

ユーザーズマニュアル

ロボット移動軌跡表示システム

発行日 2018年3月28日
公立大学法人会津大学
株式会社東日本計算センター

目次

1	はじめに	1
1.1	前提知識.....	1
1.2	システム概要.....	2
1.3	動作環境.....	4
1.3.1	GUI RTC	4
1.3.2	RDB RTC	5
1.3.3	二次 DB ファイル生成ツール.....	6
1.3.4	Choreonoid 入力ファイル作成ツール.....	6
2	インストール.....	7
2.1	GUI RTC	7
2.2	RDB RTC.....	8
2.3	二次 DB ファイル生成ツール.....	9
2.4	Choreonoid 入力ファイル作成ツール	10
3	基本設定	11
3.1	GUI RTC	11
3.2	RDB RTC.....	11
3.3	RTC 同士を接続する	12
3.4	二次 DB ファイル生成ツール.....	14
3.5	Choreonoid 入力ファイル作成ツール	15
4	操作	16
4.1	GUI RTC	16
4.2	RDB RTC.....	17
4.3	二次 DB ファイル生成ツール.....	18
4.4	Choreonoid 入力ファイル作成ツール	19
4.5	Choreonoid ビューア	20
5	出力されるメッセージ	21
5.1	GUI RTC	21
5.2	RDB RTC.....	21
5.3	二次 DB ファイル生成ツール.....	22
5.4	Choreonoid 入力ファイル作成ツール	22
6	ログファイル.....	23
6.1	GUI RTC	23
6.2	RDB RTC.....	23

1 はじめに

1.1 前提知識

国立研究開発法人産業技術総合研究所(以下、産総研)が開発・配布している RT ミドルウェア”OpenRTM-aist”、およびロボット用統合 GUI ソフトウェア”Choreonoid(コレオノイド)”に関する基礎知識が必要です。

このユーザーズマニュアルは、下記の知識がある前提で記述されています。

- 1 OS(Linux/Windows)の基本設定ができる
- 2 Choreonoid のインストールと基本設定、および操作ができる¹
- 3 OpenRTM-aist Python のインストールと基本設定ができる
- 4 omniORB の基本設定ができる
- 5 RTC 同士のポートを接続して動作させることができる

※Choreonoid は「不定間隔時系列データのインポート機能」に対応している version をインストールして下さい。²

※Choreonoid ビューアに表示する 3D MAP およびロボット 3D モデルは別途用意して下さい。

¹ Choreonoid の詳しい操作方法については以下マニュアルを参照して下さい。

<http://choreonoid.org/ja/manuals/latest/index.html>

² 本マニュアル執筆時点ではこの機能に対応したリリース版は無いため、開発版を入手し使用して下さい。詳細は以下の URL を参照して下さい。

<http://choreonoid.org/ja/manuals/1.5/install/build-ubuntu.html#id11>

1.2 システム概要

本システムではクラウド Server 上に構築された一次 DB(MongoDB)から任意のロボットのセンサーデータを取得し、クライアント PC 上に二次 DB を構成します。構成した二次 DB のデータを基に、Choreonoid にインポートできるファイル(不定間隔時系列の標準ボディモーションファイル)の作成を行います。尚、ファイル作成は表示するロボット毎(1 ロボット 1 ファイル)に行います。

