

ROSCon JP 2021

# ROBOTIS Dynamixelのための ROS 2 Control実装 dynamixel\_control

株式会社Preferred Networks エンジニア

近藤 豊

<https://youtalk.jp>

- 博士（工学）
- カワダロボティクス開発部主任  
→ Preferred Networks エンジニア
- 著書「ROS2ではじめよう  
次世代ロボットプログラミング」
  - 増刷版はROS 2 Foxyに対応
- ROSCon JP 2021実行委員
- ROS Japan Users Group元主宰
- NVIDIA Jetson Influencer
- 2児の父

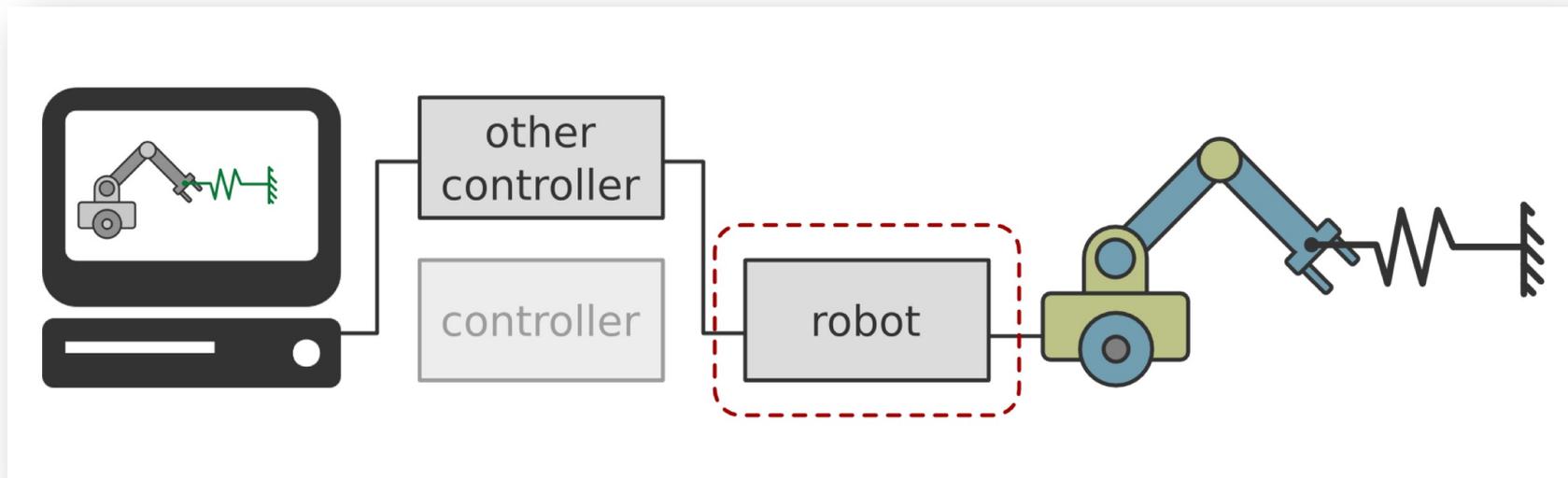


<http://amzn.to/2Ji8mQI>

# ROS 2 Control

# ROS 2 Control

- ROS 2ロボットの（リアルタイム）制御のためのフレームワーク
- ROS 1版ROS Controlの長所を引き継ぎ、短所を改善



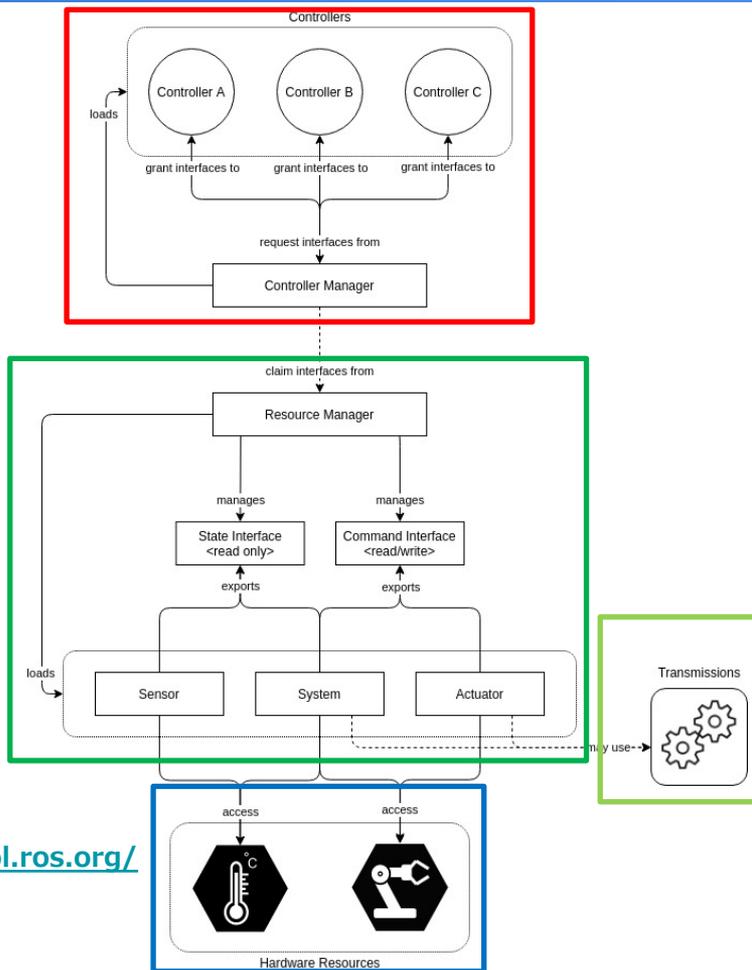
<https://roscon.ros.org/2015/presentations/RealtimeROS2.pdf>

