



JAEA-Testing

2021-004

DOI:10.11484/jaea-testing-2021-004

## HAIROWorldPlugin マニュアル (改訂版)

HAIROWorldPlugin Operation Manual (Revised Edition)

鈴木 健太 八代 大 川端 邦明

Kenta SUZUKI, Hiroshi YASHIRO and Kuniaki KAWABATA

福島研究開発部門

福島研究開発拠点

楢葉遠隔技術開発センター

モックアップ試験施設部

Demonstration Test Department

Naraha Center for Remote Control Technology Development

Fukushima Research Institute

Sector of Fukushima Research and Development

March 2022

Japan Atomic Energy Agency

日本原子力研究開発機構

JAEA-Testing

本レポートは国立研究開発法人日本原子力研究開発機構が不定期に発行する成果報告書です。本レポートはクリエイティブ・コモンズ表示 4.0 国際 ライセンスの下に提供されています。本レポートの成果（データを含む）に著作権が発生しない場合でも、同ライセンスと同様の条件で利用してください。（<https://creativecommons.org/licenses/by/4.0/deed.ja>）  
なお、本レポートの全文は日本原子力研究開発機構ウェブサイト（<https://www.jaea.go.jp>）より発信されています。本レポートに関しては下記までお問合せください。

国立研究開発法人日本原子力研究開発機構 JAEA イノベーションハブ 研究成果利活用課  
〒 319-1195 茨城県那珂郡東海村大字白方 2 番地 4  
電話 029-282-6387, Fax 029-282-5920, E-mail:ird-support@jaea.go.jp

This report is issued irregularly by Japan Atomic Energy Agency.

This work is licensed under a Creative Commons Attribution 4.0 International License (<https://creativecommons.org/licenses/by/4.0/deed.en>).

Even if the results of this report (including data) are not copyrighted, they must be used under the same terms and conditions as CC-BY.

For inquiries regarding this report, please contact Institutional Repository and Utilization Section, JAEA Innovation Hub, Japan Atomic Energy Agency.

2-4 Shirakata, Tokai-mura, Naka-gun, Ibaraki-ken 319-1195 Japan

Tel +81-29-282-6387, Fax +81-29-282-5920, E-mail:ird-support@jaea.go.jp

## HAIROWorldPlugin マニュアル (改訂版)

日本原子力研究開発機構 福島研究開発部門 福島研究開発拠点  
檜葉遠隔技術開発センター モックアップ試験施設部

鈴木 健太、八代 大<sup>\*</sup>、川端 邦明

(2021年11月30日受理)

本稿は、日本原子力研究開発機構（以下、原子力機構）が開発した HAIROWorldPlugin の利用手順書（HAIROWorldPlugin マニュアル[JAEA-Testing 2020-009]）の改訂版である。原子力機構檜葉遠隔技術開発センターでは、東京電力ホールディングス株式会社（以下、東京電力 HD）福島第一原子力発電所におけるロボットを用いた遠隔による廃炉作業を支援するための技術として、ロボット等の動力学シミュレーションが可能なオープンソースソフトウェアであるロボット用統合 GUI ソフトウェア Choreonoid を基盤としたロボットシミュレータを開発している。HAIROWorldPlugin は、ロボットの遠隔操作により行われる廃炉作業を模擬するための拡張機能を Choreonoid プラグインとしてまとめたものである。

改訂版の本利用手順書では、最新の HAIROWorldPlugin が廃炉作業時を想定したロボットの遠隔操縦体験を提供するための機能（水中ロボット/無人航空機の挙動模擬、ノイズ・歪み・色相変化等を含む不鮮明なカメラ画像提示、遅延・帯域減少・パケットロス等の通信障害効果の付与）に加えて、新たに追加された機能の設定と操作手順を説明している。

具体的には、シミュレーションをしているロボットと他の物体との間の干渉状態を記録する機能、ジョイスティックの入力状態を表示・記録する機能、環境物体のモデル（パイプ・グレーチング・スロープ）を生成する機能、慣性モーメントを計算する機能、使用頻度の高いシミュレーションの設定を簡易な操作で呼び出す機能、シミュレーションの履歴を記録・表示する機能について説明を追記した。

本稿では、Ubuntu20.04 LTS に Choreonoid を導入して HAIROWorldPlugin をインストールする手順と、HAIROWorldPlugin が提供する機能を利用する際の操作手順やパラメータの設定方法を、Choreonoid を初めて使用するユーザにもわかりやすいように画面のスナップショットを用いて説明している。

---

檜葉遠隔技術開発センター：〒979-0513 福島県双葉郡檜葉町大字山田岡字仲丸 1-22

※ 技術開発協力員

HAIROWorldPlugin Operation Manual (Revised Edition)

Kenta SUZUKI, Hiroshi YASHIRO<sup>✧</sup> and Kuniaki KAWABATA

Demonstration Test Department, Naraha Center for Remote Control Technology Development,  
Fukushima Research Institute, Sector of Fukushima Research and Development  
Japan Atomic Energy Agency  
Naraha-machi, Futaba-gun, Fukushima-ken

(Received November 30, 2021)

This report is updated HAIROWorldPlugin Operation Manual (JAEA-Testing 2020-009). Our motivation is to develop a robot simulator based on Choreonoid for technological development to contribute the decommissioning work at the Fukushima Daiichi Nuclear Power Station of Tokyo Electric Power Company Holdings, Inc. Choreonoid is an open source simulator which calculates the behavior of robots. We are developing HAIROWorldPlugin which is an extended function of Choreonoid for providing a virtual decommissioning experience.

In the latest HAIROWorldPlugin, the several functions were additionally implemented. In particular, we implemented new functionalities: logging a collision status between a robot and an object, showing and logging a status of the joystick input signals, generating an environmental object model (pipe, grating, and slope), bookmarking an often used simulation settings, and recording histories of a simulation settings. In addition, this report describes the installation of the plugin to Choreonoid on Ubuntu20.04-LTS and parameter settings of the plugin by presenting snapshots of operation windows.

Keywords: Robot Simulator, Dynamics Simulation, Choreonoid

---

✧Collaborating Engineer

# 目次

1 HAIROWorldPlugin 入門 .....	1
1-1 はじめに .....	1
1-2 Choreonoid と HAIROWorldPlugin とは .....	5
1-3 Choreonoid のセットアップ .....	6
1-4 HAIROWorldPlugin のセットアップ .....	10
2 ROV/UAV の挙動の模擬 .....	16
2-1 概要 .....	16
2-2 ROV/UAV の挙動の模擬 .....	18
3 通信障害効果の付与 .....	31
3-1 概要 .....	31
3-2 通信障害効果の設定 .....	32
4 カメラ画像への視覚効果の付与 .....	39
4-1 概要 .....	39
4-2 視覚効果の設定 .....	41
5 ロボットの移動軌跡の記録 .....	48
5-1 概要 .....	48
5-2 ロボットの移動軌跡の記録 .....	49
6 ロボットの干渉状態の記録 .....	53
6-1 概要 .....	53
6-2 ロボットの干渉状態の記録 .....	54
6-3 ロボットの干渉状態の保存 .....	58
6-4 ロボットの干渉状態のグラフ表示 .....	59
7 ジョイスティックの入力状態の表示 .....	61
7-1 概要 .....	61
7-2 ジョイスティックの入力状態の表示 .....	62
8 ジョイスティックの入力状態の記録 .....	65
8-1 概要 .....	65
8-2 ジョイスティックの入力状態の記録 .....	66
8-3 ジョイスティックの入力状態の保存 .....	69
8-4 ジョイスティック状態のグラフ表示 .....	70
9 ジョイスティックによるシミュレーションの操作 .....	72
9-1 概要 .....	72
9-2 機能の有効化／無効化 .....	75
9-3 ジョイスティックによるシミュレーションの操作 .....	76
9-4 ダイアログの呼び出し .....	78

10	ファイル/ディレクトリの表示	79
10-1	概要	79
10-2	ファイル/ディレクトリの表示	80
11	プロジェクトのブックマーク登録	83
11-1	概要	83
11-2	ブックマークマネージャの表示	84
11-3	プロジェクトの登録	85
11-4	プロジェクトの読み込み	86
11-5	プロジェクトの削除	87
12	プロジェクトの履歴の表示	88
12-1	概要	88
12-2	プロジェクトの履歴の表示と読み込み	89
13	ロボット/環境物体モデルの生成	91
13-1	概要	91
13-2	パイプモデルの生成	92
13-3	グレーチングモデルの生成	94
13-4	スロープモデルの生成	96
13-5	不整地モデルの生成	98
13-6	クローラロボットモデルの生成	106
14	慣性モーメントの計算	114
14-1	概要	114
14-2	慣性モーメントの計算	115
15	おわりに	118
	参考文献	119
	索引	120

# Contents

<b>1. Introduction to HAIROWorldPlugin</b> .....	1
1-1 Introduction.....	1
1-2 What are Choreonoid and HAIROWorldPlugin? .....	5
1-3 Setup Choreonoid.....	6
1-4 Setup HAIROWorldPlugin.....	10
<b>2. Simulating behavior of ROV/UAV</b> .....	16
2-1 Overview.....	16
2-2 Simulating behavior of ROV/UAV.....	18
<b>3. Emulating network communication failures</b> .....	31
3-1 Overview.....	31
3-2 Emulating network communication failures.....	32
<b>4. Projecting disturbed camera image</b> .....	39
4-1 Overview.....	39
4-2 Projecting disturbed camera image.....	41
<b>5. Recording trajectory of robot</b> .....	48
5-1 Overview.....	48
5-2 Recording trajectory of robot.....	49
<b>6. Recording collision status of robot</b> .....	53
6-1 Overview.....	53
6-2 Recording collision status of robot.....	54
6-3 Saving collision status.....	58
6-4 Plotting collision status.....	59
<b>7. Visualizing joystick input status</b> .....	61
7-1 Overview.....	61
7-2 Visualizing joystick input status.....	62
<b>8. Recording joystick input status</b> .....	65
8-1 Overview.....	65
8-2 Recording joystick input status.....	66
8-3 Saving joystick input status.....	69
8-4 Plotting joystick input status.....	70
<b>9. Operating simulation with a joystick</b> .....	72
9-1 Overview.....	72
9-2 Enable/Disable the function.....	75
9-3 Operating the simulation with a joystick.....	76
9-4 Calling the dialog.....	78

<b>10. Opening file and directory</b> .....	79
10-1 Overview.....	79
10-2 Opening file and directory.....	80
<b>11. Bookmarking project</b> .....	83
11-1 Overview.....	83
11-2 Calling bookmarks manager.....	84
11-3 Adding bookmark.....	85
11-4 Loading bookmarked projects.....	86
11-5 Deleting bookmark.....	87
<b>12. Referencing histories of simulation</b> .....	88
12-1 Overview.....	88
12-2 Referencing and loading history.....	89
<b>13. Generating robot / environmental object model files</b> .....	91
13-1 Overview.....	91
13-2 Generating model of pipe.....	92
13-3 Generating model of grating.....	94
13-4 Generating model of slope.....	96
13-5 Generating model of box-aligned terrain.....	98
13-6 Generating model of crawler-type robot.....	106
<b>14. Calculating moment of inertia</b> .....	114
14-1 Overview.....	114
14-2 Calculating moment of inertia.....	115
<b>15. Conclusion</b> .....	118
<b>References</b> .....	119
<b>Index</b> .....	120

# 1 HAIROWorldPlugin 入門

## 1-1 はじめに

東京電力ホールディングス株式会社（以下、東京電力 HD）福島第一原子力発電所の事故以降、多くの遠隔操作ロボットが高放射線環境下での廃炉作業に用いられている。具体的には、無人地上走行ロボット（Unmanned Ground Vehicle: 以下、UGV）や無人航空機（Unmanned Aerial Vehicle: 以下、UAV）が原子炉建屋内の調査や線量計測のために活用されてきた<sup>[1]</sup>。また、水中ロボット（Remotely Operated Vehicle: 以下、ROV）は、1号機・2号機及び3号機の原子炉格納容器内の水中探査に用いられた<sup>[2], [3]</sup>。遠隔操作ロボットを用いた廃炉作業において、ロボットを操作するオペレータは、予測が難しい空間内でロボットを操作しなければならない。つまり、ロボットに搭載したカメラやセンサの情報を参照して対象となるロボットを的確に遠隔操作することが求められる。ロボットを現場に実際に投入した訓練の実施は、現実的には不可能であることから、環境模擬体（以下、モックアップ）を利用して、作業手順を確認しながらロボットを遠隔操作する訓練を行うことが、オペレータがロボットの遠隔操作に習熟するための一般的なアプローチである。しかしながら、廃炉作業の進捗状況に合わせてモックアップを構築するには大きなコスト（時間・費用）が必要となるという課題がある。

このことから日本原子力研究開発機構（以下、原子力機構）櫛葉遠隔技術開発センターでは、オペレータがロボットの遠隔操作に習熟する訓練環境提供のために、ロボットの遠隔操作を擬似体験するためのロボットシミュレータの開発を行っている<sup>[4]</sup>。ロボットシミュレータを活用することで、計算機を用意するだけで、訓練環境を構築でき、訓練中のロボットの故障トラブルリスクを排除できるといったメリットがある。

我々は、これまでに実施された原子炉建屋内での廃炉作業事例や東京電力 HD により公開されている資料や報告されている事象等を参考に、ロボットの遠隔操作の習熟に必要と考えられる廃炉作業環境及び状況を以下のように整理した。

- ROV を使った原子炉格納容器内の調査において、水の濁りや粉塵等の影響で ROV の搭載カメラが撮影した画像の視認性が悪化することがある<sup>[5]</sup>。
- ROV を使った原子炉格納容器内の調査において、ROV の挙動が影響して粉塵を巻き上げる可能性があるため、それを抑制する操作が必要となることがある<sup>[5]</sup>。
- UAV を使って原子炉建屋内を調査するためには、UAV を目視外で操作して、屋内を安全に飛行させることが必要となる<sup>[6]</sup>。
- 遠隔操作時に通信障害が発生する／通信が不安定化することがある<sup>[7]</sup>。

これらはロボットを活用して廃炉作業を行う上で、作業の目的や内容が廃炉作業の進捗状況によって異なる場合においても、今後も直面し得る環境や状況であることから、ロボットの遠隔操作訓練にも重要な課題となると考えた。

そこで、これらの環境や状況を訓練環境内で体験できるようにするために、Choreonoid<sup>[8]</sup>をプラットフォームとしたロボットシミュレータを開発し、さらに以下の機能を付加するプラグイン (HAIROWorldPlugin) を開発した<sup>[4]</sup>。

- A) ROV/UAV が流体（水・空気等）から受ける浮力・抵抗力を計算する機能
- B) ロボットに取り付けたスラスタ・ロータを通じて ROV/UAV に推進力とトルクを与える機能
- C) 通信遅延・帯域減少・パケットロスを擬似的に発生させる機能
- D) ノイズ・歪み等の影響を受けた不鮮明なカメラ画像を提示する機能

その他にも、ユーザが簡便に仮想訓練環境を設定するための機能として以下のものが含まれている。

- E) ロボットの移動軌跡を記録・表示する機能
- F) Choreonoid のシミュレーション実行に関する状態（開始・再開・一時停止・停止）をジョイスティックの入力に基づいて制御する機能
- G) シミュレーションに使用しているロボットの設定が記述されたファイルやファイルが格納されているディレクトリを探索し、直接開く機能
- H) ジョイスティックの入力状態を表示する機能
- I) シミュレーション中のジョイスティックの入力状態を記録する機能
- J) シミュレーションの設定が記述されたファイル（プロジェクトファイル）を「ブックマーク」リストに登録する機能
- K) プロジェクトファイルの使用履歴を表示する機能
- L) シミュレーションに使用できるモデルを、ユーザが設定したパラメータに基づいて生成する機能（パイプ・グレーチング・スロープ・不整地及びサブクローラ付きクローラロボット）

本稿では、既存の(A)から(G)の HAIROWorldPlugin の機能と、新たに追加された(H)から(L)の機能の設定方法と操作手順を記している。本章では、Choreonoid と HAIROWorldPlugin のセットアップ手順を説明する。第 2 章以降では各章毎に、HAIROWorldPlugin が提供する機能の概要・設定方法と操作手順を説明する。本稿で使用する用語の詳細については、3-4 ページまたは Choreonoid マニュアル<sup>[9]</sup>等を参照いただきたい。

【本稿で用いる用語の定義】

本稿で用いる用語を以下のように定義する。

AISTSimulator	: Choreonoid で用いられる標準の物理演算エンジン
CSV	: カンマ区切りでデータが記述されたテキストファイル
Git	: ファイルの変更履歴を記録・追跡するツール
GitHub	: Web 上でファイルの変更履歴を記録・追跡できるサービス
GUI	: Graphical User Interface の略称
HAIROWorldPlugin	: 原子力機構が開発した Choreonoid の拡張機能をまとめたもの。 HAIROWorld は、日本語の「廃炉」と、プログラミング入門時にしばしば用いられるキーワード「HelloWorld」を組み合わせた造語
Makefile	: プログラムを実行できるようにするための設定が記述されたテキストファイル
TC	: Traffic Control の略
YAML	: YAML Ain't Markup Language という書式。または、YAML の書式で記述されたテキストファイル
アイテム	: プロジェクトを構成しているデータや設定等のものの単位
インクルード	: 他のファイルに記述されている内容を取り込むこと
オペレータ	: ロボットの操縦者
グレーチング	: 格子状の構造材
クローラ	: 履帯を回転させて走行する無限軌道式の走行機構の一種
子アイテム	: アイテムをツリー構造で管理したときの下位のアイテム
コントローラ	: ロボットを制御するためのプログラム
サブクローラ	: 災害対応ロボットなどに用いられる補助的な役割の無限軌道式の走行機構。ロボットの前後左右などに取り付けられ、ロボットの姿勢を制御するためなどに用いられる <sup>9)</sup>
ジョイスティック	: ゲームパッド等の入力装置
シンプルコントローラ	: ロボットを制御するためのプログラムを設定するアイテム
スラスト	: ROV に推力・トルクを与える装置
スロープ	: 斜面
ディレクトリ	: フォルダ
デバイス	: Choreonoid に実装されているセンサ・アクチュエータの総称
パイプ	: 管、導管、または筒状の物体
パス	: ファイルまたはディレクトリが格納されている場所
パッケージ	: 関連するプログラムがまとめられたもの
ビルド	: プログラムを実行できるようにする計算機上の作業
ビルドオプション	: プログラムをビルドする時に、関連する機能を含めるか除外するかを選択するための設定
不整地	: 平らにならされていない地形、土地

プラグイン	: Choreonoid の拡張機能のこと
プラグインシステム	: 新しい機能を Choreonoid に追加するための枠組みのこと
プラットフォーム	: プログラムの実行に必要な基盤のソフトウェア
プロジェクト	: シミュレーションに必要なデータや設定一式をまとめたもの。または、それらを記述したテキストファイル
ヘッダファイル	: 他のファイルに記述されたプログラムを使用する場合に読み込むファイル
ポインタ	: データが格納されている場所を示すもの
ポップアップメニュー	: マウスの副ボタンを押したときに表示されるメニュー
ボディ	: ロボットや環境物体等のモデル
ラウンドトリップ時間	: パケットが送信元と送信先の間を 1 往復するときにかかる時間
リンク	: ボディを構成する部品
ロータ	: UAV に推力・トルクを与える装置

なお、ロボットシミュレータの基盤となっているロボット用統合 GUI ソフトウェア Choreonoid に関する用語については<sup>[10]</sup>を参照のこと。

## 1-2 Choreonoid と HAIROWorldPlugin とは

Choreonoid は、株式会社コレオノイド（代表取締役：中岡慎一郎）が GitHub リポジトリ（<https://github.com/choreonoid/choreonoid>）で公開しているオープンソースのロボット用統合 GUI ソフトウェアである。動力学シミュレーション機能や動作振り付け機能を備えており、更に Choreonoid に実装されているプラグインシステムと呼ばれる機能拡張によってユーザ独自の機能も追加実装することができる。

また、Choreonoid は、Windows、Ubuntu OS がインストールされた PC で利用することができ、また MIT ライセンスに準拠している。MIT ライセンスは以下のように定義されたソフトウェアライセンスである。

- ・ このソフトウェアを誰でも無償で無制限に扱って良い。ただし、著作権表示および本許諾表示をソフトウェアのすべての複製または重要な部分に記載しなければならない
- ・ 作者または著作権者は、ソフトウェアに関してなんら責任を負わない

我々が開発した HAIROWorldPlugin は、前述のプラグインシステムを利用して実装した Choreonoid の拡張機能であり、Choreonoid と同様に MIT ライセンスに準拠している。HAIROWorldPlugin は、GitHub リポジトリ（<https://github.com/k38-suzuki/hairo-world-plugin>）で公開しており、Choreonoid のアップデートに合わせて随時更新している。

次節では、Choreonoid と HAIROWorldPlugin を使用するためのセットアップ手順を説明する。

## 1-3 Choreonoid のセットアップ

本節では、Choreonoid のセットアップ手順を説明する。セットアップにおいて、コマンドを入力する際は、半角スペースの有無を十分に注意・確認すること。

尚、Choreonoid のセットアップの詳細については、以下の Web ドキュメントを参照していただきたい。

**Choreonoid 開発版ドキュメント**：<https://choreonoid.org/ja/manuals/latest/index.html>

以下の手順に従い、Choreonoid をセットアップする。本節では、インターネット経由で Choreonoid をダウンロードするため、PC がインターネットに接続されていることを事前に確認すること。また、パスワードの入力を求められた場合は、管理者権限のパスワードを入力すること。ここでは、Ubuntu20.04 LTS を用いて、Choreonoid をセットアップする。

### ① 端末（ターミナル）の起動

(1) Ubuntu のデスクトップ画面表示中に、キーボードで[Ctrl]+[Alt]+[T]を同時に押す。

[Ctrl]+[Alt]+[T]を同時に押すと、以下のように端末（ターミナル）が起動する（図 1-1）。

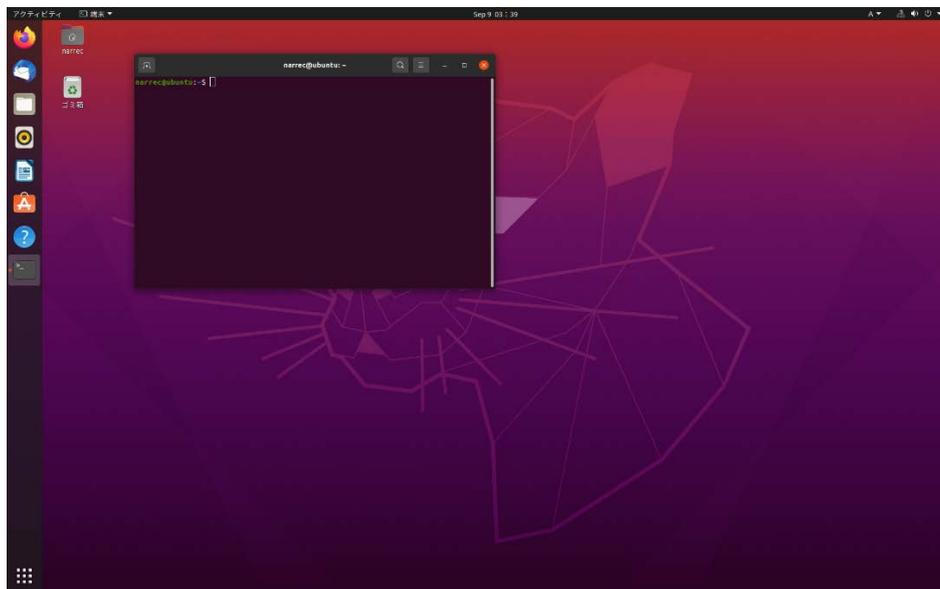


図 1-1 端末の起動

### ② Ubuntu のパッケージ一覧の更新

(1) 端末にキーボードで以下のコマンドを入力し、[Enter]を押す。

```
sudo apt update
```

### ③ Git パッケージのインストール

Choreonoid 及び HAIROWorldPlugin をインストールするために必要な Git パッケージをダウンロードする。

- (1) 端末にキーボードで以下のコマンドを入力し、[Enter]を押す。

```
sudo apt install git
```

[Enter]を押すと、インストールによるディスク消費量とインストールの続行を確認するメッセージが表示される。

- (2) 端末にキーボードで以下のコマンド入力し、[Enter] を押す。

```
y
```

### ④ Choreonoid のダウンロード

Choreonoid を GitHub リポジトリからダウンロードする。

- (1) 端末に以下のコマンドを入力し、[Enter]を押す。

```
git clone https://github.com/choreonoid/choreonoid.git
```

ダウンロードが開始されると、ダウンロードの状況が表示される。ダウンロードが完了すると、端末に「100%(\*\*/\*\*).done.」が表示される。

### ⑤ 依存パッケージのインストール

- (1) 端末に以下のコマンドを順に入力し、[Enter]を押す。

```
cd choreonoid
./misc/script/install-requisites-ubuntu-20.04.sh
```

### ⑥ Choreonoid のビルド

- (1) 端末に以下のコマンドを順に入力し、[Enter]を押す。

```
mkdir build
cd build
cmake ..
make
```

マルチコアの PC を使用している場合、make コマンドにオプション「-j」を用いて「make -j+数字」と入力することで、ビルドの時間短縮が可能である。例えば、CPU が 4 コアの場合、「make -j4」とすることで、4 つのビルドプロセスを並列で実行できる。

ビルドが完了すると、端末に[100%]が表示される。

## ⑦ Choreonoid のインストール

- (1) 端末に以下のコマンド入力し実行する。

```
sudo make install
```

## ⑧ Choreonoid の起動

- (1) 端末に以下のコマンドを入力し、[Enter]を押す。

```
choreonoid
```

[Enter]を押すと、Choreonoid の以下の画面が表示される（図 1-2）。

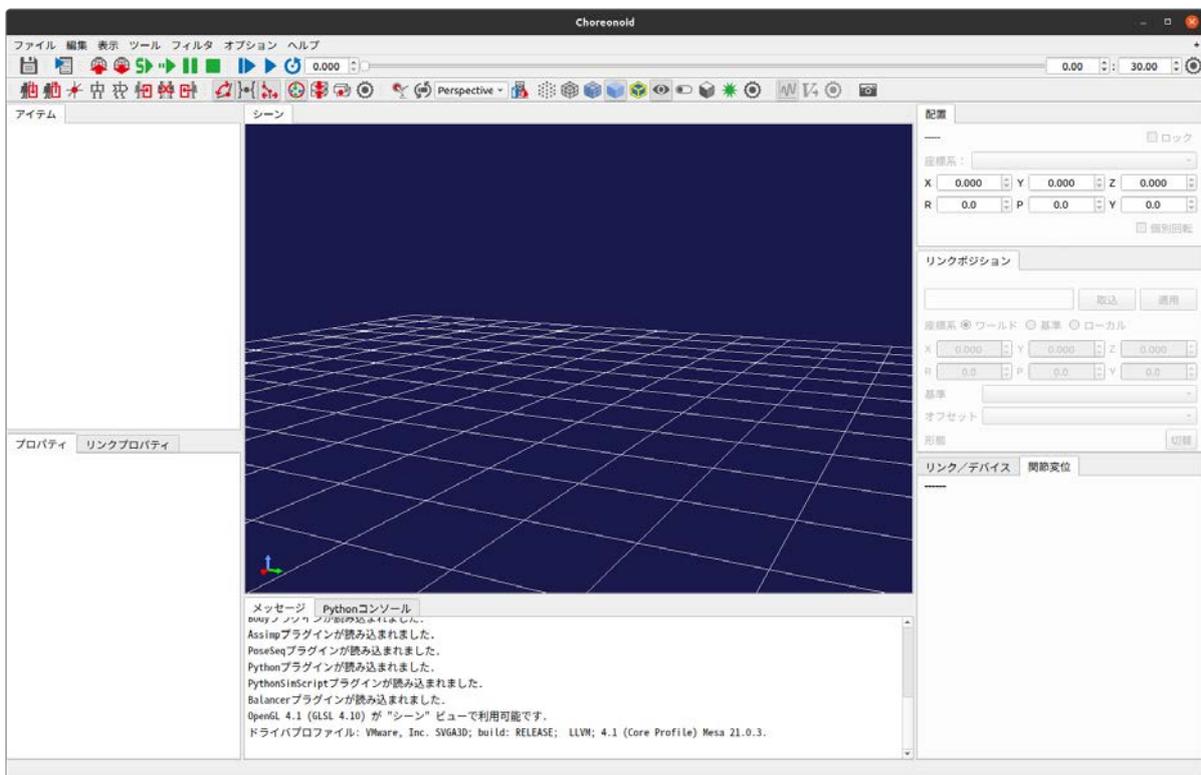


図 1-2 Choreonoid の起動

Choreonoid の画面構成と各部の名称は以下のとおりである (図 1-3)。

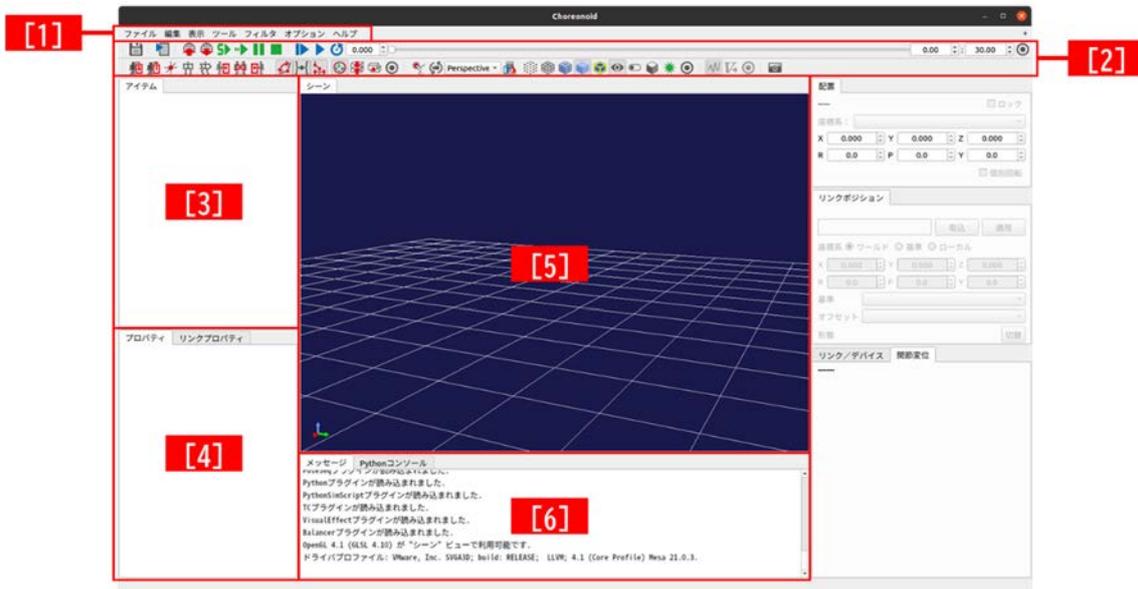


図 1-3 Choreonoid の画面構成

- [1] メインメニュー  
ファイルの読み込みやアイテムの登録等に使用する。
- [2] ツールバー  
シミュレーションの実行や設定の変更等に使用する。
- [3] アイテムツリービュー  
アイテムの選択に使用する。
- [4] プロパティビュー  
アイテムの設定の変更に使用する。
- [5] シーンビュー  
アイテムの表示に使用する。
- [6] メッセージビュー  
Choreonoid の動作状況やエラーメッセージの確認等に使用する。

