

ROSCon JP 2023

# ROS2を用いたKHI製産業用ロボットアーム制御

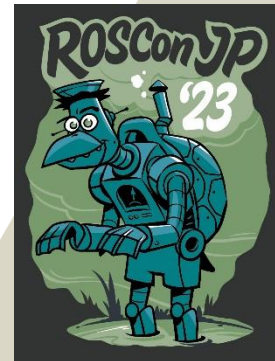
2023/09/26

川崎重工業株式会社

技術開発本部

システム技術開発センター

システム基盤技術開発部 第三課



カワる、  
サキへ。

Changing forward

# 目次

～ROS2を用いたKHI製産業用ロボットアーム制御～

---

01

川崎重工のロボット

p.03-p.06

---

02

発表サマリー

p.07

---

03

システム構成

p.08-p.09

---

04

開発内容

p.10-p.15

---

05

動作品質

p.16-p.20

---

06

まとめ

p.21

---

# 川崎重工が描くロボットビジネス

今後懸念される世界的な労働力不足を解決するソリューションサービスの提供を目指しています

## Nyokkey



## 当社産業用ロボット



<https://answers.khi.co.jp/ja/connected-society/20220527j-01/>

# 産業用ロボットの用途と特徴

産業用ロボットメーカーのパイオニアとして、  
自動車業界や電機・電子業界をはじめとするさまざまな業界向けに、  
**溶接、組立・ハンドリング、塗装、荷物積み下ろし**など多数のロボットを供給しています

## 荷物積み下ろし



## 溶接



当社産業用ロボットは当社専用ロボット制御言語(**AS言語**)を用いることで、  
ティーチングされた動作を高い精度かつ高速で繰り返すことが可能

# ご利用シーンの拡大に向けた取り組み

当社ロボットをROSで制御したいとのご要望に応じ、  
Kawasaki Robotics/khi\_robot (ROS1)を公開中

カワる、  
サキへ。  
Changing forward

機能開発や研究目的で利用いただいています

## 対象機種



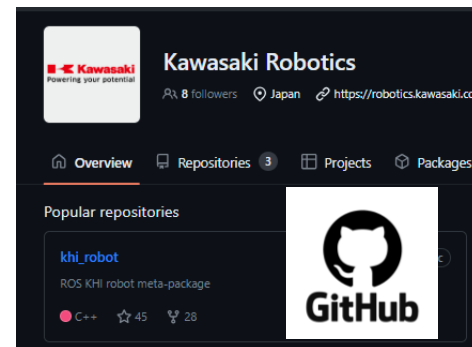
RSシリーズ

小型・中型汎用ロボット



duAro

人共存型双腕スカラロボット



# ROS2向けパッケージ公開のご紹介

FY2023. **KHI制御パッケージ(ROS2)**を公開予定

## 対象機種



RSシリーズ

小型・中型汎用ロボット

## リリース情報

公開時期

2024.03 (予定)

