H-ROS
Hardware Robot Operating System

D-module

STUDY CASE

Han's Robot D-module

Han's Robot

Han's Robot is a global supplier in direct drive technology, ranging from linear motors, torque motors, servo drives, robots to customized motors and other industrial automation solutions. Founded in Shenzhen, China in 2005, Han's Robot has rapidly grown to the leading company in its domestic market, fully equipped with over 400 dedicated people in R&D, production, supply chain, quality, sales, and support functions. The firm combines its technology strengths and domain expertise to provide a vast range of customer-focused solutions and tens of thousands of direct drive motors are successfully applied in a wide spectrum of industries every year. Today, Han's Robot successfully stands as one of the leading suppliers of innovative direct-drive products and solutions for industry customers.

Han's Robot D-module

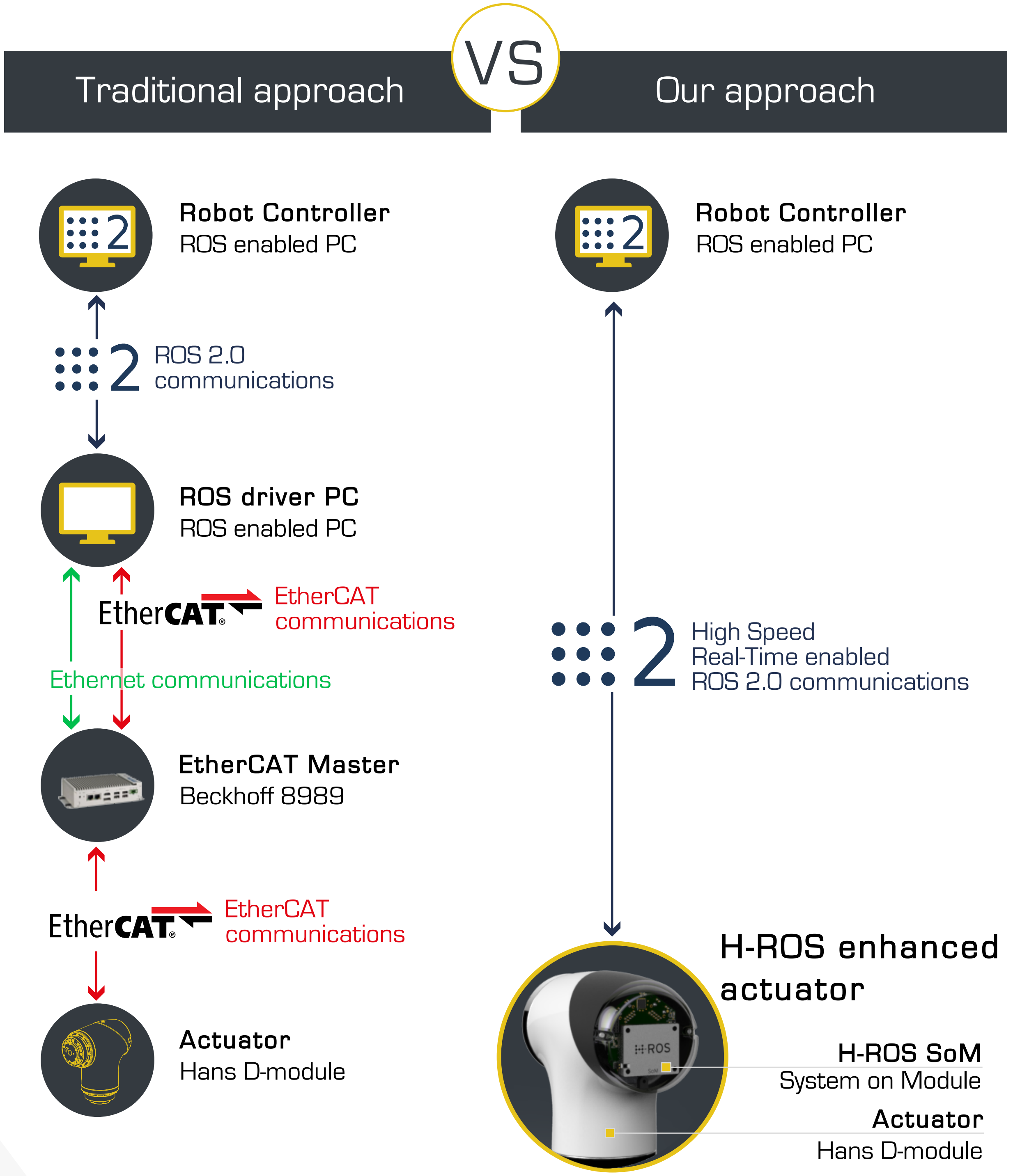
The Han's D-motor actuator is an industrial-grade 2 DoF electrical motor that includes encoders and an electro-mechanical breaking system. The device is mostly used on industrial applications and is available on a variety of different torque and size combinations. The actuator operates as an EtherCAT slave and can be controlled through this bus.

