Contents

Jason Bishop's Personal Notebook for Robotic/Embedded Programming

Hi everybody!

This is my first post for this course and I thought I'd just introduce myself. I am currently enrolled at Athabasca in the BSCCIS program. I plan on wrapping the whole thing up before the end of the year. I have been looking forward to this course for a long time, as I think this type of programming is very interesting. I have no experience in any type of electronics or engineering, but I've heard of Arduino before, and I think the potential it offers is pretty amazing.

On a personal note, I am currently an English teacher in South Korea and have been working on my degree for the past couple of years while working out here. It's been an awesome in both living abroad and earning a degree in something that I truly want to do. I have a passion for video gaming, so I have been taking most of the game programming and development courses on offer. However, I am finding so many aspects of computing very interesting from the other courses too. I am excited to be finishing soon and I look forward to seeing all the cool things I can develop with my new found skills and knowledge. After that, it's break time! I'll be travelling the world for a year (mostly South America) and working on personal development projects, and making videos of all the countries I'll be visiting. I hope all of you out there have big plans too, and I hope we can all help each other achieve success in this and other courses.

Let's have fun and see what develops!

-Jason Bishop

P.S. Why are you reading this? Can't you see it says *PERSONAL* notebok?

Units

Unit 0: Orientation

Assignment 0 - Arduino Basics

Summary of Arduino Platform and History

The Arduino is basically a kit for building interactive electronic devices. It is an open-source project that began life in Italy in 2005. The aim of the project is to provide an inexpensive way for developers of all levels to create interactive devices. The main component of an Arduino kit is the micro-controller board. Since the project is open-source, there are many types of designs, but the basic functionality is a simple processor and board with I/O ports. These ports can interact with expansion devices called "shields" (and other circuits) that greatly expand the functionality of the main board. These "shields" allow the developer to use all kinds of other devices with the Arduino board, such as LEDs, actuators, and light sensors. The programs for Arduino are written in C or C++, but the kit includes the Arduino Integrated Development Environment that makes programming easier for novices. The compact design, ease of use, and low cost of the Arduino platform has led to users creating and experimenting with all sorts of new and interesting devices.

-from the Wikipedia entry Arduino
http://en.wikipedia.org/wiki/Arduino

Further Resources

The Making of Arduino - an excellent in-depth history
http://spectrum.ieee.org/geek-life/hands-on/the-making-of-arduino

Why the Arduino Won and Why It?s Here to Stay - a great editorial on Arduino features
http://makezine.com/2011/02/10/why-the-arduino-won-and-why-its-here-to-stay/

Arduino the Documentary - a video for the TLDR crowd, it's even got Nathan Seidle from SparkFun!
https://vimeo.com/18539129

Examining the SparkFun Inventor's Kit

This is my first real foray into the world of electronics. The most I've ever dealt with before is learing about circuitry in physics class. The ideas of voltage, resistance, etc. are familiar to me, but not in any practical way. I've never built any type of circuit or basic functioning machine. I've had an interest, but never the inclination to explore this area... until I saw this course on offer at Athabasca! I love programming because you get to create something that actually works and accomplishes some kind of task. Embedded programming and robotics takes this one step further into incorporating the software into an actual machine that can do a physical task.

When I first got the kit in the mail, I was surprised at how much there was inside. I've heard of Arduino before, and I knew that it was effectively a mini-computer that you could use to program stuff and make things, but I had no idea that there were already so many components to add on to it. I'm also glad that the kit includes a breadboard, because I've never done any soldering either. I can already see a lot of potential with everything included in the kit. I made an short unboxing video of the kit which includes the Arduino board and several components.

Unboxing the SparkFun Inventor's Kit

Installing the Arduino Development Environment

After opening the kit up and examining the guide that comes with it, I discovered that installing the Arduino IDE is a relatively simple process. Everything is available on the Internet. Installing all the software and drivers was extremely easy to do, and I'm ready to get to it! I made another video documenting this process as well.

Installing the Arduino IDE

Writing and Executing Your First Arduino Program

After everything has been set up, running the first program was no problem at all. I followed the guide included with the kit for the first program - a blinking LED. Inserting the wires was a bit tricky because the holes and text are extremely small and hard to see, but it just takes some patience and a steady hand. When I hooked it up, I was surprised that the light started flash on it's own, without me having to upload any code to the board. Huh? Maybe it came pre-installed or something? I have no idea. I went through the process anyway to confirm that it worked. The code for accomplishing this task was quite simple to follow. I used the sample code provided by SparkFun so I didn't type anything in myself, but I feel that I still understood what was going on well. Anyway, everything worked as expected and yet again I made a video of this process.

My first program

Unit 1: Introduction to Robotics

Unit 1 covers the basics of robotics, including some terminology and history.

