



ソフトウェアフレームワークChoreonoidの ROS 2プラグインの設計と実装

吉本 幸太郎¹ 大西 祐輝² 中岡 慎一郎³

1: 個人 2: 東京工業大学 3: 株式会社コレオノイド

※ Choreonoid ROS 2プラグインの開発の一部は,
「World Robot Summit (WRS) 2025 過酷環境F-REIチャレンジ/シミュレーション災害チャレンジ」
の協力を受けています。

目次

1. Choreonoidの紹介
2. ChoreonoidとROS
3. ROS 2 プラグインの実装方針
4. ROS 2 へのフィードバック



 **ハンス** @Hans_Robo · 9月25日

16:35から #ROSConJP2024 で「ソフトウェアフレームワークChoreonoidのROS 2プラグインの設計と実装」の発表です！



ソフトウェアフレームワークChoreonoidの
ROS 2プラグインの設計と実装

吉本 幸太郎¹ 大西 祐輝² 中岡 慎一郎³

1: 個人 2: 東京工業大学 3: 株式会社コレオノイド

ソフトウェアフレームワークChoreonoidのROS 2プラグインの設計と実装 / Des...
の協力を受けています。

speakerdeck.comから

🗨️ 🔄 ❤️ 1 📊 12 📌 🔄

目次

1. **Choreonoidの紹介**
2. ChoreonoidとROS
3. ROS 2 プラグインの実装方針
4. ROS 2 へのフィードバック

Choreonoidとは

ロボット開発を支援する統合グラフィカルフレームワーク.

ROS的には..... rqt + RViz + Gazebo を一つのアプリにまとめたようなもの.

特徴:

- 速い！ (高速な起動, 軽量な動作)
- うまい！ (自作プラグインによる容易な機能拡張)
- 安い！ (MITライセンスのオープンソースで無償利用可能)

