Title:	RObot MANipulation of Deformables through dynamIC actions
Acronym:	**ROMANDIC**
Type of Action:	HORIZON Coordination and Support Actions
Grant Agreement No.:	101159522
Starting Date:	01-10-2024
Ending Date:	30-09-2027

ROMANDIC

Robot Manipulation of Deformables through Dynamic Actions

Deliverable Number:	D2.2
Deliverable Title:	Curricula for a new PhD course
Type:	Report
Dissemination Level:	Public
Authors:	Andrej Gams, Mihael Simonič, Aleš Ude
Contributing Partners:	JSI

Estimated Date of Delivery to the EC: 31-12-2025
Actual Date of Delivery to the EC: 12-01-2026

Contents

Executive summary

Deliverable **D2.2** deals with the proposal of a new PhD course.

Section 1 details the motivation for the content of the course, and its design choices are discussed in Section 2. Since JSI by itself does not provide courses to students, but acts as a base for research for many institutions, the relations and administrative background are provided in Section 3. The content of the course is given in 4.

The course provides theoretical background and foundational and practical knowledge for developing robotic applications using ROS 2. Participants will learn Linux basics relevant to robotics development, master ROS 2 core concepts (nodes, messaging, services), and learn how to use containers to build, deploy and manage reproducible ROS 2 environments for development and distribution. By the end of the course, students will be able to build, run, and manage ROS 2-based robotics applications both on local machines and in containers. The course was designed based on the identified needs, observed lack of knowledge of students, and on the basis of similar courses worldwide.

Attached are the formal application for the new course and the cover letter.

1 Motivation

The rapid evolution of robotics research and its increasing integration into real-world environments demand a new generation of researchers who are not only theoretically proficient, but also capable of efficiently developing, deploying, and evaluating complex robotic systems. Within **ROMANDIC**, which addresses dynamic manipulation of deformable objects with robots, robust, modular, and reproducible and transferable robotic solutions are needed. The proposed PhD course Developing and Programming Robotic Applications directly addresses a critical skills gap observed in doctoral education.

Through several years of teaching experience at the doctoral level and within engineering programs, particularly through recent work with students at the Jožef Stefan International Postgraduate School (IPS - MPŠ in Slovene) and the Faculty of Electrical Engineering, University of Ljubljana, it has become evident that many doctoral students lack systematic knowledge of modern robotics software infrastructures. While numerous powerful tools, frameworks, and open-source solutions are readily available online (e.g., ROS and ROS 2 packages, simulation frameworks, perception and planning libraries, open-source control toolkits), their effective, reliable, and scientifically sound use requires a deeper understanding of software architectures, communication paradigms, timing constraints, and deployment practices. Without such knowledge, students are often limited to ad-hoc implementations and face difficulties with scalability, reproducibility, and transferability of solutions between platforms and from simulation to real robotic systems.

Component-based and modular software architectures, distributed systems, and research-oriented development practices, with the Robot Operating System (ROS 2) serving as a unifying framework, are crucial for modern robotics development. ROS 2 has effectively become a de facto standard in both academic and industrial robotics. By mastering core ROS 2 concepts such as nodes, topics, services, and actions, doctoral students gain the ability to rapidly construct functional robotic applications that provide a solid baseline for advanced, hypothesis-driven research.

A key motivation for this course is its strong emphasis on reproducibility and scientific rigor. Infrastructure choices, including middleware, communication models, development environments, and deployment strategies, directly influence the validity and repeatability of experimental results. By introducing containerization technologies (e.g., Docker) alongside ROS 2, the course enables students to create reproducible development and execution environments, thereby facilitating collaboration across institutions and supporting open science principles promoted by EU research initiatives. This aspect is particularly relevant for ROMANDIC, where shared experimental platforms and comparable research methodologies are essential for effective twinning activities, i.e., secondments.

The course also bridges the gap between simulation and real-world deployment. Through the use of simulation tools such as Gazebo, standardized robot descriptions, and peripheral device integration, students learn how to transition seamlessly from simulated experiments to physical robotic systems. This capability is crucial for doctoral research, where rapid prototyping, iterative experimentation, and validation on real hardware are central to scientific progress. Similar doctoral-level courses focusing on robotics software architectures and ROS-based application development are already established at leading European institutions, such as ETH Zurich (ETHZ) and the Universitat Polit‘ecnica de Catalunya (UPC). Aligning the proposed course with such internationally recognized curricula further strengthens compatibility with partner institutions and facilitates student and staff mobility within the ROMANDIC framework.

From a strategic perspective, the course contributes directly to multiple ROMANDIC work packages, most notably capacity building and research collaboration, by establishing a shared technical and methodological foundation with international partner institutions. Moreover, by relying on widely adopted tools and portable development environments, the course facilitates increased inward and outward mobility of qualified researchers and encourages creative new approaches to research and innovation collaboration.

