

Course Syllabus

TABLE OF CONTENTS

- 1 **About**
- 2 **Modality:**
 - a **On-campus mode:**
 - b **UNITE mode:**
- 3 **Discussion Forum:**
- 4 **Topics and Course Structure**
- 5 **Prerequisites:**
- 6 **Textbook:**
- 7 **Guided Programming Projects:**
- 8 **Open-ended Final Project:**
 - a **Use of AI Resources on Projects**
- 9 **Quizzes:**
 - a **Use of AI Resources on Quizzes:**
- 10 **Grading Policy:**
 - a **Late Day Tokens**
- 11 **Collaboration Policy**

About

The goal of this course is to introduce students to robotics principles, covering key topics such as 3D transformations, robot kinematics, forward and inverse kinematics, path planning, configuration spaces, sampling-based planning, basic motion control algorithms, and state estimation for mobile robots, which includes introduction to mapping, localization, and SLAM. Students will gain hands-on experience in programming robots in the **Java threejs** environment. In a later project, we plan to have a real-world robot challenge. There will be an open-ended final project where students can apply their skills acquired throughout the semester to explore new

ideas. They will present their projects to a wider audience through a poster presentation with videos and demos.

This course builds on and is indebted to materials from -

- [Prof. Chad Jenkins](#) (Univ of Michigan) and the staff of [autorob.org](#)
- [Prof. Dieter Fox](#) (Univ of Washington),
- [Prof. Cyrill Stachniss](#) (Univ of Bonn),
- [Prof. Nikolaos Papanikolopoulos](#) (Univ of Minnesota),
- [Prof. Junaed Sattar](#) (Univ of Minnesota)

Modality:

This section of the course CSCI5551 is scheduled to be held both on-campus and on UNITE.

On-campus mode:

The On-campus class will be in Keller Hall 3-125 on Mondays and Wednesdays 1:00PM-2:15PM Central Time. Students registered for the on-campus mode of the class are required to attend lectures in-person, as long as health and safety concerns and university regulations do not prevent them from doing so. In any case, it is highly recommended to keep the instructor informed of any circumstances that might be of concern for administering the course. The students enrolled in the on-campus section of this course will be able to access the lecture video archives on a **10 day delay** on UNITE portal (**not immediately**). Lecture slides will be made available on the [calendar](#) before each lecture.

UNITE mode:

The UNITE mode will live-stream lectures during the in-person session. The UNITE-enrolled students will have access to the lecture videos on their UNITE portal.

There will be no exams (i.e., midterms or final) in this year's offering; all evaluation will thus take place in the form of programming projects, quizzes, and final project presentations.

Discussion Forum:

The [Ed Stem](#) discussion forum is available for discussion of course materials including lectures and projects. **Any discussion of quizzes and verbatim code on the Ed Stem forum must be posted privately.**

Topics and Course Structure

- Transformations
- Forward Kinematics
- Inverse Kinematics
- Jacobians
- Path Planning
- Motion Control
- Reactive Controllers
- Configuration Spaces
- Sampling-based Planning
- Potential Fields
- Mobile Robot State Estimation
- Robot Programming

Prerequisites:

- Strongly encouraged prerequisites:
 - Linear Algebra, Calculus, and Probability
 - Programming fluency in data structures in a classical programming language is essential.
 - Prior experience with the **Python programming language** or JavaScript is strongly recommended.

Textbook:

There is no required textbook for this course, however optional readings will be suggested from the textbook:

- [1] (Optional) K. Lynch, F. Park, "Modern Robotics: Mechanics, Planning, and Control", Cambridge University Press New York, NY, USA, 1st Edition, 2017 (ISBN-13:9781107156302).
- [2] (Optional) J. Craig, "Introduction to Robotics: Mechanics and Control," Pearson Prentice Hall, NJ, 4th edition, 2017 (ISBN-13: 9780133489798).
- [3] (Optional but recommended for ROS) Wyatt Newman. "A Systematic Approach to Learning Robot Programming with ROS". Chapman & Hall. Print ISBN: 9781315152691, 131515269X, eText ISBN: 9781498777872, 1498777872, Edition: 1st. This is freely available through the UMN libraries to students and staff here - [link](#).

- [4] (Optional) Sebastian Thrun, Wolfram Burgard and Dieter Fox. "Probabilistic Robotics," The MIT Press; Intelligent Robotics and Autonomous Agents series edition (August 19, 2005) (ISBN:9780262201629).