- **ros2\_control**
  - ROS 2 Controlのフレームワーク
- **ros2\_controllers**
  - コントローラの各種実装
- **control\_msgs**
  - ROS 2 Controlのメッセージ定義
- **control\_box**
  - 補間、PIDなど制御理論のツール
- **realtime\_tools**
  - リアルタイム制御のためのツール
- **ros2\_control\_demos**
  - ROS 2 Controlを使ったデモ

# フレームワーク概要

- **Controller Manager**
  - コントローラ（制御器）の読み込み、実行管理
- **Resource Manager**
  - ロボット全体、アクチュエータ、センサ単位で指令値と現在値の読み書き管理
- **Transmissions**
  - 指令値とアクチュエータ入力値の相互変換
- **Hardware Resources**
  - ロボット、センサそのもの

[https://ros-controls.github.io/control.ros.org/getting\\_started.html#architecture](https://ros-controls.github.io/control.ros.org/getting_started.html#architecture)



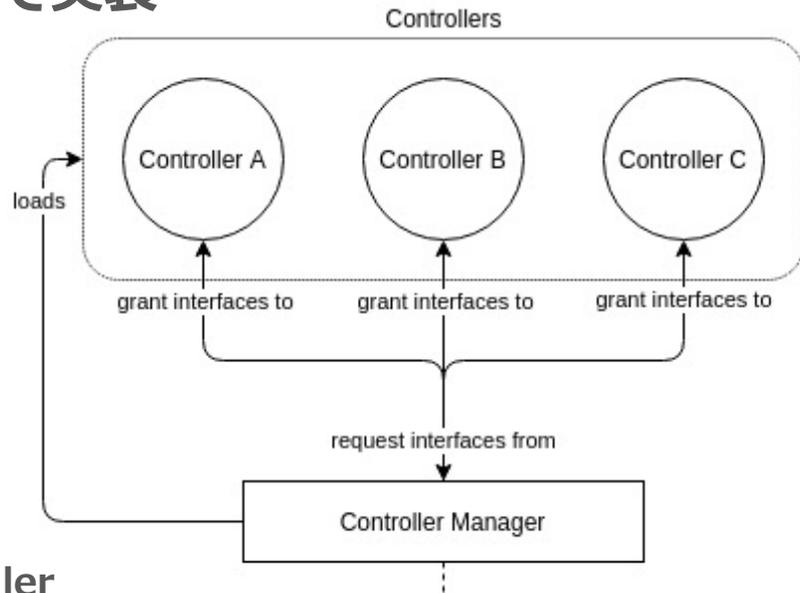
# Controller Manager

- 各コントローラは状態遷移モデルとして実装

- configure : 読み込み
- start (activate) : 実行開始
- stop (deactivate) : 実行終了

- 標準コントローラ

- ForwardCommandController
- JointGroupPosition/Velocity/EffortController
- **JointTrajectoryController** : 補間軌道生成
- **JointStateBroadcaster** : JointStateメッセージ送信
- ...



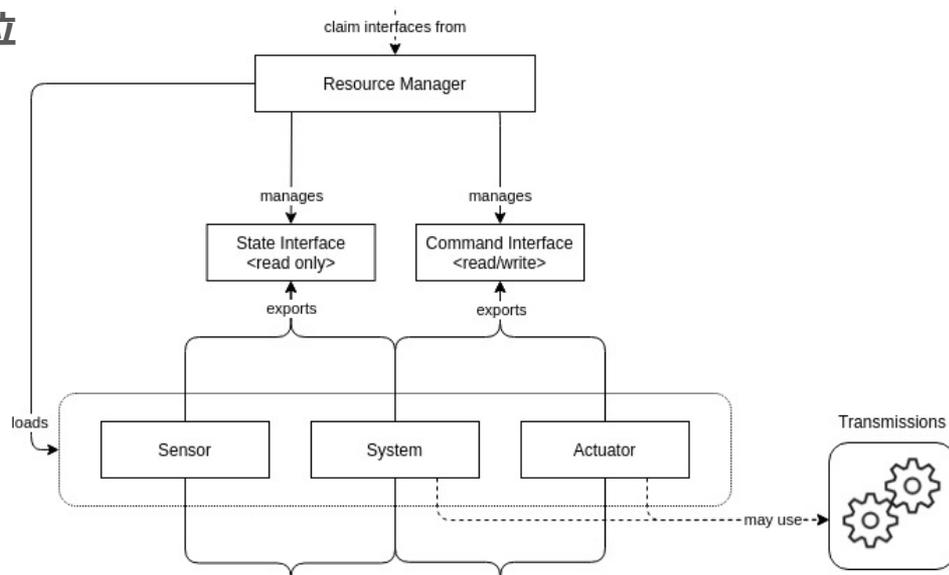
# Resource ManagerとTransmissions

- リソースの組み合わせで多様なロボットに対応

- **SystemInterface** : ロボット全体
- **ActuatorInterface** : アクチュエータ単位
- **SensorInterface** : センサ単位

