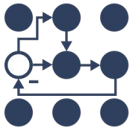
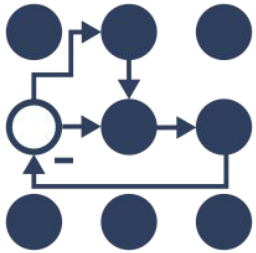
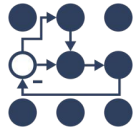


ros-controls project update



Denis Stogl, Bence Magyar

Thank you for being here!



Bence Magyar, Denis Štogl, Christoph Froehlich, Sai Kishor Kothakota, Alejandro Hernández Cordero, Karsten Knese, Jordan Palacios, Shane Loretz, Dave Coleman, Jaron Lundwall, Jonathan Bohren, Felix Exner, Victor Lopez, Paul Gesel, Tyler Weaver, Manuel Muth, Julia Jia, Olivier Stasse, Soham Patil, Marq Rasmussen, Noel Jiménez García, Reza Kermani, Silvio Traversaro, Wiktor Bajor, Márk Szitanics, Andy Zelenak ..... and many more!

ros-controls Organization

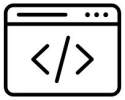
Charter

Project charter

Meetings

CEST

Repositories

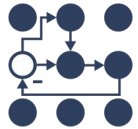


GitHub ros-controls

Docs

control.ros.org

ros-controls Organization



Committers: ...

Repositories: ...

Strategy:

- Releases
- Versioning
- Public API
- Deprecations

https://control.ros.org/rolling/doc/governance/governance.html

Thanks to our maintainers!

Bence Magyar – [Dr. Bent'seh]

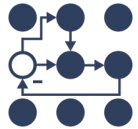
Denis Štogl – [Dr. Denis]

Sai Kishor Kothakota – [The Code-Wizard]

Christoph Fröhlich – [Dr. Christoph]

*all from different companies

Code Management Strategy - **tl;dr**



- We don’t guarantee ABI stability (Rebuild after any upstream package update!).
- We allow code to be deprecated in every release (don’t use -Werror=deprecated-declarations).
- We define the release within a distro “stable” at **October 1st** after an official distro release. The goal is that we get the **stable release available by ROSCon**.
- We still allow **API breaks and behavior breaking** changes within such stable releases in case of **safety concerns**.
- We continually try to give useful hints in the migration guides and deprecation notices.

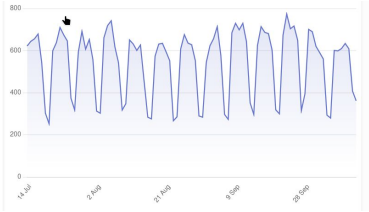
Did you know? *You can always build the rolling version, including the latest features, back on all active distros!*

https://discourse.openrobotics.org/t/releases-versioning-and-public-api-claim-of-ros-controls/50388

Other **perks** for your project!

1. **ros2_control_ci** - reusable CI templates *(you can use it too!)*
2. **ros2_control_cmake** - reusable CMake definitions *(clean up your files from boilerplate!)*
3. Pre-commit ❣️ *(linting and testing are two separate stages!)*
4. Documentation is placed next to the code! (*Easy to convince people to actually write it!)*

Enjoy public stats: https://control.ros.org/rolling/doc/statistics.html



control.ros.org stats 2025 October (YTD)

Unique visitors: 161k (105k)

Total pageviews: 673k (476k)

Total visits: 202k (144k)

Singapore: 5.2k (compared USA: 24.5k)

5. Roadmap repository: https://github.com/ros-controls/roadmap *(actual design drafts)*
6. Repository with demos: https://github.com/ros-controls/ros2_control_demos *(reference code)*


