Tallinn University of Technology
School of Information Technologies

Robotics (IAS0060)

Extended syllabus
Spring 2026

Course aims/objectives:	The aim of this course is to learn to solve mobile robotics problems including sensor signal processing, actuator control and robot control tasks.
Learning outcomes:	The student has demonstrated the ability to: • solve robotics tasks of moderate complexity by processing and fusing sensor data and controlling a mobile robot; • use the Robot Operating System (ROS) and/or embedded systems for building a robot control architecture; • solve robotics problems both in simulations and in real-world environment, and understand the advantages and limits of both environments; • manage teamwork for solving complex robotics problems; • clearly document and present their work to others; • examine and evaluate advantages and limits of typical hardware and software solutions; • choose the most suitable technical approaches for solving robotics tasks and substantiate those decisions.
Brief description of the course (topics):	The course is project-based. During the semester, the teams consisting of 2-3 students solve a task comprising decision making, robot control and data fusion. Building hardware is not required. Teams will give a weekly overview of their progress and discuss their activities with the supervisor and other students. During the week the students work at flexible schedule on their task as agreed with the supervisor of the work group. The maximum number of course participants is 20. The participants are required to have C++ or Python programming skills and a basic knowledge of ROS/ROS2 would be beneficial. At the beginning of the semester every student takes an entrance test (C++ or Python programming) that determines if the student is allowed to take the course. The students are divided into workgroups based on their background and preferences. Usually, there is a choice between 3 different course projects (https://taltech.ee/en/biorobotics-courses#p1441).
Language of the course:	English
ECTS credits:	6 ECTS
Students:	This course is addressed mainly to Master students of Information Technology and Mechatronics.
Special needs:	Persons with disabilities can participate in this course. Please inform the professor(s) in the beginning of the course of any special instruction, or assessments of this course that may be necessary to enable you to fully participate in this course.
Registration:	Students who would like to take the course should declare the course in the ÕIS (Student Information System) by deadlines set in the academic calendar.
Prerequisite courses and/or knowledge:	The participants are required to have C++ or Python programming skills and a basic knowledge of ROS\ROS2 would be beneficial
Prerequisite resources:	ROS and ROS-based simulators are used. Those are provided during the course.

Professor(s):	Maarja Kruusmaa, PhD, Tenured Full Professor, Maarja.kruusmaa@taltech.ee.
Contacting Professor(s):	Preferred means of contact: email
Schedule for classes:	Thursday (10:00 to 13:15)
Study process description:	Each week the students will: • Attend weekly meeting and lecture. • Submit previous assignment. • Start working on new assignment.
Course's e-support:	Course materials can be accessed via the e-learning environment Moodle under the course title "IAS0060 Robotics (2025/26 Spring)" https://moodle.taltech.ee/course/view.php?id=36405
Study literature:	• "Robotics – Modelling, Planning and Control", B. Siciliano; L. Sciavicco; L. Villani, G. Oriolo • "(Introduction to) Autonomous Mobile Robots", R. Siegwart and I. R. Nourbakhsh • "Springer Handbook of Robotics", B. Siciliano and O. Khatib
Continuous assessment:	The continuous assessment consists of weekly assignments which contribute 80 % to the final grade. The assignments are divided into: • Weekly assignment (70% / group grade) • Lab exercises (30% / individual grades) • Weekly progress reports (not graded)
Evaluation criteria for continuous assessment:	Each group submits one solution for the weekly assignment. The evaluation is based on the level of achievement of task, documentation, code clarity and reporting. Throughout the semester, groups need to do lab exercises. Here they transfer the work, which they developed in simulations, to a real robot. Lab exercise evaluation is individual and is based on level of preparation and quality of lab work (independent problem solving, level of achievement of task, engagement).
Exam:	The final exam contributes 20% to the final grade and takes the form of an oral presentation and defence of the groups work
Evaluation criteria for the exam:	50 % of the presentation grade is based on presentation content (i.e., how well the group presents their progress on the project) and is assigned to everyone in the group equally, while the remaining 50% of the presentation grade is assigned individually based on presentation performance, answers to individual questions and the students' knowledge about their finished project.
Final grade:	From 5 (maximum) to 0 (minimum) The sum of points for each weekly assignment, the lab exercises and the final presentation is converted into a grade using the following principles: "5" excellent 91-100 "4" very good 81-90 "3" good 71-80 "2" satisfactory 61-70 "1" poor 51-60 "0" fail less than 51
Academic integrity:	As a student at Tallinn University of Technology, you have an obligation to conduct your academic work with honesty and integrity according to university standards. It is expected that all work that you submit will be your own, and that you have actually done the work that you are submitting. Plagiarism and cheating will not be tolerated. Should you be found to be guilty of such activities, it will be followed with grade "0" for the assignment/exam and a notice will be filed to the School's Committee for

Handling Violations of Academic Practice and Contemptible Behaviour. Depending on the Committee's proposal, it may lead to Dean issuing a letter of reprimand or in case of repeated or very severe misconduct, exmatriculation from the University.

