UDACITY

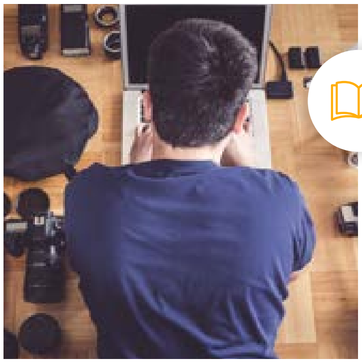
NANODEGREE PROGRAM SYLLABUS

Flying Car and Autonomous Flight Engineer

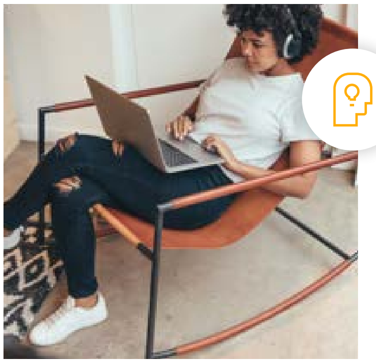
Overview

In this program, you'll learn the core concepts required to design and develop robots that fly. Working with the quadrotor test platform and our custom flight simulator, you will implement planning, control, and estimation solutions in Python and C++.

Estimated Time:
4 Months at
15hrs/week



Prerequisites:
Mathematics &
Programming



Flexible Learning:
Self-paced, so
you can learn on
the schedule that
works best for you

Need Help?
udacity.com/advisor
Discuss this program
with an enrollment
advisor.

Course 1: Introduction

In this course, you will get an introduction to flight history, challenges, and vehicles. You will learn about our quadrotor test platform, work in our custom simulator, and build your first project—getting a quadrotor to take-off and fly around a backyard!

Course Project
Backyard Flyer

In this project, you will write event-driven code in Python to get your drone to takeoff, fly a predetermined path, and land in a simulated backyard environment.

	LEARNING OUTCOMES
LESSON ONE	Welcome
LESSON TWO	Autonomous Flight
LESSON THREE	Project: Backyard Flyer
LESSON FOUR	Drone Integration

Course 2: 3D Motion Planning

Flying robots must traverse complex, dynamic environments. Wind, obstacles, unreliable sensor data, and other randomness all present significant challenges. In this course, you will learn the fundamentals of aerial path planning. You will begin with 2D problems, optimize your solutions using waypoints, and then scale your solutions to three dimensions. You will apply these skills in your second project—autonomously navigating your drone through a dense urban environment.

Course Project
3D Motion Planning

In this project, you will move beyond the backyard test grounds and fly a drone around a complex urban simulated environment. To do so, you will load a map of a real city, plan a collision-free path between buildings, and watch your drone fly above city streets.

	LEARNING OUTCOMES
LESSON ONE	Planning as Search
LESSON TWO	Flying Car Representation
LESSON THREE	From Grids to Graphs
LESSON FOUR	Moving into 3D
LESSON FIVE	Real World Planning
LESSON SIX	Project: 3D Motion Planning

Course 3: Controls

In the previous course, we implemented 3D path planning but assumed a solution for actually following paths. In reality, moving a flying vehicle requires determining appropriate low-level motor controls. In this course, you will build a nonlinear cascaded controller and incorporate it into your software in the project.

Course Project
Building a Controller

In this project, you will no longer assume vehicle actuation but rather implement your very own cascaded controller in C++. You will attempt different motions (slow, fast, slalom, etc.) and analyze performance under different conditions.

	LEARNING OUTCOMES
LESSON ONE	Vehicle Dynamics
LESSON TWO	Introduction to Vehicle Control
LESSON THREE	Control Architecture
LESSON FOUR	Full 3D Control

Course 4: Estimation

In this course, we will finish peeling back the layers of your autonomous flight solution. Instead of assuming perfect sensor readings, you will utilize sensor fusion and filtering. You will design an Extended Kalman Filter (EKF) to estimate attitude and position from IMU and GPS data of a flying robot.

Course Project
Estimation

In this project, you will implement an EKF to estimate attitude and position from IMU and GPS data of a flying robot. After doing so, you will have implemented the full-stack for a single aerial robot!