Choreonoid のセットアップは以上である。

## 1-4 HAIROWorldPlugin のセットアップ

本節では、HAIROWorldPlugin のセットアップ手順を説明する。本節においても、コマンドを入力する際は、半角スペースの有無を十分に注意・確認すること。

尚、HAIROWorldPlugin のセットアップの詳細については、以下の Web ドキュメントを参照していただきたい。

**HAIROWorldPlugin マニュアル** : <https://k38-suzuki.github.io/hairo-world-plugin-doc/>

以下の手順に従い、HAIROWorldPlugin をセットアップする。本節においても、HAIROWorldPlugin をインターネット経由でダウンロードするため、PC がインターネットに接続されていることを事前に確認すること。また、パスワードの入力を求められた場合は、管理者権限のパスワードを入力すること。

### ① 端末（ターミナル）の起動

(1) Ubuntu のデスクトップ画面表示中に、キーボードで[Ctrl]+[Alt]+[T]を同時に押す。

[Ctrl]+[Alt]+[T]を同時に押すと、以下のように端末（ターミナル）が起動する（図 1-4）。

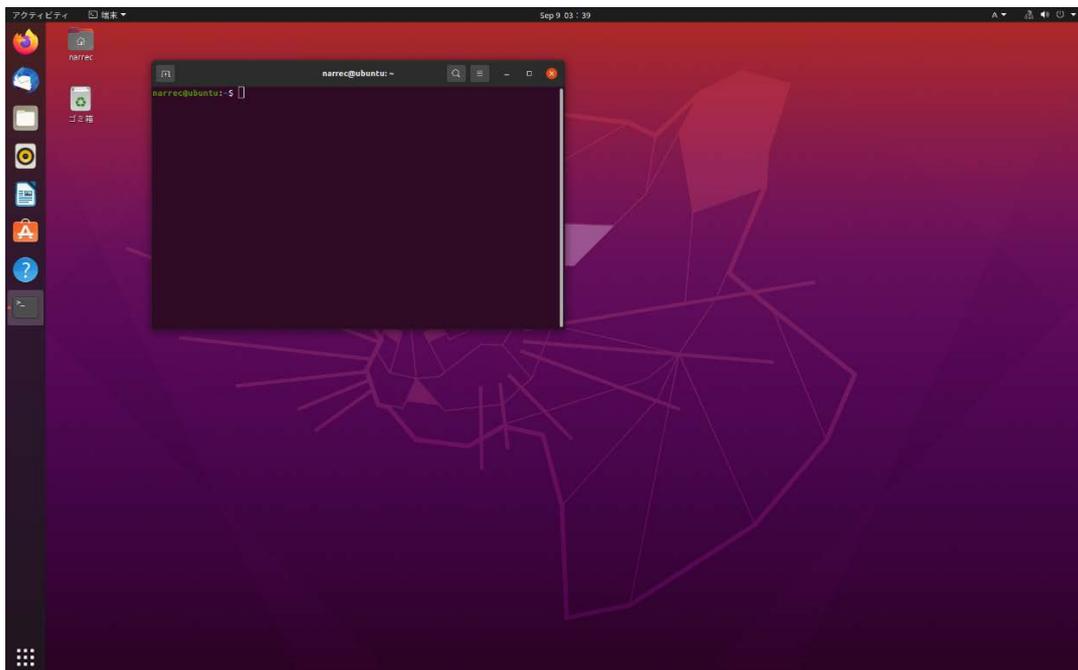


図 1-4 端末の起動

### ② 格納先ディレクトリへの移動

(1) 端末に以下のコマンドを順に入力し、[Enter]を押す。

```
cd choreonoid/ext
```

### ③ HAIROWorldPlugin のダウンロード

HAIROWorldPlugin を GitHub リポジトリからダウンロードする。

- (1) 端末に以下のコマンドを順に入力し、[Enter]を押す。

```
git clone https://github.com/k38-suzuki/hairo-world-plugin.git
```

[Enter]を押すと、ダウンロードが開始され、ダウンロードの状況が表示される。ダウンロードが完了すると端末に「100%(\*\*/\*\*).done.」が表示される。

### ④ ディレクトリ構成の確認

HAIROWorldPlugin のディレクトリ構成を確認する。

- (1) 端末に以下のコマンドを順に入力し、[Enter]を押す。

```
cd hairo-world-plugin  
ls
```

[Enter]を押すと、以下のようにディレクトリ内のファイルとディレクトリの一覧が表示される。

```
CMakeLists.txt LICENSE README.md ext include misc sample src
```

## ■ HAIROWorldPlugin のディレクトリ構成

```

hairo-world-plugin/
├── ext
├── include
├── misc
├── sample
│   ├── FileBoxPlugin
│   ├── HAIROWorld
│   │   ├── controller
│   │   ├── model
│   │   │   ├── MockupStairs
│   │   │   ├── RobotTestPool
│   │   │   ├── TestField
│   │   │   └── TrainingStage
│   └── project
├── OMPLPlugin
├── StartupPlugin
├── tutorial
└── src
    ├── BookmarkPlugin
    ├── BoxTerrainBuilderPlugin
    ├── CollisionSeqPlugin
    ├── CrawlerRobotBuilderPlugin
    ├── FileExplorerPlugin
    ├── FluidDynamicsPlugin
    ├── InertiaCalculatorPlugin
    ├── JoystickStartPlugin
    ├── JoystickStatusPlugin
    ├── MotionCapturePlugin
    ├── ObjectBuilderPlugin
    ├── TCPlugin
    └── VisualEffectPlugin

```

### ⑤ HAIROWorldPlugin の設定

- (1) 端末に以下のコマンドを順に入力し、[Enter]を押す。

```
sudo sh misc/script/install-requisites-ubuntu-20.04.sh
```

### ⑥ HAIROWorldPlugin のビルド

- (1) 端末に以下のコマンドを順に入力し、[Enter]を押す。

```
cd ../../build
cmake ..
ccmake ..
```

[Enter]を押すと、以下のように端末に Choreonoid のビルドオプションが表示される (図 1-5)。

```

narrec@ubuntu: ~/choreonoid/build
Page 1 of 4
ADDITIONAL_CXX_FLAGS
ADDITIONAL_EXT_DIRECTORIES
AGX_USE_AGGRESSIVE_WARNINGS OFF
ASSIMP_DIR /usr/lib/x86_64-linux-gnu/cmake/assimp-5.0
BODY_CUSTOMIZERS
BUILD_AGX_DYNAMICS_PLUGIN OFF
BUILD_BALANCER_PLUGIN ON
BUILD_CHOREONOID_EXECUTABLE ON
BUILD_COMPETITION_PLUGIN OFF
BUILD_DOCUMENTS OFF
BUILD_FCL_PLUGIN OFF
BUILD_GROBOT_PLUGIN OFF
BUILD_HAIRO_WORLD_PLUGIN OFF
BUILD_MANIPULATOR_PLUGIN OFF
BUILD_MEDIA_PLUGIN OFF
BUILD_MULTICOPTER_PLUGIN OFF
BUILD_ODE_PLUGIN OFF

ADDITIONAL CXX FLAGS: Additional c++ compiler optimization flags
Press [enter] to edit option Press [d] to delete an entry CMake Version 3.16.3
Press [c] to configure
Press [h] for help Press [q] to quit without generating
Press [t] to toggle advanced mode (Currently Off)

```

図 1-5 ビルドオプションの表示

- (2) キーボードで[↓]を押し、端末に表示されている「BUILD\_HAIRO\_WORLD\_PLUGIN」にカーソルを移動する (図 1-6)。

```

BUILD_GROBOT_PLUGIN OFF
BUILD_HAIRO_WORLD_PLUGIN OFF
BUILD_MANIPULATOR_PLUGIN OFF

```

図 1-6 カーソルの移動

[Enter]を押し、「BUILD\_HAIRO\_WORLD\_PLUGIN」の設定を「OFF」から「ON」に切り替えると、HAIROWorldPlugin のビルドが有効となる (図 1-7)。

```

BUILD_GROBOT_PLUGIN OFF
BUILD_HAIRO_WORLD_PLUGIN ON
BUILD_MANIPULATOR_PLUGIN OFF

```

図 1-7 ビルドの有効化

「BUILD\_HAIRO\_WORLD\_PLUGIN」の設定が「ON」に切り替えられたことを確認する。

- (3) キーボードで [c]を押す。  
(4) キーボードで [g]を押す。

[g]を押すと、Makefile の更新が完了し、端末が元の表示に戻る。

### ⑦ Choreonoid のビルド

1-3 節⑥で作成した「build」ディレクトリ内で、Choreonoid を再度ビルドする。

- (1) 端末に以下のコマンドを順に入力し、[Enter]を押す。

```
cd build
cmake ..
make
```

### ⑧ Choreonoid のインストール

- (1) 端末に以下のコマンドを順に入力し、[Enter]を押す。

```
sudo make install
```

### ⑨ Choreonoid の起動

- (1) 端末に以下のコマンドを順に入力し、[Enter]を押す。

```
Choreonoid
```

[Enter]を押すと、Choreonoid の以下の画面が表示される（図 1-8）。

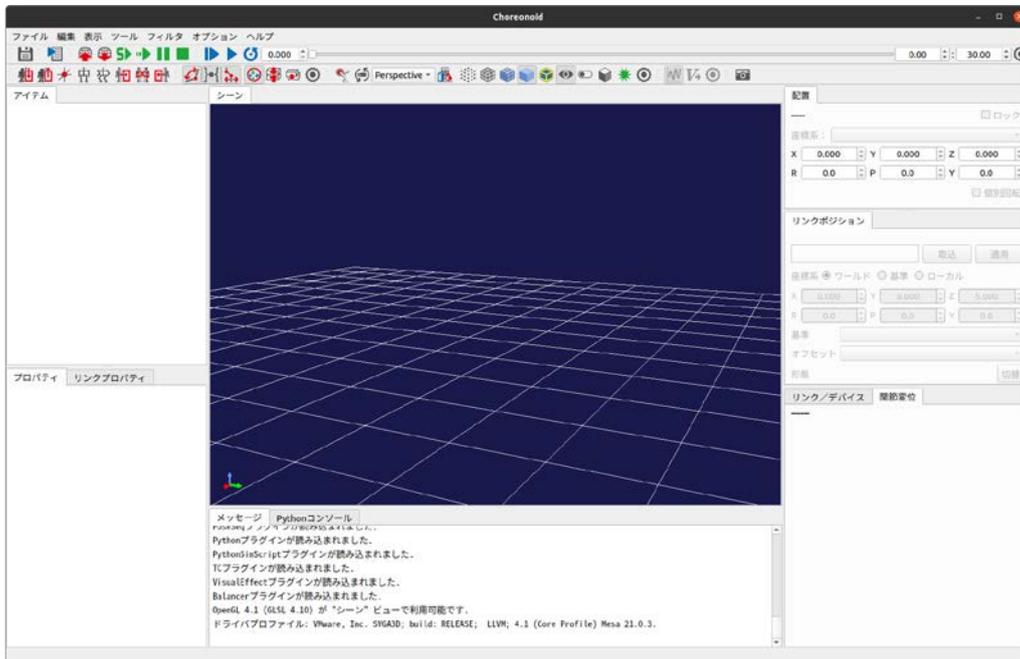


図 1-8 Choreonoid の起動

**⑩ セットアップ結果の確認**

(1) メインメニューの「ヘルプ」→「プラグインについて」にマウスカーソルを合わせる。

マウスカーソルを合わせると、以下のプラグインが表示される（表 1-1）。

表 1-1 ヘルプに表示される機能の一覧

	プラグイン名
1	BodyGenerator プラグイン
2	Bookmark プラグイン
3	CollisionSeq プラグイン
4	FileExplorer プラグイン
5	FluidDynamics プラグイン
6	InertiaCalculator プラグイン
7	JoystickStart プラグイン
8	JoystickStatus プラグイン
9	MotionCapture プラグイン
10	TC プラグイン
11	VisualEffect プラグイン

HAIROWorldPlugin のセットアップは以上である。

## 2 ROV/UAV の挙動の模擬

本章では、流体（水・空気等）の中を移動するロボット（ROV や UAV 等）の挙動を模擬するための設定と操作手順を説明する。

---

### この章の主な内容

- ・ボディの設定
  - ・スラストの設定
  - ・ロータの設定
  - ・コントローラの実装
  - ・流体エリアの設定
  - ・流体力学シミュレータの設定
- 

### 2-1 概要

本機能は、流体（水・空気等）の中を移動するボディ（ロボット）に作用する浮力・抵抗力を計算し、そのボディの挙動に反映させるための機能である（図 2-1）。また、本機能が提供する「スラスト」及び「ロータ」を用いることで、それらに取り付けられているリンクの中心に推進力・トルクを与え、ボディが流体の中を移動する様子を模擬することができる。

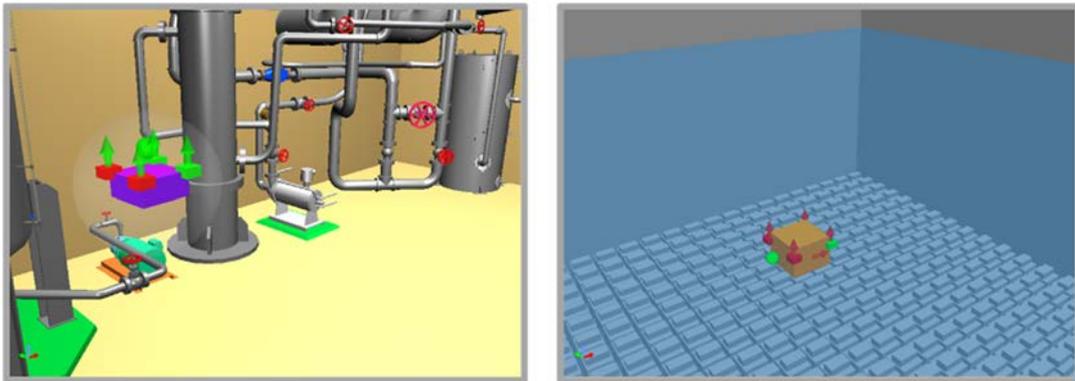


図 2-1 UAV と ROV のシミュレーション例

ここでは、プロジェクト「BoxROV.cnoid」を用いて、水中遊泳型ロボット（BoxROV）の挙動を模擬する場合を例に説明する。

BoxROV は、簡易な形状によって設計された水中遊泳型ロボット（ROV）である（図 2-2）。

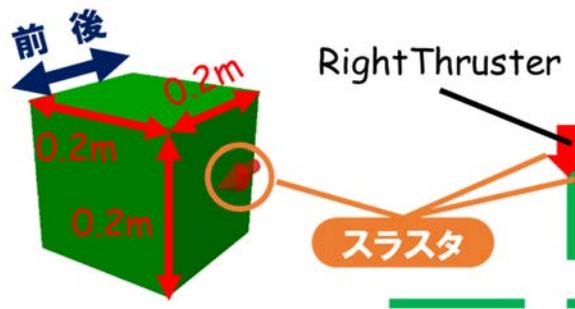


図 2-2 BoxROV の構造

BoxROV は、緑色の 1 つの箱型のリンクのみで構成され、水中を移動するためのスラストがモデル後部の左右に 1 つずつ取り付けられている。赤い矢印は、スラストの取り付け位置とスラストによって推力が加わる向きを表している。

以降の説明に用いる「BoxROV.cnoid」は、「choreonoid/ext/hairo-world-plugin/sample/tutorial」以下に格納されている。

## 2-2 ROV/UAV の挙動の模擬

以下の手順によって、ROV/UAV の挙動が模擬される。

### ① プロジェクトファイルの読み込み

- (1) メインメニューの「ファイル」→「プロジェクトを開く」を選択する。
- (2) 「プロジェクトを開く」ダイアログから、「chreonoid/ext/hairo-world-plugin/sample/tutorial」に格納されているファイル「BoxROV.cnoid」を選択する。
- (3) 「開く」ボタンを押し、プロジェクトを読み込む。

「開く」ボタンを押すと、以下のようにプロジェクトが読み込まれる（図 2-3）。

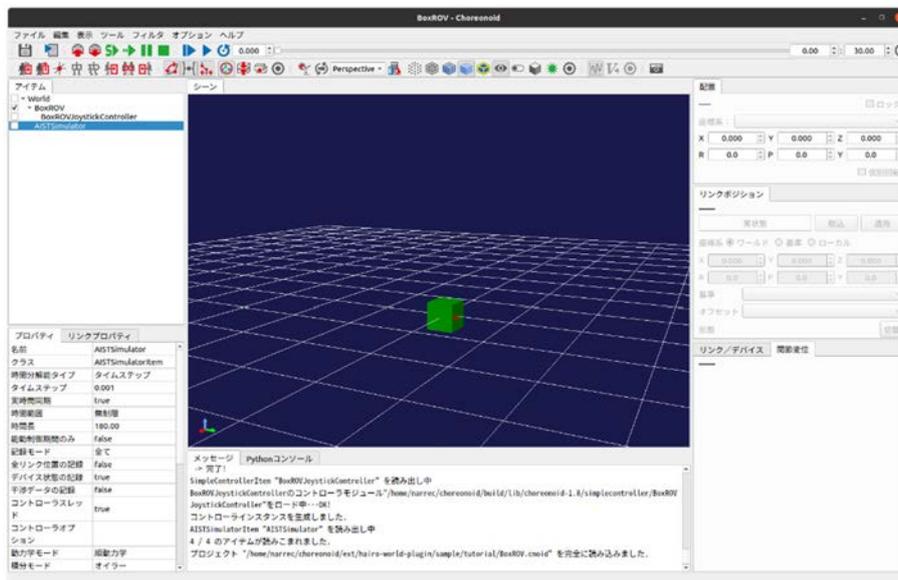


図 2-3 プロジェクトの読み込み

① リンクの設定

リンク毎に浮力・抵抗力の計算に用いるパラメータを設定する。

(1) ボディの任意のリンクに以下のように記述する。

```
density: 1000.0
centerOfBuoyancy: [ 0.0, 0.0, 0.0 ]
cdw: 1.2
tw: 0.2
cv: 10.0
surface: [ 0.04, 0.04, 0.04, 0.04, 0.04, 0.04 ]
```

リンクに記述するパラメータの詳細は以下のとおりである（表 2-1）。

表 2-1 浮力・抵抗力を計算するためのリンクのパラメータ

パラメータ	デフォルト値	単位	意味
density	0.0	kg/m <sup>3</sup>	リンクの密度を指定する。例えば、リンクの材質をアルミと仮定した場合、アルミの密度 2700 (kg/m <sup>3</sup> ) を入力する
centerOfBuoyancy (x, y, z)	0.0 0.0 0.0	m	浮力の中心の座標を指定する
cdw	0.0	-	水中での抗力係数を指定する
cda	0.0	-	空気中での抗力係数を指定する
tw	0.0	-	水中で回転運動をしている場合の抵抗力の係数を指定する。値を大きくすると回転運動を妨げるためのトルクが大きくなる
surface	0.0 0.0 0.0 0.0 0.0 0.0	m <sup>2</sup>	各座標軸方向から見たリンクの代表面積 (X+, X-, Y+, Y-, Z+, Z-) を指定する。抗力を発生させない面がある場合は、その面の代表面積を 0.0 に設定する
cv	0.0	Ns/m	粘性力の計算に用いる任意の係数。粘性力は本係数×リンクを包含している流体エリアの粘度×リンクの速度で計算される

BoxROV の場合、各面の面積が「0.04」 (= 「0.2」 × 「0.2」) [m<sup>2</sup>] であるため、「surface」は「0.04, 0.04, 0.04, 0.04, 0.04, 0.04」 (= 前面, 後面, 左面, 右面, 上面, 底面) のように記述している。

② スラスタの設定

スラスタは、水中を移動するロボットに推力とトルクを与えるためのものである。個数に制限なくリンク毎に設定することができ、リンクのローカル座標系で X 軸+方向を初期方向としている。

(1) ボディの任意のリンクの elements に以下のように記述する。

```
elements:
-
  type: Thruster
  name: Thruster
  forceOffset: 1.0
  torqueOffset: 0.1
```

スラスタのパラメータは、以下のとおりである（表 2-2）。

表 2-2 スラスタのパラメータ

パラメータ	デフォルト値	単位	意味
type	-	-	デバイスの種類。“Thruster”を入力することで、スラスタが設定される
name	-	-	デバイス名。任意のデバイス名を入力する
forceOffset	0.0	N	定常出力するスラスタの推力
torqueOffset	0.0	Nm	定常出力するスラスタのトルク
symbol	true	-	スラスタの向きを表すシンボルの表示/非表示を指定する

③ ロータの設定

ロータは、空中を移動するロボットに推力とトルクを与えるためのものである。個数に制限なくリンク毎に設定することができ、リンクのローカル座標系で Z 軸+方向を初期方向としている。

(1) ボディの任意のリンクの `elements` に以下のように記述する。

```
elements:
-
  type: Rotor
  name: Rotor
  forceOffset: 1.0
  torqueOffset: 0.1
```

ロータのパラメータは、以下のとおりである（表 2-3）。

表 2-3 ロータのパラメータ

パラメータ	デフォルト値	単位	意味
type	-	-	デバイスの種類。“Rotor” を入力することで、ロータが設定される
name	-	-	デバイス名。任意のデバイス名を入力する
forceOffset	0.0	N	定常出力するロータの推力
torqueOffset	0.0	Nm	定常出力するロータのトルク
symbol	true	-	ロータの向きを表すシンボルの表示/非表示を指定する

#### ④ シンプルコントローラの実装

ボディに設定したスラストまたはロータに推力及びトルクを発生させるためには、シンプルコントローラと呼ばれるプログラムを作成し、推力とトルクを入力する必要がある。ここでは、シンプルコントローラの実装方法について説明する。

シンプルコントローラは、シミュレーション中のロボットへの指令の送信やロボットに搭載されているデバイス・センサ類とのデータの入出力に用いられる Choreonoid のプログラムである。スラストまたはロータは、デバイスの一種として実装されているため、Choreonoid に標準で実装されている他のデバイス・センサ類と同様にシンプルコントローラを用いることで、推進力とトルクの入力が可能となっている。ここでは例として、シンプルコントローラを使用してスラストに推進力を入力する場合について説明する。

HAIROWorldPlugin では、シンプルコントローラをビルドするための ext ディレクトリを設けている。ext 内に任意の名前でディレクトリを作成し、シンプルコントローラのファイルと CMakeLists.txt を格納することで、Choreonoid マニュアル<sup>[10]</sup>の Tank チュートリアル-ステップ 2 に記載されている手順と同じ手順でシンプルコントローラのビルドと導入が可能である。また、ext には、スラスト及びロータが定義されたヘッダファイルへのパスが設定されており、シンプルコントローラ内でそれらのヘッダファイルのインクルードが可能となっている。BoxROV のコントローラでは、ジョイスティックの入力に基づいて、図 2-4 のように BoxROV に取り付けられた左右のスラストに推力が与えられる。



図 2-4 DUALSHOCK®4 での BoxROV の操作方法

以下は、BoxROV のシンプルコントローラの記述例である。

```

1  #include <cnoid/SimpleController>
2  #include <cnoid/Joystick>
3  #include <cnoid/Thruster>
4
5  using namespace std;
6  using namespace cnoid;
7
8  class BoxROVJoystickController : public SimpleController
9  {
10     Thruster* thrusterL;
11     Thruster* thrusterR;
12     Joystick joystick;
13
14 public:
15
16     virtual bool initialize(SimpleControllerIO* io) override
17     {
18         ostream& os = io->os();
19         Body* body = io->body();
20         thrusterL = body->findDevice<Thruster>("LeftThruster");
21         thrusterR = body->findDevice<Thruster>("RightThruster");
22
23         if(!thrusterL || !thrusterR){
24             os << "The thrusters are not found." << endl;
25             return false;
26         }
27
28         return true;
29     }
30
31     virtual bool control() override
32     {
33         joystick.readCurrentState();
34
35         double pos[2];
36         for(int i=0; i < 2; ++i){
37             pos[i] = joystick.getPosition(
38                 i==0 ? Joystick::L_STICK_H_AXIS : Joystick::L_STICK_V_AXIS);
39             if(fabs(pos[i]) < 0.2){
40                 pos[i] = 0.0;
41             }
42         }
43
44         double k = 0.05;
45         thrusterL->force() = k * (-2.0 * pos[1] + pos[0]);
46         thrusterR->force() = k * (-2.0 * pos[1] - pos[0]);
47         thrusterL->notifyStateChange();
48         thrusterR->notifyStateChange();
49
50         return true;
51     }
52 };
53
54 CNOID_IMPLEMENT_SIMPLE_CONTROLLER_FACTORY(BoxROVJoystickController)

```

前述のシンプルコントローラの詳細は以下のとおりである。

(1) ヘッダファイルの読み込み (3 行目)

スラスタに推力・トルクを入力するためには、

```
#include <cnoid/Thruster>
```

として、スラスタが定義されているヘッダファイルを読み込む。また、ロータに推力・トルクを入力するためには、

```
#include <cnoid/Rotor>
```

として、スラスタが定義されているヘッダファイルを読み込む。

(2) スラスタのポインタの宣言 (10~11 行目)

スラスタに推力・トルクの入力するためには、任意のポインタの名前を<name>として、

```
Thruster* <name>
```

のように、スラスタの数だけポインタを宣言する。また、ロータに推力・トルクの入力する場合は、

```
Rotor* <name>
```

のように、ロータの数だけポインタを宣言する。

(3) スラスタのポインタの取得 (20~21 行目)

スラスタのポインタを取得するためには、(2)で宣言した任意のポインタの名前が<name>、ROV のモデルファイルに記述したスラスタの名前が<thruster\_name>の場合、

```
<name> = body->findDevice<Thruster> (“<thruster_name>”);
```

と記述する。この<thruster\_name>で指定したスラスタが推力・トルクの入力を行う対象となる。

また、ロータのポインタを取得するためには、(2)で宣言した任意のポインタの名前が<name>、UAV のモデルファイルに記述したロータの名前が<rotor\_name>の場合、

```
<name> = body->findDevice<Rotor> (“<rotor_name>”);
```

と記述する。この<rotor\_name>で指定したロータが推力・トルクの入力を行う対象となる。

(4) 推力の入力 (45~46 行目)

スラスタに推力を入力するためには、(2)で宣言した任意のポインタの名前が<name>の場合、入力する任意の推力を<force\_value>[N]として、

```
<name>->force() = <force_value>;
```

と記述する。また、スラスタにトルクを入力する場合は、入力する任意のトルクを<torque\_value>[Nm]として、

```
<name>->torque() = <torque_value>;
```

と記述する。同様に、ロータに推力を入力するためには、入力する任意の推力を<force\_value>[N]として、

```
<name>->force() = <force_value>;
```

と記述し、トルクを入力する場合は、入力する任意のトルクを<torque\_value>[Nm]として、  
`<name>->torque() = <torque_value>;`  
 と記述する。

(5) 入力した推力の反映 (47~48 行目)