Detailed schedule and topics

The semester plan is preliminary and might be changed in case of cancellations, changes in available reading material, etc.
Weekly meetings and lectures take place in person in room ICT-315 (and on MS Teams if requested).
Weekly assignments are due on Wednesday of the following week.

Project 2 (μ-CAT) includes weekly visits to the Centre for Biorobotics for final testing of each assignment.
Projects 1 (TurtleBot) and 3 (U-CAT) include visits to the Centre for Biorobotics at specific weeks for final testing of assignments.

Week 1:

Course introduction and entrance test.

The test is available on February 6th and expires 90 minutes after it is started. Results to the test are announced after the final deadline passes.

Week 2:

Setting up the software framework for the class.

Students set up Ubuntu 22.04 and ROS Humble, as well as simulators and other supplemental packages that will be used throughout the semester.

Week 3:

Simple control

Introduction to PID control. First contact with the specialized simulators. The students get familiarized with their robots and simulators by investigating closed-loop control for different functions of their systems.

Week 4:

Mapping / Motion Gaits / Object Detection

Project 1 (TurtleBot) groups work on mapping for mobile robots, including learning about world models and coordinate transformations. The students will prepare an overview of mapping and localization methods.

Project 2 (μ-CAT) groups work on the derivation and implementation of the robot's kinematics in simulation and on the robot itself.

Project 3 (U-CAT) groups work on object detection, including learning about available algorithms and implementing object tracking in a simulated environment.

Week 5:

Mapping / Depth Control / Visual Tracking

Project 1 (TurtleBot) groups implement a mapping algorithm of their choice and test it in a simulation framework using a virtual robot including a laser-range finder as primary sensor.

Project 2 (μ-CAT) groups work on implementing pressure sensor feedback and a simplified closed loop depth controller of the robot.
Project 3 (U-CAT) groups work on object detection, including learning about available algorithms and implementing object tracking in a simulated environment.

Week 6:

Mapping / Control methods / Visual Tracking

Project 1 (TurtleBot) groups continue and finish their work on the mapping algorithm.

Project 2 (μ-CAT) groups learn about different control strategies for mobile robots.

Project 3 (U-CAT) groups work on tracking a static object underwater using computer vision.

Week 7:

Mapping / Depth Control (PID) / Moving Object Tracking

Project 1 (TurtleBot) groups implement their mapping algorithm on the real robot and test it in a lab exercise.

Project 2 (μ-CAT) groups work on the derivation and implementation of discretized PID variants for depth control.

Project 3 (U-CAT) groups work on tracking a moving object underwater using computer vision.

Week 8:

Localization / Depth-Surge Control (Fuzzy) / Fuzzy Logic Control

Project 1 (TurtleBot) groups implement a localization algorithm of their choice and test it in a simulation framework using a virtual robot and the map they have created in week 6.

Project 2 (μ-CAT) groups work on augmenting the robot's control system with a Fuzzy logic - PID combination for control of multiple degrees of freedom.

Project 3 (U-CAT) groups work on designing and implementing a Fuzzy logic controller.

Week 9:

Localization / Orientation Control / Acoustic Detection

Project 1 (TurtleBot) groups continue and finish their work on the localization algorithm.

Project 2 (μ-CAT) groups learn about rotation formalisms, orientation sensing, and implement yaw control for the robot.

Project 3 (U-CAT) groups work on tracking a static object underwater relying on acoustic signals.

Week 10:

Path Planning / Signal Filtering / Target Motion Estimation

Project 1 (TurtleBot) groups implement their localization algorithm on the real robot and test it in a lab exercise. Additionally, the groups work on an overview for path planning and obstacle avoidance algorithms.

Project 2 (μ-CAT) groups learn about signal filtering and implement various filtering algorithms for noise reduction.

Project 3 (U-CAT) groups learn about Kalman Filters and implement a motion estimator to predict the target's position.

Week 11:

Path Planning / Position Control / Object Detection with Data Fusion

Project 1 (TurtleBot) groups implement a path planning algorithm of their choice and test it in a simulation framework using a virtual robot and their map from week 6 and their localization from week 9.

Project 2 (µ-CAT) groups learn about pose control and implement a position/heading controller in simulation.

Project 3 (U-CAT) groups implement a Kalman filter to fuse acoustic and visual signals.

Week 12:

Path Planning / Position Control / Experiments with U-CAT

Project 1 (TurtleBot) groups implement their path planning algorithm on the real robot and test it in a lab exercise.

Project 2 (µ-CAT) groups implement a vision-based tracking algorithm for tracking the robot's pose and finish their work on position control with the physical prototype.

Project 3 (U-CAT) groups implement their object tracking control scheme on U-CAT and test it in field experiments.