

Open Source Robot

Vladimír Petřík
vladko.petrik@gmail.com



Alpaca

<http://alpaca.sh.cvut.cz>



Silicon Hill

<http://www.siliconhill.cz>

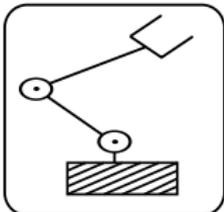
November 3, 2014

Table of Contents

- 1 Robots
 - What is a typical Robot
 - Interfaces
 - Common Sensors
- 2 Robotic Operating System
 - Introduction to ROS
 - Why ROS?
 - Examples
- 3 Agile Infrastructure
 - Build server
 - What to test
 - Tests feedback

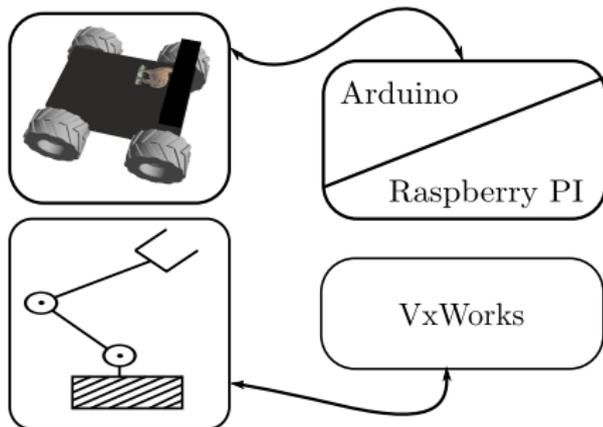
What is a typical Robot

- Hardware entity which is able to move and sense
- Controlled by low-level architecture
 - Arduino
 - Raspberry PI (RealTime kernel)
 - VxWorks (RealTime OS)
- Connected to computer



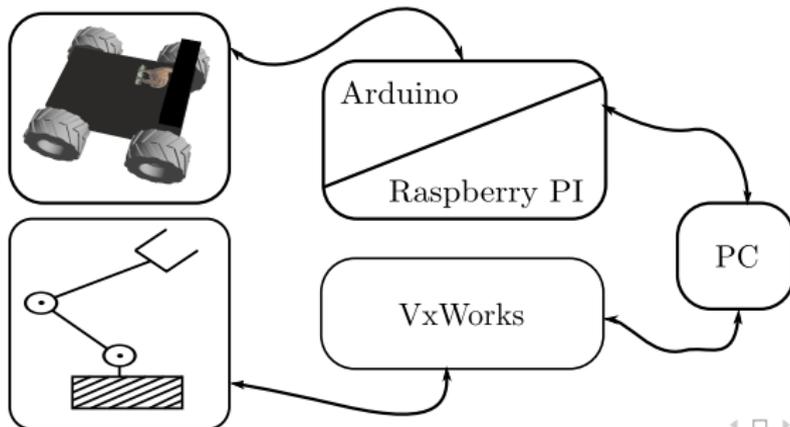
What is a typical Robot

- Hardware entity which is able to move and sense
- Controlled by low-level architecture
 - Arduino
 - Raspberry PI (RealTime kernel)
 - VxWorks (RealTime OS)
- Connected to computer



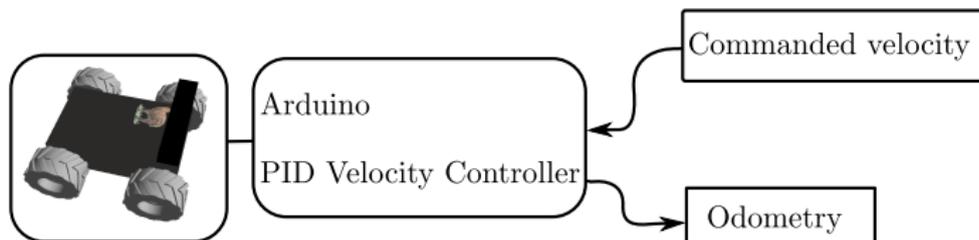
What is a typical Robot

- Hardware entity which is able to move and sense
- Controlled by low-level architecture
 - Arduino
 - Raspberry PI (RealTime kernel)
 - VxWorks (RealTime OS)
- Connected to computer