フレームワークとしてのChoreonoid

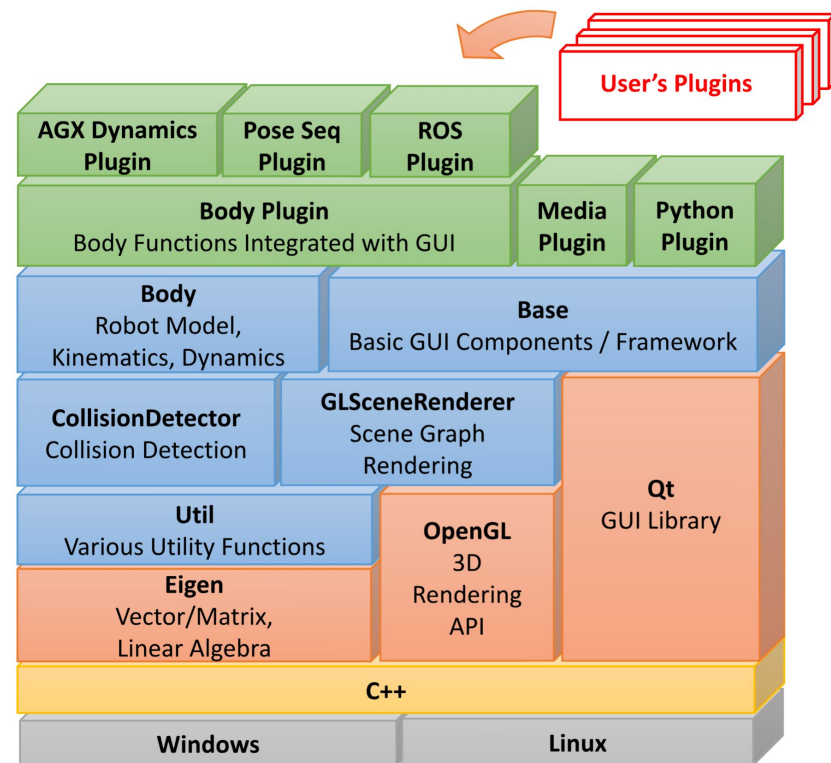
Q. Choreonoidってシミュレータ？

A. 半分正解

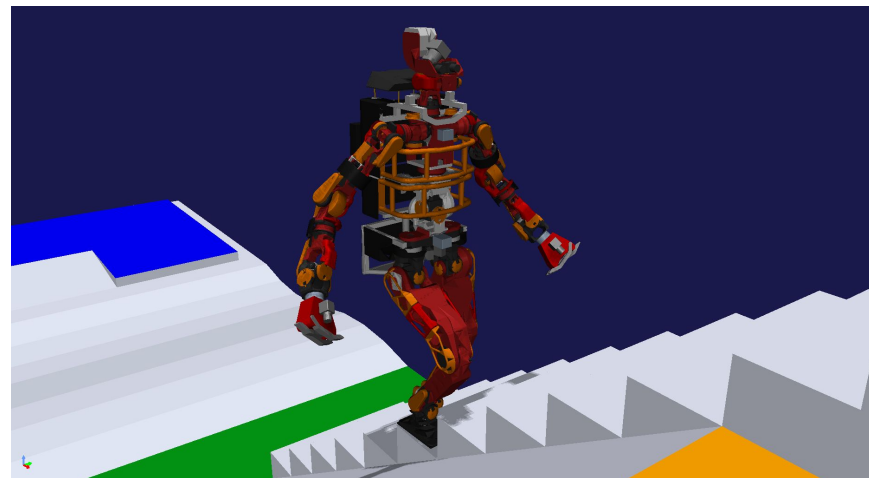
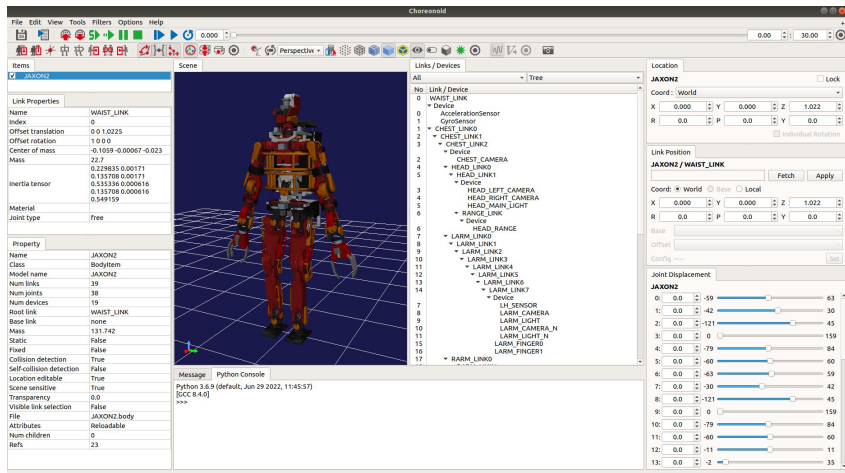
フレームワーク

「拡張可能なロボット用統合GUI環境」

- ロボット開発支援C++ライブラリ群
- ユーザープラグインシステム



Choreonoidの標準プラグイン群が提供する機能



ビジュアライザ

- ロボットモデルやセンサ情報の可視化, GUI操作, 衝突判定, etc.

動力学シミュレータ

- 産総研製の物理エンジンを搭載
- オープンソース / 商用エンジンにも対応 (例) ODE / Bullet / PhysX / AGX Dynamics

Choreonoidの活用事例



Development of HRP-series Humanoid Robots



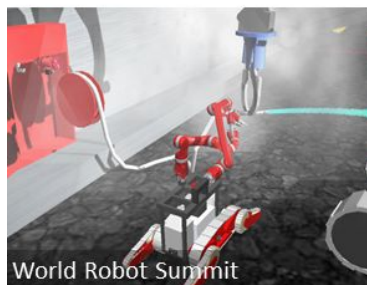
Motion Planning for Industrial Manipulators



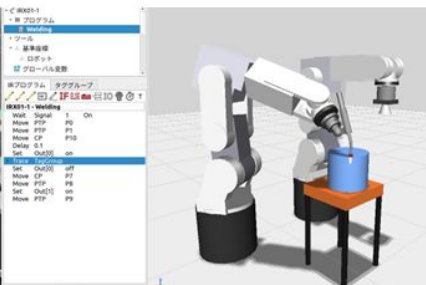
Dancing a Humanoid Robot



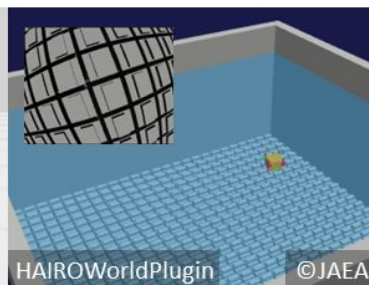
Integrating Tele-operated Semi-autonomous Systems