ディストリ  
ビューション

Foxy・Humble

対象機種

RSシリーズ

用途

研究開発

# KHI制御パッケージ(ROS2)の開発概要

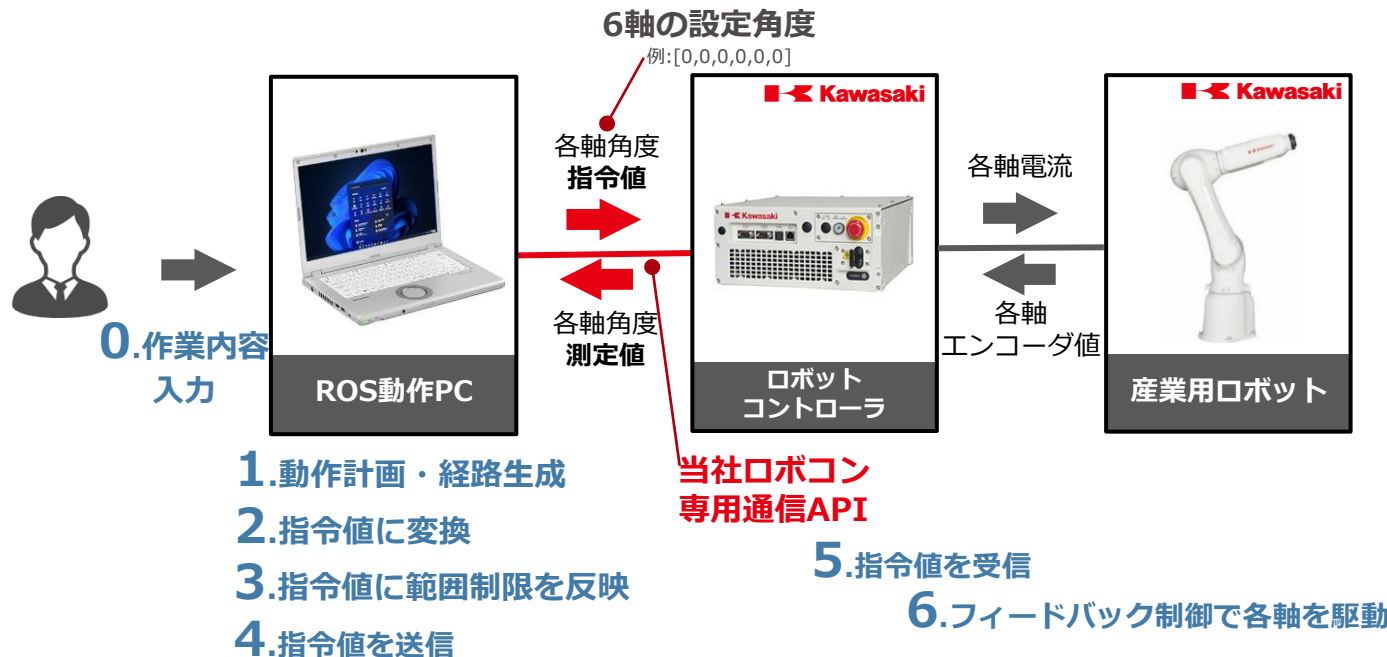
**産業用ロボットの動作品質**に留意しながら

ROS1と構成が異なるROS2パッケージの開発をおこなっています

	定義	KHI制御パッケージ (ROS2) 特記	
安定性	意図しない速度急変や 非常停止等が発生せず <b>連続動作</b> すること	<b>意図しない異常停止を防ぐ機能を導入</b>	● 本発表
再現精度	<b>目標位置姿勢</b> の再現精度	公開中のROS1パッケージと同程度	● 本発表
動作速度	教示された動作を実行するときの <b>ロボット手先速度</b> (ツール先端速度)	公開中のROS1パッケージと同程度	● 本発表
安全性	ロボットコントローラの状態と 連動した緊急停止機能	ROS2上でロボットコントローラを 制御するためのドライバ状態を定義	開発中

