UNIVERSITY OF TWENTE.

Connecting Two Robot-Software Communication Architectures: ROS and LUNA

Communicating Process Architectures 2016, Copenhagen, DK

W. Mathijs van der Werff, **Jan F. Broenink**

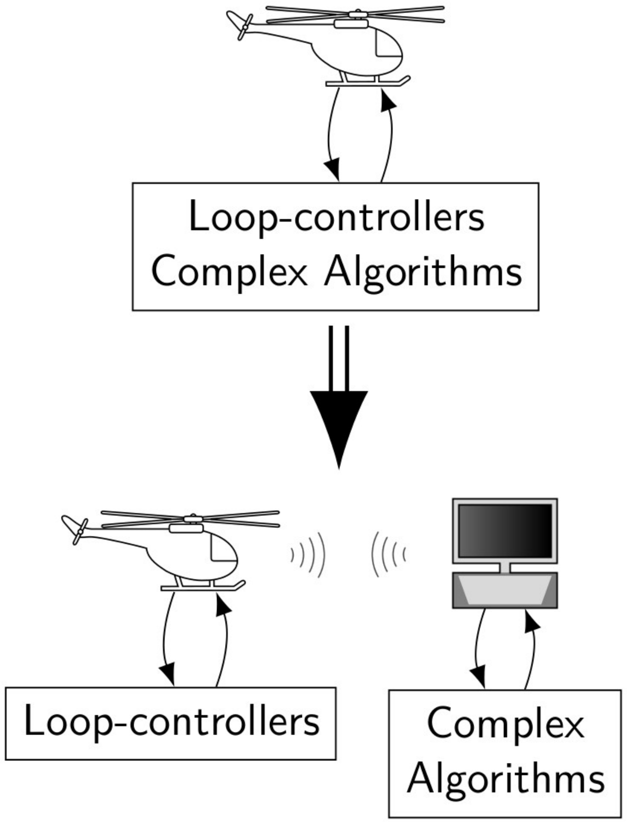
University of Twente
CTIT institute, Robotics & Mechatronics
Enschede, Netherlands

RAM ROBOTICS AND MECHATRONICS

CTIT

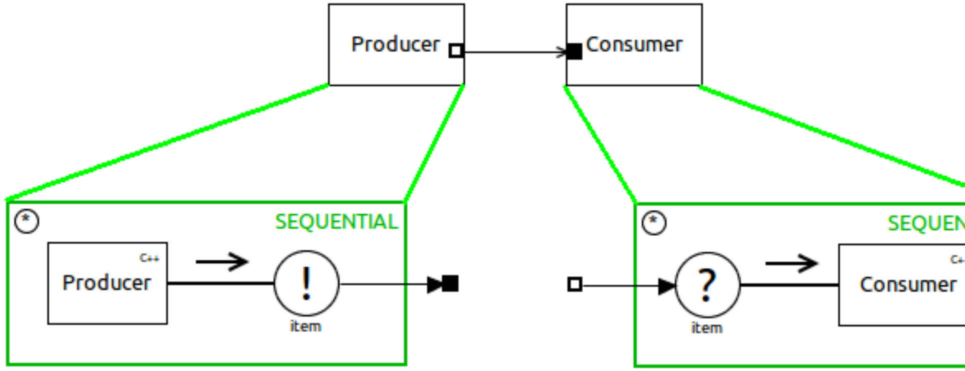
Introduction — Motivation

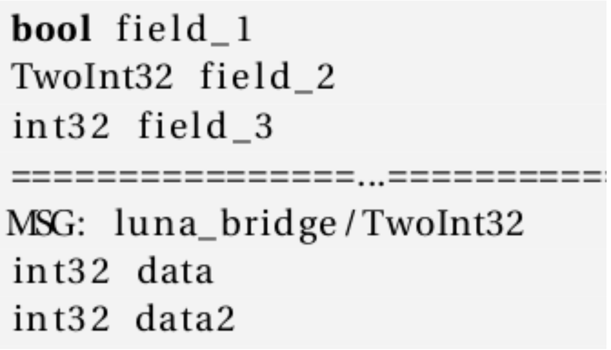
- Two trends in robotics — Conflicting!
  - More complex algorithms
    - Computer vision, area mapping, planning
  - More light weight, energy efficiency
    - Mobile robots, unmanned aerial vehicles (drones)
- Possible Solution
  - Offloading algorithms to base station
    - Development of algorithms easier
    - More resources, like computer power
    - Easier upgradable
  - Connection between two environments needed
    - Algorithms
      - Robotic Operating System – ROS
    - Loop Controllers, i.e. hard-real time code
      - LUNA Universal Network Architecture -- LUNA



Introduction — Some Background

- Hard real time
  - Controlling robots, i.e. fast mechanics
- LUNA run-time framework
  - Hard real-time execution, precompiled
  - Design Flow
    - Graphically designed CSP processes in TERRA, and verified
    - Code generated, linked to LUNA lib
- ROS – Robot Operating System
  - Open source / large community
  - Publisher - Subscriber pattern: nodes and messages
  - Design Flow
    - Design algorithms and message types
    - Connect nodes via message exchange
    - (re) compile

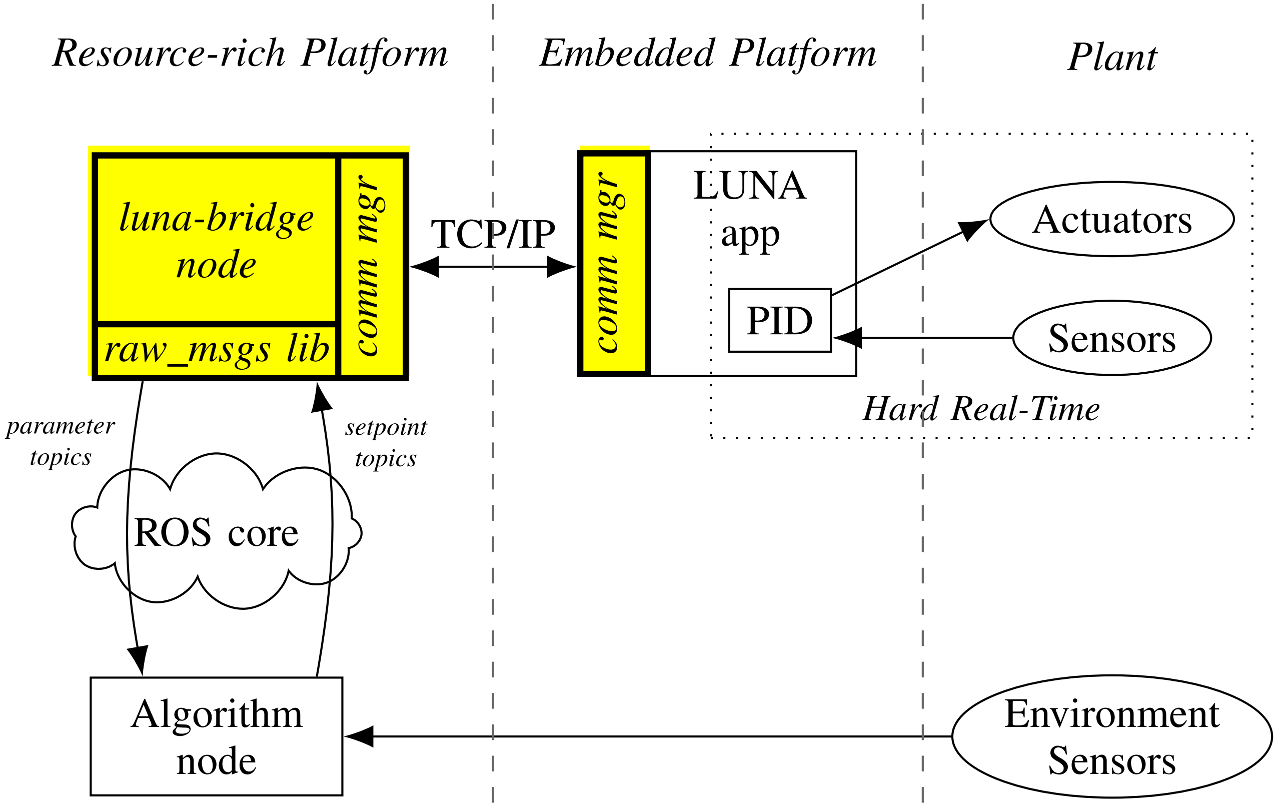




```
bool field_1
TwoInt32 field_2
int32 field_3
================...==========
MSG: luna_bridge/TwoInt32
int32 data
int32 data2
```

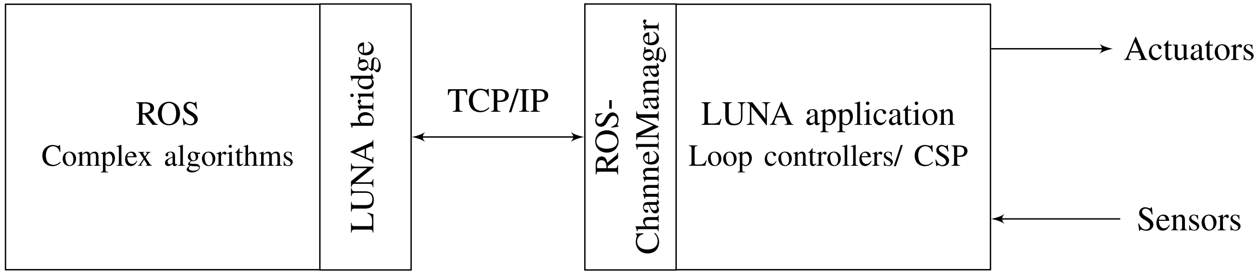
Introduction — Prototype, earlier made

- Prototype ROS-LUNA bridge made
  - Algorithms in ROS and hard real-time controllers in LUNA
  - Problem: ros :: Publisher pub = n. advertise <template T>(”topic”, 10);
    - so source-code level in ROS to be connected to precompiled library in LUNA
  - Bezemer et al. at ETFA 2015
- Prototype
  - Based on ShapeShifter class
  - Integer LUNA → ROS
  - Limited support messagetypes
    - only basic datatypes



Design and Implementation

- Essential Requirements
  - Versatile / Reusable
  - Compiled program
  - SRT - HRT connection
    - Asynchronous data connection



