Syllabus: Software for Robotics Today

Course Details

Course CSC 591/791 Software for Robotics Today

Meeting Times Tu/Th 11:45–01:00p

Credits 3 credit hours

Instructor Dr. John-Paul Ore

Email [*JPO: add NCSU email*]

Office TDB Engineering Building 2

Office Hours Mondays 1:00-2:30p and by email appointment.

Grader TBD

Course Prerequisites

Formal: A software engineering course at the undergraduate or graduate level (equivalent to CSC 326, 510); or permission from the instructor.

Informal: Mathematical maturity, comfort reading and writing journal articles; other helpful courses include CSC 333, 512, 707.

Overview

Advances in software-enabled robotics are unlocking disruptive capabilities in diverse areas, including transportation, space exploration, domestic assistants, agriculture, and medical robots. This course explores fundamental algorithms and AI-enabled capabilities for robots operating in real-world environments. The focus of the course will be on ways to make software-enabled robotics more reliable through automated testing and analysis. Topics covered in this class include robotic software architecture, motion planning, testing, state estimation, deep learning, AI, manipulation, multi-robot systems, and flying robots. We will explore these topics through the critical analysis of classic and contemporary articles in the fields of robotics and software engineering. By examining algorithms and systems used in the real world, we will determine what it takes to go from theory to implementation. In this class, students will present, discuss, and write analyses of current software engineering and robotics research. Further, students will work in groups on a semester-long project to explore areas of interest in software engineering and robotics. For 500-level students, the final projects can optionally be a replication study. The goal of this course is for students to gain an understanding of the challenges of the current state of the art in software-enabled robotics and to improve problem-solving and critical analysis skills.

Topics Covered

Topics covered in this class include robotic software architecture, motion planning, testing, state estimation, deep learning, AI, manipulation, multi-robot systems, and flying robots.

Depending on student interest, we will adjust the syllabus to cover areas in more or less detail. If you are particularly interested in covering a topic, please let me know.

Student Learning Outcomes

By the end of this course, students will be able to:

- Explain the design and function of robotic software middleware architecture.
- Identify, explain, and critique recent robotics and software engineering research.
- Summarize and communicate a research idea clearly with compelling motivation.
- Know the most prominent conferences and journals for software engineering and robotics.
- Identify current trends and open problems in software-enabled robotics research.
- Write ideas and results in a clear, technically appropriate way.

Textbooks and Reading Material

No textbook, readings will be available as links to online journals.

Article Reviews

You will do five article reviews during the semester: two of your choice; one for the article you present; and two reviews of classmates' final projects.

The first two, self-selected reviews, can be done at any time up until Thurs, Nov 14, 2019. For each of these reviews, you will review one of the articles we are reading in class. You are encouraged to do these early in the semester. Do not leave them until the last days of class. The review is due at the beginning of the class during which the article is presented. The third is due 24 hours before the class during which you present an article (see below). The final reviews, of two of your peers' final papers, is due Tues, Nov 26, 2019, at the beginning of class.

Writing detailed and constructive reviews of academic articles is a crucial part of being a researcher in both academia and industry. You should write your review as if you were evaluating the article for publication (e.g. the review is for a journal editor and the author, although I expect your reviews to be more detailed than a standard review). There is no standard format for the review, although it should be detailed and typically around 4 pages double spaced (this is somewhat longer than you would submit for most article reviews). It should include, but by no means be limited to:

- A brief summary of the paper and technical approach;
- A brief summary of your review (e.g., identify the key strength and weakness);
- Discussion of the assumptions made in the paper;
- Questions the paper raises for you and the community;
- Strengths and weaknesses of the paper (you should make sure to quantify these, for instance, "a minor weakness of this paper is" or "a significant strength");
- Constructive feedback (e.g. areas to be expanded or improved);
- Analysis of the related work and its completeness (or lack of);
- Comments on the quality of the organization, writing style, and grammar;
- Would you recommend this article for publication?

Although the articles we are reading have already been published, pretend that it has not for this review (there is always room for improvement) and that you have been asked to review the paper. Remember to give positive feedback that will allow the authors to improve the article, but do not be afraid to critical.

Article Presentation and Write-up

In this course you (and possibly a partner) will lead one class in the discussion and analysis of an article. I will assign presentation dates in the first two weeks of the course.

