



# INTRODUCTION TO ROBOTICS (CS460)

PROF. ANIS KOUBAA

---

## Overview

<https://www.udemy.com/user/anis-koubaa/>

## INSTRUCTOR

- ▶ Prof. Anis Koubaa
- ▶ Full Professor in Computer Science
- ▶ Research Associate CISTER, Portugal
- ▶ Director of RIOTU Research Lab
- ▶ Research Interest
  - ▶ Internet of Things
  - ▶ Mobile Robots
  - ▶ Deep Learning

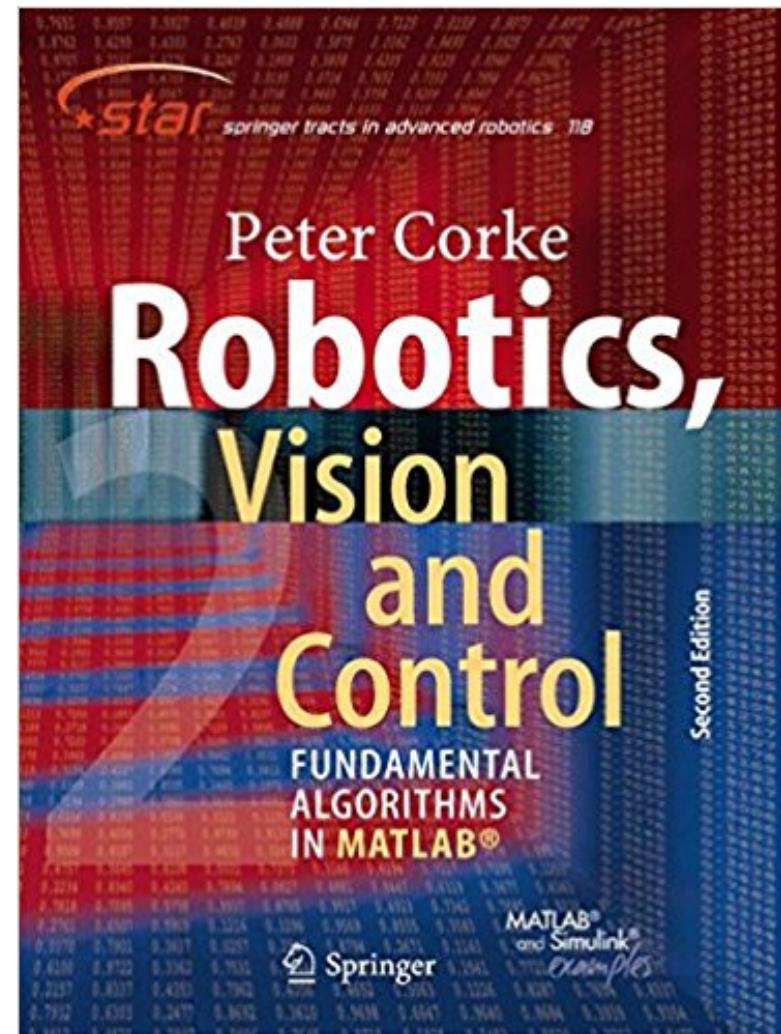
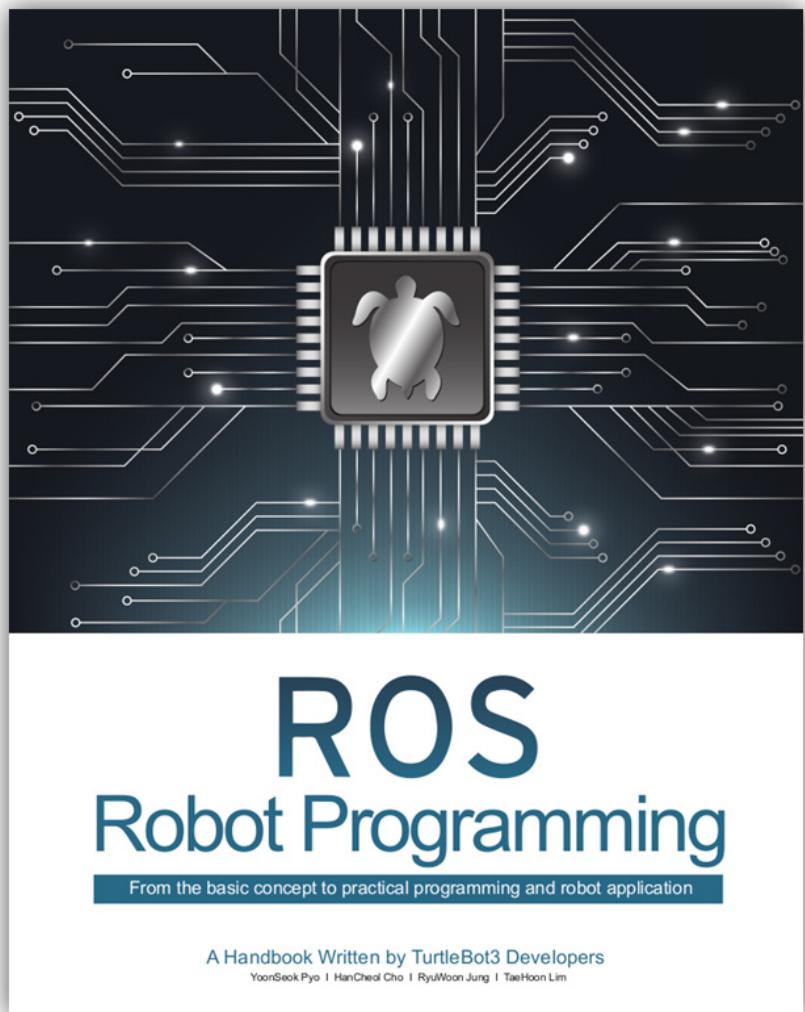
## HOW TO REACH ME

- ▶ Best way: by email [anis.koubaa@gmail.com](mailto:anis.koubaa@gmail.com)
- ▶ Other alternatives
  - ▶ Office Hours: typically from 9:00 to 10:00 am every day
  - ▶ Location: Office E265 or Robotics and IoT Lab
  - ▶ Take an appointment
  - ▶ Call in office: 011 494 8851

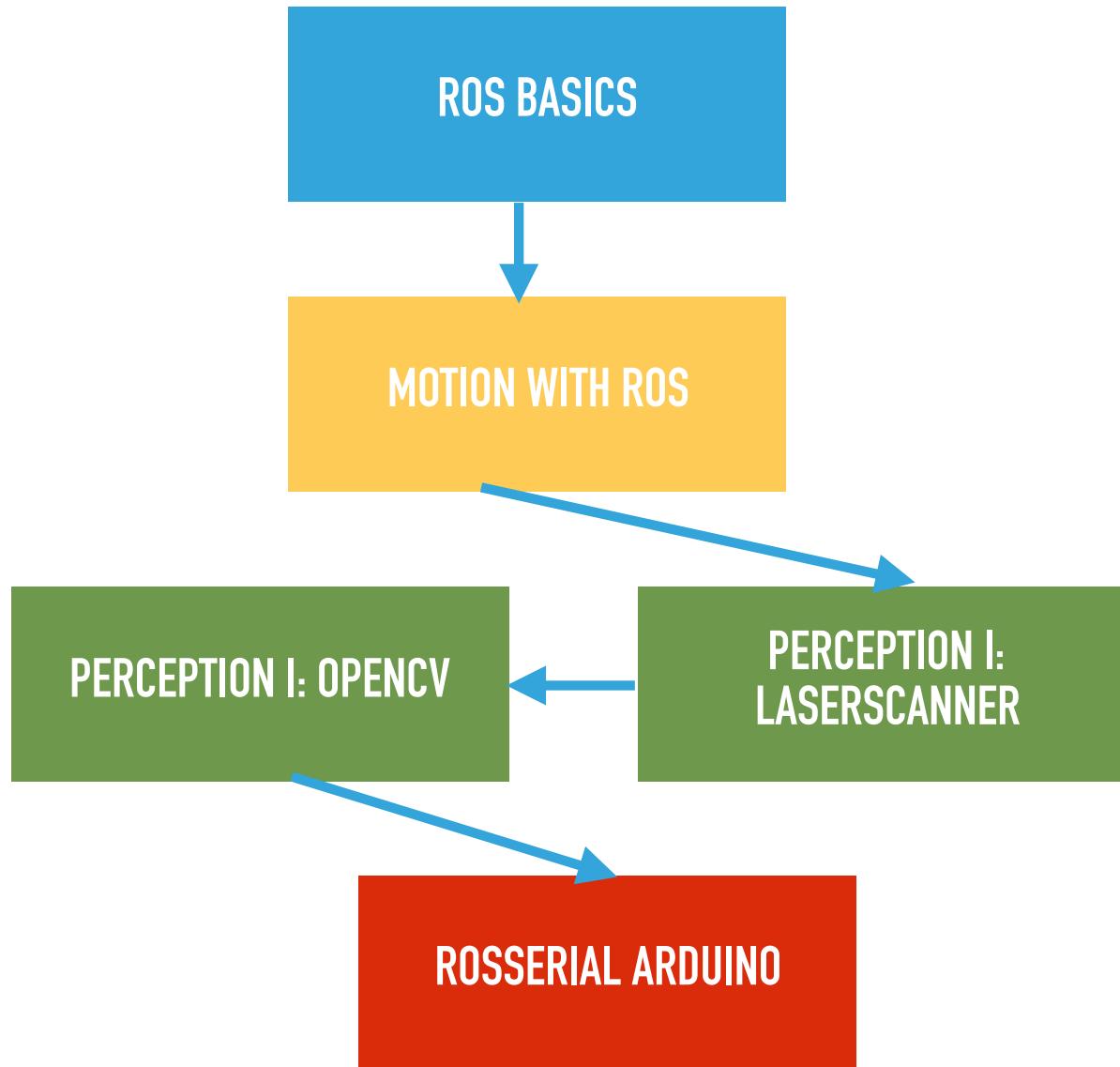
## WHAT YOU SHOULD KNOW?

- ▶ Programming
- ▶ Python and C++ will be used, but no need to have prior knowledge

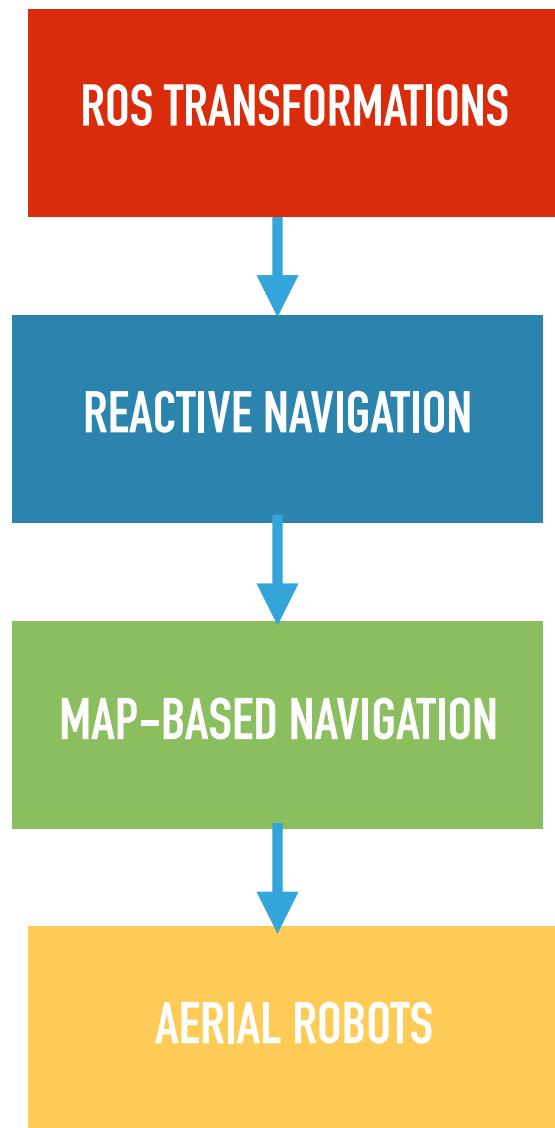
## TEXTBOOK



## LEARNING PATH



# LEARNING PATH



## LEARNING PATH



## LEARNING OUTCOMES

- ▶ Understand ROS Ecosystem (topics, nodes, messages, services, actionlib)
- ▶ Develop simple applications to control robot motion
- ▶ Understand how a position and orientation are represented in ROS
- ▶ Recognize how to develop a C++/Python ROS project
- ▶ Develop simple computer vision programs with ROS and OpenCV
- ▶ Develop application for robot navigation