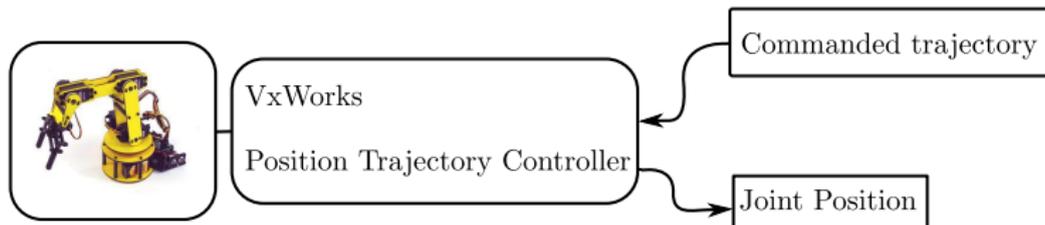
Mobile Robots

- Controlled by velocity command
- Provides odometry information (number of wheel rotations)
- Can be integrated to estimate relative position



Robotics manipulators

- Controlled by position based trajectory command ¹
- Provides absolute (precise) position information
- It can be used to compute position of the end effector



¹Video of robotics folding

Commonly used sensors

- Camera
- Ultrasonic range sensor
- Laser scanner
- Xtion / Kinect / LeapMotion



- Is there a framework/standard for all of mentioned?
- **Robotic Operating System**

Commonly used sensors

- Camera
- Ultrasonic range sensor
- Laser scanner
- Xtion / Kinect / LeapMotion



- Is there a framework/standard for all of mentioned?
- **Robotic Operating System**

ROS

ROS.org

- 6 years old, OSRF
- Ubuntu, [Arch Linux]
- Meta-operating system, middleware
 - Process communication
 - Package management
 - Hardware abstraction
 - Language independence (Python, C++, Lisp)

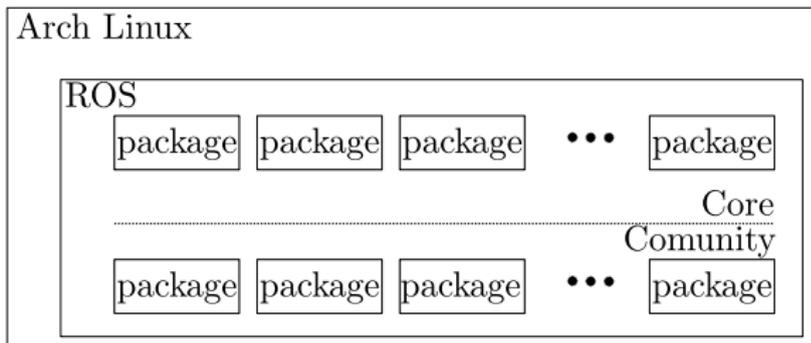
ROS

ROS.org

- 6 years old, OSRF
- Ubuntu, [Arch Linux]
- Meta-operating system, middleware
 - Process communication
 - Package management
 - Hardware abstraction
 - Language independence (Python, C++, Lisp)

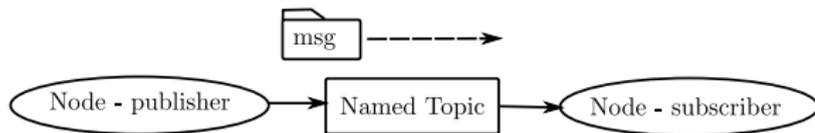
Package Management

- Everything is organized in packages
- Package consists of several nodes (piece of code)
- Nodes communicates (share information) with each other
- Example of package: *sb_image_proc*
 - Node: *number_detector*
 - Node: *image_divider*



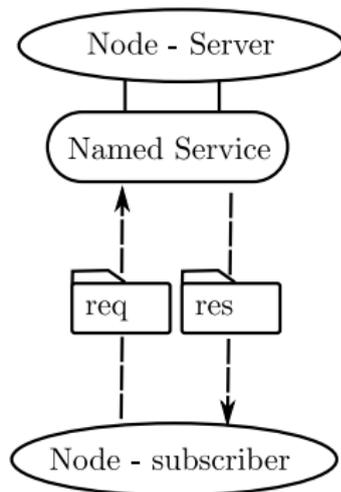
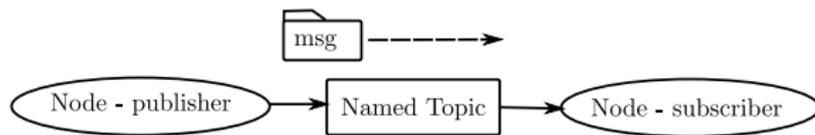
Nodes communication

- Publisher - Subscriber Architecture (Topic)
- Client - Server Architecture (Service)
- Rosparam for parametrization of nodes