In summary, the aim of the proposed course is to equip PhD students with foundational and practical competencies that are indispensable for contemporary robotics research. It empowers them to effectively leverage existing tools and methodologies, rapidly reach a functional application level, and subsequently develop novel, high-impact research contributions on top of robust and reproducible software infrastructures. As such, the course represents a key educational and strategic instrument for achieving the scientific and institutional objectives of the ROMANDIC project.

2 Examples of Similar Courses

As mentioned in the introduction, similar courses are listed in other institutions.

2.1 From project partners

CSIC from Barcelona is associated with the Universitat Polit‘ecnica de Catalunya (UPC), where its students study. Their course on ROS 2 was used to partially model our proposed course. The website for their course, titled *Introduction to ROS 2*, says the following:

> The objective of this course is to introduce students to the use of ROS 2 as a powerful robotics tool. Specifically, a familiarization with the middleware concept and the software structure of a robotic system. There will be a special emphasis on sensing and control of robotic systems using ROS 2, both in simulation and in real environments.
>
> The teaching methodology will combine lectures together with supervised exercises based on the current ROS version and tools. All classes will be organized with the theoretical sessions at the beginning and practical exercises and team work at the end. The initial part will consist on the explanation of theoretical concepts by the lecturer, promoting the active participation of students. The practical part will be focused on the student's solving skills. The main theoretical concepts will be shown in practical simulation examples and finally on a real robot test.
>
> The source code of the examples of these tutorials is organized in packages stored in the GIT repository. The course is divided in 7 tutorials and some appendices.

A screenshot of their course introduction is presented in Figure 1.

KIT offers many courses on the topic or robotics, programming and computer science in general, making KIT one of the most prestigious engineering institutions in Germany and the world.

One course dealing with similar topics is Design and Architectures for Embedded Systems (ESII). However, robotics is a complete study module at KIT (lead by **ROMANDIC**'s Prof. Asfour), with a diverse list of courses (Robotics at KIT). Specifically, until recently, two course were obligatory in this study module, namely Robotik I - Einführung in die Robotik and Echtzeitsysteme (real-time systems).

While **JSI** is an institute and does not offer courses, its Jožef Stefan International Postgraduate School (IPS - MPŠ in Slovene), where many researchers from JSI teach (including **ROMANDIC**'s Prof. Gams), does. IPS offers a robotics module called Intelligent systems and robotics: Intelligent Systems and Robotics are advanced information sciences and technologies comprising the following fields: intelligent systems, ambient intelligence, business intelligence, evolutionary computation, cognitive sciences, humanoid and service robotics, computer vision, as well as controlling robots and systems. These are the key technologies that enable the development of the information and knowledge-based societies.

Robotics courses are Humanoid and service robotics, see also Figure 2 (**ROMANDIC**'s Prof Gams is lecturer), Intelligent Robot Control and Biorobotics. Additionally, Sensors in Robotics and Biocybernetics is offered in the Sensor technologies programme.