スラスタに入力した推力・トルクを反映させるためには、(2)で宣言した任意のポインタの名前が<name>の場合、

`<name>->notifyStateChange();`

と記述する。同様に、ロータに入力した推力・トルクを反映させるためには、

`<name>->notifyStateChange();`

と記述する。

⑤ 流体エリアの設定

流体エリアは、リンク毎に計算した浮力・抵抗力を与える領域とその領域を満たす流体の物性値を設定するものである。流体エリアで設定した領域内にロボットが進入したときに浮力・抵抗力がロボットの挙動に反映される。

- (1) アイテムツリービューの「World」を選択する。
- (2) メインメニューの「ファイル」→「新規」→「流体エリア」を選択する。
- (3) 「流体力学エリアの新規生成」ダイアログの「作成」ボタンを押す。

「作成」ボタンを押すと、「FluidArea」がアイテムツリービューに登録される (図 2-5)。

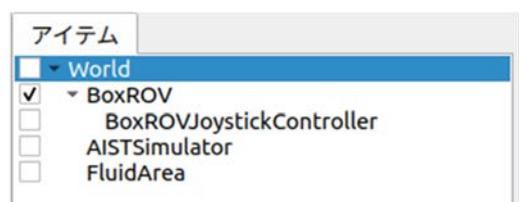


図 2-5 流体エリアの登録

(4) アイテムツリービューの「FluidArea」のチェックボックスにチェックを入れる。

チェックボックスにチェックを入れると、「FluidArea」がシーンビューに表示される（図 2-6）。

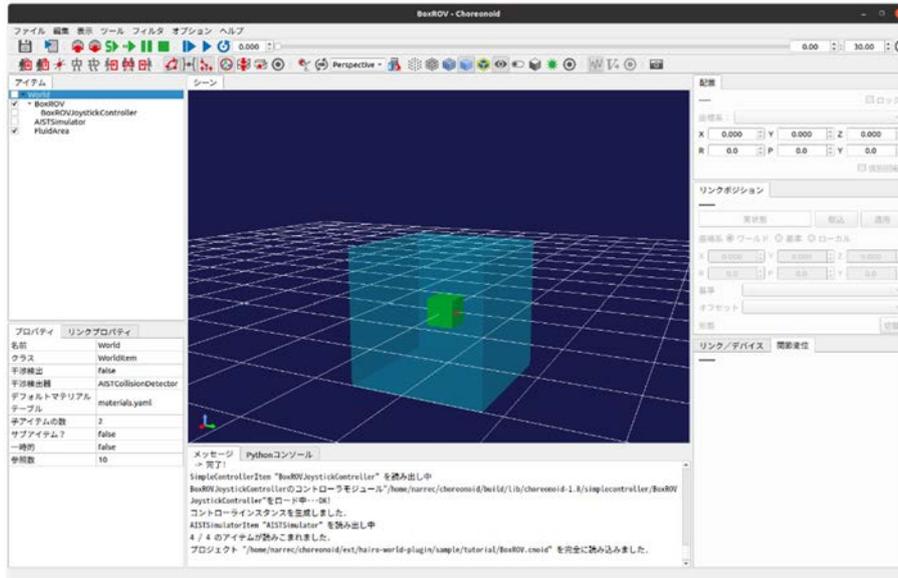


図 2-6 流体エリアの表示

(5) アイテムツリービューの「FluidArea」を選択する。

「FluidArea」を選択すると、プロパティビューに以下のプロパティが表示される（表 2-4）。

表 2-4 流体エリアのプロパティ

パラメータ	デフォルト値	単位	意味
密度	0	kg/m <sup>3</sup>	流体の密度を指定する
粘度	0	Pa*s	流体の粘度を指定する
定常流	0, 0, 0	N	流体エリア中にある物体に与える外力を X, Y, Z の順で指定する
形状	Box	-	流体エリアの形状を指定する (Box/Cylinder/Sphere)
サイズ	1.0, 1.0, 1.0	m	流体エリアのサイズを X, Y, Z の順で指定する (形状が Box の場合のみ)
半径	1	m	流体エリアの半径を指定する (形状が Cylinder・Sphere の場合のみ)
高さ	1	m	流体エリアの高さを指定する (形状が Cylinder の場合のみ)
位置	0, 0, 0	m	流体エリアの位置を X, Y, Z で指定する
RPY	0, 0, 0	deg	流体エリアの向きを Roll, Pitch, Yaw の順で指定する
拡散色	0, 0, 0	-	流体エリアの拡散色を R, G, B の順で指定する
透過度	0	-	流体エリアの透過度を指定する

パラメータ「形状」では、流体エリアの形状が以下の 3 種から選択できる（図 2-7）。

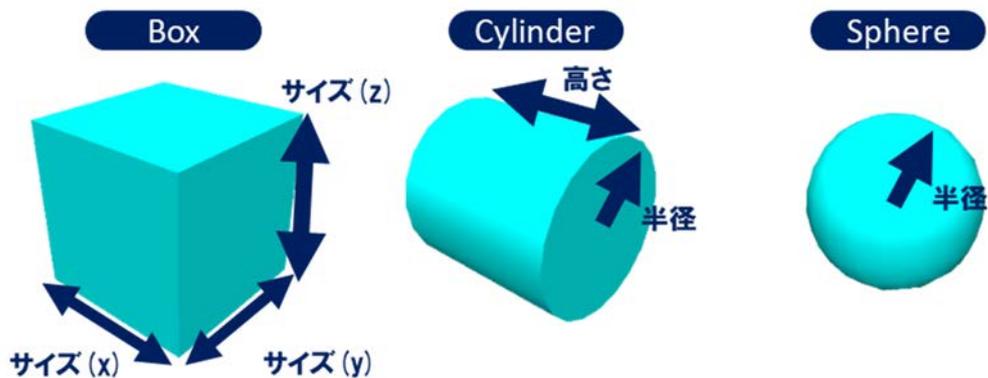


図 2-7 流体エリアの形状

水と空気の密度及び粘度は以下のとおりである（表 2-5、表 2-6）。

表 2-5 水の密度と粘度<sup>[11]</sup>

温度【℃】	密度【kg/m <sup>3</sup> 】	粘度【Pa・s】
0	999.89	0.001792
10	999.69	0.001307
20	998.22	0.001002
30	995.67	0.000797
40	992.24	0.000653
50	988.02	0.000548

表 2-6 空気の密度と粘度<sup>[12]</sup>

温度【℃】	密度【kg/m <sup>3</sup> 】	粘度【Pa・s】
-10	1.33	0.00001673
0	1.28	0.00001724
10	1.23	0.00001772
20	1.19	0.00001823
30	1.15	0.00001872
40	1.11s	0.00001920

- (6) プロパティビューの「密度」（単位：kg/m<sup>3</sup>）の「0」を選択し、密度を入力する。ここでは例として、水の密度「1000」を入力する。
- (7) プロパティビューの「粘度」（単位：Pa・s）の「0」を選択し、粘度を入力する。ここでは例として、水の粘度「0.001」を入力する。

プロパティの設定が完了すると、プロパティビューは以下ようになる（図 2-8）。

プロパティ	リンクプロパティ
名前	FluidArea
クラス	FluidAreaItem
密度	1000
粘度	0.001
定常流	0.0 0.0 0.0
形状	Box
サイズ	1.0 1.0 1.0
位置	0.0 0.0 0.0
RPY	0.0 0.0 0.0
拡散色	0.0 1.0 1.0
透過度	0.800000
子アイテムの数	0
サブアイテム?	false
一時的	false
参照数	10

図 2-8 FluidArea のプロパティの設定

#### ⑥ 流体力学シミュレータの設定

流体力学シミュレータは、流体エリア内にロボットが進入したときに、そのロボットに作用する浮力と抵抗力を流体エリアに設定された流体の物性値に基づいて計算し、ロボットの挙動に反映するものである。また、スラストまたはロータによって与えられた推力とトルクもロボットの挙動に反映する。

- (1) アイテムツリービューの「AISTSimulator」を選択する。
- (2) メインメニューの「ファイル」→「新規」→「流体力学シミュレータ」を選択する。
- (3) 「流体力学シミュレータの新規生成」ダイアログの「作成」ボタンを押す。

「作成」ボタンを押すと、「FluidDynamicsSimulator」が「AISTSimulator」の子アイテムとして、アイテムツリービューに登録される（図 2-9）。

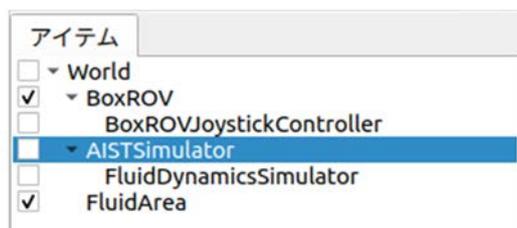


図 2-9 FluidDynamicsSimulator の登録

⑦ ROV/UAV の挙動の模擬

(1) ツールバーの「初期位置からシミュレーションを開始」ボタン  を押す。

シミュレーションが開始と、BoxROV の密度と FluidArea で設定した水の密度がそれぞれ 1000 (kg/m<sup>3</sup>) に設定されているため、重力と BoxROV に作用する浮力が釣り合い、BoxROV が FluidArea 内で静止する (図 2-10)。

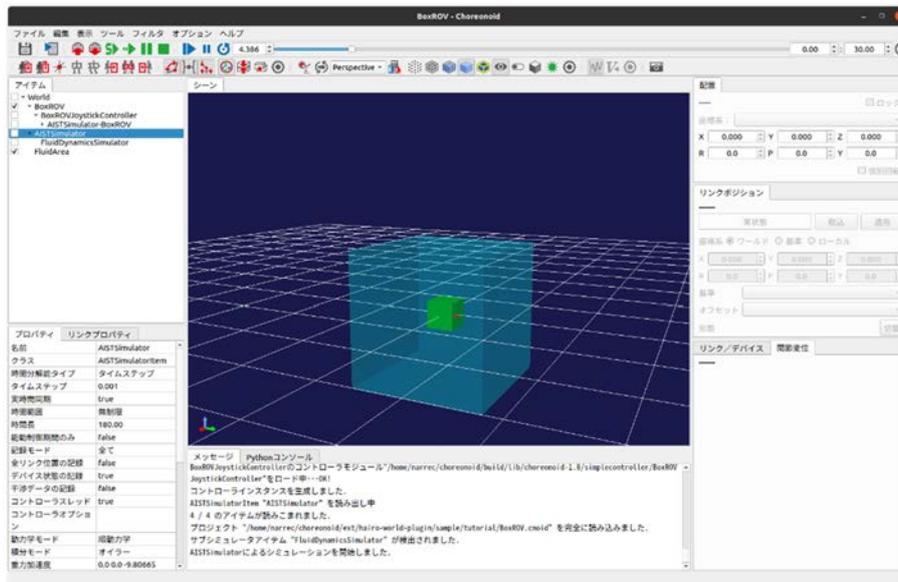


図 2-10 BoxROV のシミュレーションの開始

流体 (水・空気等) の中を移動するロボット (ROV や UAV 等) の挙動を模擬するための設定と操作手順は以上である。

## 3 通信障害効果の付与

本章では、シミュレーション中に通信障害効果を適用するための設定と操作手順を説明する。

---

### この章の主な内容

- ・通信シミュレータの設定
  - ・通信エリアの設定
- 

### 3-1 概要

本機能は、Choreonoid を起動している PC が送信または受信するパケットに通信障害効果（遅延・帯域制限・パケットロス）を付与するための機能である。本機能では、以下の通信障害効果を付与することができる（表 3-1）。

表 3-1 通信障害効果

通信障害効果名	意味
内向き遅延	パケットを受信する際の遅延時間
内向きレイテンシ	パケットを受信する際のレイテンシ（通信速度の上限）
内向きパケット損失	パケットを受信する際のパケット損失率
外向き遅延	パケットを送信する際の遅延時間
外向きレイテンシ	パケットを送信する際のレイテンシ（通信速度の上限）
外向きパケット損失	パケットを送信する際のパケット損失率

本機能では、「通信エリア」と「通信シミュレータ」を用いて、上述の通信障害効果を模擬するための設定を行う。

ここでは、プロジェクト「Tank.cnoid」を用いて、通信エリアの領域内に Tank が進入したときに通信障害の効果を発生させる場合を例に説明する。

尚、説明に用いる「Tank.cnoid」は、「choreonoid/ext/hairo-world-plugin/sample/tutorial」以下に格納されている。

## 3-2 通信障害効果の設定

以下の手順によって、通信障害効果が設定される。

### ① プロジェクトファイルの読み込み

- (1) メインメニューの「ファイル」→「プロジェクトを開く」を選択する。
- (2) 「プロジェクトを開く」ダイアログから、「choreonoid/ext/hairo-world-plugin/sample/tutorial」に格納されているファイル「Tank.cnoid」を選択する。
- (3) 「開く」ボタンを押し、プロジェクトを読み込む。

「開く」ボタンを押すと、以下のようにプロジェクトが読み込まれる（図 3-1）。

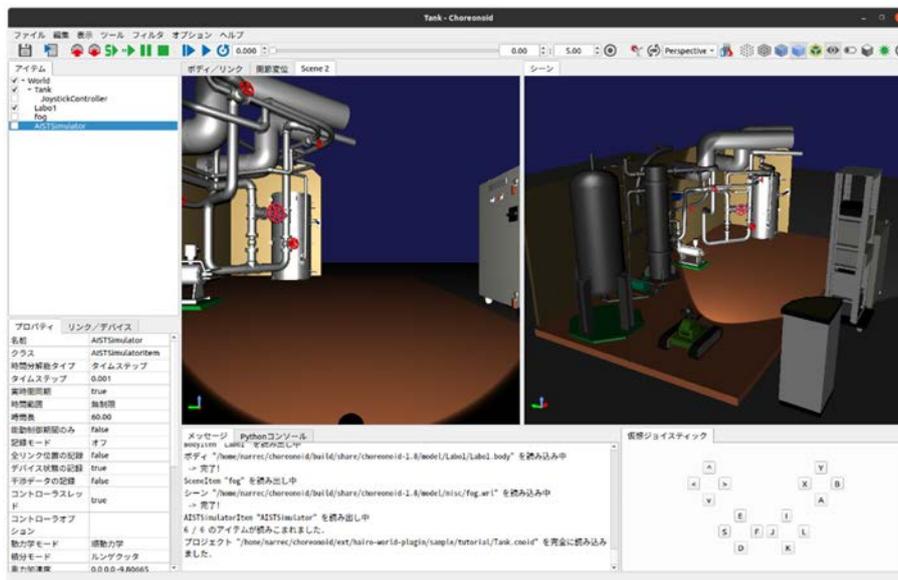


図 3-1 プロジェクトの読み込み

### ② 通信エリアの設定

通信エリアは、通信障害効果を与える領域を設定するものである。通信エリアで設定した領域内にロボットが進入したときに通信障害の効果が適用される。

- (1) アイテムツリービューの「World」を選択する。
- (2) メインメニューの「ファイル」→「新規」→「通信エリア」を選択する。
- (3) 「通信エリアの新規生成」ダイアログの「作成」ボタンを押す。

「作成」ボタンを押すと、「TCArea」がアイテムツリービューに登録される（図 3-2）。

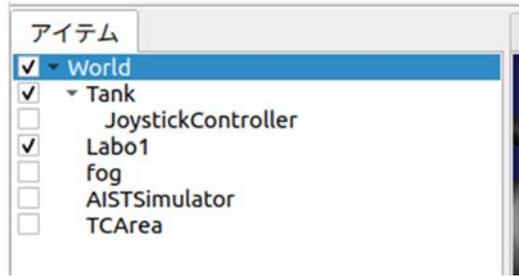


図 3-2 通信エリアの登録

(4) アイテムツリービューの「TCArea」左のチェックボックスにチェックを入れる。

チェックボックスにチェックを入れると、通信エリアがシーンビューに表示される（図 3-3）。



図 3-3 通信エリアの表示

(5) アイテムツリービューの「TCArea」を選択する。

「TCArea」を選択すると、プロパティビューに以下のプロパティが表示される（表 3-2）。

表 3-2 通信エリアのプロパティ

パラメータ	デフォルト値	単位	意味
内向き遅延	0	ms	パケットを受信する際の遅延時間を指定する
内向きレイテンシ	0	kbit/s	パケットを受信する際のレイテンシ（通信速度の上限）を指定する
内向きパケット損失	0	%	パケットを受信する際のパケット損失率を指定する
外向き遅延	0	ms	パケットを送信する際の遅延時間を指定する
外向きレイテンシ	0	kbit/s	パケットを送信する際のレイテンシ（通信速度の上限）を指定する
外向きパケット損失	0	%	パケットを送信する際のパケット損失率を指定する
送信元 IP アドレス	0.0.0.0	-	パケットの送信元を限定する際の IP アドレスを指定する
送信先 IP アドレス	0.0.0.0	-	パケットの送信先を限定する際の IP アドレスを指定する
形状	Box	-	通信エリアの形状を指定する (Box/Cylinder/Sphere)
サイズ	1.0, 1.0, 1.0	m	通信エリアのサイズを XYZ で指定する (形状が Box の場合のみ)
半径	1	m	通信エリアの半径を指定する (形状が Cylinder・Sphere の場合のみ)
高さ	1	m	通信エリアの高さを指定する (形状が Cylinder の場合のみ)
位置	0, 0, 0	m	通信エリアの位置を XYZ で指定する
RPY	0, 0, 0	deg	通信エリアの向きを Roll, Pitch, Yaw で指定する
拡散色	0, 0, 0	-	通信エリアの拡散色を RGB で指定する
透過度	0	-	通信エリアの透過度を指定する

ms : 1ms は 0.001s (秒) を表す。

kbit/s : 1kbit/s は 1 秒間に 1000bit の情報を送信または受信することを表す。

deg : 1deg は、1 度 (°) を表す。

パラメータ「形状」では、通信エリアの形状が以下の3種から選択できる（図3-4）。

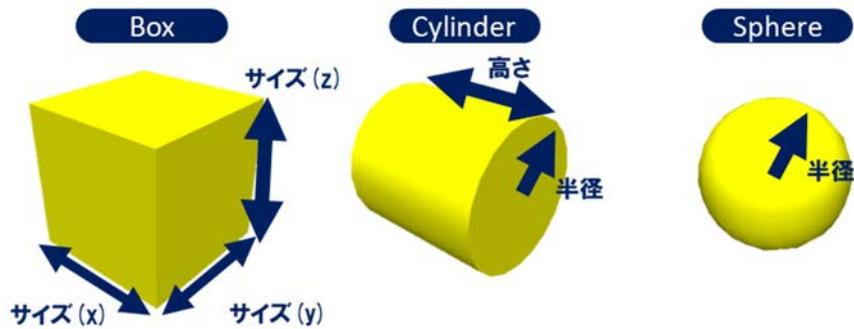


図3-4 通信エリアの形状

- (6) プロパティビューの「位置」(単位: m, m, m) の「000」を選択し、通信エリアの領域の中心の座標をXYZ (m, m, m) で入力する。ここでは例として、「0-0.5 0.6」を入力する。

「位置」のパラメータを変更すると、シーンビューに表示されている「TCArea」の位置が以下のように変更される（図3-5）。

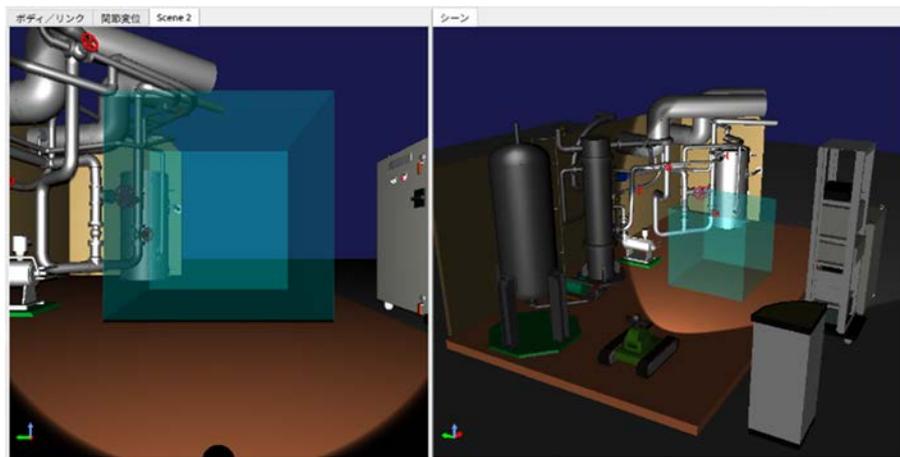


図3-5 通信エリアの表示（「位置」を変更した場合）

- (7) プロパティビューの「内向き遅延」(単位: ms) の「0」を選択し、通信元の PC（ここでは Choreonoid がインストールされている PC）がパケットを受信するときに発生させる遅延時間を入力する。ここでは例として、「100」を入力する。

プロパティの設定が完了すると、プロパティビューは以下ようになる（図 3-6）。

プロパティ	リンクプロパティ
名前	TCArea
クラス	TCAreaitem
内向き遅延	100
内向きレイテンシ	0
内向きパケット損失	0
外向き遅延	0
外向きレイテンシ	0
外向きパケット損失	0
送信元IPアドレス	
送信先IPアドレス	
形状	Box
サイズ	1.0 1.0 1.0
位置	0.0 -0.5 0.6
RPY	0.0 0.0 0.0
拡散色	0.0 1.0 1.0
透過度	0.800000
子アイテムの数	0
サブアイテム?	false
一時的	false
参照数	10

図 3-6 TCArea のプロパティの設定

### ③ 通信シミュレータの設定

通信シミュレータは、通信エリアの領域内にロボットが進入したときに、通信エリアの設定に従って通信障害効果を付与するものである。

- (1) アイテムツリービューに登録されている「AISTSimulator」を選択する。
- (2) メインメニューの「ファイル」→「新規」→「通信シミュレータ」を選択する。
- (3) ダイアログ「通信シミュレータの新規生成」の「作成」ボタンを押す。

「作成」ボタンを押すと、「TCSimulator」が「AISTSimulator」の子アイテムとして、アイテムツリービューに登録される（図 3-7）。

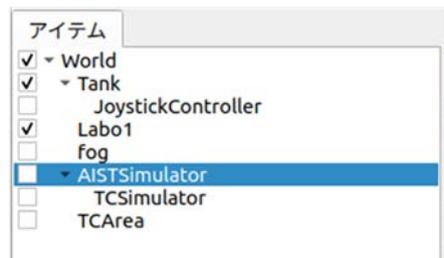


図 3-7 TCSimulator の登録

- (4) アイテムツリービューの「TCSimulator」を選択する。

- (5) プロパティビューの「インタフェース」から通信障害効果を設定する任意の LAN アダプタを選択する。ここでは例として、使用している PC 自身と通信を行うため「lo」を選択している (図 3-8)。

プロパティ	リンクプロパティ
名前	TCSimulator
クラス	TCSimulatorItem
有効	true
インタフェース	lo
IFBデバイス	ifb0
子アイテムの数	0
サブアイテム?	false
一時的	false
参照数	11

図 3-8 TCSimulator のプロパティの設定

#### ④ 通信障害効果の確認

以下の手順に従い、通信障害効果が適用されていることを確認する。ここでは例として、「ping」を用いて、設定した遅延時間を計測する。

- (1) ツールバーの「初期位置からシミュレーションを開始」ボタン  を押し、シミュレーションを開始する。
- (2) キーボードで[Ctrl]+[Alt]+[T]を同時に押し、端末（ターミナル）を起動する。
- (3) 端末にキーボードで以下のコマンドを入力し、[Enter]を押す。ここでは、通信シミュレータのパラメータ「インタフェース」で、「lo」を選択しているため、「localhost」を指定している。  
「lo」以外のインタフェースを指定している場合は、「localhost」を通信先の IP アドレスに置き換える。

```
ping localhost
```

ジョイスティック等で Tank を操縦し、通信エリアの領域内に Tank を進入させると（図 3-9）、端末に遅延を含むラウンドトリップ時間が表示される（図 3-10）。



図 3-9 通信エリアの領域内への Tank の進入

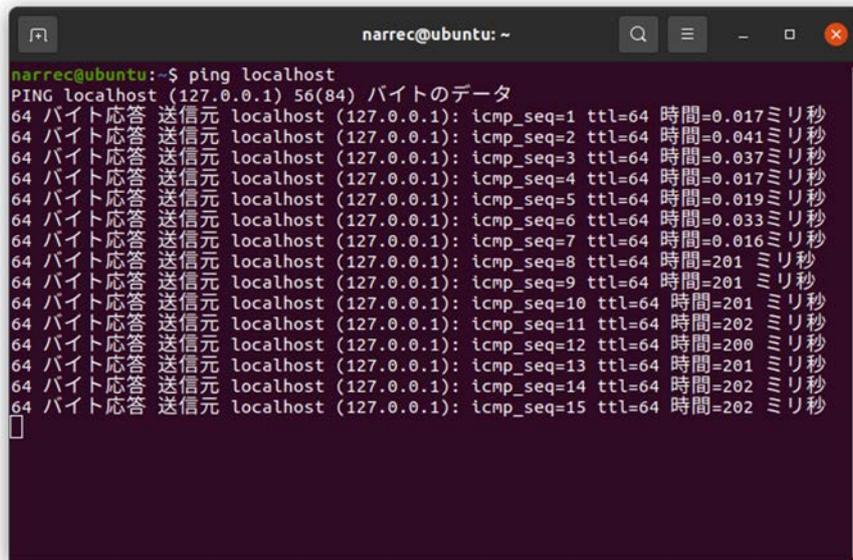


図 3-10 ラウンドトリップ時間の表示

シミュレーション中に通信障害効果を適用するための設定と操作手順は以上である。

## 4 カメラ画像への視覚効果の付与

本章では、シミュレーション中のロボットに取り付けられているカメラで撮影された画像にノイズ・歪み・色相変化の視覚効果を付与するための設定と操作手順を説明する。

### この章の主な内容

- ・ビジュアルエフェクタの設定
- ・視覚効果の設定

### 4-1 概要

本機能は、シミュレーション中のロボットに取り付けられているカメラで撮影された画像にノイズ・歪み・色相変化の視覚効果を付与するための機能である。

本機能では、「視覚効果」を用いて、視覚効果をカメラ画像に付与するための設定を行う。

本機能により、以下の視覚効果をカメラ画像に付与することができる（表 4-1、図 4-1）。

表 4-1 視覚効果の一覧

パラメータ	デフォルト値	単位	意味
色相	0	-	色相を調整する
彩度	0	-	彩度を調整する
明度	0	-	明度を調整する
赤	0	-	赤色の輝度を調整する
緑	0	-	緑色の輝度を調整する
青	0	-	青色の輝度を調整する
歪み	0	-	樽型歪みの度合いを調整する
拡大率	0	-	拡大率を調整する
標準偏差	1	-	ガウシアンノイズの度合いを調整する
白色ノイズ	0	-	ゴマ塩ノイズの白色の生成確率を調整する
黒色ノイズ	0	-	ゴマ塩ノイズの黒色の生成確率を調整する
反転	false	-	画像の上下と左右を反転する
フィルタ	フィルタなし	-	画像に指定したフィルタ（ガウシアン 3×3・ガウシアン 5×5、ソーベル、プリューウィット）を適用する

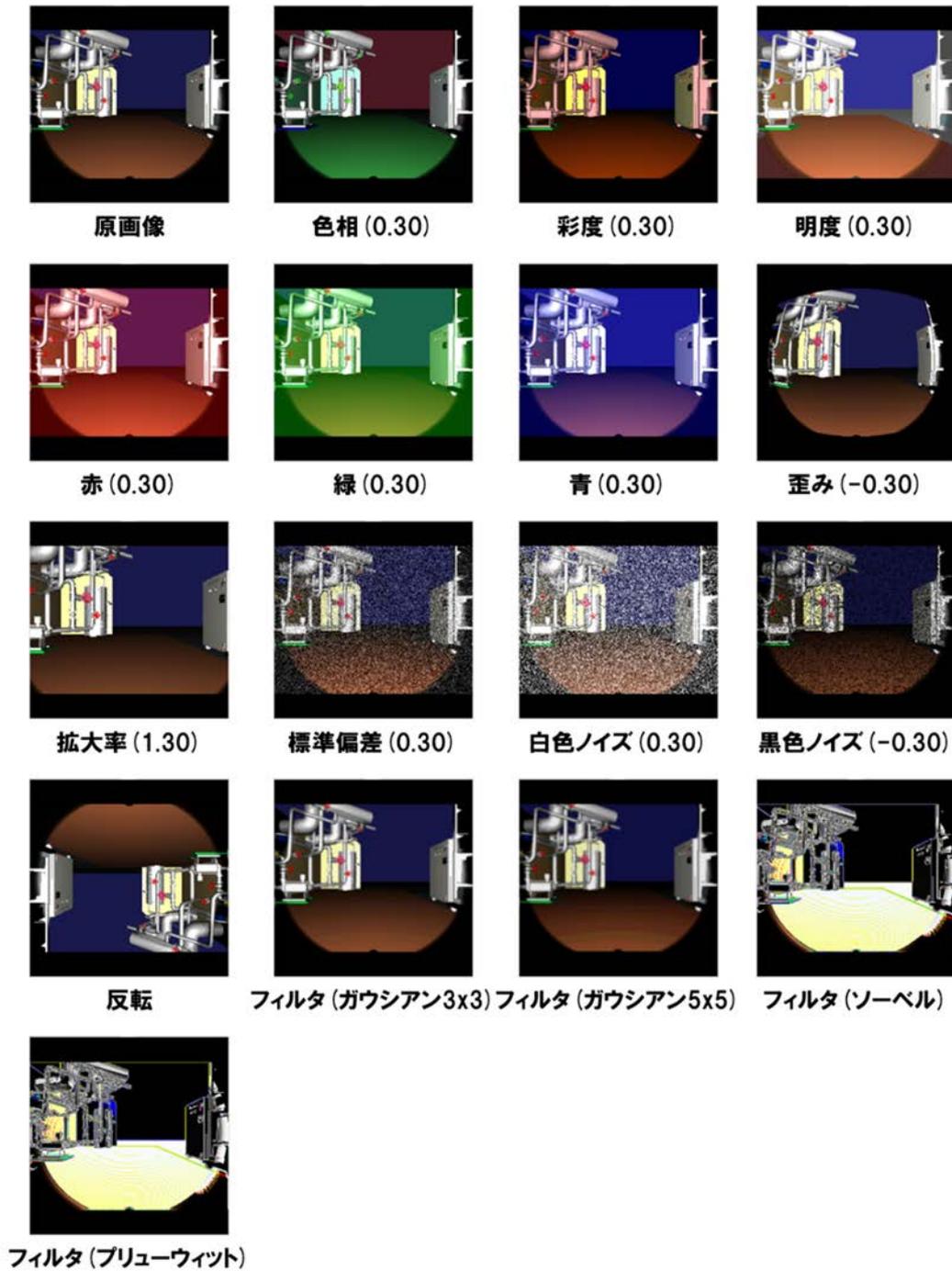


図 4-1 視覚効果を付与したカメラ画像例

ここでは、プロジェクト「Tank.cnoid」を用いて、「Tank」に取り付けられているカメラ「Kinect」によって取得されたカメラ画像に視覚効果を付与する場合を例に説明する。