r/ROS • 2y ago
to ros2_control or to not ros2_control

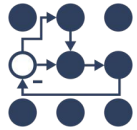

ROS


NAV 2
OPEN
NAVIGATION


MoveIt2

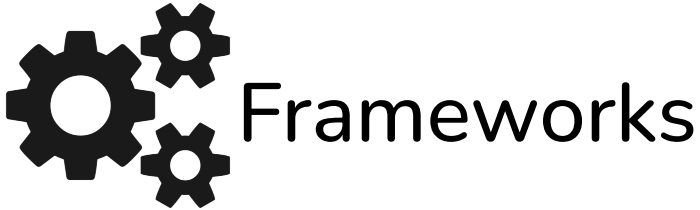

OpenCV


pcl




Open-RMF

dora-rs


Frameworks


Pinocchio
Efficient and versatile rigid body dynamics algorithms


YARP


RBDL


GAZEBO


NVIDIA
ISAAC


MuJoCo


MATLAB

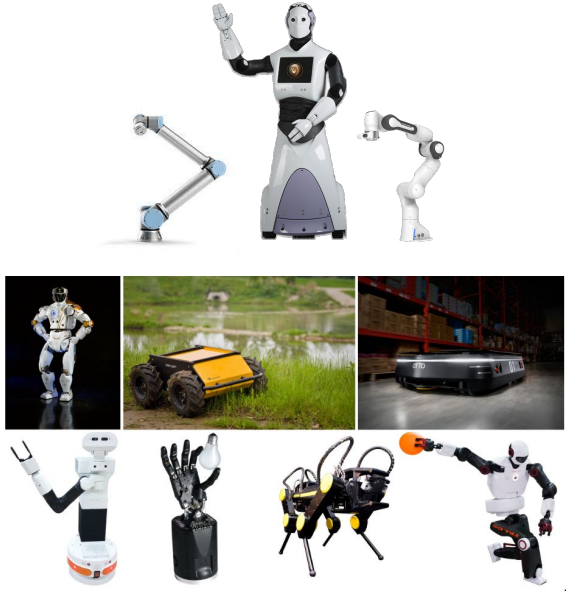
History

pr2_controller_manager

(pr2_mechanism)

