Real-Time Extension to the Robot Operating System

Jan Carstensen
Axel Rauschenberger

This work has been supported by the Federal Office of Bundeswehr Equipment, Information Technology and In-Service Support (BAAINBw).
RTROS – Goals and Architecture

Design Goals

- Usability
- Hard Real-Time
- Safety

Architecture

- ROS Node
- RTROS Node
- RTROS
- ach
- Ubuntu
- Xenomai

RTROS – Real-Time Extension to the Robot Operating System
Overview

- Publish/Subscribe
- Services
- Time
- Transform (TF)
- Tools

Version 1.0

RTROS – Real-Time Extension to the Robot Operating System
Rate Benchmark

- **ROS**
  - Average jitter: 2.740.15 μs
  - Maximum jitter: 21.846.05 μs

- **RTROS**
  - Average jitter: 0.66 μs
  - Maximum jitter: 114.89 μs
Communication

Publish/Subscribe

- Publisher
  - Main Thread
  - publish(\text{\ldots})
  - serialize
  - put
  - shared memory

- Subscriber
  - Main Thread
  - poll
  - deserialize
  - callback(\text{\ldots})
  - get

Services

- ServiceServer<\text{\ldots}>
- ServiceConnectionRequest
- ServiceClient<\text{Time}>
- ServiceServer<\text{Time}>
- ServiceRequest<\text{Time}>
- ServiceConnectionResponse
- ServiceResponse<\text{Time}>
Publish/Subscribe Benchmark

- ROS:
  - 210.24 μs (avg)
  - 27.701.19 μs (max)
  - 99.843/100.000 (99%)

- RTROS:
  - 45.96 μs (avg)
  - 179.93 μs (max)
  - 100.000/100.000 (100%)
Services Benchmark

- ROS
  - 26,317,33 μs (avg)
  - 47,986,3 μs (max)
  - 100.00/100.000 (100%)

- RTROS
  - 57,35 μs (avg)
  - 187,75 μs (max)
  - 100.000/100.000 (100%)

04.11.2016

RTROS – Real-Time Extension to the Robot Operating System
Current Work

- Testing and Improving
- RTROS – Compatibility
  - Preempt_RT
  - Xenomai 3.0
- External Evaluation
  - Open-Source Release
- New Features
  - Network Communication
  - Time Synchronisation