また、本システムは一次 DB にロボットセンサーデータが既に入力されていることを前提としています。

以下にシステム概要図を記載します。

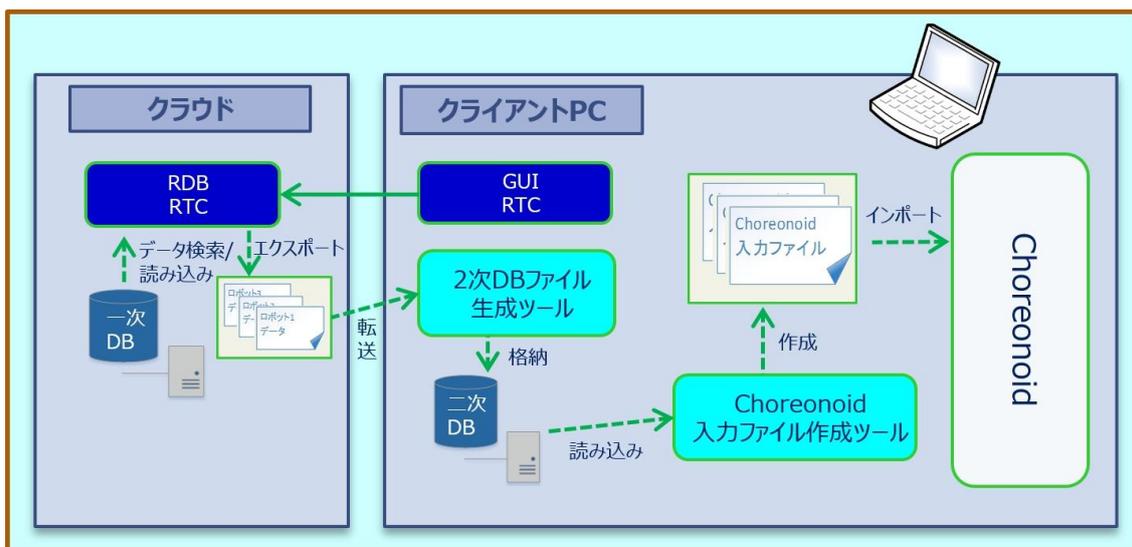


図 1 システム概要図

一次 DB からのセンサーデータ取得

クライアント PC から GUI 操作でクラウド内の DB RTC に一次 DB から取得したいロボットの情報を指定し、取得を行います。取得した情報は CSV 形式のファイルにエクスポートし、クラウドの所定の場所に保存します。

二次 DB の生成

クライアント PC からコンソールを通じてクラウド上の CSV ファイルを取得し、データの補正・変換を行い再度 CSV ファイルを作成します。ロボット毎に作成したファイルで二次 DB を構成します。

Choreonoid 入力ファイル変換

二次 DB のファイル(CSV 形式)を Choreonoid にインポートできる標準ボディモーションファイル(YAML 形式)に変換します。

Choreonoid ビューア表示

Choreonoid を起動し、3D MAP・ロボット 3D モデルを表示し所定の位置に配置し、作成した標準ボディモーションファイルをインポートし、軌跡表示を行います。

1.3 動作環境

1.3.1 GUI RTC

当 RTC は Windows で動作します。必須動作環境を以下に記載します。

表 1 ハードウェア必須動作環境 (Windows)

項目	必要条件	動作確認済み環境
対応 OS	Windows	8.1 pro
CPU	デュアルコア以上	Core i7 4710MQ
メモリ	512MB 以上	2.5GHz / 4 コア HT
補助記憶装置	OS インストール済みのドライブに 1MByte 以上の空き容量が必要 ³	HDD 910GB
ディスプレイ	解像度 800x600 以上 16bit カラー推奨	1024 x 768 32bit カラー
その他	python が動作する	Python2.7

当 RTC を動作させるための前提条件

(ア) OS の基本設定が完了していること

(イ) “OpenRTM-aist Python 1.1.2” がインストールされており正常に動作すること

OpenRTM-aist Python 1.1.2 のインストール方法は、OpenRTM-aist 公式 Web サイト⁴ を参照してください。

³ ログを出力する場合は更に空き領域が必要です

⁴ <http://www.openrtm.org/openrtm/ja/node/798>

1.3.2 RDB RTC

当 RTC は Linux で動作します。必須動作環境を以下に記載します。

表 2 ハードウェア必須動作環境 (Linux)

項目	必要条件	動作確認済み環境
対応 OS	Linux	Ubuntu 16.04 LTS
CPU	デュアルコア以上	Core i7 4710MQ
メモリ	512MB 以上	2.5GHz / 4 コア HT
補助記憶装置	OS インストール済みのドライブに 1MByte 以上の空き容量が必要 ⁵	HDD 32GB
ディスプレイ	解像度 800x600 以上 16bit カラー推奨	1024 x 768 32bit カラー
その他	python が動作する	Python2.7