ros_control

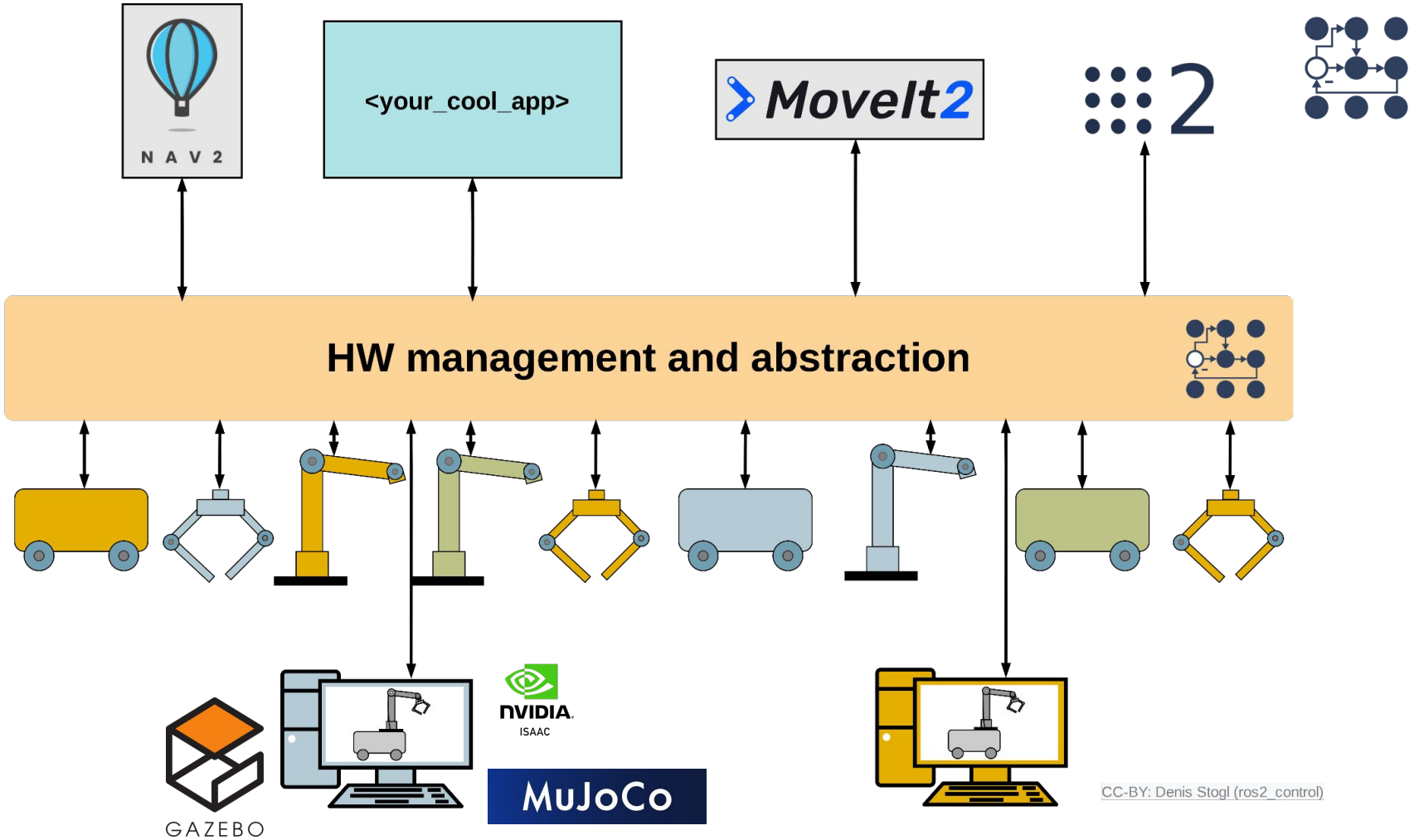
2012/2017

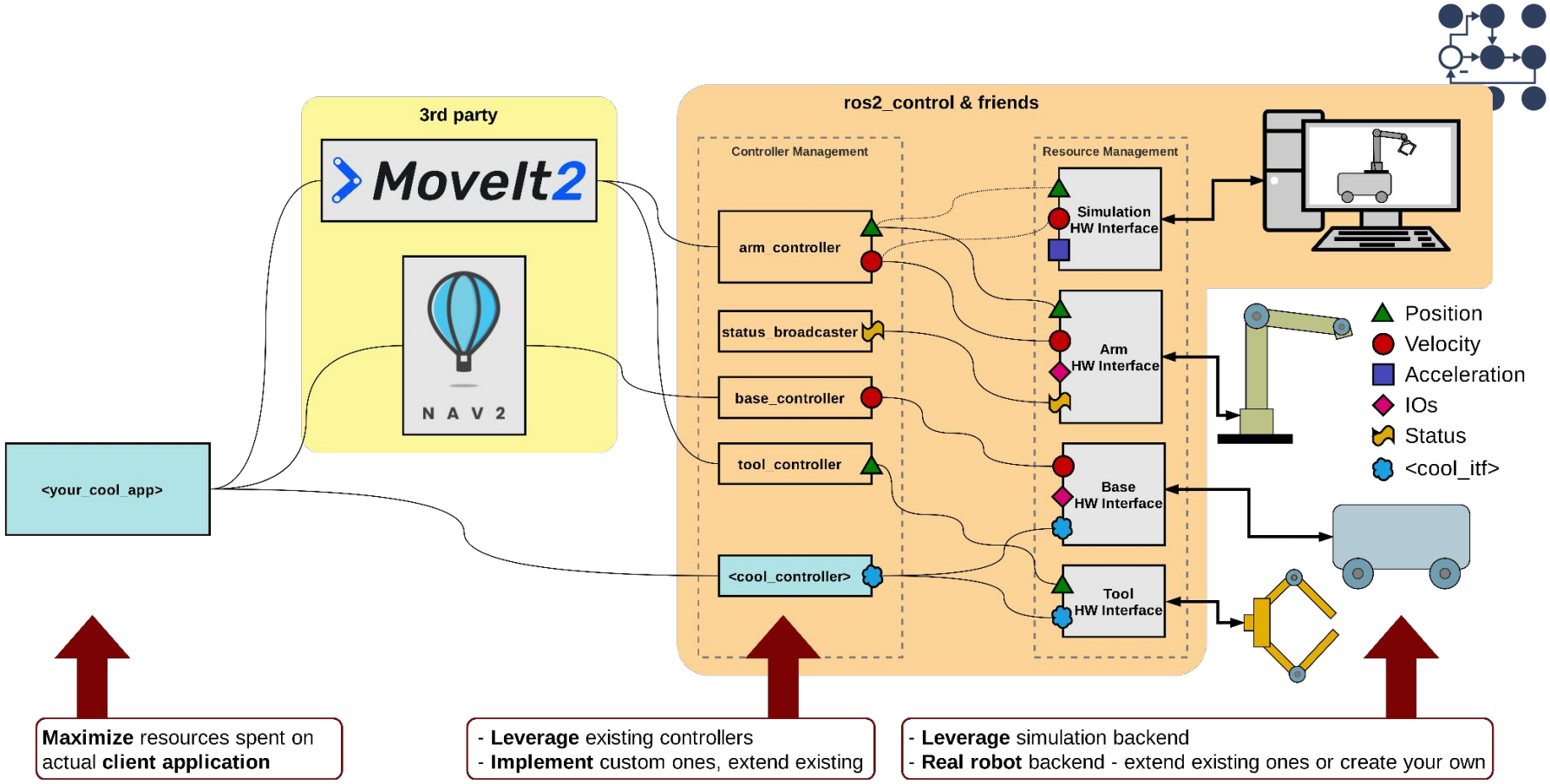


ros2_control

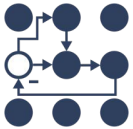
2017/today

https://control.ros.org/master/doc/supported_robots/supported_robots.html




3rd party
MoveIt2
N A V 2
ros2_control & friends
Controller Management
Resource Management
arm_controller
status_broadcaster
base_controller
tool_controller
<cool_controller>
Simulation HW Interface
Arm HW Interface
Base HW Interface
Tool HW Interface
<your_cool_app>
Position
Velocity
Acceleration
IOs
Status
<cool_itf>
Maximize resources spent on actual **client application**
- **Leverage** existing controllers
- **Implement** custom ones, extend existing
- **Leverage** simulation backend
- **Real robot** backend - extend existing ones or create your own

Overview of Standard Controllers



Generics

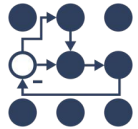
- PID Controller
- Forwarding Controller
    - Forward Command – multiple joints, one interface
    - Multi Interfaces Fwd. Cmd. – one joint, multiple interfaces
- GPIO Command Controller - sends values on set of GPIO interfaces

Mobile Robots - ***Nav2***

- Steering Controllers
    - Bicycle – 1 drive joints, 1 steering joint
    - Tricycle – 2 drive joints, 1 steering joint
    - Ackerman – 2 drive joints, 2 steering joints
- Omni Wheel Drive
- Tricycle controller (1 drive + steering joint)
- Mecanum drive
- **Differential Drive (Diff drive)** / Skid steer