尚、説明に用いる「Tank.cnoid」は、「choreonoid/ext/hairo-world-plugin/sample/tutorial」以下に格納されている。

## 4-2 視覚効果の設定

以下の手順によって、視覚効果が設定される。

### ① プロジェクトファイルの読み込み

- (1) メインメニューの「ファイル」→「プロジェクトを開く」を選択する。
- (2) 「プロジェクトを開く」ダイアログから、「choreonoid/ext/hairo-world-plugin/sample/tutorial」に格納されているファイル「Tank.cnoid」を選択する。
- (3) 「開く」ボタンを押し、プロジェクトを読み込む。

「開く」ボタンを押すと、以下のようにプロジェクトが読み込まれる（図 4-2）。

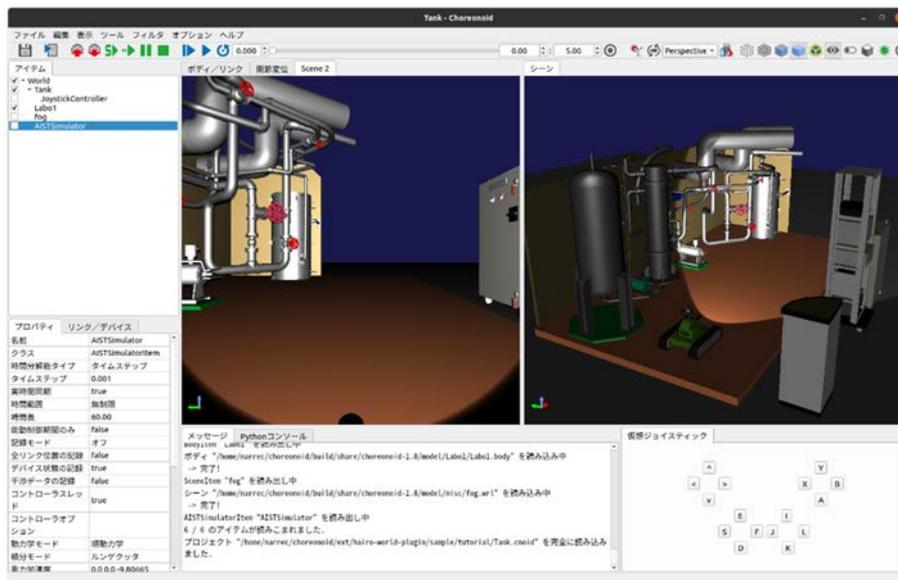


図 4-2 プロジェクトの読み込み

### ② 画像ビューバーの表示

- (1) メインメニューの「表示」→「ツールバーの表示」→「画像ビューバー」を選択する。

「画像ビューバー」を選択すると、以下のようにツールバーに「画像ビューバー」が表示される（図 4-3）。



図 4-3 画像ビューバーの表示

### ③ 画像ビューの表示

- (1) メインメニューの「表示」→「ビューの表示」→「画像」を選択する。

「画像」を選択すると、以下のように「画像ビュー」が表示される（図 4-4）。



図 4-4 画像ビューの表示

### ④ 視覚効果の設定

「視覚効果」は、カメラ画像にノイズ・歪み・色相変化等の視覚効果を付与するものである。

- (1) アイテムツリービューの「Tank」を選択する。
- (2) メインメニューの「ファイル」→「新規」→「視覚効果」を選択する。
- (3) 「視覚効果の新規生成」ダイアログの「作成」ボタンを押す。

「作成」ボタンを押すと、以下のように「VisualEffector」が「Tank」の子アイテムとして、アイテムツリービューに登録される（図 4-5）。

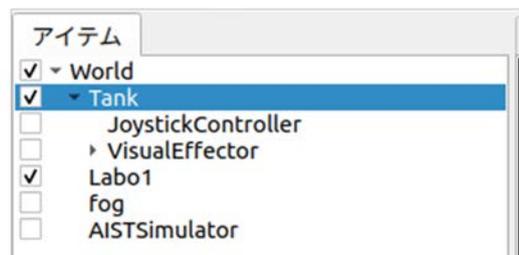


図 4-5 VisualEffector の登録

- (4) アイテムツリービューの「VisualEffector」左の「▶」マークを押す。

「▶」マークを押すと、「VisualEffector」の子アイテムに「Kinect\_Image」が表示される（図 4-6）。

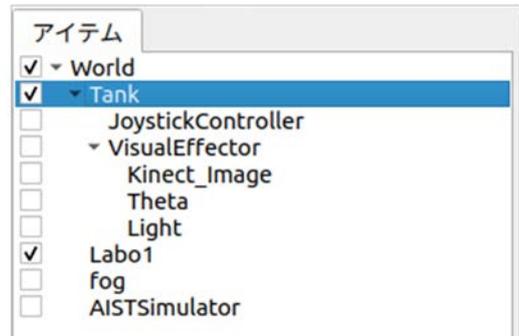


図 4-6 VisualEffector の子アイテムの表示

- (5) アイテムツリービューの「Kinect\_Image」左のチェックボックスにチェックを入れる（図 4-7）。このチェックボックスにチェックが入っているときに視覚効果を付与したカメラ画像の生成が有効となり、チェックが外れているときは、無効となる。

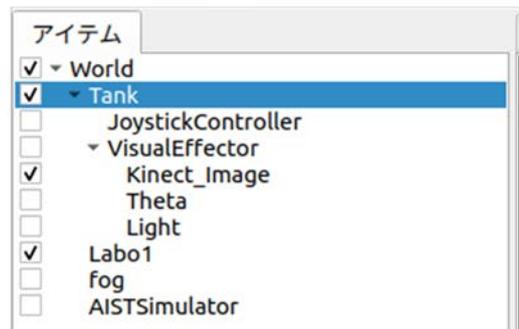


図 4-7 カメラ画像生成の有効化

## ⑤ GL ビジョンシミュレータの設定

「GL ビジョンシミュレータ」は、カメラ画像を生成するためのものである。

- (1) アイテムツリービューの「AISTSimulator」を選択する。
- (2) メインメニューの「ファイル」→「新規」→「GL ビジョンシミュレータ」を選択する。
- (3) 「GL ビジョンシミュレータの新規生成」ダイアログの「作成」ボタンを押す。

「作成」ボタンを押すと、以下のように「GLVisionSimulator」が「AISTSimulator」の子アイテムとして、アイテムツリービューに登録される（図 4-8）。

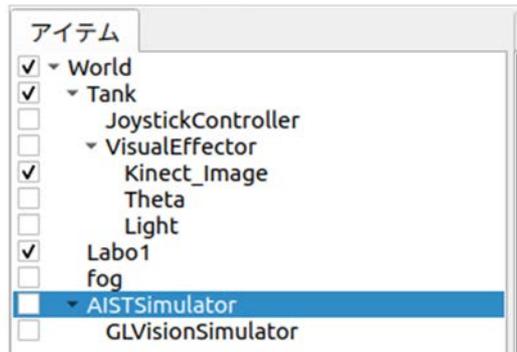


図 4-8 GLVisionSimulator の登録

- (4) アイテムツリービューの「GLVisionSimulator」を選択する。
- (5) プロパティビューの「ビジョンデータの記録」を選択し、「True」を選択する（図 4-9）。

プロパティ		リンク/デバイス	
名前	GLVisionSimulator		
クラス	GLVisionSimulatorItem		
有効	True		
対象ボディ			
対象センサ			
最大フレームレート	1000.00		
最大レイテンシ[秒]	1.00		
ビジョンデータの記録	True		
スレッドモード	センサ		
ベストエフォート	False		
全てのシーンオブジェクト	False		

図 4-9 GLVisionSimulator のプロパティの設定

⑥ 視覚効果の確認

以下の手順に従い、カメラ画像に視覚効果が付与されることを確認する。

- (1) ツールバーの「初期位置からシミュレーションを開始」ボタン  を押す。
- (2) 「画像ビュー」を選択する。このとき、特に表示は変化しない。
- (3) ツールバーの「画像ビューバー」の「No Selection」を選択し、「Kinect\_Image」を選択する。

「Kinect\_Image」を選択すると、画像ビューに「Kinect」のカメラ画像が表示される（図 4-10）。

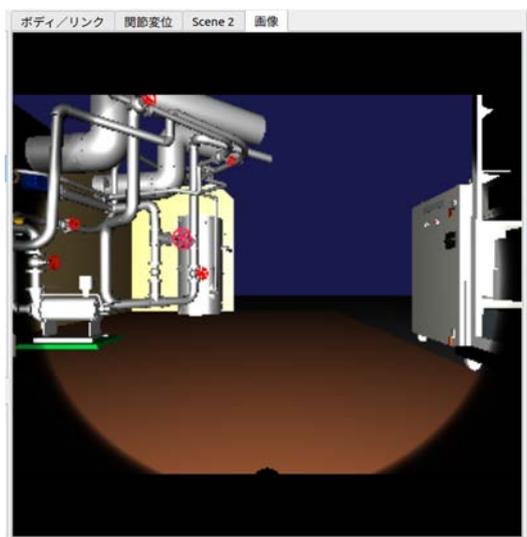


図 4-10 カメラ画像の表示

- (4) アイテムツリービューの「Kinect\_Image」を選択し、右クリックする。

右クリックすると、ポップアップメニューが表示される（図 4-11）。

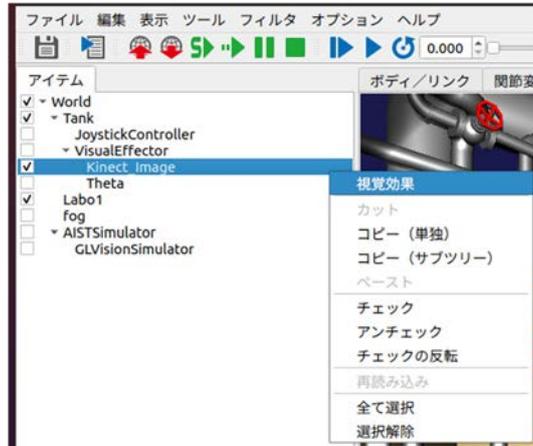


図 4-11 ポップアップメニューの表示

(5) ポップアップメニューの「視覚効果」を選択する。

「視覚効果」を選択すると、「視覚効果の設定」ダイアログが表示される（図 4-12）。



図 4-12 設定ダイアログの表示

(6) 「視覚効果の設定」ダイアログに任意のパラメータを入力する。ここでは例として、「赤」に「0.30」を入力している（図 4-13）。



図 4-13 パラメータの設定（「赤」を入力した場合）

パラメータを入力すると、入力したパラメータに基づく視覚効果が画像ビューに表示されているカメラ画像に付与される。ここでは、「赤」のパラメータを変更しているため、カメラ画像全体の赤色の輝度が大きくなっている（図 4-14）。



図 4-14 視覚効果を付与したカメラ画像の表示

シミュレーション中のロボットに取り付けられているカメラで撮影された画像にノイズ・歪み・色相変化の視覚効果を付与するための設定と操作手順は以上である。

## 5 ロボットの移動軌跡の記録

本章では、シミュレーション中のロボットの移動軌跡を記録するための設定と操作手順を説明する。

---

### この章の主な内容

- ・パッシブマーカの設定
  - ・モーションキャプチャシミュレータの設定
- 

### 5-1 概要

本機能は、ボディ（ロボット）の移動軌跡を記録するための機能である。本機能が提供する「パッシブマーカ」を用いて、パッシブマーカが取り付けられているリンクの位置とその位置にパッシブマーカが到達した時間を記録する。

ここでは、プロジェクト「Crawler.cnoid」を用いて、Crawler の移動軌跡を記録する場合を例に説明する。

尚、説明に用いる「Crawler.cnoid」は、「choreonoid/ext/hairo-world-plugin/sample/tutorial」以下に格納されている。

## 5-2 ロボットの移動軌跡の記録

以下の手順によって、ロボットの移動軌跡が記録される。

### ① プロジェクトファイルの読み込み

- (1) メインメニューの「ファイル」→「プロジェクトを開く」を選択する。
- (2) 「プロジェクトを開く」ダイアログから、「choreonoid/ext/hairo-world-plugin/sample/tutorial」に格納されているファイル「Crawler.cnoid」を選択する。
- (3) 「開く」ボタンを押し、プロジェクトを読み込む。

「開く」ボタンを押すと、以下のようにプロジェクトが読み込まれる（図 5-1）。

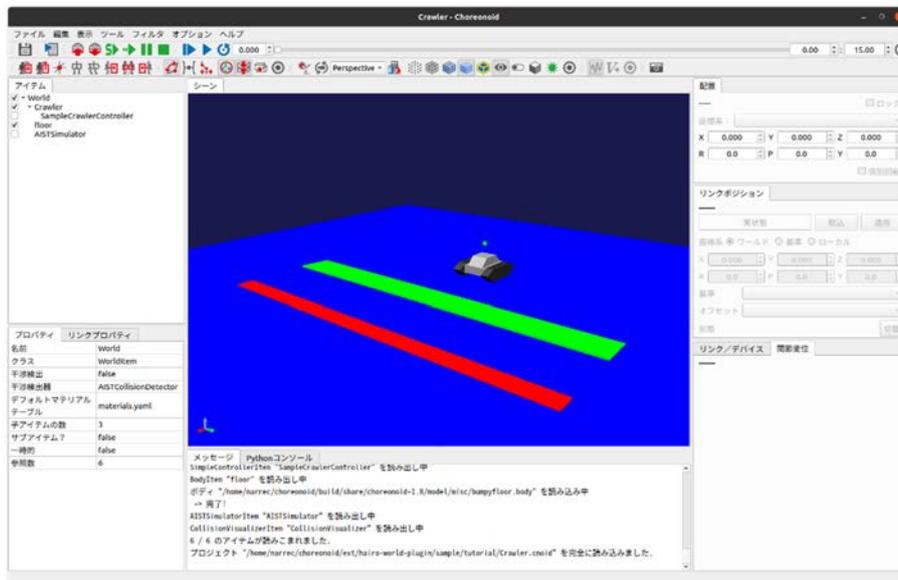


図 5-1 プロジェクトの読み込み

### ② パッシブマーカの設定

パッシブマーカは、ロボットの移動軌跡の記録するときの座標の目印である。パッシブマーカは、個数に制限なく、リンク毎に設定することができる。

- (1) ボディの任意のリンクの elements に以下のように記述する。

```

elements:
-
  type: PassiveMarker
  name: GreenMarker
  translation: [ 0.0, 0.0, 0.3 ]
  radius: 0.03
  color: [ 0.0, 1.0, 0.0 ]
  transparency: 0.3
    
```

パッシブマーカのパラメータは、以下のとおりである（表 5-1）。

表 5-1 パッシブマーカのパラメータ

パラメータ	デフォルト値	単位	意味
type	-	-	デバイスの種類。“PassiveMarker”を入力することで、パッシブマーカが設定される
name	-	-	デバイス名。任意のデバイス名を入力する
radius	1.0	m	パッシブマーカの半径を指定する
color	1.0, 0.0, 0.0	-, -, -	パッシブマーカの色を RGB で指定する
transparency	0.0	-	パッシブマーカの透過度を指定する

### ③ モーションキャプチャシミュレータの設定

モーションキャプチャシミュレータは、パッシブマーカの中心の座標とその座標にパッシブマーカが到達したときの時間を記録するためのものである。

- (1) アイテムツリービューの「AISTSimulator」を選択する。
- (2) メインメニューの「ファイル」→「新規」→「モーションキャプチャシミュレータ」を選択する。
- (3) 「モーションキャプチャシミュレータの新規生成」ダイアログの「作成」ボタンを押す。

「作成」ボタンを押すと、「MotionCaptureSimulator」が「AISTSimulator」の子アイテムとして、アイテムツリービューに登録される（図 5-2）。

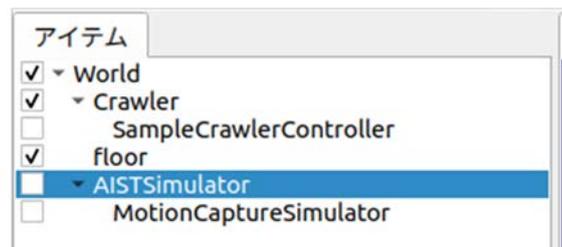


図 5-2 MotionCaptureSimulator の登録

- (4) アイテムツリービューの「MotionCaptureSimulator」を選択する。

「MotionCaptureSimulator」を選択すると、プロパティビューに以下のプロパティが表示される(表 5-2)。

表 5-2 MotionCaptureSimulator のプロパティ

パラメータ	デフォルト値	単位	意味
記録	True	-	True の場合にパッシブマーカの中心座標の記録を有効にする。False に設定した場合は、パッシブマーカの中心座標を記録せず、シーンビューにもパッシブマーカの中心座標を表示しない。また、「記録間隔」・「CSV 出力」のプロパティが非表示となる
記録間隔	0.10	s	パッシブマーカの軌跡を記録する時間間隔を指定する

- (5) プロパティビューの「記録間隔」の「0.10」を選択する。
- (6) 「記録間隔」に任意の値を入力する。

#### ④ ロボットの移動軌跡の記録

- (1) ツールバーの「初期位置からシミュレーションを開始」ボタン  を押す。

シミュレーションが開始すると、「MotionCaptureSimulator」の子アイテムとして、マーカポイント「Marker Points」が生成される(図 5-3)。生成されたこのマーカポイントには、シミュレーション中のパッシブマーカの位置とその位置に到達した時間が記録されている。生成されたマーカポイントの名前は「年/月/日/時間」となる。例えば、2021年9月13日05時03分53秒にパッシブマーカの座標の記録が開始した場合、マーカポイントの名称は「Marker Points - 20210913050353」となる。

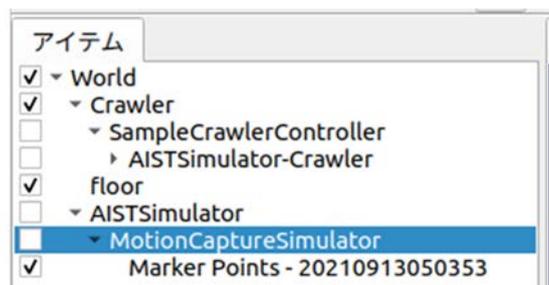


図 5-3 MarkerPoint の生成

(2) ツールバーの「シミュレーションの停止」  を押す。

「シミュレーションの停止」を押すと、マーカポイントに記録されたパッシブマーカの位置がシーンビューに表示される（図 5-4）。アイテムツリーに登録されているマーカポイントのチェックボックスのチェックを外すと非表示となる。

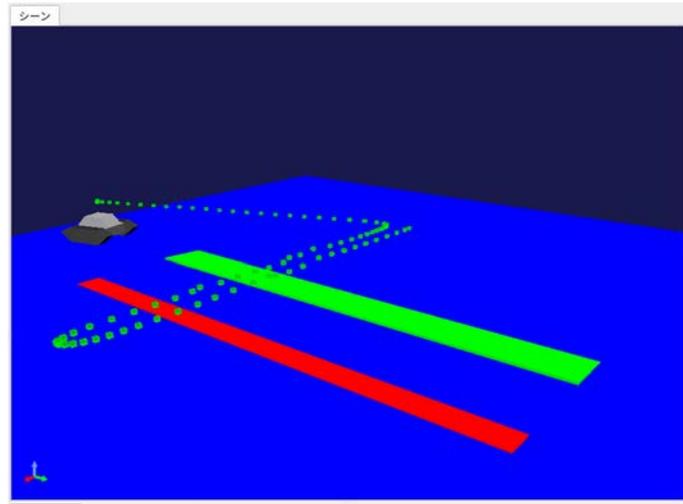


図 5-4 マーカポイントの表示

シミュレーション中のロボットの移動軌跡を記録するための設定と操作手順は以上である。

## 6 ロボットの干渉状態の記録

本章では、シミュレーションに使用しているボディ（ロボット）と他の物体との間の干渉状態を記録・表示するための設定と操作手順を説明する。

---

### この章の主な内容

- ・ コリジョンビジュアライザの設定
  - ・ コリジョンセンサの設定
- 

### 6-1 概要

本機能は、ボディ（ロボット）と他の物体との間の干渉状態をリンク毎に記録するための機能である。また、本機能が提供する「コリジョンセンサ」を用いることで、コリジョンセンサが取り付けられたリンクに干渉が生じたときに、そのリンクの色を任意の色に変更して表示することができる。

本機能では、「コリジョンビジュアライザ」を用いて、前述のリンク毎の干渉状態の記録とコリジョンセンサによるリンクの色の変更するための設定を行う。

ここでは、プロジェクト「Crawler.cnoid」を用いて、Crawler の干渉状態を記録する場合を例に説明する。

尚、説明に用いる「Crawler.cnoid」は、「choreonoid/ext/hairo-world-plugin/sample/tutorial」以下に格納されている。

## 6-2 ロボットの干渉状態の記録

以下の手順によって、ロボットと他の物体との間の干渉状態が記録される。

### ① プロジェクトファイルの読み込み

- (1) メインメニューの「ファイル」→「プロジェクトを開く」を選択する。
- (2) 「プロジェクトを開く」ダイアログから、「chreonoid/ext/hairo-world-plugin/sample/tutorial」に格納されているファイル「Crawler.cnoid」を選択する。
- (3) 「開く」ボタンを押し、プロジェクトを読み込む。

「開く」ボタンを押すと、以下のようにプロジェクトが読み込まれる（図 6-1）。

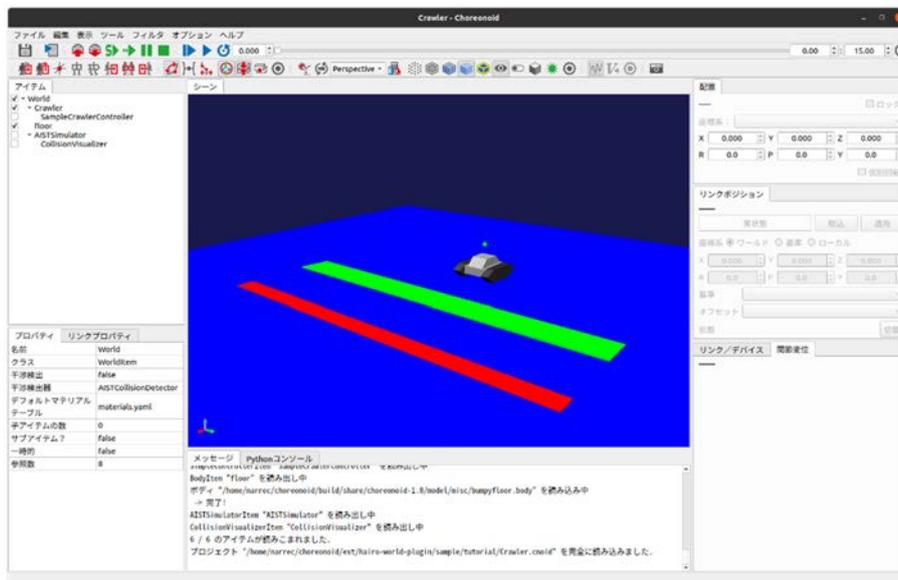


図 6-1 プロジェクトの読み込み

### ② コリジョンビジュアライザの設定

コリジョンビジュアライザは、リンク毎の干渉状態の記録とコリジョンセンサによるリンクの色の変更をするものである。

- (1) アイテムツリービューの「AISTSimulator」を選択する。
- (2) メインメニューの「ファイル」→「新規」→「干渉可視化」を選択する。
- (3) 「コリジョンビジュアライザの新規生成」ダイアログの「作成」ボタンを押す。

「作成」ボタンを押すと、「CollisionVisualizer」が「AISTSimulator」の子アイテムとして、アイテムツリービューに登録される（図 6-2）。

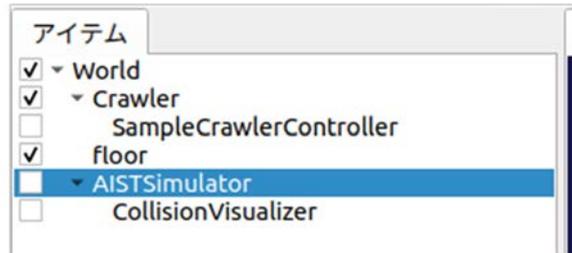


図 6-2 CollisionVisualizer の登録

(4) アイテムツリービューの「CollisionVisualizer」を選択する。

「CollisionVisualizer」を選択すると、プロパティビューに以下のプロパティが表示される（表 6-1）。

表 6-1 CollisionVisualizer のプロパティ

パラメータ	デフォルト値	単位	意味
対象ボディ		-	干渉を可視化するボディを指定する。指定がない場合は、全てのボディが対象になる。複数選択する場合は、スペース区切りで指定する
干渉の記録	False	-	True の場合に干渉を記録する

(5) プロパティビューの「干渉の記録」を選択し、「True」を選択する（図 6-3）。

プロパティ	リンクプロパティ
名前	CollisionVisualizer
クラス	CollisionVisualizerItem
有効	True
対象ボディ	
干渉の記録	True
子アイテムの数	0
サブアイテム?	False
一時的	False
参照数	11

図 6-3 CollisionVisualizer のプロパティの設定

### ③ コリジョンセンサの設定

コリジョンセンサは、このセンサが取り付けられているリンクに干渉が生じたときに、ユーザがコリジョンセンサのパラメータで指定した色にリンクの色を変更するためのものである。

尚、このコリジョンセンサの設定は任意であり、干渉が生じたときにリンクの色を変更しない場合は不要である。

(1) ボディの任意のリンクの elements に以下のように記述する。

```
elements:
-
  type: CollisionSensor
  name: CollisionSensor
  translation: [ 0.0, 0.0, 0.0 ]
  rotation: [ 0, 1, 0, 0 ]
  color: [ 1.0, 0.0, 0.0 ]
```

コリジョンセンサのパラメータは、以下のとおりである (表 6-2)。

表 6-2 コリジョンセンサのパラメータ

パラメータ	デフォルト値	単位	意味
color	1.0, 0.0, 0.0	-	干渉しているときのリンクの色を RGB で指定する

### ④ ロボットの干渉状態の記録

(1) ツールバーの「初期位置からシミュレーションを開始」を押す。

シミュレーションが開始すると、ボディの干渉を記録したアイテム「Collision States」が「CollisionVisualizer」の子アイテムとして生成される (図 6-4)。ただし、コリジョンビジュアライザの「干渉の記録」が「False」の場合は、「Collision States」は生成されない。

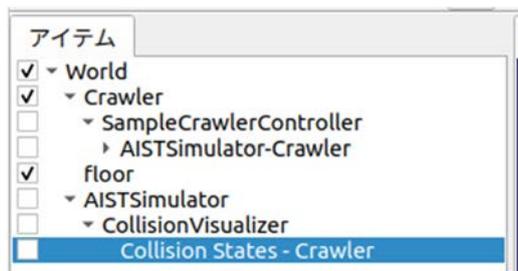


図 6-4 Collision States の生成

また、ボディにコリジョンセンサが取り付けられている場合、コリジョンセンサが取り付けられているリンクの色がパラメータで指定した色に変更される（図 6-5、図 6-6）。ここでは例として、Crawlerの左右のクローラにコリジョンセンサを取り付け、干渉が生じたときに赤色に変更されるように設定している。