当 RTC を動作させるための前提条件

(ア) OS の基本設定が完了していること

(イ) “OpenRTM-aist Python 1.1.2” がインストールされており正常に動作すること

OpenRTM-aist Python 1.1.2 のインストール方法は、OpenRTM-aist 公式 Web サイト⁶を参照してください。

⁵ ログを出力する場合は更に空き領域が必要です

⁶ <http://www.openrtm.org/openrtm/ja/node/798>

1.3.3 二次 DB ファイル生成ツール

当ツールは Linux で動作します。必須動作環境を以下に記載します。

表 3 ハードウェア必須動作環境 (Linux)

項目	必要条件	動作確認済み環境
対応 OS	Linux	Ubuntu 16.04 LTS
CPU	デュアルコア以上	Core i7 4710MQ
メモリ	512MB 以上	2.5GHz / 4 コア HT
補助記憶装置	OS インストール済みのドライブに 1MByte 以上の空き容量が必要	HDD 32GB
ディスプレイ	解像度 800x600 以上 16bit カラー推奨	1024 x 768 32bit カラー
その他	python が動作する	Python2.7

1.3.4 Choreonoid 入力ファイル作成ツール

当ツールは Linux で動作します。必須動作環境を以下に記載します。

表 4 ハードウェア必須動作環境 (Linux)

項目	必要条件	動作確認済み環境
対応 OS	Linux	Ubuntu 16.04 LTS
CPU	デュアルコア以上	Core i7 4710MQ
メモリ	512MB 以上	2.5GHz / 4 コア HT
補助記憶装置	OS インストール済みのドライブに 1MByte 以上の空き容量が必要	HDD 32GB
ディスプレイ	解像度 800x600 以上 16bit カラー推奨	1024 x 768 32bit カラー
その他	python が動作する	Python2.7

2 インストール

2.1 GUI RTC

ファイルの展開

ファイルを任意の場所に展開してください。

展開されたファイルのうち、動作に必要なファイルは下記のとおりです。

表 5 動作に必要なファイル

フォルダ名	ファイル名
RobotMoveLocusRDBGui	RobotMoveLocusRDBGui.conf RobotMoveLocusRDBGui.py RobotMoveLocusRDBGuiDesign.py rtc.conf

“*.py” ファイルに実行権限が与えられていることを確認してください。

ライブラリのインストール

当 RTC は以下の python ライブラリを使用します。別途インストールして下さい。

インストール方法や詳細は各ライブラリのマニュアル等を参照して下さい。

表 6 動作に必要な python ライブラリ

ライブラリ名	動作確認済みの version
Pyqt	4.11.4

2.2 RDB RTC

ファイルの展開

ファイルを任意の場所に展開してください。

展開されたファイルのうち、動作に必要なファイルは下記のとおりです。

表 7 動作に必要なファイル

フォルダ名	ファイル名
RobotMoveLocusRDB	RobotMoveLocusRDB.conf RobotMoveLocusRDB.py rtc.conf

“*.py” ファイルに実行権限が与えられていることを確認してください。

ライブラリのインストール

当 RTC は以下の python ライブラリを使用します。別途インストールして下さい。インストール方法や詳細は各ライブラリのマニュアル等を参照して下さい。

表 8 動作に必要な python ライブラリ

ライブラリ名	動作確認済みの version
pymongo	3.3.0
python-dateutil	1.5

2.3 二次 DB ファイル生成ツール

ファイルの展開

ファイルを任意の場所に展開してください。

展開されたファイルのうち、動作に必要なファイルは下記のとおりです。

表 9 動作に必要なファイル

フォルダ名	ファイル名
2DBFileCreate	2DBFileCreate.py completion_1DBcsv.py conf.ini consts.py Convert_yaml_parameter.py drone_distance_cal.py