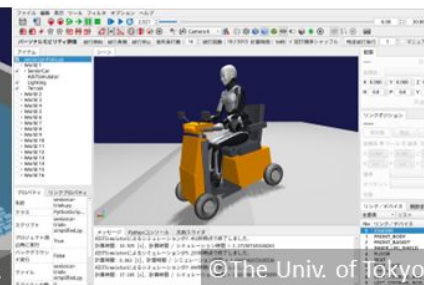
Robot Simulation in Disaster Response



Teaching Industrial Robots with Graphic User Interfaces



Virtual Tele-operation Environment for Nuclear Decommissioning



Risk Assessment of Assistive Products

目次

1. Choreonoidの紹介
- 2. ChoreonoidとROS**
3. ROS 2 プラグインの実装方針
4. ROS 2 へのフィードバック

ChoreonoidのROS対応

ChoreonoidのROS対応の内容は、大別して次の三点にまとめられる:

- Choreonoidのノード化
- ROS標準のURDFで書かれたロボットモデルの読み込み
- 物理シミュレーションで、ROSインターフェースを提供するプラグイン
(**ROSプラグイン**, 次頁で説明)

Choreonoidにおけるプラグインの立ち位置

Choreonoid の設計思想

「プラグインを作って機能を増やす」

→プラグイン自体も作りやすくしている。

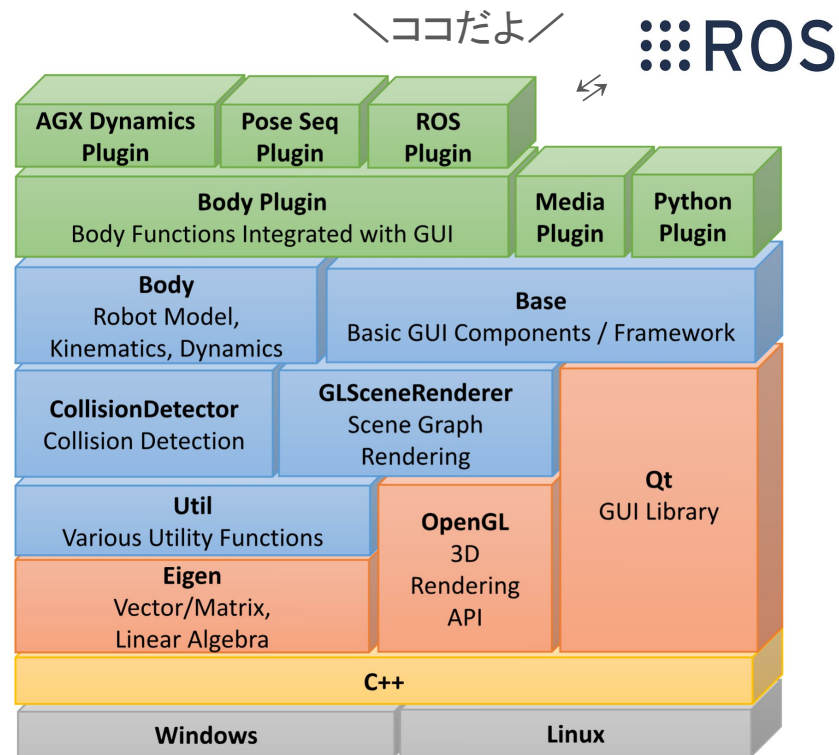
ROSプラグイン

ROSのライブラリや

Choreonoidの他のプラグインを利用して

, Choreonoidの内部情報と

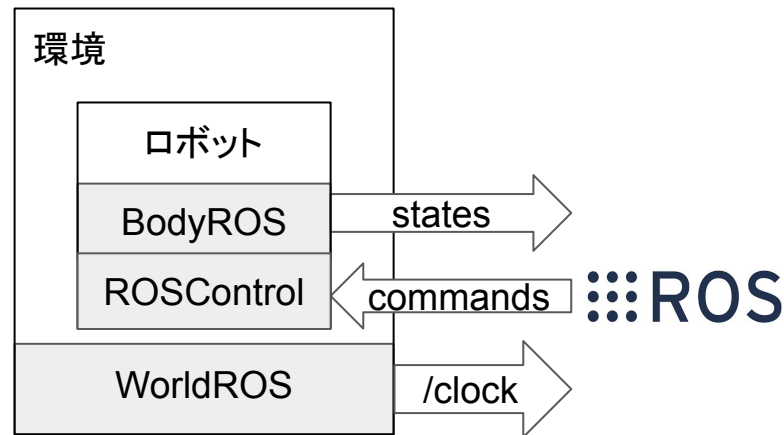
ROS での通信情報を相互に変換する。



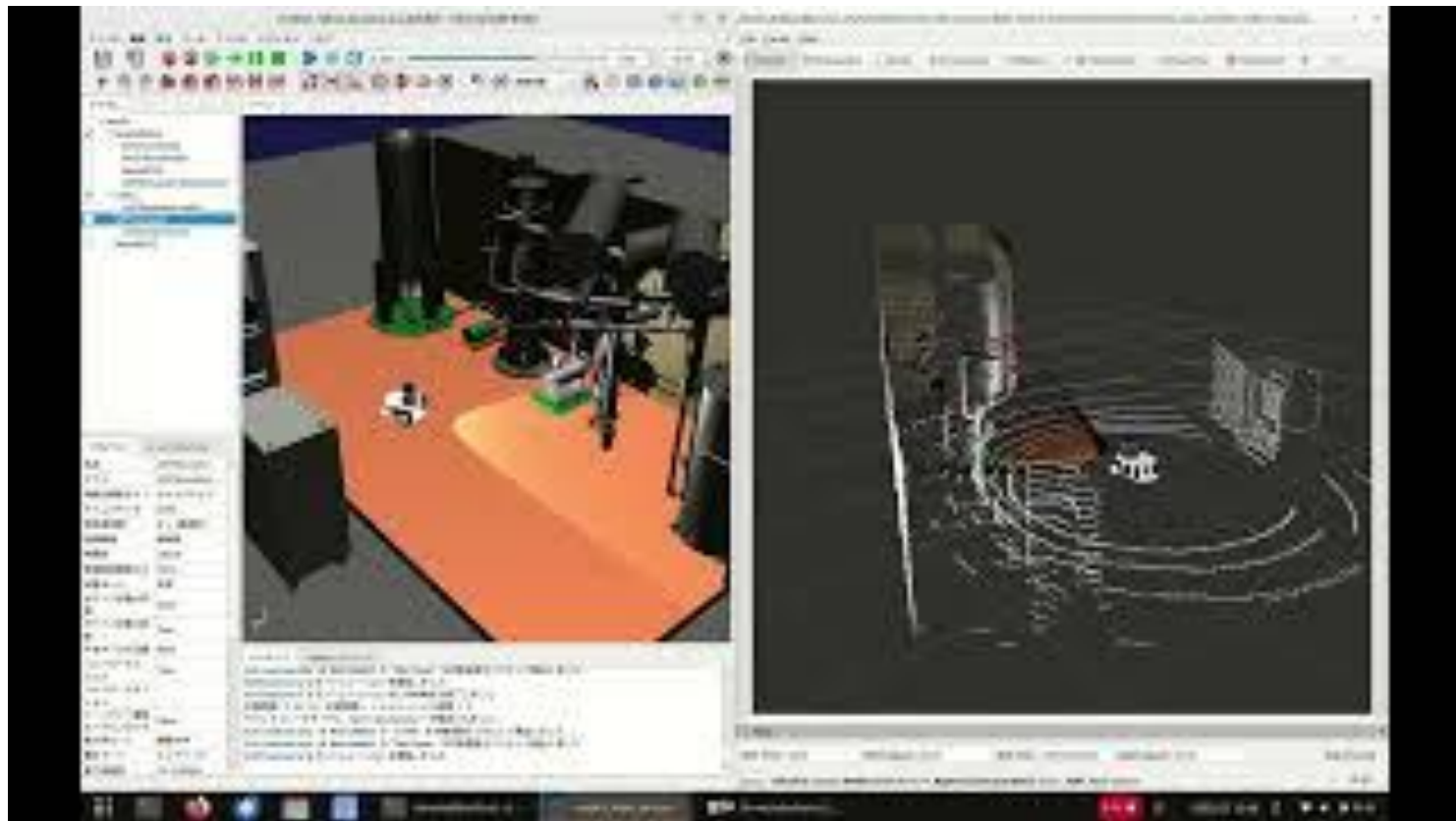
ChoreonoidのROSプラグイン

ChoreonoidをROS環境でシミュレータとして使うためのプラグインアイテム群

- WorldROS(2): シミュレーション環境に1つだけ配置
 - シミュレーション時間を/clockトピックで出力
- BodyROS(2): ロボットごとに配置
 - ロボットのセンサ情報をトピックで出力 (joint statesを含む)
- ROSControl: ロボットごとに配置
 - ros_control の制御インターフェイスを提供
 - ※ ros2_control 向けは現在開発中