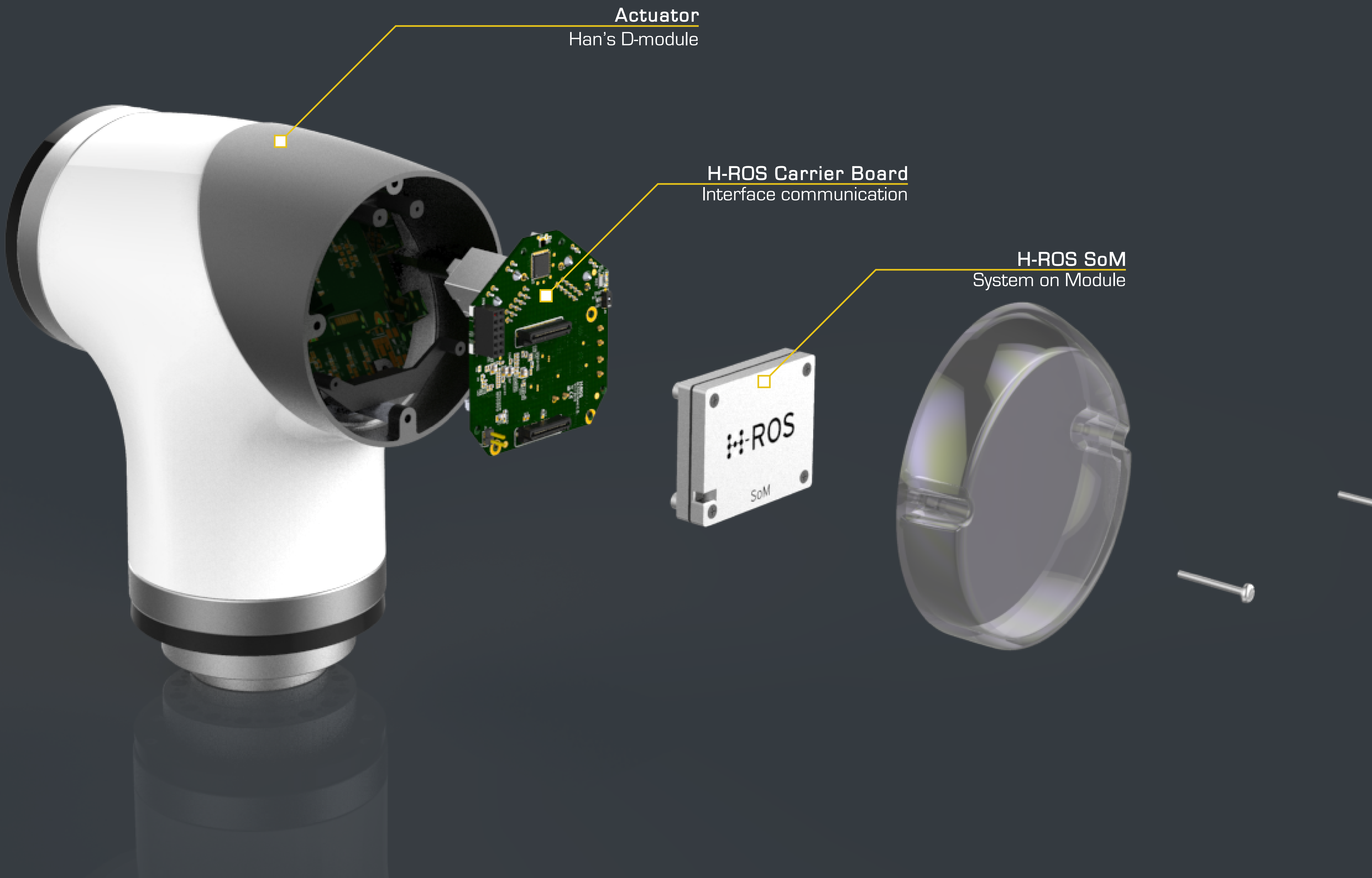
	14 series	17 series	20 series	25 series	32 series
rated torque	9.4Nm	30Nm	49Nm	85Nm	156Nm
peak torque	34Nm	69Nm	104Nm	200Nm	420Nm
maximum rotation speed	90°/s	90°/s	90°/s	90°/s	90°/s
weight	2.8kg	3.8kg	5.8kg	9.6kg	17kg
rotation angle	±360°	±360°	±360°	±360°	±360°
repeatability	20arcsec	20arcsec	20arcsec	20arcsec	20arcsec
communication	EtherCAT	EtherCAT	EtherCAT	EtherCAT	EtherCAT
supply voltage	DC 48V	DC 48V	DC 48V	DC 48V	DC 48V
IP class	IP54	IP54	IP54	IP54	IP54
main material	aluminum alloy	aluminum alloy	aluminum alloy	aluminum alloy	aluminum alloy
working temperature	0-50℃	0-50℃	0-50℃	0-50℃	0-50℃
working humidity	10-80%	10-80%	10-80%	10-80%	10-80%

