UNIVERSITÀ
DI TRENTO
Department of
Information Engineering and Computer Science

Course Introduction

Edoardo Lamon

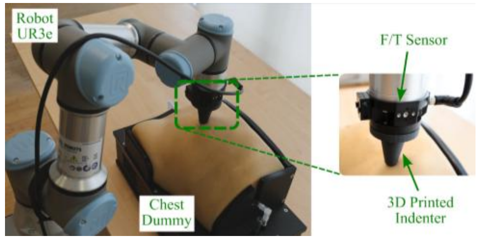
Software Development for Collaborative Robots

Academic Year 2025/26

About Me

- **Name**: Edoardo Lamon, PhD
- **Role**: Assistant Professor, DISI, Researcher at the Interdepartmental Robotics Labs (IDRA), Responsible of the IoT and Robotics Lab
- **Background**: PhD in Robotics (UniPi), researcher at HRII, IIT@Genoa
- **Research Interests**: human-robot interaction & collaboration, medical robotics, robot control & planning.
- **Contact & Office Hours**: edoardo.lamon@unitn.it, office 211, 2nd floor @Povo 1, available by appointment (email first)

IDRA
INTERDEPARTMENTAL
ROBOTICS LABS



The Course (SDCR)

- **Software development** is the process of designing, creating, testing, and maintaining different software applications.

  It requires:
  - high proficiency in programming;
  - effective collection and keen understanding of the requirements;
  - testing and documentation.

- **Collaborative robotics** is a form of robotic automation built to work safely alongside human workers in a shared, collaborative workspace.

  In most applications, the robot is responsible for repetitive, strenuous tasks, while a human worker completes more complex and thought-intensive tasks.

  Why to learn the principles of collaborative robotics?
  - Different robotic platforms (in this course we will use robotic arms)
  - Different (human-populated) environments: manufacturing, healthcare, logistics, agriculture

Scope of the Course (from syllabus)

- Develop a **collaborative robotics application** that includes interaction, motion, and task planning and execution;
- Integrate advanced programming aspects in C++ and Python to promote **code modularity** and **reusability**;
- **Manage** and **maintain software** in a **team** setting, including documentation, testing, continuous integration, and version control;
- **Present** and **highlight**, within a group, the distinctive features of their own work.

Technical Learning Objectives (from syllabus)

1. use the ROS 2 programming framework;
2. design and develop modular, well-documented, and thoroughly tested code;
3. apply advanced features of the C++ programming language in the context of ROS 2;
4. use the Python programming language within the ROS 2 framework.

Prerequisites (from syllabus)

- Basic knowledge of **C++** and **object-oriented programming**.
- Basic knowledge of **robotics** (transformations, kinematics, planning).
- Experience with **programming IDEs** and **build systems**.
- Basic knowledge of **mathematics** and **mechanics**.

Content (from syllabus)

- Fundamental concepts of collaborative robotics;
- Software development and simulation using ROS 2;
- Team-based software management with version control (git/GitHub);
- Software documentation, testing, and continuous integration;
- Presentation of the different project topics;
- Project development (laboratory activities).

NB: The program is WIP, as I will modify it according to the comments of last year students.

Teaching and Material (from syllabus)

Teaching Methods:

- Theoretical lectures, classroom exercises, project-related practical sessions

Requirements:

- Your own laptop, a physical partition or an emulation (Windows Subsystem for Linux/Docker/Virtual Machine) with Ubuntu 22.04

Material:

- Moodle (slides, registrations, exercises)
- Marco Matteo Bass: A Very Informal Journey Through ROS2: Patterns, Anti-patterns, Frameworks, and Best Practices
- B. Stroustrup: Programming Principles and Practices using C++ (Second Edition)

Test and Assessment

- **Deliverables**: GitHub project + presentation + half-course report;

- **Team-work**: groups of 2 people (max. 3 exceptionally);

- **Project content**: simulated collaborative robotic application (robot+human):
  - ➢ Theme selected by the group (healthcare, manufacturing, agriculture);
  - ➢ Specific topic according group-interest (you can re-use knowledge from other courses, like computer vision, AI, etc.), agreed with teacher.
  - ➢ Possibility to test on a real robots in lab setup.

- **Goal**: plenary presentation of the projects at the end of the course:
  - ➢ Project development within the course;
  - ➢ **Half-course report** on proposed project objectives and features, and draft individual contribution and methodologies;
  - ➢ **Presentation** also **peer-reviewed** (focus on clarity and impact).

Grading

The deliverables will be evaluated according to:

- Level of **functionality** reached and **performance**;
- **Modularity** and **re-usability** of the code (adoption of clever solutions is welcome)
- Quality of **development** and **documentation**:
  - ➢ Code (doxygen, sphinx, etc.)
  - ➢ Functionality (logic behind design choices and modularity)
  - ➢ Testing and continuous integration
- Quality of the **oral presentation**:
  - ➢ Slide design (images, videos, etc.)
  - ➢ Speaker in command of the presented topics;
  - ➢ Impactful demonstration of the project functionality;
- **Collaboration** between the **group members** (design awareness, using of tools for cooperative development of software)
- Testing the solution in **a real setup** (IoT and Robotics lab) is a recognized plus!