ROS 2 プラグインの動作例



目次

1. Choreonoidの紹介
2. ChoreonoidとROS
- 3. ROS 2 プラグインの実装方針**
4. ROS 2 へのフィードバック

ROS 1/2 両対応の形態

ROS 1 パッケージを ROS 2 に対応させるとき, いくつかの形態が存在

- 別リポジトリ運用
 - リポジトリ名の末尾に「2」が付きがち
 - 例: rviz2 / navigation2
- 同一リポジトリ・別ソース運用
 - ブランチごとでそれぞれのバージョンに特化
 - 例: image_transport
- 同一リポジトリ・同一ソース運用
 - 一つのソースコードで全てのバージョンへ対応
 - 例: BehaviorTree.CPP

ROS 1/2 両対応の形態

ROS 1 パッケージを ROS 2 に対応させるとき, いくつかの形態が存在

- 別リポジトリ運用
 - リポジトリ名の末尾に「2」が付きがち
 - 例: rviz2 / navigation2
- 同一リポジトリ・別ソース運用
 - ブランチごとでそれぞれのバージョンに特化
 - 例: image_transport
- 同一リポジトリ・同一ソース運用
 - 一つのソースコードで全てのバージョンへ対応
 - 例: BehaviorTree.CPP



メンテナンスコスト低

BugFixのbackportがいらぬ

ROS 1/2 同時対応の方法

環境変数 ROS_VERSION:

有効なROSバージョンとして、1または2の値を保持する。

CMakeLists.txt と package.xml のそれぞれが、
この環境変数を参照することで、ビルドが分岐可能になる。


CMakeLists.txt:

```
if(DEFINED ENV{ROS_VERSION})  
  if($ENV{ROS_VERSION} EQUAL 1)  
    find_package(  
      catkin REQUIRED COMPONENTS  
      roscpp
```

package.xml:

```
<depend condition="$ROS_VERSION == 1">roscpp</depend>  
<depend condition="$ROS_VERSION == 2">roscpp</depend>
```

ソースコード

```
#ifdef USING_ROS  CMakeで要マクロ定義  
#include <ros/package.h>  
#endif
```


ノード管理

Choreonoidは1プロセスで動作し、各部品はスレッドで動作するが.....

ROS 1: 1プロセス1ノード

- 複数のロボットのトピックを全て出力する神ノードが誕生 🙄
- 外部でROS Masterを起動する必要がある
- 処理の負荷を分散が難しい

ROS 2: Executorレベルも柔軟にプログラム可能に

1ロボット1ノード1スレッドの綺麗な設計が可能に！

ネットワーク分離(実装中)

ROS 1: マルチROSマスター

ROSマスターの設定が必要. ノード側の設定も必要.

ROS 2: ドメインIDの切り替えでネットワーク分離が可能

1プロセス内のノードでも, Contextを別に持つことで
別々のネットワークに所属可能

並列シミュレーションの ROS対応の可能性が開けた！！

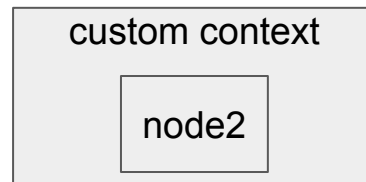
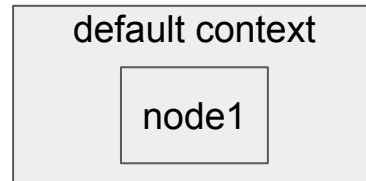
目次

1. Choreonoidの紹介
2. ChoreonoidとROS
3. ROS 2 プラグインの実装方針
- 4. ROS 2 へのフィードバック**

rclcppのバグ

rclcpp::spin_some / spin_all 関数は、
ノードの context によらず、
強制的に default context を使用する
仕様になっていた。