Chapter 1:
What Is a Robot?
Defining Robotics

My first impressions of the book are that it is informative and easy to read. I have enjoyed learning about the history of robotics and learning the new terms that used in the field. In particular, having a hard definition of what a robot actually is makes me think of robotics in a new light. Things that I thought of as robots, like those used in robot assisted surgery, are actually not robots by definition, because they require human controllers to perform actions. Robotics is a fascinating subject, and I am already eager to learn more after this short introductory chapter.

Chapter 2:
Where Do Robots Come From?
A Brief but Gripping History of Robotics

I am still finding the book pleasant to read. The author's style of writing makes things enjoyable and easy to understand. This chapter goes into much more depth about the history of robots and robotics, covering many things I did not know before. I found Grey Walter's tortoises particularly interesting, as he created robots with a surprising amount of functionality so early in the history of robotics. This leads into the concept of artificial intelligence (AI), which I find a particularly interesting topic, and I hope the book eventually covers more on this subject.

Chapter 3:
What's in a Robot?
Robot Components

This chapter begins by asking the question "Have you wondered what makes up a robot, what makes it tick?" My answer to this question is a resounding "Yes." I remember when I was younger, taking apart electronics like my alarm clock and video game controllers to see what was inside and trying to figure out how they worked. This chapter focuses a lot on the terminology of robotic parts, allowing me to better understand the robotic/electronic components I had no names for in my youth. I had never really put much thought into the concept of the division between sensors and effectors, and I feel this chapter makes the concepts quite clear.

Questions to Ponder

Food for Thought - Chapters 1, 2, 3

Exercises

Exercises - Companion workbook and SIK circuits

U1 - Food for Thought

Chapter 1

What else can you do from afar, by means of teleoperation?

There are many things that can be done from afar using teleoperation, including the examples of telephone, telegraph, and television provided. Business conferencing can be done from afar, as in teleconferencing. Another example would be surgery. There are machines that doctors use to perform complicated surgery without having to get hands on with a scalpel. And with the advent of the internet, there are many ways to have a telepresence in another location. For example, there is the ability to hunt in a real world location from your computer. The user can move a robot around and fire a gun. With computers and networking becoming a ubiquitous part of are lives, the ability to teleoperate devices continues to grow.

Is a thermostat a robot?

If we examine the definition of a robot provided in this chapter, the key points are that a robot is autonomous, exists physically, has sensors, acts on information from these sensors, and achieves a goal through these actions. Based on these points, it seems that a simiple thermostat CAN be defined as a robot. It does exist in the physical world, as there is usually a control device to be used. It can use sensors to sense the temperature of the environment. It acts on the temperature information it receives to adjust the temperature of a room or building. The goal of these actions is to control the climate of the room or building to be within a certain temperature range. Finally it acts autonomously because all a human operator has to do is select the desired temperature. The thermostat will control the operation of the heating/cooling/other systems on it's own after this selection.

Is a toaster a robot?

Using the defintion points discussed above, it would seem that a toaster CANNOT be classified as a robot. It too will act autonomously, as all one has to do is push the button and later the toast will pop up by itself when finished. And of course, a toaster is a physical object that is used to toast bread. However, a toaster does not really have a sensor. It does not sense the state of the toast, or temperature inside the toaster (unless we are talking about a toaster oven). All it has is a timer which allows a user to select how long it will be in operation for. Accordingly, the toaster cannot act on any type of information, because it receives no such information. Finally, the goal of the human operator of a toaster is to have perfectly browned and delicious toast. However, a toaster has no goals to achieve. It simply turns on and off according to the amount of time selected by the human operator.

Are Webcrawlers, called softbots, robots?

Again the definition of a robot provided points us to the conclusion that a Web crawler CANNOT be called a robot. Web crawlers are autonomous because they act on their own to gather information from the Web. However, the web is a virtual world, and therefore a Web crawler does not exist in the physical world. This means that they can't really have sensors either, because there is nothing real to actually be sensed. However, they do have goals to achieve, usually retrieving or gathering information, and they will take actions to achieve these goals, but this is not enough to make them true robots.

Is HAL from the movie 2001 a robot?

Applying the definition of a robot to HAL, we can see that it CAN be called a robot. HAL exists in the physical world. The astronauts interact with it and we know of its presence on screen with its ominous red light. It also clearly acts autonomously, because it acts according to its own wishes, despite the fact that its human operators wish otherwise. HAL also has sensors placed throughout the environment, such as cameras and microphones, which it uses to track the humans aboard. It uses this information to make decisions and act, killing the humans aboard for example. The reason for these actions was to accomplish the goal of preventing these humans from deactivating it and "jeopardizing" the mission. HAL fits the definition of a robot easily, just not a very nice one!

Chapter 2

How important is it for robots to be inspired by biological systems? Does it matter what kind of robot your are building? Does it matter if it is going to interact with people?

If we consider the field of robotics overall, I feel that the importance of biological inspiration is not as important as it was in the early stages of robotics and cybernetics. The question provides the example of airplanes and helicopters being very different from birds. I would add that there are many machines that are used which have no good biological counterpart. For example, there are the many types of robots used in manufacturing. These robots are designed to act in ways that no other animals would need to. As such, they often look interesting and move in unique ways. However, this does not mean that biological inspiration has no importance to robotics whatsoever.