H-ROS enhanced module

Traditional approach with ROS
For most robotics applications, roboticists will tend to use the Robot Operating System (ROS). Provided such interest, a typical configuration for its use would be one where the actuator is interfaced with an EtherCAT master that is later connected to a ROS-enabled PC that contains the appropriate ROS driver. From such "ROS driver PC", interactions from the ROS network can happen as usual.

H-ROS
Our approach involves integrating the Hardware Robot Operating System (H-ROS) through the System on Module (SoM). This facilitates the whole process of integration and delivers a ROS native hardware actuator that can be used from the ROS network seamlessly through a High Speed and Real-Time enabled communication interface. Moreover, the inclusion of H-ROS provides security protections and enhances the Han's D-module with additional capabilities such as inertial sensing information, hardware power lifecycle, etc.




Actuator
Han's D-module
H-ROS Carrier Board
Interface communication
H-ROS SoM
System on Module
H-ROS
SoM

Real-Time analysis

We compare the traditional ROS approach with H-ROS (our approach) taking into account the following aspects:

1. Synchronization

The lack of synchronization has several implications for real-time robotic systems such as a lack of coordination among software and hardware components, added latencies due to stochastic arrival time offsets or inaccurate data reconstruction or inference. In addition, communication latency cannot be measured because of clock misalignment. An alternative approach to measure communication latencies in unsynchronized environments was demonstrated by Gutiérrez et al. and consists of using the so called ping-pong setup. This method, while popular, does not allow to measure the communication performance of "real systems" and only simulated scenarios can be tackled.

2. Operating System

The operating system is critical for real-time robotic control. Having a deterministic behavior through time and through different conditions allows robots to respond appropriately.

3. Networked environments with traffic

Most robot components interact in a networked environment together with other robot components. Real-time systems require their communication latencies to be contained even in the worst scenarios. We analyze what happens when we attach the D-module in a saturated robot network.

[1] Gutiérrez, C. S. V., Juan, L. U. S., Ugarte, I. Z., & Vilches, V. M. (2018). Real-time Linux communications, an evaluation of the Linux communication stack for real-time robotic applications. arXiv preprint arXiv:1808.10821.