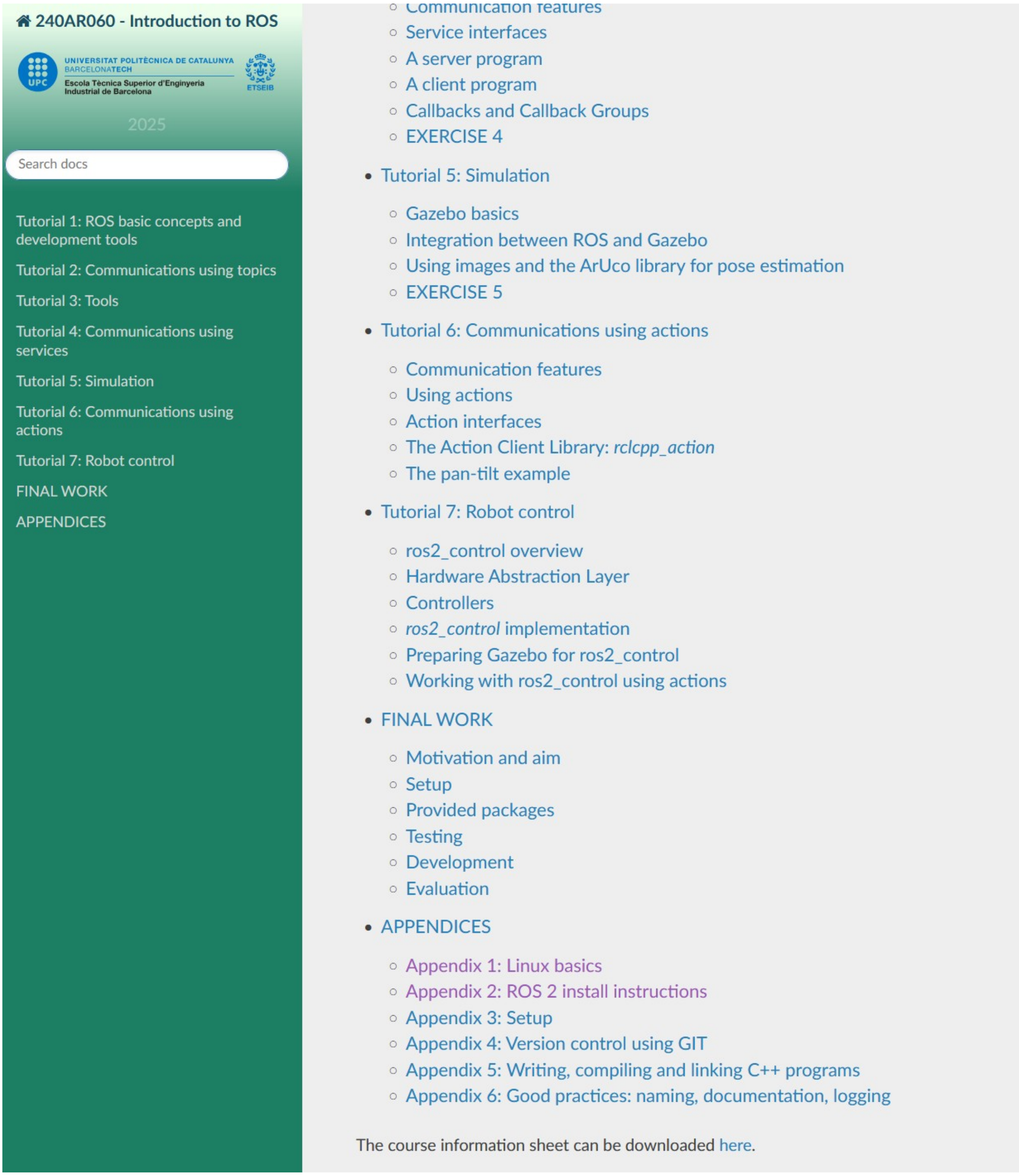

Figure 1: Screenshot of course *Introduction to ROS 2*, given at UPC.

2.2 Others

Of course, other institutions in the world offer similar courses, for example, below please find a screenshot of the course held at ETH Zurich in Fig. 3.

Učni načrt predmeta

Predmet: Humanoidna in servisna robotika

Course: Humanoid and Service Robotics

Študijski program in stopnja / Study programme and level	Študijska smer / Study field	Letnik / Academic year	Semester / Semester
Informacijske in komunikacijske tehnologije, 3. stopnja	Inteligentni sistemi in robotika	1	1
Information and Communication Technologies, 3rd cycle	Intelligent Systems and Robotics	1	1

Vrsta predmeta / Course type Izbirni / Elective

Univerzitetna koda predmeta / University course code: IKT3-623

Predavanja Lectures	Seminar Seminar	Vaje Tutorial	Klinične vaje work	Druge oblike študija	Samost. delo Individ. work	ECTS
30	30			30	210	10

*Navedena porazdelitev ur velja, če je vpisanih vsaj 15 študentov. Drugače se obseg izvedbe kontaktnih ur sorazmerno zmanjša in prenese v samostojno delo. / This distribution of hours is valid if at least 15 students are enrolled. Otherwise the contact hours are linearly reduced and transfered to individual work.

Nosilec predmeta / Course leader: izr. prof. dr. Bojan Nemec

Sodelavci / Lecturers: izr. prof. dr. Andrej Gams

Figure 2: Screenshot of course curricula *Humanoid and Service Robotics*, given at IPS.

ETH zürich Robotic Systems Lab

Lab Research Robots Scientific Events Education Publications Partnerships & Spin-offs

Homepage > Education > Lectures > Programming for Robotics - ROS

Programming for Robotics - ROS

Abstract: This course gives an introduction to the Robot Operating System (ROS2) including many of the available tools that are commonly used in robotics. With the help of different examples, the course should provide a good starting point for students to work with robots. They learn how to create software including simulation, to interface sensors and actuators, and to integrate control algorithms.

Objective:

- ROS2 architecture: Master, nodes, topics, messages, services, parameters and actions
- Console commands: Navigating and analyzing the ROS2 system and the catkin workspace
- Creating ROS packages: Structure, launch-files, and best practices
- ROS2 C++ client library (rclcpp): Creating your own ROS2 C++ programs
- Simulating with ROS2: Gazebo simulator, robot models (URDF) and simulation environments (SDF)
- Working with visualizations (RViz) and user interface tools (rqt)
- Inside ROS2: TF transformation system, time, bags

Content: This course consists of a guided tutorial and exercises with increasing level of difficulty when working with an autonomous robot. You learn how to setup such a system from scratch using ROS2, how to interface the individual sensors and actuators, and finally how to implement first closed loop control systems.