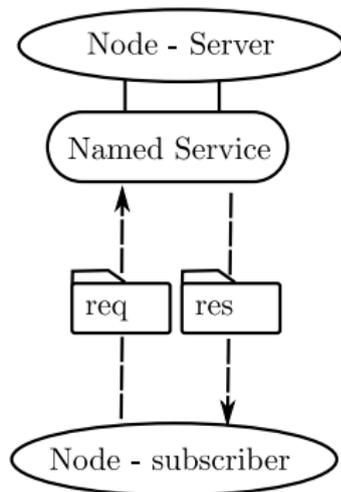
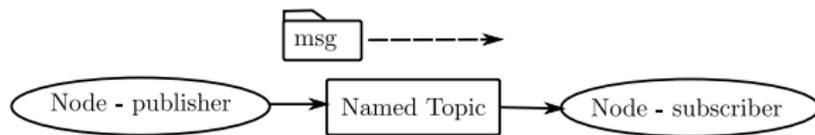
Nodes communication

- Publisher - Subscriber Architecture (Topic)
- Client - Server Architecture (Service)
- Rosparam for parametrization of nodes



Nodes communication

- Publisher - Subscriber Architecture (Topic)
- Client - Server Architecture (Service)
- Rosparam for parametrization of nodes



Nodes communication advance

Services are powerfull but not suitable for longtime goals.

ActionLib

- Suitable for longtime running tasks (> 1 sec).
- Build on top of Topics.
- Non blocking. Preemptable.
- Example *GoTo*
 - Plan collision free path
 - Parametrize and filter path to trajectory
 - Execute trajectory

Nodes communication advance

Services are powerfull but not suitable for longtime goals.

ActionLib

- Suitable for longtime running tasks (> 1 sec).
- Build on top of Topics.
- Non blocking. Preemptable.
- Example *GoTo*
 - Plan collision free path
 - Parametrize and filter path to trajectory
 - Execute trajectory

Nodes communication advance

Services are powerfull but not suitable for longtime goals.

ActionLib

- Suitable for longtime running tasks (> 1 sec).
- Build on top of Topics.
- Non blocking. Preemptable.
- Example *GoTo*
 - Plan collision free path
 - Parametrize and filter path to trajectory
 - Execute trajectory

Nodes communication advance

Services are powerfull but not suitable for longtime goals.

ActionLib

- Suitable for longtime running tasks (> 1 sec).
- Build on top of Topics.
- Non blocking. Preemptable.
- Example *GoTo*
 - Plan collision free path
 - Parametrize and filter path to trajectory
 - Execute trajectory

Why ROS?

- That's all. Nothing special. So why is ROS used across whole robotics community?
- Open Source and simple to use
- Great marketing (PR2 Robot, turtlebot)
- Many packages maintained by large community.
- Interesting for academic researchers as reference.
- **Many "core" packages for visualization and common robotics tasks.**

Why ROS?

- That's all. Nothing special. So why is ROS used across whole robotics community?
- Open Source and simple to use
- Great marketing (PR2 Robot, turtlebot)
- Many packages maintained by large community.
- Interesting for academic researchers as reference.
- Many "core" packages for visualization and common robotics tasks.

Why ROS?

- That's all. Nothing special. So why is ROS used across whole robotics community?
- Open Source and simple to use
- Great marketing (PR2 Robot, turtlebot)
- Many packages maintained by large community.
- Interesting for academic researchers as reference.
- **Many "core" packages for visualization and common robotics tasks.**

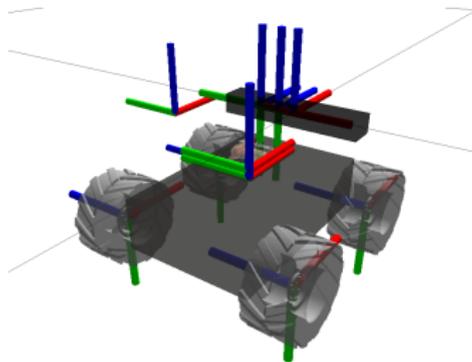
Visualization package

3D visualization tool for ROS ²

- Marker, Markers Array
- Point cloud
- Images
- 2D / 3D Map
- ...

²Video Polygon in RVIZ

Example of transformations



- Transformations in tree structure
- Any transformation relative to any frame
- Buffered in time

Questions ROS

Questions? How would you do the robotics mapping?

- Create low level controller connectable to PC
- Write ROS Node sending cmd_vel to PC
- Run odometry integration node which has been already written
- Run mapping and navigation nodes which have been already written ³

³Video of PR2 Mapping and Mapping Real

Questions ROS

Questions? How would you do the robotics mapping?

- Create low level controller connectable to PC
- Write ROS Node sending cmd_vel to PC
- Run odometry integration node which has been already written
- Run mapping and navigation nodes which have been already written ³

³Video of PR2 Mapping and Mapping Real

Questions ROS

Questions? How would you do the robotics mapping?