https://control.ros.org/rolling/doc/ros2_controllers/doc/controllers_index.html

Overview of Standard Controllers



Industrial Robotics (Arms) - ***MoveIt2***

- Joint Trajectory Controller (JTC) - scaled 🎊
  - The most used one – interface for MoveIt2 and similar frameworks
- Admittance Controller – force-position control in Cartesian space (using IK library from KDL)
- (Industrial) Motion Primitives Controllers - move LIN, PTP, CIRC

Grippers / Tools

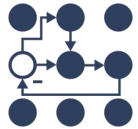
- Parallel Gripper Controller – 1 DoF gripper with position and optionally max vel and max effort interfaces
- GPIO Tool Controller - generic tools and grippers (engaging, disengaging, and configuring)

Not Controllers → Broadcasters

- Joint State Broadcaster – **nothing works without it!!!**
- Force Torque Sensor Broadcaster
  - Has funky stuff in it, like filtering—cool for using in chain ahead of Admittance Controller
- IMU, GPS, Battery, Range Sensor Broadcaster
- Pose Broadcaster

https://control.ros.org/rolling/doc/ros2_controllers/doc/controllers_index.html

Overview of Hardware Components (Drivers)



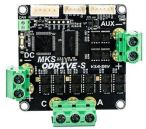
Automation / Communication

CANopen

EtherCAT

ctrlX AUTOMATION

BECKHOFF TwinCAT ADS



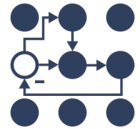
Non-robot devices

End-effectors

DG 5F

DG 4F

https://control.ros.org/rolling/doc/supported_robots/supported_robots.html



Overview of Hardware Components (Drivers)