## TEACHING STRATEGY

- ▶ Teaching by Demonstration
  - ▶ Live Coding
  - ▶ Concepts Explanation
- ▶ Post-Lecture Hands-on Activities
  - ▶ Quizzes
  - ▶ Short coding exercises
  - ▶ Programming assignments
- ▶ Written and/or Video solutions

## TOOLS NEEDED

- ▶ Virtual Machine with ROS Kinetic Pre-Installed on Ubuntu 16.04
- ▶ Laptop recommended requirements
  - ▶ 8 GB of RAM
  - ▶ SSD Hard drive

## COURSE INTERACTION

- ▶ Post questions on the course Q/A Forum
- ▶ Send me feedback
- ▶ Email: [anis.koubaa@gmail.com](mailto:anis.koubaa@gmail.com)

## GRADING

- ▶ Assignments: 10%
- ▶ Major Exam 1: 15%
- ▶ Major Exam 2: 15%
- ▶ Project: 20%
- ▶ Final Exam: 40%

## LEARNING PLATFORMS

- ▶ LMS-Moodle: <https://lms.psu.edu.sa/>
- ▶ Udemy: [udemy.com/ros-essentials](https://www.udemy.com/ros-essentials)
- ▶ Course Website: <http://cs460.coins-lab.org/>



# INTRODUCTION TO ROBOTICS (CS460)

PROF. ANIS KOUBAA

---

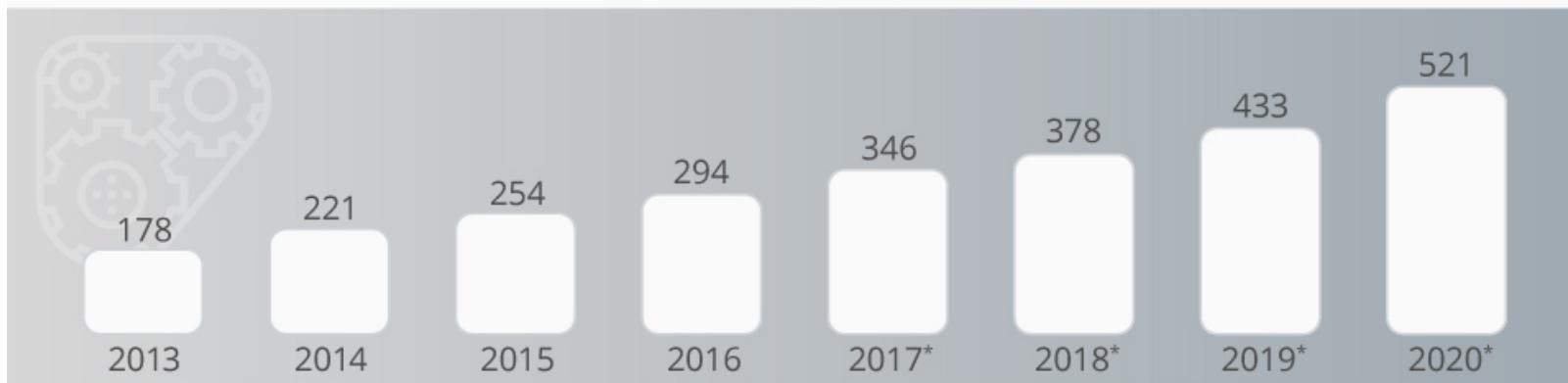
## What is a robot?

<https://www.udemy.com/user/anis-koubaa/>

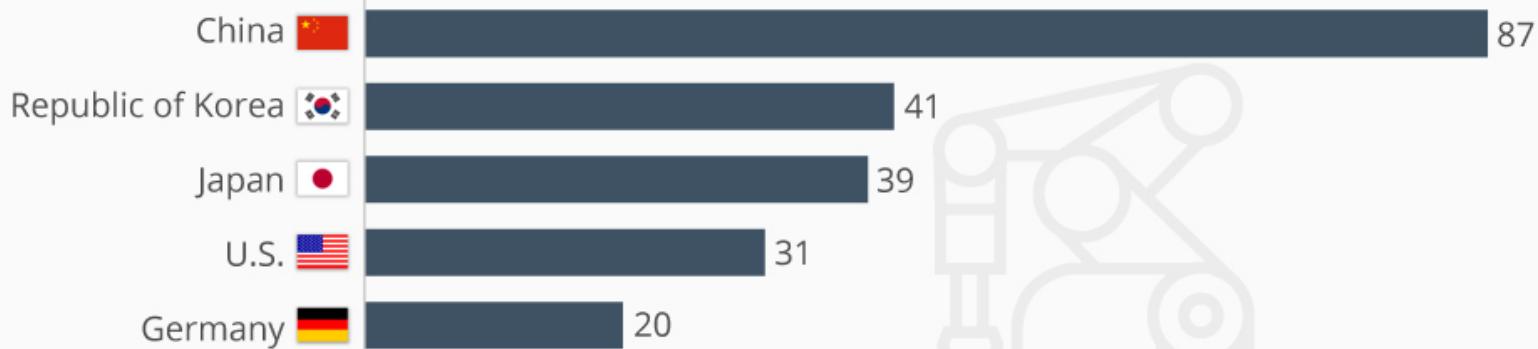
## IMPACT

### Rise of the Industrial Robots

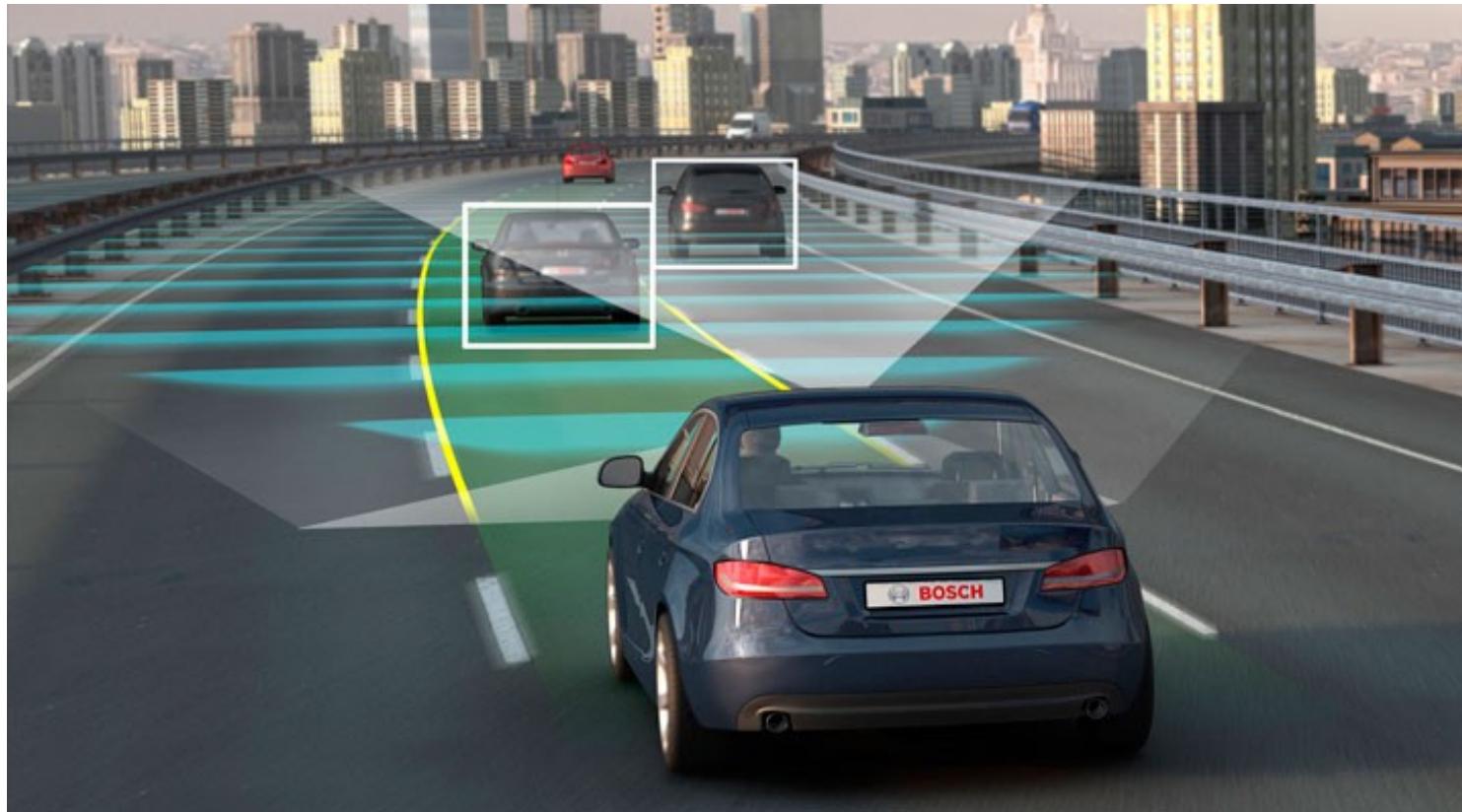
Estimated and forecast supply of industrial robots worldwide 2013–2020 (in thousand units)