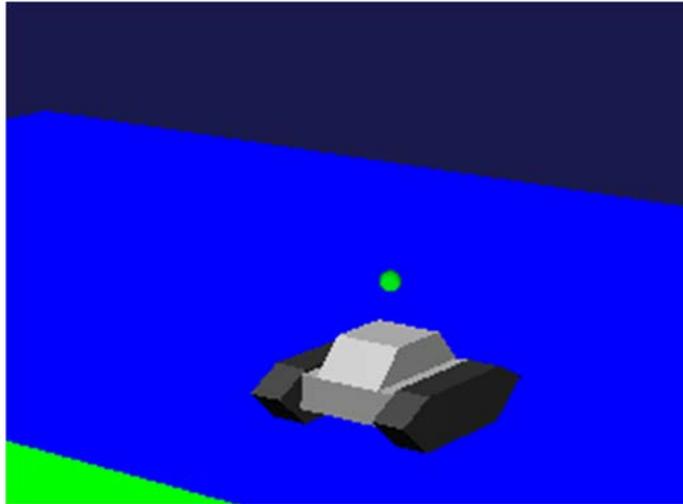


図 6-5 干渉が生じる前の表示例

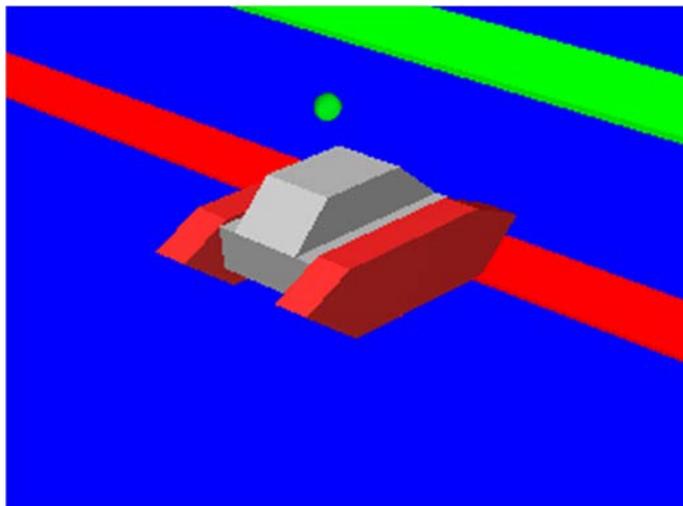


図 6-6 干渉が生じた後の表示例

## 6-3 ロボットの干渉状態の保存

以下の手順によって、ロボットと他の物体との間の干渉状態を保存する。

- (1) アイテムツリービューの「Collision States」を選択する。
- (2) メインメニューの「ファイル」→「選択したアイテムのエクスポート」を選択する。
- (3) ダイアログに任意のファイル名と保存先を指定する。
- (4) ダイアログの「保存」ボタンを押す。

「保存」ボタンを押すと、ロボットの干渉状態がファイルに保存される。保存したファイルには、アイテムツリーに登録されている順番でロボットの干渉状態が記録される。例えば、リンクが2つ設定されているロボットAとロボットBの干渉状態を記録した場合は、ロボットAのリンク1、ロボットAのリンク2、ロボットBのリンク1、ロボットBのリンク2の順番で記録される。

## 6-4 ロボットの干渉状態のグラフ表示

以下の手順によって、ロボットと他の物体との間の干渉状態をグラフで表示する。

- (1) メインメニューの「表示」→「ビューの表示」→「Multi Value Seq」を選択する。

「Multi Value Seq」を選択すると、以下の「Multi Value Seq」ビューが表示される（図 6-7）。

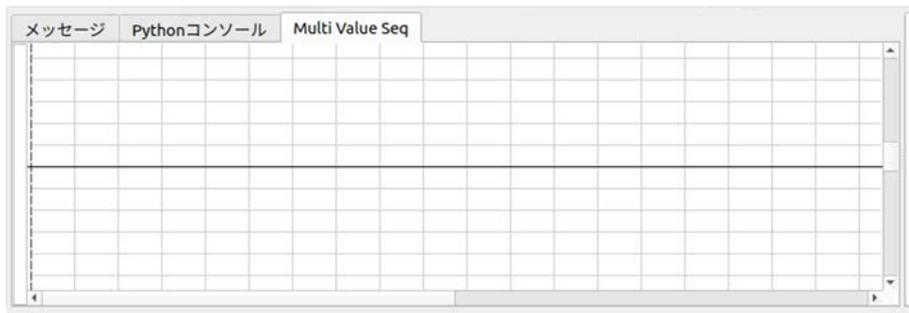


図 6-7 Multi Value Seq ビューの表示

- (2) アイテムツリーの「CollisionVisualizer」を選択し、左の「▶」を押す（図 6-8）。

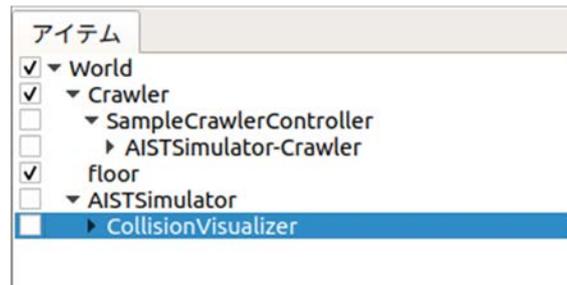


図 6-8 CollisionVisualizer の選択

「CollisionVisualizer」左の「▶」を押すと、ロボットの干渉状態が記録された「Collision States」が表示される（図 6-9）。

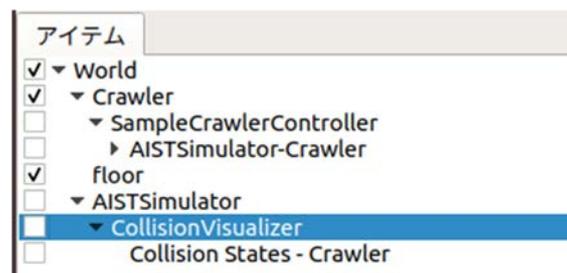


図 6-9 Collision States の表示

- (3) アイテムツリーの「Collision States」を選択する。
- (4) Multi Value Seq ビューの左側に表示されている数字を選択する。

Multi Value Seq ビューの左側に表示されている数字は、アイテムツリーに登録されているロボットのリンクに対応している。例えば、リンクが2つ設定されているロボットAとロボットBの干渉状態を記録した場合は、0から順番に、ロボットAのリンク1、ロボットAのリンク2、ロボットBのリンク1、ロボットBのリンク2の順番で数字が割り当てられている。

数字を選択すると、対応しているリンクの干渉状態がMulti Value Seq ビューに表示される（図6-10）。

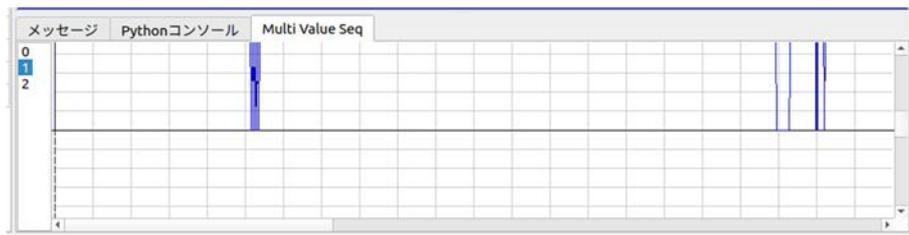


図 6-10 干渉状態の表示

シミュレーションに使用しているボディ（ロボット）の干渉状態を記録・表示するための設定と操作手順は以上である。

## 7 ジョイスティックの入力状態の表示

本章では、ジョイスティックの入力状態を表示するための設定と操作手順を説明する。

---

### この章の主な内容

- ・ジョイスティックの入力状態の表示
- 

### 7-1 概要

本機能は、使用している PC に接続されているジョイスティックのスティックとボタンの入力状態を表示するための機能である。また、本機能を利用して、キーボードからジョイスティックの入力を行うこともできる。

尚、本機能では最初に接続されたジョイスティック (`/dev/input/js0`) の入力状態のみ表示できる。

## 7-2 ジョイスティックの入力状態の表示

以下の手順によって、ジョイスティックの入力状態が表示される。

(1) メインメニューの「表示」→「ビューの表示」→「ジョイスティック状態」を選択する。

「ジョイスティック状態」を選択すると、「ジョイスティック状態」ビューが以下のように表示される（図 7-1）。



図 7-1 「ジョイスティック状態」ビュー

「ジョイスティック状態」ビューのパラメータと、ジョイスティックのスティック及びボタンは以下のように割り当てられている（表 7-1）。

表 7-1 ジョイスティック (DUALSHOCK®4) の入力とビューの表示の割り当て

DUALSHOCK®4の入力	「ジョイスティック状態」ビューの表示
左スティック (横)	「F」ボタン (正)、「S」ボタン (負)、「L_STICK_H_AXIS」バー
左スティック (縦)	「D」ボタン (正)、「E」ボタン (負)、「L_STICK_V_AXIS」バー
L2トリガ	「Q」ボタン、「L_TRIGGER_AXIS」バー
右スティック (横)	「L」ボタン (正)、「J」ボタン (負)、「R_STICK_H_AXIS」バー
右スティック (縦)	「K」ボタン (正)、「I」ボタン (負)、「R_STICK_V_AXIS」バー
R2トリガ	「P」ボタン、「R_TRIGGER_AXIS」バー
十字キー (横)	「>」ボタン (正)、「<」ボタン (負)、「DIR_PAD_H_AXIS」バー
十字キー (縦)	「^」ボタン (正)、「v」ボタン (負)、「DIR_PAD_V_AXIS」バー
×ボタン	「A」ボタン
○ボタン	「B」ボタン
□ボタン	「X」ボタン
△ボタン	「Y」ボタン
L1ボタン	「W」ボタン
R1ボタン	「O」ボタン
SHAREボタン	「G」ボタン
OPTIONSボタン	「H」ボタン
ロゴボタン	「_ (スペース)」ボタン
左スティック (押し込み)	「C」ボタン
右スティック (押し込み)	「M」ボタン

各ボタンの入力状態は、「ジョイスティック状態」ビュー左のボタンに表示される。

例として、DUALSHOCK®4の「○」ボタンを押すと、「ジョイスティック状態」ビューの「B」ボタンが「赤色」になる (図 7-2)。



図 7-2 DUALSHOCK®4 の○ボタンを押した場合

また、各スティックに入力状態は、「ジョイスティック状態」ビュー左のボタンと右のバーに表示される。スティックの入力が負の向きの場合は、バーの色が「マゼンタ色」、正の向きの場合は、「シアン色」で表示される。また、スティックの入力の具合は、同バーに0~100%の範囲で表示される。

例として、DUALSHOCK®4 の左スティックを最大まで奥に倒した（負の入力）場合は、「ジョイスティック状態」ビューの「L\_STICK\_V\_AXIS」右のバーが「マゼンタ色」になり、「100%」が表示される。また、「E」ボタンが赤色になる（図 7-3）。



図 7-3 DUALSHOCK®4 の左スティックを奥に倒した場合

また反対に、左スティックを最大まで手前に倒した（正の入力）場合は、「ジョイスティック状態」ビューの「L\_STICK\_V\_AXIS」右のバーが「シアン色」になり、「100%」が表示される。また、「D」ボタンが赤色になる（図 7-4）。



図 7-4 DUALSHOCK®4 の左スティックを手前に倒した場合

ジョイスティックの入力状態を表示するための設定と操作手順は以上である。

## 8 ジョイスティックの入力状態の記録

本章では、シミュレーション中のジョイスティックの入力状態を記録するための設定と操作手順を説明する。

---

### この章の主な内容

- ・ ジョイスティックロガーの設定
  - ・ ジョイスティックの入力状態の書き出し
  - ・ 記録したジョイスティックの入力状態の表示
- 

### 8-1 概要

本機能は、使用している PC に接続されているジョイスティックのスティックとボタンの入力状態を記録するための機能である。

本機能では、「ジョイスティックロガー」を用いて、ジョイスティックの入力状態を記録するための設定を行う。

ここでは、プロジェクト「Tank.cnoid」を用いて、シミュレーション中のジョイスティックの入力状態を記録する場合を例に説明する。

尚、説明に用いる「Tank.cnoid」は、「choreonoid/ext/hairo-world-plugin/sample/tutorial」以下に格納されている。また、本機能では最初に接続されたジョイスティック (`/dev/input/js0`) の入力状態のみ表示できる。

## 8-2 ジョイスティックの入力状態の記録

ジョイスティックロガーは、シミュレーション中のジョイスティックの入力状態を記録するものである。

以下の手順によって、ジョイスティックロガーが設定される。

### ① プロジェクトファイルの読み込み

- (1) メインメニューの「ファイル」→「プロジェクトを開く」を選択する。
- (2) 「プロジェクトを開く」ダイアログから、「chreonoid/ext/hairo-world-plugin/sample/tutorial」に格納されているファイル「Tank.cnoid」を選択する。
- (3) 「開く」ボタンを押し、プロジェクトを読み込む。

「開く」ボタンを押すと、以下のようにプロジェクトが読み込まれる（図 8-1）。

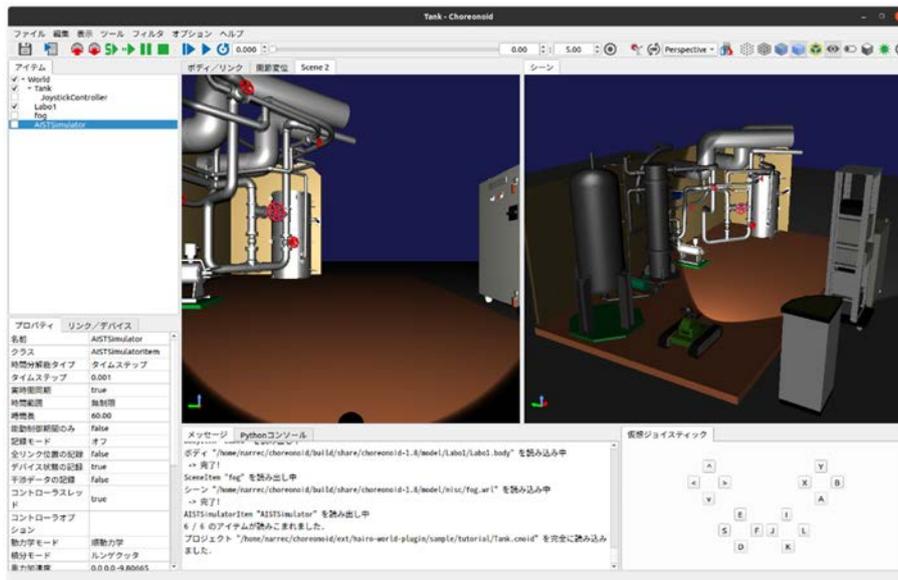


図 8-1 プロジェクトの読み込み

### ② ジョイスティックロガーの設定

- (1) アイテムツリービューの「AISTSimulator」を選択する。
- (2) メインメニューの「ファイル」→「新規」→「ジョイスティックロガー」を選択する。
- (3) 「ジョイスティックロガーの新規生成」ダイアログの「作成」ボタンを押す。

「作成」ボタンを押すと、「JoystickLogger」が「AISTSimulator」の子アイテムとして、アイテムツリービューに登録される（図 8-2）。

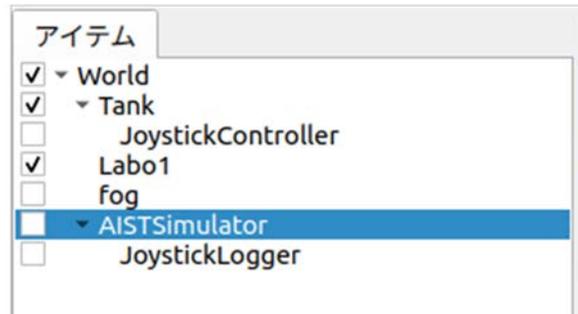


図 8-2 JoystickLogger の登録

(4) アイテムツリービューの「JoystickLogger」を選択する。

「JoystickLogger」を選択すると、プロパティビューに以下のプロパティが表示される（表 8-1）。

パラメータ	デフォルト値	単位	意味
ジョイスティック状態の記録	False	-	True の場合にジョイスティック状態を記録する

(5) プロパティビューの「ジョイスティック状態の記録」を選択し、「True」を選択する（図 8-3）。

プロパティ	リンク/デバイス
名前	JoystickLogger
クラス	JoystickLoggerItem
有効	True
ジョイスティック状態の記録	True
子アイテムの数	0
サブアイテム?	False
一時的	False
参照数	11

図 8-3 JoystickLogger のプロパティの設定

(6) ツールバーの「初期位置からシミュレーションを開始」ボタン  を押す。

シミュレーションが開始されると、ジョイスティック状態を記録した「Joystick States」が「JoystickLogger」の子アイテムとして生成される（図 8-4）。ただし、ジョイスティックロガーの「ジョイスティック状態の記録」が「False」の場合は、「Joystick States」は生成されない。



図 8-4 Joystick States の生成

### 8-3 ジョイスティックの入力状態の保存

以下の手順によって、ジョイスティックの入力状態が保存される。

- (1) アイテムツリービューの「Joystick States」を選択する。
- (2) メインメニューの「ファイル」→「選択したアイテムのエクスポート」を選択する。
- (3) ダイアログに任意のファイル名と保存先を指定する。
- (4) ダイアログの「保存」ボタンを押す。

「保存」ボタンを押すと、記録したジョイスティックの入力状態がファイルに保存される。保存したファイルには、記録したジョイスティックの入力状態が、全スティック-全ボタンの順番で記録される。例えば、スティック (Axis) が 8 入力、ボタン (Button) が 13 入力のジョイスティックを使用している場合は、Axis-0, Axis-1, ..., Axis-7, Button-0, Button-1, ..., Button-12 の順番で記録される。

## 8-4 ジョイスティック状態のグラフ表示

以下の手順によって、ジョイスティックの入力状態がグラフで表示される。

- (1) メインメニューの「表示」→「ビューの表示」→「Multi Value Seq」を選択する。

「Multi Value Seq」を選択すると、以下の「Multi Value Seq」ビューが表示される（図 8-5）。

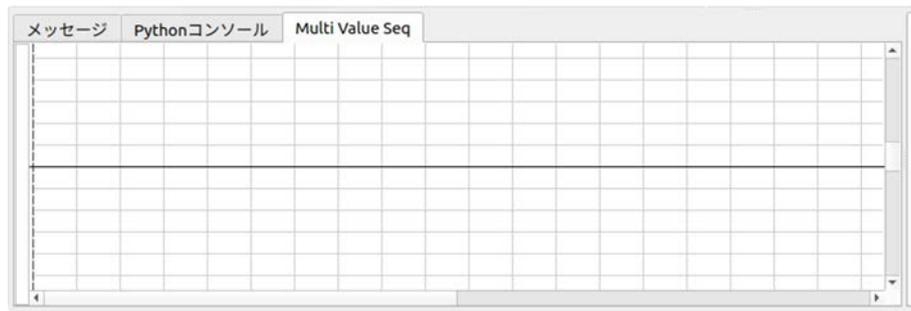


図 8-5 Multi Value Seq ビューの表示

- (2) アイテムツリーの「JoystickLogger」を選択し、左の「▶」を押す（図 8-6）。

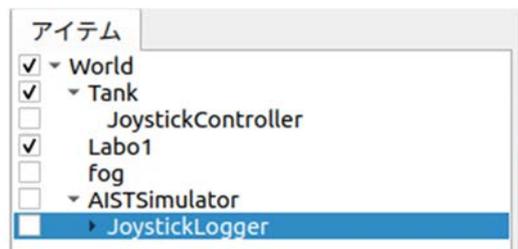


図 8-6 JoystickLogger の選択

「JoystickLogger」左の「▶」を押すと、ジョイスティックの入力状態が記録された「Joystick States」が表示される（図 8-7）。

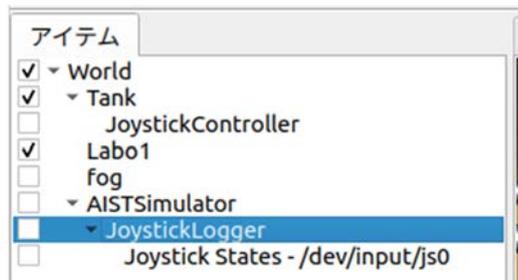


図 8-7 Joystick States の表示

- (3) アイテムツリーの「Joystick States」を選択する。
- (4) Multi Value Seq ビューの左側に表示されている数字を選択する。

Multi Value Seq ビューの左側に表示されている数字は、ジョイスティックの各スティック・ボタンに対応しており、全スティック-全ボタンの順番で割り当てられている。例えば、スティック (Axis) が 8 入力、ボタン (Button) が 13 入力のジョイスティックを使用している場合は、Axis-0, Axis-1, ..., Axis-7, Button-0, Button-1, ..., Button-12 の順番で数字が割り当てられ、0 から 7 はスティック、8 から 20 はボタンに対応している。

数字を選択すると、対応しているスティック、またはボタンの入力状態が Multi Value Seq ビューに表示される (図 8-8)。

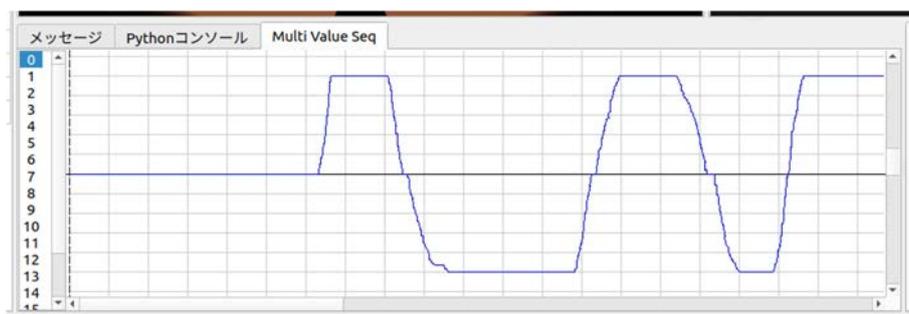


図 8-8 ジョイスティックの入力状態の表示

シミュレーション中のジョイスティックの入力状態を記録するための設定と操作手順は以上である。

## 9 ジョイスティックによるシミュレーションの操作

本章では、ジョイスティックを用いたシミュレーションの開始・一時停止・再開・停止と、プロジェクトを読み込むためのダイアログの呼び出しのための設定と操作手順を説明する。

### この章の主な内容

- ・ジョイスティックの操作によるシミュレーションの開始・一時停止・再開・停止
- ・プロジェクトを読み込むためのダイアログの呼び出し

### 9-1 概要

本機能は、ジョイスティックの「START/OPTIONS」・「BACK/SHARE」ボタンを用いてシミュレーションの開始・一時停止・再開・停止と、「LOGO」ボタンを使用してプロジェクトを開くダイアログの呼び出しを行うための機能である。

ジョイスティックによるシミュレーションの操作には、SONY の「DUALSHOCK<sup>®</sup> 4」、Microsoft<sup>®</sup> の「Xbox360<sup>®</sup> Controller」、Logicool の「F310r GAMEPAD<sup>®</sup>」の3種類のジョイスティックが使用できる。本機能でシミュレーションの操作に使用する「START/OPTIONS」・「BACK/SHARE」とダイアログの呼び出しに使用する「LOGO」のボタンは以下のとおりである（図 9-1）。

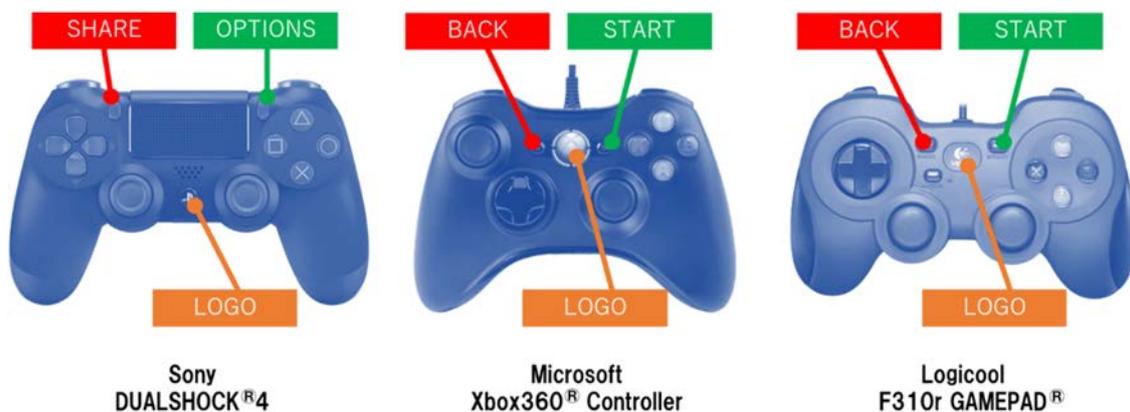


図 9-1 ジョイスティックの種類とボタンの位置

本機能によって提供される動作と、その時のジョイスティックの操作は以下のようになっている（表 9-1）。

表 9-1 ジョイスティックによる操作の種類

動作の種類	ジョイスティックの操作
初期位置からシミュレーションを開始	シミュレーションが「停止」しているときに「START/OPTIONS」ボタンを1回押す
現在位置からシミュレーションを開始	シミュレーションが「停止」しているときに「BACK/SHARE」ボタンを1回押す
シミュレーションの停止	シミュレーションが「一時停止」しているとき、またはシミュレーション中に「BACK/SHARE」ボタンを1回押す
シミュレーションの一時停止	シミュレーション中に「START/OPTIONS」ボタンを1回押す
現在位置からシミュレーションを再開	シミュレーションが「一時停止」しているときに「START/OPTIONS」ボタンを1回押す
プロジェクトを開くためのダイアログの呼び出し	「LOGO」ボタンを1回押す

前述のジョイスティックの操作によって、ユーザはシミュレーションの状態を以下のように遷移させることができる（図 9-2）。

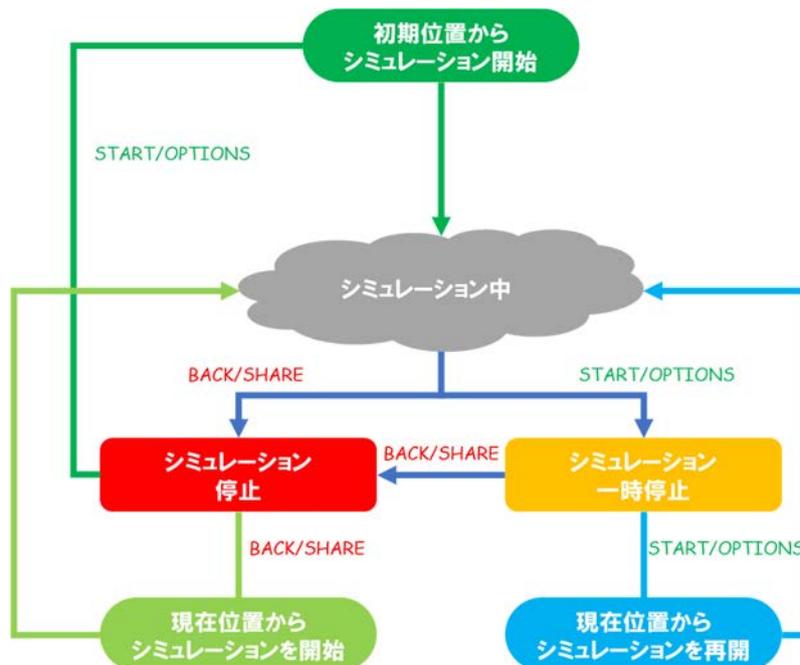


図 9-2 シミュレーションの状態遷移

ここでは、プロジェクト「Tank.cnoid」を用いて、ジョイスティックの操作によってシミュレーションを開始・一時停止・再開・停止させる場合を例に説明する。加えて、ジョイスティックの操作によって、プロジェクトを読み込むためのダイアログを呼び出して、「Tank.cnoid」を読み込む例について説明する。

尚、説明に用いる「Tank.cnoid」は、「choreonoid/ext/hairo-world-plugin/sample/tutorial」以下に格納されている。また、本機能では、前述の「START/OPTIONS」・「BACK/SHARE」・「LOGO」ボタンを占有するため、他の用途にこれらのボタンを使用する場合は、本機能は使用しないようにしていただきたい。

## 9-2 機能の有効化／無効化

以下の手順によって、本機能が有効化／無効化される。

- (1) メインメニューの「オプション」→「ジョイスティック」→「シミュレーションを開始（スタートボタン）」を選択する（図 9-3）。

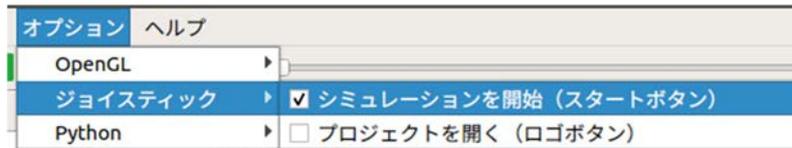


図 9-3 ジョイスティックによるシミュレーションの操作の有効化

「シミュレーションを開始（スタートボタン）」左のチェックボックスにチェックが入っているときに本機能が有効化され、ジョイスティックを用いてシミュレーションを開始・一時停止・再開・停止させることが可能となる。チェックを外すと無効化される。