- 追加機能

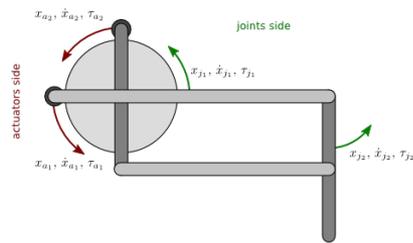
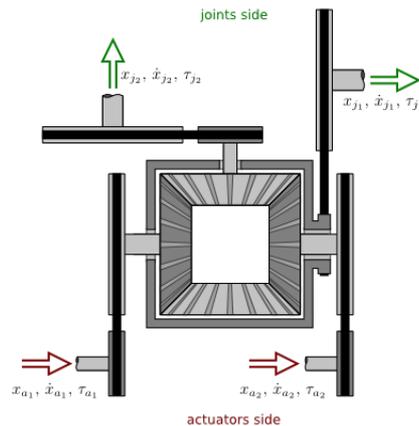
- **Transmission** :  
指令値をアクチュエータ入力値に変換
- **JointLimitHandle** :  
指令値をアクチュエータ限界値内に補正



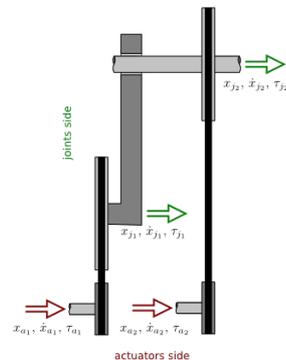
# Transmission

## ● 指令値とアクチュエータ入力の変換

- SimpleTransmission
  - 減速比とオフセット量に基づく変換
- DifferentialTransmission
  - 差動装置を使った2軸アクチュエータのための変換
- FourBarLinkageTransmission
  - 四節リンク構造を使った2軸アクチュエータのための変換