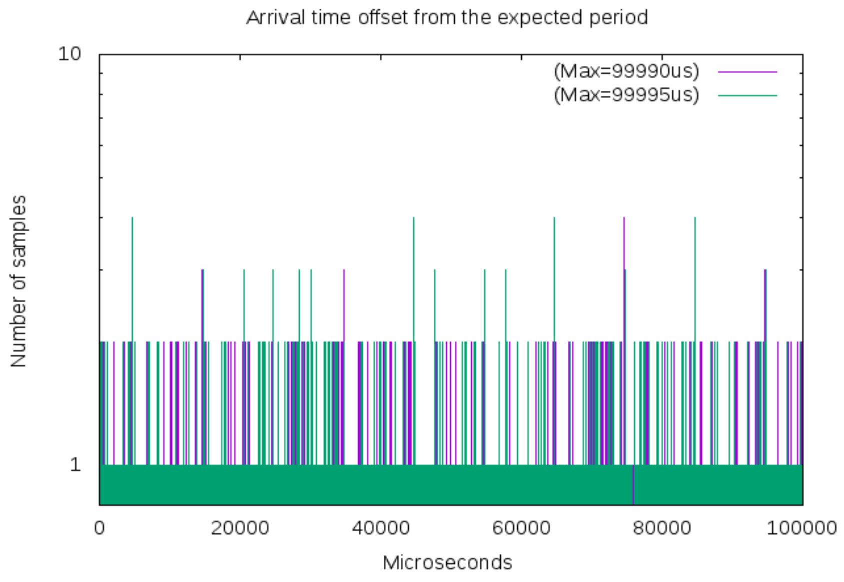
1. Synchronization

VS

Traditional approach: no synchronization

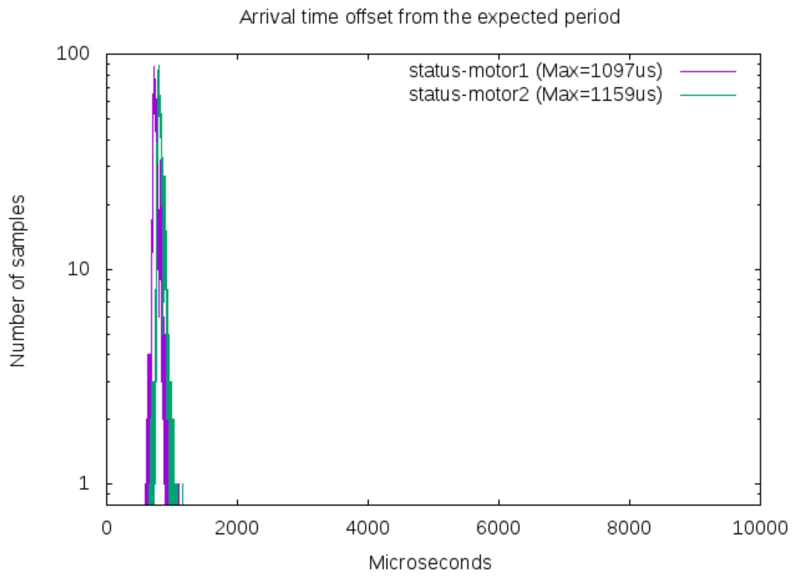
In the traditional ROS approach, most commonly, none of the devices have been synchronized appropriately. Given this lack of synchronization, the traditional ROS setup does not allow to measure communication delays. However, it is possible to measure when messages arrive to the robot controller (from where we operate the actuation module).

ROS is not designed to provide synchronization by default. For the D-module, when using the default ROS timers and a 100 millisecond cycle, the arrival time offset (from the expected period) can be as high as the complete window, 100 milliseconds (ms) or 0.1 seconds. Synchronization delays will be added to the overall system delay. These results are thereby unacceptable for most real-time use cases because it will mean that any communication latency will receive offsets up to 0.1 second (s) stochastically.



Our approach: H-ROS

Our approach includes a full synchronization scheme that allows H-ROS powered components to have arrival time offsets below 1 millisecond (ms), even in the most extreme circumstances. This means our solution provides synchronization between ROS communications bellow 1 millisecond end-to-end.

