを参照してインストールを行ってください．

`OpenRTM` 環境構築マニュアル

<http://www2.meijo-u.ac.jp/~kohara/cms/technicalreport/mobilerobotnavigationframework-table-of-contents>

`Choreonoid` のインストールは，

<https://github.com/rsdlab/CRANE-simulation> から Download してきた中の Documents にある `Choreonoid` 導入マニュアルを参照してください．

1.3 動作環境

本 RTC の動作確認環境を以下に示す.

OS	Ubuntu14.04
RT ミドルウェア	OpenRTM-aist-1.1.2
Choreonoid	Choreonoid-1.5.0

※Choreonoid の動作検証には, Panasonic Let's Note(CF-MX3)で行った結果, Choreonoid が落ちてしまう現象が確認されたので, 高スペック(高性能グラフィックボード搭載等)のデスクトップパソコンでの動作を推奨しています.

1.4 開発環境

本 RTC の開発環境を以下に示す.

OS	Ubuntu14.04
RT ミドルウェア	OpenRTM-aist-1.1.2
Choreonoid	Choreonoid-1.5.0

2. シミュレーション, 実機操作の準備

2.1 MobileRobotNavigationFramework のダウンロード

MobileRobotNavigationFramework は sugarsweetrobotics から MobileRobotNabigationFramework_dist を github することで利用することできるほか, MobileRobotNavigationFramework を利用するために必要なソフトウェアのインストール手順やソース単位でコンポーネントをコンパイルして実行するための手順をまとめたマニュアルも先ほどの“OpenRTM の環境構築マニュアル”と同じページで公開しているのでそれらを参考に MobileRobotNavigationFramework が利用できる環境を整備してください.

2.2 dialout にユーザーの追加

実機のロボット用コンポーネントを用いる際には, デフォルトの設定では root 権限で実行しなければコンポーネントを実行することができない. そのため, dialout にユーザーを登録しておくと, コンポーネントを起動する際に, 毎回 root 権限で実行しなくてもよくなる. 以下にコマンドを示す.

```
$sudo gpasswd -a ユーザー名 dialout
```

2.3 VelocityConverterRTC のコンパイル

github からダウンロードしてきたディレクトリ内にある VelocityConverterRTC のコンパイル方法を以下に示す. VelocityConverterRTC は, SFMLJoystick の速度入力用データポートのデータ型である TimedVelocity2D 型を Choreonoid のモデルに対応する TimedDoubleSeq 型に変換するためのコンポーネントである. コンパイルは以下の手順で行う.

```
$cd VelocityConverterRTC
$mkdir build
$cd build
$cmake ..
$make
```

2.4 PoseConverterRTC のコンパイル

github からダウンロードしてきたディレクトリ内にある PoseConverterRTC のコンパイル方法を以下に示す. PoseConverterRTC は, Choreonoid から TimedPose3D 型で出力される位置座標データを MobileRobotNavigationFramework に対応する TimedPose2D 型に変換するためのコンポーネントである. コンパイルは以下の手順で行う.

```
$cd VelocityConverterRTC
$mkdir build
$cd build
$cmake ..
$make
```

3. シミュレータを用いた移動ロボットの操作方法

3.1 eclipse の起動

①ネーミングサービスを実行する

```
$rtm-naming
```

②eclipse を起動する.

③eclipse が立ち上がったたら, 右上のその他から RT System Editor を選択する. (Fig.1~3)