“*.py” ファイルに実行権限が与えられていることを確認してください。

ライブラリのインストール

以下の python ライブラリを使用します。別途インストールして下さい。インストール方法や詳細は各ライブラリのマニュアル等を参照して下さい。

表 10 動作に必要な python ライブラリ

ライブラリ名	動作確認済みの version
pandas	0.17.1
numpy	1.11
configparser	3.5.0

2.4 Choreonoid 入力ファイル作成ツール

ファイルの展開

ファイルを任意の場所に展開してください。

展開されたファイルのうち、動作に必要なファイルは下記のとおりです。

表 11 動作に必要なファイル

フォルダ名	ファイル名
ChoreonoidInputFileCreate	ChoreonoidInputFileCreate.py consts.py

“*.py” ファイルに実行権限が与えられていることを確認してください。

ライブラリのインストール

以下の python ライブラリを使用します。別途インストールして下さい。インストール方法や詳細は各ライブラリのマニュアル等を参照して下さい。

表 12 動作に必要な python ライブラリ

ライブラリ名	動作確認済みの version
pandas	0.17.1

3 基本設定

本マニュアルでは、RTC 独自の設定項目とツールの設定項目について説明します。

3.1 GUI RTC

独自の設定項目はありません。

3.2 RDB RTC

RTC 起動時に下記のファイルが読み込まれ、RTC の振る舞いが決定します。

1. “RobotMoveLocusRDB.conf”

上記ファイルをテキストエディタで開き、環境に沿った設定を行って下さい。

※System Editor 上での設定も可能です。(4.2 RDB RTC 章参照)

表 13 “RobotMoveLocusRDB.conf” の設定項目

設定項目	意味
conf.default.export_path	CSV ファイルをエクスポートするフォルダの指定
conf.default.robot_GPS_key	MongoDB からロボットの GPS データを検索する際の検索 Key の指定
conf.default.robot_gyro_key	MongoDB からロボットの gyro データを検索する際の検索 Key の指定

3.3 RTC 同士を接続する

1. クライアント PC 側で "RobotMoveLocusRDBGui.py" を実行します。
2. クラウド Server 側で "RobotMoveLocusRDB.py" を実行します。⁷
3. "RT System Editor" を使用して、RTC 同士を接続してください(図 2)。⁸

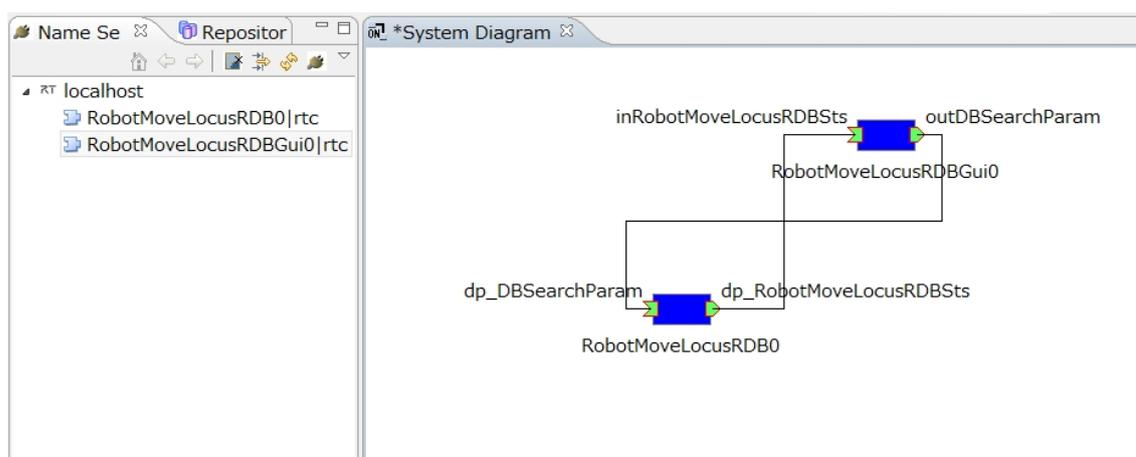


図 2 RT System Editor によって RTC を接続した状態

1. RobotMoveLocusRDB の [dp_DBSerchParam] ポートを RobotMoveLocusRDBGui の [outDBSerchParam] ポートに接続します。
2. RobotMoveLocusRDB の [dp_RobotMoveLocusRDBSts] ポートを RobotMoveLocusRDBGui の [inRobotMoveLocusRDBSts] に接続します。