# システム構成と各部説明

ROSパッケージを導入した外部PCで経路計画を行い、当社ロボットコントローラに動作指令を連続送信し、6軸のロボットアームを制御します

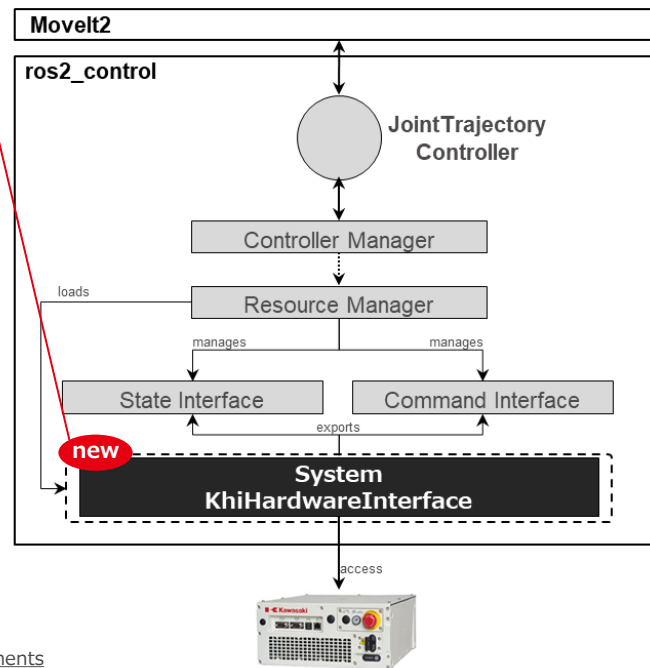
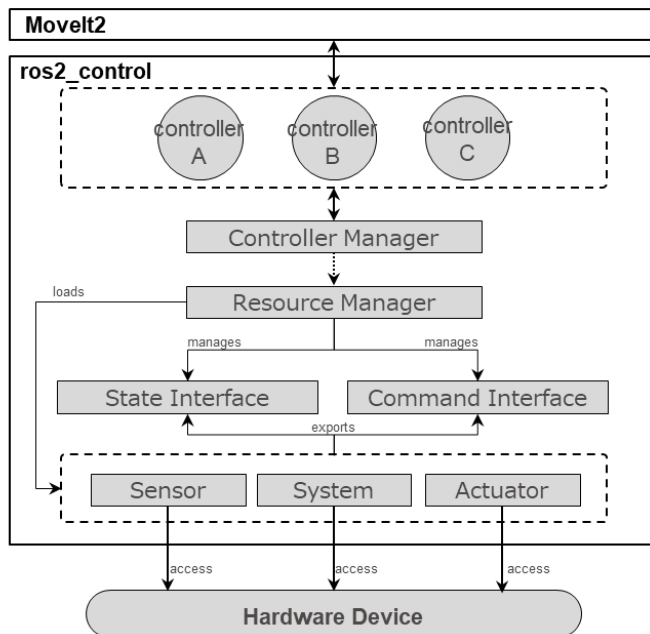




# KHI制御パッケージ(ROS2)の構成図

公開パッケージ“**ros2\_control**”を基本構成とし

KHIロボットに接続するために“**System/KhiHardwareInterface**”クラスを開発しました



[https://control.ros.org/foxy/doc/getting\\_started/getting\\_started.html#hardware-components](https://control.ros.org/foxy/doc/getting_started/getting_started.html#hardware-components)



# KHIロボットのための制約と設定

“System/KhiHardwareInterface” には

当社ロボコン専用通信APIを含み、ハード的制約を各部で設定しました

Time Parametrization Algorithm

```
moveit::planning_interface::
MoveGroupInterface::Plan
```

ライブラリ指定

rs007l\_joint\_limits.yaml

```
joint_limits:
  joints:
    hes_position_limits: true
    min_position: -3.14159265359
    max_position: 3.14159265359
    hes_velocity_limits: true
    max_velocity: 0.4577
    hes_acceleration_limits: true
    max_acceleration: 5.65950345333
```

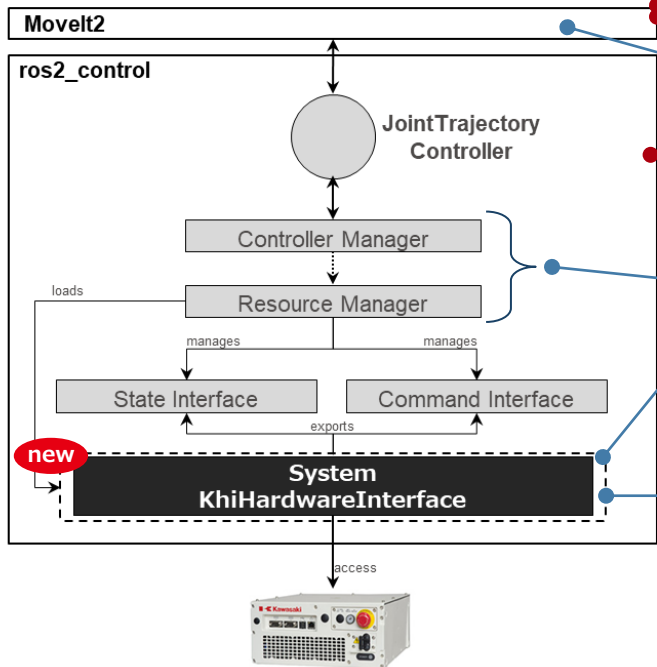
当社ロボット用初期設定値

- ・ 軸動作範囲
- ・ 軸速度上限
- ・ 軸加速度上限
- ・ ...

```
controller_manager:
  ros_parameters:
    update_rate: 100 # Hz
    update_rate:
  rs007l_arm_controller:
    type: joint_trajectory_control
```

当社ロボコン用設定値

- ・ 制御周期
- ・ ...



