Robotics Engineering

Steve Cline

2020-07-08

Contents

6   Robotic Types   21

7   Capstone Project   23

8   Going Further with Robotics   25

9   Course Materials and Instructions   27

"' options(tinytex.verbose = TRUE)

———————————————

Chapter 1

Prerequisites

While we will use Canvas as much as possible in this course, it is not a suitable platform for a course which relies heavily on programming. As a result, you will need to become familiar with using sites like GitHub. Familiarity with these sites will prepare you for future work in college and beyond.

For this course you will need to do the following to prepare to learn:

Locate the Course Lecture Notes section of this site. Browse through the notes to gain an understanding of what we will be covering this year.

Sign up for a GitHub account and fill out this form.

Bookmark the course GitHub Repository. This is where many of the programming resources will be found. In addition, assignments will be submitted here.

Locate our two textbooks for the course. They are both online and free. The first is an electronics textbook called Lessons in Electric Circuits. The second is a programming textbook called Think Python.

Subscribe to the course YouTube Channel.

Directions for how to do this are found in the Follow the Course YouTube Channel section of this site.

Log on to the class Jupyter Notebook server

Sign up for an OnShape account by following the instructions in the OnShape Account Setup section of this site.

Complete the WHS Mechatronics Lab Code of Conduct Agreement.

Chapter 2

Introduction

2.1   A brief Overview of the Course

Remember that engineers spend a majority of their time debugging their designs. You will too. It is not because this course is made too hard for you...it is just the way engineering works.

"They say that no plan survives first contact with implementation. I'd have to agree." – Mark Watney, Sol 40

Why begin with this quote? One of my favorite books and movies of all time is The Martian. The struggles of the main character, Mark Whatney, demonstrate the challenges facing engineers. While we will not be risking our lives on another planet, our solutions to the problems in this course, follow the same pattern as his. Where he sought to survive being the only human on another planet by engineering his way out, we will survive this course by learning to attack problems in the same way and remember not to give up. We must keep in mind, no plan will work the first time.

During this class students will explore the field of mechatronics using a variety of hands-on activities. Students begin the semester with an introduction to basic history and theory of robotics, the engineering process and tools and processes used to create robotic devices. We will introduce basic electronics concepts. Moving forward programming becomes an essential and vital element. Students program the onboard micro-processor found on a Raspberry Pi 3B+. This control board will use the Raspian OS which is a version of Linux. While students will work within the Linux shell, the programming language of this course is Python. Students work individually and in teams to design and build simple and comples mechatronic systems capable of meeting a variety of criteria including driving, pushing, controlling speed, etc. Sensors are introduced to allow robotic devices to interact with the environment. Actuator design is dis-

cussed and different manipulator systems are introduced. As an essential part of mechanical design, students will be exposed to CAD concepts using OnShape.

As you can see from this Euler diagram here, Mechatronics is a complicated field of engineering that combines many areas of study. This is the more technical name for robotics and was created by Tetsuro Mori in 1971 and has served as the name for this field since. In this course, you will be introduced to several of the areas on this diagram.

Since much of this course will be delivered remotely, be sure that you consider creating a good remote learning environment by doing the following:

Create a dedicated workspace for yourself that you can keep neat and organized throughout the year. Pick a place that is as quiet as possible so that you can concentrate. Do not use this space for anything else. In this course that space should include a place for your computer as well as an area to work on electronics projects - preferable with a hard surface like a desktop or workbench.

Manage your sleep schedule. You will need to be able to concentrate and stay focused since you do not have a teacher hovering over your workspace to keep you focused. If you go to bed at a consistent time and wake up at a consistent time, you will be able to learn more effectively.

Make sure you know how to log on to every account needed for the course. See the Prerequisites for all of the details for this course. Be sure to ask questions if you encounter any difficulties.