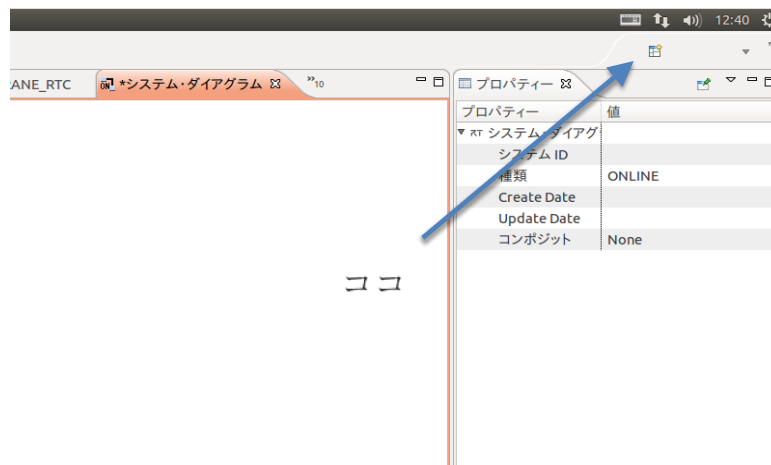


Fig.1 Eclipse



Fig.2 その他

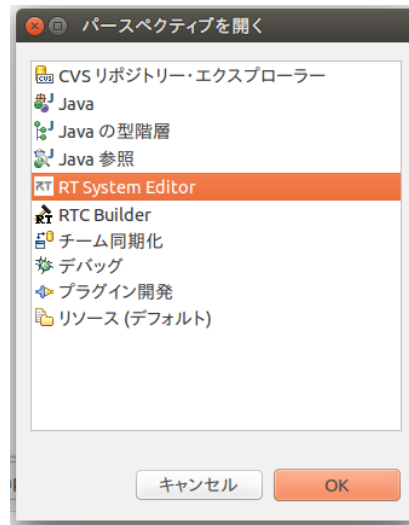


Fig.3 RT System Editor

⑤ネームサービスに何も無い場合，Fig.4 のようにネームサーバーの追加から localhost を追加する．（このとき失敗する場合ネーミングサービスが立ち上がっていない→3 章の手順を再び行ってください．）

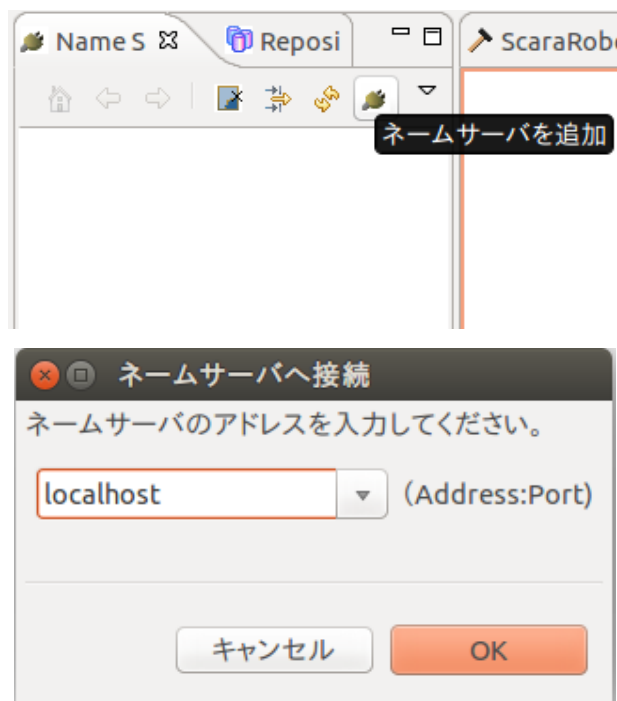


Fig.4 ネームサーバーの追加

3.2 Choreonoid の起動

①Choreonoid を起動する

```
$cd choreonoid-1.5.0/build/bin/  
$./choreonoid
```

Choreonoid のファイル→新規からプロジェクトの読み込みを選択し, Kobuki_simulation/model_project 内の kobuki-TurtleBot.cnoid を読み込む(Fig.5, 6)