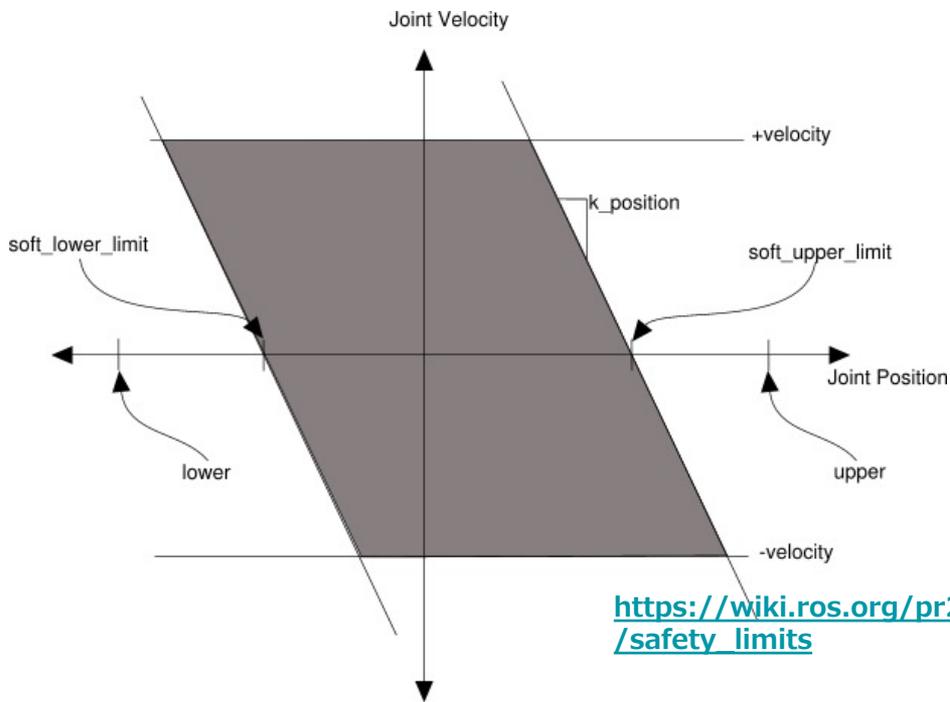
Four-bar linkage



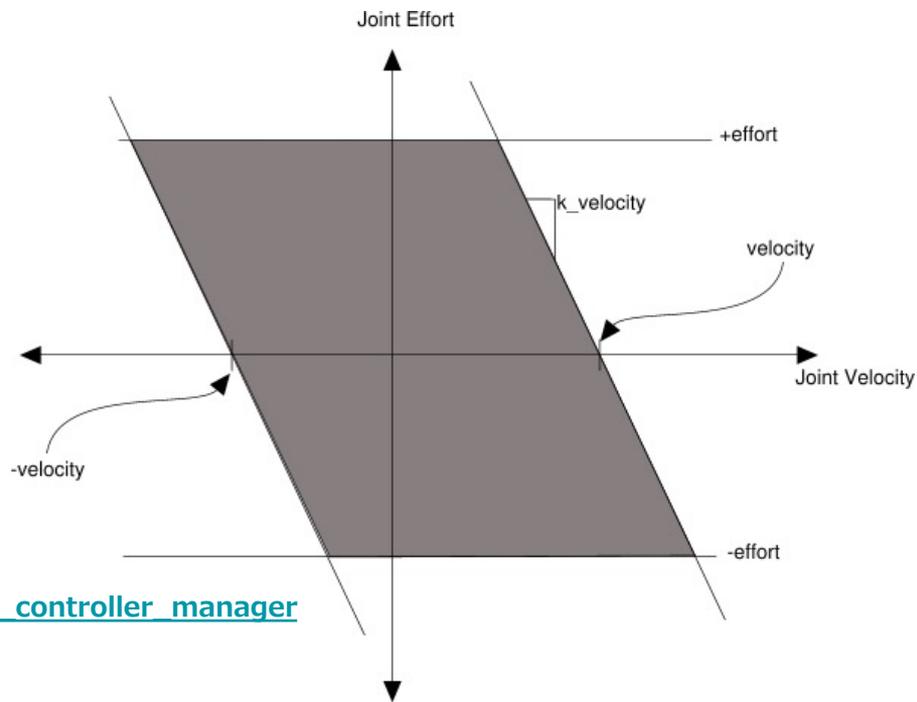
Remote second joint actuation

# JointLimitHandle

- 関節の機械的リミットから少しマージンを取った安全リミットの設定



関節角リミット



関節角速度リミット

[https://wiki.ros.org/pr2\\_controller\\_manager/safety\\_limits](https://wiki.ros.org/pr2_controller_manager/safety_limits)

- **ros2cliのサブコマンドros2 control**
  - **ros2 control list\_controllers**
    - Controller Managerが管理するコントローラのリストアップ
  - **ros2 control list\_hardware\_interfaces**
    - Resource Managerが管理するリソースのリストアップ
  - **ros2 control load\_controller**
    - コントローラの読み込み
    - --set-stateオプションで読み込みのみか、即実行開始か選択可能
  - **ros2 control switch\_controllers**
    - コントローラの実行開始、実行終了の切り替え

# ROS 2 Controlに貢献しよう

- ROS 2 Controlのコンセプトは素晴らしい
- が、**実装、デモ、テスト、ドキュメンテーション**は追いついていない
- ROS 2 Control WGメンバだけでなく、**私たちも貢献していくべき**

🔍 0 Open ✓ 5 Closed

🔗 [ros2\_control\_test\_assets] Fix typo ✖  
#371 by youtalk was merged on Apr 7 · Approved

🔗 Sleep accurate duration on ros2\_control\_node ✓  
#302 by youtalk was merged on Jan 26 · Approved

🔗 Call template methods instead of copy and paste ✓  
#78 by youtalk was merged on Jun 11, 2020 · Approved 📄 1 task

🔗 Fix typo ✓  
#77 by youtalk was merged on Apr 24, 2020 · Approved

🔗 Use dashing for Travis CI ✓  
#34 by youtalk was merged on Jan 21, 2020

## Contributors 46

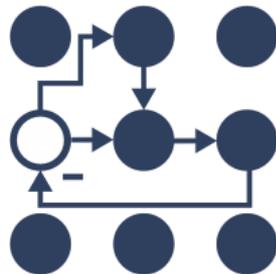


+ 35 contributors

[https://github.com/ros-controls/ros2\\_control/pulls?q=is%3Apr+is%3Aclosed+author%3Ayoutalk](https://github.com/ros-controls/ros2_control/pulls?q=is%3Apr+is%3Aclosed+author%3Ayoutalk)  
[https://github.com/ros-controls/ros2\\_control/graphs/contributors](https://github.com/ros-controls/ros2_control/graphs/contributors)

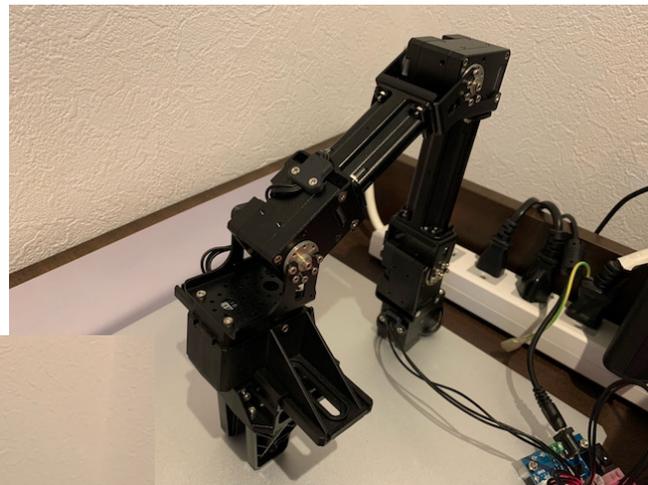
# ROBOTIS Dynamixelのための ROS 2 Control実装 dynamixel\_control