そのため、
custom context を使用しようとする
と、エラーで異常終了するバグが存在した。

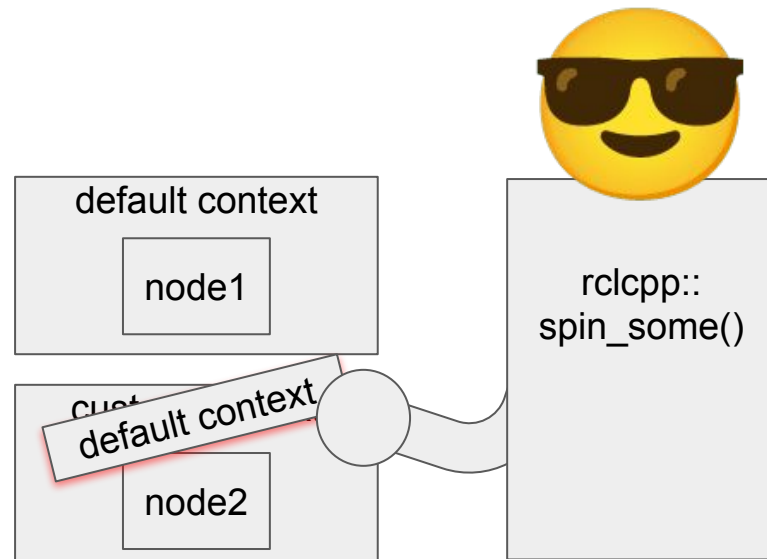


rclcpp::
spin_some()

rclcppのバグ

rclcpp::spin_some / spin_all 関数は、
ノードの context によらず、
強制的に default context を使用する
仕様になっていた。

そのため、
custom context を使用しようとする
と、エラーで異常終了するバグが存在した。



ROS 2本体へのフィードバック

修正内容自体は大したことはないが、rclcppの修正というのは貴重な体験.....

1. プルリクエスト作成

2. レビュー

3. CIテスト

4. マージ

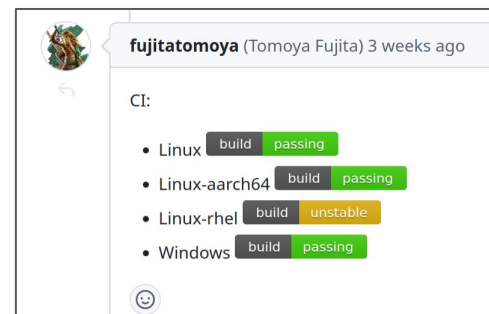
5. リリース

6. バックポート

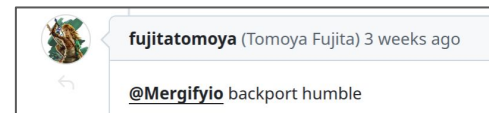
メンテナが回してくれる
不安定なテストあり

月1のリリースで使えるようになる

申し出るとHumbleなどへ遡って修正できる
※ABIが壊れないものに限る



CIを回してくれる藤田さん



バックポートの呪文を唱える藤田さん

プルリクエストの作法

- バグの修正
- テストの追加
 - 修正前に失敗する
 - 修正後に成功する
- ターゲットはrollingブランチ
 - rollingに入れてからバックポートしないと
将来のリリースでバグが復活しちゃう, なんてことも

まずはrollingに入れてから
Humbleなどにバックポートする

反応がないときは気軽に催促！

※メンテナは忙しいので時々忘れられます



HansRobo 3 weeks ago

 @fujitatomoya friendly ping



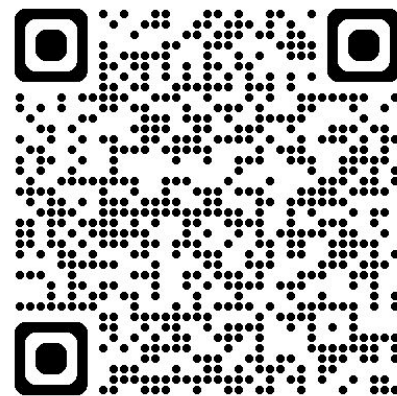
Try Choreonoid & ROS 2!

まとめ: ChoreonoidのROS 2プラグインを作成した.

- ROS 1/2 同時対応の方法
- ROS 2 でのノード管理の改善
- ROS 2 でのネットワーク管理の改善
- rclcpp への貢献手順



choreonoid_ros2_
mobile_robot_tutorial



doc/ に200ページ超の
教材PDFあります！
(スターください！)

付録: ROS 1 から ROS 2 への移植の参考リンク集

- ROS 2 Documentation (Humble), Migrating from ROS 1 to ROS 2
<https://docs.ros.org/en/humble/How-To-Guides/Migrating-from-ROS1.html>
- ros2_control, Migration Guides
<https://control.ros.org/humble/doc/migration/migration.html>

※ リンクは全て Humble 向け