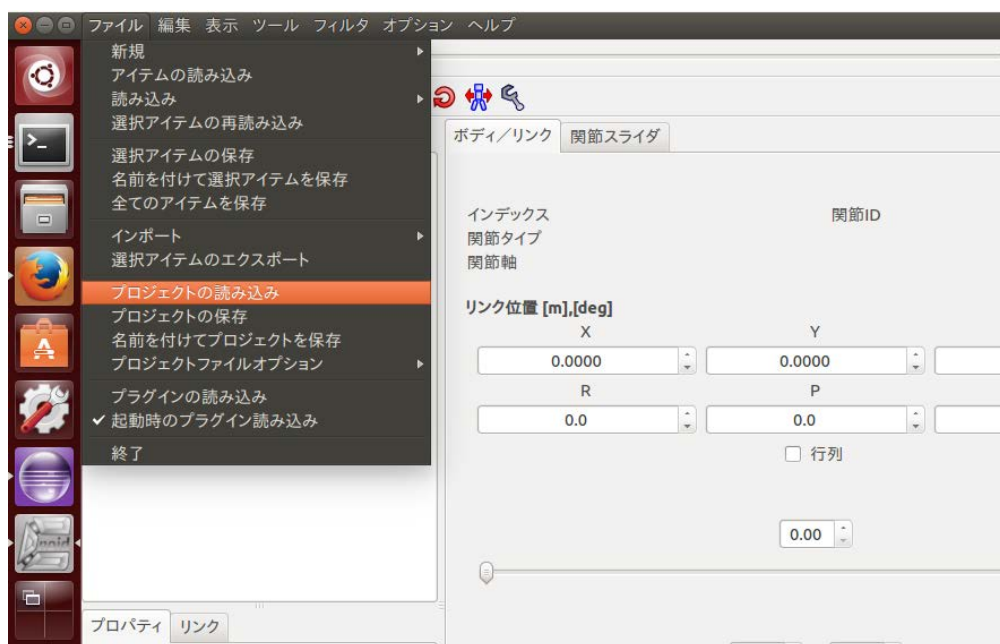


Fig.5 プロジェクトの読み込み手順

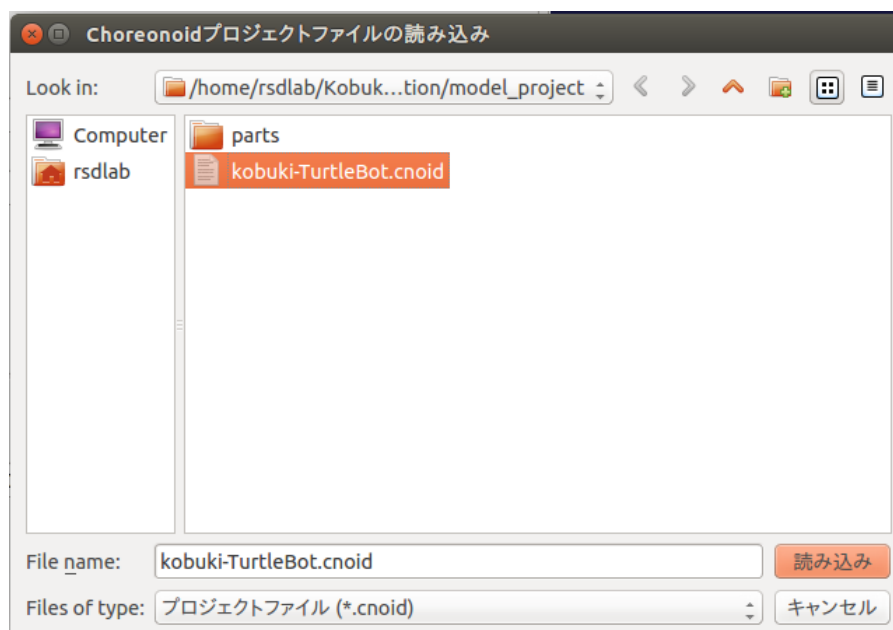


Fig.6 プロジェクトの読み込み

アイテムツリーが以下の Fig.7 のように表示されるのを確認する。

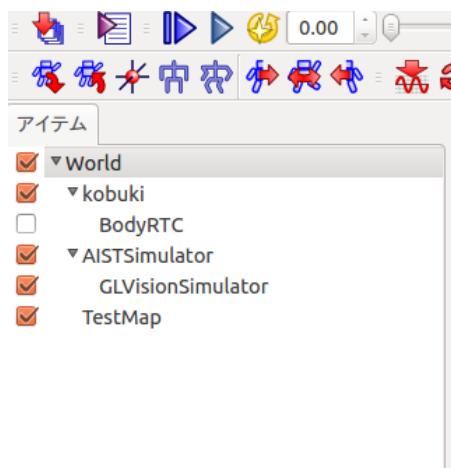


Fig.7 ItemTree

このとき，Choreonoid 上には以下のようなモデルが生成されているのを確認する．(Fig.8
左：移動ロボットモデル，右：TestMap)

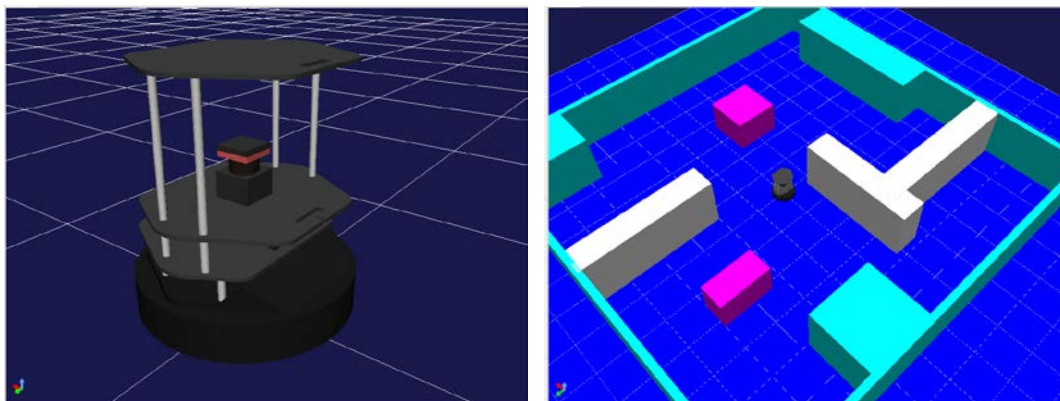


Fig.8 Model