- ROBOTISサーボモータDynamixelのためのROS 2 Control実装
  - [https://github.com/youtalk/dynamixel\\_control](https://github.com/youtalk/dynamixel_control)
  - すべてのDynamixelの組み合わせに対応
  - URDFにros2\_controlタグを追加するだけで適用可能
  - ROS 2 Foxy対応



# 2

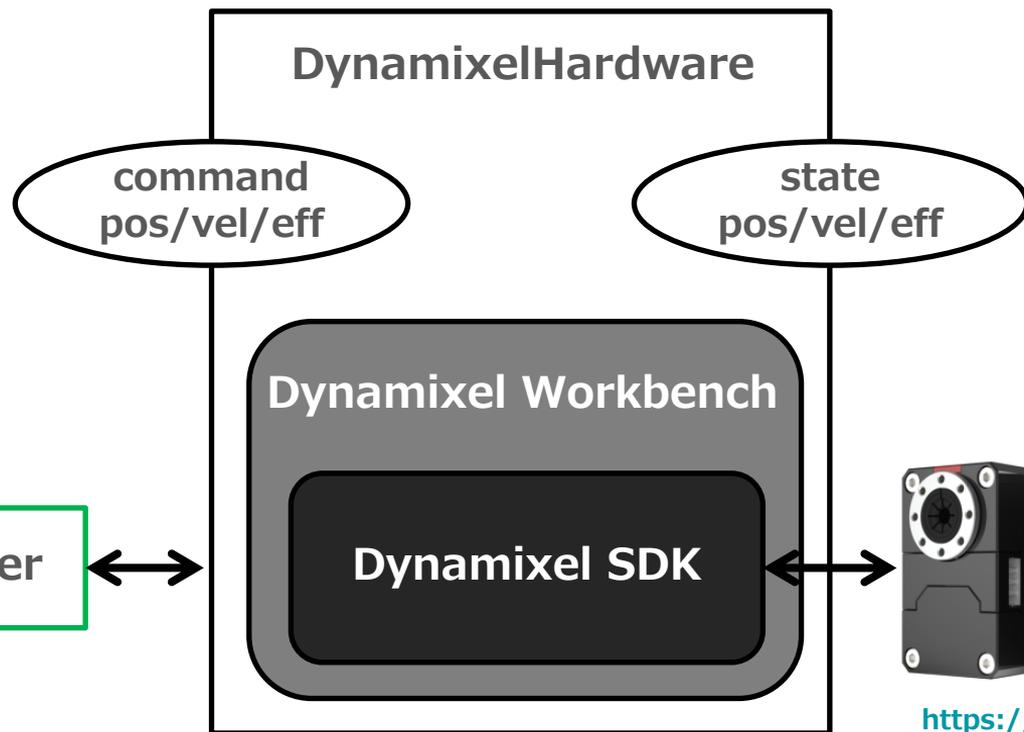
- dynamixel\_hardware
  - DynamixelロボットのためのSystemInterface実装
- open\_manipulator\_x\_description
  - ROBOTIS Open Manipulator-Xへの適用デモ
- pantilt\_bot\_description
  - パンチルト軸のみの2軸ロボットへの適用デモ



open\_manipulator\_x



pantilt\_bot



<https://emanual.robotis.com/docs/en/dxl/x/>

- **DynamixelHardware**

- 現在値 (state) 取得 : 位置、速度、トルク
- 指令値 (command) 命令 : 位置、速度、トルク
- 変速 : 擬似SimpleTransmission

- **コントローラ**

- ForwardCommandController
- JointGroupPosition/Velocity/EffortController
- JointTrajectoryController
- JointStateBroadcaster

```
youtalk ~/ros/foxy % ros2 launch open_manipulator_x_robot open_manipulator_x_robot.launch.py
```

```
youtalk ~/ros/foxy % ros2 topic echo /joint_states
```

```
youtalk ~/ros/foxy % ros2 control switch_controllers --start-controllers velocity_controller
```

<https://youtu.be/DzizMZp3KqU>



# URDF ros2\_controlタグ

- リソース設定
- hardwareタグ
  - plugin : DynamixelHardware指定
  - usb\_port : USB接続ポート
  - baud\_rate : ボーレート
  - use\_dummy : ダミーモード有効化
- jointタグ
  - id : DynamixelモータID
  - command : 指令値インタフェース
  - state : 現在値インタフェース

