COURSE DESCRIPTION CARD - SYLLABUS

Course name
Tools and software for robotic systems [S1AiR1E>PO5-NiOdSR]

Course

Field of study
Automatic Control and Robotics

Year/Semester
4/7

Area of study (specialization)
–

Profile of study
general academic

Level of study
first-cycle

Course offered in
English

Form of study
full-time

Requirements
elective

Number of hours

Lecture
30

Laboratory classes
30

Other
0

Tutorials
0

Projects/seminars
0

Number of credit points

5,00

Coordinators

dr inż. Krzysztof Walas
krzysztof.walas@put.poznan.pl

Lecturers

Prerequisites

Has structured knowledge of selected algorithms and data structures as well as methodology and techniques of procedural and object-oriented programming. Knows and understands the basic processes occurring in the software development cycle. Has structured knowledge of classification, construction and kinematic structures, mathematical description, principles of operation and programming of manipulation robots; knows and understands the mathematical description, properties and principles of operation and programming of simple mobile robots to an advanced degree. Knows and understands the theory and methods of operation of basic electronic, analogue and digital components, selected electronic circuits and systems to an advanced degree.

Course objective

In-depth understanding of practical problems related to robot programming and the tools supporting the use of sensors and actuators.

Course-related learning outcomes

Knowledge:
Knows and understands typical engineering technologies, principles and techniques of construction of

simple automation and robotics systems; knows and understands the principles of selection of executive systems, computational units and measurement and control elements and devices [K1_W20 (P6S_WG)].
Is familiar with the current status and latest development trends of the field of automation and robotics [K1_W21 (P6S_WG)].
Knows and understands the fundamental dilemmas of modern civilisation related to the development of automation and robotics [K1_W28 (P6S_WK)].
Skills:
Is able to plan, prepare and simulate the operation of simple automation and robotics systems [K1_U10 (P6S_UW)].
Is able to select the type and parameters of the measurement system, control unit and peripheral and communication modules for the selected application and integrate them in the form of the resulting measurement and control system [K1_U22 (P6S_UW)].
Is able to develop a solution to a simple engineering task and implement, test and run it in a selected programming environment on a PC for selected operating systems [K1_U26 (P6S_UW)].
Social competences:
Is aware of the importance and understands the non-technical aspects and consequences of engineering activities, including their impact on the environment and the related responsibility for decisions; is ready to care for the achievements and traditions of the profession [K1_K2 (P6S_KR)].
The graduate is aware of the need for a professional approach to technical issues, meticulous familiarization with the documentation and environmental conditions in which the equipment and its components can operate. The graduate is ready to observe the rules of professional ethics and to demand it from others, to respect the diversity of opinions and cultures [K1_K5 (P6S_KR)].

Methods for verifying learning outcomes and assessment criteria

Learning outcomes presented above are verified as follows:

Learning outcomes presented above are verified as follows:
Lectures
– assessment of knowledge and skills through written examination
Laboratory classes:
– assessment of knowledge and skills through a written test on robot programming
– assessment of knowledge during classes
Additionall points:
– efficiency of using gathered knowledge while solving problems
– proposal of discussing additional aspects of the issue

Programme content

1. Introduction to Linux
2. Principal tools in Linux systems, working in console: Nano, VIM, SSH, MC, apt-get,gcc
3. Git and version control, certain aspects of project management
4. Make, Cmake, Doxygen (under Linux)
5. Creating libraries, linking process, structure of the projects
6. Using example libraries: Eigen, OpenCV, Boost (e.g. asio) OpenGL
7. Principal design patterns.
8. Robot Operating System
9. Python

Course topics

The topics of the lectures are related to learning practical problems of robot programming and tools to support the operation of sensors and actuators of robotic systems.

Teaching methods

A) Lecture: multimedia presentations (slides) illustrated with examples analyzed on the board and program code fragments implementing selected content described during the lecture
B) Laboratory: Classes will be conducted using a problem-solving approach. The student will receive an introduction to the laboratory, where the link between the topic of classes and the content of the lecture will be described. Then, with the help of the teacher, the student will solve subsequent problems that will be presented.

Bibliography

Basic
1. Mark Mitchell, Jeffrey Oldham, Alex Samuel, Advanced Linux Programming, New Riders Publishing
2. Robot Operating System (ROS), Springer 2016
Additional
1. Tutorial ROS: http://wiki.ros.org/ROS/Tutorials
2. Tutorial Python: https://docs.python.org/3/tutorial/
3. Turorial Linux: http://linuxcommand.org/learning_the_shell.php

Breakdown of average student's workload

	Hours	ECTS
Total workload	120	5,00
Classes requiring direct contact with the teacher	60	2,50
Student's own work (literature studies, preparation for laboratory classes/ tutorials, preparation for tests/exam, project preparation)	60	2,50