Eclipse のネームサーバーに kobukiComp が実行されていることを確認する．(Fig.8)

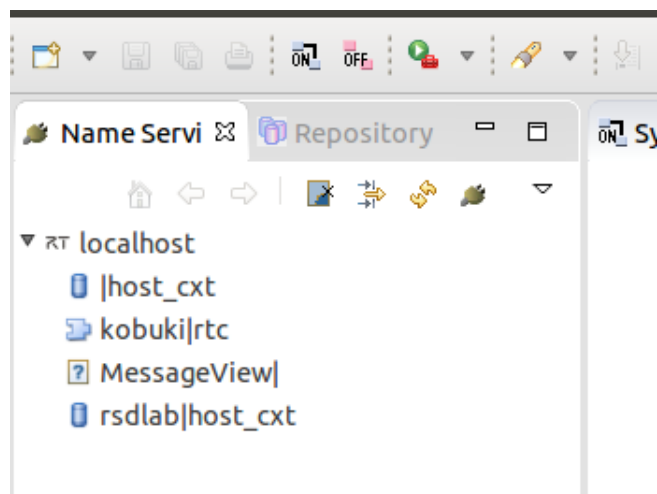


Fig.8 ネームサーバーの kobuki|rtc の確認

3.3 シミュレータ操作コンポーネントの実行

シミュレータを実行する上で必要となるコンポーネントを示す.

- SFMLJoystick
- SFMLJoystickToVelocity
- MapperMRPT
- NavigationManager
- VelocityConverterRTC
- PoseConverterRTC

使用するコンポーネントの中で上からの 4 つのコンポーネントに関しては, MobileRobotNavigationFramework の中にあるので, それらを実行する.