2.2   History of Robotics

Adapted from Robotics: A Brief History, Stanford University

Origins of "Robot" and "Robotics"

The word "robot" conjures up a variety of images, from R2D2 and C3PO of Star Wars fame; to human-like machines that exist to serve their creators (perhaps in the form of the cooking and cleaning Rosie in the popular cartoon series the Jetsons); to the Rover Sojourner, which explored the Martian landscape as part of the Mars Pathfinder mission. Some people may alternatively perceive robots as dangerous technological ventures that will someday lead to the demise of the human race, either by outsmarting or outmuscling us and taking over the world, or by turning us into completely technology-dependent beings who passively sit by and program robots to do all of our work. In fact, the first use of the word "robot" occurred in a play about mechanical men that are built to work on factory assembly lines and that rebel against their human masters. These machines in R.U.R. (Rossum's Universal Robots), written by Czech playwright Karl Capek in 1921, got their name from the Czech word for slave.

The word "robotics" was also coined by a writer. Russian-born American science-fiction writer Isaac Asimov first used the word in 1942 in his short story

"Runabout." Asimov had a much brighter and more optimistic opinion of the robot's role in human society than did Capek. He generally characterized the robots in his short stories as helpful servants of man and viewed robots as "a better, cleaner race." Asimov also proposed three "Laws of Robotics" that his robots, as well as sci-fi robotic characters of many other stories, followed:

A robot may not injure a human being or, through inaction, allow a human being to come to harm.

A robot must obey the orders given it by human beings except where such orders would conflict with the First Law.

A robot must protect its own existence as long as such protection does not conflict with the First or Second Law.

Early Conceptions of Robots

One of the first instances of a mechanical device built to regularly carry out a particular physical task occurred around 3000 B.C.: Egyptian water clocks used human figurines to strike the hour bells. In 400 B.C., Archytus of Taremtum, inventor of the pulley and the screw, also invented a wooden pigeon that could fly. Hydraulically-operated statues that could speak, gesture, and prophecy were commonly constructed in Hellenic Egypt during the second century B.C.

In the first century A.D., Petronius Arbiter made a doll that could move like a human being. Giovanni Torriani created a wooden robot that could fetch the Emperor's daily bread from the store in 1557. Robotic inventions reached a relative peak (before the 20th century) in the 1700s; countless ingenius, yet impractical, automata (i.e. robots) were created during this time period. The 19th century was also filled with new robotic creations, such as a talking doll by Edison and a steam-powered robot by Canadians. Although these inventions throughout history may have planted the first seeds of inspiration for the modern robot, the scientific progress made in the 20th century in the field of robotics surpass previous advancements a thousandfold.

The First Programmable System

There is very little doubt today about both how essential programming is to robotics and who the earliest computer programmer was. While many would immediately think of modern computer scientists like Steve Wozniak or earlier programmers like Alan Turing, the first person to successfully put forth the idea of a programmable system like a computer was Ada Lovelace. In her article of 1842 (Yes, during the John Tyler administration) she proposed that numbers and objects could be used to "express abstract scientific operations" and create action based on rules. This was the first time anyone had proposed programming a device to act on inputs based on rules. This is the very heart of what it is to be a robot. Despite the great disadvantage of being born a woman in a very male dominated 19th century, she founded the field which would become the backbone of mechatronics.

The Origins of the Math of Robotics

The backbone of decision making in the mechatronics is a field of math called Boolean Algebra. As a professor of mathematics at Queen's College Cork in Ireland, George Boole published a work titled An Investigation of the Laws of Thought in 1854. Boole seeks to prove mathematically the existence of god through the application of logic. While his success at justifying the existence of god has not been supported, the underlying mathematical system that he invented would be used a century later as the foundation for modern day computers.

Withough Boole's contribution to mathematics and logic, modern microprocessers built from transistors would not be possible. While he lived in obscurity during his life, the long legacy of George Boole is found in every computer programming language, including Python. The code below, which you will become familiar with this year, demonstrates this application.

```
While True:
     print("Boolean Algebra makes computer decision making possible.")
```

For a more in-depth understanding of his contributions to mechatronics see the embedded video in the Supplemental Videos Section.