In addition to the presentation, you will need to write an article review (see above for format) for the article you are presenting. If you are presenting with a partner, each partner is responsible for writing a separate, individual article review. This is due 24 hours before the start of the class you are leading. I also encourage you to meet with me to discuss the article ahead of time. Please do not wait until the last minute to read the article.

You can prepare slides for the presentation or use the board. At the start of the class, each person in the class will be asked to give some comments about the article (e.g. comment on the strength, weakness, assumptions, etc. of the article). This can be used as the basis for discussion, although summarizing the article at this point is also useful. Also, it is often useful to follow up on references in the paper and present more background than is present in the paper. Similarly, it is often interesting to look to see if there are any more recent papers that build on the original paper.

I will suggest a paper for you to present, however, if there is an alternative, related paper you would like the class to read instead, that is possible as well. Just please make sure to talk with me at least 4 days in advance to give the class sufficient time to read the article.

You will be graded on both your presentation and article review writeup. In addition, the students in the class will fill out evaluations on your article presentation that I will summarize and use to aid me in grading.

Final Project*

The final project for this course is to write an article related to the fields of software engineering and robotics. The final writeup is due Tues, Nov 19, 2019, at the start of class in ACM two-column format (`https://www.acm.org/publications/proceedings-template`). This is an opportunity to combine your research with what you have learned in this course. This can be an individual or small group project. Project proposals will be due early in the semester, and you will be working on your project throughout the semester. There will also be periodic project status reports. Also, you will put together a short video describing your project. More details on the proposal and project will be discussed in class.

We will use the last week of class (and if necessary, the final examination timeslot) to do final project presentations. You will be graded on your article, presentation, and video. You will also receive article reviews from your classmates prior to your presentation.

*591 students' final projects can optionally be a replication study, whereas 791 students will be expected to conduct original research and produce a final project report worthy of submission to a conference or workshop in software engineering or robotics.

Final Project Reviews

Each student will review two of their classmates' final project articles. I will assign the project you will review. The review will be single-blind (the author will not know who reviewed it) and should follow the guidelines of the article review described above. The review is due Tues, Nov 26, 2019, at the start of class.

Weekly Course Schedule

See the attached course schedule (subject to change, updated frequenty—please refer to the course website for the most up-to-date version).

Course Structure and Grading

All assignments are due via email to [*JPO: add NCSU email*]. Please include 'CSC 791' or 'CSC 591' at the beginning of the subject line. I will respond to you to acknowledge that I received the assignment. If you do not receive a response from me within 24 hours, assume I did not receive your assignment and try to contact

me again. Assignments and due dates will be announced in class. Unless otherwise noted, all assignments are due at the beginning of class on the day they are due.

Your grade will be determined by a number of components:

PERCENTAGE	ASSIGNMENT
20%	Class Participation
20%	Article Reviews
20%	Article Presentation and Write-up
30%	Final Project*
10%	Final Project Review

The final letter grade will be based on the final percentage as follows:

A+ $\leq$ 97% < A $\leq$ 93% < A− $\leq$ 90% < B+ $\leq$ 87% < B $\leq$ 83% < B− $\leq$ 80% < C+ $\leq$ 77% < C $\leq$ 73% < C− $\leq$ 70% < D+ $\leq$ 67% < D $\leq$ 63% < D− $\leq$ 60% < F

REG 02.50.03 describes the grade point interpretation of letter grades.

*591 students' final projects can optionally be a replication study, whereas 791 students will be expected to conduct original research and produce a final project report worthy of submission to a conference or workshop in software engineering or robotics.

Assignment Lateness Policy

Assignment deadlines will be hard. Article reviews submitted after class will not be graded. Other assignments, where applicable, will be assessed a 20% per-day late penalty, with a maximum of 3 days. Unless the problem is apocalyptic, excuses will not be effective in reducing the penalty. Students with legitimate reasons who contact the professor before the deadline may apply for an extension.

Absences

Students are expected to attend all classes. Any absences will lead to a reduction in your class participation grade. If you need to be absent (e.g., to attend a conference), please discuss this with me as early as possible in the semester. Not showing up for class is not an excuse for not turning in an assignment or giving a presentation.

The university policy on excused absences will be observed (see REG 02.20.03). Late submission of non-exam assignments due to excused absences is not subject to the policies on late assignments.