Robotics is often concerned with efficiency. As with the manufacturing robots mentioned, we want them to move and act in the most efficient manner to produce maximum output, reduce wear and tear, and a number of other reasons. The billions of years of evolution on Earth have helped to shape some very efficient designs. Accordingly, we can then look to biological systems to provide us with efficient solutions to some of our robotics challenges. This is often the case with many robots, as seen with MIT's famous Cheetah Robot (http://newsoffice.mit.edu/2014/mit-cheetah-robot-runs-jumps-0915).

Additionally, if we are building a biomemetic robot, biological systems will have greater importance. It's right there in the name! Building a robot to model a biological system requires the study of that system. A robot that interacts with people is similar in concept. People feel uncomfortable with odd interactions. This is demonstrated by the "uncanny valley" effect often found in video games and humanoid robots. Therefore, it is necessary to study and understand the intricacies of human behaviour to make a robot interact in a comfortable way for a human. Nevertheless, these are more specific applications of robotics. Biological systems may be the "best" model for robotics in many cases, but certainly not all of them, and they are certainly not the "only" model for robotics.

Chapter 3

What do you think is more difficult, manipulation or mobility?

At the basic levels of both manipulation and mobility, it would seem that manipulation is the more difficult task. Humans consistently use machines to get around, with devices like cars, airplanes and tanks. These concepts can be fairly easily applied to robotics for basic mobility. Basic manipulation on the other hand, can be much more difficult. Each type of manipulation can require a specialized part or ability. For example, flipping a simple light switch doesn't require much in the way of manipulation. However, even a basic task like turning the page of a book, requires much finer control, that can be much more difficult to implement.

However, as the complexity increases for both types of tasks, it seems that mobility becomes more difficult than manipulation. We have modern robots today, that can perform manipulations with more accuracy than any human can achieve. This is necessary, for example, in the production of microchips or other manufacturing processes. Complex mobility, on the other hand, does not seem to have reached the same level of precision. Honda's robot ASIMO (https://en.wikipedia.org/wiki/ASIMO), for instance, is one of the first robots in the world that can accurately simulate human locomotion, and we are still waiting on Google to release it's autonomous cars. However, these types of mobility tasks seem to be complex because they require fine manipulation of the parts of the robot itself to respond to the environment. So while it seemed that mobility is harder at more complex levels, the required manipulation to provide complex mobility seems to show that mobility is the more difficult of the two.

How large do you think your sensor space is?

I think my sensor space is extremely large! Most people remember learning about the 5 senses in school - sight, hearing, smell, taste and touch. However, humans really have an abundance of different senses. These include our sense of balance, our sense of tempurature, our sense of pain, and so many more (not including sense of humour of course!). All of these senses combine to provide a large sensor space just in number of sensors alone. However, many of these sensors operate on a continuum. For example, we can hear over a range of sounds (about 64 - 23,000 Hz) and see a over range of electromagnetic radiation (about 390 - 700 nm). When we consider these values combined with the number of senses we have, the sensor space is HUGE!

Can you think of things or information in your life that is observable, partially observable, or hidden?

There are many types of information that are observable in everyday life. The colour of the chair I'm sitting in, the sound of the music I'm listening to, the feel of the keys underneath my fingers as I'm typing, are all perfectly observable pieces of sensory information. An example of something partially observable is the universe itself. Our current scientific theories tell us that nothing can move faster than light. As such, any part of the universe that is sufficiently far away from us is unobserveable. So while we can detect a whole host of stars and galaxies with our modern telescopic equipment, there are parts of the universe that we will never be able to gather information about, without some type of extreme scientific breakthrough. There is also information that his hidden from us everyday. Like when I look at my pet cat, I wonder what she is thinking - or even if she is "thinking" at all? This type of information is hidden from us, and as is the case with the observeable universe, it would require a major scientific breakthrough to gather this information.

U1 - Exercises

I quickly read through the information here as it seems to be covered well in section 3.2 of the book. I have attempted to answer the questions based on the SparkFun Inventor's Kit (SIK), as opposed to the iRobot Create, because that is what we are using for our course.

The Robotics Primer Workbook
Robot Components

Exercise 1: Sensors and Sensor Space

1. List each of the SIK's sensors. Refer to the SIK Guide for help as some sensors may not be obvious.

The SparkFun Inventor's Kit contains several sensors. These are:

- a temperature sensor
- a photo resistor
- a flex sensor
- 4 push button switches
- a potentiometer
- a soft potentiometer
- a sealed relay

2. Define the SIK's sensor space. For each of the sensors, identify what it perceives and the range of possible sensor readings.

The sensor space is the space of all possible sensory readings. For the SparkFun Inventor's Kit, this includes:

Temperature sensor
This is a linear temperature sensor. It adjusts output voltage based on the surrounding temperature.