- Overview
  - Communication
  - LUNA
  - ROS

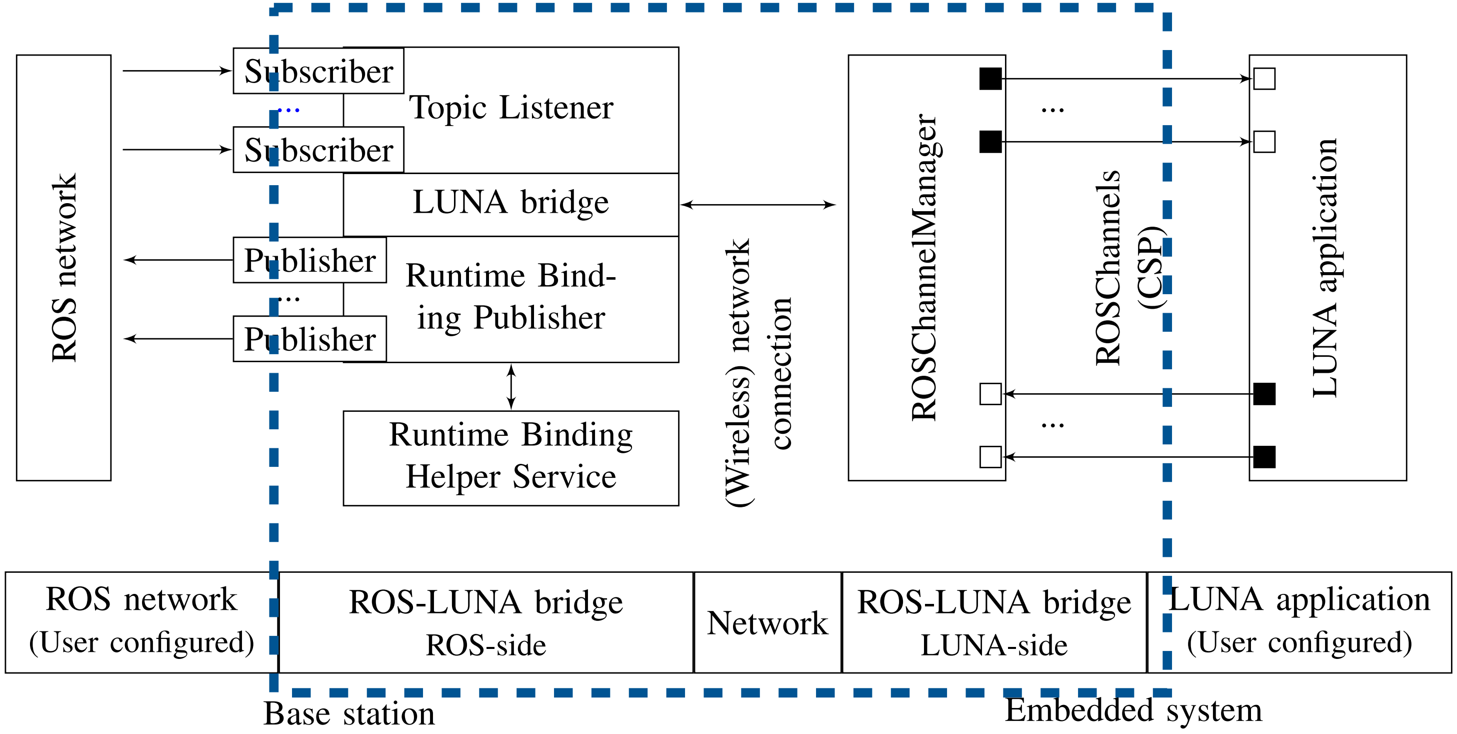
ROS network (User configured)	ROS-LUNA bridge	LUNA application (User configured)	Robotic setup

Base station

Embedded system

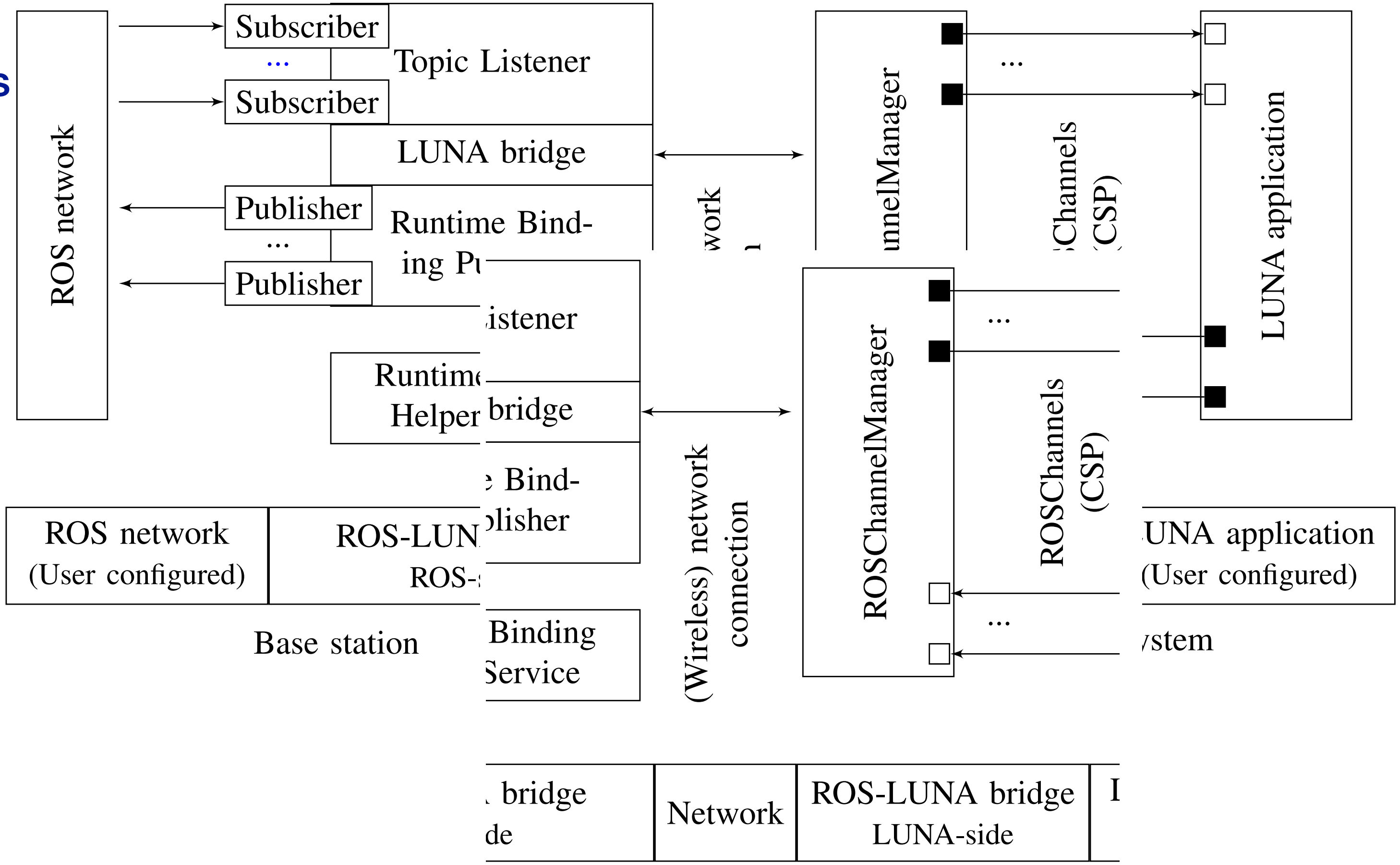
ROS-LUNA Bridge Architecture

- Overview
  - Communication
  - LUNA
  - ROS



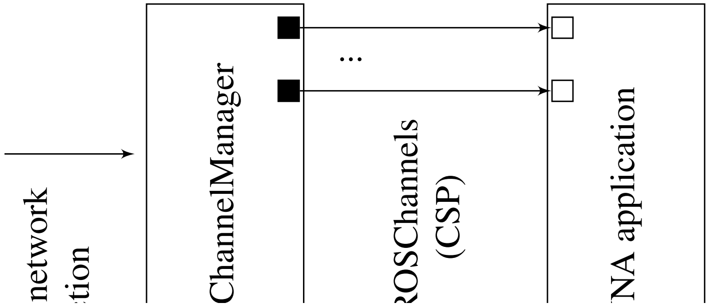
Implementation — Communication

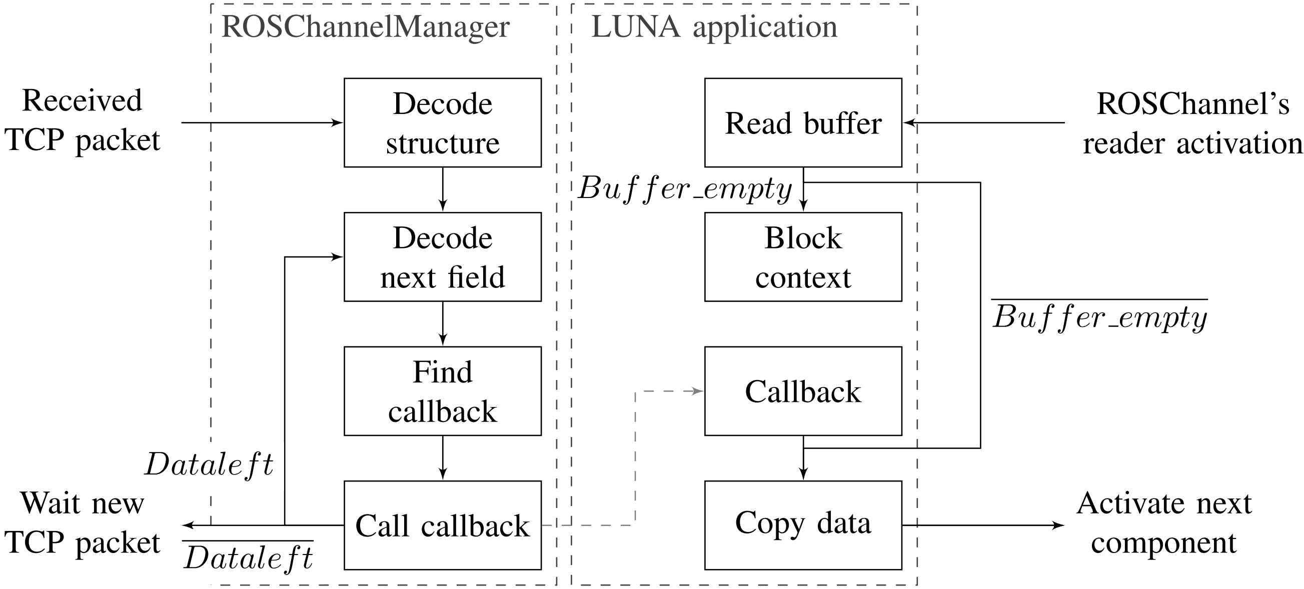
- Communication Protocol
  - Serialise, Deserialise
    - to fill up TCP/IP packets
      - use bandwidth effectively
    - tailored solution
      - reduce overhead
  - Extendible
  - ROS channels
    - >>