The First Modern Robots

The earliest robots as we know them were created in the early 1950s by George C. Devol, an inventor from Louisville, Kentucky. He invented and patented a reprogrammable manipulator called "Unimate," from "Universal Automation." For the next decade, he attempted to sell his product in the industry, but did not succeed. In the late 1960s, businessman/engineer Joseph Engleberger acquired Devol's robot patent and was able to modify it into an industrial robot and form a company called Unimation to produce and market the robots. For his efforts and successes, Engleberger is known in the industry as "the Father of Robotics." Academia also made much progress in the creation new robots. In 1958 at the Stanford Research Institute, Charles Rosen led a research team in developing a robot called "Shakey." Shakey was far more advanced than the original Unimate, which was designed for specialized, industrial applications. Shakey could wheel around the room, observe the scene with his television "eyes," move across unfamiliar surroundings, and to a certain degree, respond to his environment. He was given his name because of his wobbly and clattering movements. The Supplemental Videos section of this text has a video made about Shakey.

In order to better understand the field of mechatronics, you will choose one person who has been foundational to the development of the field.

Assignment: The Innovators Project

Chapter 3

Electrical Concepts

This class and modern life relies heavily on electricity and electronics. Our course electronics text points this out quite eloquently:

The history of electricity starts more than two thousand years ago, with the Greek philosopher Thales being the earliest known researcher into electricity. But it was Alessandro Volta who created the most common DC power source, the battery (for this invention the unit Volt was named after him).

Direct current (also known as DC) is the flow of charged particles in one unchanging direction (most commonly found as electron flow through conductive materials). DC can be found in just about every home and electronic device, as it is more practical (compared to AC from power stations) for many consumer devices. Just a few of the places where you can find direct current are batteries, phones, computers, cars, TVs, calculators, and even lightning.

We will begin exploring a number of concepts relating to electricity as we will essentially be using electrons as our modes of information processing. From the electrons that flow through a button or distance sensor to be read by the microprocessor to those which supply the power for our motors and servos, we are heavily dependent on electricity. Before we get into this, we need to go over some very important concepts. Take out your Raspberry Pi and follow along with the video below.

Video here Rules of this course in regards to electricity:

Always work only with DC electricity, never use AC voltage except for when you plug your Raspberry Pi into the outlet.

Always work on your circuits when the devices are unplugged and batteries removed. Not only is this a safety issue, it will prevent the need to replace electrical components

Only use the components listed in the laboratory activity on which you are working. Do not try new components you have never used before or are unfamiliar with.

Keep your work area clean and uncluttered. Do not keep beverages on your workstation.

Circuit lab setup

3.1   Circuits

A basic understanding of circuit electricity is essential going forward. For this section of the course, we will begin using our online textbook. It gives us an excellent overview of electrons, circuits and polarity. Please keep in mind that, in order to begin using the Raspberry Pi (i.e. receiving the power supply) you will need to pass all of the required quizzes with at least a 70%.

Key Concepts of Section 3.1

A circuit is a loop of conductive material.

In order for electrons to flow a closed circuit is required.

A circuit is open or broken if a complete path for the electrons to flow no longer exists.

Rather than me trying to do a better job than our textbook, you are going to read the short section on circuits. These questions will be similar to those that you will see on your unit exam (The one you need at least a 70% on to move on). So…that means that you are not completing this worksheet for a grade. Instead, you are trying to measure how well you learned the concepts in the reading. Much of this year's content will be delivered this way. Get your head around the idea of learning for the sake of learning.

To help you understand this concept, here is an excellent video from SparkFun Electronics on the topic.

Assignments for section 3.1:

After reading the section on circuits, complete the worksheet

Using (tool for this), design a closed circuit with the following components:

5V DC power supply

Red LED

330 Ohm resistor

Submit the (method of submission) here.

3.2 Voltage and Current

The movement of electrons from one pole to another is what we use in robotics to make our robots work. By altering the volume of electrons moving through the wires connected to a motor we can alter the speed of the wheel attached to the motor. The potential energy coming in to our microprocessor pin over a wire attached to a photoresistor tells us the level of brightness or lumosity in a room. Putting these two concepts together we could create a robotic system that opens the blinds on our lab when the sun sets in the west. That is, of course, if we can program our system to make that decision on its own. We will get to the programming part later. For now, we will examine the concepts of voltage and current which allow us, along with resistance, to receive information through sensor and send information out to an actuation device like a motor.