⁷ クラウド側でのコンポーネントの起動には、事前にクラウド管理者への連絡が必要です。

⁸ "RTShell"を使用する方法もあります。詳細は OpenRTM-aist 公式 Web サイトを参照してください。 <http://www.openrtm.org/openrtm/ja/node/869>

表 14 各 RTC のデータポート名

RTC 名	データポート名
RobotMoveLocusRDB	dp_DBSerchParam
	dp_RobotMoveLocusRDBSts
RobotMoveLocusRDBGui	outDBSerchParam
	inRobotMoveLocusRDBSts

ポートを接続したら、すべての RTC を "Activate" します。
 "RT System Editor" を右クリックして "All Activate" を選択します(図 3)。
 "Activate" 状態になると、RTC の色が青から緑に変わります。

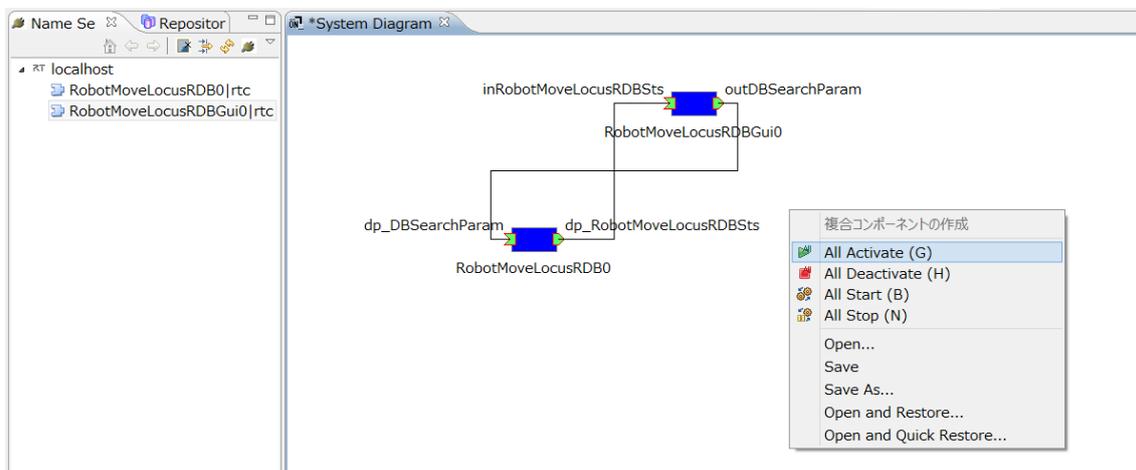


図 3 RTC を Activate する

3.4 二次 DB ファイル生成ツール

ツール実行時に以下のファイルが読み込まれます。

1. conf.ini

上記ファイルをテキストエディタで開き、環境に沿った設定を行って下さい。

設定項目

[base_gps_position]

コレオノイドビューア上で表示する 3D MAP の基準点となる地点の座標を指定します。ここで指定した値を基に、ロボットの位置情報をコレオノイドで使用する形式に変換します。

表 15 base_gps_position 指定内容

項目	内容
lat	基準点となる地点の緯度を指定(十進法度単位)
lng	基準点となる地点の経度を指定(十進法度単位)
alt	基準点となる地点の高度を指定(m)

[gyro_def_position]

コレオノイドビューア上で表示するロボットの角度の初期値を指定します。
ここで指定した値を、gyro 情報を持っていないロボット、または欠損しているデータの初期値として使用します。