Photo resistor
This is a passive sensor. It detects the amount of available light and responds by adjusting the resistance.

Flex sensor
Perceives the amount of "bend" in the sensor and responds by adjusting the resistance.

Push buttons
There are 4 button switches available in the kit. They can provide binary inputs.

Potentiometer
This is a variable knob that adjusts resistance depending on the knob position.

Soft potentiometer
This sensor reacts to pressure on its surface and responds by adjusting the resistance.

Sealed relay
This causes a switch to trip when it receives a jolt of energy.

3. Now imagine a sonar sensor has been added to the SIK. What is the new sensor space?

With a sonar sensor, the SparkFun Inventor's Kit sensor space includes the above, plus:

Sonar sensor
Detects the echoes of sound waves and responds by adjusting the electrical signal in some fashion. If it is an active sensor, it will also transmit the waves for reflection and sense them.

Exercise 2: Effectors and Actuators

1. Describe the difference between effectors and actuators.

An effector is a device that affects the environment. For example, legs and wheels allow a robot to move. A gripper or hand can manipulate objects.

Actuators are the components that allow effectors to execute actions. For example, legs and wheels are often driven by motors. Hands or grippers often use servos.

2. List each of the SIK's actuators and effectors.

The SparkFun Inventor's Kit includes the following actuators:

- a small servo
- 2 DC motors
- a piezo buzzer

The SparkFun Inventor's Kit includes the following effectors:

- various small plastic pieces included with the servo

3. List the degrees of freedom for the SIK.

With 2 motors and a servo the SparkFun Inventor's Kit has 3 DOF. However, the only effectors are the plastic pieces that can attach to the servo. The servo only controls rotation, therefore there is only 1 controllable DOF.

SparkFun Inventor's Kit
Circuits #2 - #7

Building the circuits has been quite simple but still quite interesting. This is the first time I am encountering a lot of these types of electronic components. The SIK is making things easy to understand and use so far. I can see many of the real-world applications of these components and how they can be used in many interesting ways. The kit is certainly flooding my imagination with possibilities.

I have made videos of all the circuits completed for this unit as a YouTube playlist:

https://www.youtube.com/watch?v=WktAo1TREtU&list=PLGwxDzcLDnGlTi_mU6fJAl5s_xDVYPDeV

My favorite circuit so far has been circuit #4 using multiple LEDs. I currently live in Korea, which has a lot of LED and neon signage that can be quite fascinating to watch to watch in operation. I found this circuit to be interesting to watch as well.

https://www.youtube.com/watch?v=g2wpLDcGoIU

Unit 2: Robotic Movement 1 - Locomotion

Unit 2 covers the topic of robotic movement, focusing on how locomotion is achieved using particular components.

Chapter 4:
Arms, Legs, Wheels, Tracks, and What Really Drives Them
Effectors and Actuators

This chapter goes into detail about effectors and actuators used to achieve robotic movement. Motors are mentioned as the most common actuators, and the chapter goes in-depth with their function. I remember creating a simple motor for high school physics class so the general concept is somewhat familiar to me. I feel that the gearing concepts are well explained and easy to understand, which was not the case when I first learned about them. Also well explained is the concept of degrees of freedom (DOF), a concept that I had never heard of before reading this chapter. It was very interesting to learn about.

Chapter 5:
Move It!
Locomotion

This chapter dives deep into robotic locomotion. I was already aware of the difficulties of locomotion, namely that it can be very challenging to balance and compute movements. Even the best robotics of today, like the MIT Cheetah robot, still don't seem to have the fluidity and ease of movement of biological creatures. I was particularly interested in the section on gaits and number of legs. I find this a very interesting robotic challenge because moving on a flat surface is not very difficult, but handling rough terrain, changes in elevation, and three-dimensional movement are things that humans do easily, but is so difficult to get robots to do well. I am still enjoying the readings so far, and I am looking forward to future chapters.

Questions to Ponder

Food for Thought - Chapter 4, 5

Exercises

Exercises - Companion workbook + SIK circuits

U2 - Food for Thought

Chapter 4

How would you measure a motor's torque? What about it's speed?

To measure any aspect of a robot, we require a sensor. This of course applies to motors that we use inside the robot. Therefore, to measure torque, we would need some kind of sensor to measure the rotational force generated by the motor. To measure this externally, some sort of resistant device is needed. A material with known properties can be used to resist the force generated by the motor and measuring the amount of bend in the material will allow us to calculate the force generated.

Measuring the speed of the motor would require some kind of sensor to measure the rotational speed of the motor axis. This can be done externally by something on the motor and using it to measure the rotation. For example, a simple black piece of tape can be applied, with a sensor to count how many times the tape passes the sensor within a minute or second.

Measuring these things internally is much more complicated, but could probably use similar methods. Also, the book mentions that the amount of power used is proportional to the torque and rotational velocity of the motor. Therefore, if we know the amount of electricity being used by the motor, we should be able to measure only one of the values and use the properties of the motor to calculate the other value.