- (2) メインメニューの「オプション」→「ジョイスティック」→「プロジェクトを開く（ロゴボタン）」を選択する（図 9-4）。

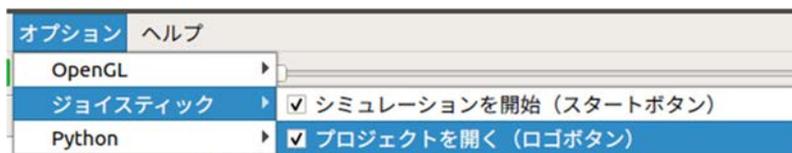


図 9-4 ジョイスティックによるプロジェクトを開くためのダイアログ呼び出しの有効化

「プロジェクトを開く（ロゴボタン）」左のチェックボックスにチェックが入っているときに本機能が有効化され、プロジェクトを開くダイアログの呼び出しが可能となる。チェックを外すと無効化される。

## 9-3 ジョイスティックによるシミュレーションの操作

以下の手順によって、シミュレーションが操作される。

### ① プロジェクトファイルの読み込み

- (1) メインメニューの「ファイル」→「プロジェクトを開く」を選択する。
- (2) 「プロジェクトを開く」ダイアログから、「choreonoid/ext/hairo-world-plugin/sample/tutorial」に格納されているファイル「Tank.cnoid」を選択する。
- (3) 「開く」ボタンを押し、プロジェクトを読み込む。

「開く」ボタンを押すと、以下のようにプロジェクトが読み込まれる（図 9-5）。

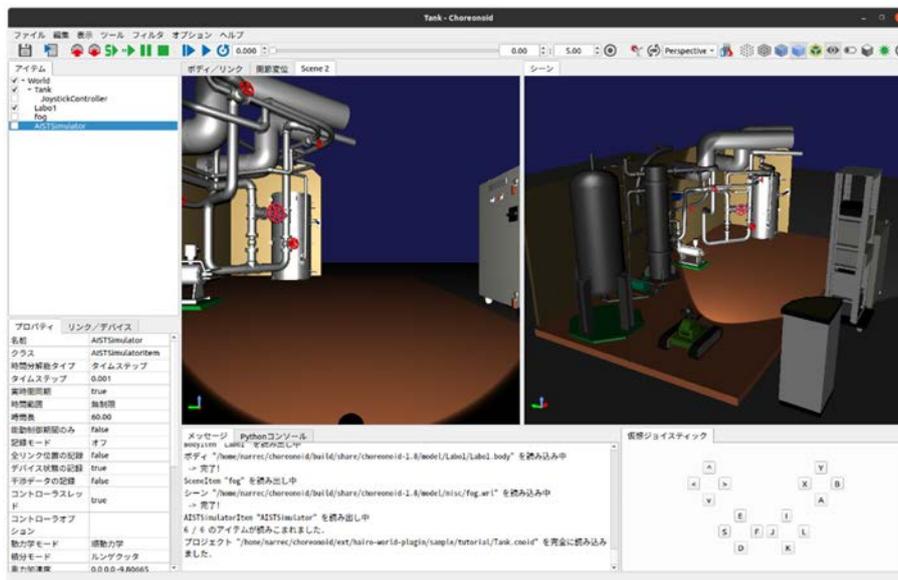


図 9-5 プロジェクトの読み込み

### ② 初期位置からのシミュレーションの開始

- (1) シミュレーションが「停止」しているとき（または開始前）にジョイスティックの「START/OPTIONS」ボタンを押す。

「START/OPTIONS」ボタンを押すと、初期位置からシミュレーションが開始する。

尚、このときの挙動は、ツールバーの「初期位置からシミュレーションを開始」ボタン  を押した場合と同じである。

### ③ シミュレーションの一時停止・再開

- (1) シミュレーション中にジョイスティックの「START/OPTIONS」ボタンを押す。

シミュレーション中に「START/OPTIONS」ボタンを押すと、シミュレーションが一時停止する。また、シミュレーションが「一時停止」しているときに「START/OPTIONS」ボタンを押すと、シミュレーションが再開する。

尚、このときの挙動は、ツールバーの「シミュレーションの一時停止」ボタン  を押した場合と同じである。

### ④ シミュレーションの停止

- (1) シミュレーション中に（または「一時停止」しているとき）にジョイスティックの「BACK/SHARE」ボタンを押す。

「BACK/SHARE」ボタンを押すと、シミュレーションが停止する。

尚、このときの挙動は、ツールバーの「シミュレーションの停止」ボタン  を押した場合と同じである。

### ⑤ 現在位置からのシミュレーションの開始

- (1) シミュレーションが「停止」しているときにジョイスティックの「BACK/SHARE」ボタンを押す。

「BACK/SHARE」ボタンを押すと、現在位置からシミュレーションが開始する。

尚、このときの挙動は、ツールバーの「現在位置からシミュレーションを開始」ボタン  を押した場合と同じである。

ジョイスティックの操作によるシミュレーションの開始・一時停止・再開・停止のための設定と操作手順は以上である。

## 9-4 ダイアログの呼び出し

以下の手順によって、プロジェクトを読み込むためのダイアログが呼び出される。

(1) ジョイスティックの「LOGO」ボタンを押す。

「LOGO」ボタンを押すと、ダイアログ「Choreonoidプロジェクトファイルの読み込み」が表示される（図 9-6）。表示されたダイアログで任意のプロジェクトファイルを選択し、「読み込み」ボタンを押すと、プロジェクトが読み込まれる。

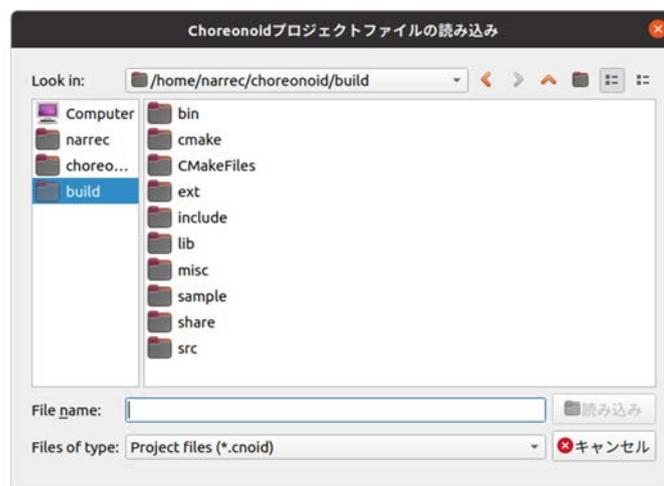


図 9-6 Choreonoid プロジェクトファイルの読み込みのダイアログの表示

プロジェクトを読み込むためのダイアログの呼び出しの設定と操作手順は以上である。

## 10 ファイル／ディレクトリの表示

本章では、シミュレーションに使用しているモデルの設定が記述されているファイル、または、そのファイルが格納されているディレクトリを簡易な操作で開くための設定と操作手順を説明する。

---

### この章の主な内容

- ・モデルの設定が記述されているファイルの表示
  - ・モデルの設定が記述されているファイルが格納されているディレクトリの表示
- 

### 10-1 概要

本機能は、アイテムツリービューに登録されているボディの設定が記述されたファイル、または、そのファイルが格納されているディレクトリを表示するための機能である。

ここでは、プロジェクト「Tank.cnoid」を用いて、プロジェクト内で使用されているボディ「Tank」の設定が記述されたファイル及びそのファイルが格納されているディレクトリを開くための操作手順を説明する。

尚、説明に用いる「Tank.cnoid」は、「choreonoid/ext/hairo-world-plugin/sample/tutorial」以下に格納されている。

## 10-2 ファイル/ディレクトリの表示

以下の手順によって、ファイルまたはディレクトリが表示される。

- (1) ボディをアイテムツリービューで選択し、右クリックする。ここでは例として、「Tank」を選択する。

右クリックすると、以下のようにポップアップメニューが表示される（図 10-1）。



図 10-1 ポップアップメニューの表示

- (2) ファイルを表示する場合は、ポップアップメニューの「開く」→「ファイル」を選択し（図 10-2）、ディレクトリを表示する場合は、「開く」→「ディレクトリ」を選択する（図 10-3）。



図 10-2 メニュー「ファイル」の選択

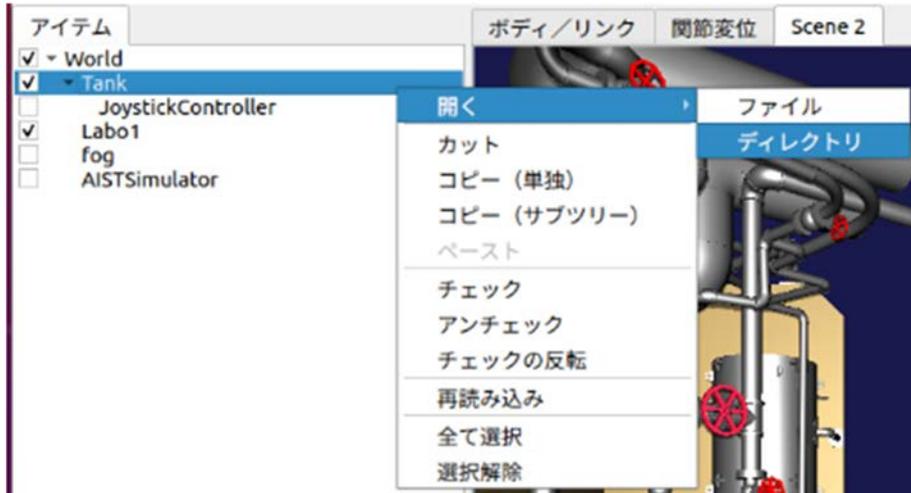


図 10-3 メニュー「ディレクトリ」の選択

「ファイル」を選択すると、アイテムツリービューで選択した「Tank」の設定が記述されているファイル「Tank.body」がテキストエディタで以下のように表示される（図 10-4）。

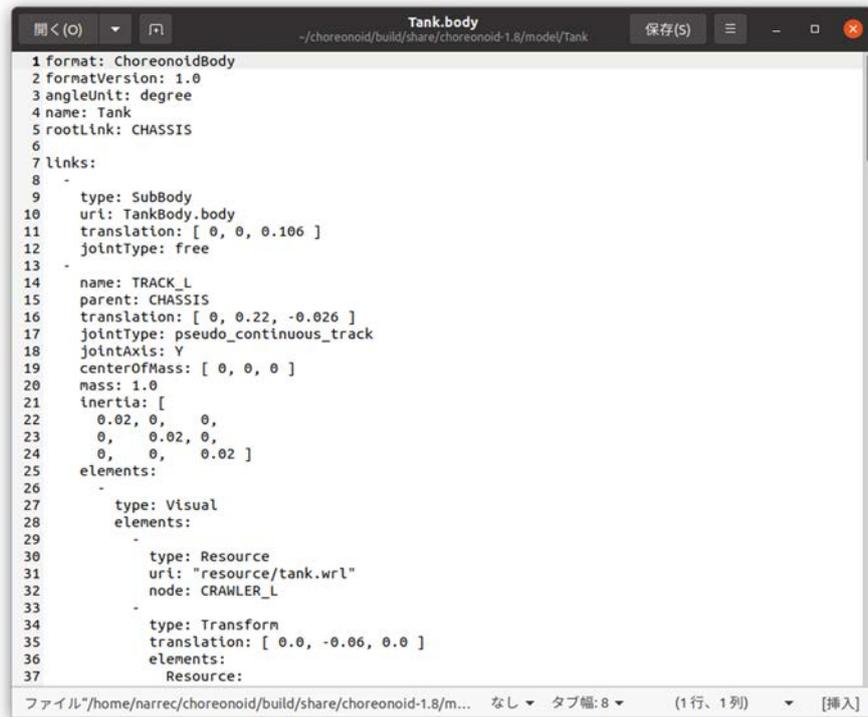


図 10-4 ファイルの表示

「ディレクトリ」を選択すると、アイテムツリービューで選択した「Tank」の設定が記述されているファイル「Tank.body」が格納されているディレクトリが以下のように表示される（図 10-5）。

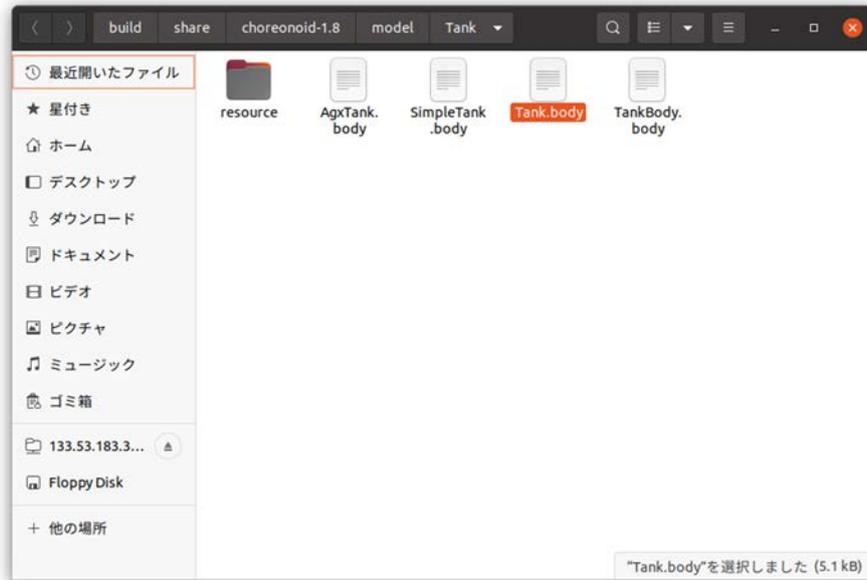


図 10-5 ディレクトリの表示

シミュレーションに使用しているモデルの設定が記述されているファイル及びそのファイルが格納されているディレクトリを表示するための設定と操作手順は以上である。

# 11 プロジェクトのブックマーク登録

本章では、プロジェクトの「ブックマーク」リストへの登録と、「ブックマーク」リストに登録されているプロジェクトを読み込むための設定と操作手順を説明する。

---

## この章の主な内容

- ・プロジェクトの「ブックマーク」リストへの登録・削除
  - ・「ブックマーク」リストに登録されたプロジェクトの読み込み
- 

## 11-1 概要

本機能は、使用頻度の高いプロジェクトを「ブックマーク」リストに登録し、簡易な操作で読み込むための機能である。登録したプロジェクトは記録され、Choreonoid を起動したときに自動的に読み込まれる。

ここでは、プロジェクト「Tank.cnoid」の「ブックマーク」リストへの登録と読み込み、更に「ブックマーク」リストに登録した「Tank.cnoid」の削除を例に説明する。

尚、説明に用いる「Tank.cnoid」は、「choreonoid/ext/hairo-world-plugin/sample/tutorial」以下に格納されている。

## 11-2 ブックマークマネージャの表示

本機能では、プロジェクトの「ブックマーク」リストを、「ブックマークマネージャ」を用いて作成する。この「ブックマークマネージャ」に任意のプロジェクトを登録しておくことで、使用頻度の高いプロジェクトを簡易な操作で読み込むことができる。

以下の手順によって、ブックマークマネージャが表示される。

- (1) メインメニューの「表示」 - 「ツールバーの表示」から「ブックマークバー」を選択する（図 11-1）。

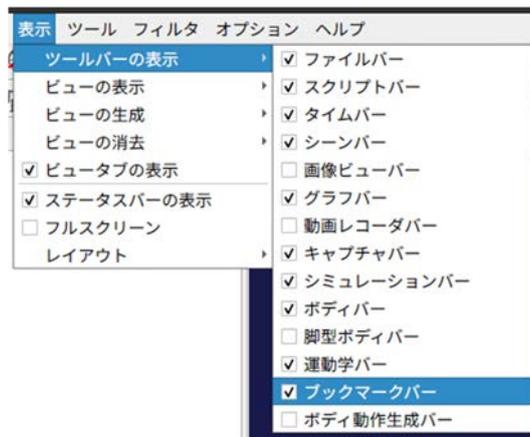


図 11-1 ブックマークバーの表示

「ブックマークバー」を選択すると、ツールバーに  ボタンが表示される。

- (2) ブックマークバーの  ボタンを押す。

ボタンを押すと、ブックマークマネージャが表示される（図 11-2）。

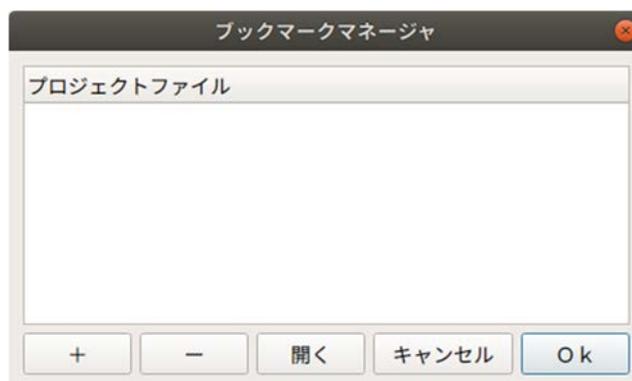


図 11-2 ブックマークマネージャの表示

## 11-3 プロジェクトの登録

以下の手順によって、ブックマークマネージャにプロジェクトが登録される。

- (1) ブックマークマネージャの「+」ボタンを押す。
- (2) 「Choreonoid プロジェクトファイルの選択」ダイアログから、登録するプロジェクトファイルを選択する（図 11-3）。ここでは例として、「choreonoid/ext/hairo-world-plugin/sample/tutorial/Tank.cnoid」を選択している。

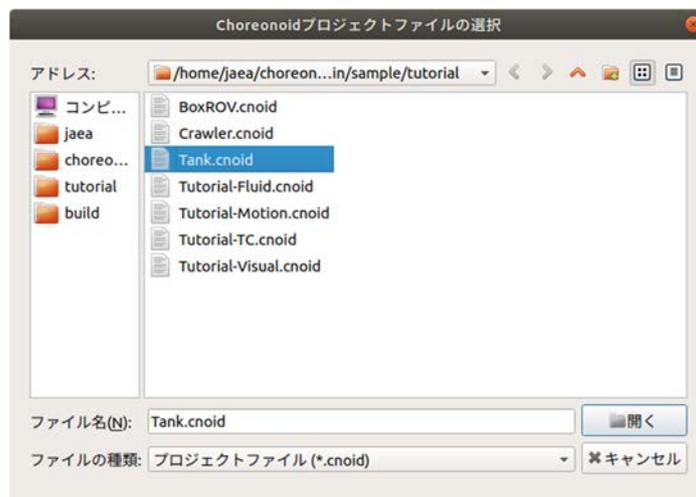


図 11-3 Choreonoid プロジェクトファイルの選択のダイアログ

- (3) ブックマークマネージャの「開く」ボタンを押す。

「開く」ボタンを押すと、選択したプロジェクトがブックマークマネージャに登録される（図 11-4）。

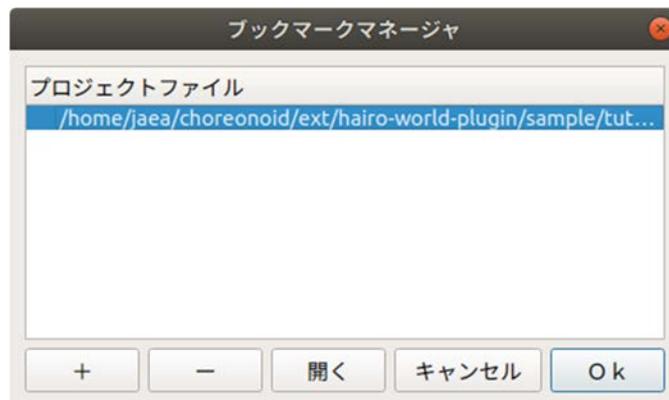


図 11-4 「ブックマーク」リストへの登録

## 11-4 プロジェクトの読み込み

以下の手順によって、ブックマークマネージャに登録されているプロジェクトが読み込まれる。

- (1) ブックマークマネージャに表示されている任意のプロジェクトを選択する。ここでは例として、「choreonoid/ext/hairo-world-plugin/sample/tutorial/Tank.cnoid」を選択している（図 11-5）。

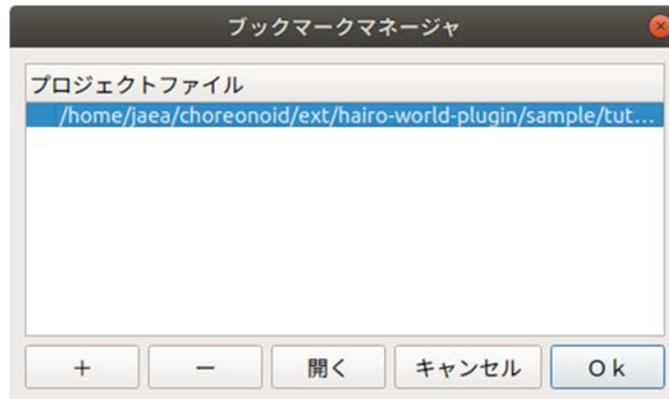


図 11-5 プロジェクトの選択

- (2) ブックマークマネージャの「開く」ボタンを押す。

「開く」ボタンを押すと、選択したプロジェクトが読み込まれる（図 11-6）。

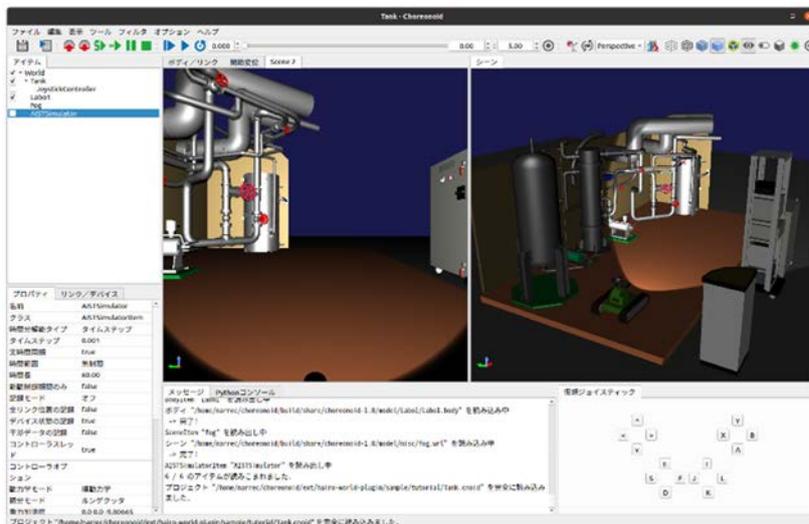


図 11-6 プロジェクトの読み込み

## 11-5 プロジェクトの削除

以下の手順によって、ブックマークマネージャに登録されたプロジェクトが削除される。

- (1) ブックマークマネージャに表示されている任意のプロジェクトを選択する。ここでは例として、「choreonoid/ext/hairo-world-plugin/sample/tutorial/Tank.cnoid」を選択している（図 11-7）。

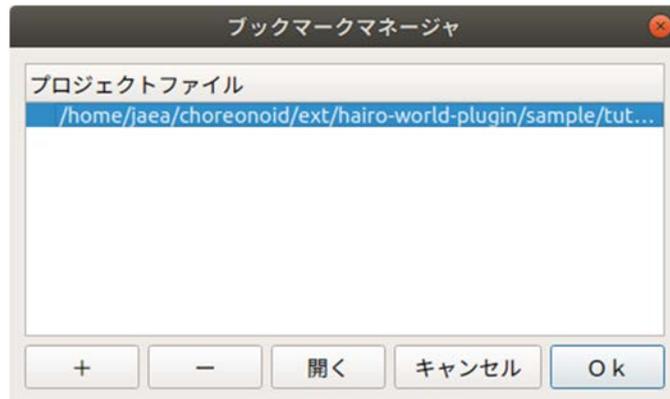


図 11-7 プロジェクトの選択

- (2) ブックマークマネージャの「-」ボタンを押す。

「-」ボタンを押すと、ブックマークマネージャに登録されているプロジェクトが削除される（図 11-8）。



図 11-8 プロジェクトの削除

プロジェクトの登録、読み込みと削除のための設定と操作手順は以上である。

## 12 プロジェクトの履歴の表示

本章では、プロジェクトの履歴を表示し、履歴に記録されているプロジェクトを読み込むための設定と操作手順を説明する。

---

### この章の主な内容

- ・プロジェクトの履歴の表示
  - ・プロジェクトの履歴からプロジェクトの読み込み
- 

### 12-1 概要

本機能は、読み込みを行ったプロジェクトを履歴として記録・表示するための機能である。履歴には、直近で使用したプロジェクトが 10 件記録され、記録されているプロジェクトを簡単に読み込むことができる。

ここでは、プロジェクトの履歴を表示して、履歴に記録されているプロジェクト「Tank.cnoid」を読み込む例について説明する。

尚、説明に用いる「Tank.cnoid」は、「choreonoid/ext/hairo-world-plugin/sample/tutorial」以下に格納されている。

## 12-2 プロジェクトの履歴の表示と読み込み

以下の手順によって、プロジェクトの履歴が表示され、プロジェクトが読み込まれる。

- (1) メインメニューの「ツール」→「履歴」を選択する。

「履歴」を選択すると、直近で使用したプロジェクトの履歴（最大 10 件）が以下のように表示される（図 12-1）。

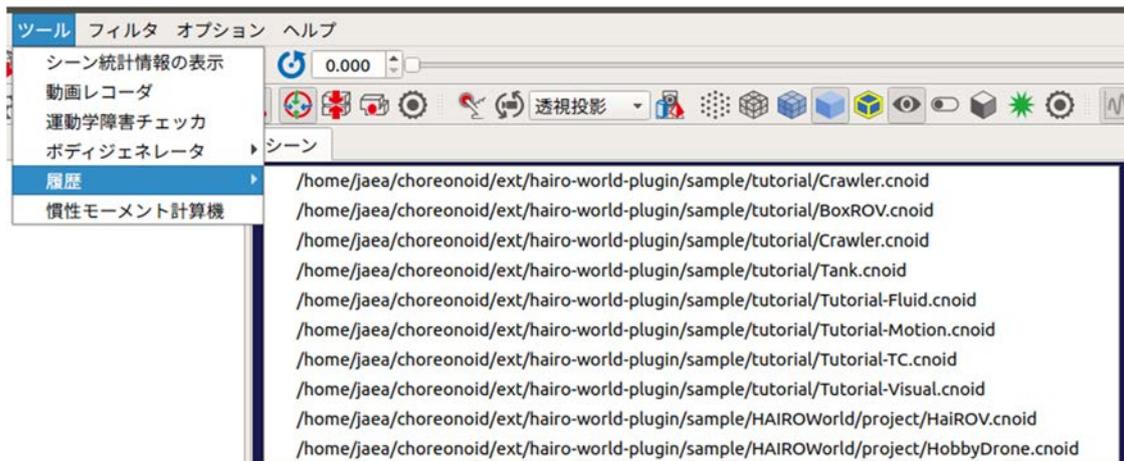


図 12-1 履歴メニューの表示

- (2) 履歴から任意のプロジェクトを選択する。ここでは、例として「choreonoid/ext/hairo-world-plugin/sample/tutorial/Tank.croid」を選択している（図 12-2）。

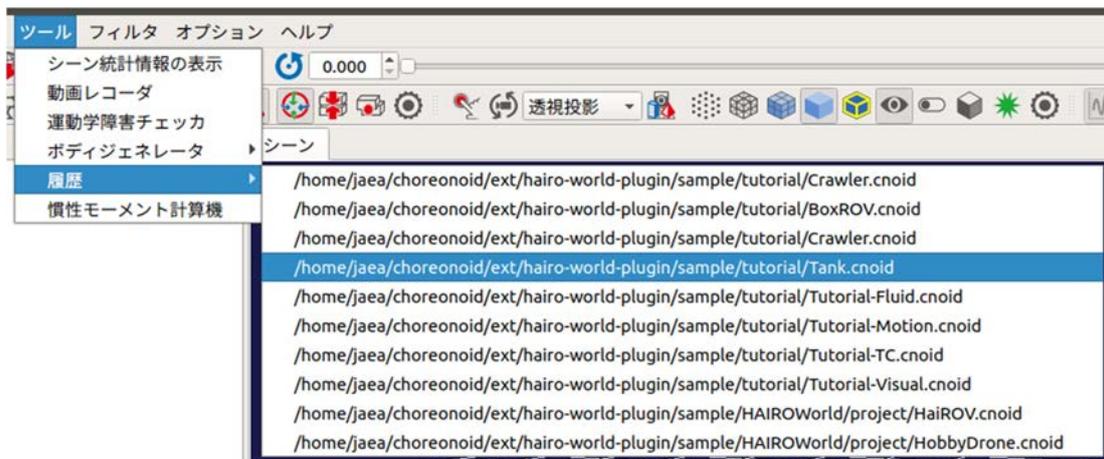


図 12-2 履歴からのプロジェクト選択

プロジェクトを選択すると、プロジェクトが読み込まれる（図 12-3）。

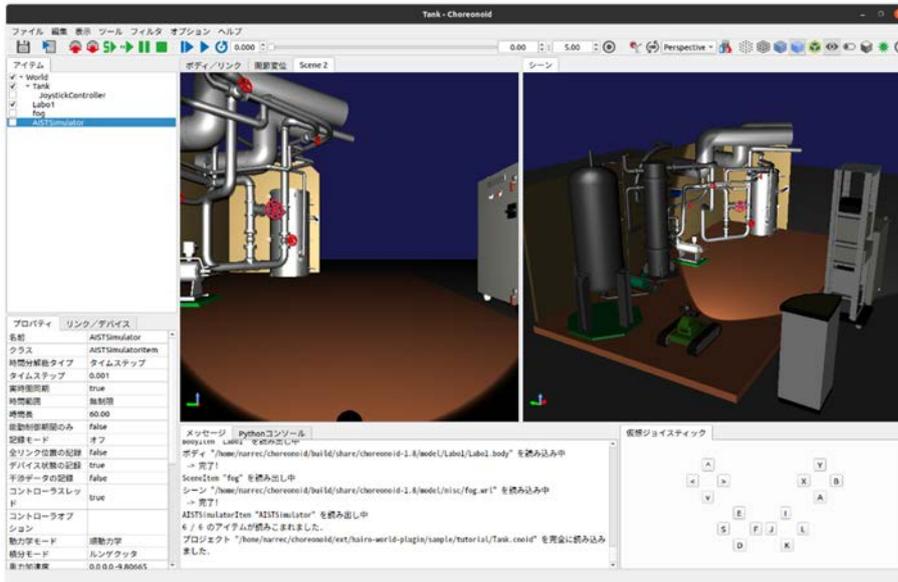


図 12-3 プロジェクトの読み込み

プロジェクトの履歴を表示し、履歴に記録されているプロジェクトを読み込むための設定と操作手順は以上である。

## 13 ロボット/環境物体モデルの生成

本章では、シミュレーションに使用できるロボットや環境物体等のモデル（ボディ）を生成するための設定と操作手順を説明する。

---

### この章の主な内容

- ・パイプモデルの生成
  - ・グレーチングモデルの生成
  - ・スロープモデルの生成
  - ・不整地モデルの生成
  - ・サブクローラ付きクローラロボットモデルの生成
- 