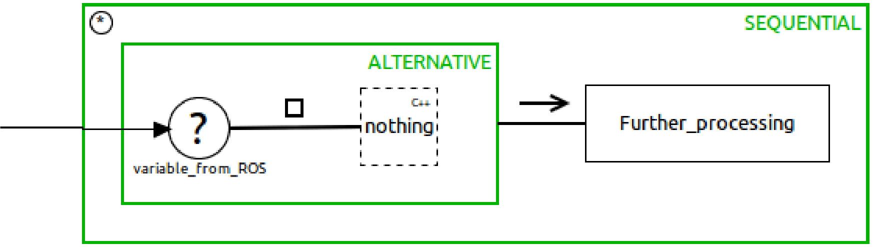


Implementation — specific channels in LUNA

- LUNA — ROS channels
- Allows modeling in TERRA
- Channel modifications
  - non-blocking write to ROS
    - from HRT to SRT
    - 2 data buffers
  - blocking read from ROS
    - synchronisation…
- Non-blocking read
  - using ALT: ROSread [] SKIP

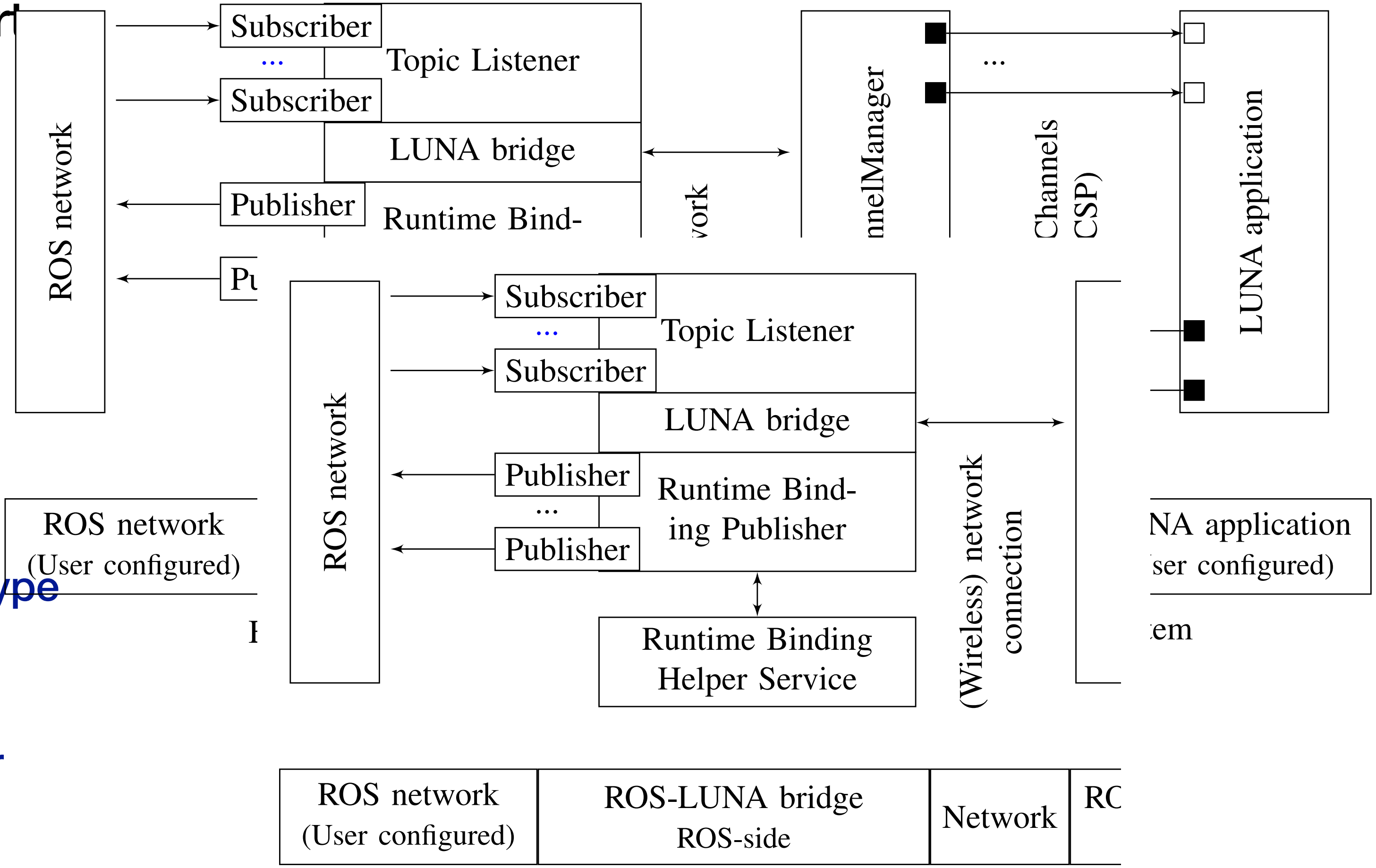






Implementation — ROS topic listeners…

- ROS — Topic Listeners
  - topic = data to transpor
  - run-time topic binding
    - specific Publisher
  - specific configuration
    - through the network
- Implementation
  - ShapeShifter class
    - publish & subscribe
    - without specifying data type
  - Needs specific
    - serialiser, deserialiser
    - RuntimeBindingPublisher
    - extended TopicListener

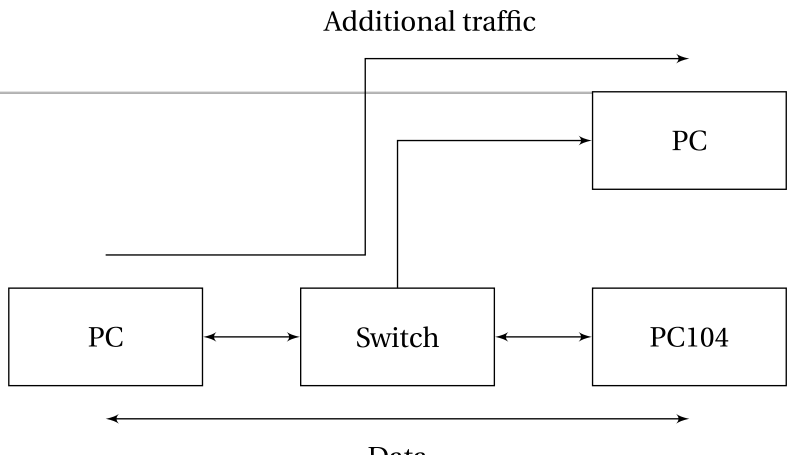


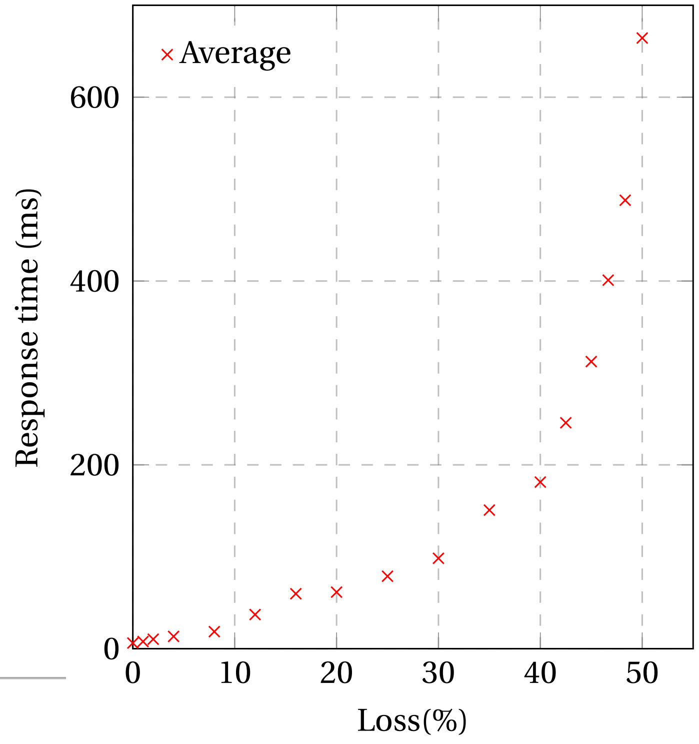
Testing

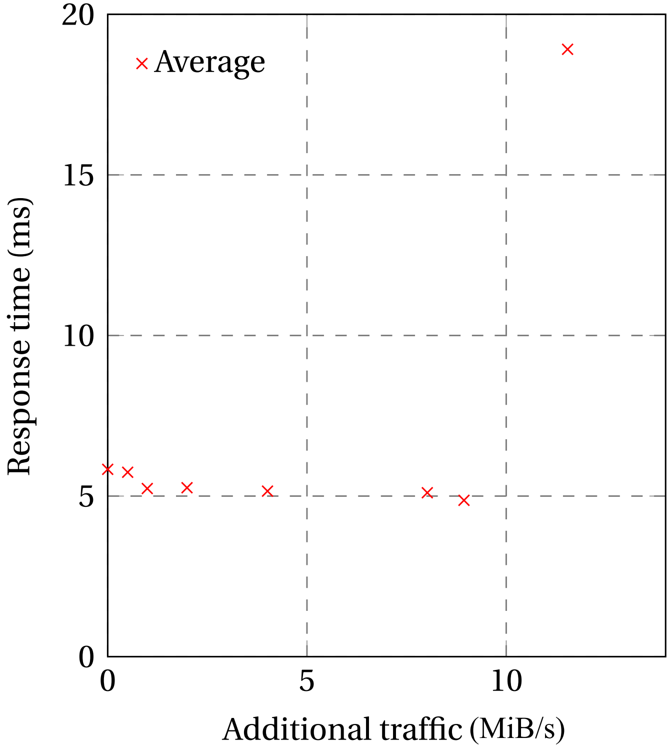
- Initial Tests
  - on bandwidth
  - packet loss
- Verification, Performance
  - RBP - RuntimeBindingPublisher
  - Performance
    - Publishers
    - Subscribers
- Demonstration
  - timing
  - robotic system