## 1. 動作計画・経路生成

rs007l\_controller.yaml

## 2. 指令値に変換

## 3. 指令値に範囲制限を反映

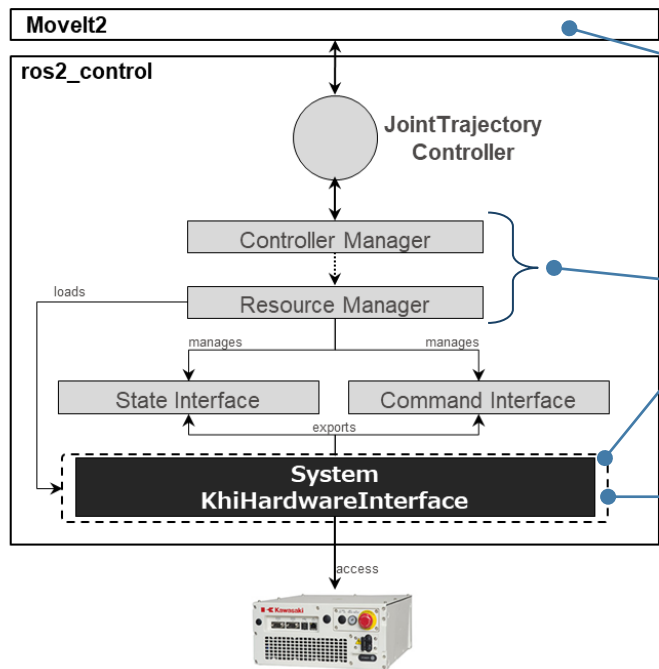
ros2\_controlとして、  
速度上限などを超えないように指令値を調整する