表 16 gyro_def_position 指定内容

項目	内容
roll	roll 角を指定(rad)
pitch	pitch 角を指定(rad)
yaw	yaw 角を指定(rad)

conf.ini 記載例

```
[DEFAULT]

[base_gps_position]
lat = 35.681301
lng = 139.767066
alt = 2.0

[gyro_def_position]
roll = 0.0
pitch = 0.0
yaw = 1.57
```

上記例では基準点を[北緯 35.681301 度/東経 139.767066/高度 2.0m]とし、
角度の初期値を[roll 角 0.0 rad/pitch 角 0.0 rad/yaw 角 1.57 rad]に設定して
います。

3.5 Choreonoid 入力ファイル作成ツール

独自の設定項目はありません。

4 操作

4.1 GUI RTC

GUI で一次 DB からデータを検索するパラメータ (DB Collection、RobotID、開始/終了日時) を設定し、Start ボタンを押すことで RDB RTC に通知され、エクスポートが開始されます。

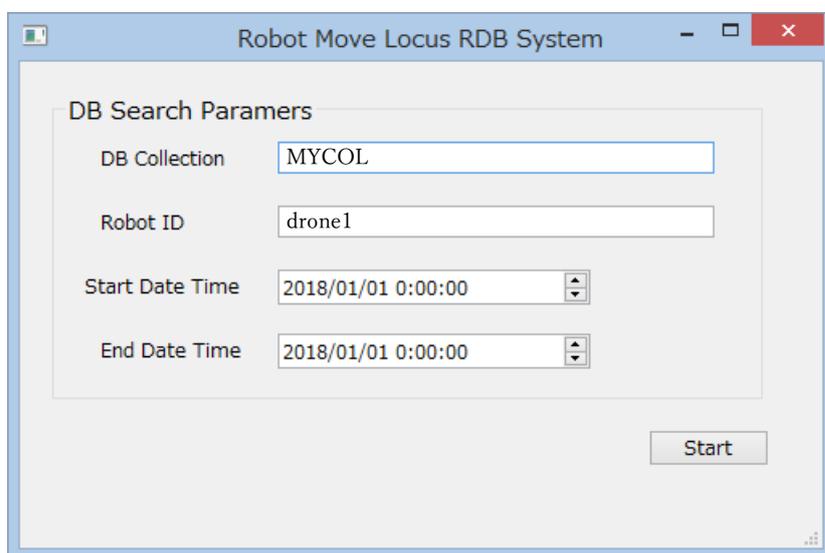


図 4 パラメータ指定 GUI

データのエクスポートが正常に完了した場合、クラウド Server 上の所定のフォルダに CSV ファイルが作成されます。これを対象となるロボットの台数分を行います。全ての作成が完了したら、クライアント PC にエクスポートファイルをダウンロードし、二次 DB ファイルの生成を行って下さい。(4.3 章)

4.2 RDB RTC

3.2 章でコンフィグファイルの設定を行っている場合はここでの操作は不要です。設定を変更したい場合は、System Editor 上で RDB RTC のコンフィグ編集ウィンドウ(図 5)を開き、設定を行って下さい。