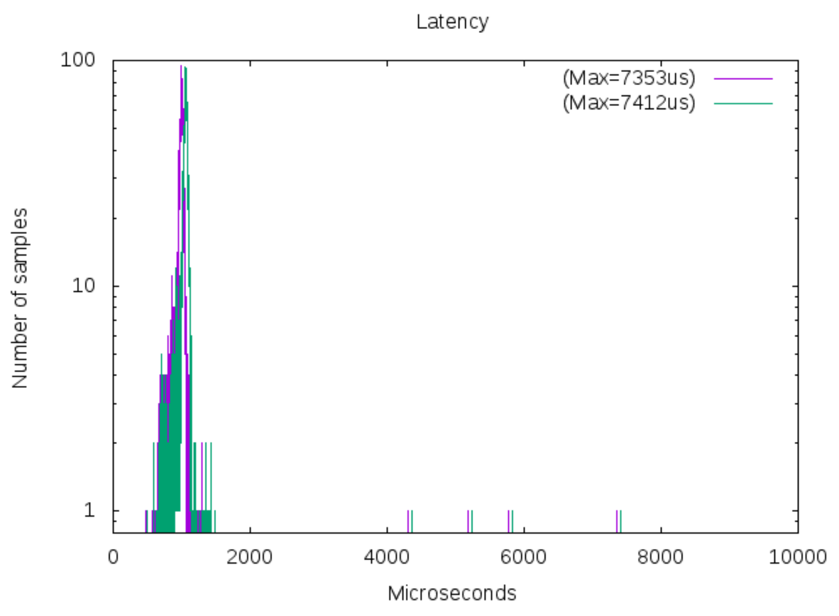


2. Operating System

VS

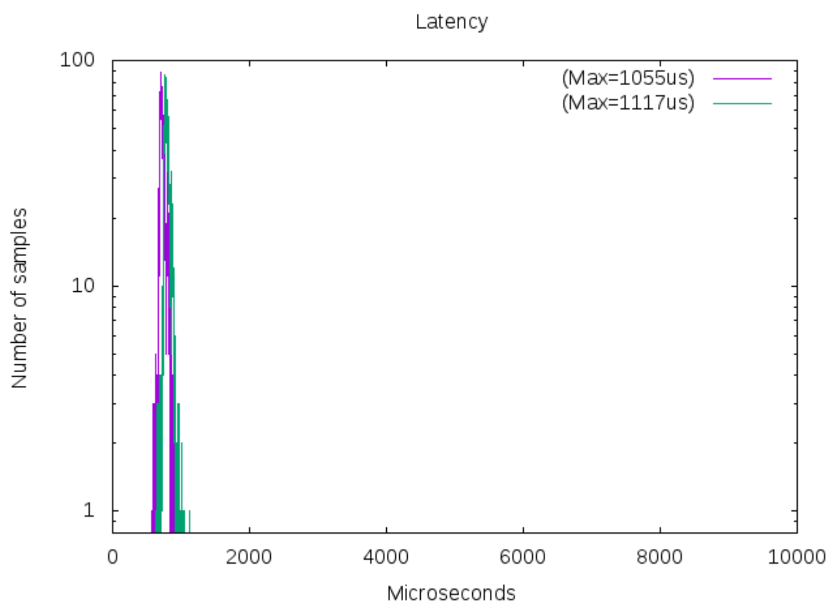
Traditional approach: Ubuntu 16.04

Assuming there is synchronization between robot components, and using a popular fresh Ubuntu 16.04 installation, we obtain communication latencies with peaks that go beyond 7 milliseconds (ms).



Our approach: H-ROS

The H-ROS SoM deploys an optimized Linux-based Real-Time Operating System in a hybrid architecture that has been particularly customized for the D-module needs. With our solution, the communication latencies of the modified D-module stay below 2 milliseconds (ms).