Initial Tests

- Packet loss
  - to mimic WiFi
- Additional traffic
  - network sharing

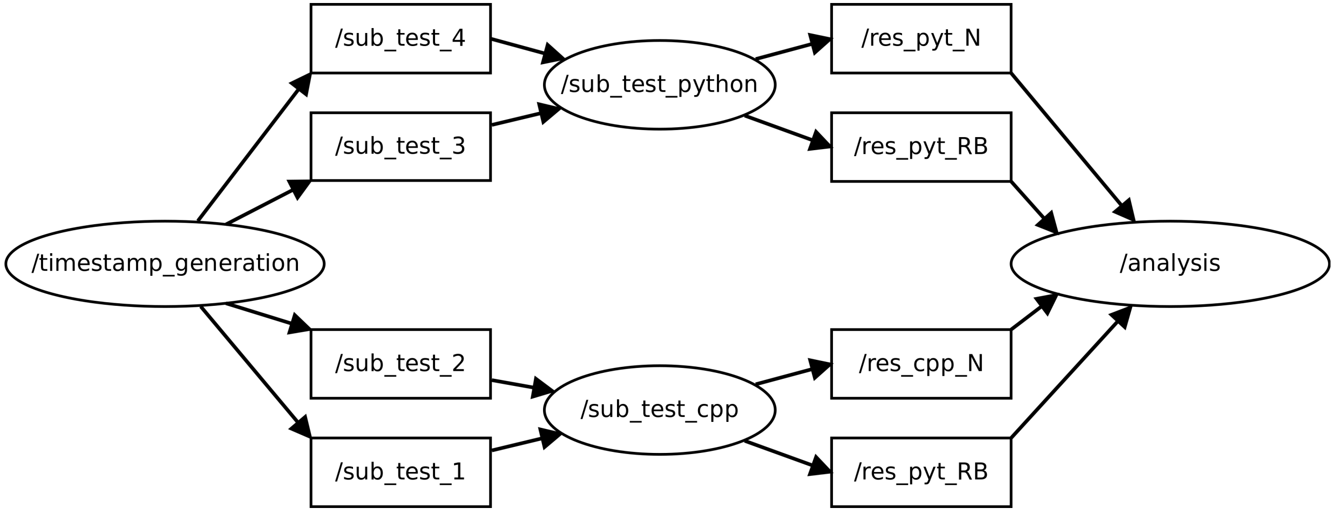






Verification tests

- Verify RuntimeBindingPublisher
  - correct serializing / deserializing
- auto-generated ROS structure of test
  - time stamp test:

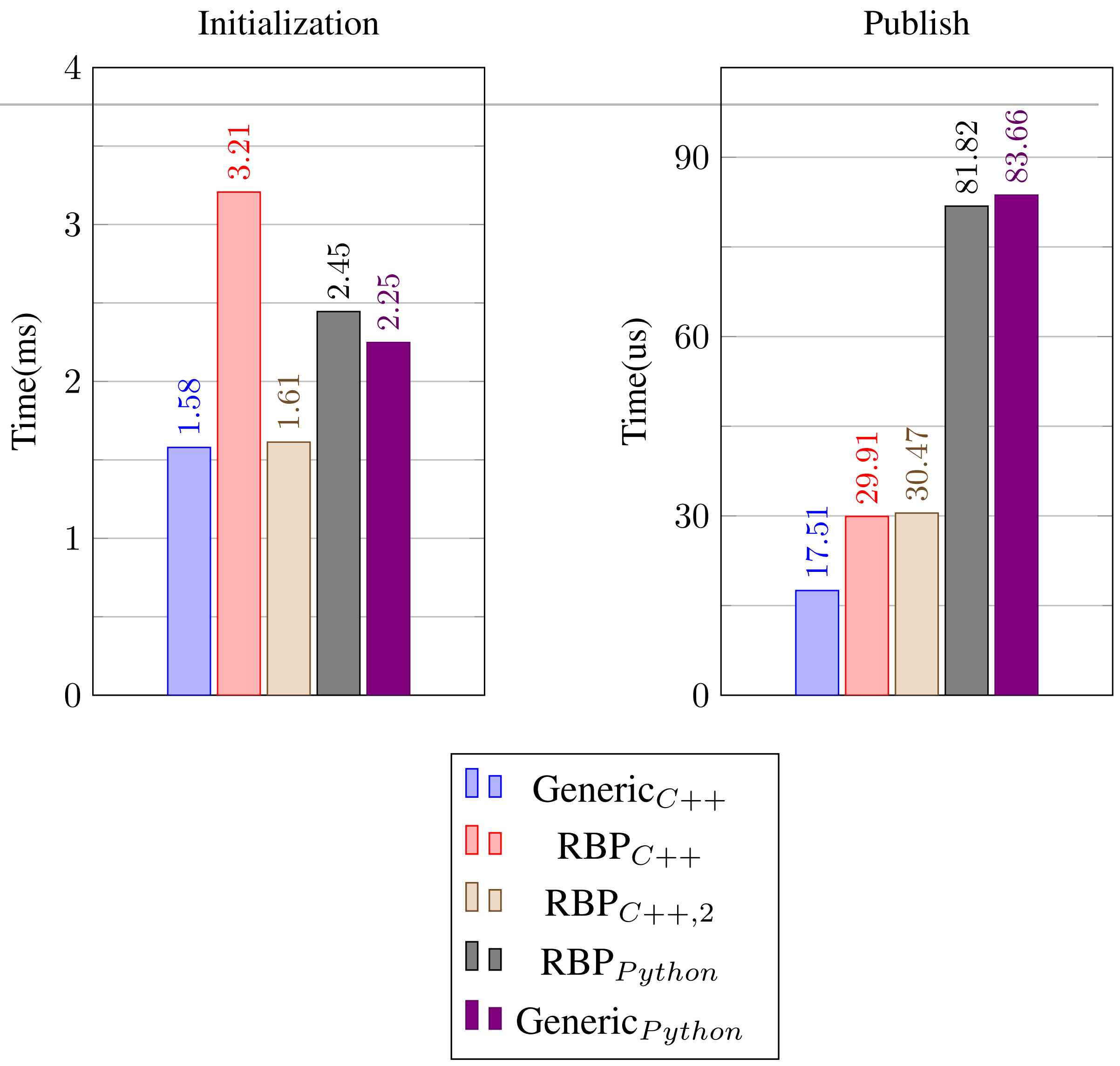


Performance Tests - Publishers

- Five different implementations of ROS publishers
  - generic ROS Publisher in C++
  - generic ROS Publisher in Python
  - RuntimeBindingPublisher with prior msg info
  - RuntimeBindingPublisher without prior msg info
  - simplified RuntimeBindingPublisher in Python
- Tests
  - average of 100 tests
  - per test 50 x init and publishing of 100 samples
  - 10 tests in 1 run
    - 100 tests in 1 run makes ROS core crash
  - On intel i5@2.53 GHz, 4 GB RAM, Ubuntu 15.10, ROS Jade

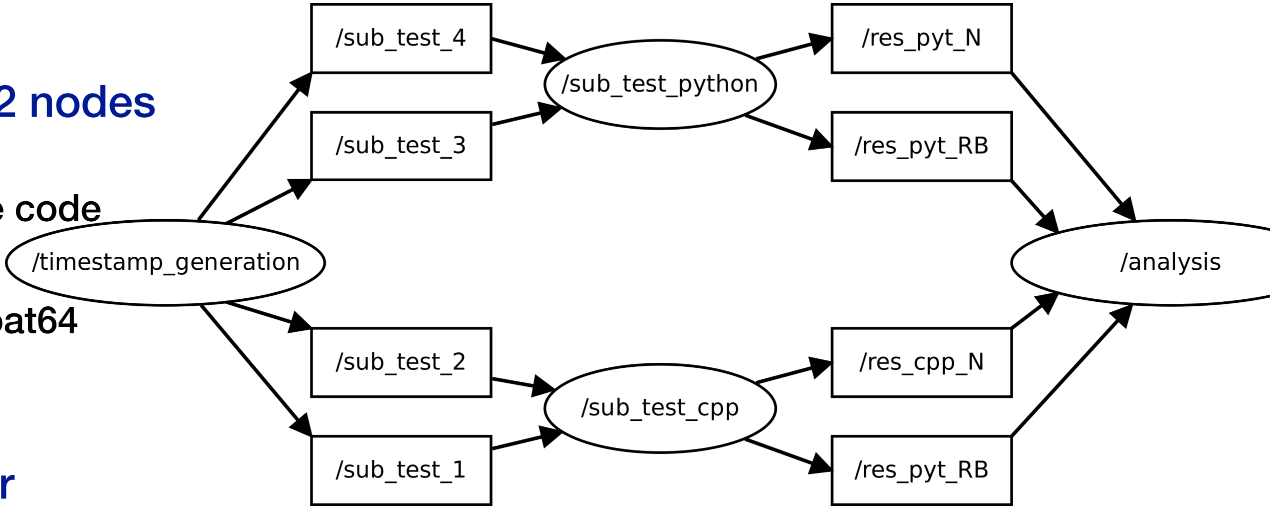
Publishers

- Initialisation
  - RBPc++ slowest
    - due to external Python helper node
    - RBPc++2:
      - not needed as used from previous call
  - Python slower than C++
    - RBPPython slower than Python
      - additional fu calls needed
- Runtime
  - RBPs are comparable
    - only initialisation is different
  - RBP slower than C++
    - due to additional var name look ups
  - Python slowest