#### Countries which supplied the most in 2016



## HOT AREAS: SELF-DRIVING CARS



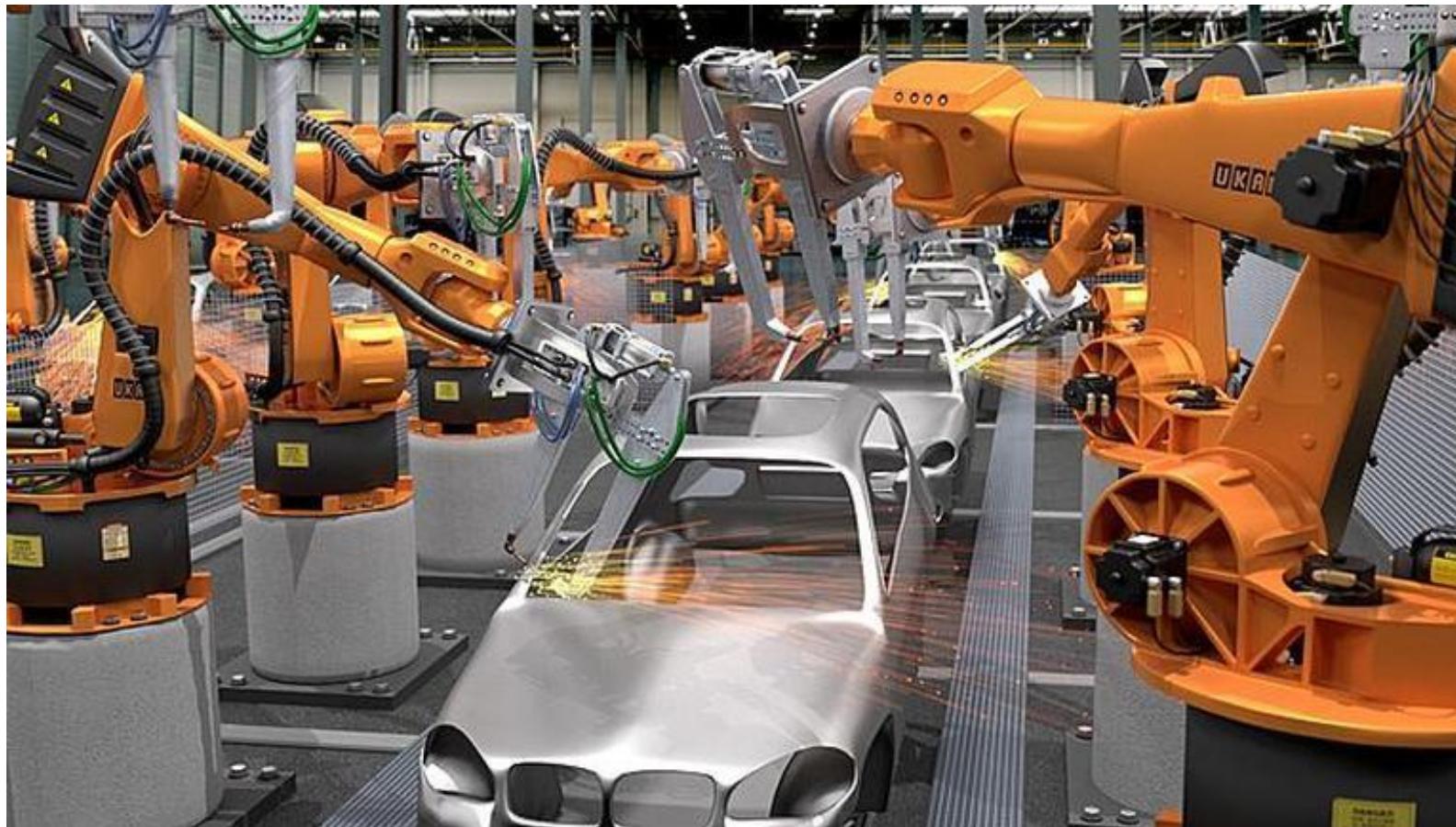
## HOT AREAS: DRONES



## HOT AREAS: SERVICE ROBOTS



## HOT AREAS: INDUSTRIAL ROBOTS





# INTRODUCTION TO ROBOTICS (CS460)

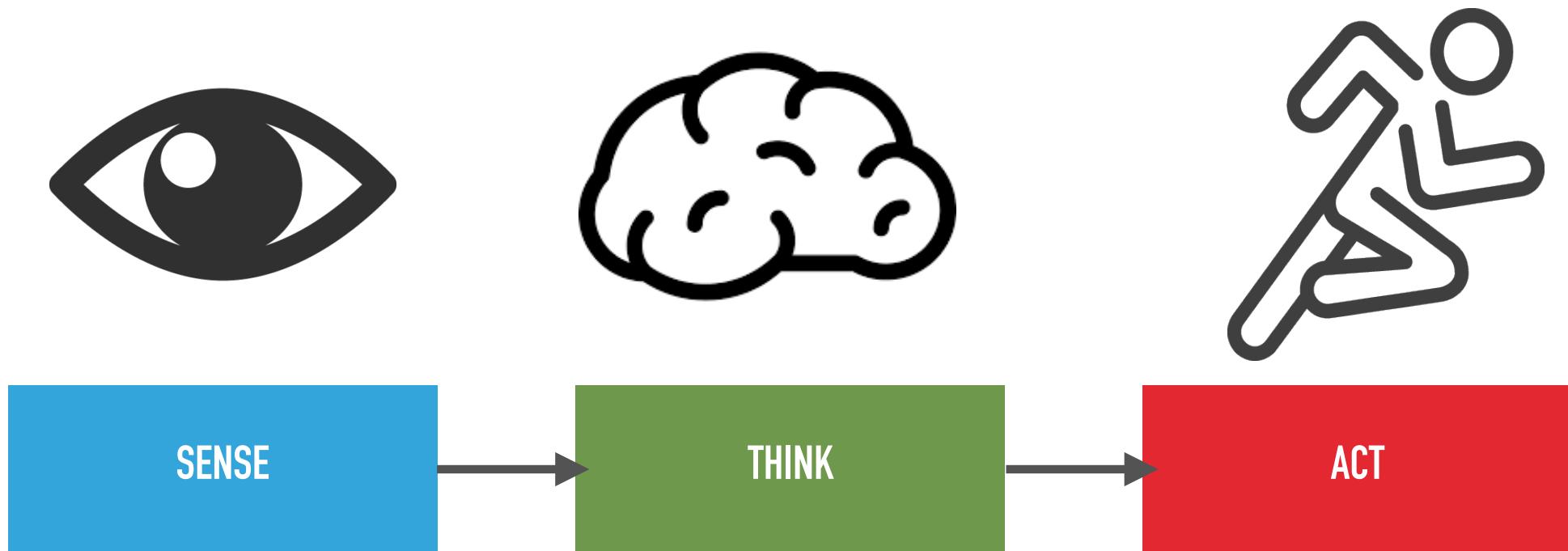
PROF. ANIS KOUBAA

---

## Robot Process Cycle

<https://www.udemy.com/user/anis-koubaa/>

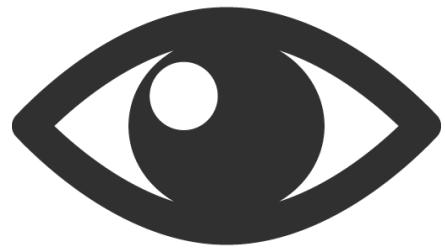
# ROBOT PROCESS CYCLE



## EXAMPLE



SENSE



SENSE

CAMERA



LASERS



ULTRASONIC



GPS



THINK



THINK

ARTIFICIAL INTELLIGENCE

MACHINE LEARNING

SIGNAL PROCESSING

ACT



ACT





# INTRODUCTION TO ROBOTICS (CS460)

PROF. ANIS KOUBAA

---

Robot Operating System  
(ROS)

<https://www.udemy.com/user/anis-koubaa/>



# UDEMY COURSE ROBOT OPERATING SYSTEM BASICS, MOTION, AND OPENCV

PROF. ANIS KOUBAA

---

## ROS Overview

<https://www.udemy.com/user/anis-koubaa/>

## WHAT IS ROS?

- ▶ ROS is an open-source software framework for robot software development
- ▶ Provides an operating system-like functionality



## WHAT IS ROS?

- ▶ Provides OS services
- ▶ Hardware abstraction
- ▶ Low-level device control
- ▶ Implementation of commonly-used functionality
- ▶ Message-passing between processes
- ▶ Package management

# WHAT IS ROS?

USER APPS

ROS

ROBOT HARDWARE  
(SENSORS, ACTUATORS)

## ROS ECOSYSTEM

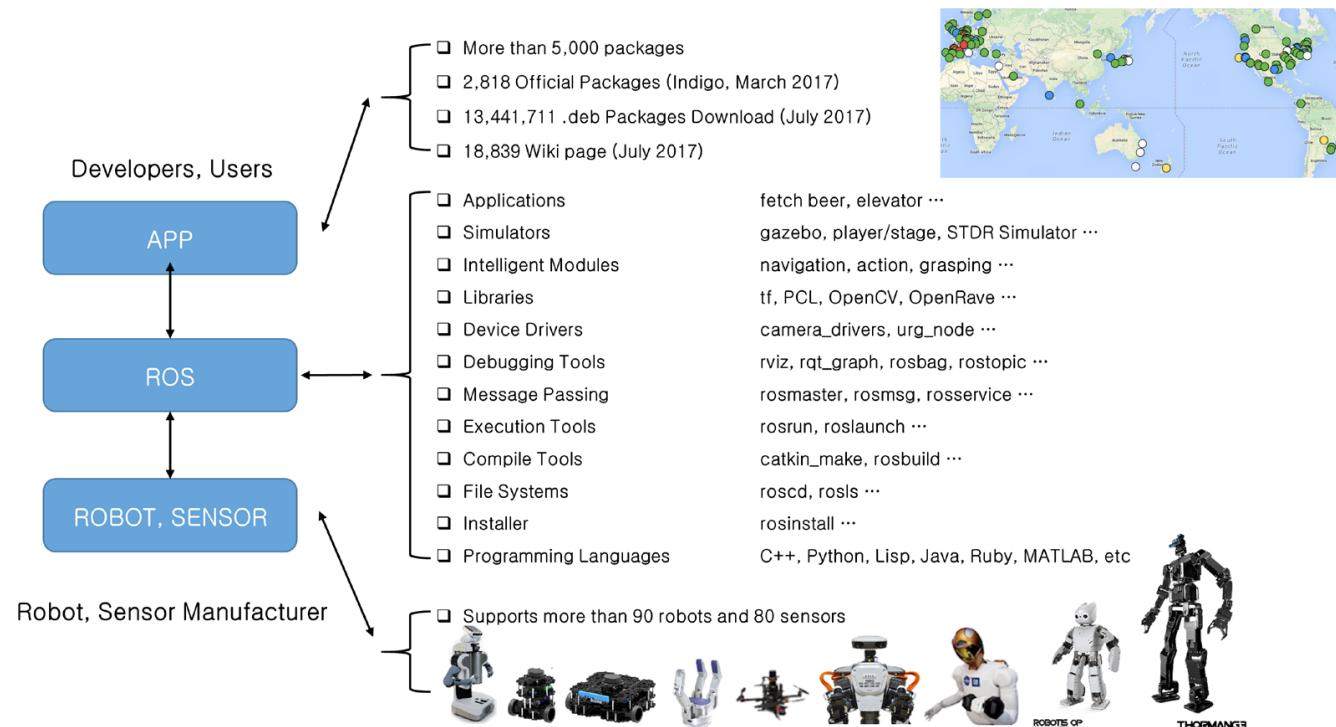


FIGURE 2-4 ROS Ecosystem

Reference Book: **ROS Robot Programming** (in English). Authors: Yoonseok Pyo, Hancheol Cho, Leon Jung, Darby Lim

# INTRODUCTION TO ROBOTICS (CS460)

## ROS DISTRIBUTIONS

- Dec 08, 2017 - Release of ROS 2.0
- Sep 21, 2017 - ROSCon2017 (Canada)
- May 23, 2017 - Release of Lunar Loggerhead
- May 16, 2017 - Changed name from OSRF to Open Robotics
- Oct 8, 2016 - ROSCon2016 (South Korea)
- May 23, 2016 - Release of Kinetic Kame
- Oct 3, 2015 - ROSCon2015 (Germany)
- May 23, 2015 - Release of Jade Turtle
- Sep 12, 2014 - ROSCon2014 Conference (U.S.)
- Jul 22, 2014 - Release of Indigo Igloo
- Jun 6, 2014 - ROS Kong 2014 Conference (Hong Kong)
- Sep 4, 2013 - Release of Hydro Medusa
- May 11, 2013 - ROSCon2013 Conference (Germany)
- Feb 11, 2013 - Open Source Robotics Foundation takes on development/management
- Dec 31, 2012 - Release of Groovy Galapagos
- May 19, 2012 - ROSCon2012 Conference (U.S.)
- Apr 23, 2012 - Release of Fuerte
- Aug 30, 2011 - Release of Electric Emys
- Mar 2, 2011 - Release of Diamondback
- Aug 2, 2010 - Release of C Turtle
- Mar 2, 2010 - Release of Box Turtle
- Jan 22, 2010 - Release of ROS 1.0
- Nov 1, 2007 - Willow Garage starts development under the name 'ROS'
- May 1, 2007 - Switchyard Project, Morgan Quigley, Stanford AI LAB, Stanford University
- 2000 - Player/Stage Project, Brian Gerkey, University of Southern California (USC)

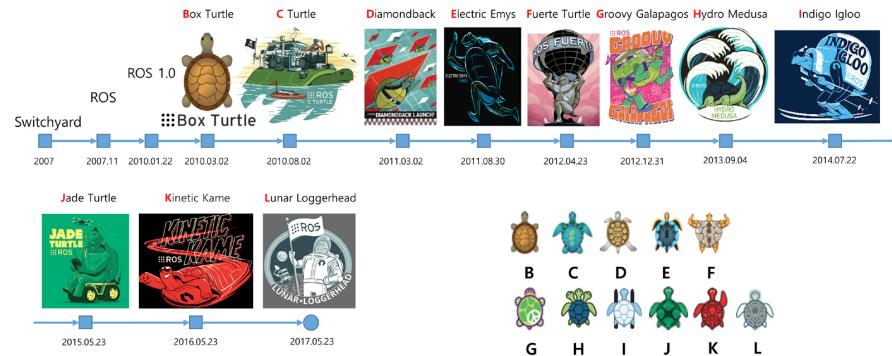


FIGURE 2-7 ROS Versions(<http://wiki.ros.org/>)



FIGURE 2-8 Turtle icons for each version of ROS

Distro	Release Date	Poster	Symbol	EOL Date
Lunar Loggerhead	2017.05.23			2019.05
Kinetic Kame (Recommended)	2016.05.23			2021.04 (Xenial EOL)
Jade Turtle	2015.05.23			2017.05
Indigo Igloo	2014.07.22			2019.04 (Trusty EOL)

FIGURE 2-9 Recent ROS versions and end of support date

## WHY ROS?

- ▶ Code reuse in Robotics R&D
- ▶ Ready-to-use development environment
- ▶ Comprehensive tools and client API libraries
- ▶ Scalable (distributed network of processes loosely coupled)
- ▶ Large community
- ▶ Continuous support