## 4. 指令値を送信

当社ロボコン専用通信APIが指令値の速度上限を  
チェックする。上限を超えるとエラー停止する。

# KHI制御パッケージ(ROS2)の特徴

リアルタイムに指令値を受信することを想定している  
当社ロボコンに適合するように、**送信機能を強化しました**



1.動作計画・経路生成

2.指令値に変換

3.指令値に範囲制限を反映

4.指令値を送信

**ロボコンが求める制御指令値を確実に  
送信するよう実装**

# 当社ロボコンの受信特性

“ros2\_control”の動作計画を  
ロボコンが正しく解釈できない場面を考慮しています



1. 動作計画・経路生成

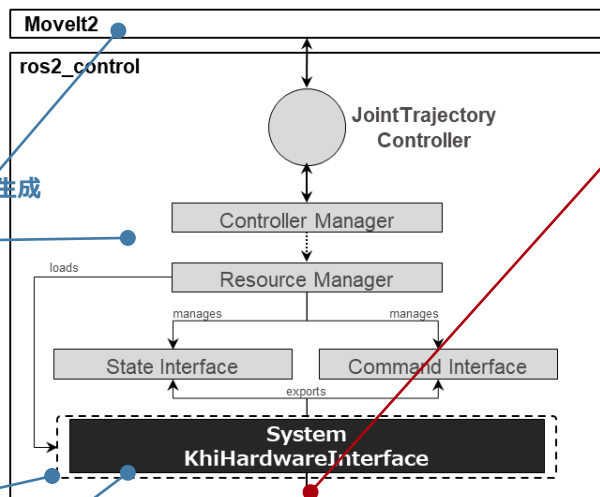
2. 指令値に変換

3. 指令値に範囲制限を反映

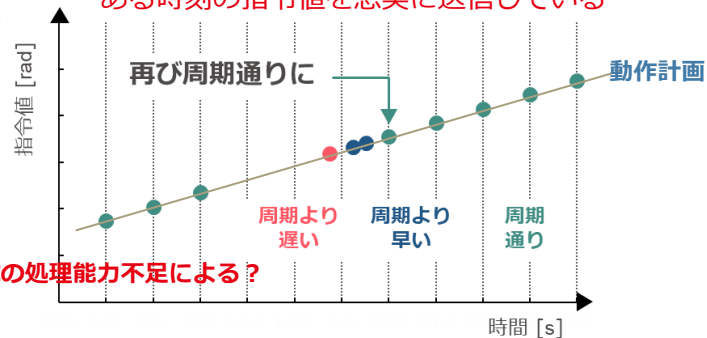
速度上限などを超えないように指令値を調整

4. 指令値を送信

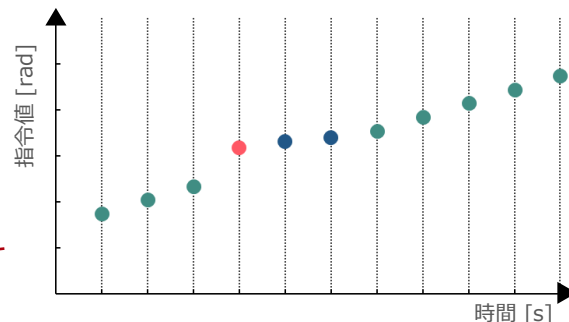
当社ロボコン専用通信APIが指令値の速度上限をチェックする。  
不可の場合は非常停止となる



一定の制御周期が保てないときでも、動作計画通りに、  
ある時刻の指令値を忠実に送信している



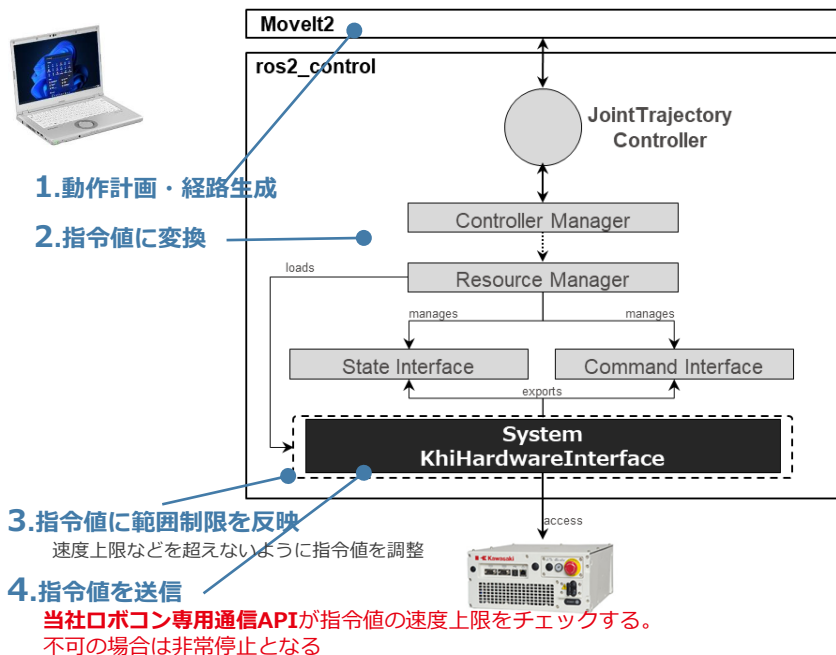
制御PCの処理能力不足による?