※export_path を変更する場合は、一度 RTC を Deactivate してから再度 Activate して下さい。フォルダが生成されます。

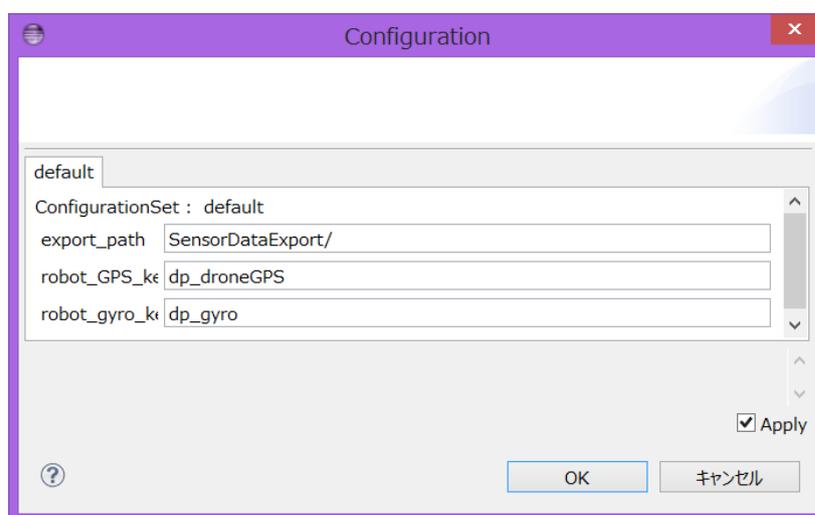


図 5 コンフィグ編集ウィンドウ

4.3 二次 DB ファイル生成ツール

コンソールを開き実行ファイルが保存してあるフォルダに移動し、以下のコマンドを実行して下さい。

```
$ python 2DBFileCreate.py ファイル名 [--mode ] [--rate ]
```

- ファイル名：2次DB生成対象となるCSVファイルの場所と名前。
例： ./input/MYCOL_drone1.csv
- [--mode]：不正値の補完方法を指定するオプション。
指定が無い場合はデフォルト設定(破棄)。

表 17 [--mode]オプション指定

オプション指定	内容
--mode pad	不正値を直前データ穴埋め
--mode linear	不正値を線形補間
指定なし	不正値を破棄

- [--rate]：2次DBファイルデータのダウンサンプリングを指定するオプション。指定が無い場合はダウンサンプリングなし。

表 18 [--rate]オプション指定

オプション指定	内容
--rate 整数	指定の値でダウンサンプリングを行う。 例:データ量を 1/2 にダウンサンプリングしたい場合は"2"を指定
指定なし	ダウンサンプリングなし

ファイルの作成が正常に完了した場合、以下のフォルダにファイルが作成されま
す。

./2DBFileCreate/export2DB/(作成日フォルダ)/ファイル名

作成されたファイルを二次DB格納場所に保存し、Chreonoid入力ファイルの
作成を行って下さい。(4.4章)

4.4 Choreonoid 入力ファイル作成ツール

コンソールを開き実行ファイルが保存してあるフォルダに移動し、以下のコマンドを実行して下さい。

```
$ python ChoreonoidInputFileCreate.py ファイル名
```

- ファイル名：二次 DB ファイル生成ツールで作成した CSV ファイルの場所と名前

例： `./RobotMoveLocus2DB/converted-sensordata_MYCOL_drone1.csv`

ファイルの作成が正常に完了した場合、以下のフォルダにファイルが作成されません。

`./ChoreonoidInputFileCreate/timed_frame_body_motion/(作成日フォルダ)/`
ファイル名

ファイルの作成が完了したら、Choreonoid を起動しファイルのインポートを行って下さい。(4.5 章)

4.5 Choreonoid ビューア

Choreonoid を起動後、3D MAP とロボット 3D モデルをビューアに表示します。

本システムではロボットの GPS 位置情報を Choreonoid で表示する為に距離情報に変換している為、3D MAP とロボットを重ねて表示する際には下記のように 3D MAP を配置する必要があります。

1. 3D MAP 上の基準点となる場所を Choreonoid の(X=0,Y=0,Z=0)となるように MAP を配置する。(基準点：3.4 章で設定した座標)
2. MAP の向きは、X 軸は緯線、Y 軸は経線とそれぞれ平行となるように配置する。
X 軸は東方向に対しての軸を正、西方向に対して負とする。
Y 軸は北方向に対しての軸を正、南方向に対して負とする。