RSL Robotic Systems Lab

Course dates

Room: 17.02 HG D1.2; 19.02 HG D1.2; 21.02 HG D7.2; 24.02 HG G5; 28.02 HG D 7.2

Time: 08:00 - 12:00

Dates: 17.02, 19.02, 21.02, 24.02, 28.02

Lecturers: Fan Yang, Lennart Werner, Lorenzo Terenzi, Pol Eyschen, Marco Hutter

Figure 3: Screenshot of course *Programming for Robotics - ROS*, given at ETHZ.

3 Applying for a new course at IPS

3.1 Institutions & Organization

ROMANDIC's JSI is a research institute and as such, it does not offer education and courses. However, as is the case with many such institutions, JSI collaborates with Universities and higher education institutions, where early-career researchers can get degrees and where JSI's staff teaches.

JSI collaborates with several institutions, but due to geographical and organizational (and institution size) considerations, mostly with the University of Ljubljana and with the The Jožef Stefan International Postgraduate School (IPS).

The Jožef Stefan International Postgraduate School (IPS) was established in 2004 as an independent higher education institution. Its study programmes were approved by the Slovenian National Council for Higher Education. The initiative for the establishment of IPS came from the Jožef Stefan Institute (JSI). It was strongly supported by industry and an international network of cooperating universities and research institutes from the European Union, USA, Japan, and many other countries. The Jožef Stefan Institute provides the central research and educational basis, whereas other partners, such as invited research institutes, industrial and other enterprises also contribute their knowledge and innovation capacities for solving developmental problems.

Jožef Stefan Institute (JSI) is the main research base of IPS with over four hundred researchers with doctoral degrees. More than one third among them are elected university teachers. Researchers and university teachers are engaged by IPS for cooperation in research-innovation projects in the interest of IPS partners.

3.2 Study programmes

There are 4 doctoral (Third-cycle) programmes at IPS: Nanosciences and nanotechnologies, Information and Communication Technologies (IKT), Ecotechnologies and Sensor Technologies.

At the Department of Automatics, Biocybernetics and Robotics we mostly offer courses in the IKT programme. It offers four modules:

- Knowledge technologies
- Computer structures and systems
- Advanced internet technologies
- Intelligent systems and robotics

Each programme, as well as each modules, has an appointed head.

The programmes, modules and courses were approved by the Senate. It is the highest expert body of the IPS. It consists of at least 15 members, 12 of which are professors, researchers and other experts appointed by the Academic Council.

3.3 Application Procedure

To apply with a new course at MPS, we first consulted with the Vice-Dean for Study Matters, Prof. Dr. Aleksander Zidanšek. A new course can be proposed at the end of each year.

To submit the proposal, one has to send the application and a cover letter to to the study committee and to the IPS senate. However, a reasonable course of action is to first obtain general approval of the head of the module and the head of the programme.

We contacted the Head of the Information and Communication Technologies (IKT) programme, Prof. Dr. Nada Lavrač, and the head of the Robotics module, Prof. Dr. Aleš Ude, and sent them the draft proposal of the course. They also shared our proposal with the president for IKT affairs of the IPS Study Committee, Prof. Dr. Marko Bohanec, and the deputy chair of IKT programme Prof. Dr. Gregor Papa. We expanded the course to the form it is listed below based on their comments.

We submitted the proposal to the IPS Study Committee and the IPS Senate on Dec. 22nd, 2025. The application and the cover letter are attached at the end of the deliverable.

4 Content of the course

The course is titled *Developing and programming robot applications*.

Study programme and level *Information and Communication Technologies, 3rd cycle*.

Module *Intelligent Sytems and Robotics*, meant for 1 semester of studies.

Course type: Elective

Duration: Lectures: 15h; Tutorial: 35h, Other forms: 15h, Individual work: 85h.
Note: This distribution of hours is valid if at least 15 students are enrolled. Otherwise the contact hours are linearly reduced and transferred to individual work.

ECTS: 5.

Course leader: prof. dr. Andrej Gams **Lecturer:** dr. Mihael Simonič

Languages: Lectures and Tutorial: Slovenian, English

Prerequisites: Completed Bologna second cycle study program or an equivalent pre-Bologna university study program. This course requires knowledge of computer programming and basics of robotics. Recommended courses:

- Humanoid and Service Robotics[1]
- Robot vision

Content (Syllabus outline):

- *Programming paradigms in robotics:* component- based and modular software architectures, separation of perception, planning, control and execution of robot movements.
- *Communication in distributed systems:* communication models, synchronization, timing considerations, reliability, and robustness.
- *Reproducibility, experimentation, and research methodology:* experiment design, experiment setup versioning, and best practices for evaluating robotic experiments.
- *Development environment basics:* basic Linux command-line tools, file system, permissions, package management, version control software.
- *Introduction to the Robot Operating System (ROS):* middleware for distributed robotic systems and ecosystem of libraries, tools, and conventions that standardize robotic application development and hardware access.
- *Installing ROS 2:* workspace management and package setup, build tools, compiling and running simple nodes.
- *Core concepts and communication in ROS:* a distributed system is represented as a graph of nodes, which are independent processes, that can communicate via topics (data streaming via the publisher-subscriber pattern), services (request-response pattern), or actions (long-running goals using a request-feedback-result cycle).

