COURSE DESCRIPTION CARD - SYLLABUS

Course name
Tools and Software for Robotic Systems [S1AiR2>PO2-NiOdSR]

Course

Field of study
Automatic Control and Robotics

Year/Semester
3/5

Area of study (specialization)
–

Profile of study
general academic

Level of study
first-cycle

Course offered in
Polish

Form of study
full-time

Requirements
elective

Number of hours

Lecture
15

Laboratory classes
30

Other
0

Tutorials
0

Projects/seminars
0

Number of credit points

3,00

Coordinators

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

dr hab. inż. Dominik Belter prof. PP
dominik.belter@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:
1. Has organized and extended knowledge on programming tools for robotics.
2. Has basic knowledge essential for designing robot control and perception system.
3. Has organized and extended knowledge on Linux operating system and Robot Operating System (ROS)

Skills:
1. Is able to design robot control and perception system architecture.
2. Is able to select programming tool essential for implementing control system of a robot
3. Is able to gather information from literature, databases and other sources, is able to integrate and interpret gathered information. Additionally, is able to take conclusions and formulate and justify opinions.

Social competences:
1. Is ready to critically assess his knowledge. He understands the need for and knows the opportunities for continuous professional, personal and social skills development, he is able to inspire and organize the learning process of others
2. Is ready to think and act in an entrepreneurial way.

Methods for verifying learning outcomes and assessment criteria

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

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	75	3,00
Classes requiring direct contact with the teacher	45	2,00
Student's own work (literature studies, preparation for laboratory classes/ tutorials, preparation for tests/exam, project preparation)	30	1,00