

Key Features

The key theme of PolyScope 6.0 is **ROBUSTNESS**.

We have emphasized addressing key customer pains related to conflicts and scalability issues between URCaps. Therefore we are introducing a set of modern, scalable technologies to support URCaps becoming even richer, and more focused on creating value through solutions, while reducing their chance of conflicts towards other URCaps and providing more freedom of choice and powerful interfaces for our development partners.

PolyScope 6.0 boasts a major software platform upgrade, including a new Operating System (OS) that improve cybersecurity and provide increased options for URCap developers.

NOTE: PolyScope 6 is a breaking change for existing URCaps, and introduce a new URCapX-format that must be used for all URCaps. All existing URCaps must be adapted to install on PolyScope 6. See compatibility notices below.

- **Updated OS, 64-bits, and new Java LTS**
The OS is upgraded to 64-bit Debian 11, Java and thus the URCap frontend will follow the latest LTS version (Java 17). Some libraries have been removed, for example python. These libraries should now be used via the appropriate container base image
- **New URCap file format: URCapX**
URCaps will be packaged in a new, extensible format: URCapX. This zip-archive format will contain existing Java OSGi-bundles (jars, like in PolyScope 5), as well as docker-images for backend services. The format allows extending what URCaps can embed in the future.
- **URCap backend services (daemons) are run in Docker containers**
This reduce URCap reliance on the OS, avoid conflicts between multiple URCaps, and gives freedom of technology choice for developers.
- **New URCap development environment and SDK packaging**
To ease URCap development, the URCap SDK is now packaged as a development-container. This makes it easier to setup your development environment across Windows, Mac, or Linux, and work in your favorite IDE against the Dockerized development-environment. The URSim is also packaged as a Docker image, making it easier to simulate PolyScope from your local machine.
- **Struct datatype in URScript**
Set of variables can be aggregated into structs, and thus transferred and stored as a single variable
- **ROS2 state publishers**
Realtime data (joint positions, I/O, robot state, program state, etc.) is available through ROS2 topics.

- **ROS2 publishers and subscribers in urscript**
Publishing or subscribing to a ROS2 topic can now be done from urscript
- **Kinematic Tree in urscript**
Added coordinate frames as an object to urscript. They can be named, and referred to by existing motion commands (move!, movep, etc). They can also be attached/detached during a running program, and assigned inertia properties.
- **Disable/enable high holding torque**
Prevent false positive protective stop when the robot is standing still but the end-effector is doing some work such as screwing. However, if there is an external axis, this prevents collision detection if only the external axis is moving but the robot is in collision.
- **Improved Modbus Client script functions**
Added support for more Function Codes allowing for broader device support and increased communication efficiency. Added error handling script functions - now user can detect, and react to communication errors without stopping the program.
- **Improved Modbus Server**
General Purpose Coils added (bit registers)
- **Part Positioner - External Axis support in urscript**
UR Script adds support for external axis control through Axis Groups, Definitions, and extended motion script functions. This includes kinematic tree definition, frames, and real time control of coordinated motion with external axis via EtherCAT.

Known Issues

- URcap meta-data, such as the URcap name, URcap vendor, etc., might be missing or not properly displayed in some screens and dialogs in PolyScope
- The presentation of some URcap installation errors in PolyScope does not have the level of details and maturity of the previous PolyScope version

Fixed Bugs

- Fixed "No Controller issue when a non-existing control command is run" there were nullptr dereferences in case of not correctly formed commands. nullptr guards were added.
- Modulo 0 of an integer will be displayed as a float NaN
- Compare operations using bool_val(s) in URScripts now behave similar to other programming languages, like in Python
- Corrected a bug which was preventing correct transfer of an XMLRPC struct into the ValueStack
- Fixed a bug around anonymous struct to xmlrpc value conversion. If the struct_val is an anonymous struct (has no named members) it will be converted to a pose like xmlrpc struct (if member count is 6) or to an xmlrpc array.

Change log

INU	3/28/2023