- Create low level controller connectable to PC
- Write ROS Node sending cmd_vel to PC
- Run odometry integration node which has been already written
- Run mapping and navigation nodes which have been already written ³

³Video of PR2 Mapping and Mapping Real

Questions ROS

Questions? How would you do the robotics mapping?

- Create low level controller connectable to PC
- Write ROS Node sending `cmd_vel` to PC
- Run odometry integration node which has been already written
- Run mapping and navigation nodes which have been already written ³

³Video of PR2 Mapping and Mapping Real

Questions ROS

Questions? How would you do the robotics mapping?

- Create low level controller connectable to PC
- Write ROS Node sending `cmd_vel` to PC
- Run odometry integration node which has been already written
- Run mapping and navigation nodes which have been already written ³

³Video of PR2 Mapping and Mapping Real

ROS largest problem?

Let's criticise

- ROS is new and thus changing too fast
- Too many not maintained packages
- Many academic researchers provides their code as ROS package
- Code / Publish / Throw away

Is there any solution how to separate well designed codes?
Agile development

ROS largest problem?

Let's criticise

- ROS is new and thus changing too fast
- Too many not maintained packages
- Many academic researchers provides their code as ROS package
- Code / Publish / Throw away

Is there any solution how to separate well designed codes?
Agile development

ROS largest problem?

Let's criticise

- ROS is new and thus changing too fast
- Too many not maintained packages
- Many academic researchers provides their code as ROS package
- Code / Publish / Throw away

Is there any solution how to separate well designed codes?

Agile development

ROS largest problem?

Let's criticise

- ROS is new and thus changing too fast
- Too many not maintained packages
- Many academic researchers provides their code as ROS package
- Code / Publish / Throw away

Is there any solution how to separate well designed codes?

Agile development

Build server

Goal: Needs something to build required codes periodically and checking for possible defects.

Build server

- Connected to git and issue tracker
- Will automatically report errors



Build server

Goal: Needs something to build required codes periodically and checking for possible defects.

Build server

- Connected to git and issue tracker
- Will automatically report errors

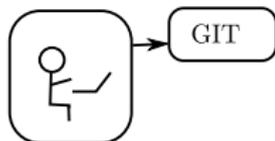


Build server

Goal: Needs something to build required codes periodically and checking for possible defects.

Build server

- Connected to git and issue tracker
- Will automatically report errors

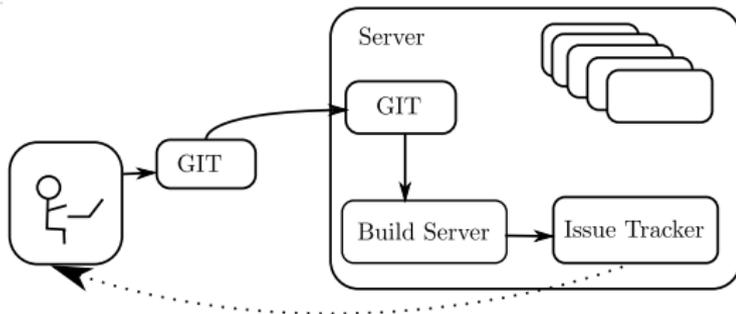


Build server

Goal: Needs something to build required codes periodically and checking for possible defects.

Build server

- Connected to git and issue tracker
- Will automatically report errors



Tests feedback

How to motivate programmers to use test driven development?

- They are experienced and know it is needed
- We will use some sort of measurement function and specify threshold
- **Test Coverage Tools**

Tests feedback

How to motivate programmers to use test driven development?

- They are experienced and know it is needed
- We will use some sort of measurement function and specify threshold
- **Test Coverage Tools**

Tests feedback

How to motivate programmers to use test driven development?

- They are experienced and know it is needed
- We will use some sort of measurement function and specify threshold
- Test Coverage Tools

Tests feedback

How to motivate programmers to use test driven development?

- They are experienced and know it is needed
- We will use some sort of measurement function and specify threshold
- **Test Coverage Tools**

	Hit	Total	Coverage
Lines:	20	22	90.9 %
Functions:	3	3	100.0 %
Branches:	8	10	80.0 %

Conclusion

To conclude:

- ROS is great tool but it is not perfect.
- It provides core infrastructure simplifying the development.
- Hardware abstraction and community packages.
- There is nothing better yet.

But:

- ROS is under heavy development.
- Changing too often and usually versions incompatibility.
- Many non tested packages.

In Alpaca Robotics Team we decided to focus on agile development to lowerize this problems for our purposes.

Questions

Questions?

Thanks for attention.

Questions

Questions?

Thanks for attention.