Academic Integrity Policy

Recall the Pack Pledge: *"I have neither given nor received unauthorized aid on this test or assignment."* (source: `https://studentconduct.dasa.ncsu.edu`)

Students in this class are welcome and strongly encouraged to discuss assignments outside of class, including to have other students review and provide feedback on all aspects of presentations or project progress. However, the actual work of the assignment should be done by the assigned parties. Article reviews should be completed individually before class. Students are explicitly forbidden from copying the work of others (with or without superficial modification). This includes Internet or text sources for code or prose.

I expect that the students in this course will complete assignments in an honest fashion.

However, should an incident arise where I believe academic misconduct has occured, the university, college, and department policies against academic dishonesty will be strictly enfofficed. You may obtain copies of the NCSU Code of Student Conduct from the Office of Student Conduct. The instructor has a zero tolerance policy for violations of academic integrity. which include but are not limited to plagiarism and illegal collaboration. If a student is in doubt about the conduct of themselves or others, the instructor welcomes questions about this

policy. In this case, it is far better to ask permission, as there will not be forgiveness of academic misconduct. The penalties for academic misconduct will include assigning at least a negative grade and refering the student to the appropriate University bodies for possible further action.

It is the understanding and expectation of instructor that the students signature on any assignment means that the student neither gave nor received unauthorized aid. For additional information, visit `https://studentconduct.dasa.ncsu.edu`

Resources for Support

The instructor's goal is to help students gain a clear understanding of the course material, to foster a deep interest in the topic of software for robotics, and develop the basic research skills essential to a career at the frontiers of technology. With software for robotics, crucial understanding often relies on subtleties. Accordingly, it is natural for students to struggle both with the content of this course and with requisite background material.

To this end, the instructional staff are providing a number of mechanisms for support. These include:

- **Piazza** The course will feature a Piazza message board, available here. This should be your first go-to resource for any questions about course structure, deadlines, class material, or anything else that could possibly be relevant to other students. Note that active participation in Piazza will enhance your participation grade. The instructional staff receives emails from Piazza, so any questions posted to Piazza will be addressed as fast or faster than those sent by email. Piazza will be the main form of out-of-class communication.
- **Office Hours** The instructor will hold office hours weekly. Students are highly encouraged to come to office hours with the instructor to discuss doubts about course material, cofficerns about course performance, consult on the course project, or to discuss computer security beyond what can be discussed in class. No appointment is needed to attend office hours. The instructor is also available by appointment outside of office hours when meeting is impractical.
- **Email** The instructional staff strongly requests that you limit individual emails to communications regarding private questions (like grade concerns), appointment and make up exam requests, and other communications that are not suitable for Piazza. Note that emails that are of a general nature will be posted anonymously to Piazza on a students behalf. To ensure that student emails receive a high priority, students should place the string [CSC XXX] somewhere in the subject line. If at any time you have constructive suggestions about how to improve the course, feel free to share them with the instructor during office hours or via an email.

NCSU Writing Center

The North Carolina State University Graduate Writing Center can provide you with meaningful support as you write for this class as well as for every course in which you enroll. Consultants are available to talk with you as you plan, draft, and revise your writing. Please check the Writing Center website (`https://tutorial.dasa.ncsu.edu/wsts-overview-programs/gwc/`) for locations, hours, and information about scheduling appointments.

Statement on Identity

I endeavor to foster an environment of respect, including addressing each other with our preferred names and pronouns. If you would like to be called by a different name or pronoun other than what is in the directory, please let me know (in person or email). Also, I seek to pronounce your name correctly and would be grateful if you notify me of the correct pronunciation.

Statement on transportation

Students have to provide their own transportation for any and all class related trips.

Statement on safety and risk assumption

This course does not require activities that pose physical risk to students.

Statement for students with disabilities

Reasonable accommodations will be made for students with verifiable disabilities. In order to take advantage of available accommodations, students must register with Disability Services for Students at 1900 Student Health Center, Campus Box 7509, 919-515-7653. For more information on NC States policy on working with students with disabilities, please see the Academic Accommodations for Students with Disabilities Regulation (REG 02.20.01).

Statement on Class Evaluation

Online class evaluations will be available for students to complete during the last 2 weeks of the semester for full semester courses and the last week of shorter sessions. Evaluations then become unavailable at 8am on the first day of finals. For full semester courses, evaluations will be available: 8am November 25th, 2019 through 8 am December 9th, 2019.

N.C. State University Polices, Regulations, and Rules (PRR)

Students are responsible for reviewing the PRRs which pertain to their course rights and responsibilities. These include: Equal Opportunity and Non-Discrimination Policy Statement, Office for Institutional Equity and Diversity, Code of Student Conduct, and Grades and Grade Point Average.