How many DOF are there in the human hand?

The human hand has 22 degrees of freedom. Each finger has 4 DOF; the 3 knuckles allow the finger to bend, and the base knuckle, where the finger meets the hand, also allows for a side to side movement. The thumb only has 3 DOF; the 2 knuckles allow for the same bending as the fingers, the base of the thumb has 1 DOF because it can move side-to-side, but the rotation of the thumb is not really controllable so is not counted. The remaining 3 DOF are located in the wrist; the wrist can move up/down, side-to-side, and rotate. 4 x 4 (fingers) + 3 (thumb) + 3 (wrist) = 22 DOF for the hand in total.

Chapter 5

How does an automated Web path planner, such as Mapquest or Google Maps, which finds a route for, say, a trip from your house to the airport, find the optimal path?

Google and Mapquest use algorithms to find the optimal path between two destinations. The specifics of the algorithms are not available publicly because these are private companies protecting their intellectual property. However, I can guess at the general algorithm used. Basically, the goal of the algorithm is to compute the fastest and most direct route. The algorithm probably looks at the road connections between the two end points. Each route between road connections is likely given a weight, based on the characteristics of the road. For example, a highway would be given more weight than a city street. Each intersection may be given a weight as well, depending on if the driver will turn left or right because turning left usually requires more time. Things get even more complicated if you include Google's traffic tracking abilities, which alter the weights even further. With all the weights in place, the algorithm chooses the best path based on these weights. There is probably some sort of cyclical algorithm to determine the best possible route by using the sum of the weights between the endpoints. The best weighting calculated will be the route displayed to the user.

U2 - Exercises

Reading through the workbook, it covers the concept of error in calculating mechanical systems. This idea was also brought up in the book.

The Robotics Primer Workbook
Locomotion

Questions from the Instructor's Notebook

Compare what they are doing [in the workbook exercises] with the iRoomba to robot kits like Lego Mindstorms. How do they compare?

I assume that by iRoomba, we are to examine the iRobot Create, as this is what is used in the exercises and the Roomba robot comes already pre-programmed to vacuum, whereas the Create is a programmable robot. Therefore after examining both the iRobot Create 2 and the Lego Mindstorms EV3, I have discovered a few similarities and differences.

The Create kit comes as a pre-assembled robot that is programmable. Motors are already attached to wheels and buttons and other sensors are built-in. These sensors seem to be of good quality (since they are based on the iRoomba) and there are also small extras included, like a fourth wheel. Overall however, adding sensors or other components to the robot seems to be done in more of a hacking fashion without slots or positions for predefined pieces. The exercises in the workbook start out with some simpler locomotion exercises, moving on to homing and teleoperation.

The Mindstorms product on the other hand, does not come pre-assembled. Numerous pieces are included to be able to construct the robot body and effectors in a typical Lego fashion. This makes the robot highly customizable in more of a planned fashion compared to the hacking approach of the Create. The kit also includes wheels and motors, just like the Create, as well as color, touch and infra-red sensors, however, I have no clue regarding the actual quality of these sensors. There are no specific exercises that I could determine, other than building the robots suggested with the kit. However, it seems that the exercises in the workbook could be completed using the Mindstorms product.

It seems that the degrees of freedom available with the Mindstorms kit is much greater, because it includes parts to create all kinds of different effectors, and there are 3 motors included. The Create 2, by contrast, is limited to wheels with motors already attached, and anything else must be purchased separately and hacked onto the robot body. However, depending on the limitations of the Mindstorms sensors, there may be extra fidelity and quality available with the Create 2's sensor space. This can allow the robot to have more of a sense of it's environment, allowing for more information for the robot to make decisions to act on.

How do the motors used in the iRoomba compare to the various motors in your Inventor's kit?

There doesn't seem to be much information about the actual motors in the Create 2. However, it is clear that they are attached to the main wheels to provide the robot with locomotion. The DC motors in the SIK come with no effectors whatsoever, and seem quite small. I would guess that they are less powerful than those available in the Create 2. The SIK also comes with a servo, with some small plastic parts to use as effectors with limited functionality. This servo is also very small, and obviously not very strong, but the Create 2 comes with no such functionality.

Discuss how you might build a robot like the iRoomba using your Inventor's kit, including and additional parts you might have to acquire.

To build a robot like the Create 2 from the SIK, the most important thing I would need is wheels! I would attach them to the motors available with the kit. This would allow for similar movement between the two. The first exercise is getting the Create 2 to move in a square, which could be done with the setup described for the SIK. The next exercise uses photoresistors to move along a path. The SIK has only one photoresistor, another would need to be purchased to complete the exercise. The third exercise could be completed with this set up as well, as it again uses 2 photoresitors, this time to move toward a light. The fourth exercise is teleoperation of the Create 2. This would be much harder to achieve with the SIK, as some type of separate input device would be required. Additionally, to allow the SIK to complete all of these exercises, additional parts would be required to construct a body for the robot.