	LEARNING OUTCOMES
LESSON ONE	Introduction to Estimation
LESSON TWO	Introduction to Sensors
LESSON THREE	Extended Kalman Filters
LESSON FOUR	The 3D EKF and UKF
LESSON FIVE	Project: Estimation
LESSON SIX	GPS Denied Navigation

Course 5: Fixed Wing Aircraft

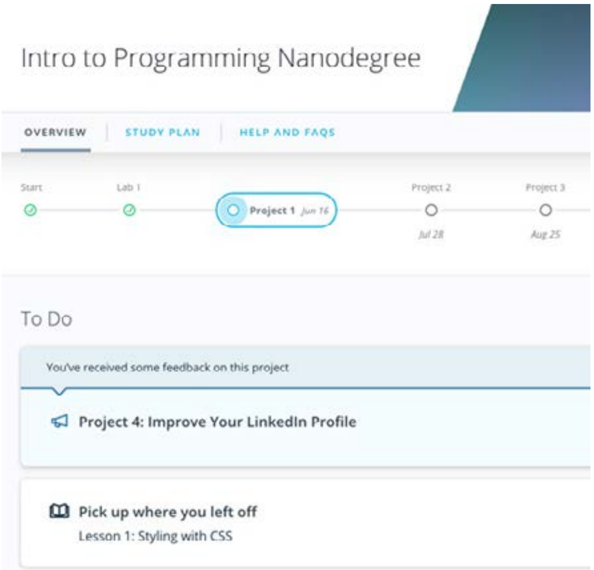
While quadrotors are the ideal test platform for aerial robotics, flying cars and other long-range aircrafts leverage the aerodynamic efficiencies of fixed-wing flight. In this course, you will learn how to adapt the concepts you've learned so far and successfully fly a fixed-wing aircraft in simulation.

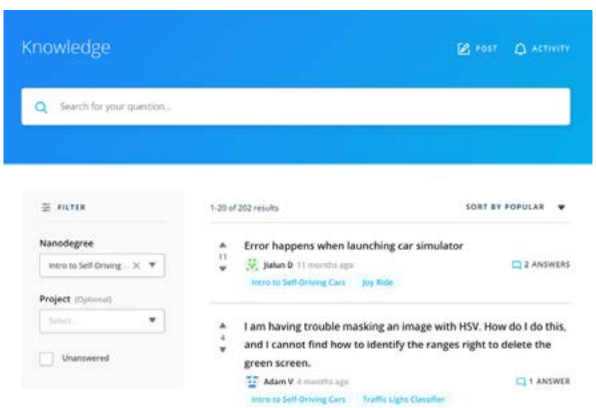
Course Project
Fixed-Wing Control

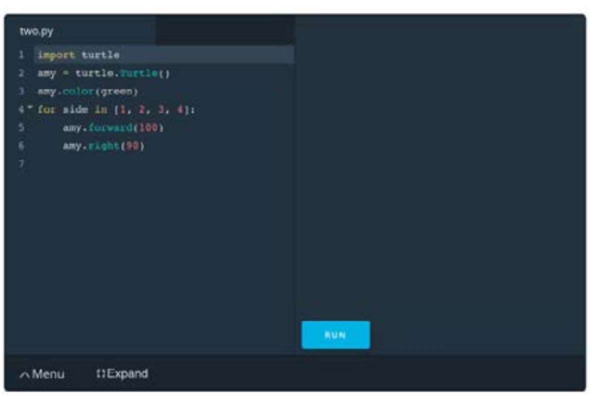
In this project you will code a fixed-wing aircraft, and then implement solutions to a significantly more challenging control problem.

	LEARNING OUTCOMES
LESSON ONE	Introduction to Fixed-Wing Flight
LESSON TWO	Lift and Drag
LESSON THREE	Longitudinal Model
LESSON FOUR	Lateral-Directional Model
LESSON FIVE	Fixed-Wing Autopilot
LESSON SIX	Project: Fixed-Wing Control

Our Classroom Experience







REAL-WORLD PROJECTS

Build your skills through industry-relevant projects. Get personalized feedback from our network of 900+ project reviewers. Our simple interface makes it easy to submit your projects as often as you need and receive unlimited feedback on your work.