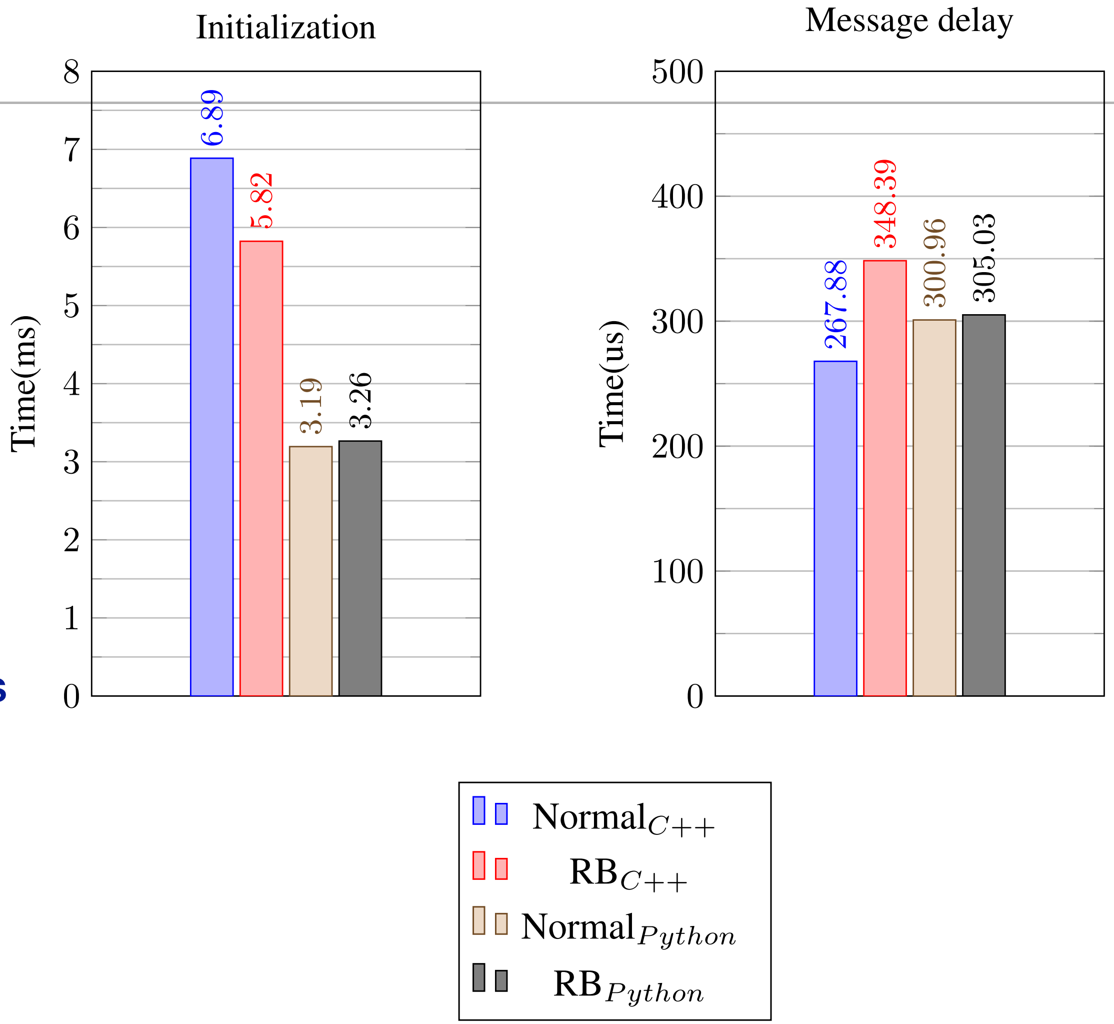
Performance Tests - Subscribers

- Four different implementations of ROS subscribers
  - normal subscribers in C++ / Python
  - extended TopicListener in C++ / simple runtime binding in Python
- Tests
  - custom type: header and 2 float64
  - average of 100 test, for initialisation
  - 6,000 msg @ 200 Hz:
    - time stamp send as float64
    - published over 4 topics, connected to 2 nodes
      - 1 node C++, 1 node Python
      - both have runtime binding and normal node code
    - received data
      - elapsed time is measured and put in 2nd float64
    - analysed
      - in analysis node
    - delay: publisher + network + subscriber
      - network delay can be subtracted as common factor

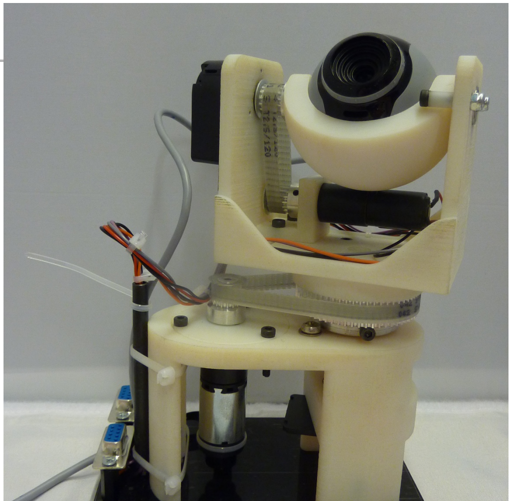


Subscribers

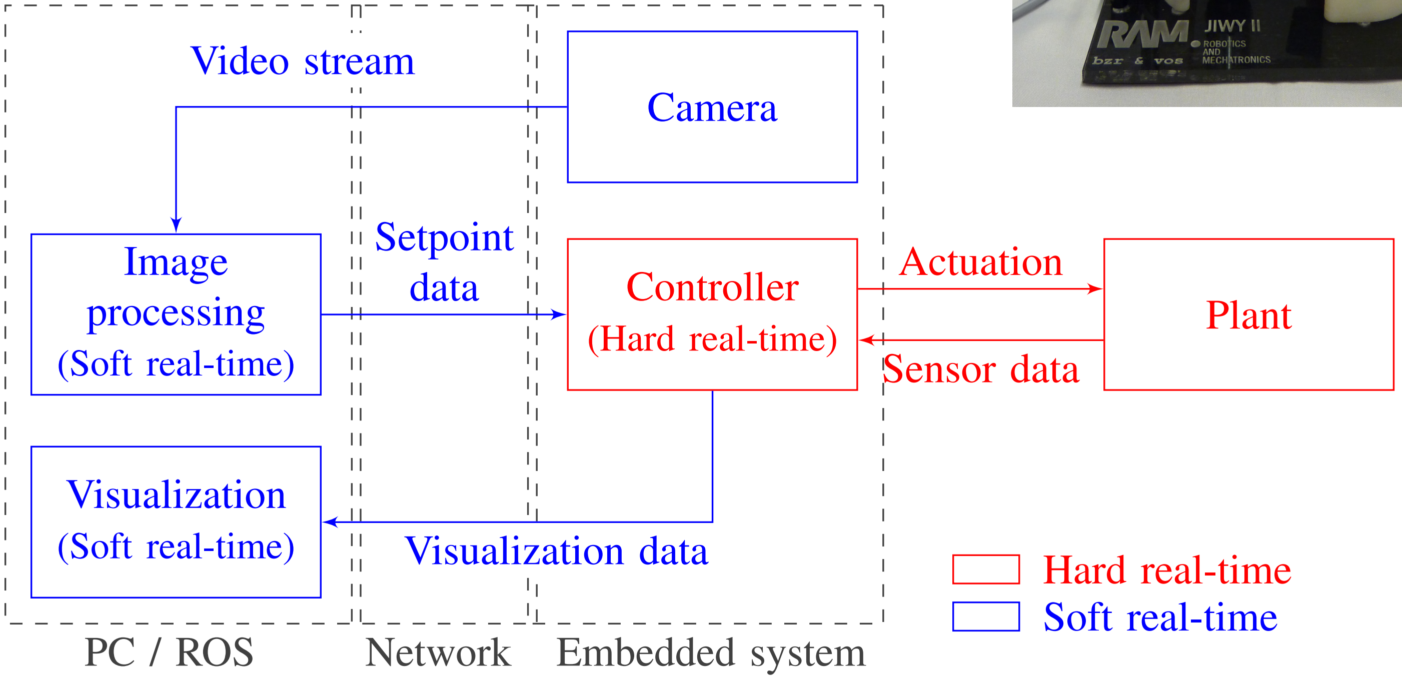
- Initialisation
  - C++ slowest
    - due to tasks others do at runtime
      - like registering the callback
  - Python seems to optimize
    - due to repeating of runs
- Runtime
  - C++ slowest
    - has to iterate over description fields
  - Python faster than RBPc++
    - due to optimizations
- Overall conclusion
  - C++ faster than Python
  - RBPc++ is in between



Demonstration Tests

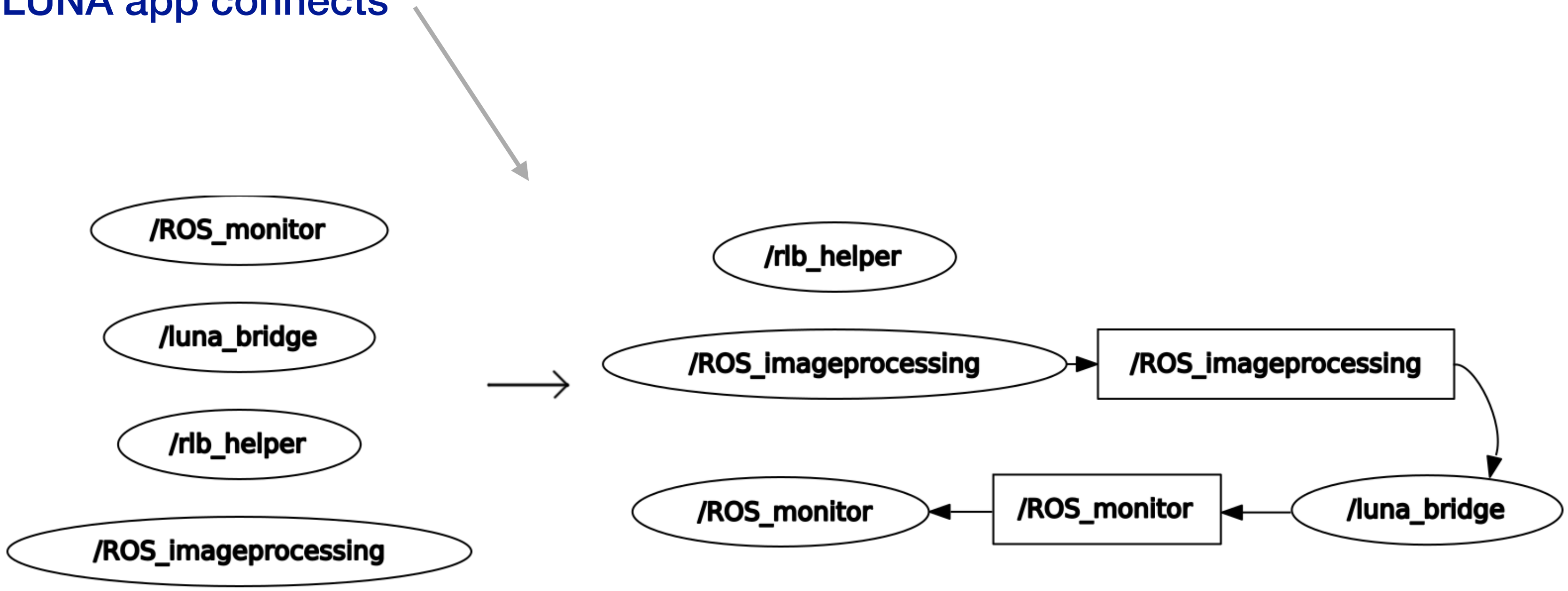


- Robotic setup: vison in the loop
  - our favorite JIWY test setup
    - pan-tilt gimball, DC-motor driven
  - RaMstix embedded board:
    - Gumstix over fire, Linux 3.2.21, Xenomai HRT patch 2.6.3
    - FPGA for PWM pulse generation and encoder pulse counting
  - Notebook for ROS
  - Tests
    - initialisation
    - timing
    - real action