Web Resources

iRobot Create 2

Webpage - http://store.irobot.com/education-research-robots/irobot-create-programmable-robot/family.jsp?categoryId=2591511
Wikipedia - https://en.wikipedia.org/wiki/IRobot_Create

Lego Mindstorms EV3

Webpage - http://www.lego.com/en-us/mindstorms/products/31313-mindstorms-ev3
Wikipedia - https://en.wikipedia.org/wiki/Lego_Mindstorms

SparkFun Inventor's Kit
Circuits #8 - #12

Continuing on with the circuits in the SIK guide, I am continuing to enjoy the learning process and discover all the possibilities of the kit and the Arduino IDE. One gripe I do have is that the motors included with the kit have no effectors to attach to them. It would be much more fun to have some wheels or something cool to use, but I suppose that it can be much more expensive and complex than the goals that the guide is trying to accomplish. Anyway, there have been no real challenges so far, and I find the variety of sensors intrguing.

YouTube playlist

Unit 3: Robotic Movement 2 - Affectors

Unit 3 covers the topic of robotic movement, focusing on affectors that allow a robot to complete manipulation activities

Chapter 6:
Grasping at Straws
Manipulation

This chapter covers the robotic manipulators of all kinds. It talks about the different types of joints and the complexities of manipulating the robotic effectors to get them in the position we want. I found the chapter short but insightful and it made me think about how easily humans can move and position ourselves compared to getting a robot to do the same thing.

Questions to Ponder

Food for Thought - Chapter 6

Exercises

Exercises - SIK Circuits + Design Question

U3 - Food for Thought

Chapter 6

How many DOF are there in the human hand? Can you control each of them independently?

As mentioned in the previous Food for Thought question, there are 21 DOF in the human hand. 4 in each finger, 4 for the thumb, and 3 in the wrist. These are all controllable through the wrist and finger joints, with associated muscles and ligaments.

Which of the two joint types we discussed, rotational and prismatic, is more commonly found in biological bodies? Can you think of examples of specific animals?

In biological bodies, the rotational joint is much more common than the prismatic joint. There are many examples of rotational joints in biology. One example is the mammalian shoulder. This is the ball and socket joint that we all have and are very familiar with. Prismatic joints seem to be pretty rare. In fact, I can think of no examples in any biological form. However, a quick Google search mentions that proton pumps in biological cells are used to drive proton pumps. As mentioned in the textbook, a piston is a type of prismatic joint. In this case, it's just not the type of bone joints that we are used to.

Are astronaut suits exoskeletons? What about a lever-controlled backhoe?

The term exoskeleton simply means "outside skeleton." In robotics, it is a hard outer structure worn to enhance human capabilities. I would say that an astronaut suit is an exoskeleton. An astronaut wears the suit to help survive the vacuum of space, which would be impossible for a human to do without this protection. The suit provides temperature control and oxygen to the wearer, and allows him/her to operate tools and manipulate objects.

I also think that a lever-controlled backhoe can be classified as an exoskeleton. The backhoe allows the operator to teleoperate the backhoe device to perform various functions. It extends human capabilities by allowing the operator to pick up heavy objects, or dig through hard surfaces. The backhoe can also move around with its driving controls. There is also protection from the metal body of the backhoe. While the skeletal form is not exactly human shaped or sized, it still qualifies as an exoskeleton of sorts.

Imagine an exoskeleton which has its own sensors, makes its own decisions, and acts on them. This is definitely a robot, yet it is also controlled by a human and attached to the humans body. Can you imagine where this could be useful?

A situation where this type of exoskeleton might be useful in a military context. For example, the suit could protect its human operator from explosives and small arms fire. The robot portion could be programmed to fire weapons and move around the battlefield. This could allow for faster reaction and greater accuracy. The human could also take control when the situation dictates some other type of tactics, such as an order to retreat or cease fire. Writing this answer makes me feel like I play too many video games!

Robotic dogs have been used to play soccer. What are the end effectors of the dogs in that case? Are the dogs now purely mobile robots?

The end effectors of the soccer playing robot dogs are the front paws of the robot. These are used to grasp or kick the ball by moving the legs and paws into the correct position and orientation. The dogs are no longer purely mobile robots because the can use these effectors to manipulate the environment, in this case, kicking the ball. The robots are still mobile, but also have manipulation capabilities, so they are not purely mobile.

U3 - Exercises

SparkFun Inventor's Kit Guide

As mentioned in Unit 2, I have finished all circuits from #1 - #12. I found the circuits simple and fun to do.
Here is a YouTube video link to a playlist of the videos I have made for each circuit:

YouTube

Design Question:

How many motors, and of what type, would you require to make a fully functional robotic arm that had a working elbow, wrist, and end effector (i.e. a simple clamp)?