Official Robot Drivers









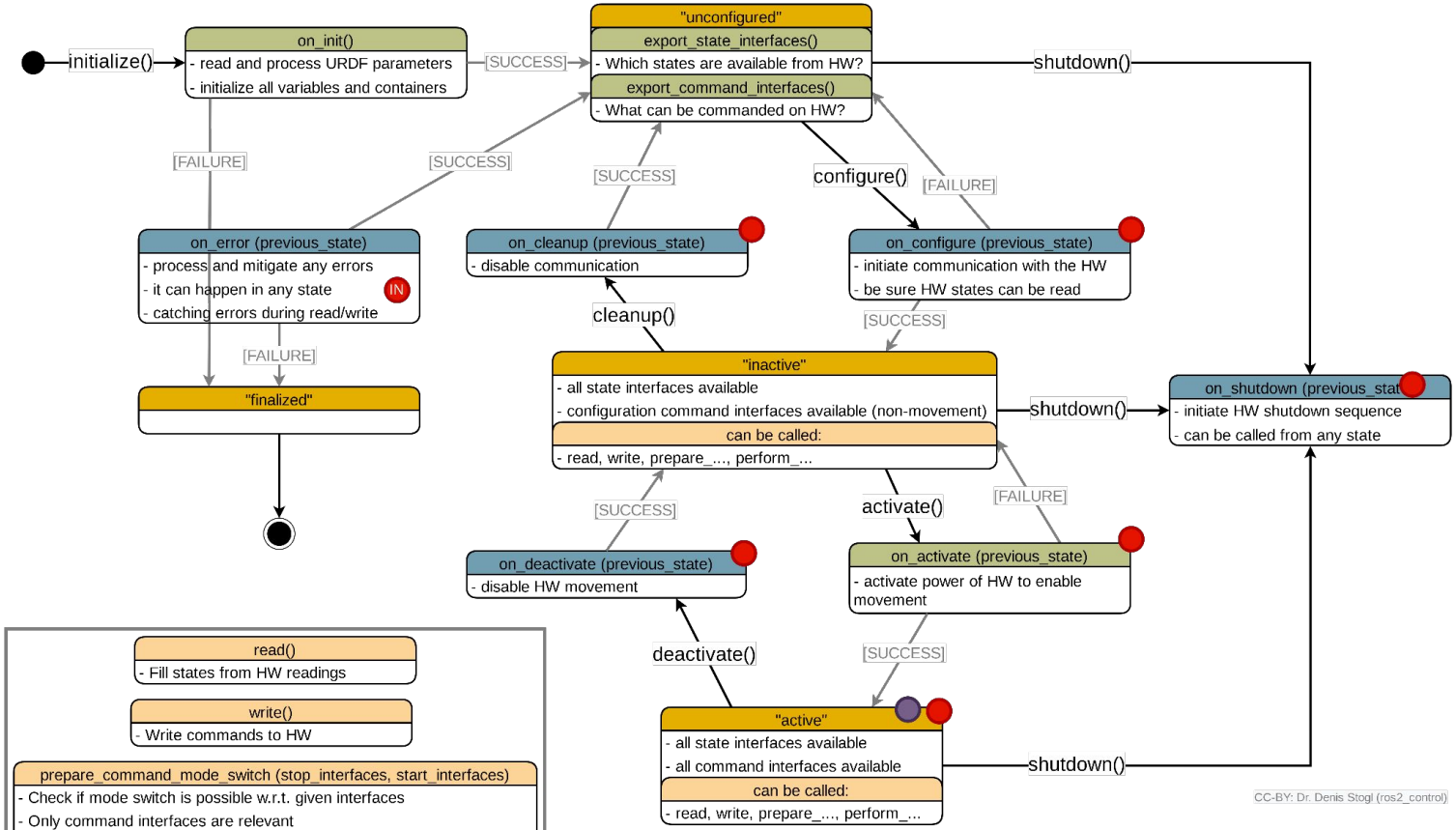





From Community



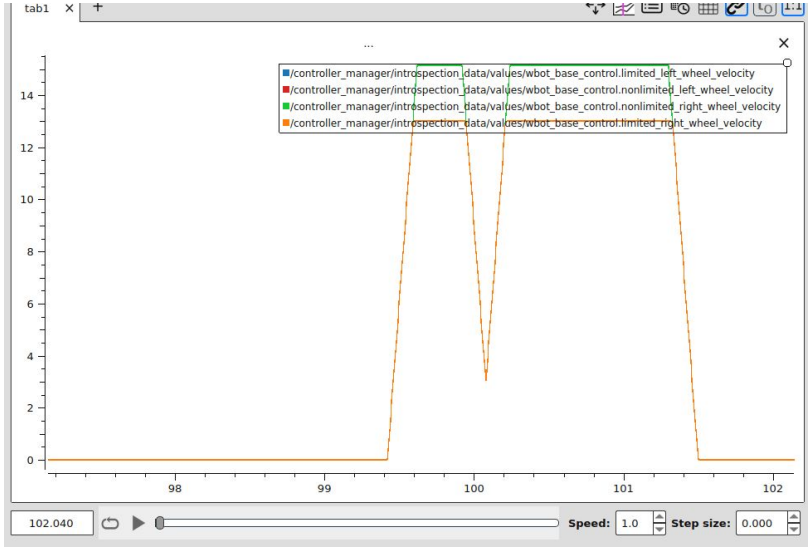

https://control.ros.org/rolling/doc/supported_robots/supported_robots.html

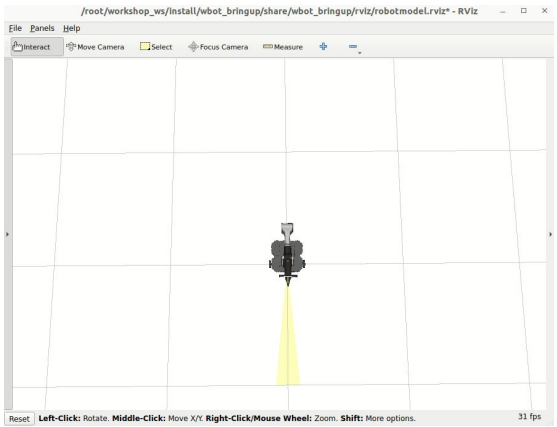


Lifecycle everywhere!

Workshop

- Successful workshops with 100+ people in total at ROSCon and ROSCon UK 2025!
- ESP32 board speaking ROS natively over Zenoh Pico