# INTRODUCTION TO ROBOTICS (CS460)

PROF. ANIS KOUBAA

---

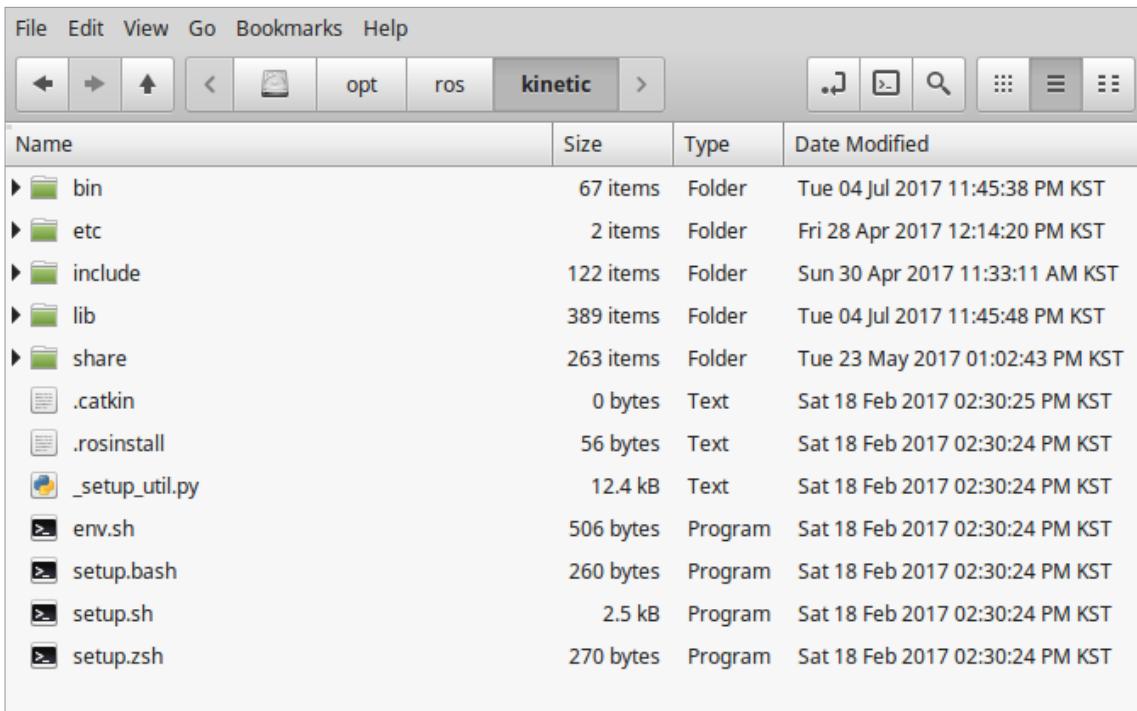
## ROS File System

<https://www.udemy.com/user/anis-koubaa/>

## HANDS-ON ACTIVITY

- ▶ Open Virtual Machine
- ▶ The instructor will explain the file ecosystem through a live demonstration
- ▶ Follow the instructions of the instructors
- ▶ Take note the new terminologies.

# ROS FILE SYSTEM CONFIGURATION



- /bin Executable Binary Files
- /etc ROS and Catkin related Configuration Files
- /include Header Files
- /lib Library Files
- /share ROS Packages
- env.\* Environment Configuration Files
- setup.\* Environment Configuration Files

FIGURE 4-19 ROS File Configuration

# ROS FILE SYSTEM CONFIGURATION

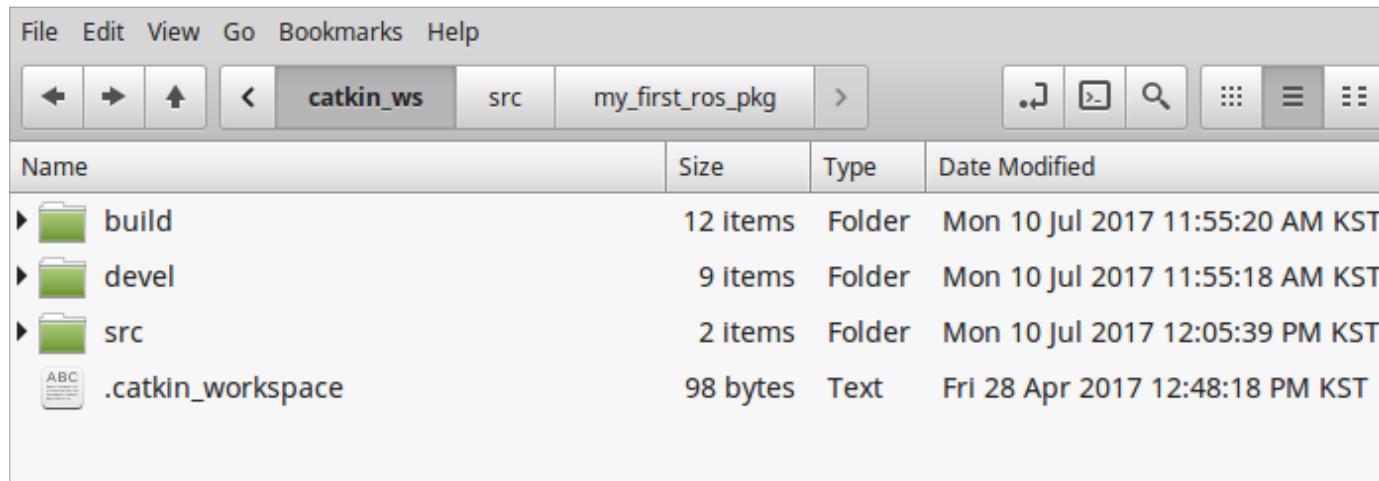


FIGURE 4-20 File Configuration of catkin workspace

- /build      Build Related Files
- /devel      msg, srv Header Files and User Package Library, Execution Files
- /src        User Packages

## ROS FILE SYSTEM CONFIGURATION

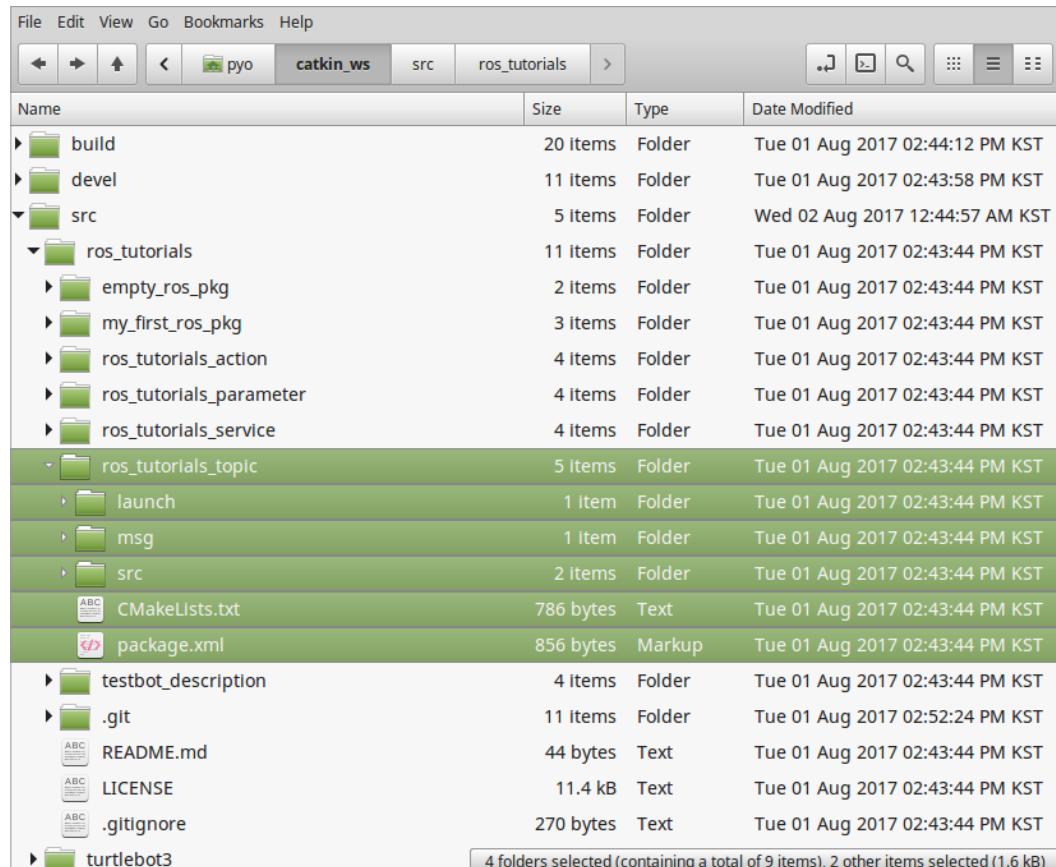


FIGURE 4-21 File Configuration of the User Package

- /include Header Files
- /launch Launch Files Used with roslaunch
- /node Script for rospy
- /msg Message Files
- /src Source Code Files
- /srv Service Files
- CMakeLists.txt Build Configuration File
- package.xml Package Configuration File



UDEMY COURSE  
ROBOT OPERATING SYSTEM  
BASICS, MOTION, AND OPENCV

PROF. ANIS KOUBAA

---

ROS File System Hands-On  
Notes

<https://www.udemy.com/user/anis-koubaa/>

ROS FILE SYSTEM

---

HANDS-ON

---

## CONFIGURE WORKSPACE

- ▶ `source /opt/ros/kinetic/setup.bash`
  - ▶ activate the ROS default workspace
- ▶ Add the command into `.bashrc` file
  - ▶ `.bashrc` is a file executed automatically every time a new terminal is open
- ▶ `roscd`
  - ▶ take you to the default workspace

---

## CONFIGURE WORKSPACE

- ▶ `cd`
  - ▶ takes you to HOME directory
  - ▶ `.bashrc` is in the HOME directory
- ▶ you can create shortcuts and aliases in `.bashrc`
  - ▶ `alias gb="gedit /home/riotu/.bashrc"`
- ▶

---

# CREATE WORKSPACE

- ▶ create your own **catkin** workspace (catkin\_ws) in your HOME directory
  - ▶ your catkin workspace will be used to create and store your own ROS packages (project)
  - ▶ **catkin** is the name of the build tool used to compile and execute programs in ROS
  - ▶ reference: <http://wiki.ros.org/ROS/Tutorials/InstallingandConfiguringROSEnvironment>

```
mkdir -p ~/catkin_ws/src  
$ cd ~/catkin_ws/  
$ catkin_make
```

---

# CREATE A ROS PACKAGE

- ▶ Create your ROS package (project) that you will use to develop programs.
- ▶ the package must be created inside the src folder
- ▶ first move to the folder
  - ▶ `cd ~catkin_ws/src/`
- ▶ Create your package (specify the dependencies)
- ▶ `catkin_create_pkg cs460_package std_msgs rospy roscpp`
- ▶ Go to catkin\_ws and compile
  - ▶ `cd ..`
  - ▶ `catkin_make`
- ▶ this will generate executable and configuration files for the project

---

## MAKE THE NEW PACKAGE THE DEFAULT ONE

- ▶ add the following command in .bashrc
  - ▶ `source /home/riotu/catkin_ws/devel/setup.bash`
  - ▶ *replace riotu by your username*