KNOWLEDGE

Find answers to your questions with Knowledge, our proprietary wiki. Search questions asked by other students, connect with technical mentors, and discover in real-time how to solve the challenges that you encounter.

WORKSPACES

See your code in action. Check the output and quality of your code by running them on workspaces that are a part of our classroom.

QUIZZES

Check your understanding of concepts learned in the program by answering simple and auto-graded quizzes. Easily go back to the lessons to brush up on concepts anytime you get an answer wrong.

CUSTOM STUDY PLANS

Create a custom study plan to suit your personal needs and use this plan to keep track of your progress toward your goal.

PROGRESS TRACKER

Stay on track to complete your Nanodegree program with useful milestone reminders.

Learn with the Best

Nicholas Roy

INSTRUCTOR

Nicholas Roy is a Professor in the Department of Aeronautics & Astronautics, and a member of the Computer Science and Artificial Intelligence Laboratory, at MIT. He also founded Project Wing at X.

Angela Schoellig

INSTRUCTOR

Angela is an Assistant Professor at the University of Toronto Institute for Aerospace Studies (UTIAS), and an Associate Director of the Center for Aerial Robotics Research and Education (CARRE) at the University of Toronto.

Sebastian Thrun

INSTRUCTOR

As the founder and president of Udacity, Sebastian's mission is to democratize education. He is also the founder of Google X, where he led projects including the Self-Driving Car, Google Glass, and more.

Raffaello D'Andrea

INSTRUCTOR

Raffaello is a Professor of Dynamic Systems and Control at the Swiss Federal Institute of Technology (ETH) in Zurich. He is also the founder of Verity Studios, and a co-founder of Kiva Systems (now Amazon Robotics).

Learn with the Best

Sergei Lupashin

INSTRUCTOR

Sergei has a PhD in MechE from ETH Zurich and a BS in ECE from Cornell. He brings experience from projects such as industrial drones, self-driving cars and controls testbeds. He is a TED Fellow and founder of Fotokite.

Jake Lussier

PRODUCT LEAD

Jake is a PhD Candidate in AI at Stanford University focused on robotics, perception, and human-centered design. Prior to serving as Product Lead at Udacity, he founded an early-stage food-technology startup and consulted on flying cars.

Andy Brown

CURRICULUM LEAD

Andy has a bachelor's degree in physics from MIT, and taught himself to program after college (mostly with Udacity courses). He has been helping Udacity make incredible educational experiences since the early days of the company.

All Our Nanodegree Programs Include:

EXPERIENCED PROJECT REVIEWERS

REVIEWER SERVICES

- Personalized feedback & line by line code reviews
- 1600+ Reviewers with a 4.85/5 average rating
- 3 hour average project review turnaround time
- Unlimited submissions and feedback loops
- Practical tips and industry best practices
- Additional suggested resources to improve

TECHNICAL MENTOR SUPPORT

MENTORSHIP SERVICES

- Questions answered quickly by our team of technical mentors
- 1000+ Mentors with a 4.7/5 average rating
- Support for all your technical questions

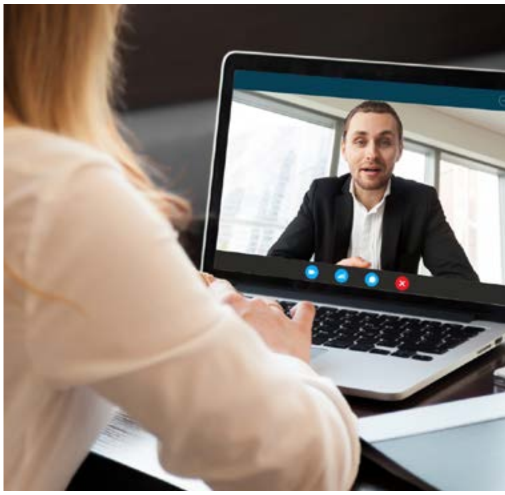
PERSONAL CAREER SERVICES

CAREER SUPPORT

- Resume support
- Github portfolio review
- LinkedIn profile optimization

Frequently Asked Questions

PROGRAM OVERVIEW



WHY SHOULD I ENROLL?