実行方法のマニュアルは OpenRTM のインストール方法と同じ HP に記載してある.

<http://www2.meijo-u.ac.jp/~kohara/cms/technicalreport/mobilerobotnavigationframework-table-of-contents/mobile-robot-navigation-framework-utilization/mapping-components-execution>

上の URL にあるページを参照の上, 実行してください.

また, VelocityConverterRTC と PoseConverterRTC の実行手順を以下に示す.

◆ VelocityConverterRTC

```
$cd VelocityConverterRTC/build/src  
$./VelocityConverterRTCComp
```

◆ PoseConverterRTC

```
$cd PoseConverterRTC/build/src  
$./PoseConverterRTCComp
```

3.4 コンポーネント接続

Choreonoid のプロジェクトファイルの読み込みと必要な RTC の実行できたら, Eclipse の RTSystemEditor でシステムダイアグラムを ON にし, コンポーネントをシステムダイアグラムに表示し, 以下のように接続する. (Fig.9)

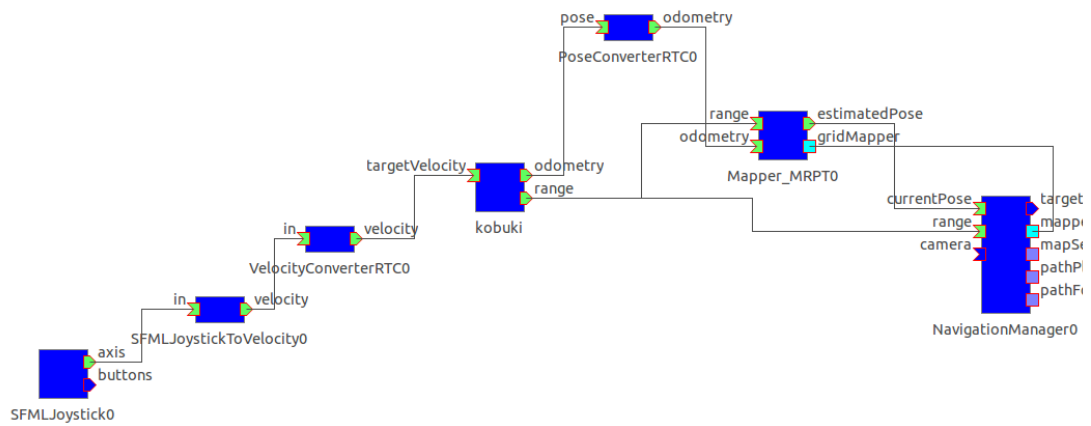


Fig.9 環境地図生成コンポーネント群構築のための RT システム

3.5 シミュレーションの実行

Choreonoid のシミュレーションの実行をクリックする。(Fig.10)



Fig.10 シミュレーションの開始

シミュレーションが開始されると Choreonoid モデルコンポーネントが自動的に Activate される。

その後、すべてのコンポーネントを Activate 状態にすると、Mapping のための GUI が起動した状態になり、その画面上には移動ロボットの位置とレーザレンジファインダからのデータが表示される。GUI 画面の左上にある“Start Mapping”ボタンをクリックすることで地図生成を行っていくことができる。(Fig.11)