ROS FILE SYSTEM

---

# PRACTICE TEST 1

## PRACTICE TEST 1

- ▶ Open Udemy Website
- ▶ Respond to Practice Test 1
- ▶ Submit your answer



# INTRODUCTION TO ROBOTICS (CS460)

PROF. ANIS KOUBAA

---

End of Week 1

<https://www.udemy.com/user/anis-koubaa/>



# INTRODUCTION TO ROBOTICS (CS460)

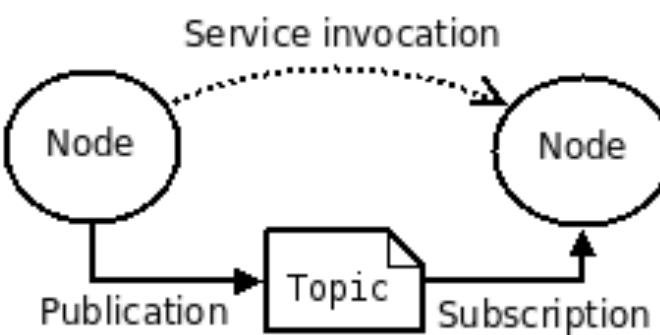
PROF. ANIS KOUBAA

---

## ROS Computation Graph

<https://www.udemy.com/user/anis-koubaa/>

## COMPUTATION GRAPH



## ROS COMPUTATION GRAPH

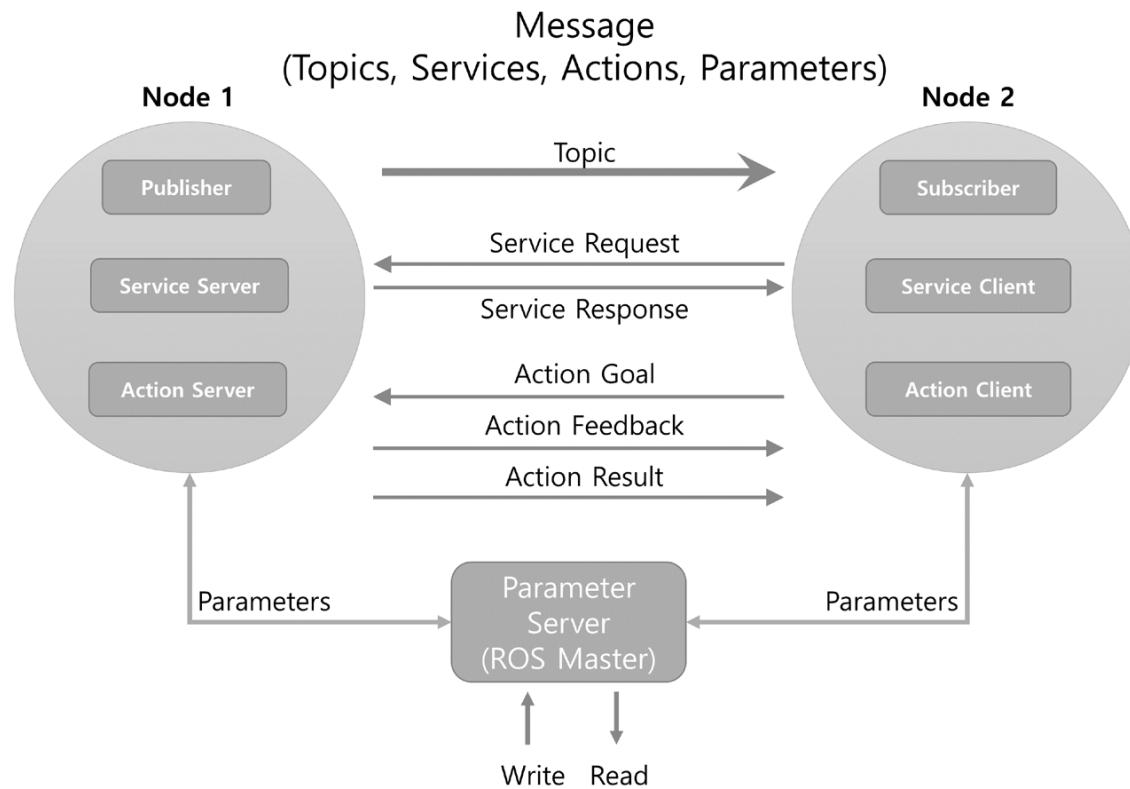
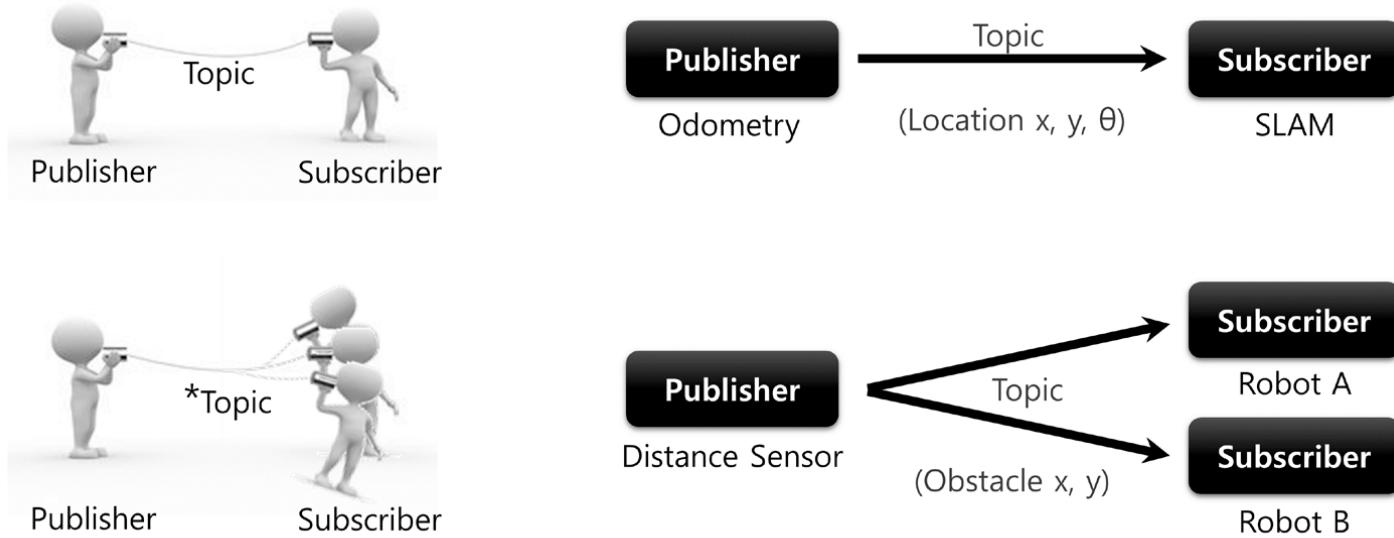


FIGURE 4-1 Message Communication between Nodes

Type	Features		Description
Topic	Asynchronous	Unidirectional	Used when exchanging data continuously
Service	Synchronous	Bi-directional	Used when request processing requests and responds current states
Action	Asynchronous	Bi-directional	Used when it is difficult to use the service due to long response times after the request or when an intermediate feedback value is needed

TABLE 4-1 Comparison of the Topic, Server, and Action

## TOPIC



\*Topic not only allows 1:1 Publisher and Subscriber communication, but also supports 1:N, N:1 and N:N depending on the purpose.

FIGURE 4-2 Topic Message Communication

# SERVICE

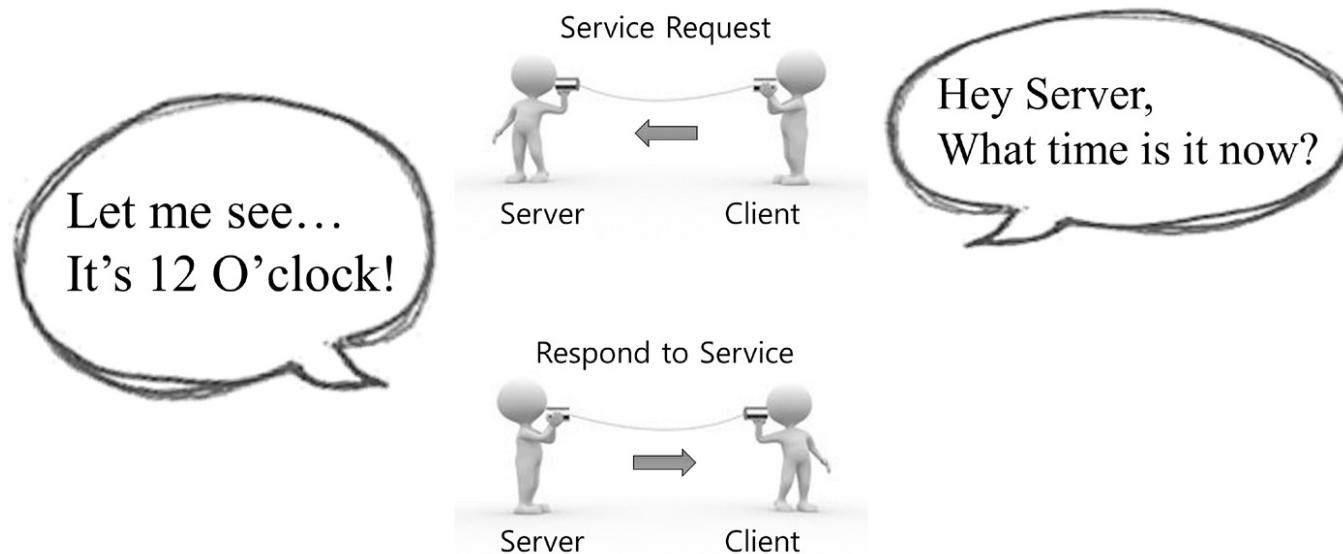


FIGURE 4-3 Service Message Communication

## ACTION

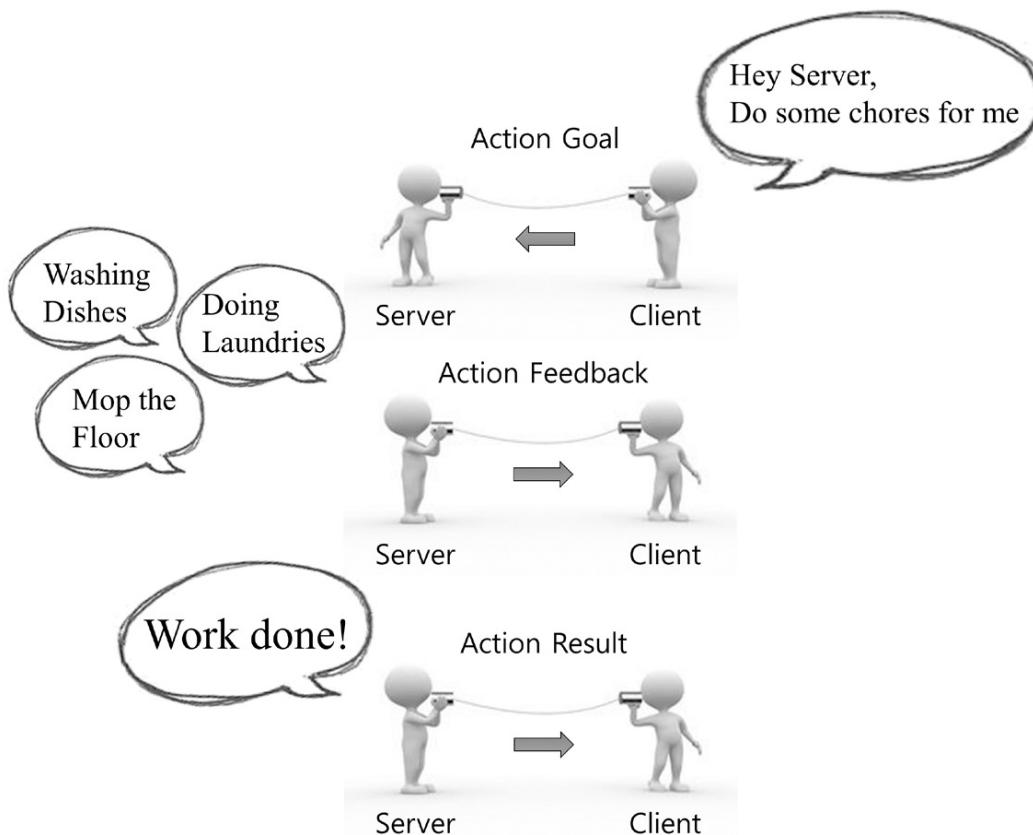


FIGURE 4-4 Action Message Communication

ROS FILE SYSTEM

---

HANDS-ON

## HANDS-ON ACTIVITY

- ▶ The instructor will explain the concepts of ROS computation graph through live demonstration
- ▶ Follow and make the same instructions
- ▶ Take note of the commands used

ROS FILE SYSTEM

---

# ASSIGNMENT

# ASSIGNMENT

- ▶ Open Udemy Website
- ▶ Respond to Assignment 1: ROS Filesystem and Ecosystem
- ▶ Submit your answer



# INTRODUCTION TO ROBOTICS (CS460)

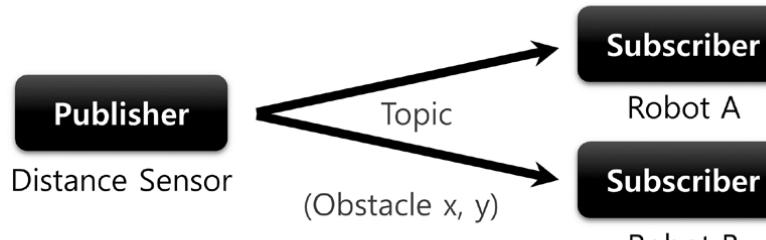
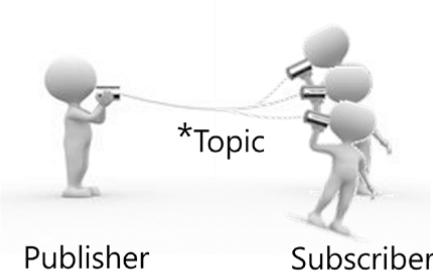
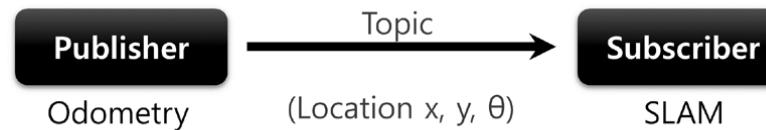
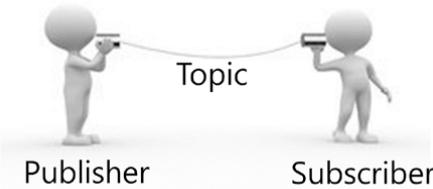
PROF. ANIS KOUBAA