3.3 Resistance

3.4 Ohm's Law

3.5 Using a Digital Multimeter

This section will become active in the event that we are not using remote instruction.

3.6 Basic Soldering

This section will become active in the event that we are not using remote instruction.

Chapter 4

Mechanical Design

Need to teach them a few things.

4.1   Basic Mechanics?

4.2   Advanced Mechanics?

4.3   Computer Aided Design

Chapter 5

Python Programming

Finally! We are going to start learning how to make the Raspberry Pi control components like lights and motors and receive input from sensors. This is what will make your creations become real robots. For this unit we will refer to the course programming text quite often. Many of the programming problems come from the Downey text as well. For all of your assignments in this section you will start here though. Learn the basic concepts of that assignment, read more about it on the course text, go to GitHub Classroom to practice the program and finally, transfer what you have learned to the Raspberry Pi.

Let's get started by setting up the Pi and learning more about Python and it's wide range of applications.

5.1   Setting up the Raspberry Pi

These procedures are for students in the Robotics Engineering course at Windsor High School. As such, some of the initial steps required in setting up a Raspberry Pi have already been completed. If you are using a new, out of the box, Pi refer to one of the many tutorials online.

Equipment Needed:

Raspberry Pi 3B+

5V | 2.5A power supply

32GB Micro SD card with pre-installed Linux based OS

Monitor

HDMI cable

Keyboard

Mouse

Optional: Windows, iOS or Linux Computer (not a Chromebook)

Connection to a LAN. In the lab this is "Mechlab". At home it is your router.

NOTE: For the initial setup we will use the monitor, keyboard and mouse. Eventually, we will only use the command line functions which will allow you to eliminate those components and control the Pi from your laptop remotely if you choose. If not, you can use the Pi as a computer itself but just enter the command line controls of the Linux terminal. Eventually, you will need to be able to do this though as we will disconnect the Pi from all physical I/O devices and make it the brains of a self-sufficient robotic system.

5.1.1   Putting the Pi Together

To begin, assemble the Pi by putting the SD card in the slot. Connect the monitor to the HDMI port, and the keyboard and mouse to the USB ports. Plug in the power cord and turn it on.

(Create a custom background for the RPi and install all of the programs needed for the class.)

5.2   The parts of the RPi and Pinout

Let's examine the parts of this tiny computer that you will be using for the rest of the year…it is really quite amazing how much can be packed on the small device.

There are several things worth noting here about the Pi:

While it is possible to power the device with a micro USB power supply that provides only 1A of current (older Android phones use this), it is best to use the full 2-2.5A power supply, especially when we start hooking up components to the GPIO pins

Be careful not to short two GPIO pins together or connect an external DC power supply incorrectly to one of the GPIO pins. These actions can damage essential components of the Pi.

TBC

A powerful feature of the Raspberry Pi is the row of GPIO (general-purpose input/output) pins along the top edge of the board. There are 40 pins which we can use for a wide range of robotics applications. It is important to remember which pins are which as the locations do not represent the GPIO labels. If you forget or cannot find this diagram, use the command line in the terminal.

This is as good a place as any to start using the terminal. Before the introduction of the Apple Lisa with a graphical user interface (GUI), all computers used something like the terminal as the interface with humans. All computers still contain a terminal. The Raspberry Pi OS we are using is based on the Linux kernel so we will use Linux commands but the procedure is similar for other systems such as Windows and macOS. With the Raspberry Pi powered on press CTRL+ALT+T to open a terminal session.

After the terminal opens, type the following command: `pinout`

You should see something like this in the terminal:

Any of the GPIO pins can be designated (in software) as an input or output pin and used for a wide range of purposes. We will use them to send and receive electrical signals. Two 5V pins and two 3V3 pins are present on the board, as well as a number of ground pins (0V), which are unconfigurable. The remaining pins are all general purpose 3V3 pins, meaning outputs are set to 3V3 and inputs are 3V3-tolerant.