Guided Programming Projects:

You will complete 7 guided programming **projects** over the course of the semester. Projects 1-6 will use JavaScript webbased simulation environment. Please find the schedule for the same at **calendar**. Project 7 will be a group project aimed toward a real-robot challenge. Number of students in a group is **TBD**. This same group will be retained for the Open-ended Final Project.

Open-ended Final Project:

You will have opportunity to explore your ideas in an open-ended setting where you can use the tools you have built over the semester to do a final project. You will have opportunity to present your project to wider audience (students, faculty at the U) via poster presentation. This will be a group project with the same group as in for Project 7. Number of students in a group is **TBD**.

Use of AI Resources on Projects

The project and assignments must be your own original work. However, the use of AI tools such as ChatGPT and GitHub Copilot to assist with portions of the implementation are permitted. Although AI can be a powerful tool for a programmer, their use comes with additional responsibilities:

- The bar for getting an A in this class is higher than just "getting it working."
- If you integrate AI-generated code, you are responsible for understanding how it works. Course staff will have many opportunities to check-in with the students over the course leading to requesting you to walk us through the project structure and code.
- You should expect that the course staff will ask you questions about how the implementation works, especially for complicated functions, and you should be able to provide an explanation that shows you have a thorough understanding of all code in your project.

You are expected to read the University policy listed **here**.

Quizzes:

Throughout the semester, there will be a total of 12 quizzes administered via **Gradescope**. They will be released on Tuesdays 6pm and will be available till 12pm on Wednesdays. Each quiz will have 2 questions for 0.5 pts each. These questions will be within the scope of previously covered lectures and graded projects. They are designed to be answered in less than 5 mins each after

you have reviewed the previous week's course material. **When you start the quiz, you will have 20 mins to answer them.**

Your final grade for the quizzes will be based on the best 10 quizzes out of 12 in case you miss any of the quizzes. No additional quizzes will be provided.

Use of AI Resources on Quizzes:

Use of AI tools is **NOT PERMITTED** on quizzes and will be considered an academic integrity violation. This includes, but is not limited to, services such as ChatGPT, Claude, Bard, Bing Chat, and GitHub Copilot. At the beginning of each quiz, you will need to complete an honor statement affirming that the answers on the quiz are the result of your own work using only the allowed course materials. You are expected to read the University policy listed [here](#).

Grading Policy:

Course grades will be determined according to the following criteria:

- Project 1: 10%
- Project 2: 10%
- Project 3: 10%
- Project 4: 10%
- Project 5: 10%
- Project 6: 10%
- Project 7: 15%
- Final Project: 15%
 - Project proposal slides + presentation: 2%
 - Final project video: 5%
 - Poster presentation (evaluation by judges): 3%
- Best 10 out of 12 In-class Quizzes: 10% (1% each)

Late Day Tokens

P1-6: 3 total late day tokens are available for students. After which a 25% daily penalty will be applied for late submissions. P7 and Open-ended final projects: No late days are available for these group projects.

The grading in this course is on an absolute scale. This means that the performance of others in the class will not affect your grade. Letter grades will be assigned using the following scale:

- $A \geq 93.0\%$
- $A- \geq 90.0\%$ and $< 93.0\%$
- $B+ \geq 87.0\%$ and $< 90.0\%$
- $B \geq 83.0\%$ and $< 87.0\%$
- $B- \geq 80.0\%$ and $< 83.0\%$
- $C+ \geq 77.0\%$ and $< 80.0\%$
- $C \geq 73.0\%$ and $< 77.0\%$
- $C- \geq 70.0\%$ and $< 73.0\%$
- $D+ \geq 67.0\%$ and $< 70.0\%$
- $D \geq 60.0\%$ and $< 67.0\%$
- $F < 60.0\%$

For S/N grading, a satisfactory grade (S) requires a grade of 70.0% or above.

Collaboration Policy

The free flow of discussion and ideas is encouraged. **But, everything you turn in must be your own work**, and you must note the names of anyone you collaborated with on each problem and cite resources that you used to learn about the problem. If you have any doubts about whether a particular action may be construed as cheating, ask the instructor for clarification before you do it. Cheating in this course will result in a grade of F for course and the **University policies** will be followed.

No code can be communicated, including verbally. Explicit use of external sources must be clearly cited. **DO NOT post your code in public repositories (Github, Gitlab, Google-drive, Website, etc).** If we find it, your grades will be retracted.