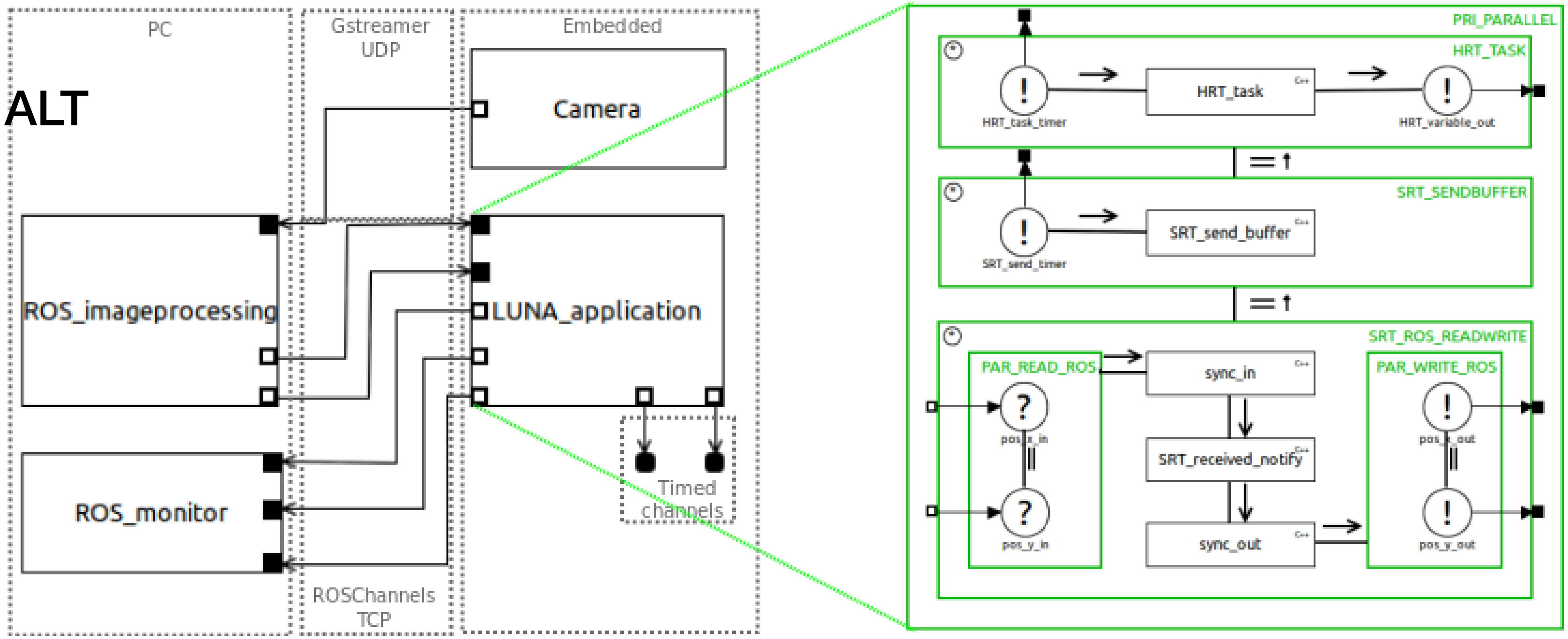
Initialisation JIWY setup

- Initialisation
  - of ROS nodes and topics
  - via the ROS-LUNA bridge
  - ROS topic / message graphs
    - before, after LUNA app connects
- Tests
  - as expected



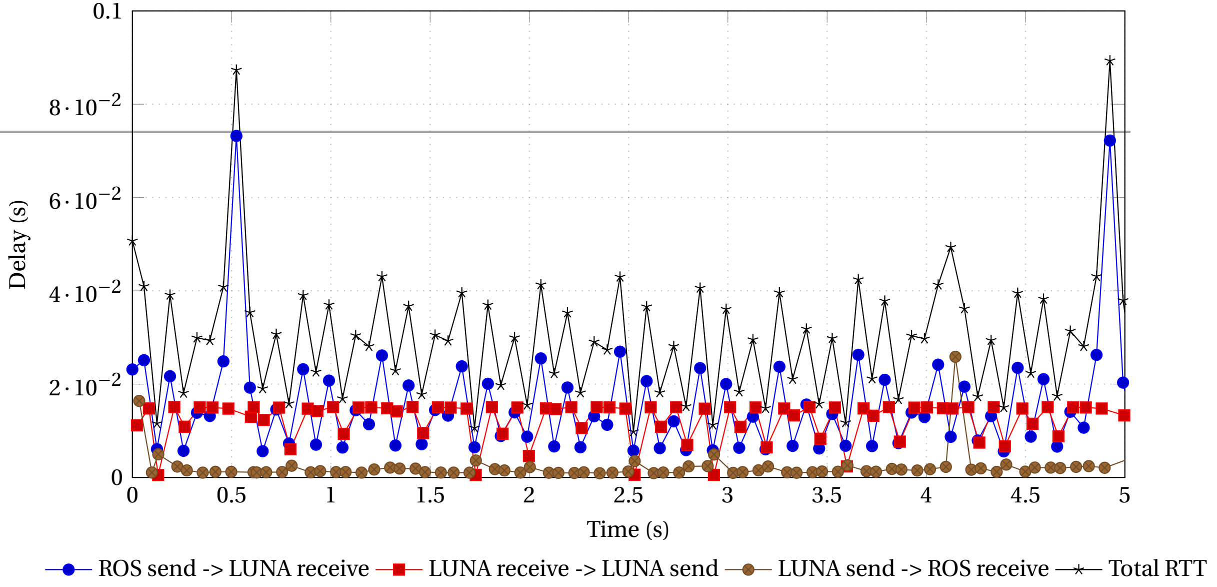
Timing tests JIWY setup

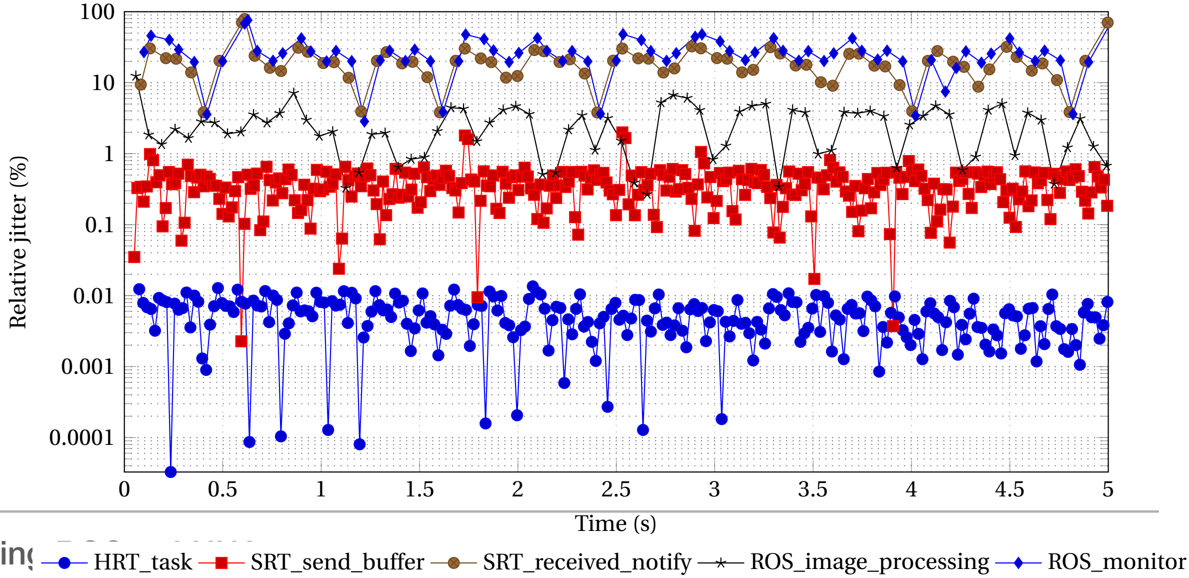
- Only ROS-LUNA bridge over the network
- two tasks concurrently
  - transporting images
    - video file and camera images
  - hard-real time task @ higher freq: 500 Hz
    - writing packages to ROS @ 62.5 Hz
- In LUNA
  - priority via PRI ALT