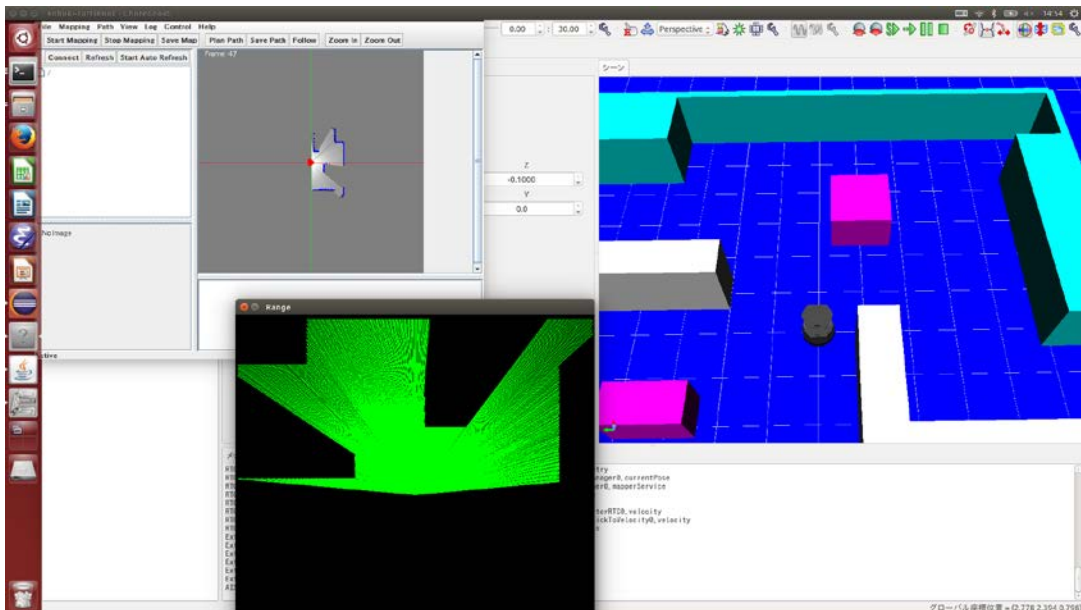


Fig11 Strat Mapping

環境地図生成が完了したら、GUI の左上にある“Save Map”のボタンをクリックすることで作成した環境地図を保存する。なお、取得される Map データの保存は以下の二つのファイルである。

- **xxx.png**：ビットマップ画像データを扱うファイル
- **xxx.yaml**：マップの原点やピクセル/長さ比の情報データを扱うファイル

xxx の部分には、各自が決めたファイル名で保存すること。

(例) Map_Data.png

yaml ファイルは、png ファイルと同じファイル名となる。

TestMap で Mapping を行った結果を以下に示す。

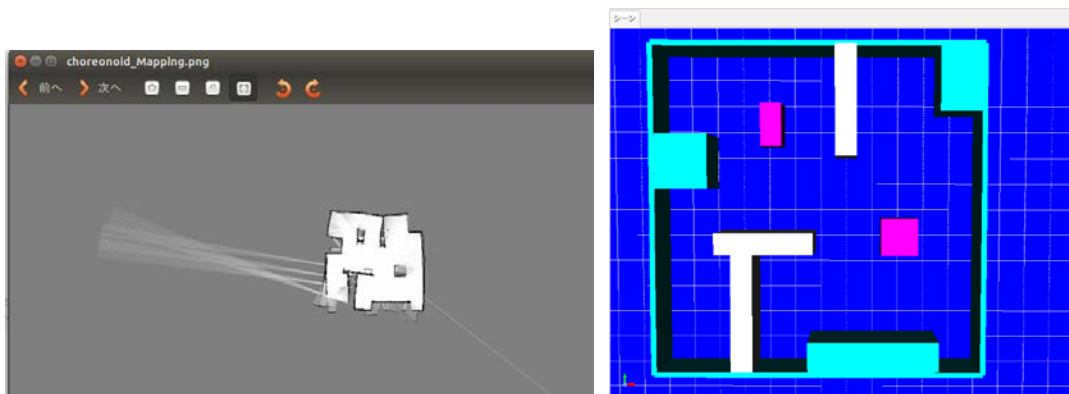


Fig.12 Mapping

読み込んだプロジェクトには、テスト用の Map をデフォルトで入れてあるが、ほかの Map での検証を行いたい場合には、アイテムツリー内の TestMap を消して新たに Map 用の wrl を読み込ませることで Map を変更することができる。

wrl を読み込ませるときは、word を選択した状態で、「読み込み」→「OpenHRP モデルファイル」から対象の wrl を追加することができる。