[https://github.com/youtalk/dynamixel\\_control/blob/main/pantilt\\_bot\\_description/urdf/pantilt\\_bot.ros2\\_control.xacro](https://github.com/youtalk/dynamixel_control/blob/main/pantilt_bot_description/urdf/pantilt_bot.ros2_control.xacro)

```
<ros2_control name="${name}" type="system">
```

```
<hardware>  
  <plugin>dynamixel_hardware/DynamixelHardware</plugin>  
  <param name="usb_port">/dev/ttyUSB0</param>  
  <param name="baud_rate">1000000</param>  
  <!-- <param name="use_dummy">true</param> -->  
</hardware>
```

```
<joint name="joint1">  
  <param name="id">11</param>  
  <command_interface name="position"/>  
  <command_interface name="velocity"/>  
  <state_interface name="position"/>  
  <state_interface name="velocity"/>  
  <state_interface name="effort"/>  
</joint>
```

```
<joint name="joint2">  
  <param name="id">12</param>  
  <command_interface name="position"/>  
  <command_interface name="velocity"/>  
  <state_interface name="position"/>  
  <state_interface name="velocity"/>  
  <state_interface name="effort"/>  
</joint>
```

```
</ros2_control>
```

# controllers.yaml

- コントローラ設定
- **controller\_manager**
  - Controller Managerの必要とするパラメータを記述
  - コントローラプラグインと名前をリストアップ
- **各コントローラ名**
  - 各コントローラの必要とするパラメータを記述

[https://github.com/youtalk/dynamixel\\_control/blob/main/pantilt\\_bot\\_description/controllers/controllers.yaml](https://github.com/youtalk/dynamixel_control/blob/main/pantilt_bot_description/controllers/controllers.yaml)

```
controller_manager:  
  ros__parameters:  
    update_rate: 100 # Hz  
  
  velocity_controller:  
    type: velocity_controllers/JointGroupVelocityController  
  
  joint_trajectory_controller:  
    type: joint_trajectory_controller/JointTrajectoryController  
  
  joint_state_broadcaster:  
    type: joint_state_broadcaster/JointStateBroadcaster
```

```
velocity_controller:  
  ros__parameters:  
    joints:  
      - joint1  
      - joint2
```

```
joint_trajectory_controller:  
  ros__parameters:  
    joints:  
      - joint1  
      - joint2  
  
  command_interfaces:  
    - position  
  
  state_interfaces:  
    - position
```

# 適用事例

<https://twitter.com/shirokunet/status/1436579497810169857>

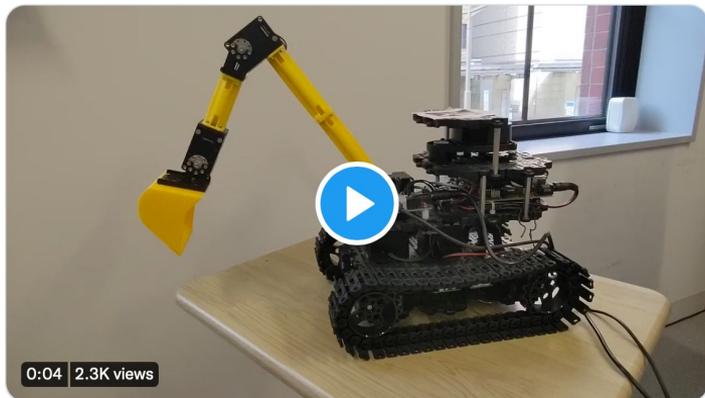


shirokunet  
@shirokunet

ros2\_control、youtalkさんの dynamixel\_control 触りながら勉強してます！感謝！！

過去に OpenMANIPULATOR-X ベースで作ったミニ建機でテスト 😊

[Translate Tweet](#)



3:36 PM · Sep 11, 2021 · Twitter Web App

12 Retweets 2 Quote Tweets 91 Likes

<https://youtu.be/B2hrMPb5IpY>

#ROSDevDay #ROSdevelopers #ROS  
Build Your Own ROS 2 Robot from Scratch | Yutaka Kondo | ROSDevDay 2021  
2,961 回視聴 · 2021/08/04

## TODO

- ~~起動時の初期姿勢問題の解決~~ ✓
- ~~ギア比対応~~ ✓
  - ROS 2 Control版transmission\_interfaceが未完成...
- ~~速度制御への対応~~ ✓
- ~~電流制御への対応~~ ✓
- forward\_command\_controller以外のros2\_controllerへの対応
  - ~~位置、速度制御への対応~~ ✓
  - ~~トルク制御への対応~~ ✓
  - 複数ros2\_controllerの読み込み ✓
    - ソフトウェアリミット制御 🦵
      - ROS 2 Control版joint\_limits\_interfaceが未完成...
- ~~Movelt2連携~~ 📝