For this task, all motors I would use would be servo motors so that the arm components stay in position when any forces are applied. Beginning with the elbow joint, I would use a servo motor with a high degree of torque. The elbow is a simple joint that only moves in one plane, so only one motor is necessary. The high torque servo is required because the elbow will have to lift the itself, the remaining arm components, and whatever payload is to be carried by the end effector. The speed is not as important a factor in this case.

The wrist joint has more DOF than the elbow joint and will require more servo motors. The elbow joint will allow the arm to get in an approximate position, but the wrist will fine tune the positioning of the end effector to properly manipulate an object. Therefore, I would want servo motors that are focused on speed as opposed to torque. Once the arm is in approximate position, we will want to be able to quickly move the effector into position, and the wrist joint will not be carrying as much of the payload weight as the elbow joint. To properly model the wrist joint, I would need 3 servos to model the roll/pitch/yaw movements.

Finally, the number of servos required for the end effector depend on the type of effector being used. In this case, we are using a simple clamp. Each side of the clamp could be controlled independently with 2 servos. However, controlling each side independently is may not that useful a function. We could use the clamp like a pair of scissors, where each side comes together uniformly. This would require one servo to pull the ends together. The design I prefer is to have one end of the clamp be static, while the other end moves toward/away from the static end to open and close. This is like many clamps used as tools in the real world. We position the clamp and then screw it in to tighten it by moving only one end of the clamp. This design also requires the use of only 1 servo. In any case, I would want servos that have a higher degree of torque than speed. With the elbow and wrist doing the work of positioning, the clamp simply needs to move in one direction to open and close. Therefore, instead of focusing on speed, we can use the torque to be able have a secure grip on an object.

What components would you add if you wanted the clamp to be able to tell how hard it was grabbing an object such as an egg (i.e. to avoid crushing it)?

To be able to tell how hard the clamp was gripping an object, I would add pressure sensors to each ends of the clamp that come into contact with the object. Increasing pressure on the sensors will change some value, such as resistance. Monitoring this value will allow us to determine the amount of force being applied to the object.

Programming/Circuit Task:

Create an elbow joint.

Planning

To create this elbow joint, I will use the micro servo from the SIK. This will allow me to easily position the elbow as demonstrated in circuit #8 of the guide. To simulate the arms I will use wooden chopsticks, because these are easy to get where I live (in Korea). If I'm feeling artistic and have time, I will attach some paper to the sticks to make them look nicer. To control the movement, I will try to use buttons to control the direction of movement, similar to how they were used to control an LED in circuit #5. Each button will control movement in one direction, pressing both will do nothing, and pressing the corresponding button to move the joint beyond its limits will also do nothing. If I can't get the buttons to work, I will use serial input to control the exact servo position.

For the project code, I will attempt to use the servo controls from circuit #8 and the button logic from circuit #5. Here is some basic pseudo code for the program:

```
declare constants/variables

setup
    designate the wiring positions of the servo/buttons

end setup

loop
    if button1 is pushed and button2 is not
  if arm is not at max degrees
   move the arm up by x degrees
   wait for the movement

 if button2 is pushed and button1 is not
  if arm is not at min degrees
   move the arm down by x degrees
   wait for the movement

end loop
```

I have made a drawing of the circuit I plan to implement:

Here is the testing plan for the complete circuit:

press buttons to check for proper movement
move elbow to max position, push button to try and move past max
move elbow to min position, push button to try and move past min
adjust movement rate and delay to acceptable values

Implementation

I completed the circuit as shown in the following picture:

Here is the code used to implement the behaviour:

elbow_sketch.ino

Finally, I made a video of the complete project:

YouTube

Discussion

I had fun finally creating my own project with the SIK. I basically combined the designs from two of the example circuits; the one dealing with buttons and the other dealing with servos. Setting up the circuit itself was pretty easy, but I was a little nervous setting it up. Usually for hooking up the circuit, the guide makes everything very clear so I feel confident about it working and not damaging any components. Doing it by myself on the other hand, I made sure to double check all my connections before plugging in the board.

This is my first time really writing my own code in the Arduino IDE, but I found it super-easy to use. I've taken quite a few programming courses at Athabasca so far, so this task was relatively simple for me, but I am happy I've had a chance to get familiar with the workings of the IDE. I basically followed the logic of my pseudocode, but used the handy constrain function to keep values within range instead of using if statements.

Once the circuit was built and the programming uploaded, everything worked on the first try exactly as expected. I tested the button functions as laid out in my test plan. The joint moved as expected and stayed within the limits of motion it was supposed to. I played around with the values for amount of movement with each loop the button is pressed, as well as the delay value between. I basically tuned it by hand until I found values I liked, moving the servo by 5 degrees each loop, and having the delay around 50 ms so that it didn't update too fast, but fast enough to not slow the movement or make it jerky.

Upon finishing this project, I also made a little guide for it. I modeled it after the SIK Guide with a circuit diagram, connections, and troubleshooting section. You can take a look here.

Unit 4: Robotic Sensing

Unit 4 covers the area of robotic sensing. Robots need to be aware of both their internal and external environments using sensors. This allows them to adjust to these conditions to make useful actions.