一定の制御周期で処理をすることが前提なので、  
図のような解釈で、ロボットを動作させようとする

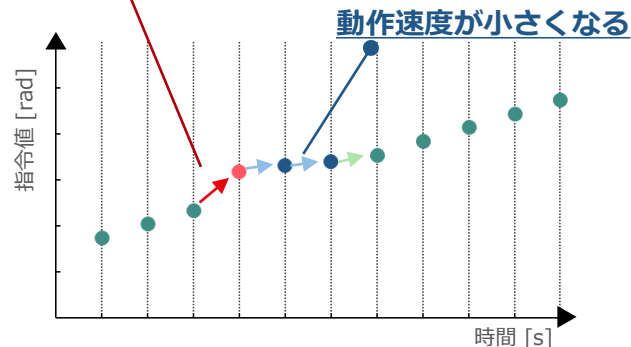
# ros2\_controlの制御周期が不安定で生じる問題

ロボットアームの動作速度に急変が発生し、異常停止の要因になります



## 動作速度が大きくなる

極端に大きくなる場合、通信専用APIが不可を判定しロボット動作が停止してしまう可能性がある



# 周期補正機能の導入

他のROSパッケージに影響を与えず、  
本来のインターロック機能を無効にしない方法にしました



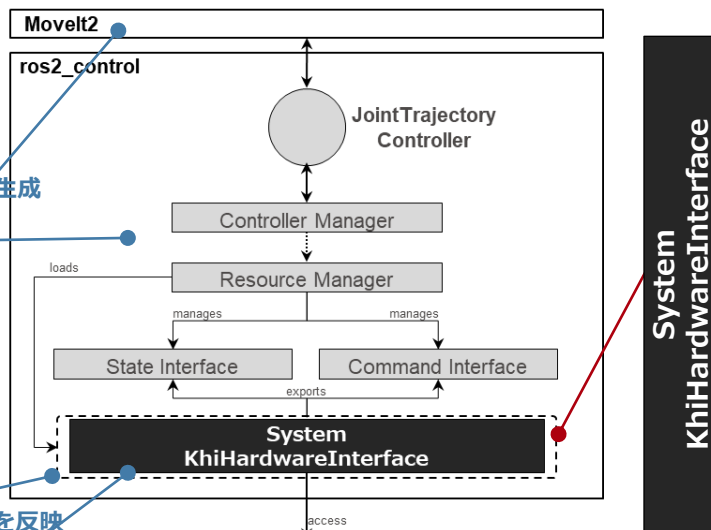
1. 動作計画・経路生成

2. 指令値に変換

3. 指令値に範囲制限を反映  
速度上限などを超えないように指令値を調整

4. 指令値を送信

当社ロボコン専用通信APIが指令値の速度上限をチェックする。  
不可の場合は異常停止となる



補正機能なし

(不定期の) 各軸角度指令値

専用通信APIの  
速度上限判定

非常停止要因

専用通信APIで送信

補正機能あり

(不定期の) 各軸角度指令値

周期補正機能

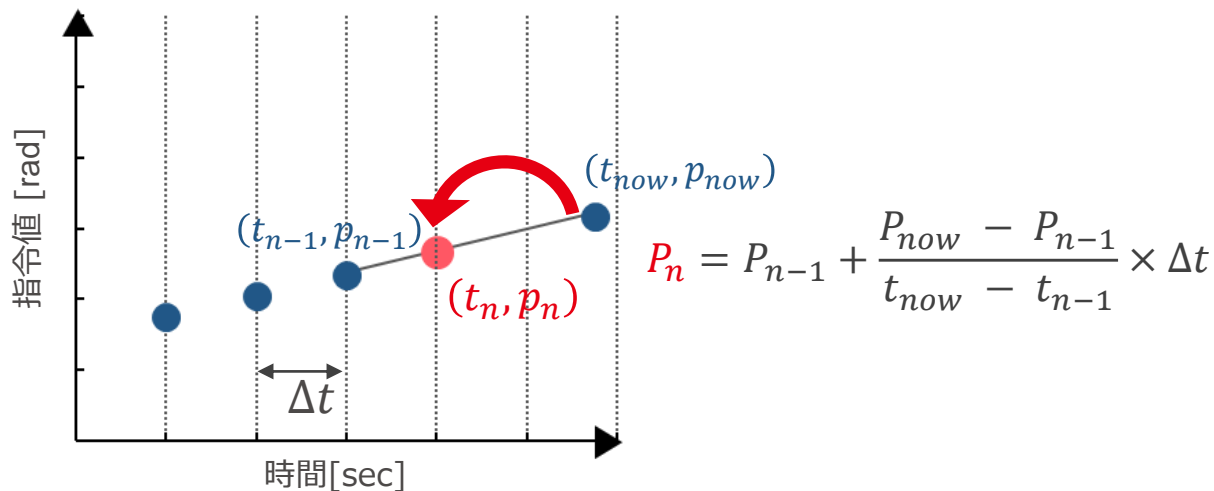
(定周期の) 各軸角度指令値

専用通信APIの  
速度上限判定

専用通信APIで送信

# 周期補正機能の詳細

不定期で送られてくる各軸角度指令値に対して、直線補完で周期相当の値を算出しました



最新値( $t_{now}, p_{now}$ )と、1回前の送信値( $t_{n-1}, p_{n-1}$ )から  
直線補間を用い周期相当の軸値を算出

## 補正機能あり

(不定期の) 各軸角度指令値

**周期補正機能**

(定周期の) 各軸角度指令値

専用通信APIの  
速度上限判定

専用通信APIで送信

# 補正機能による安定性の比較

補正機能が無い場合はしばしば異常停止が発生しますが、補正機能がある場合は停止することなく動作します

## 積み下ろし動作を繰り返し行った際の 途中停止確率

(停止回数/繰り返し動作回数)