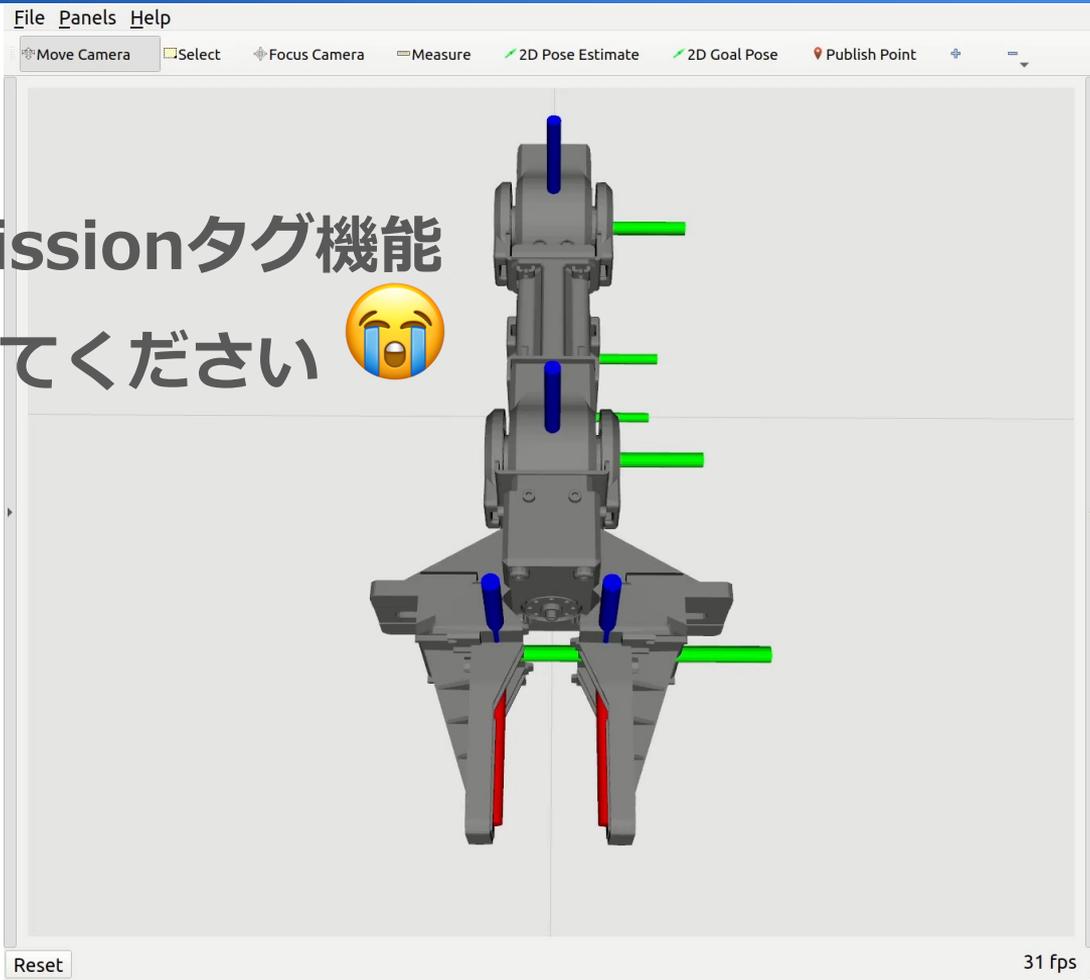
+ 3 more things

# SimpleTransmission擬似対応

- まだtransmissionタグは実装されていない -> 独自実装
  - jointタグに独自パラメータ追加
    - **mechanical\_reduction** : 減速比
    - **mechanical\_offset** : オフセット量
  - [https://github.com/youtalk/dynamixel\\_control/pull/11](https://github.com/youtalk/dynamixel_control/pull/11) (ドラフトPR)

```
<joint name="gripper">
  <param name="id">15</param>
  <!-- TODO: Replace to transmission tag -->
  <param name="mechanical_reduction">${0.0275/pi}</param>
  <param name="mechanical_offset">-1</param>
  <command_interface name="position"/>
  <command_interface name="velocity"/>
  <state_interface name="position"/>
  <state_interface name="velocity"/>
  <state_interface name="effort"/>
</joint>
```

# SimpleTransmission擬似対応



早くtransmissionタグ機能を  
をリリースしてください 🥲

# Effort Control対応

- Dynamixelの電流制御モードを使ったトルク制御への対応
  - トルク = 電流値 x 係数
  - すでに各軸トルクの現在値はJointStateに反映済み
  - [https://github.com/youtalk/dynamixel\\_control/pull/9](https://github.com/youtalk/dynamixel_control/pull/9) (ドラフトPR)
    - 実機テスト不足...

早く柔らかい制御に入門したい 

でも実機は一組しかないから壊したらお終い 

# MoveIt 2対応

- ROS 2 Control対応ロボットはMoveIt 2対応も簡単はず
  - すでにJointTrajectoryControllerとJointStateBroadcasterには対応済み
  - [https://github.com/youtalk/dynamixel\\_control/pull/4](https://github.com/youtalk/dynamixel_control/pull/4) (ドラフトPR)
    - [https://github.com/ros-planning/moveit\\_resources/tree/ros2](https://github.com/ros-planning/moveit_resources/tree/ros2)を参考に結構あがいたが、いまだ正常起動せず

moveit\_setup\_assistantのROS 2版を  
早くリリースしてください🥲

# Gazebo対応

- ROS 2 Control対応ロボットはGazebo対応も簡単はず
  - ただしGazebo 11 (**Ignition Gazeboではない**)
  - [https://github.com/youtalk/dynamixel\\_control/pull/8](https://github.com/youtalk/dynamixel_control/pull/8) (マージ済み)
    - URDFのSystemInterfaceとGazebo Pluginを変更・追記するだけ
    - 一応動いた！でもGazeboSystemの起動に数分かかる！？
  - Ignition Gazeboを使う方が良さそう
    - <https://discourse.ros.org/t/ignition-gazebo-ros-2-august-community-meeting-cross-post/21777>