作成した Choreonoid 入力ファイルをロボット毎にインポートし、再生を開始して下さい。ロボットの移動軌跡の表示が行われます。

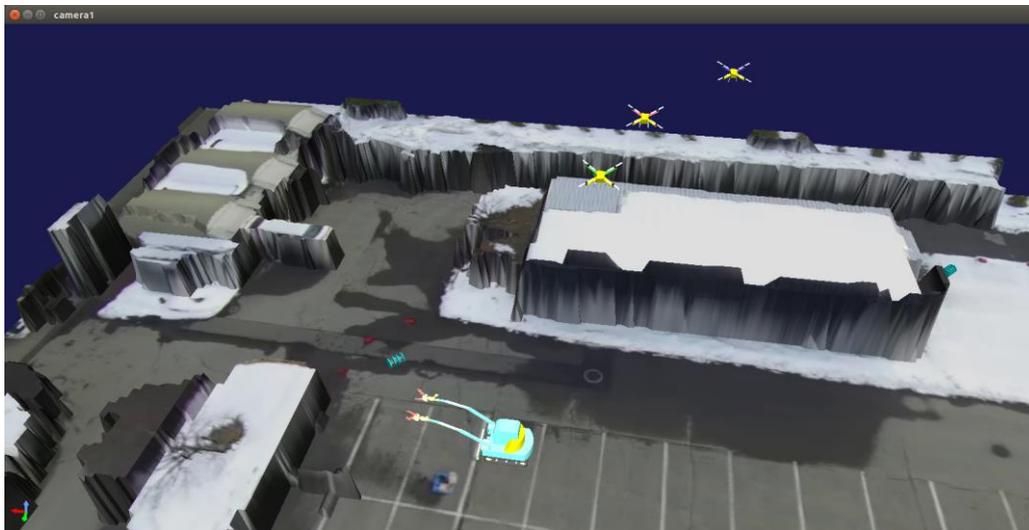


図 6 Choreonoid ビューア画面

5 出力されるメッセージ

発生した障害の原因を調査するには、下記が出力するメッセージを確認する必要があります。

- OS
- OpenRTM-aist
- omniORB
- RTC
- ツール

この章では本システムで使用される RTC・ツールで独自に出力されるメッセージについて紹介します。その他のメッセージに関してはそれぞれのマニュアル等を参照してください。

5.1 GUI RTC

表 19 エラーメッセージ(GUI RTC)

No	状態	エラーメッセージ
1	DB Collection 未入力で start ボタンを押下した時	[warning]DB Collection is not entered!!
2	Robot ID 未入力で start ボタンを押下した時	[warning]Robot ID is not entered!!
3	終了日時より大きい開始日時を指定した時	[warning]Start date and time > End date and time!!

5.2 RDB RTC

表 20 エラーメッセージ(RDB RTC)

No	状態	エラーメッセージ
1	検索条件のデータ入力ミス検出	Search conditions are wrong. Please confirm.
2	検索条件を満たすデータがデータベースに無い場合	There is no data that meet the search conditions.

5.3 二次 DB ファイル生成ツール

表 21 エラーメッセージ(二次 DB ファイル生成ツール)

No	状態	エラーメッセージ
1	CSV ファイル読み込みエラー	[warning]csv file open error
2	コマンドライン引数指定エラー ※指定した値が不正	[warning]invalid command line parameter. Please confirm.
3	コンフィグ指定エラー ※指定した値の型が不正、または範囲外の値	[warning]conf.ini (GPS) ValueError. ->Please confirm conf.ini

5.4 Choreonoid 入力ファイル作成ツール

表 22 エラーメッセージ(Choreonoid 入力ファイル作成ツール)

No	状態	エラーメッセージ
1	CSV ファイル読み込みエラー	[warning]The specified csv file is missing!!
2	コマンドライン引数指定エラー	[warning]Command line arguments are missing!!

6 ログファイル

本システムで使用する RTC のメッセージは以下のログファイルに出力されています。

6.1 GUI RTC

RobotMoveLocusRDBGui.py と同じディレクトリに “rtc<プロセス ID>.log” ファイルが作成されます。

6.2 RDB RTC

RobotMoveLocusRDB.py と同じディレクトリに “rtc<プロセス ID>.log” ファイルが作成されます。

著作権

本文書の著作権は公立大学法人 会津大学に帰属します。

この文書のライセンスは以下のとおりです。

[クリエイティブ・コモンズ 表示 2.1 日本](http://creativecommons.org/licenses/by/2.1/jp/)

<http://creativecommons.org/licenses/by/2.1/jp/>

