HANDS-ON INTERVENTION:
Vehicle-Manipulator Systems

The Project

Patryk Cieślak
patryk.cieslak@udg.edu

1. Introduction

The goal of the project is to design and implement a kinematic control system for a mobile manipulator, based on the Task-Priority redundancy resolution algorithm. The mobile manipulator under investigation is a differential drive robot (Kobuki Turtlebot 2), equipped with a 4 DOF manipulator (uFactory uArm Swift Pro). The manipulator is a typical case of a palletising robot. Moreover, the robot is equipped with a suite of sensors including wheel encoders, an RGBD camera and a 2D lidar. The manipulator is equipped with a vacuum gripper.

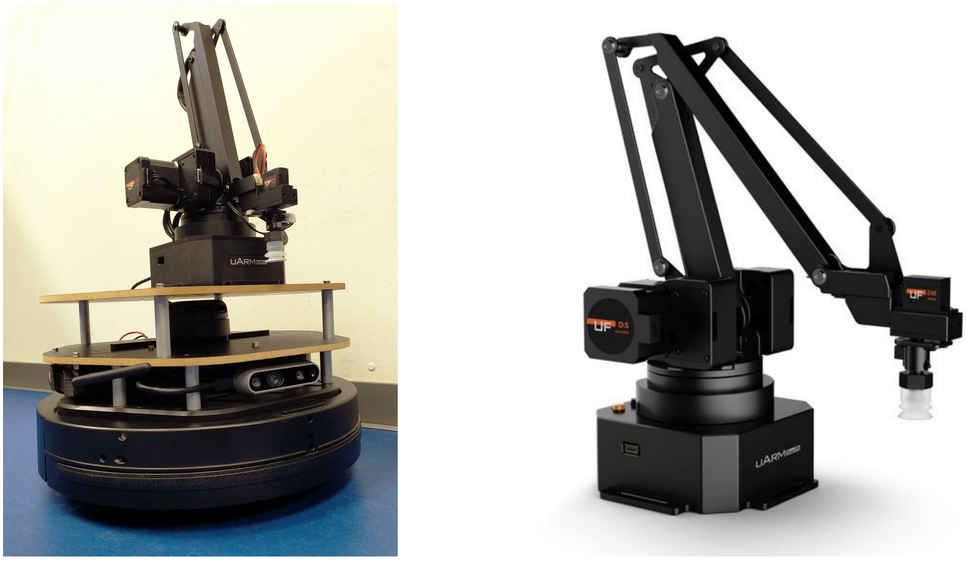

Fig 1. The experimental platform.

2. Methodology

The project is divided into four stages: the conceptual design phase, the base functionality implementation phase, supported by throughout simulation, the experimental validation phase, and the integration phase. A simulation environment was prepared by the tutor, which includes the full model of the real robotic platform, and a fully compliant software architecture. All developments will be tested thoroughly in the simulation and the successful implementation will be validated experimentally. Thanks to the fact that the simulation and the real platform follow the same interface, it will be possible to execute the same code on both of them. This will allow the students to develop virtual prototyping and hardware-in-the-loop (HIL) simulation skills.

2.1. Conceptual design phase (20 pts)

In this phase of the project students will design the solution based on the presented specification. It is also a time for deep understanding of the functioning of the robot's software and hardware architecture, to be able to discover the possibilities and the necessary developments. The following outputs are expected from each group:

	Task description	Pts
A	A high level block diagram of robot hardware architecture, including: the mechanical structure, the actuators, the sensors, and the compute units.	2
B	A detailed block diagram of the software architecture used in the robot simulation setup.	4
C	A detailed functional block diagram of the designed kinematic control system, including: all necessary components of the control system, interfacing with robot architecture, inputs, outputs, message types.	8
D	Symbolic derivation of the forward and inverse kinematics of the experimental system.	6

All of the block diagrams have to be prepared in an easily editable form to allow for discussion and correction by the teacher, e.g., using diagrams.net or gitmind.com. The symbolic derivations have to be documented using word processor's formula editor or LaTeX.