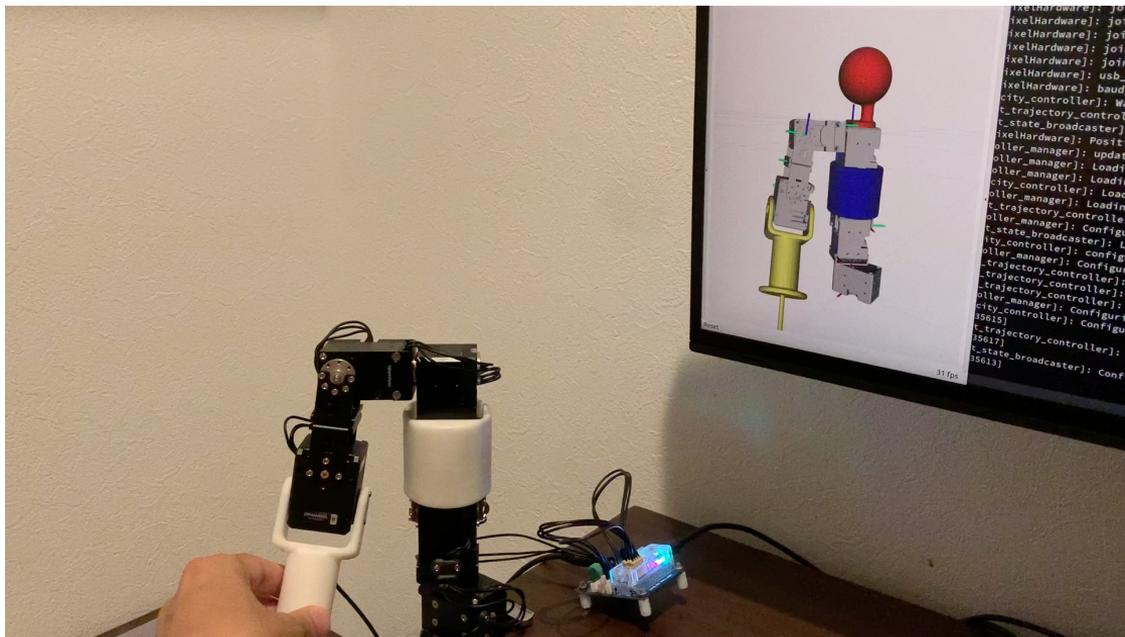
ignition\_ros2\_control (的なパッケージ) を  
早くリリースしてください🙏

# 2XC430-W250対応

- 1つの筐体に2軸分のモータが入ったDynamixelへの対応
  - <https://github.com/ROBOTIS-GIT/dynamixel-workbench/pull/342> (レビュー待ち)



<https://emanual.robotis.com/docs/en/dxl/x/2xc430-w250/>



<https://twitter.com/youtalk/status/1435953228122705920>

# dynamixel-workbenchからの卒業

- dynamixel-workbenchのros2ブランチは2年以上更新なし...

- <https://github.com/ROBOTIS-GIT/dynamixel-workbench/tree/ros2>



- Plan A : Dynamixel SDKの直接利用

- ROBOTIS社Will Sonさん（ROS 2 TSCメンバ）とともに作業進行中

- Plan B : OUXT-Polaris/dynamixel\_hardware\_interfaceへの移行

- [https://github.com/OUXT-Polaris/dynamixel\\_hardware\\_interface/pull/14](https://github.com/OUXT-Polaris/dynamixel_hardware_interface/pull/14)

- Plan C : 誰か手伝ってください 🙏

# dynamixel\_control

[https://github.com/youtalk/dynamixel\\_control](https://github.com/youtalk/dynamixel_control)



The screenshot shows the GitHub repository page for `youtalk/dynamixel_control`. The repository is public and has 3 Unstars, 38 forks, and 4 pull requests. The main branch is selected. The file list includes:

File/Folder	Description	Time
<code>dynamixel_hard...</code>	remove ids param and add id params	14 hours ago
<code>open_manipula...</code>	remove ids param and add id params	14 hours ago
<code>pantilt_bot_des...</code>	remove ids param and add id params	14 hours ago
<code>.clang-format</code>	add clang-format	5 months ago
<code>.gitignore</code>	remove ros2_controllers	4 months ago
<code>LICENSE</code>	move to root	4 months ago
<code>README.md</code>	Update README.md	14 hours ago
<code>dynamixel_cont...</code>	remove ros2_controllers	4 months ago

The right sidebar shows the repository description: "ros2\_control packages for ROBOTIS Dynamixel". It includes tags for `cpp`, `robotics`, `dynamixel`, `gazebo`, `ros2`, `robotis`, `gazebo-ros`, `ros2-foxy`, and `ros2-control`. There are no releases published.



**Making the real world computable**