[1] In the submitted document a wrong course is listed in English, but the correct one in Slovenian language.

- *Working with ROS:* launch files, parameters, logging, system state introspection tools.
- *Simulation with ROS:* Gazebo simulator, robot models (URDF), simulation environments (SDF)
- *Introduction to containerization:* basics of Docker (images, containers, volumes, networking), key commands and best practices for deployment.
- Running and testing ROS applications in containerized environments.
- Interfacing and controlling of peripheral devices, i.e., through Raspberry Pi-based nodes.
- *Final project:* Students implement a small ROS application.

Books:

- The Linux Command Line, William Shots,
- A Concise Introduction to Robot Programming with ROS 2, Francisco Martín Rico, 2023, CRC Press.; Source code
- Using Docker, Adrian Mouat.

Papers:

- S. Macenski, T. Foote, B. Gerkey, C. Lalancette, W. Woodall, "Robot Operating System 2: Design, architecture, and uses in the wild," Science Robotics vol. 7, May 2022.
- Simonič M, Pahič R, Gašpar T, Abdolshah S, Haddadin S, Catalano M G, Wörgötter F, Ude A; Modular ROS-based software architecture for reconfigurable, Industry 4.0 compatible robotic workcells Proceedings Article; 20th International Conference on Advanced Robotics (ICAR), pp. 44-51, IEEE Ljubljana, Slovenia, 2021.

Web resources:

- The Missing Semester of Your CS Education, https://missing.csail.mit.edu/
- ROS 2 Documentation, https://docs.ros.org/en/humble/index.html
- Gazebo tutorials: https://gazebosim.org/docs/fortress/tutorials
- ROS 2 on Raspberry Pi; https://docs.ros.org/en/rolling/How-To-Guides/Installing-on-Raspberry-Pi.html

Objectives and competences: This course provides theoretical background and foundational and practical knowledge for developing robotic applications using ROS 2. Participants will learn Linux basics relevant to robotics development, master ROS 2 core concepts (nodes, messaging, services), and learn how to use containers to build, deploy and manage reproducible ROS 2 environments for development and distribution. By the end of the course, students will be able to build, run, and manage ROS 2-based robotics applications both on local machines and in containers.

Intended learning outcomes: Student successfully completing this course will be able to:

- Reason about architectures and architectural trade-offs and design novel or research-oriented robotic software architectures.
- Understand timing constraints in order to evaluate suitability for advanced control and implied safety issues (e.g. for human–robot collaboration).
- Understand how infrastructure choices affect scientific validity and replicability.
- Use Linux command-line tools confidently for development and system management.
- Understand the architecture of ROS 2: nodes, topics, services, actions, launch mechanisms.
- Build and manage ROS 2 workspaces, packages, and compile/launch nodes.
- Use container environments ROS 2 applications and manage dependencies reliably across environments.
- Deploy ROS 2 applications inside containers and set appropriate hardware access rights.
- Debug and test ROS 2 applications.

Learning and teaching methods:

- Mix of theoretical tutorials, live demonstrations, and practical/lab-based sessions.
- Hands-on exercises and mini-projects
- Final Project

Assessment: (weight in %)

- Exercises/assignments (25%)
- Mini-projects (25%)
- Final project (50%)

Appendix: Copy of the cover letter

Note: The submitted document was digitally signed.

Andrej Gams, Mihael Simonič
Institut Jožef Stefan
Jamova cesta 39
1000 Ljubljana

Študijska komisija MPŠ in Senat MPŠ
Mednarodna podiplomska šola Jožefa Stefana
Jamova 39
1000 Ljubljana

Ljubljana, 22. 12. 2025

Zadeva: Prošnja za vključitev novega predmeta "Razvoj in programiranje robotskih aplikacij" v program Informacijske in komunikacijske tehnologije doktorskega študija na MPŠ

Spoštovani člani študijske komisije MPŠ, spoštovani člani senata MPŠ,

V priponki vam podpisani Andrej Gams ter Mihael Simonič, oba raziskovalca na Odseku za Avtomatiko, Biokibernetiko in Robotiko Inštituta "Jožef Stefan", pošiljava predlog novega predmeta za vključitev v program Informacijske in komunikacijske tehnologije doktorskega študija na MPŠ, v modul robotika.