- Setup via ros2_control & standard controllers
- We () want to push for more embedded support





2025 Workshop repo

TOP 5 CHARTS

— SUMMER 2025 —

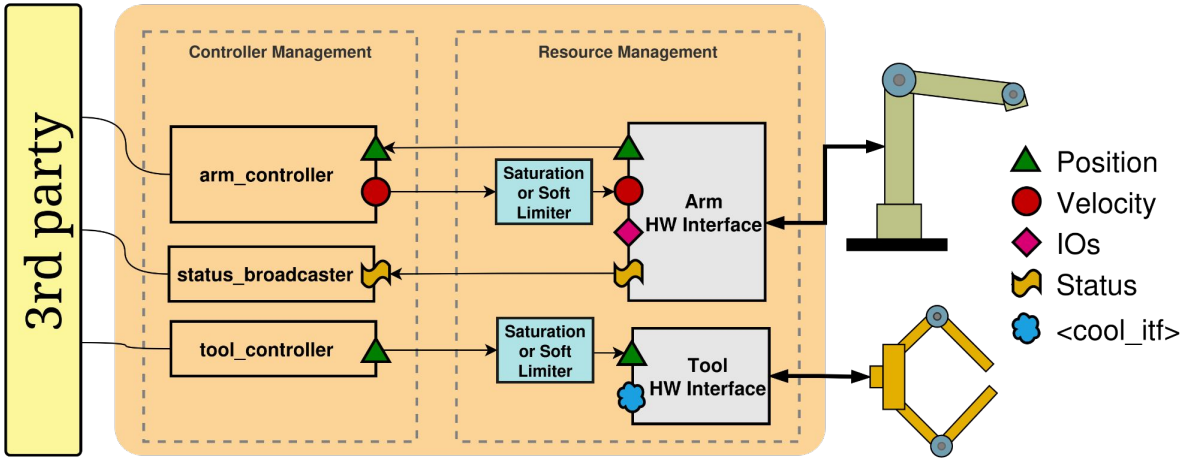
ros2_control

Joint Limits Enforcement

- Per joint limits
- Saturation, Range and Soft Limiters
- Joint Limits definitions in URDF
- <ros2_control>-tag for acceleration and jerk
- Available from *Jazzy—default* “on” from *kilted*