Timing Tests Results

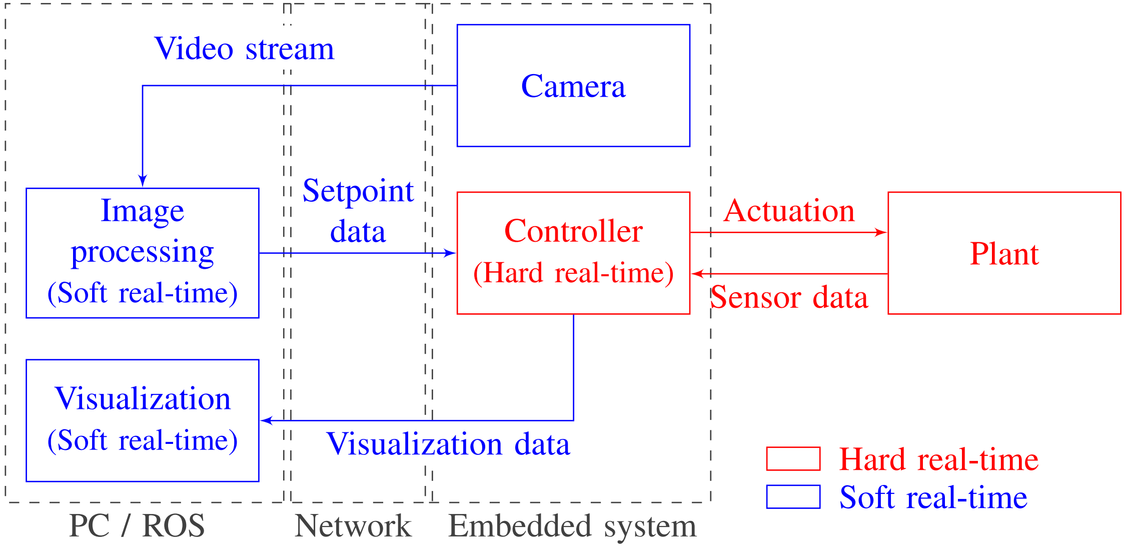
- **Tests**
  - timestamps recorded
  - variation (= jitter) calculated
- **Results - Jitter**
  - at LUNA side
    - HRT Jitter: 0.265 %
    - SRT Jitter : 0.373 %
    - both timed via timer channel
  - on PC - ROS
    - SRT notify: 18.3 %
    - ROS monitor: 21.7 %
- **Results - delays**
  - Round trip 31.5 ms, large variation
    - ROS -> LUNA 15.5
    - inside LUNA 13.4
    - back to ROS 2.6

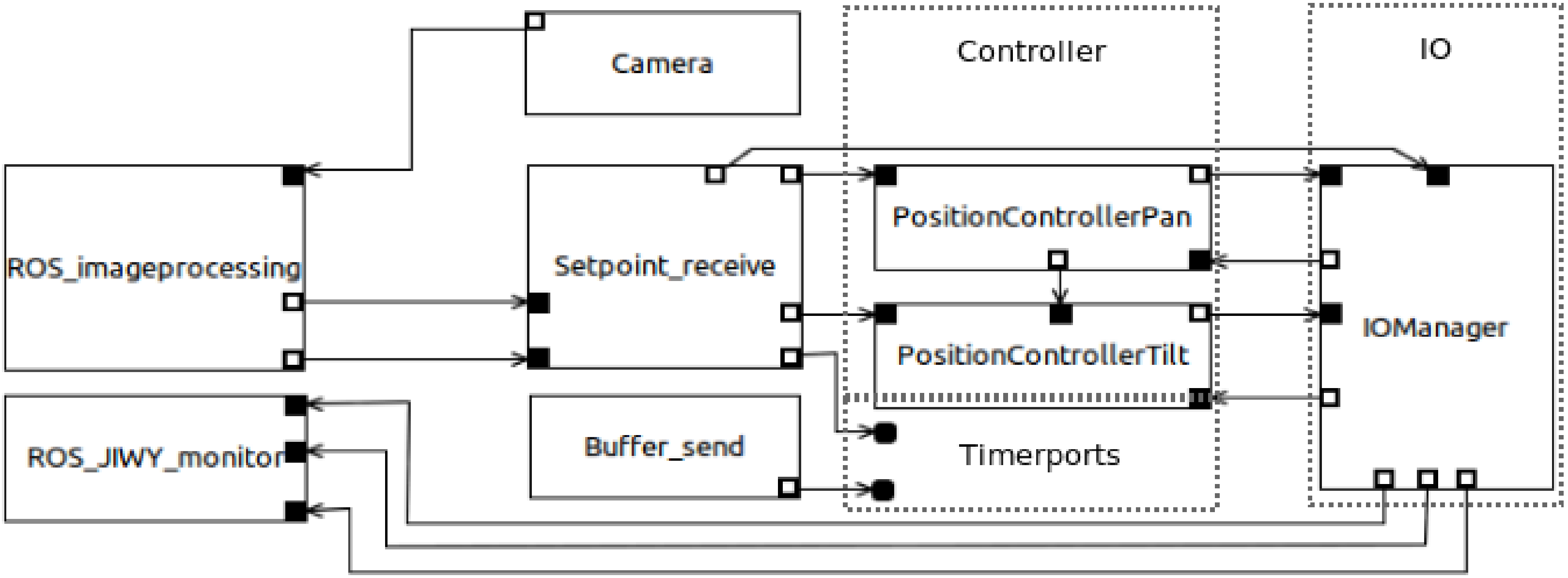




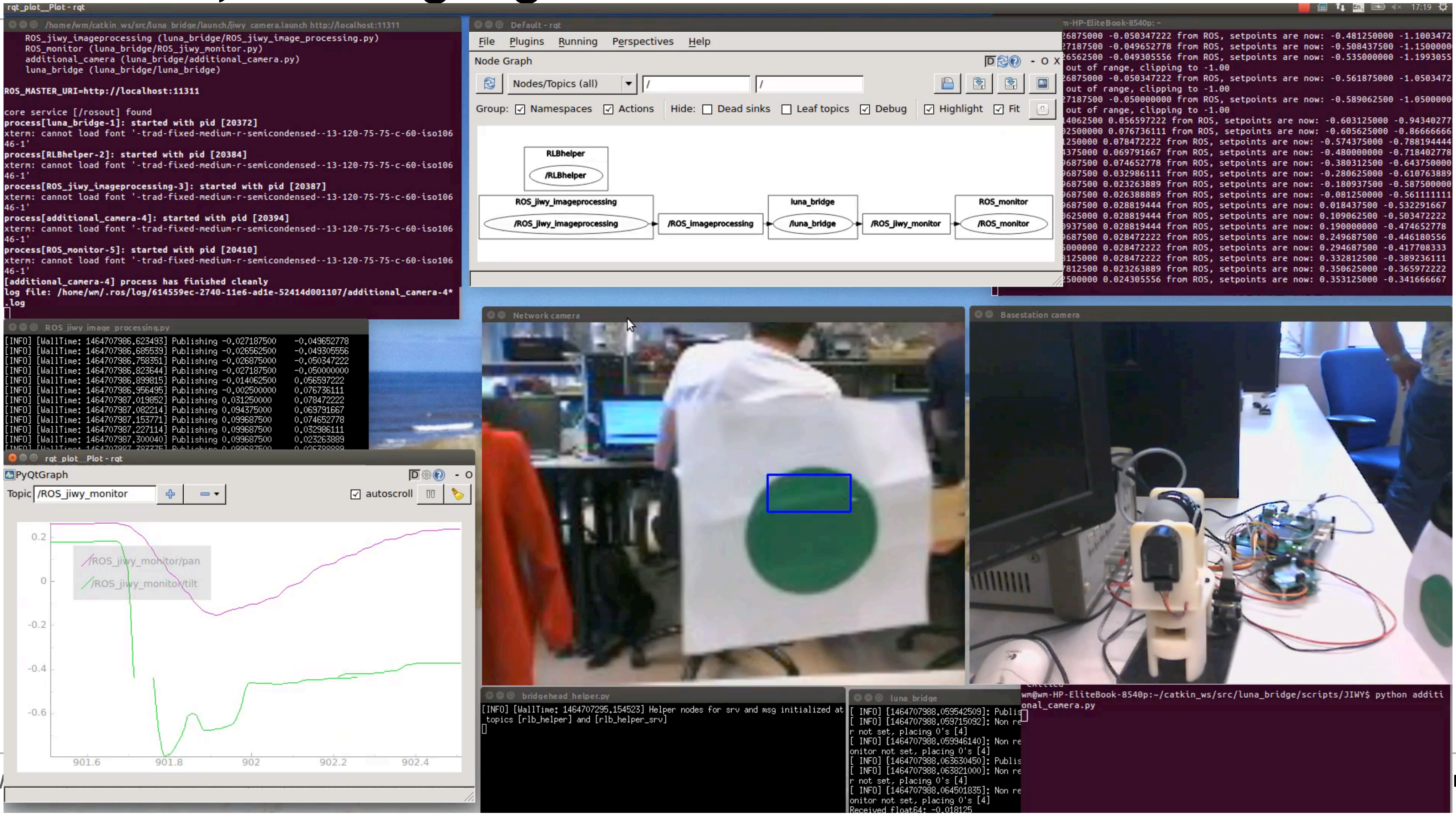
Complete Robotic system

- Controlling Robotic Setup
  - controllers @ 100 Hz
- System
  - overview
  - architecture in TERRA





