Time Commitment

This is a graduate level course. Estimated 10~20 hours of study time per week.

Prerequisites
For content covered in the course

College-level introductory linear algebra (vector spaces, linear systems, matrix decomposition)

College-level introductory calculus (partial derivatives, function gradients)

For programming projects, please note:
Basic knowledge of computer programming (variables, functions, control flow, etc.) will be needed.
We will use the Robot Operating System (ROS) which runs on Ubuntu Linux. Basic familiarity with a Linux-like environment will be needed.

Projects will be carried out in the Python language. All starter code and class examples will be in Python.

Programming the assignments in C++ is accepted, but not supported by the course staff. In other words, our technical platforms are capable of compiling and running C++ code, and our grading scripts will be able to grade their output, but the course staff will not be able to answer language-specific questions or problems. If any starter code is provided, it will be in Python, and it will be your responsibility to write equivalent C++ code.

Course Outline
Week 1: Introduction to Robotics, Robotics and AI, Introduction to ROS, Project 1 released

Week 2: 2D Transforms, Homogenous Coordinates, 3D Transforms, Thinking about Transforms, Transform Inverse, Rotation Representations, Transforms in ROS, the TF Library, Project 2 released

Week 3: Robot Arms Introduction, Kinematic Chains, Forward Kinematics: URDF, Forward Kinematics: Analytical Methods, DH Parameters, Forward Kinematics:DH Examples, Project 3 released

Week 4: Analytical IK, Robot Examples, Robot Workspaces and IK Solutions, Homework 1 released

Week 5: Differential Kinematics: Jacobian Definition and Analytical Computation, Singularities, Full Kinematics: Robot Examples, Homework 2 released

Week 6: Study week

Week 7: Numerical Jacobian Computation, Cartesian and Null Space Control, Porject 4 released

Week 8: Motion Planning: Configuration Space vs. Task Space, Stochastic Motion Planning, Project 5 released

Week 9: Mobile Robots Introduction, Mobile Robots Kinematics: Differential Drive, Other Kinematics, Path Planning for Mobile Robots

Week 10: Course Recap, Things We Have Not Covered, Robotics and AI Revisited

Grading
Programming Assignments (70%). The course has 5 programming assignments in ROS, with contributions to the final grade as follows:

- Project 1 (released in Week 1): 5%
- Project 2 (released in Week 2): 15%
- Project 3 (released in Week 3): 15%
- Project 4 (released in Week 6): 15%
- Project 5 (released in Week 8): 20%

Please note that all Projects will be due on May 3rd.
Homeworks (20%): The course has 2 homeworks, for which you will be expected to carry out pencil-and-paper derivations then input the results as answers to multiple-choice questions. Their contributions to the final grade are as follows:

- Homework 1 (released in Week 4): 10%
- Homework 2 (released in Week 5): 10%

Final quiz (10%): There will be a final quiz at the end of the course, with multiple-choice questions covering the concepts presented in the lectures. We will release an example quiz which will contain questions of the same type that you can expect to encounter on the final.

Passing Grade To pass the course, you must score 60% (60 total points) or above.

Suggested Readings
There is no required textbook for the course. The following recommended textbooks are some of the most widely used intro-level books by the Robotics community:

Lorenzo Sciavicco and Bruno Siciliano, Modelling and control of robot manipulators, Springer.

Saeed B. Niku, Introduction to Robotics, Wiley.

Mark W. Spong, Seth Hutchinson and M. Vidyasagar, Robot Modeling and Control, Wiley.

Roland Siegwart, Introduction to Autonomous Mobile Robots, MIT Press.

Peter Corke, Robotics, vision and control : fundamental algorithms in MATLAB, Springer.

Honor Code
Academic Honesty Policy You are required to read, and understand the following agreement regarding Academic Honesty. Each student is sole owner of his own code and work and must NOT: · Submit work that is not original. · Publish code or solutions online. · Post the course questions on forums including stack overflow. · Submit someone else's work, or a modification of that work, with or without that person's knowledge. · Allow someone else to submit his/her work, or a modification of that work. · Solve as a group a quiz or project. All coursework is to be done by the student working alone. · Contract course work out to others. · Plan or execute with another student a cooperative subterfuge during an exam. · Make use of unauthorized material during an exam. Project assignments will be checked with plagiarism detection software. Thank you for abiding by these rules. Doing so will ensure the experience is fair to everyone taking this class or the future sessions of this class.

Enjoy and wishing you all a wonderful learning journey!