MODULE	WEEK	TOPICS / CONCEPTS	DATE	READING	SOURCE
	1	Course Logistics and Administration, Policies & Grading, Introduction, Semester Plan, Overview (and fun videos).	22-Aug	COURSE INTRO	
Robotic Software Architecture & Robotic Software Analysis	2	Reactive Software, Modularity, Integrating Hardware & Software, Middleware	27-Aug	Robotic systems architectures and programming. Sections 8.1-8.2	Kortenkamp, David, Reid Simmons, and Davide Brugali. In Springer Handbook of Robotics, pp. 283-306. Springer, Cham, 2016.
			29-Aug	Robotic systems architectures and programming Sections 8.3-8.6	Kortenkamp, David, Reid Simmons, and Davide Brugali. In Springer Handbook of Robotics, pp. 283-306. Springer, Cham, 2016.
	3	"Robot Operating System", Message Passing, Autonomy Hierarches, Safety Layers	3-Sep	ROS: an open-source Robot Operating System, How to Read a Scientific Article	Quigley, Morgan, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y. Ng. In ICRA workshop on open source software, vol. 3, no. 3.2, p. 5. 2009. Rice University, 2004. Online.
			5-Sep	Designing autonomous robots	Bensalem, Saddek, Matthieu Gallien, Félix Ingrand, Imen Kahloul, and Nguyen Thanh-Hung. IEEE Robotics & Automation Magazine 16, no. 1 (2009): 67-77.
	4	Robotic Computation Graphs, Static vs Dynamic Analysis, Code Quality Metrics, Testing Hybrid Software/Hardware	10-Sep	Static-Time Extraction and Analysis of the ROS Computation Graph	Santos, André, Alcino Cunha, and Nuno Macedo. In 2019 Third IEEE International Conference on Robotic Computing (IRC), pp. 62-69. IEEE, 2019.
			12-Sep	A framework for quality assessment of ROS repositories	Santos, André, Alcino Cunha, Nuno Macedo, and Cláudio Lourenço. In 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 4491-4496. IEEE, 2016.
	5	Physical Unit Inconsistencies, Abstract Type Inferrence for bug detection, Type Checking vs Automated Software Testing	17-Sep	Lightweight detection of physical unit inconsistencies without program annotations	Ore, John-Paul, Carrick Detweiler, and Sebastian Elbaum.In Proceedings of the 26th ACM SIGSOFT International Symposium on Software Testing and Analysis, pp. 341-351. ACM, 2017.
			19-Sep	Phys: probabilistic physical unit assignment and inconsistency detection	Kate, Sayali, John-Paul Ore, Xiangyu Zhang, Sebastian Elbaum, and Zhaogui Xu. In Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, pp. 563-573. ACM, 2018.
Robotic Capabilities & Challenges:	6	Project ideas (Meta "Writing Academic Papers")	24-Sep	Discuss Project ideas in Groups (Bring Draft of Proposal!) How to Write an Academic Paper, English Communication for Scientists: Writing Scientific Papers	R. A. Day, ASM News, vol. 42, pp. 486-494, 1975.
		Reasoning and Higher-Level Planning, Hypergraphs, "Old-School AI"	26-Sep	Deliberation for autonomous robots: A survey	Ingrand, Félix, and Malik Ghallab Artificial Intelligence 247 (2017): 10-44.
	7	Sensors and Algorithms for Autonomous vehicles. Obstacle avoidance, costmaps. Differences between cars and roombas	1-Oct	Stanley: The robot that won the DARPA Grand Challenge	Thrun, Sebastian, Mike Montemerlo, Hendrik Dahlkamp, David Stavens, Andrei Aron, James Diebel, Philip Fong et al. Journal of field Robotics 23, no. 9 (2006): 661-692.
			3-Oct	A robust layered control system for a mobile robot	Brooks, Rodney. IEEE journal on robotics and automation 2, no. 1 (1986): 14-23.
	8	Flying Robots. Path planning, algorithmic costs and techniques for planning in 3D, motion primitives, unstable systems.	8-Oct	A survey of motion planning algorithms from the perspective of autonomous UAV guidance	Goerzen, Chad, Zhaodan Kong, and Bernard Mettler. Journal of Intelligent and Robotic Systems 57, no. 1-4 (2010): 65.
			10-Oct	NO CLASS (FALL BREAK)	
	9	Surgical Robots, Shared Autonomy, Software Reliability and the Therac-25 Disaster	15-Oct	Evolution of autonomous and semi-autonomous robotic surgical systems: a review of the literature	Moustris, George P., Savas C. Hiridis, Kyriakos M. Deliparaschos, and Konstantinos M. Konstantinidis. The international journal of medical robotics and computer assisted surgery 7, no. 4 (2011): 375-392.
			17-Oct	Present Project Status to Class (10 min on Related Work)	
	10	Swarm softare architectures, challenges of distributed systems, testing and validating swarm software.	22-Oct	Swarm robotics: A formal approach Sections 1.1 (Introdution) and 1.7 (Implementations)	Hamann, Heiko. Swarm robotics: A formal approach. Springer International Publishing, 2018.
			24-Oct	Swarm robotics: A formal approach Sections 3.1-3.9	Hamann, Heiko. Swarm robotics: A formal approach. Springer International Publishing, 2018.