Results, tracking a green blob



```
    ROS_jiwy_imageprocessing (luna_bridge/ROS_jiwy_image_processing.py)
    ROS_monitor (luna_bridge/ROS_jiwy_monitor.py)
    additional_camera (luna_bridge/additional_camera.py)
    luna_bridge (luna_bridge/luna_bridge)

ROS_MASTER_URI=http://localhost:11311

core service [/rosout] found
process[luna_bridge-1]: started with pid [20372]
xterm: cannot load font '-trad-fixed-medium-r-semicondensed--13-120-75-75-c-60-iso106
46-1'
process[RLBhelper-2]: started with pid [20384]
xterm: cannot load font '-trad-fixed-medium-r-semicondensed--13-120-75-75-c-60-iso106
46-1'
process[ROS_jiwy_imageprocessing-3]: started with pid [20387]
xterm: cannot load font '-trad-fixed-medium-r-semicondensed--13-120-75-75-c-60-iso106
46-1'
process[additional_camera-4]: started with pid [20394]
xterm: cannot load font '-trad-fixed-medium-r-semicondensed--13-120-75-75-c-60-iso106
46-1'
process[ROS_monitor-5]: started with pid [20410]
xterm: cannot load font '-trad-fixed-medium-r-semicondensed--13-120-75-75-c-60-iso106
46-1'
[additional_camera-4] process has finished cleanly
log file: /home/wm/.ros/log/614559ec-2740-11e6-ad1e-52414d001107/additional_camera-4*
.log
```

```
26875000 -0.050347222 from ROS, setpoints are now: -0.481250000 -1.1003472
27187500 -0.049652778 from ROS, setpoints are now: -0.508437500 -1.1500000
26562500 -0.049305556 from ROS, setpoints are now: -0.535000000 -1.1993055
 out of range, clipping to -1.00
26875000 -0.050347222 from ROS, setpoints are now: -0.561875000 -1.0503472
 out of range, clipping to -1.00
27187500 -0.050000000 from ROS, setpoints are now: -0.589062500 -1.0500000
 out of range, clipping to -1.00
14062500 0.056597222 from ROS, setpoints are now: -0.603125000 -0.94340277
02500000 0.076736111 from ROS, setpoints are now: -0.605625000 -0.86666666
1250000 0.078472222 from ROS, setpoints are now: -0.574375000 -0.788194444
4375000 0.069791667 from ROS, setpoints are now: -0.480000000 -0.718402778
9687500 0.074652778 from ROS, setpoints are now: -0.380312500 -0.643750000
9687500 0.032986111 from ROS, setpoints are now: -0.280625000 -0.610763889
9687500 0.023263889 from ROS, setpoints are now: -0.180937500 -0.587500000
9687500 0.026388889 from ROS, setpoints are now: -0.081250000 -0.561111111
9687500 0.028819444 from ROS, setpoints are now: 0.018437500 -0.532291667
0625000 0.028819444 from ROS, setpoints are now: 0.109062500 -0.503472222
0937500 0.028819444 from ROS, setpoints are now: 0.190000000 -0.474652778
9687500 0.028472222 from ROS, setpoints are now: 0.249687500 -0.446180556
5000000 0.028472222 from ROS, setpoints are now: 0.294687500 -0.417708333
3125000 0.028472222 from ROS, setpoints are now: 0.332812500 -0.389236111
7812500 0.023263889 from ROS, setpoints are now: 0.350625000 -0.365972222
2500000 0.024305556 from ROS, setpoints are now: 0.353125000 -0.341666667
```

```
[INFO] [WallTime: 1464707986.623493] Publishing -0.027187500    -0.049652778
[INFO] [WallTime: 1464707986.685539] Publishing -0.026562500    -0.049305556
[INFO] [WallTime: 1464707986.758351] Publishing -0.026875000    -0.050347222
[INFO] [WallTime: 1464707986.823644] Publishing -0.027187500    -0.050000000
[INFO] [WallTime: 1464707986.899815] Publishing -0.014062500    0.056597222
[INFO] [WallTime: 1464707986.956495] Publishing -0.002500000    0.076736111
[INFO] [WallTime: 1464707987.019852] Publishing 0.031250000     0.078472222
[INFO] [WallTime: 1464707987.082214] Publishing 0.094375000     0.069791667
[INFO] [WallTime: 1464707987.153771] Publishing 0.099687500     0.074652778
[INFO] [WallTime: 1464707987.227114] Publishing 0.099687500     0.032986111
[INFO] [WallTime: 1464707987.300040] Publishing 0.099687500     0.023263889
```

```
[INFO] [WallTime: 1464707295.154523] Helper nodes for srv and msg initialized at
 topics [rlb_helper] and [rlb_helper_srv]
```

```
[ INFO] [1464707988.059542509]: Publis
[ INFO] [1464707988.059715092]: Non re
r not set, placing 0's [4]
[ INFO] [1464707988.059946140]: Non re
onitor not set, placing 0's [4]
[ INFO] [1464707988.063630450]: Publis
[ INFO] [1464707988.063821000]: Non re
r not set, placing 0's [4]
[ INFO] [1464707988.064501835]: Non re
onitor not set, placing 0's [4]
Received float64: -0.018125
```

```
wm@wm-HP-EliteBook-8540p:~/catkin_ws/src/luna_bridge/scripts/JIWY$ python additi
onal_camera.py
```

Conclusions and Recommendations

- ROS - LUNA bridge runs
  - SRT - HRT connection in a natural way
  - Reusable / Flexible
    - at the price of some more delay
  - Demo application suffers from delay

- Recommendations
  - Complete support in TERRA
    - to avoid modifying generated code to use ROS-channels
  - ROS runtime binding
    - can be used in other HRT systems than LUNA

Figure 15 Setpoint Receive Blok

- to read from Im Proc and produce setpoints

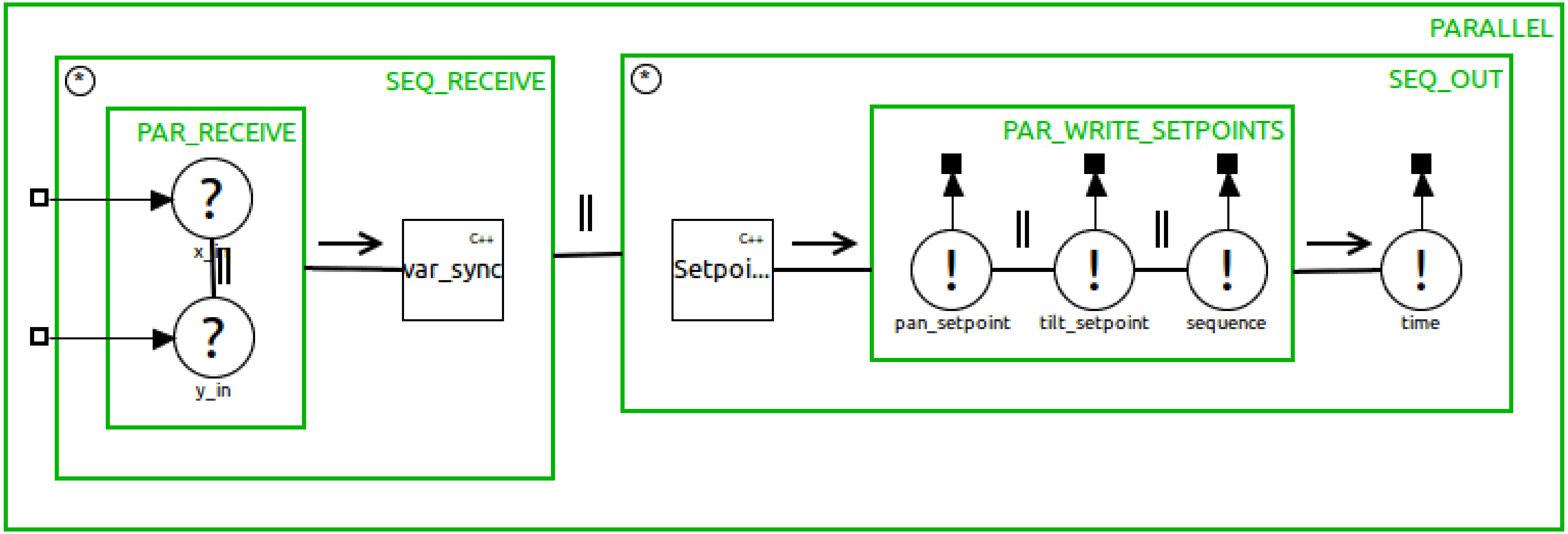
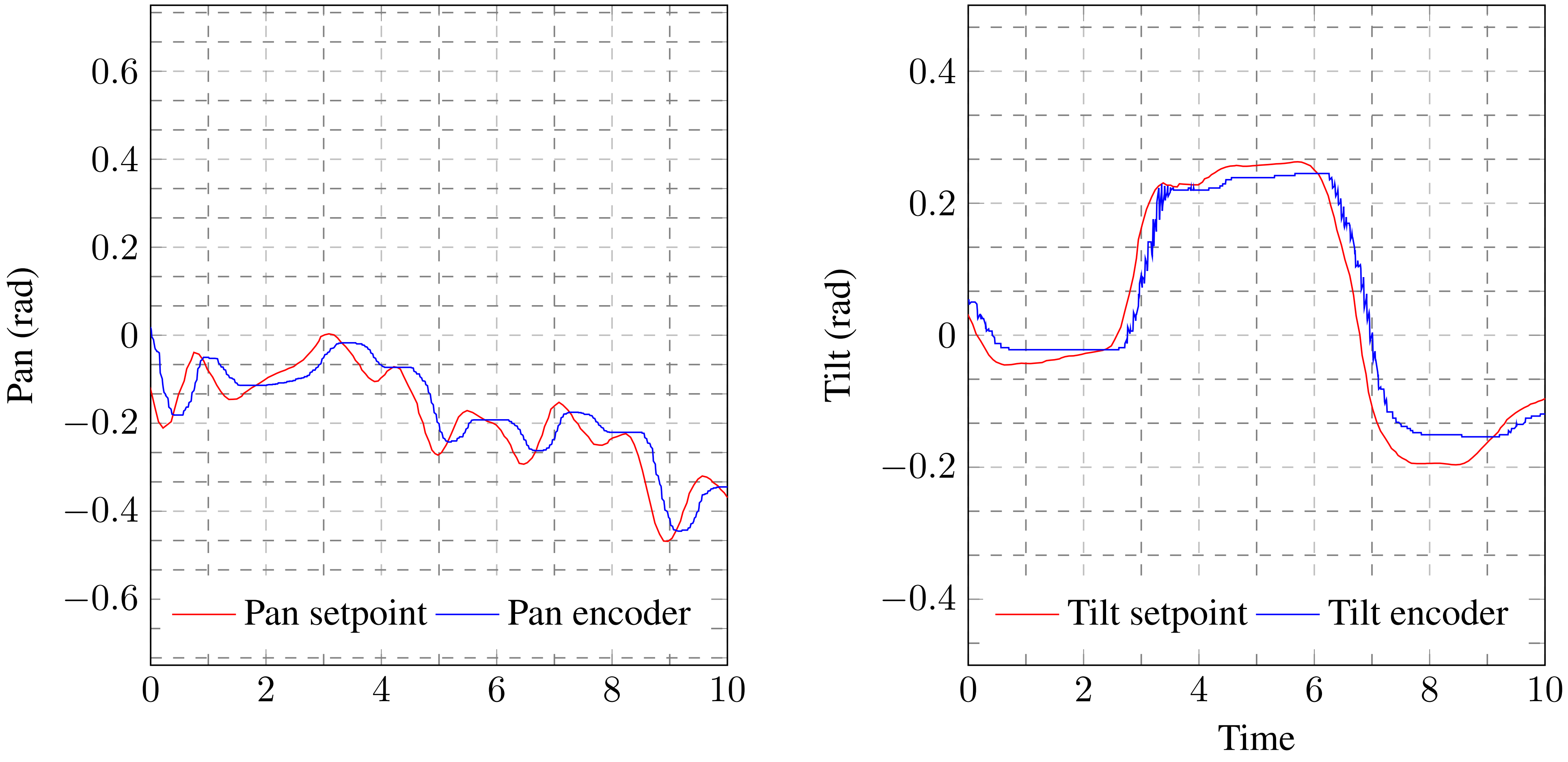


Figure 17: signals supporting the JIWY movie

-


Pan (rad)
Pan setpoint
Pan encoder
Tilt (rad)
Tilt setpoint
Tilt encoder
Time