### 13-1 概要

本機能は、ボディ（Choreonoid のシミュレーションに使用する 3D モデル）をユーザが設定したパラメータに基づいて生成する機能である。

本機能では、パイプ、グレーチング、スロープ、不整地及びサブクローラ付きクローラロボットの 5 種のボディを生成することができる。

次節以降では、前述の各ボディのパラメータの詳細と生成例について説明する。

## 13-2 パイプモデルの生成

以下の手順によって、パイプモデルが生成される。生成されるパイプモデルの構造とパラメータは、以下のとおりである（図 13-1）。

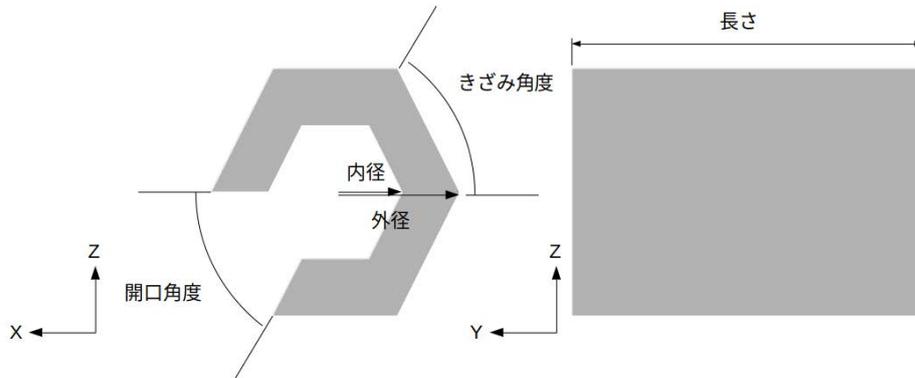


図 13-1 パイプモデルの構造

(1) メインメニューの「ツール」→「ボディジェネレータ」→「パイプ」を選択する（図 13-2）。

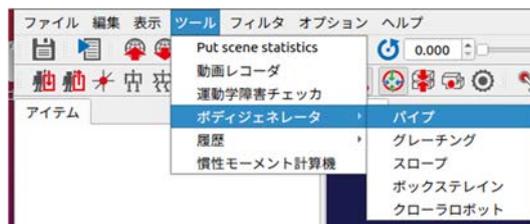


図 13-2 パイプメニューの選択

「パイプ」を選択すると、「パイプビルダ」ダイアログが表示される（図 13-3）。

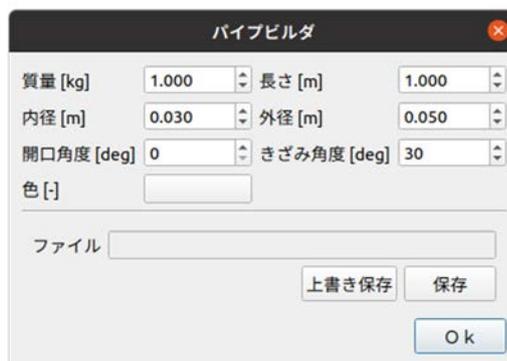


図 13-3 パイプビルダの表示

- (2) 「パイプビルダ」ダイアログに任意のパラメータを入力する。
- (3) 「保存」ボタンを押す。上書きの場合は、「上書き保存」ボタンを押す。

「保存」ボタンを押すと、「ボディを保存」ダイアログが表示される（図 13-4）。

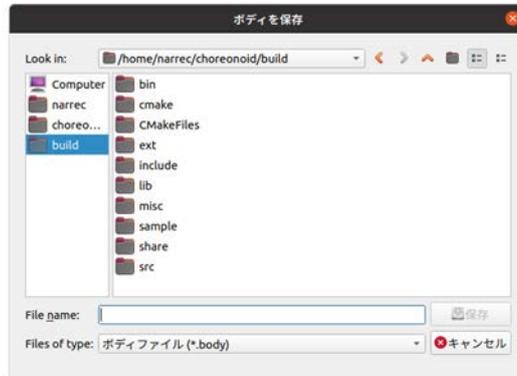


図 13-4 ボディを保存ダイアログの表示

- (4) 「ボディを保存」ダイアログにファイル名を入力し、保存先を指定する。
- (5) 「保存」ボタンを押す。

「保存」ボタンを押すと、指定した保存先にボディファイルが生成される。

### 13-3 グレーチングモデルの生成

以下の手順によって、グレーチングモデルが生成される。生成されるグレーチングモデルの構造とパラメータは、以下のとおりである（図 13-5）。

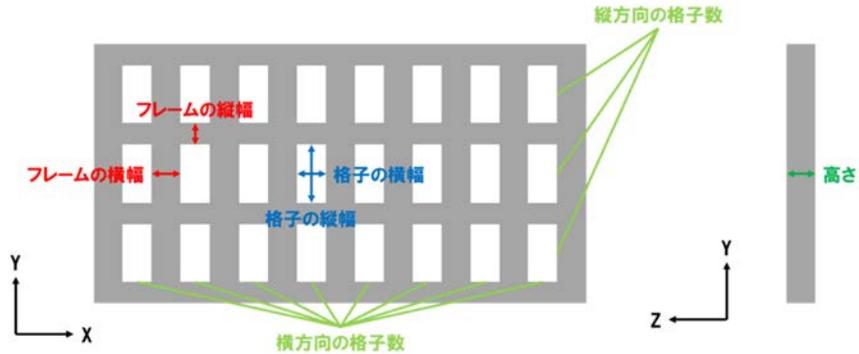


図 13-5 グレーチングモデルの構造

- (1) メインメニューの「ツール」→「ボディジェネレータ」→「グレーチング」を選択する（図 13-6）。

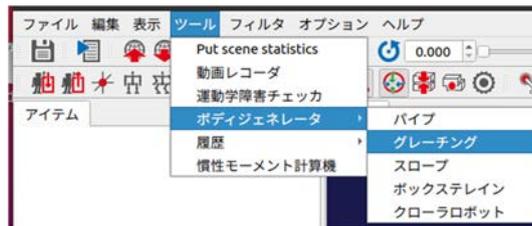


図 13-6 グレーチングメニューの選択

「グレーチング」を選択すると、「グレーチングビルダ」ダイアログが表示される（図 13-7）。



図 13-7 グレーチングビルダの表示

- (2) 「グレーチングビルダ」ダイアログに任意のパラメータを入力する。
- (3) 「保存」ボタンを押す。上書きの場合は、「上書き保存」ボタンを押す。

「保存」ボタンを押すと、「ボディを保存」ダイアログが表示される（図 13-8）。

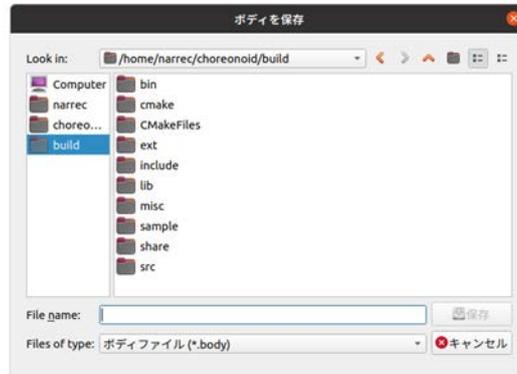


図 13-8 ボディを保存ダイアログの表示

- (4) 「ボディを保存」ダイアログにファイル名を入力し、保存先を指定する。
- (5) 「保存」ボタンを押す。

「保存」ボタンを押すと、指定した保存先にボディファイルが生成される。

## 13-4 スロープモデルの生成

以下の手順によって、スロープモデルが生成される。生成されるスロープモデルの構造とパラメータは、以下のとおりである（図 13-9）。

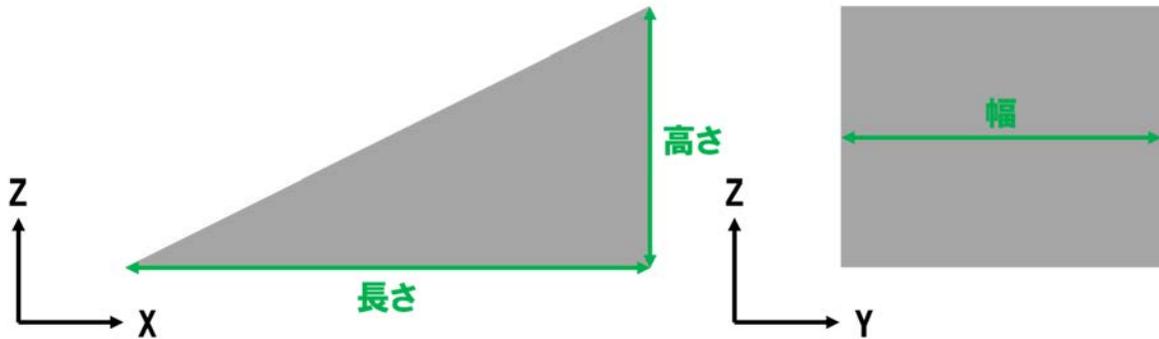


図 13-9 スロープモデルの構造

- (1) メインメニューの「ツール」→「ボディジェネレータ」→「スロープ」を選択する（図 13-10）。

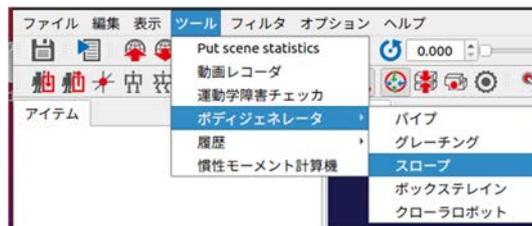


図 13-10 スロープメニューの選択

「スロープ」を選択すると、「スロープビルダ」ダイアログが表示される（図 13-11）。



図 13-11 スロープビルダの表示

- (2) 「スロープビルダ」ダイアログに任意のパラメータを入力する。
- (3) 「保存」ボタンを押す。上書きの場合は、「上書き保存」ボタンを押す。

「保存」ボタンを押すと、「ボディを保存」ダイアログが表示される（図 13-12）。

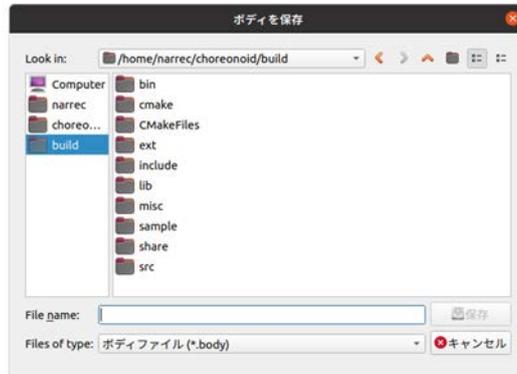


図 13-12 ボディを保存ダイアログの表示

- (4) 「ボディを保存」ダイアログにファイル名を入力し、保存先を指定する。
- (5) 「保存」ボタンを押す。

「保存」ボタンを押すと、指定した保存先にボディファイルが生成される。

## 13-5 不整地モデルの生成

本機能では、縦 0.1m×横 0.1m×高さ 0.1m の箱形状を単位として構成した不整地モデルが生成できる（図 13-13）。

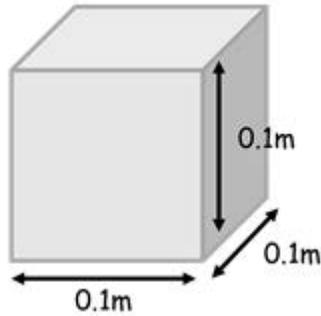


図 13-13 縦 0.1m×横 0.1m×高さ 0.1m の箱形状

以下は、生成された不整地モデルの例である（図 13-14）。箱形状の並べ方・積み上げる個数により、平面・段差・階段を模擬した地形の簡易なモデルを生成できる。

以下の手順によって、不整地モデルが生成される。

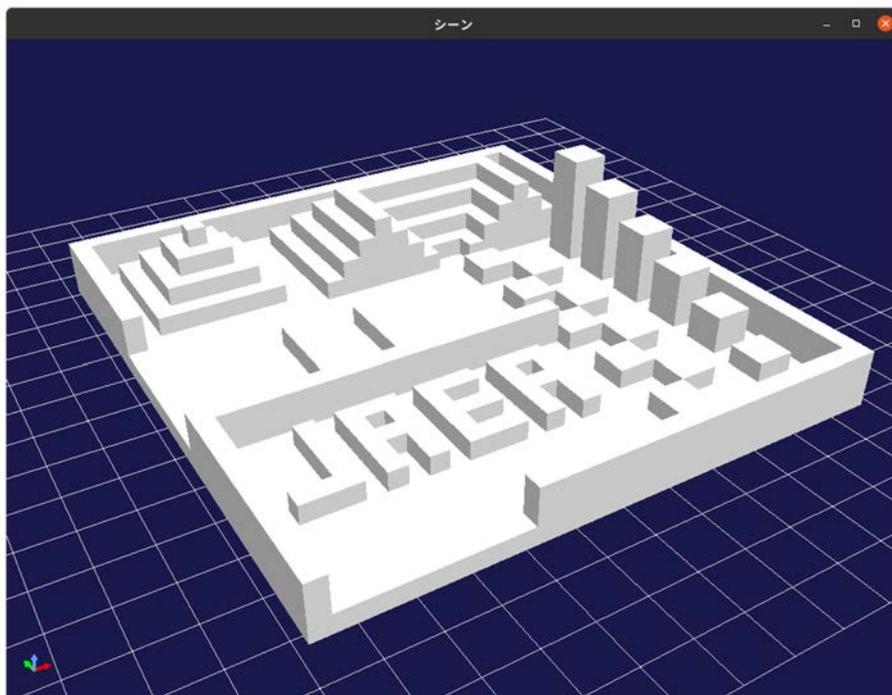


図 13-14 不整地モデルの生成例

## ① 入力ファイルの作成

本機能では、不整地モデルの形状を、入力ファイル（ファイル拡張子：CSV）を用いて指定する。

ここでは例として、入力ファイルを Ubuntu20.04 LTS に予めインストールされている表計算ソフトウェア「Calc」を用いて、以下の不整地モデルを生成する（図 13-15）。

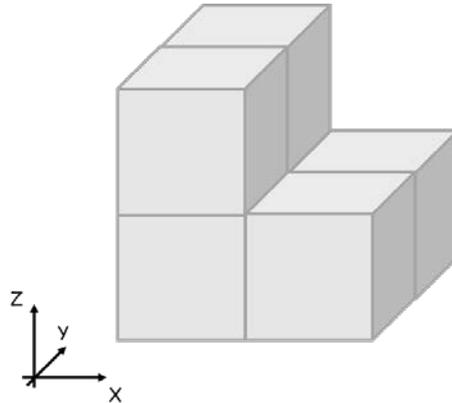


図 13-15 生成する不整地モデル例

(1) Ubuntu のデスクトップ画面表示中に、キーボードで[Ctrl]+[Alt]+[T]を同時に押す。

[Ctrl]+[Alt]+[T]を同時に押すと、以下のように端末（ターミナル）が起動する（図 13-16）。

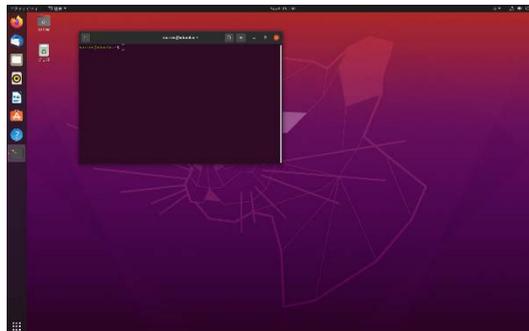


図 13-16 端末の起動

(2) 端末に以下のコマンドを順に入力し、[Enter]を押す。

```
soffice --calc
```

[Enter]を押すと、「Calc」が起動する（図 13-17）。

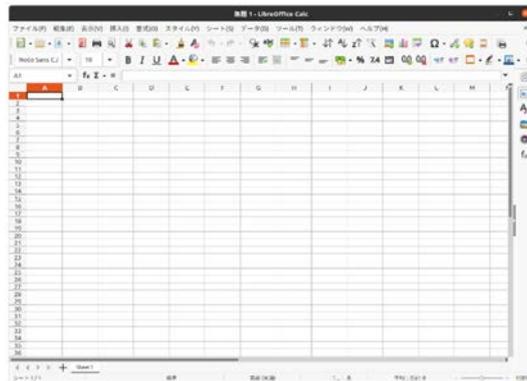


図 13-17 Calc の起動

(3) Calc の表の各セルにパラメータを入力する。

入力ファイルには、前述の箱形状の並べ方と Z 軸方向から見て何個積み上げるかを記述する。ここでは、Calc の表の各セルを不整地モデルを直上から見た時の箱形状の並べ方、また、各セルの値を、箱形状を積み上げる個数と見なし、形状を指定する。このとき、表の行数・列数は制限なく記述できる。ただし、全ての行の列数は同じ数にする必要がある。例えば、箱形状を 2 行×2 列に並べる場合、2 行全ての列数を 2 にする。

前述の不整地モデル例（図 13-15）を生成する場合、Calc の表には以下のように入力する（図 13-18）。

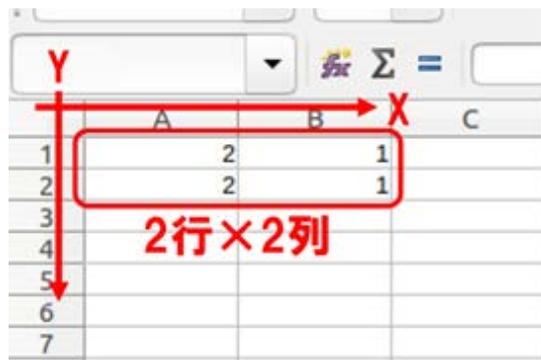


図 13-18 不整地モデル例と入力ファイルの記述例

1行はX軸方向、1列はY軸方向に対応しており、この例では、2行×2列が以下のように対応している（図13-19）。

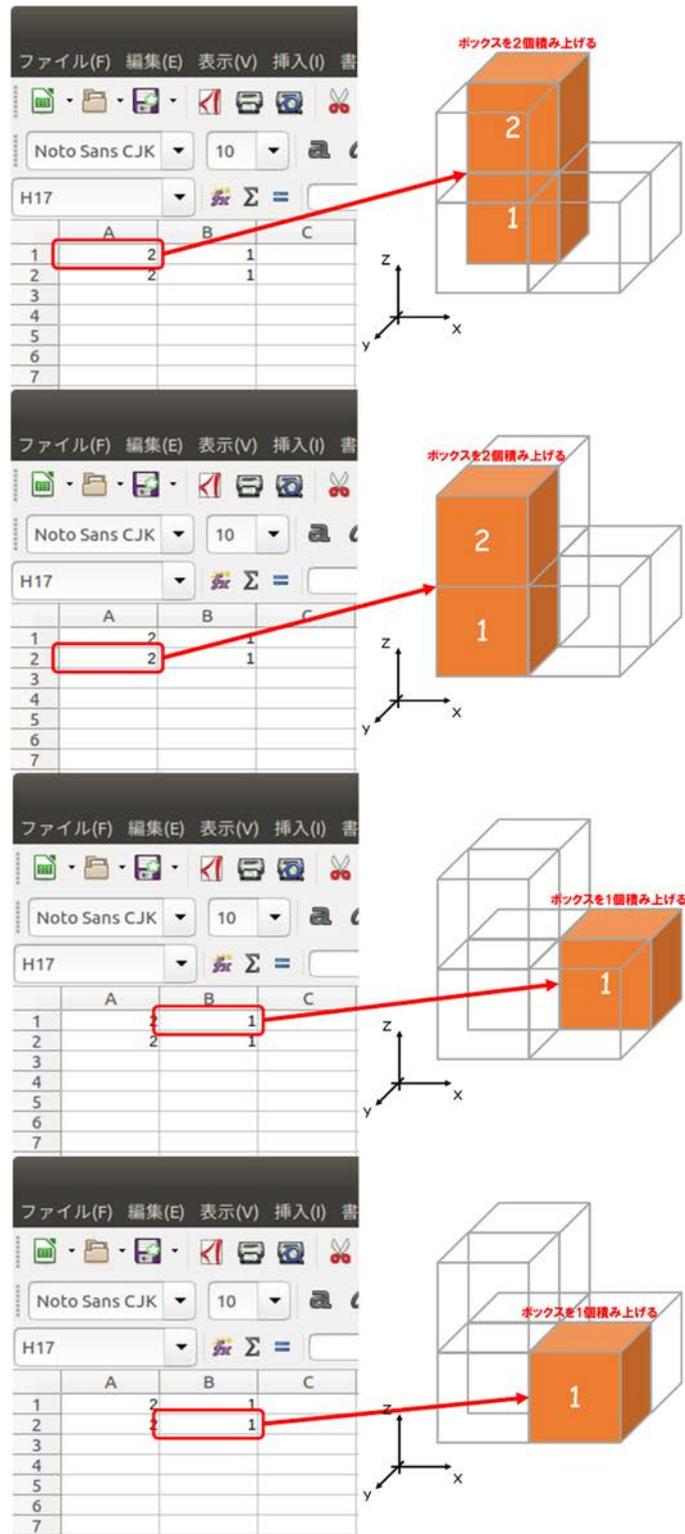


図 13-19 セルと箱形状の対応関係

尚、ここでは2行×2列の場合について説明したが、生成したい形状に合わせてユーザ自身が行数・列数を10行×10列や50行×80列のように自由に変更してよい。

(4) Calcのメニュー「ファイル」→「名前を付けて保存(A)」を選択する。

「名前を付けて保存(A)」を選択すると、以下のダイアログが表示される(図13-20)。



図13-20 名前を付けて保存時のダイアログの表示

(5) ダイアログにファイル名を入力し、保存先を指定する。

(6) 「ODF 表計算ドキュメント(.ods)」を選択し、コンボボックスから「テキスト CSV(.csv)」を選択する。

(7) 「保存(S)」ボタンを押す。

「保存(S)」ボタンを押すと、以下が表示される(図13-21)。

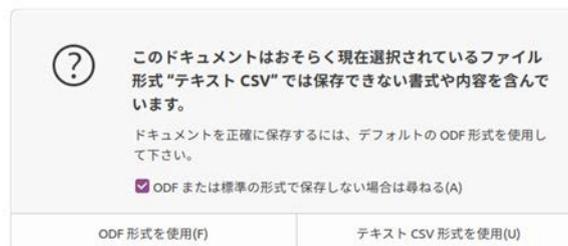


図13-21 書式の確認画面

(8) 「テキスト CSV 形式を使用(U)」を押す。

「テキスト CSV 形式を使用(U)」を押すと、「テキストファイルのエクスポート」ダイアログが表示される (図 13-22)。



図 13-22 テキストファイルのエクスポートダイアログの表示

(9) 「OK(O)」を押す。

「OK(O)」を押すと、指定した保存先に入力ファイルが生成される。

## ② 不整地モデルの生成

(1) メインメニューの「ツール」→「ボディジェネレータ」→「ボックスステレイン」を選択する (図 13-23)。

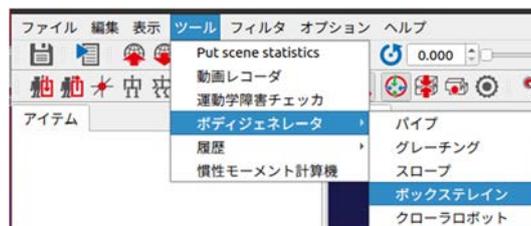


図 13-23 ボックスステレインの選択

「ボックスステレイン」を選択すると、「ボックスステレインビルダ」ダイアログが表示される（図 13-24）。



図 13-24 ボックスステレインビルダの表示

(2) 「ボックスステレインビルダ」ダイアログの「読み込み」ボタンを押す。

「読み込み」ボタンを押すと、「CSV ファイルの読み込み」ダイアログが表示される（図 13-25）。

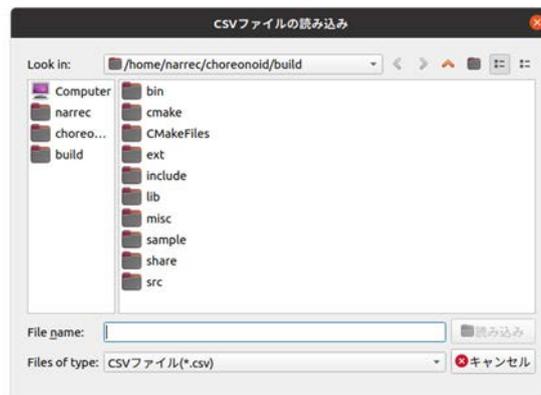


図 13-25 CSV ファイルの読み込みダイアログの表示

(3) 「CSV ファイルの読み込み」ダイアログから、入力ファイルを選択する。

(4) 「読み込み」ボタンを押す。

(5) 必要に応じて、ダイアログ「ボックスステレインビルダ」の「スケール」を入力する。

スケールが 1.0 の場合、縦 0.1m×横 0.1m×高さ 0.1m の箱形状が等倍の大きさとなる。スケールが 2.0 の場合は、1 つの箱形状が縦 0.2m×横 0.2m×高さ 0.2m の大きさとなる。

(6) 「保存」ボタンを押す。上書きの場合は、「上書き保存」ボタンを押す。

「保存」ボタンを押すと、「ボディを保存」ダイアログが表示される（図 13-26）。

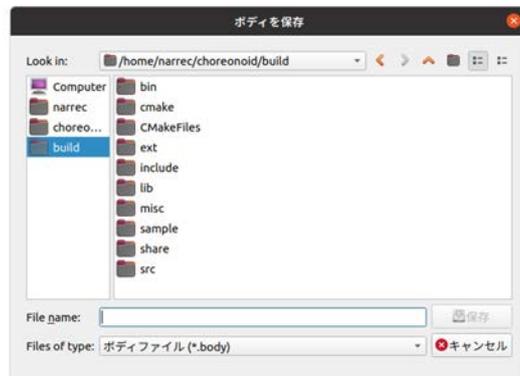


図 13-26 ボディを保存ダイアログの表示

- (7) 「ボディを保存」ダイアログにファイル名を入力し、保存先を指定する。
- (8) 「保存」ボタンを押す。

「保存」ボタンを押すと、ボディファイルが指定した保存先に生成される。

### 13-6 クローラロボットモデルの生成

以下の手順によって、サブクローラ付きのクローラロボットモデルが生成される。生成されるクローラロボットモデルの構造とパラメータは、以下のとおりである（図 13-27）。