The emerging generation of flying car engineers will reimagine how we move and transform how we live. The Flying Car Nanodegree program will prepare you to be at the forefront of this technological and societal revolution.

In this program, you'll learn from world-class experts, work with cutting-edge tools, and tackle real-world challenges. You'll master techniques in planning, controls, and estimation. Most importantly, you will learn by doing, writing aircraft-ready code that you can run on your own drones.

If you're interested in flying cars, drones, autonomous systems, and/or the future of smart transportation, this Nanodegree program is for you!

WHAT JOBS WILL THIS PROGRAM PREPARE ME FOR?

As a graduate of the world's first flying car engineering program, you will be prepared for positions pertaining to aerial robotics, autonomy and mobility. Job titles in this industry vary, but include: Unmanned Aircraft Software Engineer, Software and Controls Engineer, Guidance Navigation and Controls (GNC) Engineer, Aerial Roboticist, and more.

With experience architecting sophisticated yet safe autonomous systems, you will also be prepared for jobs far beyond aerial systems, including: Autonomous Driving Engineer, Autopilot Engineer, Robotics Software Engineer, IoT Engineer, and more.

HOW DO I KNOW IF THIS PROGRAM IS RIGHT FOR ME?

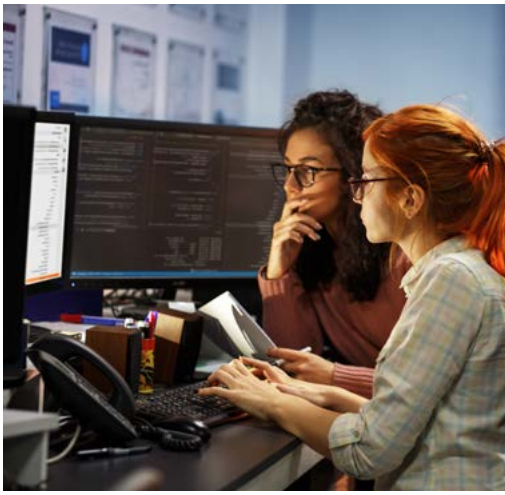
This Nanodegree program is an advanced specialized program in aerial vehicles—transformational technologies that are reshaping our future and driving amazing new innovations. If you are interested in developing the skills to build an autonomous aircraft system, and excited by the opportunity to port your code to real drones, this is the perfect way to get started.

WHAT IS THE DIFFERENCE BETWEEN THE FLYING CAR NANODEGREE PROGRAM AND THE SELF-DRIVING CAR ENGINEER NANODEGREE PROGRAM OR THE ROBOTICS SOFTWARE ENGINEER NANODEGREE PROGRAM?

The Flying Car Nanodegree program is a specialized program for aerial vehicles. The focus will be on developing the skills to build an autonomous aircraft system, with a focus on quadrotors. This means a unique emphasis on planning and autonomy for three-dimensional mobility, involving hands-on projects in simulation, with the opportunity to port your code to real drones.

The Robotics Software Engineer Nanodegree program provides an introduction to software and artificial intelligence as applied to robotics. The areas we focus on are perception, localization, path planning, deep learning, reinforcement learning, and control. These are taught using the Robot Operating System (ROS) framework.

FAQs Continued



All of the techniques required to complete the projects in the Robotics Software Engineer Nanodegree program (including machine learning) are taught as part of the program.

The Self-Driving Car Engineer Nanodegree program focuses entirely on a specialized application of robotics—it uses robotics concepts and applies them to a self-driving car. If your primary interest is in the application of robotics, machine learning, and artificial intelligence to self-driving cars, then this is the program for you. However, if you want a broader and more comprehensive robotics curriculum, with an emphasis on software engineering, then the Robotics Software Engineer Nanodegree program is your best option.

