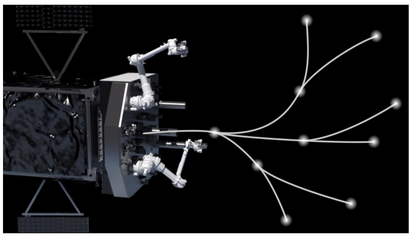
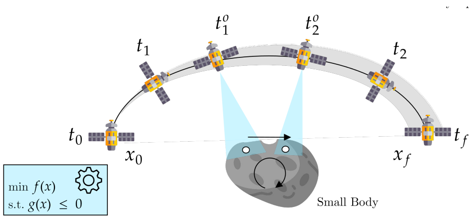
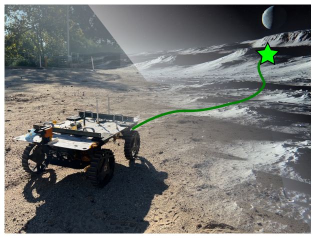
EN.530.626: Optimal Control for Space Systems

Instructor: Prof. Abhishek Cauligi

Semester: Fall 2026







Course Description

Trajectory design and control for aerospace systems encompasses a broad range of system dynamics, physical constraints, and other safety considerations. Optimal control offers a powerful paradigm to solve such problems and this course introduces the theoretical and practical foundations of optimal control as applied to aerospace and robotic applications. In particular, a strong emphasis is placed on real-time planning and control via the use of on-board numerical optimization and students will apply theoretical insights from trajectory optimization and model predictive control for developing real-time controllers. Students will apply this theory to practice through coding implementations in Python and evaluation in simple simulation environments, with applications including planetary rover path planning, rocket powered descent guidance, and spacecraft controls. Finally, a course project will be included to allow students to gain further experience on an algorithm or application of their choice.

Key words: Optimal control, trajectory optimization, model predictive control, interior point methods, sequential quadratic programming, sampling-based control, stochastic optimal control, differentiable optimization, model-based reinforcement learning.

Prerequisites: A strong foundation in linear algebra (e.g., `EN.530.616 Introduction to Linear Systems Theory`) and differential equations (e.g., `EN.530.761 (01) Mathematical Methods of Engineering 1`) and experience with a high-level programming language such as Python or Julia will be assumed. This course assumes otherwise assumes no background in optimization.

Learning Outcomes

At the end of the semester, students will be able to:

- Formulate trajectory optimization and model predictive control problems for aerospace and robotic trajectory design problems.
- Develop custom quadratic program solvers using interior point methods using automatic differentiation libraries such as `jax`.
- Understand how different optimization algorithms are suitable based upon the system dynamics, constraints, and geometric structure.
- Leverage off-the-shelf nonlinear optimization interfaces such as `acados` [1] and `casadi` [2].
- Connect optimal control and reinforcement learning using differentiable optimization theory.

Instructor

Abhishek Cauligi
Email: `cauligi@jhu.edu`

Teaching Assistants

Lectures

TBD.
Lectures will be recorded and posted online.

Assignments

- There will be short quizzes held every other week in class that are written in-person and closed note. These quizzes should take approximately 15 - 20 minutes and are intended to provide students the opportunity to track their theoretical understanding of course material.

Office Hours

Office hours will begin in the second week of the semester.
Prof. Cauligi's office hours TBD.

Grading Policy

- Assignments: 25%
- Quizzes: 40%
- Final Project: 30%
- Participation: 5%

Course Policies

Late Assignments: Late submissions will be penalized by 10% per day and the final deadline for consideration is two days after the assignment due date.

Attendance: Regular attendance is expected and in-class attendance will be taken.

Textbook

There is no required textbook for this class. However, the following resources may be useful:

- J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed. Springer, 2006

Ethics

Academic Integrity: All students must adhere to university policies on plagiarism and cheating. Students are allowed to collaborate to complete problem sets, however all submitted material (including code, proofs, derivations, among others) must have been independently completed. If a student does indeed collaborate on a homework assignment, then they must include this detail and the nature of the collaboration as a part of the submitted assignment. Students can find further information on academic integrity policies for undergraduates here and for graduate students here. Any violations of these policies as determined by the instruction team will subsequently dealt with appropriately [1]

Course Policy on Generative AI: Students are permitted to use AI assistants (e.g., chatbots, large language models, etc.) to assist them in the competition of homework and final project assignments. However, students are not permitted to directly submit and claim as their own any material that is substantially copied over from the output of such an AI coding agent. This includes, but is not limited to, e.g., providing an AI assistant the homework files and asking to complete it, Suspected breaches of this policy will be treated and addressed as violations of the aforementioned academic integrity standards.

References

[1] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi: A software framework for nonlinear optimization and optimal control," vol. 11, no. 1, pp. 1–36, 2019.

[2] R. Verschueren, G. Frison, D. Kouzoupis, J. Frey, N. van Duijkeren, A. Zanelli, B. Novoselnik, T. Albin, R. Quirynen, and M. Diehl, "acados—a modular open-source framework for fast embedded optimal control," *Mathematical Programming*, vol. 14, pp. 147–183, 2022.

[3] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed. Springer, 2006.

[1] As written on April 30, 2026, the academic misconduct policy details can be found here.