Course Outline Student

Edit Resource (/COMP3431/20T3/resources/52239/edit)

Contents

- Course Details
- Course Summary
- Course Timetable
- Course Aims
- Student Learning Outcomes
- Assumed Knowledge
- Teaching Rationale
- Teaching Strategies
- Assessment
- Academic Honesty and Plagiarism
- Course Schedule
- Resources for Students
- Course Evaluation and Development

Course Details

Course Code	COMP3431/COMP9434
Course Title	Robot Software Architectures

Units of Credit	6
Course Website	http://cse.unsw.edu.au/~cs3431 (http://cse.unsw.edu.au/~cs3431)
Handbook Entry	http://www.handbook.unsw.edu.au/undergraduate/courses/current/COMP3431.html (http://www.handbook.unsw.edu.au/undergraduate/courses/current/COMP3431.html)

Course Summary

This course gives a practical introduction to the intelligent control of robots. In labs, students will get hands on experience with some of these planning systems.

We start with an introduction to the structure of robot software, including the Robot Operating System, ROS (http://www.ros.org/) . We will be using TurtleBot3 Waffle Pi (http://turtlebot.com/) Robots.

At the end of this course, students should have:

- experience with the details of one robot architecture,
- an overview of a range of other architectures and decision making systems,
- an understanding of the practicalities of programming physical robots as opposed to purely software systems,
- an understanding of a robot as a complete system, as well as its component parts.

Course Timetable

The official course timetable is available here (http://timetable.unsw.edu.au/2020/COMP3431.html) . Please note that the timetables for COMP3431 and COMP9434 are the same. We will use the 12pm-2pm Monday slot for lectures and the labs will be on Tuesday afternoon 12pm-3pm and Wednesday 3pm-6pm. Lectures will be online and the labs will be in J18-212 (The Willis Annex, next door to J17 and K17). If possible, we will also use.

Robotics and COVID-19: Because of the ongoing pandemic, lectures will be online, either recorded or live using Moodle's Collaborate facility. Because this course is meant to provide practical experience with robots, we will hold labs, but we will have to be careful to maintain physical distancing. Students will be organised into groups of 4 or 5 students, with one robot assigned to each group. **To maintain a safe working environment, each group should only have one student handling the robot at any one time. This may mean having one student in the lab working on the robot, with the others online. This will be very experimental and we ask for you patience as we work out the best way of handling access to the robots.**

Course Aims

This course gives both a theoretical and a practical introduction to the control of robotic systems. The course is in two parts:

An overview of robot software architectures :

This part of the course gives students an introduction to the structure of robot software, including the theory of decision making: How can an agent build up knowledge of its world? Having done that, how can an agent choose its actions? A variety of different decision making and planning architectures are covered; including classical planning, reinforcement learning and cognitive architectures. In labs, students will get hands on experience with some of these planning systems.

Practical Robotics

This part of the course gives students a practical introduction to some specific robot architectures; the main system architecture will be the Robot Operating System, ROS (http://www.ros.org/) . We will be using TurtleBot (http://turtlebot.com/) Robots, or other robots as available.

Student Learning Outcomes

At the end of this course, students should have:

- experience with the details of one robot architecture,
- an overview of a range of other architectures and decision making systems,

- an understanding of the practicalities of programming physical robots as opposed to purely software systems,
- an understanding of a robot as a complete system, as well as its component parts.

This course contributes to the development of the following graduate capabilities:

Graduate Capability	Acquired in
scholarship: understanding of their discipline in its interdisciplinary context	1 - 12
scholarship: capable of independent and collaborative enquiry	1 - 12
scholarship: rigorous in their analysis, critique, and reflection	1 - 12
scholarship: able to apply their knowledge and skills to solving problems	1 - 12
scholarship: ethical practitioners	1 - 12
scholarship: capable of effective communication	1 - 12
scholarship: information literate	1 - 12
scholarship: digitally literate	1 - 12
leadership: enterprising, innovative and creative	1 - 12
leadership: capable of initiating as well as embracing change	1 - 12
leadership: collaborative team workers	1 - 12
professionalism: capable of independent, self-directed practice	1 - 12
professionalism: capable of lifelong learning	1 - 12
professionalism: capable of operating within an agreed Code of Practice	1 - 12
global citizens: capable of applying their discipline in local, national and international contexts	1 - 12

global citizens: culturally aware and capable of respecting diversity and acting in socially just/responsible ways	1 - 12
global citizens: capable of environmental responsibility	1 - 12

Assumed Knowledge

This course is for postgraduate students (COMP9434) and advanced undergraduates (COMP3431). It is a 3rd year course because it has few pre-requisites.

Students in this course are expected to be able to program in C++, Java or Python.

While not a formal pre-requisite or co-requisite, it is recommended that students take a machine learning course if they are interested in artificial intelligence, eg COMP9417 (http://www.cse.unsw.edu.au/~cs9417/) . That course does not need to be taken before this one. The artificial intelligence course, COMP3411 (http://www.cse.unsw.edu.au/~cs3411/) , is also recommended but it does not give the depth of Machine Learning knowledge that is useful for this course. While it isn't a formal pre-requisite, students who have already taken COMP3411 or an equivalent course often find this course easier.