---

## Understanding ROS Topics

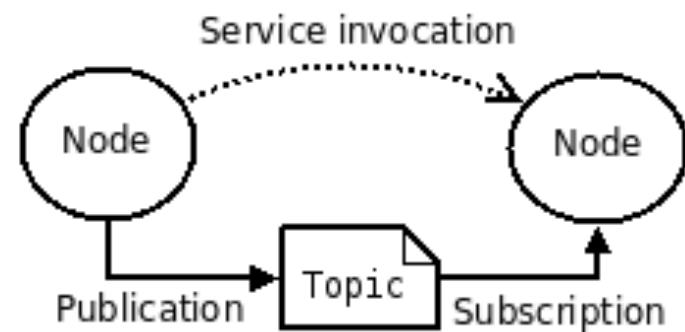
<https://www.udemy.com/user/anis-koubaa/>

# Topic

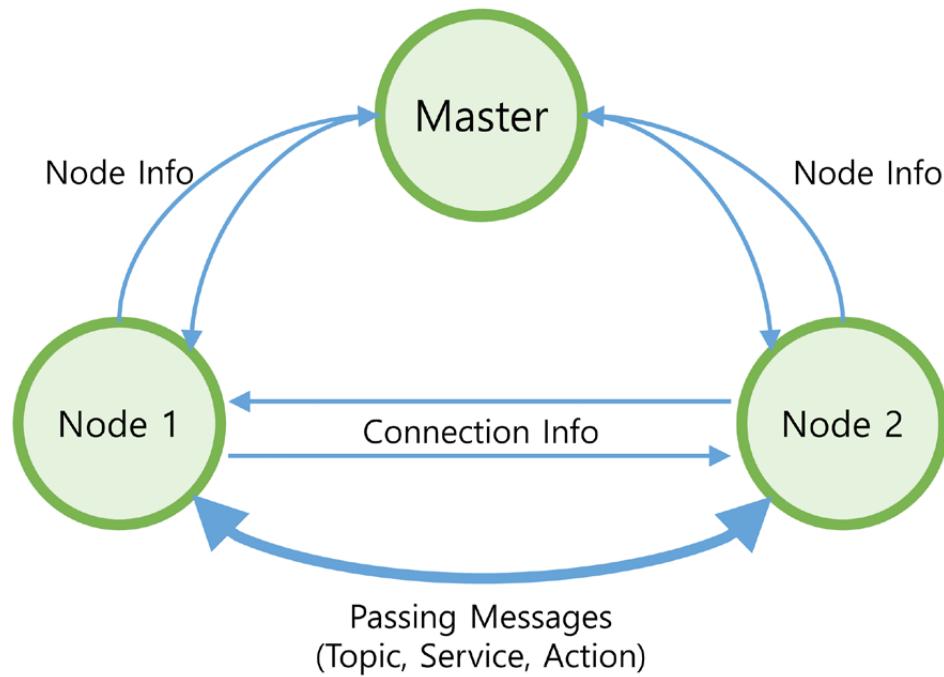


\*Topic not only allows 1:1 Publisher and Subscriber communication, but also supports 1:N, N:1 and N:N depending on the purpose.

Reference Book: **ROS Robot Programming** (in English).  
Authors: Yoonseok Pyo, Hancheol Cho, Leon Jung, Darby Lim



# Message Communication



Reference Book: **ROS Robot Programming** (in English).  
Authors: Yoonseok Pyo, Hancheol Cho, Leon Jung, Darby Lim

# 1: Run the Master ROS Node

```
$ roscore
```

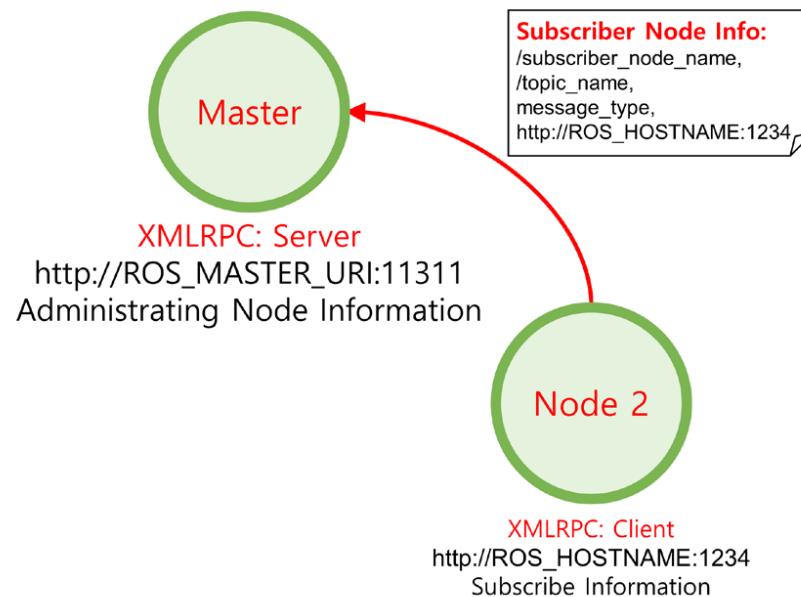


XMLRPC: Server  
[http://ROS\\_MASTER\\_URI:11311](http://ROS_MASTER_URI:11311)  
Administrating Node Information

Reference Book: **ROS Robot Programming** (in English).  
Authors: Yoonseok Pyo, Hancheol Cho, Leon Jung, Darby Lim

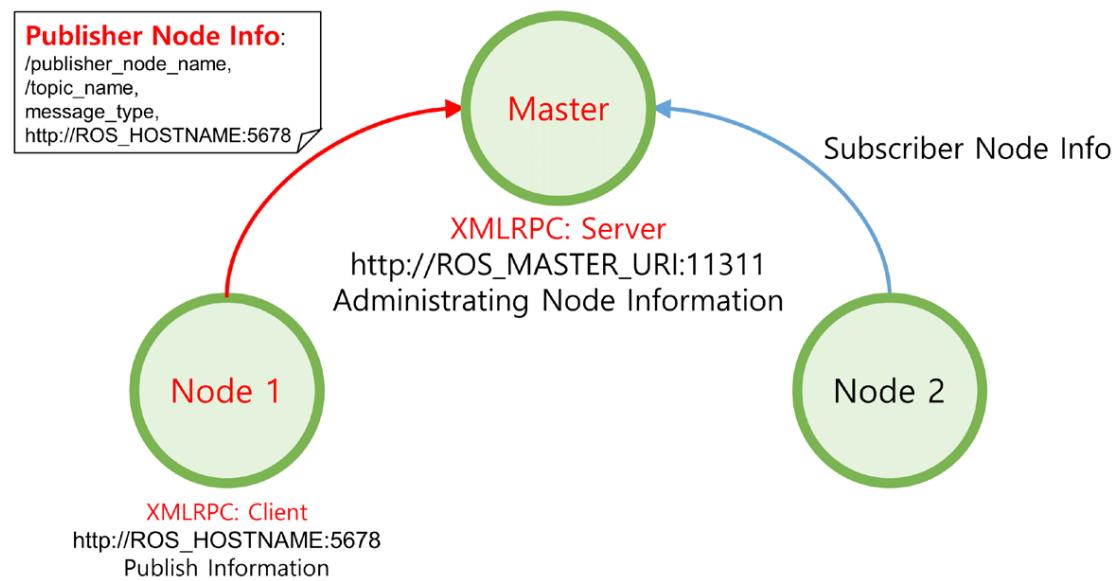
# 2: Running the Subscriber

```
$ rosrun PACKAGE_NAME NODE_NAME  
$ roslaunch PACKAGE_NAME LAUNCH_NAME
```



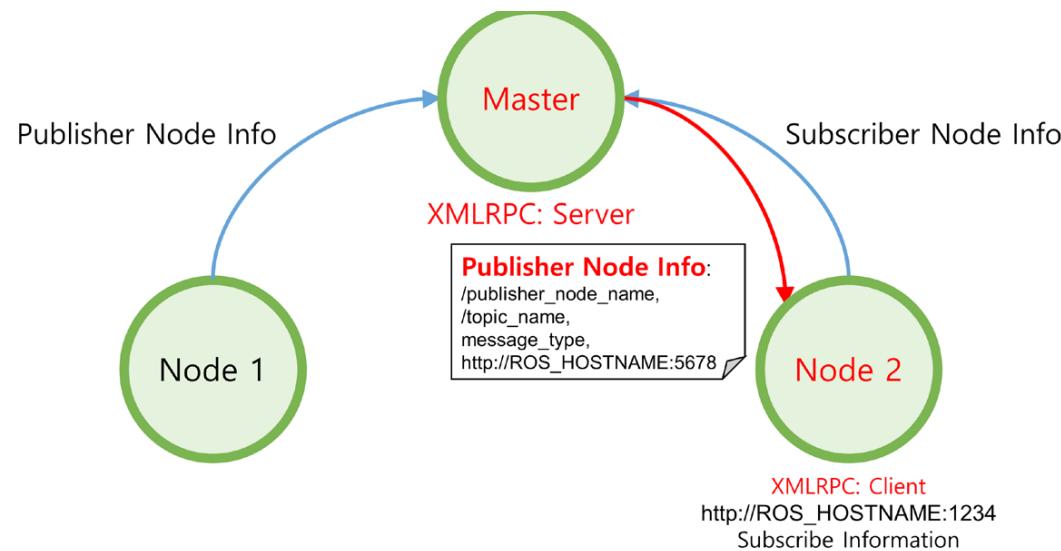
Reference Book: **ROS Robot Programming** (in English).  
Authors: Yoonseok Pyo, Hancheol Cho, Leon Jung, Darby Lim

# 3: Running the Publisher



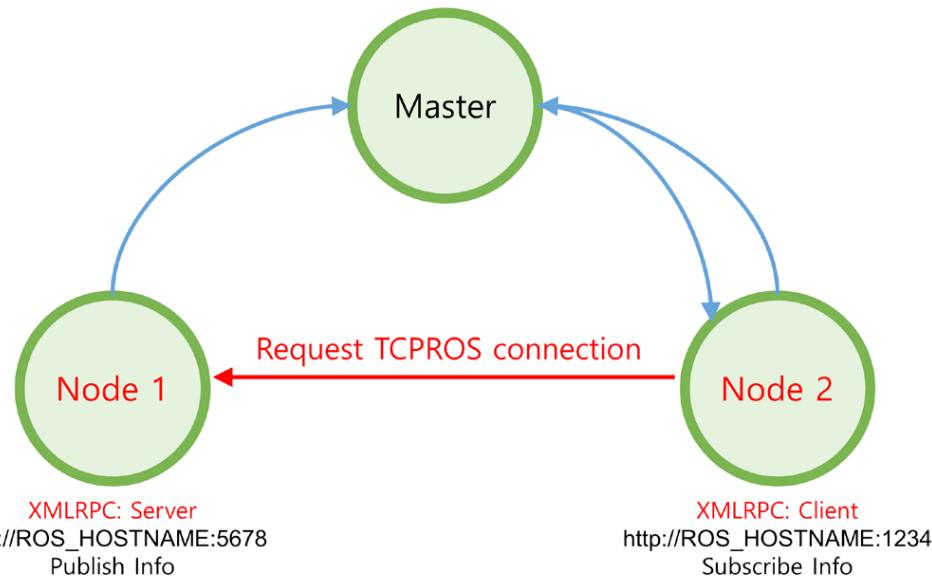
Reference Book: **ROS Robot Programming** (in English).  
Authors: Yoonseok Pyo, Hancheol Cho, Leon Jung, Darby Lim

# 4: Provide Publisher Info



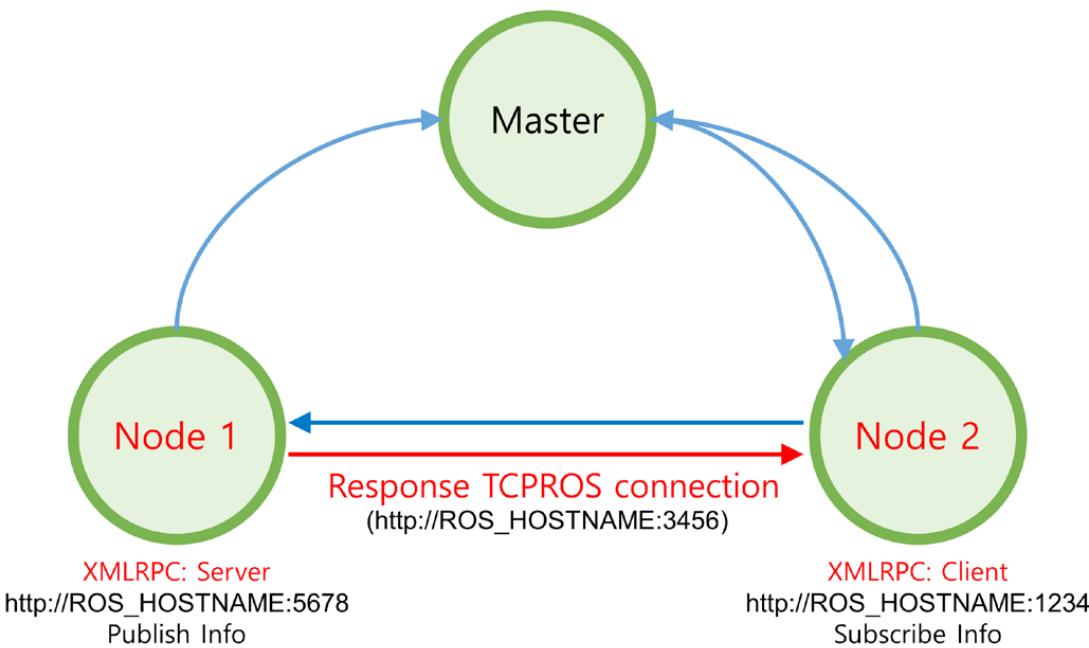
Reference Book: **ROS Robot Programming** (in English).  
Authors: Yoonseok Pyo, Hancheol Cho, Leon Jung, Darby Lim