補正無

補正有

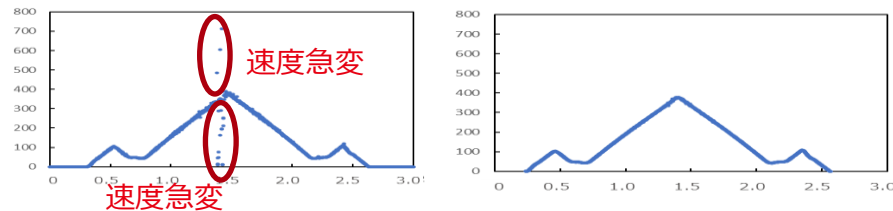
停止率

4.3%

0.0%

20/470

0/100



ツール先端速度の時間変化



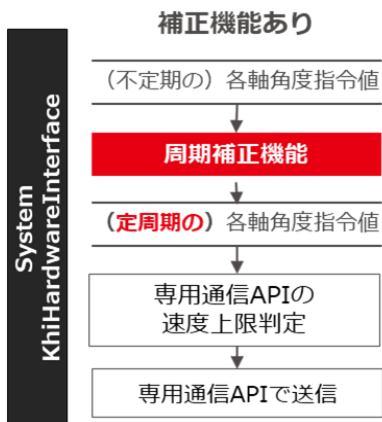


# 動作品質の確認

今回開発したKHI制御パッケージ(ROS2)の再現精度と動作速度を調査しました

## 安定性

### 確認済み



## 再現精度



手先の  
 ・座標のズレ  
 ・軸のズレ

- ・ 目標への到達精度確認
- ・ 軌跡確認

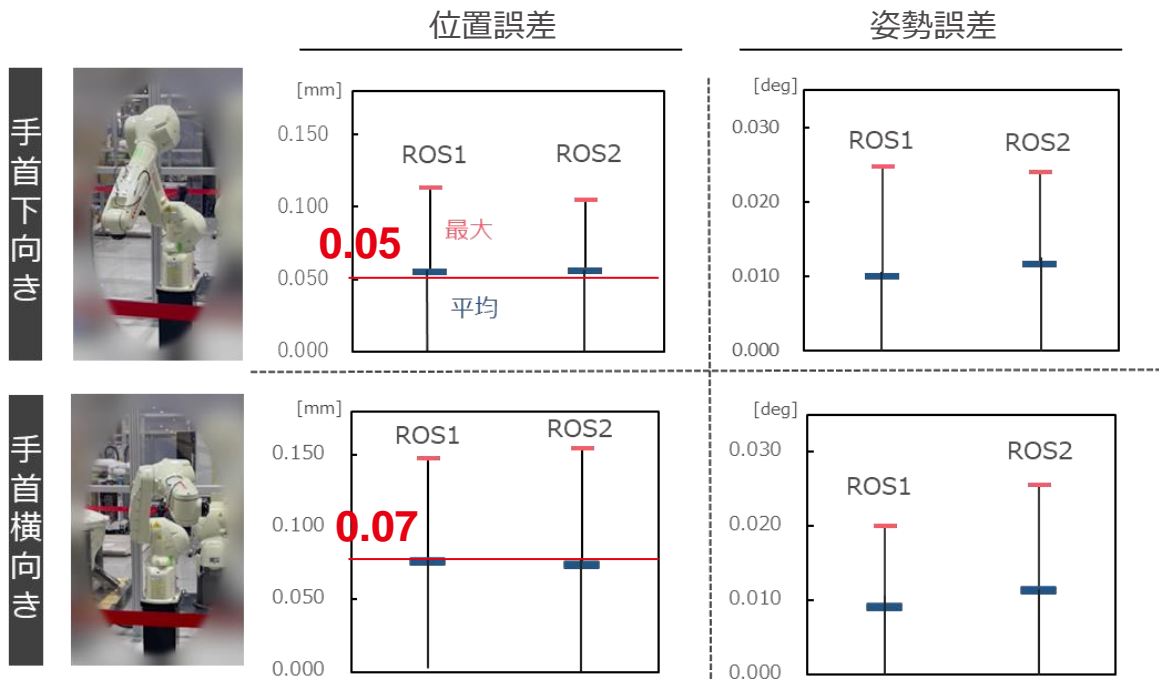
## 動作速度

```
controller_manager:
  ros_parameters:
    #update_rate: 100 # Hz
    update_rate:
      rs0071_arm_controller:
        type: joint_trajectory_control
```