Advanced Techniques for Robotic Software Reliabilty	**11**	**Model Checking of hybrid systems, repair and adaptation in robotic models.**	29-Oct	Combining Model Checking and Runtime Verification for Safe Robotics	Desai, Ankush, Tommaso Dreossi, and Sanjit A. Seshia. In International Conference on Runtime Verification, pp. 172-189. Springer, Cham, 2017.
			31-Oct	Model-Based Adaptation for Robotics Software	Aldrich, Jonathan, David Garlan, Christian Kästner, Claire Le Goues, Anahita Mohseni-Kabir, Ivan Ruchkin, Selva Samuel et al. IEEE Software 36, no. 2 (2019): 83-90.
	12	**Deep Learning & Synthesis. Role and organization of deep learning in robotic software.**	5-Nov	The limits and potentials of deep learning for robotics	N. Sunderhauf, O. Brock, W. Scheirer, R. Hadsell, D. Fox, J. Leitner, B. Upcroft, P. Abbeel, W. Burgard, M. Milford, and P. Corke, The International Journal of Robotics Research, vol. 37, no. 4-5, pp. 405-420, Apr. 2018.
			7-Nov	Synthesis of nonlinear continuous controllers for verifiably correct high-level, reactive behaviors	DeCastro, Jonathan A., and Hadas Kress-Gazit. The International Journal of Robotics Research 34, no. 3 (2015): 378-394.
	13	**Role of simulation in robotic system testing, simulation gap, coarse vs. fine-grained simulation, Training & Learning in simulation**	12-Nov	Sim-to-Real: Learning Agile Locomotion For Quadruped Robots	Tan, Jie, Tingnan Zhang, Erwin Coumans, Atil Iscen, Yunfei Bai, Danijar Hafner, Steven Bohez, and Vincent Vanhoucke. arXiv preprint arXiv:1804.10332 (2018).
			14-Nov	Crashing simulated planes is cheap: Can simulation detect robotics bugs early?	Timperley, Christopher Steven, Afsoon Afzal, Deborah S. Katz, Jam Marcos Hernandez, and Claire Le Goues. In 2018 IEEE 11th International Conference on Software Testing, Verification and Validation (ICST), pp. 331-342. IEEE, 2018.
	14	**Mixed Reality in Robotic Testing**	19-Nov	Evaluating the Effectiveness of Mixed Reality Simulations for Developing UAV Systems	Chen, Ian Yen-Hung, Bruce MacDonald, and Burkhard Wünsche. In International conference on simulation, modeling, and programming for autonomous robots, pp. 388-399. Springer,
Long-Term Autonomy		**Challenges in undewater and space robotics, watchdog mechanisms, resiliant and redundant software architecture**	21-Nov	Toward deep space humanoid robotics inspired by the NASA Space Robotics Challenge	Tanaka, Yoshimaru, Hyunhee Lee, Dylan Wallace, Youngbum Jun, Paul Oh, and Masayuki Inaba. In 2017 14th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI), pp. 14-19. IEEE, 2017.
	15		26-Nov	A Survey on Intermediation Architectures for Underwater Robotic (DUE: 2 Reviews of Peer's Final Project Reports)	Li, Xin, José-Fernán Martínez, Jesús Rodríguez-Molina, and Néstor Martínez. Sensors 16, no. 2 (2016): 190.
			28-Nov	NO CLASS (THANKSGIVING BREAK)	
	16		3-Dec	Project Presentations	
			5-Dec	Project Presentations	
	17		10-Dec	Project Presentations (8:00a-11a)	