Predlagani predmet "Razvoj in programiranje robotskih aplikacij" je nastal kot odgovor na potrebe sodobne robotike ter opaženega manjka znanja pri delu s študenti na drugih robotskih predmetih na MPŠ. V tem izbirnem predmetu predmeta bi študentje skozi predavanja, vaje in projektno delo osvojili temeljno in praktično znanje za razvoj robotskih aplikacij z uporabo robotskega operacijskega sistema ROS 2. Ob zaključku predmeta bodo študenti poznali programske paradigme v robotiki, komunikacijo v porazdeljenih sistemih ter pogoje ponovljivosti raziskovalne metodologije, hkrati pa bodo sposobni razvijati, zaganjati in upravljati robotske aplikacije, temelječe na robotskem operacijskem sistemu ROS 2, tako na lokalnih računalnikih kot v vsebnikih.

Predmet je osnovan na temeljni teoretični osnovi programskih paradigm v robotiki in komunikacije v porazdeljenih sistemih. Poleg teoretičnega temelja je dodatno precej poudarjena praktična komponenta, ki omogoča študentom dobro pripravo tako za raziskovalno delo kot za morebitno kasnejše udejstvovanje izven akademskega okolja. Vsebino predmeta so predhodno pogledali in načelno podprli prof. A. Ude, vodja modula robotika, prof. N. Lavrač, vodja programa IKT na MPŠ ter prof. Bohanec, predsednik študijske komisije za področje IKT. Njihove komentarje smo upoštevali pri pripravi končnega predloga.

Lep pozdrav, v imenu obeh predlagateljev,

izr. prof. Andrej Gams

Appendix: Copy of the Curricula application document

Note: The submitted document was digitally signed.

UČNI NAČRT PREDMETA / COURSE SYLLABUS	
Predmet:	Razvoj in programiranje robotskih aplikacij
Course title:	Developing and programming robot applications

Študijski program in stopnja Study programme and level	Modul Module	Letnik Academic year	Semester Semester
Informacijske in komunikacijske tehnologije, 3. stopnja	Inteligentni sistemi in robotika	1	1
Information and Communication Technologies, 3rd cycle	Intelligent Sytems and Robotics	1	1

Vrsta predmeta / Course type	Izbirni / Elective
Univerzitetna koda predmeta / University course code:	IKT3-xxx

Predavanja Lectures	Seminar Seminar	Sem.vaje Tutorial	Lab. vaje Laboratory work	Druge oblike	Samost. delo Individ. work	ECTS
15		35		15	85	5

**Navedena porazdelitev ur velja, če je vpisanih vsaj 15 študentov. Drugače se obseg izvedbe kontaktnih ur sorazmerno zmanjša in prenese v samostojno delo. / This distribution of hours is valid if at least 15 students are enrolled. Otherwise the contact hours are linearly reduced and transfered to individual work.*

Nosilec predmeta / Course leader	prof. dr. Andrej Gams
Sodelavec predmeta / Lecturer:	dr. Mihael Simonič

Jeziki / Languages:	**Predavanja / Lectures:**	slovenščina, angleščina / Slovenian, English
	Vaje / Tutorial:	slovenščina, angleščina / Slovenian, English

Pogoji za vključitev v delo oz. za opravljanje študijskih obveznosti:	**Prerequisites:**
Zaključena druga stopnja bolonjskega študija ali diploma univerzitetnega študijskega programa. Pri tem predmetu je potrebno predznanje programiranja ter osnov robotike. Priporočeni predmeti: - humanoidna in servisna robotika - robotski vid	Completed Bologna second cycle study program or an equivalent pre-Bologna university study program. This course requires knowledge of computer programming and basics of robotics. Recommended courses: - Intelligent robot control - Robot vision

Vsebina:	**Content (Syllabus outline):**
- *Programske paradigme v robotiki:* komponentna in modularna programska arhitektura, ločitev zaznavanja okolice, načrtovanja, vodenja in izvajanja robotskih gibov. - *Komunikacija v porazdeljenih sistemih*: komunikacijski modeli, sinhronizacija, časovne zahteve, zanesljivost in robustnost. - *Ponovljivost, eksperimentiranje in raziskovalna metodologija*: zasnova eksperimentov, različice eksperimentalnih nastavitev ter dobre prakse za vrednotenje robotskih eksperimentov. - *Osnove razvojnega okolja*: osnovna orodja Linuxove ukazne vrstice, datotečni sistem, dovoljenja, upravljanje paketov, sistemi za nadzor različic.	- *Programming paradigms in robotics:* component-based and modular software architectures, separation of perception, planning, control and execution of robot movements. - *Communication in distributed systems*: communication models, synchronization, timing considerations, reliability, and robustness. - *Reproducibility, experimentation, and research methodology*: experiment design, experiment setup versioning, and best practices for evaluating robotic experiments. - *Development environment basics:* basic Linux command-line tools, file system, permissions, package management, version control software.