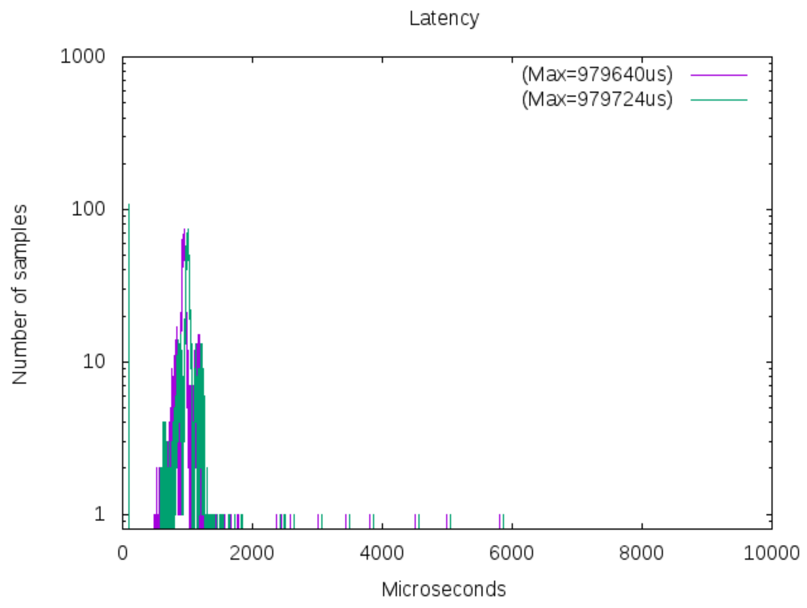


3. Networked environments with traffic

VS

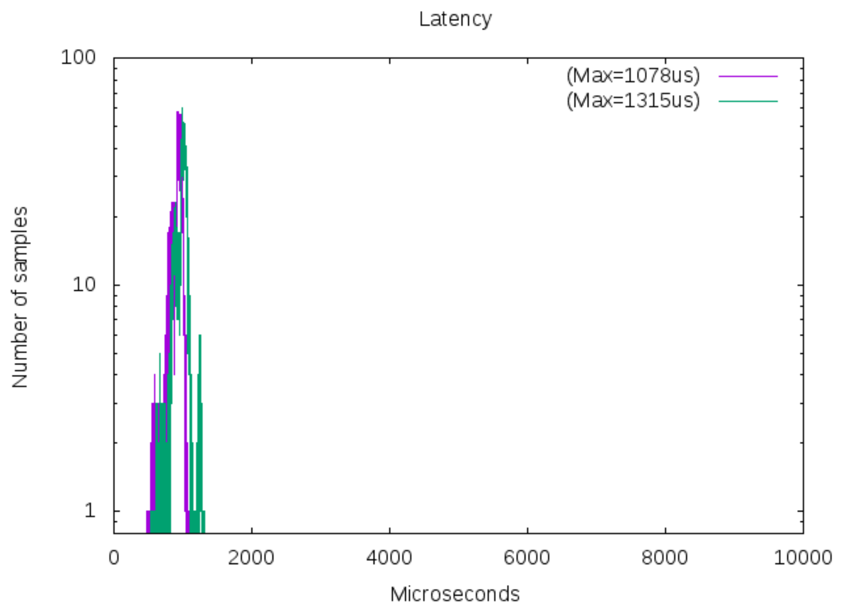
Traditional ROS approach

Using the traditional ROS approach in a saturated environment, common in a variety of applications ranging from research to commercial and professional robots, we obtain latencies that go up to 1 second (s). This means that control systems wouldn't be able to establish control loops above 1 time per second (1 Hz). Moreover, this scenario is not bounded and additional levels of saturation in the network (e.g. by adding a depth sensor) would result into latencies above 1 second (s).



Our approach: H-ROS

The H-ROS SoM technology provides capabilities to be reliable in the worst circumstances. It is able to remain stable and provide with communication latencies below 2 milliseconds (2 ms) when faced with traffic bursts of 900 Mbps (or 90% of the channel capacity).



Key benefits

Thanks to the H-ROS SoM, the augmented advantages of the D-module are:

- **ROS 2.0 Hardware**
  A purely distributed robot module built with ROS 2.0. A first class-participant of the ROS 2.0 ecosystem.

- **Security**
  An encrypted computing and communication environment. A hacker-tested robot module, secured through continuous security audits.

- **Diagnostics and telemetry**
  First hand data about robot part state through ROS topics, power consumption, processor load and much, much more.

- **Full synchronization**
  Distributed sub-microsecond clock synchronization accuracy. ROS 2.0 latencies get optimized.

- **Network resilient**
  Applications that operate predictably in the presence of network congestion. Even with traffic bursts above 90% of the channel capacity, our solution delivers.

- **Policing**
  Robot modules that meet their specifications by applying individual policy rules. This implies that even if your robot component malfunctions, we ensure that it does not compromise the rest of the robot network.

- **Hardware level life-cycle**
  ROS 2.0 life-cycle extension to hardware which allows to influence power conditions for an increased performance or adaptive behavior.