2.2. Implementation/simulation phase (50 pts)

In this phase of the project students will build the necessary software architecture to complete the tasks specified by the teacher. The following outputs are expected from each group:

	Task description	Pts
A	A detailed implementation block diagram for the designed kinematic control system, including: all implemented nodes, classes, fields, functions, ROS messages and services (UML or similar).	5
B	Implementation of the Task-Priority redundancy resolution algorithm for the experimental platform.	5
C	Implementation of the following tasks: end-effector position, end-effector orientation, end-effector configuration, base orientation, joint position, joint limits.	5
D	Implementation of a sequence of motions between desired Cartesian end-effector configurations.	5
E	Implementation of a sequence of motions necessary to pick an object with the vacuum gripper.	5
F	Implementation of a sequence of motions to place the picked object on the robot platform.	5
G	Implementation of a pick, transport and place operation, using a state machine, for a priori known pick and place locations, using dead reckoning for navigation.	10
H	Implementation of a pick, transport and place operation, using a state machine, for a priori known approximate location of pick and place locations, utilising visual feedback from ARUCO marker detection and dead reckoning for navigation.	10
I	Presentation of the software operation to the teacher, to obtain approval for experimental trials.	0

All of the implementation must be contained in one ROS package for easy testing by the teacher and transfer to the workspace of the experimental platform. Use of GIT version control system (code repository) is encouraged for safety and simple management of files.

2.3. Experimental validation phase (30 pts)

In this phase of the project, students will perform an experimental validation of the software developments and compare the results with the simulation. The following outputs are expected from each group:

	Task description	Pts
A	Transfer of the code to the experimental platform and successful compilation and execution of the programs. Tuning of gains, velocity limits and other algorithm parameters.	5
B	Execution of experimental trials representing the tasks described in 2.2.D to 2.2.H, resulting in a set of *rosbag* files and videos.	10
C	Detailed report on the functioning of the robot, based on the recorded and processed data, including, but not limited to: • Description of each of the performed tasks, • Plots presenting motion of the mobile base and the end-effector in the X-Y plane, • Plots presenting evolution of control errors over time, • Plots presenting evolution of joint velocities over time, • Discussion of results: performance, problems, comparison with the simulation.	15

In case of not being able to reach the experimental phase of the project, students are required to prepare the report based on data recorded using the simulation environment.

2.4. Integration phase

This phase of the project is not mandatory and can only be entered upon successful completion of the previous phases. The goal of this phase is to integrate the developed control system with technologies developed in other hands-on subjects, including navigation, perception and planning. The integration will be presented at a special session.

All developments of integration are highly encouraged but not mandatory.

3. Evaluation

The evaluation of the project consists of two elements: the evaluation of the design, implementation and testing of the project and the evaluation of the oral presentation, including questions.

3.1. Project development

The project development evaluation is following the methodology presented in Section 2. The complete mark of the project will be calculated in **90%** from **the hands-on intervention** subject and in **10%** from **the integration phase**. The maximum number of points possible to achieve in the HOI part is **100**, which corresponds to the highest mark. Number of points assigned to each of the tasks was specified in Section 2, together with the expected deliverables. During the continuous evaluation of the project, **the teacher will also assess the engagement of each person in the group, affecting her/his personal HOI mark.**

Project mark = 90% * HOI mark + 10% * Integration mark

3.2. Oral presentation

The oral presentation is an integral part of the project, allowing the tutor to assess the understanding of the developments by all of the students inside each project group. The presentation of the project developments will be performed during a special session. Each of the students will be taking active part in the presentation by performing a rotation of speakers, for each of the tasks specified in Section 2. Moreover, the **teacher will ask questions directed towards specific students** to further evaluate their understanding and efforts related to the project. Therefore, **marks related to the oral presentation will be individual and can differ inside the group**.