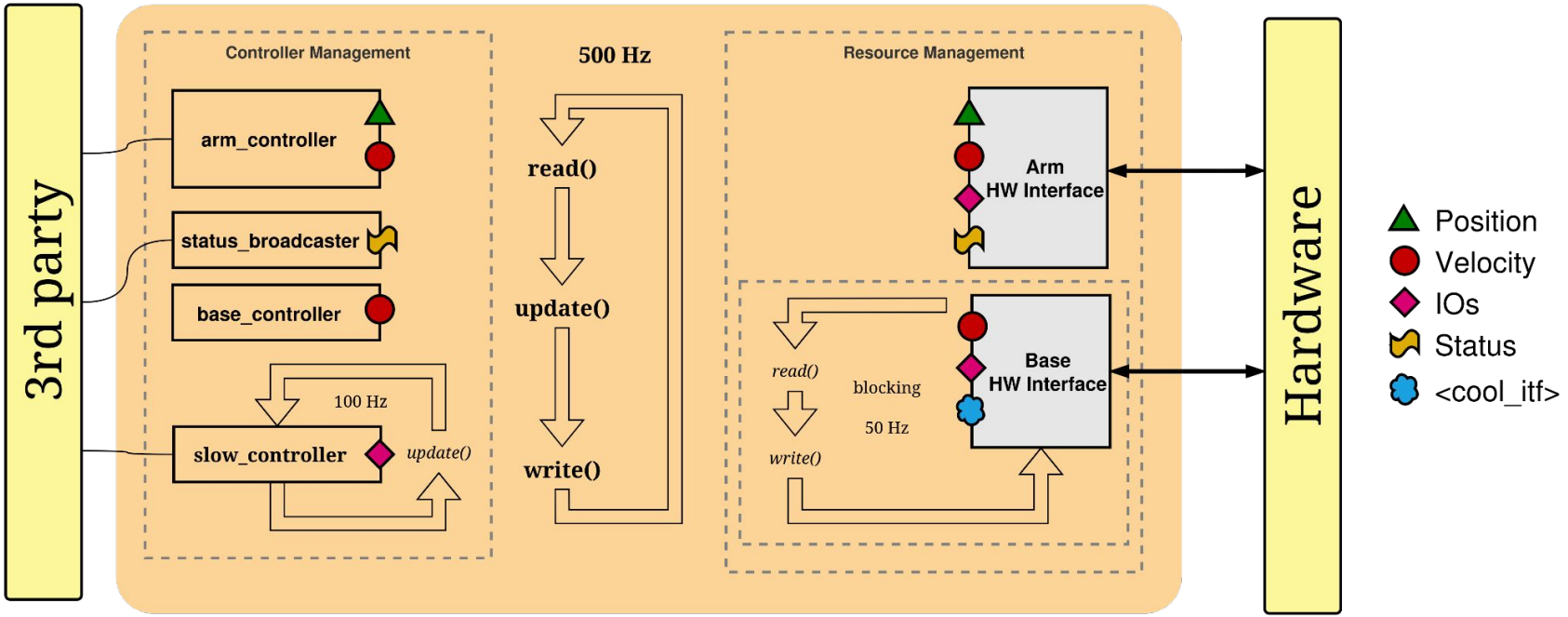
Limitations

- No sync between robot’s joints → Possible offsets in the executed trajectory if some joints are limited
- *Jerk* limiting is not fully implemented to influence its “integrals”



Asynchronous Hardware Components

```
<ros2_control name="MyBase" type="system" is_async="true" update_rate="50">
  <hardware>
    <plugin>my_hw_itf_pkg/BaseHWInterface</plugin>
  </hardware>
  ....
```

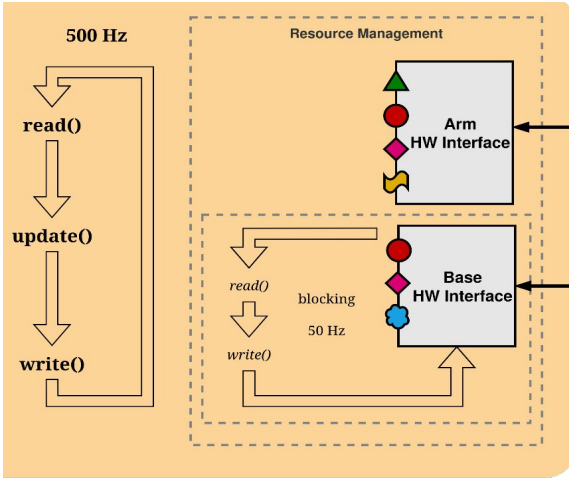


Real-time improvements

- Locking memory, CPU affinity, thread priority
- ros2_control node
- async controllers and HW components
- Monotonic clock in RT loop

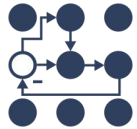
Scheduling policy

- *synchronized*—CM triggers slower loop when ready
- *detached*—independent from CM thread




```
<ros2_control name="MyBase" type="system" is_async="true" update_rate="50">
  <properties>
    <async affinity="[2,4]" scheduling_policy="synchronized" print_warnings="true" thread_priority="30"/>
  </properties>
  <hardware>
    <plugin>my_hw_itf_pkg/BaseHWInterface</plugin>
  </hardware>
  ...
```