Because there are only a small number of robots, this class is limited in size. A minimum WAM is required for the course. For people who have a WAM below the required WAM, there is a waiting list. That waiting list is used to fill up any positions remaining just before the start of the semester.

Teaching Rationale

The class has traditionally been fairly small (20-30 students). There is no textbook that covers all the topics at the right level of detail for this course, so there is no set text, but there are recommended references below.

The assignments are all group-work. Furthermore, the assignments are graded is a style more like a design class than a computer science class: you demonstrate your robot doing what it does, and then the class discusses why it behaved that way and whether that is, on balance, a good thing. You will also write a final report to accompany your main project.

Teaching Strategies

- Lectures ... introduce concepts, show examples
- Tutorials ... reinforce concepts and provide additional examples
- Lab Work ... introduce technology required for the assignments
- Assignments .. allow students to solve significant problems

Assessment

There are two large assignments worth 20% and 40%, and a final report worth 40%.

- Preliminary Assignment: 20% (Due end of week 4)
- Project: 40% (demo in week 10)
- Project report: 40% (report due end of week 11)

Lectures and Assignments are similar between COMP3431 and COMP9434 students.
Units of credit: This is a 6 UOC course

Academic Honesty and Plagiarism

Plagiarism is defined as (https://student.unsw.edu.au/what-plagiarism) *using the words or ideas of others and presenting them as your own* . UNSW and CSE treat plagiarism as academic misconduct, which means that it carries penalties as severe as being excluded from further study at UNSW. There are on-line sources to help you understand what plagiarism is and how it is dealt with at UNSW: MyUNSW: Plagiarism (https://student.unsw.edu.au/plagiarism) and Academic Misconduct (https://www.gs.unsw.edu.au/policy/documents/studentmisconductprocedures.pdf) .

Make sure that you read and understand these. Ignorance is not accepted as an excuse for plagiarism.

Course Schedule

1. Introduction & History of AI and Robotics
   Introduction to ROS (the Robot Operating System)
2. More ROS
   Robot Software Architectures
3. Mapping and Navigation

4. Robot Vision
5. More Robot vision
6. Flexibility Week
7. Human-Robot Interaction
8. Robot Learning
9. Project Work
10. Demonstrations

Resources for Students

There is no set text for this course. However, the following books are recommended:

TurtleBot3 Online Manual (http://emanual.robotis.com/docs/en/platform/turtlebot3/overview/#overview) ; Robotis

ROS Robot Programming (http://www.robotis.com/service/download.php?no=719) ; YoonSeok Pyo, HanCheol Cho, RyuWoon Jung, TaeHoon Lim

> This is an introduction to robot programming using ROS, written by the developers of the TurtleBot3. The PDF is free to download and will be our primary text.

Artificial Intelligence: A Modern Approach (http://aima.cs.berkeley.edu/) **; S. Russell and P. Norvig**

> This is a good overview textbook for artificial intelligence in general. If you were going to get one book on AI, this would be the one I'd choose.

Probabilistic Robotics (http://robots.stanford.edu/probabilistic-robotics/) **; S. Thrun, W. Burgard and D. Fox**

> This is a newer textbook covering "perception and control in the face of uncertainty". It covers those areas well, but doesn't have quite the breadth of this course.

Automated Planning: Theory and Practice (http://www.laas.fr/planning/) **; M. Ghallab, D. Nau and P. Traverso**

> This book focuses on the planning and decision making areas of artificial intelligence. It goes goes into more detail in that one area. It provides a readable introduction to this area of artificial intelligence.

Reinforcement Learning: An Introduction (http://www-anw.cs.umass.edu/%7Erich/book/the-book.html) **; R. Sutton and A. Barto**

> This is a good book on reinforcement learning and Markov Decision Processes, in the same way that "Automated Planning" is a good book on planning systems: it covers its section of AI well.

Machine Learning
(http://www.cs.cmu.edu/afs/cs.cmu.edu/user/mitchell/ftp/mlbook.html) **; T. Mitchell**

> While machine learning is only touched on in this course, I consider it a very important part of AI and robotics, so I'm putting this book here anyway.

Markov Decision Processes: Discrete Stochastic Dynamic Programming; M. Puterman

> This book focuses in great detail on the theory of Markov Decision Processes. It is almost 10 years old now, and so lacks some of the newer contributions to the field, but it is an excellent reference on the mathematical theory of stochastic decision making.

Course Evaluation and Development

This course is evaluated each session using the CATEI system.

In the previous offering of this courses, students noted that space is limited.

Based on their comments, we can now use the Design Next studio, on the same floor as the robotics lab, which is much larger.

Resource created about a year ago (Sunday 06 September 2020, 05:54:55 PM), last modified about a year ago (Sunday 13 September 2020, 10:25:15 PM).

Comments

(/COMP3431/20T3/forums/search?forum_choice=resource/52239)
(/COMP3431/20T3/forums/resource/52239)

There are no comments yet.