ENROLLMENT AND ADMISSION

DO I NEED TO APPLY? WHAT ARE THE ADMISSION CRITERIA?

There is no application. This Nanodegree program accepts everyone, regardless of experience and specific background.

WHAT ARE THE PREREQUISITES FOR ENROLLMENT?

Students should have prior experience with the following:

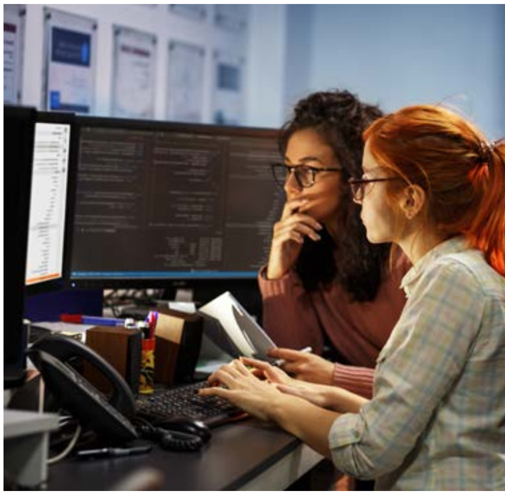
- Substantial experience programming in any language
- Intermediate-level programming experience in Python or willingness to learn
- Intermediate-level programming experience in C++ or willingness to learn (including knowledge of memory allocation, classes, and references) Basic Linear algebra Calculus (derivatives and integrals)
- Probability and statistics (mean, variance, and probability distributions)
- Basic Physics (basic mechanics including knowledge of kinematics, dynamics, and torque)
- Students will need to be able to communicate fluently and professionally in written and spoken English.

IF I DO NOT MEET THE REQUIREMENTS TO ENROLL, WHAT SHOULD I DO?

We have a number of Nanodegree programs and free courses that can help you prepare, including:

- Intro to Self-Driving Cars Nanodegree Program
- Intro to Programming Nanodegree Program
- Data Analyst Nanodegree Program
- Machine Learning Engineer Nanodegree Program
- Deep Learning Nanodegree Program
- Intro to Computer Science
- Programming Foundations with Python
- C++ for Programmers
- Intro to Statistics, Descriptive Statistics and Inferential Statistics

FAQs Continued



- Linear Algebra Refresher
- Intro to Data Science and Data Analysis
- Intro to Machine Learning
- Statistics and Probability (Khan Academy)
- Linear Algebra (Khan Academy)
- Multivariable Calculus (Khan Academy)

TUITION AND TERM OF PROGRAM

HOW IS THIS NANODEGREE PROGRAM STRUCTURED?

The Flying Car Nanodegree program is comprised of content and curriculum to support five (5) projects. We estimate that students can complete the program in four (4) months working 10 hours per week.

Each project will be reviewed by the Udacity reviewer network. Feedback will be provided and if you do not pass the project, you will be asked to resubmit the project until it passes.

HOW LONG IS THIS NANODEGREE PROGRAM?

Access to this Nanodegree program runs for the length of time specified in the payment card above. If you do not graduate within that time period, you will continue learning with month to month payments. See the **Terms of Use** and **FAQs** for other policies regarding the terms of access to our Nanodegree programs.

I HAVE GRADUATED FROM THE FLYING CAR NANODEGREE PROGRAM BUT I WANT TO KEEP LEARNING. WHERE SHOULD I GO FROM HERE?

Once you have completed the Flying Car Nanodegree program, the **Self-Driving Car Engineer Nanodegree** program and the **Robotics Software Engineer Nanodegree** program are ideal for continuing your learning.

SOFTWARE AND HARDWARE

WHAT SOFTWARE AND VERSIONS WILL I NEED IN THIS PROGRAM?

For the Flying Car Nanodegree Program, the minimum computational requirements are

- 4GB RAM
- Quad-Core i5 processor or equivalent
- 50GB free HDD space
- WiFi capability (802.11x)

We also recommend that you obtain Bitcraze's Crazyflie STEM drone bundle which you can port your code to, but it is not required.