A GPIO pin designated as an output pin can be set to high (3V3) or low (0V). In addition pins GPIO12, GPIO13, GPIO18, GPIO19 may also be used with Pulse Width Modulation (PWM). A GPIO pin designated as an input pin can be read as high (3V3) or low (0V). This is made easier with the use of internal pull-up or pull-down resistors. Pins GPIO2 and GPIO3 have fixed pull-up resistors, but for other pins this can be configured in software.

It is possible to control GPIO pins using a number of programming languages and tools. For this course we will use Bash, the scripting language of Linux, and Python.

5.3   Why Python?

It is worth spending a little time discussing the reasons why Python will be the chosen operating system for this course. Prior to Fall 2020, the course utilized the Arduino UNO as the microprocessor for the robotics systems. The Arduino is also a versatile, fully programmable microcomputer with input and output pins capable of producing and detecting low DC voltages. It utilizes the C++ programming language. There were a few limitations, however, with the Arduino UNO.

The biggest advantage of the Pi is the programming interface. Since the Pi is essentially a small Linux computer, applications and environments that function on Linux also function on the Pi. The default scripting language on Linux is called Bash. When you called up the `pinout` for the Raspberry Pi earlier, you were using Bash. It is a powerful command line interface. Why not just use Bash instead of learning another language within the Linux shell? The main reason for this is Python's usefulness outside of this class. We know from past

experience that almost all of our Robotics Engineering students will go on to some kind of technical field of study in college. While those do not all go in to robotics, they all need to learn new languages. Python is about the most useful language one can learn in high school if planning to go on to a technical degree in college like engineering, computer science, data science or machine learning as well as almost any natural science field of study. It is powerful, easy to learn and widely used. The most recent StackOverflow survey ranked Python 3rd most popular among users. By learning this language, you will be more prepared to take on the challenges of a field of study in college related to robotics. Since one of the main objectives of our program is to help students prepare for post-secondary STEM fields of study, Python is an easy choice.

Python is also an interpreted language instead of a compiled language. What is the difference? In order to get a compiled language to run on a computer, the additional step of compiling it into machine code must take place before the device will run the commands of the program. With an interpreted language, a program or virtual machine running on the device (In this case, the Python environment installed on the Raspberry Pi OS) interprets the script written by the programmer directly into machine code. This is a faster process than running a compiled language like C++. Other interpreted languages include Ruby, Java and Perl and good old BASIC (Which many of us adult programmers learned when we were younger on our Commodore 64 or Apple IIe computers). This structure makes Python faster to learn.

There are other advantages to the switch to the Pi. The first advantage that the Pi has over the UNO is an onboard wireless internet connection. The Pi can connect to other devices using either 2.4GHz or 5GHz frequencies as well as Bluetooth. The current version also includes a Power over Ethernet (PoE) capability. Making the switch to the Raspberry Pi will allow for more seamless IOT (Internet of Things) activities without the need for an additional board. Second, the Raspberry Pi can actually function like a computer. We can modify the OS easily by modifying the micro SD card used as RAM on the computer.

5.4   Hello, World...and a Little More

5.5   Functions

5.6   Conditionals

5.7   Iteration

5.8   Sensing the Environment

Chapter 6

Robotic Types

Discuss types of robotic systems.

6.1 DataBot

6.2 MobileBot

6.3 SolarBot

Chapter 7

Capstone Project

Chapter 8

Going Further with Robotics

8.1 Robotics in Higher Education

8.2 Supplemental Videos

The Genius of George Boole

Shakey:Experiments in Robotic Planning and Learning (1972)

Chapter 9

Course Materials and Instructions

9.1   Course Lecture Notes

9.2   Follow the Course YouTube Channel

9.3   Using GitHub

9.4   Think Python Textbook

9.5   All About Circuits Textbook

9.6   Download and Install PyCharm

9.7   Jupyter Notebooks

9.8   OnShape Account Setup

9.9   Controlling the Raspberry Pi Remotely

Using a remote shell, you can control your Raspberry Pi with another computer. You can do this through the linux command line or with a GUI using a service

like Tight VNC. Learning how to do this will allo