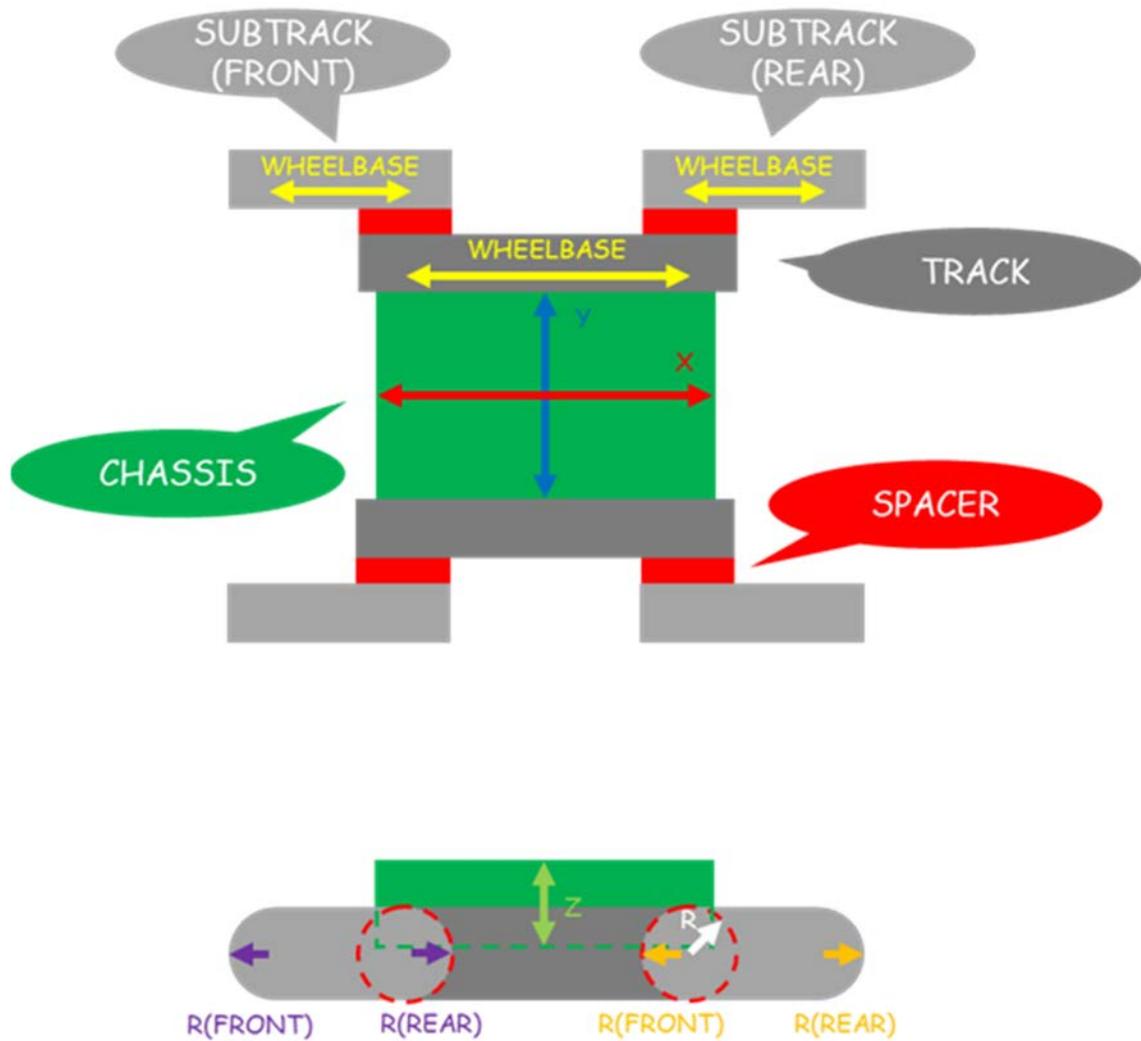


図 13-27 クローラロボットモデルの構造とパラメータ

① クローラロボットモデルの生成

- (1) メインメニューの「ツール」→「ボディジェネレータ」→「クローラロボット」を選択する (図 13-28)。

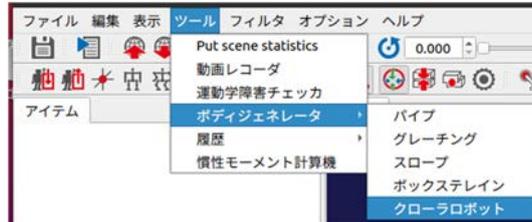


図 13-28 クローラロボットの選択

- 「クローラロボット」を選択すると、「クローラロボットビルダ」ダイアログが表示される (図 13-29)。



図 13-29 クローラロボットビルダの表示

- (2) 「クローラロボットビルダ」ダイアログに任意のパラメータを入力する。

パラメータの詳細は以下のとおりである（図 13-30 – 図 13-34、表 13-1 – 表 13-5）。

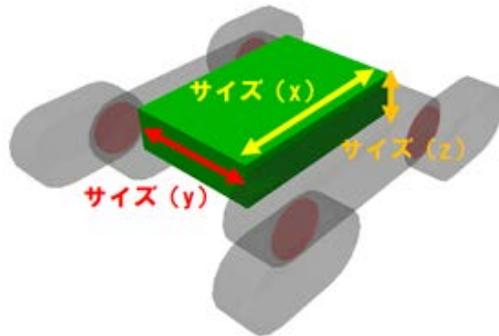


図 13-30 シャーシのパラメータ

表 13-1 シャーシのパラメータの詳細

パラメータ	デフォルト値	単位	意味
質量	8.0	kg	シャーシの質量を指定する
色	グリーン	-	シャーシの色を指定する
サイズ (x, y, z)	0.450 0.300 0.100	m	シャーシのサイズを指定する

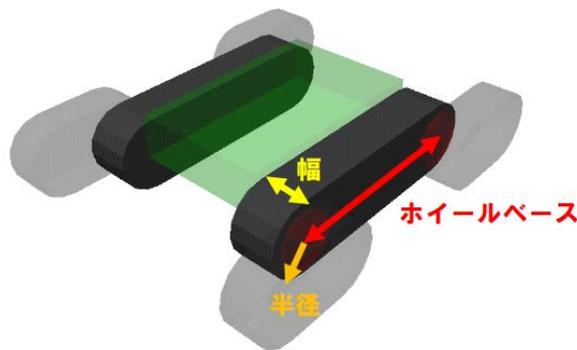


図 13-31 トラックのパラメータ

表 13-2 トラックのパラメータの詳細

パラメータ	デフォルト値	単位	意味
質量	1.0	kg	トラックの質量を指定する
色	黒	-	トラックの色を指定する
半径	0.080	m	トラックの半径を前後共通で指定する
幅	0.100	m	トラックの幅を指定する
ホイールベース	0.420	m	トラックのホイールベースを指定する

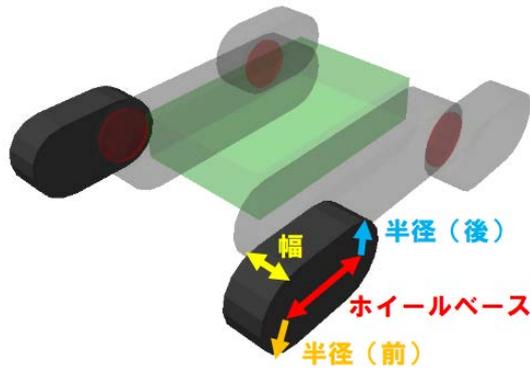


図 13-32 サブトラック（前）のパラメータ

表 13-3 サブトラック（前）のパラメータの詳細

パラメータ	デフォルト値	単位	意味
質量	1.0	kg	サブトラック(前)の質量を指定する
色	黒	-	サブトラック(前)の色を指定する
半径 (前, 後)	0.080 0.080	m	サブトラック(前)の半径を前後別で指定する
幅	0.100	m	サブトラック(前)の幅を指定する
ホイールベース	0.420	m	サブトラック(前)のホイールベースを指定する

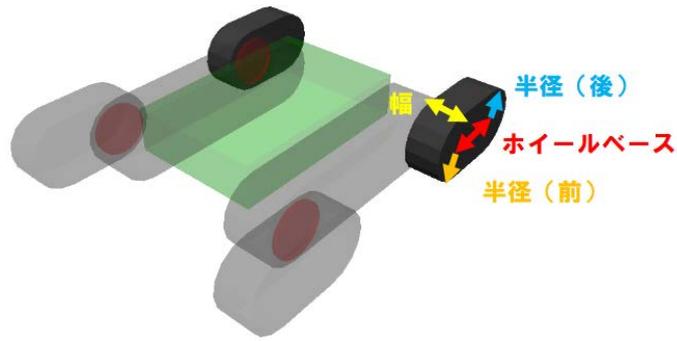


図 13-33 サブトラック (後) のパラメータ

表 13-4 サブトラック (後) のパラメータの詳細

パラメータ	デフォルト値	単位	意味
質量	0.250	kg	サブトラック(後)の質量を指定する
色	黒	-	サブトラック(後)の色を指定する
半径 (前, 後)	0.080 0.080	m	サブトラック(後)の半径を前後別で指定する
幅	0.080	m	サブトラック(後)の幅を指定する
ホイールベース	0.130	m	サブトラック(後)のホイールベースを指定する

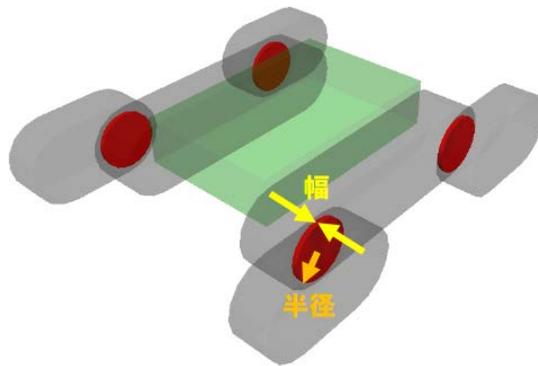


図 13-34 スペーサのパラメータ

表 13-5 スペーサのパラメータの詳細

パラメータ	デフォルト値	単位	意味
質量	0.200	kg	スペーサの質量を指定する
色	赤	-	スペーサの色を指定する
半径	0.060	m	スペーサの半径を指定する
幅	0.010	m	スペーサの幅を指定する

- (3) 「保存」ボタンを押す。上書きの場合は、「上書き保存」ボタンを押す。

「保存」ボタンを押すと、「ボディを保存」ダイアログが表示される（図 13-35）。

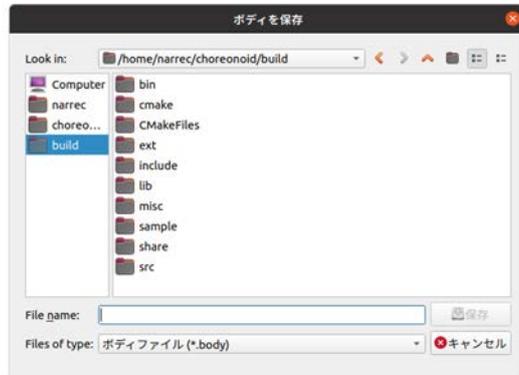


図 13-35 ボディを保存ダイアログの表示

- (4) 「ボディを保存」ダイアログにファイル名を入力し、保存先を指定する。  
 (5) 「保存」ボタンを押す。

「保存」ボタンを押すと、ボディファイルが指定した保存先に生成される。

## ② 設定ファイルの保存

以下の手順によって、「クローラロボットビルダ」ダイアログに入力したパラメータを記述した「設定ファイル」（ファイル形式：YAML）が保存される。

- (1) 「クローラロボットビルダ」ダイアログの「エクスポート」ボタンを押す（図 13-36）。



図 13-36 エクスポートボタンの選択

- (2) 「エクスポート」ボタンを押すと、「設定ファイルの保存」ダイアログが表示される（図 13-37）。

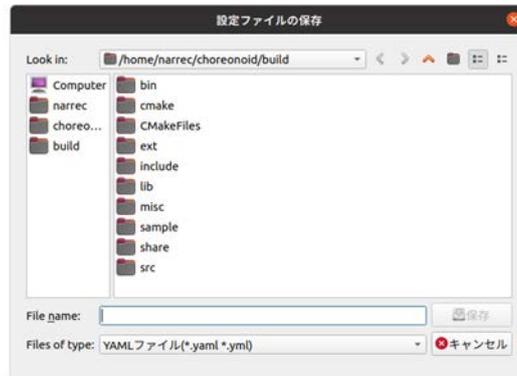


図 13-37 設定ファイルの保存ダイアログの表示

- (2) 「ボディを保存」ダイアログにファイル名を入力し、保存先を指定する。
- (3) 「保存」ボタンを押す。

「保存」ボタンを押すと、「クローラロボットビルダ」ダイアログに入力したパラメータを記述した設定ファイルが指定した保存先に生成される。

### ③ 設定ファイルの読み込み

以下の手順によって、「クローラロボットビルダ」ダイアログに保存したパラメータが読み込まれる。

- (1) 「クローラロボットビルダ」ダイアログの「インポート」ボタンを押す（図 13-38）。



図 13-38 インポートボタンの選択

- (2) 「インポート」ボタンを押すと、「設定ファイルの読み込み」ダイアログが表示される（図 13-39）。

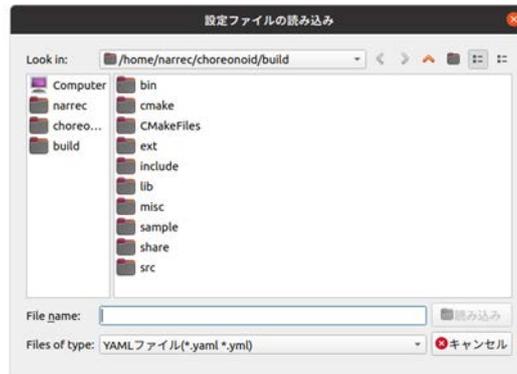


図 13-39 設定ファイルの読み込みダイアログの表示

- (2) 「設定ファイルの読み込み」ダイアログから、設定ファイルを選択する。
- (3) 「読み込み」ボタンを押す。

「読み込み」ボタンを押すと、設定ファイルに保存されているパラメータがクローラロボットビルダに反映される。

シミュレーションに使用できるボディ（パイプ、グレーチング、スロープ、不整地及びサブクローラ付きクローラロボット）を生成するための設定と操作手順は以上である。

# 14 慣性モーメントの計算

本章では、ボディファイルに記述する慣性モーメントを計算するための設定と操作手順を説明する。

---

## この章の主な内容

- ・慣性モーメント計算機による慣性モーメントの計算
- 

### 14-1 概要

本機能は、4種類のプリミティブな形状（箱・円柱・球・円錐）の慣性モーメントを簡易に計算するための機能である。本機能で慣性モーメントを計算できる形状と計算に用いるパラメータは以下のとおりである（表 14-1）。

表 14-1 慣性モーメントを計算できる形状と計算に用いるパラメータ

形状	計算に用いるパラメータ
箱 (Box)	質量[kg]、x (底面の奥行) [m]、y (底面の幅) [m]、z (高さ) [m]
円柱 (Cylinder)	質量[kg]、底面の半径[m]、高さ[m]、中心軸[-]
球 (Sphere)	質量[kg]、半径[m]
円錐 (Cone)	質量[kg]、底面の半径[m]、高さ[m]、中心軸[-]

ここでは、箱形状の慣性モーメントを計算する場合を例に説明する。

## 14-2 慣性モーメントの計算

以下の手順によって、慣性モーメントが計算される。

- (1) メインメニューの「ツール」→「慣性モーメント計算機」を選択する（図 14-1）。

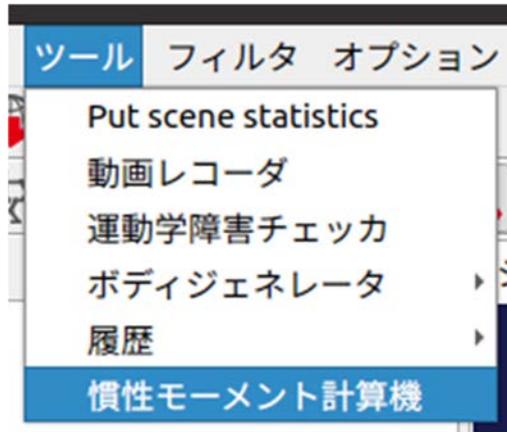


図 14-1 慣性モーメント計算機を選択

「慣性モーメント計算機」を選択すると、以下の「慣性モーメント計算機」が表示される（図 14-2）。

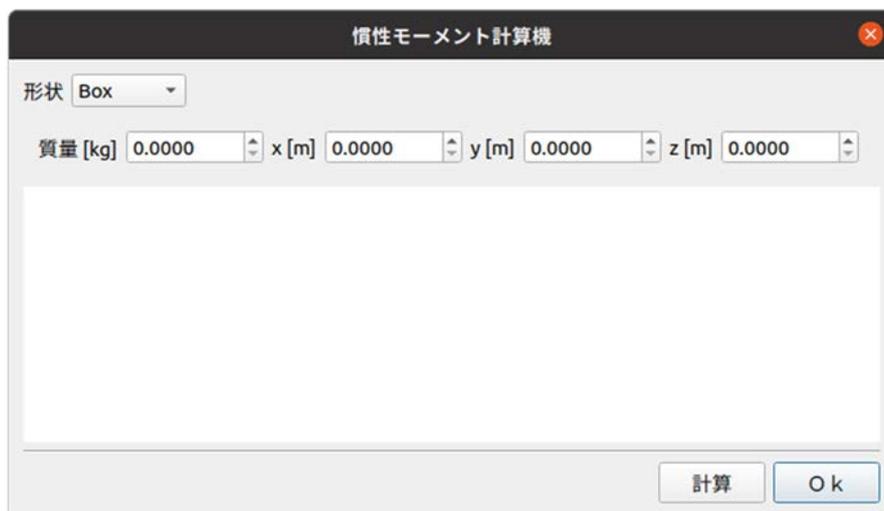


図 14-2 慣性モーメント計算機の表示

- (2) 慣性モーメント計算機の「形状」から任意の形状を選択する。ここでは、例として「Box」を選択している（図 14-3）。



図 14-3 形状の選択

任意の形状を選択すると、対応するパラメータが表示される。ここでは、「形状」に「Box」を選択しているため、「質量[kg]」、「x[m]」、「y[m]」、「z[m]」が表示される（図 14-4）。



図 14-4 Box を選択した場合のパラメータの表示

- (3) パラメータを入力する。ここでは例として、1[kg]の一边 1[m]の箱の慣性モーメントを計算するために、「質量[kg]」に「1」、「x[m]」に「1」、「y[m]」に「1」、「z[m]」に「1」を入力している（図 14-5）。



図 14-5 パラメータの入力

(4) 慣性モーメント計算機の「計算」ボタンを押す。

「計算」ボタンを押すと、計算された慣性モーメントの3×3行列が以下のように表示される(図14-6)。

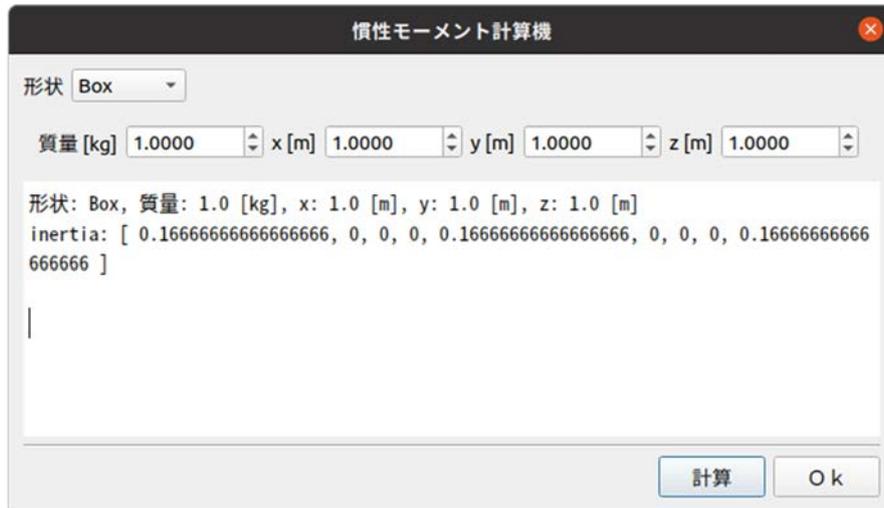


図 14-6 計算された慣性モーメントの表示

ボディファイルに記述する慣性モーメントを計算するための設定と操作手順は以上である。

## 15 おわりに

本稿では、東京電力 HD 福島第一原子力発電所におけるロボットを用いた遠隔による廃炉作業の状況を模擬するために原子力機構が開発した Choreonoid の拡張機能 HAIROWorldPlugin の最新の利用手順を記述した。特に、最新の HAIROWorldPlugin では、シミュレーションをしているロボットと他の物体との間の干渉状態を記録する機能、ジョイスティックの入力状態を表示・記録する機能、環境物体のモデル（パイプ・グレーチング・スロープ）を生成する機能、慣性モーメントを計算する機能、使用頻度の高いシミュレーションの設定を簡易な操作で呼び出す機能、シミュレーションの履歴を記録・表示する機能が追加されており、仮想的な訓練環境の構築をより簡便に行えるようになった。

我々は、今後もシミュレーションやロボットに関わる遠隔技術開発を通じて、東京電力 HD 福島第一原子力発電所の廃炉に貢献していきたいと考えている。更に我々は、原子力機構が保有する原子力緊急事態支援組織が日々実施している定常的な遠隔操作ロボットの操縦訓練に HAIROWorldPlugin を活用することも検討している。この試みを通じて、シミュレーションを活用した遠隔操作ロボットの操縦訓練の効果・効率を検証しながら、HAIROWorldPlugin の機能改良を行う予定である。

本稿で利用手順を説明した HAIROWorldPlugin は、Choreonoid の更新に合わせて今後も随時更新する予定である。HAIROWorldPlugin は、廃炉に関連する研究開発の活性化や新しい技術の創出のために、Choreonoid ユーザだけでなく、廃炉作業に関連する技術者、廃炉作業に関心のある者、他分野の研究者らなどにも広く普及し、使用されることを期待する。

## 参考文献

- [1] 国際廃炉研究開発機構, “福島第一原子力発電所 建屋内で活躍するロボットについて(その3) 水中遊泳ロボット(げんご ROV) & 床面走行ロボット(トライダイバー)”, [https://irid.or.jp/research/gengorov\\_trydiver/](https://irid.or.jp/research/gengorov_trydiver/), (参照: 2021年11月18日).
- [2] 東芝エネルギーシステムズ, “福島第一原子力発電所 3号機の内部を探るー水中遊泳ロボットの奮闘”, <https://www.toshiba-energy.com/nuclearenergy/topics/fukushima-robot.htm>, (参照: 2021年11月18日).
- [3] 東京電力ホールディングス, “福島第一原子力発電所 3号機原子炉建屋内部 ドローンによる線量調査結果”, <https://photo.tepco.co.jp/date/2018/201802-j/180228-01j.html>, (参照: 2021年11月18日).
- [4] Kenta Suzuki, Kuniaki Kawabata, “Development of a Robot Simulator for Decommissioning Tasks Utilizing Remotely Operated Robots”, *Journal of Robotics and Mechatronics*, Vol.32, No.6, pp.1292-1300, 2020.
- [5] 東京電力ホールディングス, “3号機 PCV 内部調査進捗(19日調査速報)”, [https://www.tepco.co.jp/nu/fukushima-np/handouts/2017/images2/handouts\\_170719\\_08-j.pdf](https://www.tepco.co.jp/nu/fukushima-np/handouts/2017/images2/handouts_170719_08-j.pdf), (参照: 2021年11月18日).
- [6] 原田将吾, 小林峰人, 齊藤大祐, “ドローン運用実績と今後の展開”, *アトックス技報*, No.10, 2018, pp.14-17.
- [7] Shinji Kawatsuma, Kuniaki Kawabata, Yoshihiro Tsuchida, Yuta Tanifuji, “Analysis of Emergency Response robots deployed for Fukushima Daiichi Nuclear Power Plants’ Accidents”, *Proceedings of Decommissioning and Remote Systems 2016*, pp.67-69, 2016.
- [8] 中岡慎一郎, “Choreonoid ホームページ”, <https://choreonoid.org/ja/>, (参照: 2021年11月18日).
- [9] 小柳栄次, “サブローラを持つレスキューロボット”, *日本ロボット学会誌*, Vol.28, No.2, pp.147-150, 2010.
- [10] 中岡慎一郎, “Choreonoid マニュアル”, <https://choreonoid.org/ja/manuals/latest/index.html>, (参照: 2021年11月18日).
- [11] 今木清康, “流体機械工学”, 株式会社コロナ社, p.4, 1982.
- [12] DD, “【空気の粘度と動粘度一覧】温度依存性と計算式まとめ”, *機械技術ノート* <https://tec-note.com/330>, (参照: 2021年11月18日).

# 索引

## A

AISTSimulator 3,29,36,43,44,50,54,55,66,67

## B

BodyGenerator プラグイン 15

Bookmark プラグイン 15

## C

Calc 99,100,102

ccmake 12

Choreonoid のインストール 8,14

Choreonoid のダウンロード 7

Choreonoid のビルド 7,14

Choreonoid の起動 8,14

Choreonoid 開発版ドキュメント 6

cmake 7,12,14

Collision States 56,58,59,60

CollisionSeq プラグイン 15

CollisionVisualizer 55,56,59

CollisionVisualizer のプロパティ 55

## F

FileExplorer プラグイン 15

FluidArea 25,26,27,29,30

FluidDynamicsSimulator 29

FluidDynamics プラグイン 15

## G

GitHub 3,5,7,11

Git パッケージのインストール 7

GLVisionSimulator 44

GL ビジョンシミュレータの設定 43

## H

HAIROWorldPlugin のダウンロード 11

HAIROWorldPlugin のディレクトリ構成 11,12

HAIROWorldPlugin のビルド	12,13
HAIROWorldPlugin の設定	12
HAIROWorldPlugin マニュアル	10
<b>I</b>	
InertiaCalculator プラグイン	15
<b>J</b>	
Joystick States	68,69,70,71
JoystickLogger	67,68,70
JoystickStart プラグイン	15
JoystickStatus プラグイン	15
<b>K</b>	
Kinect_Image	43,45
<b>M</b>	
make	7,8,14
Marker Points	51
mkdir	7
MotionCaptureSimulator	50,51
MotionCaptureSimulator のプロパティ	51
MotionCapture プラグイン	15
Multi Value Seq	59,60,70,71
<b>P</b>	
ping	37
<b>T</b>	
TCArea	33,34,35,36
TCSimulator	36,37
TC プラグイン	15
<b>V</b>	
VisualEffector	42,43
VisualEffect プラグイン	15
<b>W</b>	
World	25,32

## あ

アイテムツリービュー	9,25,26,29,32,33,34,36,37,42,43,44, 45,50,54,55,58,66,67,69,79,80,81,82
アイテムのエキスポート	58,69

## い

依存パッケージのインストール	7
----------------	---

## か

画像ビューの表示	42
画像ビューバーの表示	41
干渉の記録	55,56
干渉可視化	54
干渉状態の記録	53,54,56
慣性モーメントを計算できる形状と計算に用いるパラメータ	114
慣性モーメント計算機	114,115,116,117

## く

グレーチングビルダ	94,95
グレーチングモデルの生成	91,94
クローラロボットビルダ	107,112,113
クローラロボットモデルの生成	91,106,107

## け

現在位置からシミュレーションを開始	73,77
現在位置からシミュレーションを再開	73

## こ

コリジョンセンサ	53,54,56,57
コリジョンセンサのパラメータ	56
コリジョンセンサの設定	53,56
コリジョンビジュアライザ	53,54,56
コリジョンビジュアライザの設定	53,54
コントローラの実装	16,22

## さ

サブトラック（後）のパラメータ	110
サブトラック（前）のパラメータ	109

## し

シーンビュー	9,26,33,35,51,52
視覚効果	39,40,42,43,45,46,47
視覚効果の一覧	39
視覚効果の確認	45
視覚効果の設定	39,41,42,46
シミュレーションの一時停止	73,77
シミュレーションの停止	52,73,77
シャーシのパラメータ	108
ジョイスティックによる操作の種類	73
ジョイスティック（DUALSHOCK®4）の入力とビューの表示の割り当て	63
ジョイスティックロガー	65,66,68
ジョイスティックロガーの設定	65,66
ジョイスティック状態	62,63,64
ジョイスティック状態の記録	67,68
初期位置からシミュレーションを開始	30,37,45,51,56,67,73,76
シンプルコントローラ	3,22,23,24

## す

スペーサのパラメータ	110
スラストのパラメータ	20
スラストの設定	16,20
スローピルダ	96,97
スローモデルの生成	91,96

## た

端末（ターミナル）の起動	6,10
--------------	------

## つ

ツールバー	9,30,37,41,45,51,52,56,67,76,77,84
通信障害効果	31,32,36,37,38
通信障害効果の確認	37
通信エリアのプロパティ	34

通信エリアの形状	34,35
通信エリアの設定	31,32,36
通信シミュレータ	31,36,37
通信シミュレータの設定	31,36
て	
ディレクトリの表示	79,82
と	
トラックのパラメータ	108
は	
パイプビルダ	92,93
パイプモデルの生成	91,92
パッケージ一覧の更新	6
パッシブマーカ	48,49,50,51,52
パッシブマーカのパラメータ	50
パッシブマーカの設定	48,49
ひ	
ビルドオプション	4,13
ふ	
ファイルの表示	79,81
不整地モデルの生成	91,98,103
ブックマークバー	84
ブックマークマネージャ	84,85,86,87
浮力・抵抗力を計算するためのリンクのパラメータ	19
プロジェクトの削除	87
プロジェクトの登録	85
プロジェクトの読み込み	18,32,41,49,54,66,76,83,86,88,90
プロパティビュー	9,28,27,29,34,35,36,37,44,51,55,67
ほ	
ボックスステインビルダ	104
ボディジェネレータ	92,94,96,103,107

ま	
マーカポイント	51,52
め	
メインメニュー	9,15,18,25,29,32,36,41,42,43,49,50, 54,58,59,62,66,69,70,75,76,84,89,92, 94,96,103,107,115
メッセージビュー	9
も	
モーションキャプチャシミュレータ	50
モーションキャプチャシミュレータの設定	48,50
り	
流体エリアのプロパティ	27
流体エリアの形状	27
流体エリアの設定	16,25
流体力学エリア	25
流体力学シミュレータ	29
流体力学シミュレータの設定	16,29
履歴からプロジェクトの読み込み	88
リンクの設定	19
ろ	
ロータのパラメータ	21
ロータの設定	16,21
ロボットの移動軌跡	2,48,49,51,52

This is a blank page.