# 5: Establish Connection Request



Reference Book: **ROS Robot Programming** (in English).  
Authors: Yoonseok Pyo, Hancheol Cho, Leon Jung, Darby Lim

# 6: Connection Response

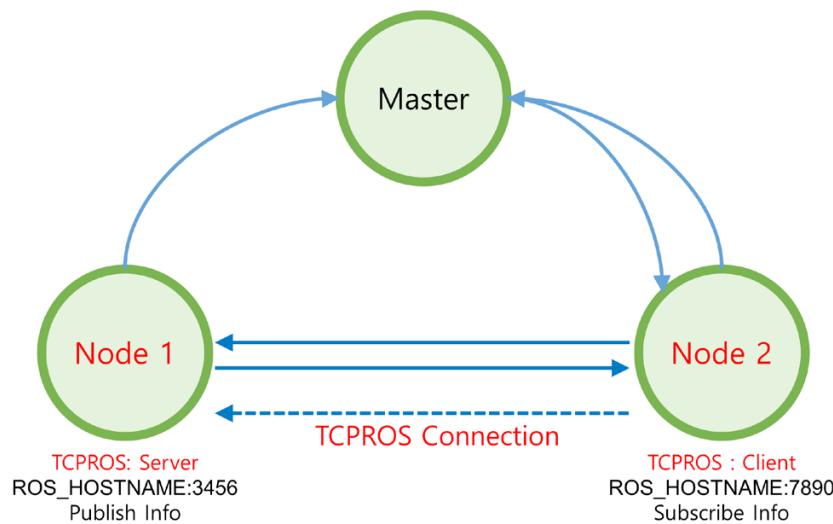


Reference Book: **ROS Robot Programming** (in English).  
Authors: Yoonseok Pyo, Hancheol Cho, Leon Jung, Darby Lim

# 7: TCP Connection

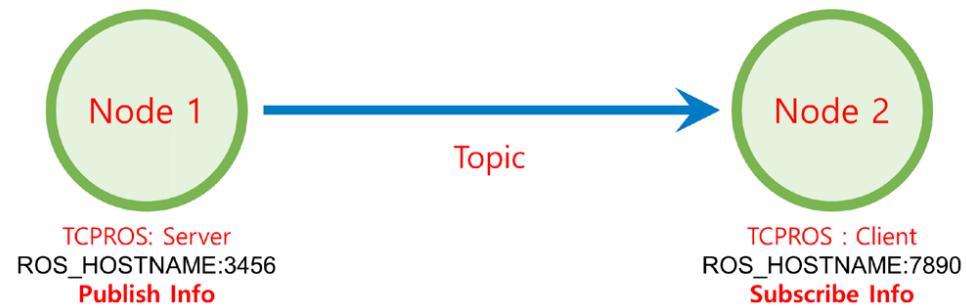
## TCPROS Connection

The subscriber node creates a client for the publisher node using TCPROS, and connects to the publisher node. At this point, the communication between nodes uses TCP/IP based protocol called TCPROS.



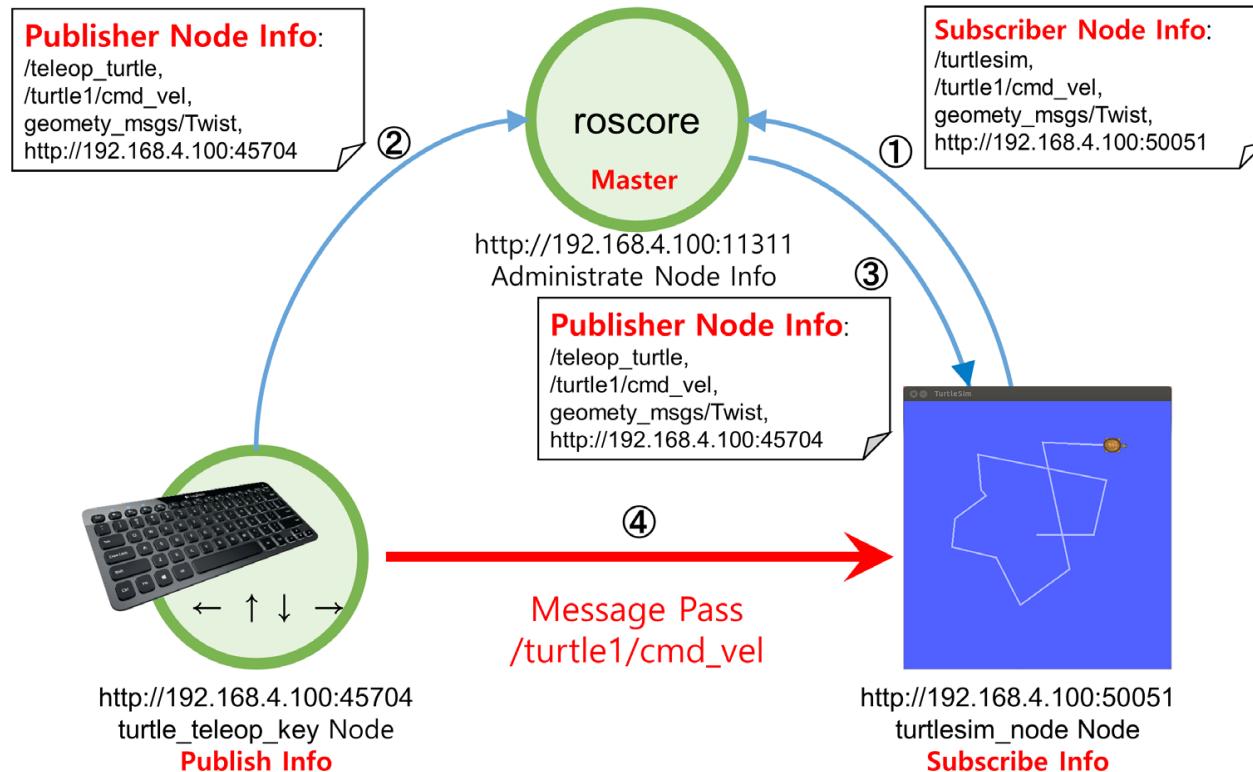
Reference Book: **ROS Robot Programming** (in English).  
Authors: Yoonseok Pyo, Hancheol Cho, Leon Jung, Darby Lim

# 8: Message Transmission



Reference Book: **ROS Robot Programming** (in English).  
Authors: Yoonseok Pyo, Hancheol Cho, Leon Jung, Darby Lim

# Example



Reference Book: **ROS Robot Programming** (in English).  
Authors: Yoonseok Pyo, Hancheol Cho, Leon Jung, Darby Lim

# Practical Tips to Write Publisher and Subscriber for ROS Topics

## ● Publisher

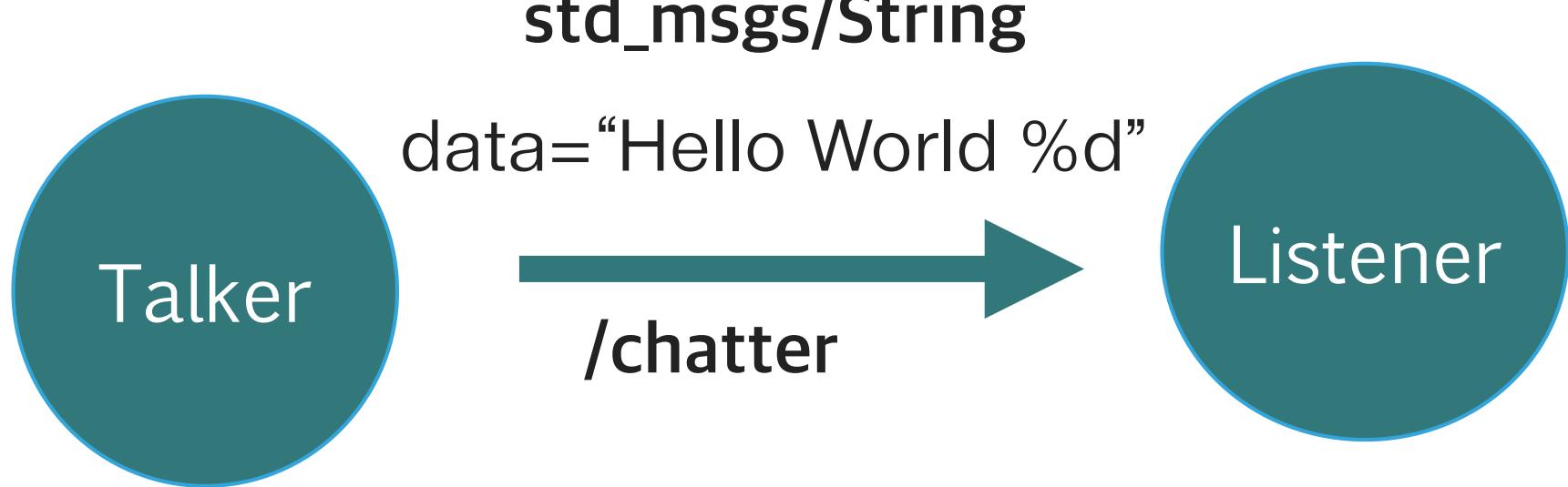
- **Step 1.** Determine a **name** for the topic to publish
- **Step 2.** Determine the **type** of the messages that the topic will publish
- **Step 3.** Determine the **frequency** of topic publication (how many message per second)
- **Step 4.** Create a publisher object with parameters chosen
- **Step 5.** Keep publishing the topic message at the selected frequency

# Practical Tips to Write Publisher and Subscribers for ROS Topics

## ● Subscriber

- **Step 1.** Identify the **name** for the topic to listen to
- **Step 2.** Identify the **type** of the messages to be received
- **Step 3.** Define a callback function that will be automatically executed when a new message is received on the topic
- **Step 4.** Start listening for the topic messages
- **Step 5.** Spin to listen for ever (in C++)