ハード的制約から定めた  
KHI制御パッケージの設定値に  
おける**サイクルタイム**を測定

# 目標への到達精度確認（再現精度）

ROS制御パッケージ（ROS1,ROS2）はともに指令値に対して、**0.07mm程度の誤差で実動作していることを確認しました**



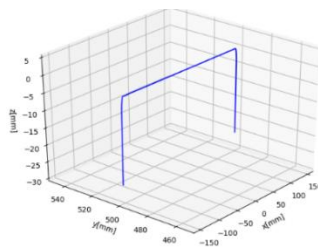
# 軌跡確認 (再現精度)

計画した軌跡通りに、実動作していることを確認できました

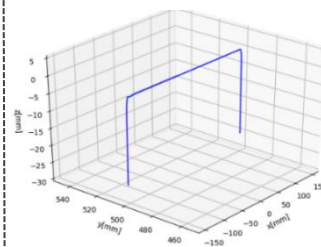


軌跡

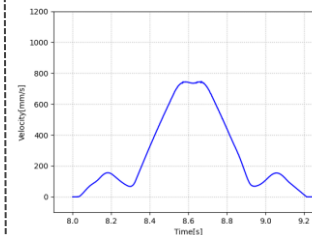
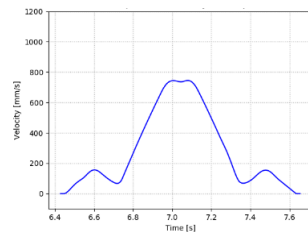
ROS1



ROS2



速度変化



# サイクルタイムの測定（動作速度）

KHI制御パッケージ(ROS2)は、KHI制御パッケージ(ROS1)と同程度であることを確認しました



ROS1

1.19±0.015秒



ROS2

1.19±0.024秒

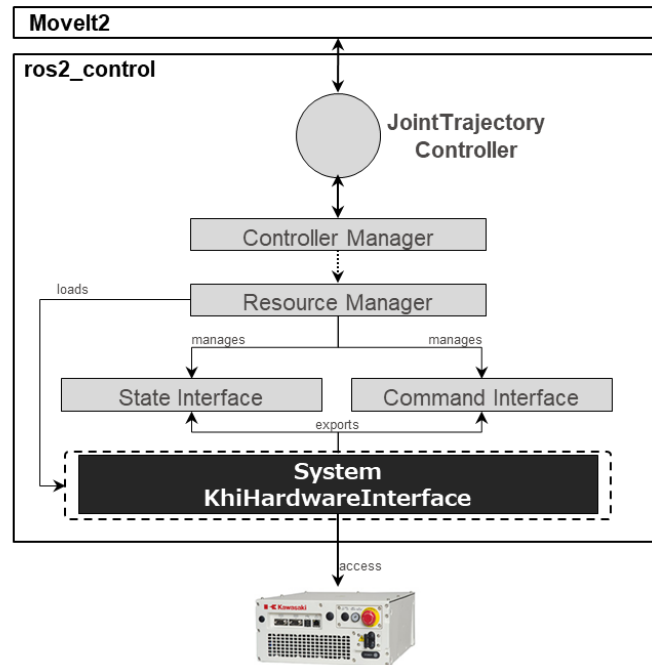


N=300

# まとめ

- 当社産業用ロボット（RSシリーズ）のKHI制御パッケージ（ROS2）を作成しました
- 制御PCの性能に依らず、KHI制御パッケージ（ROS2）の連続動作の安定性を高めました
- 再現精度・動作速度についてKHI制御パッケージのROS1とROS2は同程度であることを確認しました

**KHI制御パッケージ（ROS2）を是非使ってみてください！！！！**



世界の人々の豊かな生活と地球環境の未来に貢献する  
“Global Kawasaki”