- *Uvod v Robotski operacijski sistem (ROS)*: vmesna programska oprema (middleware) za porazdeljene robotske sisteme ter ekosistem knjižnic, orodij in konvencij, ki standardizirajo razvoj robotskih aplikacij in dostop do strojne opreme.	- *Introduction to the Robot Operating System (ROS):* middleware for distributed robotic systems and ecosystem of libraries, tools, and conventions that standardize robotic application development and hardware access.
- *Namestitev sistema ROS 2*: upravljanje delovnega prostora in paketov, orodja za prevajanje, prevajanje in zaganjanje enostavnih vozlišč.	- *Installing ROS 2*: workspace management and package setup, build tools, compiling and running simple nodes.
- *Osrednji koncepti in komunikacija v ROS-u*: porazdeljeni sistem je predstavljen kot graf vozlišč, kjer so vozlišča neodvisni procesi, ki komunicirajo prek tem (pretakanje podatkov prek pošiljanja in prejamanja sporočil), storitev (mehanizem zahteva–odziv) ali akcij (dlje časa trajajoča opravila s sprotnimi povratnimi informacijami in potrditvijo končnega rezultata).	- *Core concepts and communication in ROS*: a distributed system is represented as a graph of nodes, which are independent processes, that can communicate via topics (data streaming via the publisher-subscriber pattern), services (request-response pattern), or actions (long-running goals using a request-feedback-result cycle).
- *Delo z ROS*: uporaba zagonskih datotek, parametrov, beleženje dnevnikov, orodja za spremljanje in analizo stanja sistema.	- *Working with ROS*: launch files, parameters, logging, system state introspection tools.
- *Simulacija z ROS:* simulator Gazebo, opis robotov v formatu URDF, opis simulacijskih okolij v formatu SDF.	- *Simulation with ROS*: Gazebo simulator, robot models (URDF), simulation environments (SDF)
- *Uvod v tehnologijo vsebnikov*: osnove Dockerja (slike, vsebniki, nosilci, omrežje), ključni ukazi in dobre prakse za namestitev aplikacij.	- *Introduction to containerization*: basics of Docker (images, containers, volumes, networking), key commands and best practices for deployment.
- Zaganjanje in testiranje ROS aplikacij v vsebnikih.	- Running and testing ROS applications in containerized environments.
- Povezovanje in krmiljenje perifernih naprav, npr. prek vozlišč na platformi Raspberry Pi.	- Interfacing and controlling of peripheral devices, i.e., through Raspberry Pi-based nodes.
- Zaključni projekt: študenti implementirajo manjšo aplikacijo v ROS.	- Final project: Students implement a small ROS application.

Temeljna literatura in viri / Readings:

Knjige/Books:

- The Linux Command Line, William Shots, https://linuxcommand.org/tlcl.php
- A Concise Introduction to Robot Programming with ROS 2, Francisco Martín Rico, 2023, CRC Press.; Source code: https://github.com/fmrico/book_ros2
- Using Docker, Adrian Mouat, https://www.oreilly.com/library/view/using-docker/9781491915752/

Članki/Papers:

- S. Macenski, T. Foote, B. Gerkey, C. Lalancette, W. Woodall, "Robot Operating System 2: Design, architecture, and uses in the wild," Science Robotics vol. 7, May 2022.
- Simonič M, Pahič R, Gašpar T, Abdolshah S, Haddadin S, Catalano M G, Wörgötter F, Ude A; Modular ROS-based software architecture for reconfigurable, Industry 4.0 compatible robotic workcells Proceedings Article; 20th International Conference on Advanced Robotics (ICAR), pp. 44-51, IEEE Ljubljana, Slovenia, 2021.

Web resources:

- The Missing Semester of Your CS Education, https://missing.csail.mit.edu/
- ROS 2 Documentation: https://docs.ros.org/en/humble/index.html
- Gazebo tutorials: https://gazebosim.org/docs/fortress/tutorials
- ROS 2 on Raspberry Pi; https://docs.ros.org/en/rolling/How-To-Guides/Installing-on-Raspberry-Pi.html

Cilji in kompetence:	Objectives and competences:
Ta predmet zagotavlja teoretične osnove ter temeljno in praktično znanje za razvoj robotskih aplikacij z uporabo ROS 2. Udeleženci se bodo naučili osnov Linuxa, pomembnih za razvoj aplikacij v robotiki, osvojili ključne koncepte ROS 2 (vozlišča, sporočanje, storitve) ter se naučili uporabljati tehnologijo vsebnikov za gradnjo, nameščanje in upravljanje ponovljivih okolij za razvoj in distribucijo ROS 2 aplikacij. Ob zaključku predmeta bodo študenti sposobni razvijati, zaganjati in upravljati robotske aplikacije, temelječe na ROS 2, tako na lokalnih računalnikih kot v vsebnikih.	This course provides theoretical background and foundational and practical knowledge for developing robotic applications using ROS 2. Participants will learn Linux basics relevant to robotics development, master ROS 2 core concepts (nodes, messaging, services), and learn how to use containers to build, deploy and manage reproducible ROS 2 environments for development and distribution. By the end of the course, students will be able to build, run, and manage ROS 2-based robotics applications both on local machines and in containers.