Diagnostics and Introspection



- */controller_manager/statistics/** topics and /diagnostics topic
  - execution time and periodicity of everything RT related
  - *read()*, *update()*, *write()* for individual controllers and HW components

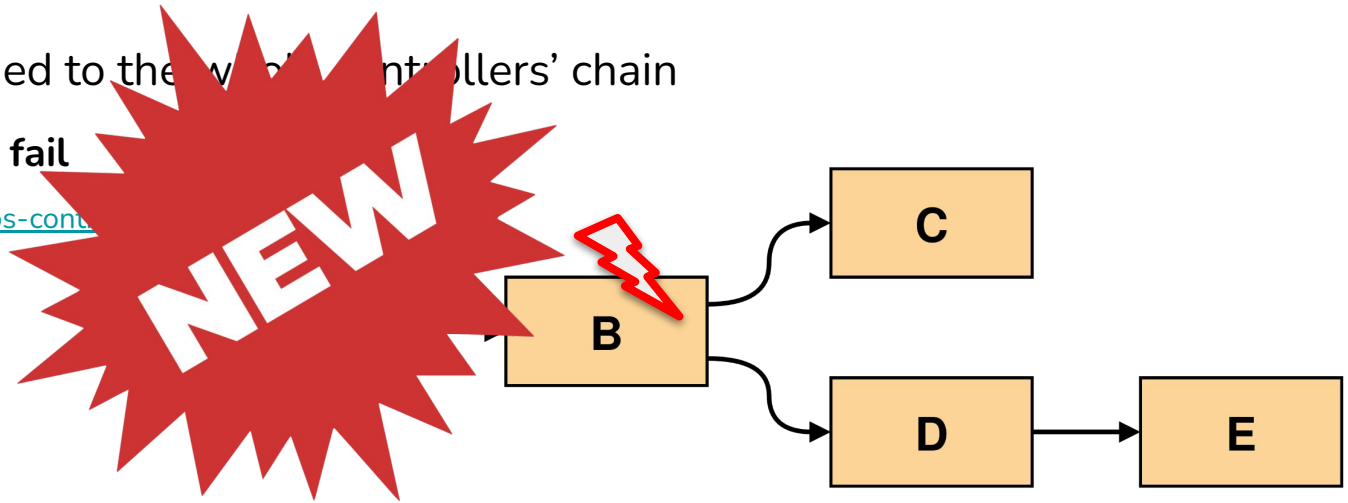
- */controller_manager/activity* topic
  - The latest state of controllers and the hardware components

- */controller_manager/introspection/** topics
  - Contain values handshaked between controllers and HW components directly
  - Contains information on which interfaces are limited

Being “strict” means being “safer”

- Parameter for overriding default strictness for activating controllers
  - *humble*, *jazzy*, *kilted*—framework default is *best-effort*
  - *rolling*—framework default is *strict*

- Strict is now applied to the w[illegible]llers’ chain
  - **any fails → all fail**
  - https://github.com/ros-cont[illegible]

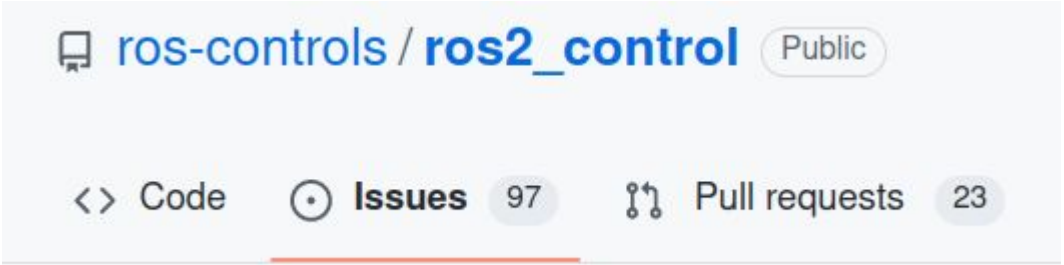


Join us!

Working Group Meetings
every second Wednesday!

Next one is 5th November!

- ros2_control presentations
  - https://control.ros.org/master/doc/resources/resources.html



- Github project to guide contributors to where they are most needed
  - https://github.com/orgs/ros-controls/projects/11

ros2_control reviewers

27 members