- **Interoperability**
  A common interface, the Hardware Robot Information Model (HRIM), that enables communication among different vendors, regardless of the manufacturer. It also enables robot modules to be added or removed to the network without interfering with the runtime operation of any other device on the network and without impacting any data flows in which they are not directly engaged.

- **Automatic updates, over the air (OTA)**
  Over-The-Air updates for robot parts. The SoM keeps robots and robot modules updated, seamlessly.

- **Real-Time Operating System**
  Deterministic operating system responses powered by a hybrid architecture featuring the most popular OS in the robotics domain: Linux.

- **Traffic shaping**
  A mechanism that allows to reserve bandwidth for high-priority traffic while, at the same time, ensuring that best effort traffic will continue to flow.

- **Bandwidth allocation**
  The option to dynamically estimate the available bandwidth in the network and to reserve an additional portion, if possible. This empowers roboticists with the status of the robot network.

H-ROS added value

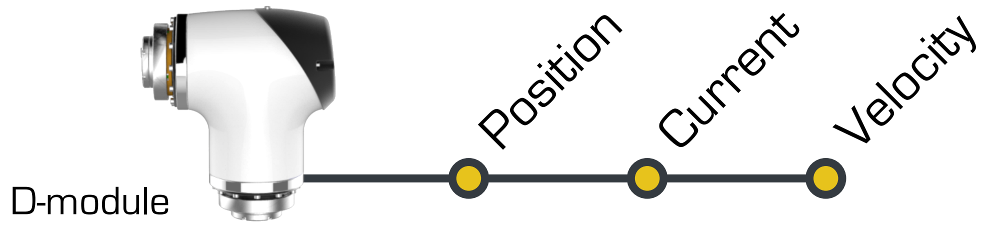
Redundancy
A variety of different network architectures that enhance the redundancy of a robot network (daisy-chain, line, ring, tree, etc.) gaining reliability.

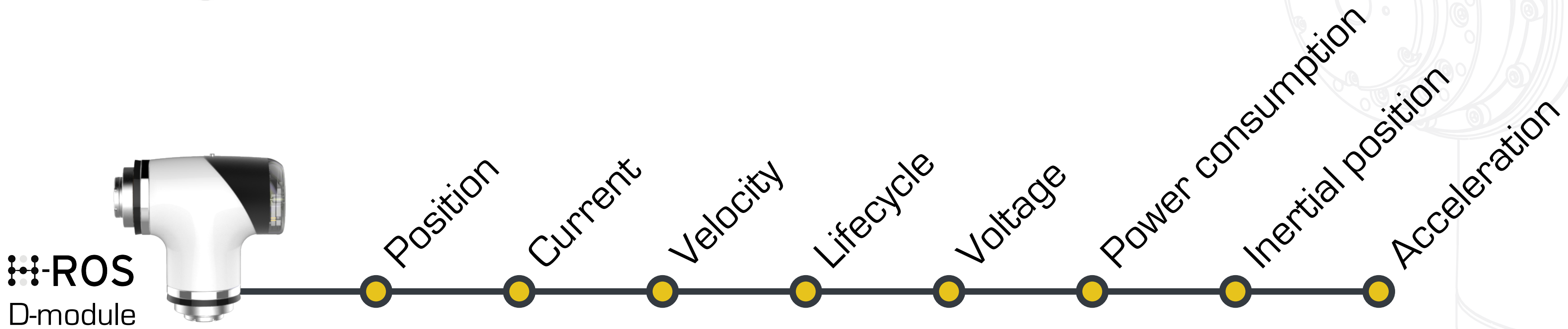
Scalability
The possibility to grow from individual modules to large robots filled with hundreds of them in multi-network setups.

Backwards compatibility
The SoM includes mechanisms to integrate any existing IEEE 802.3 (Ethernet) standard protocol providing backward compatibility.

Low cycle times
Scalable and user-selectable cyclic update rates that can meet or exceed legacy Industrial Ethernet solutions.

Converged networks
Coexistence with best effort traffic and support to multiple industrial protocols.





Acutronic
robotics

www.acutronicrobotics.com

SIMPLIFYING ROBOTICS