Predvideni študijski rezultati:	Intended learning outcomes:
Študent-ka z uspešno opravljenimi obveznostmi tega predmeta bo sposoben-na: - razumeti arhitekture in arhitekturne kompromise programske opreme ter načrtovati nove ali raziskovalno usmerjene arhitekture robotske programske opreme; - razumeti časovne omejitve, da lahko presodi primernost za napredne pristope vodenja in z njimi povezane varnostne vidike (npr. pri sodelovanju človek–robot); - razumeti, kako infrastrukturne izbire vplivajo na znanstveno veljavnost in ponovljivost raziskav; - uporabljati orodja Linuxove ukazne vrstice za razvoj in upravljanje sistema; - razumeti arhitekturo ROS 2: vozlišča, teme, storitve, akcije in zagonske mehanizme; - vzpostaviti in upravljati delovne prostore in pakete ROS 2 ter prevajati in zaganjati vozlišča; - uporabljati vsebnike za aplikacije ROS 2 ter zanesljivo upravljati odvisnosti med različnimi okolji; - namestiti aplikacije ROS 2 znotraj kontejnerjev in upravljati pravice za dostop do strojne opreme; - razhroščevati in testirati aplikacije ROS 2.	Student successfully completing this course will be able to: - Reason about architectures and architectural trade-offs and design novel or research-oriented robotic software architectures. - Understand timing constraints in order to evaluate suitability for advanced control and implied safety issues (e.g. for human–robot collaboration). - Understand how infrastructure choices affect scientific validity and replicability. - Use Linux command-line tools confidently for development and system management. - Understand the architecture of ROS 2: nodes, topics, services, actions, launch mechanisms. - Build and manage ROS 2 workspaces, packages, and compile/launch nodes. - Use container environments ROS 2 applications and manage dependencies reliably across environments. - Deploy ROS 2 applications inside containers and set appropriate hardware access rights. - Debug and test ROS 2 applications.

Metode poučevanja in učenja:	Learning and teaching methods:
- kombinacija teoretičnih predstavitev, demonstracij v živo ter praktičnih laboratorijskih vaj; - praktične vaje in mini projekti; - zaključni projekt.	- Mix of theoretical tutorials, live demonstrations, and practical/lab-based sessions. - Hands-on exercises and mini-projects - Final Project

Načini ocenjevanja:	Delež (v %) / Weight (in %)	Assessment:
Vaje	25 %	Exercises/assignments
Projektne laboratorijsje vaje	25 %	Mini-projects
Seminarska naloga	50 %	Final project

Reference nosilca / Lecturer's references:

- JAQUIER, Noemie, WELLE, Michael C., **GAMS, Andrej**, YAO, Kunpeng, FICHERA, Bernardo, BILLARD, Aude, UDE, Aleš, ASFOUR, Tamim, KRAGIČ, Danica. Transfer learning in robotics : An upcoming breakthrough? A review of promises and challenges. The international journal of robotics research. [Print ed.]. 2025, vol. 44, iss. 3, str. 465-485. ISSN 0278-3649. DOI: 10.1177/02783649241273565.
- **SIMONIČ, Mihael**, UDE, Aleš, NEMEC, Bojan. Hierarchical learning of robotic contact policies. Robotics and computer-integrated manufacturing. Apr. 2024, vol. 86, [article no.] 102657, 1-12 str., ilustr. ISSN 1879-2537. https://www.sciencedirect.com/science/article/pii/S0736584523001321, DOI: 10.1016/j.rcim.2023.102657.
- **GAMS, Andrej**, PETRIČ, Tadej, NEMEC, Bojan, UDE, Aleš. Manipulation learning on humanoid robots. Current robotics reports. 2022, vol. 3, str. 97-109. ISSN 2662-4087. DOI: 10.1007/s43154-022-00082-9.
- UDE, Aleš, **SIMONIČ, Mihael**, KUSTER, Boris, MAVSAR, Matija, BEM, Martin, WÖRGÖTTER, Florentin, et al. The challenges of using robots to automate the recycling of electronic devices : a new approach incorporating ai, modularity, and soft robotics. IEEE robotics & automation magazine. <in press> 2025, vol. , iss. , 1-15 str. ISSN 1070-9932. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=11220802, DOI: 10.1109/MRA.2025.3619331.
- LONČAREVIĆ, Zvezdan, **GAMS, Andrej**, REBERŠEK, Simon, NEMEC, Bojan, ŠKRABAR, Jure, SKVARČ, Jure, UDE, Aleš. Specifying and optimizing robotic motion for visual quality inspection. Robotics and computer-integrated manufacturing. 2021, vol. 72, str. 102200-102200-14. ISSN 0736-5845. DOI: 10.1016/j.rcim.2021.102200.