# Talker/Listener





# INTRODUCTION TO ROBOTICS (CS460)

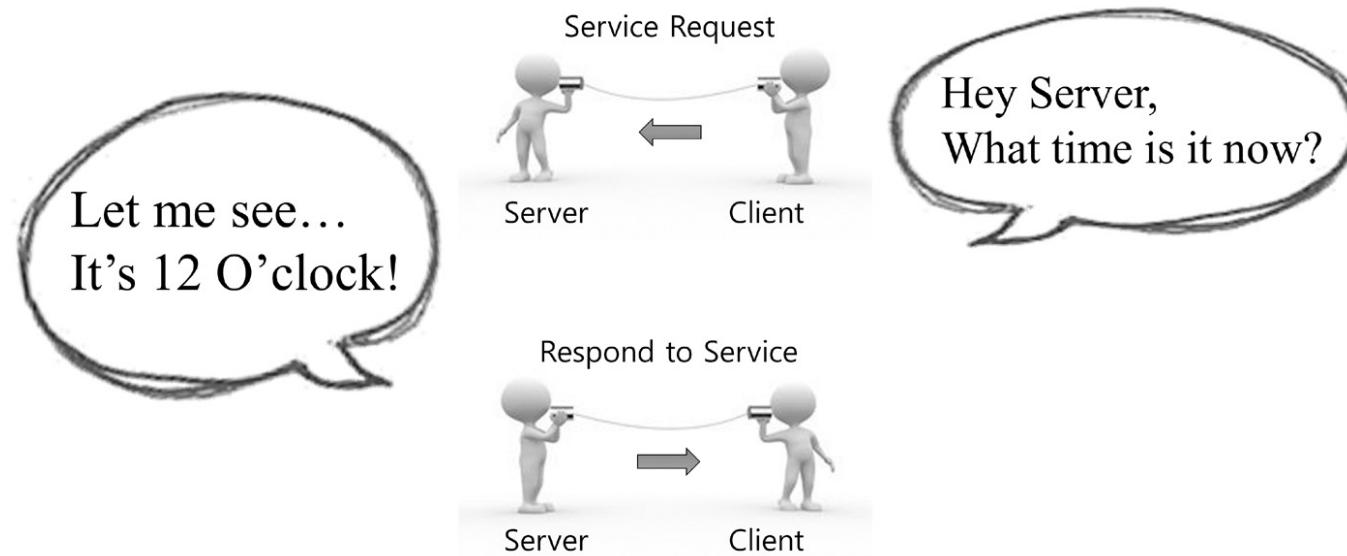
PROF. ANIS KOUBAA

---

## Understanding ROS Services

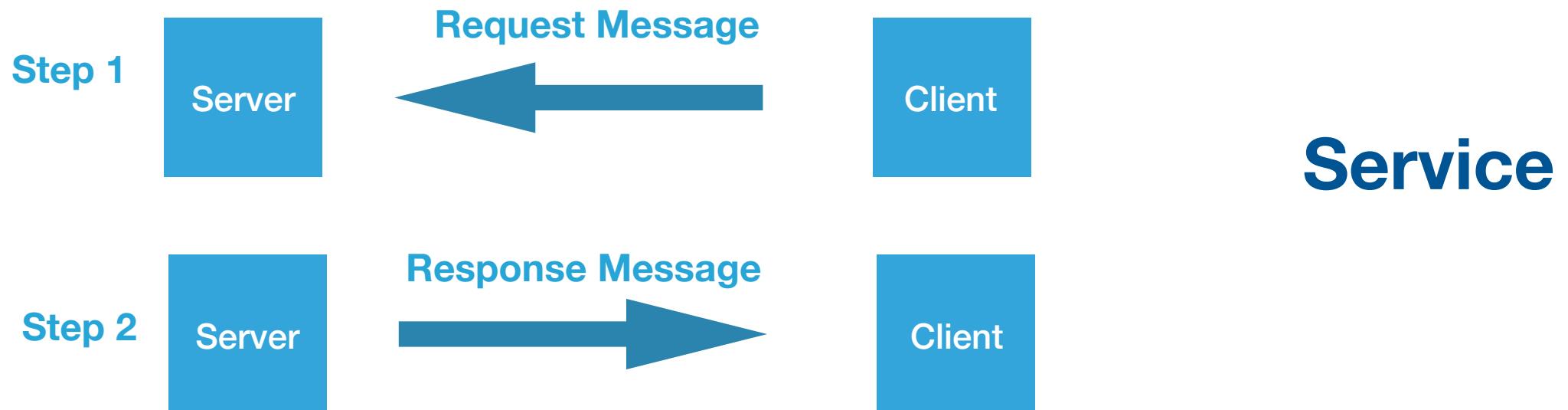
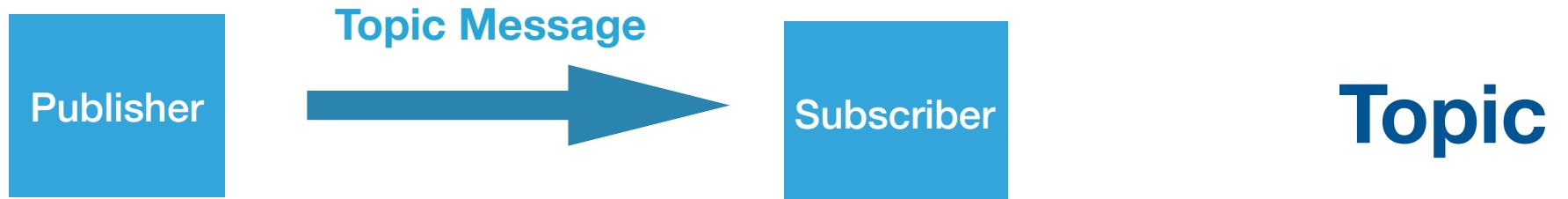
<https://www.udemy.com/user/anis-koubaa/>

# ROS SERVICES



Reference Book: **ROS Robot Programming** (in English).  
Authors: Yoonseok Pyo, Hancheol Cho, Leon Jung, Darby Lim

# ROS SERVICES



## STEPS TO CREATE A CLIENT/SERVER ROS SERVICE APP

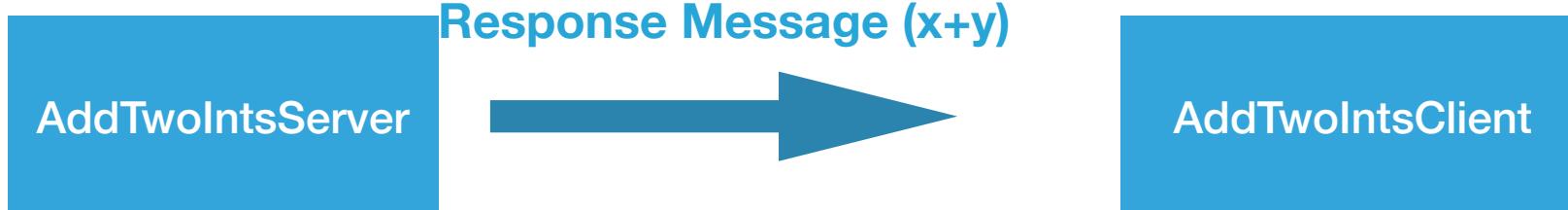
- ▶ Step 1. Define the service message (service file)
- ▶ Step 2. Create ROS Server node
- ▶ Step 3. Create ROS Client node
- ▶ Step 4. Execute the service
- ▶ Step 5. Consume the service by the client

# ROS SERVICE: ADDING TWO INTS

**Step 1**



**Step 2**



ROS FILE SYSTEM

---

HANDS-ON

## HANDS-ON ACTIVITY

- ▶ The instructor will explain how to write a ROS service in Python
- ▶ Then, you will implement a service for rectangle area.



UDEMY COURSE  
ROBOT OPERATING SYSTEM  
BASICS, MOTION, AND OPENCV

PROF. ANIS KOUBAA

---

RectangleArea ROS Services

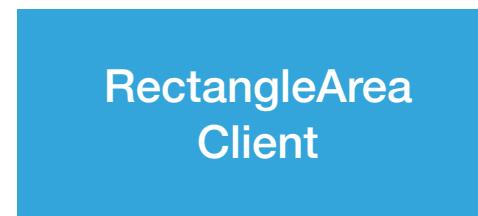
<https://www.udemy.com/user/anis-koubaa/>

# ROS SERVICE: ADDING TWO INTS

**Step 1**



**Request Message  
(width,length)**



**Response Message (area)**

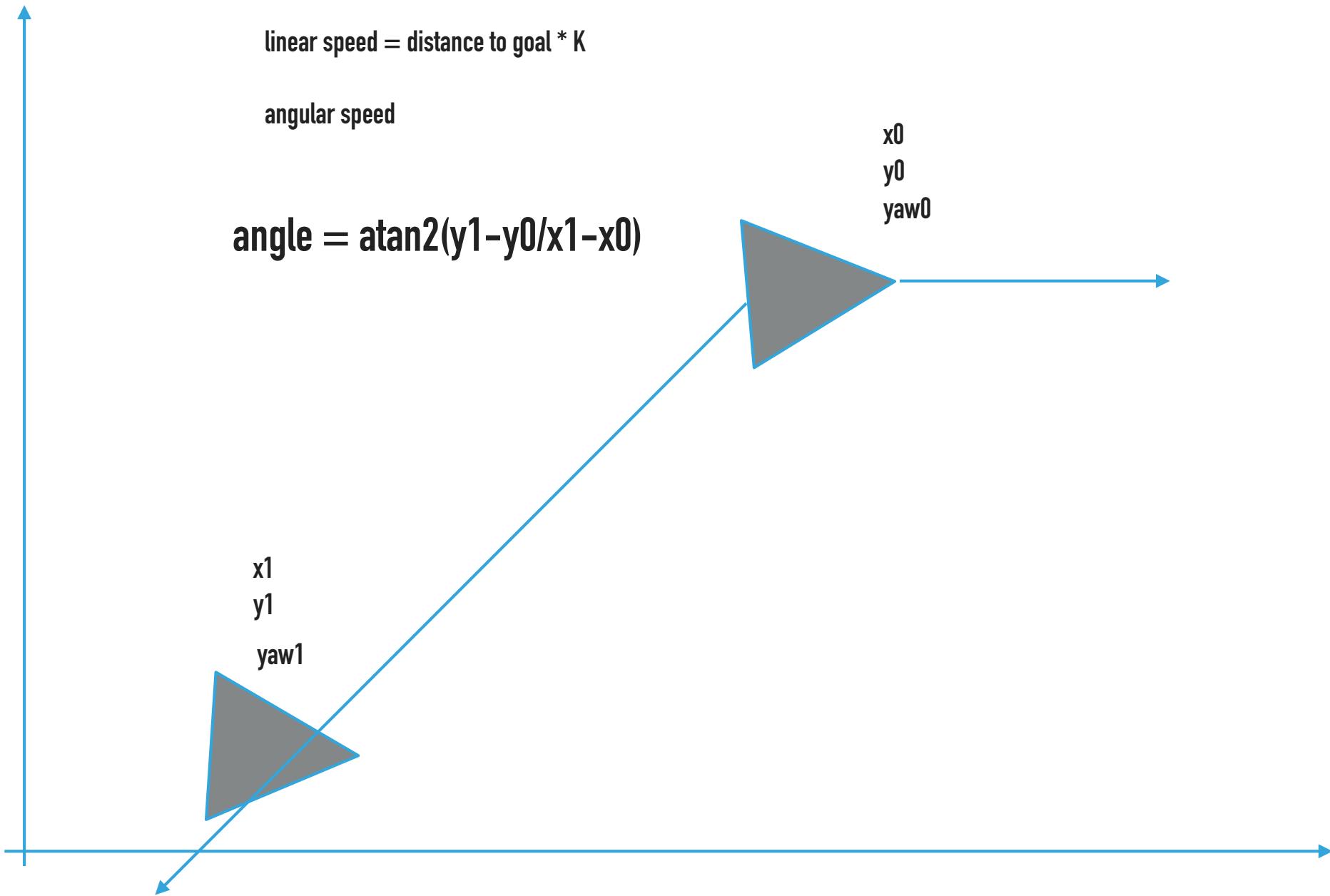
**Step 2**



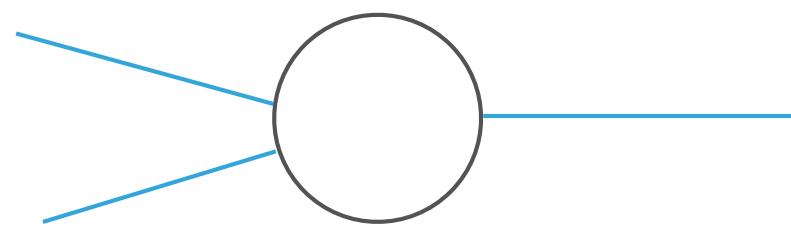
**RectangleArea  
Client**

# PID

---



---





UDEMY COURSE  
ROBOT OPERATING SYSTEM  
BASICS, MOTION, AND OPENCV

PROF. ANIS KOUBAA

---

Motion in ROS (Python)

<https://www.udemy.com/user/anis-koubaa/>



UDEMY COURSE  
ROBOT OPERATING SYSTEM  
BASICS, MOTION, AND OPENCV

PROF. ANIS KOUBAA

---

Go-To-Goal Motion in ROS  
(Python)

<https://www.udemy.com/user/anis-koubaa/>



UDEMY COURSE  
ROBOT OPERATING SYSTEM  
BASICS, MOTION, AND OPENCV

PROF. ANIS KOUBAA

---

Turtlebot 2 Robot Motion

<https://www.udemy.com/user/anis-koubaa/>



UDEMY COURSE  
ROBOT OPERATING SYSTEM  
BASICS, MOTION, AND OPENCV

PROF. ANIS KOUBAA

---

Launch File  
Including a Launch File and  
Parameters

<https://www.udemy.com/user/anis-koubaa/>

## WHAT IS A LAUNCH FILE?

- ▶ A **launch file** is XML document, which specifies:
  - ▶ which nodes to execute
  - ▶ their parameters
  - ▶ what other launch files to include
- ▶ **roslaunch** is a program that easily launches multiple ROS nodes
- ▶ Launch file has a .launch extension