Chapter 7:
What's Going On?
Sensors

This chapter discusses the different types of sensors we use in robotics. It delves into the levels of sensor processing and different types of perception. It made me realize the importance of robotic sensors, especially in creating automnomous robots. I feel that because we don't always see what sensors are doing, the are overlooked in favor of robotic movement components such as actuators and effectors. However, they are just as important.

Chapter 8:
Switch on the Light
Simple Sensors

After explaining the concept of sensor classification, the chapter goes in depth on many different types of simple sensors. The sensors covered are mostly available in our SparkFun Inventor's Kit, so I have become familiar with them while working on this course. This chapter made me think about how even simple sensors can be used in complex ways, just like the discussion in the book of using a simple light sensor for shaft encoders and break beam sensors.

Chapter 9:
Sonars, Lasers, and Cameras
Complex Sensors

In this chapter, particular focus is on ultrasound and lasers, discussing the challenges and differences. It ends with a discussion of the field of machine vision highlights the complications involved with this type of robotic perception. I found this chapter very interesting because I have no previous experience with handling robotic sensors. I was surprised to learn how cheap ultrasonic sensors are, as I assumed they would be much more expensive. However, it does not surprise me that laser sensors are still quite expensive. I didn't know that size was still a problem with these sensors, because I always thought they would be smaller, like a laser pointer.

Food for Thought

Food for Thought - Chapters 7, 8, 9

Exercises

Exercises - Robotics Primer Workbook + Questions from the Instructor's Notebook

U4 - Food for Thought

Chapter 7

Uncertainty is not much of a problem in computer simulations, which is why simulated robots are not very close to the real, physical ones. Can you figure out why?

As mentioned, a robot running in a simulation does not have to deal with uncertainty. All parameters of the simulation are known to be able to create the simulation. If these values are known to the robot, then it will not have to deal with uncertainty, and will behave differently when faced with real world uncertainty. Additionally, simulations can reflect the real world, but are not perfect representations of it. Therefore, when a robot moves into the real world, there can be uncertainties that aren't accounted for, causing the robot to sense or behave in a different manner. Finally, simulations cannot possibly take into account every single variable present in the real world. As such, a robot in the real world may encounter events that it can misinterpret because they were never encountered in the simulation. For example, a robot designed to detect kill spiders by detecting 8-legged creatures may kill on octopus that is placed in its path, since octopi are generally not in the same areas as spiders.

Some robotics engineers have argued that sensors are the main limiting factor in robot intelligence: if only we had more, smaller, and better sensors, we could have all kinds of amazing robots. Do you believe that is all that's missing?

These engineers don't know what they're talking about. All of the best sensors in the world can't make up intelligence. Processing and computation are both needed to make sense of the sensory inputs. If we look at the history of cameras, the original camera obscura is no better at detecting humans than today's high-tech digital DSLRs. What makes the digital DSLR better suited to this task is that it is electronic, and we have the computing technology available to be able to interpret the signals from the camera, and in many cases, also the computational power to be able to figure out what is in the image. Therefore, no amount of sensory refinement can truly lead the way to intelligence.

What do you think will it take to get robots to be self-aware and highly intelligent? And if some day they are both, what will their intelligence be like, similar to ours or completely different?

I have thought about this question many times, and I still have no idea what the answer could be. I believe that the best way for us to go about tackling the problem is to model robots after the only creatures we know to be fully self-aware and have consciousness - humans. The more I learn about brain research, the more it seems that scientists in the area are coming to the conclusion that consciousness is an emergent property of the brain. There are the fundamental parts and processes of the brain, such as vision, motor control, language processing, etc. And the combination of these factors allows for consciousness to emerge.

My thoughts on the matter come back to how much self awareness is required. Clearly a robot can have some form of self awareness through proprioceptive sensors. But clearly this is not enough to have consciousness arise because we haven't been enslaved by any robot overlords yet. Many animals have a higher level of self-awareness and can recognize themselves in a mirror, but as the question states, there are still arguments as to whether they are truly conscious. I think the key is that self-awareness not only involves recognizing the body and the separation as an individual entity, but also a greater sense of both internal and external awareness. What I mean is the ability we have to recognize ourselves as individuals within our own minds, as creatures that can not only make external actions in the world, but also reconstruct this in the simulated environments of our minds. I can't be much clearer, because I don't know if language can actually articulate these kinds of concepts.

With that said, I believe that if robots are intelligent one day, there consciousness will not be similar to ours. There are several factors that I feel contribute to this. First of all, the advanced electronics to create these robots are likely to exceed human capacities. The concept of time will be completely different to a robot that can do things a million times faster than us, and live a million times longer than us. Additionally, electronics allow for a connection between them that humans can't have. Robots may be able to access the future super-Internet, and also likely other robots. Not only can they have perfect recall and access to basically all information collected to the present, the connection with other robots means they can instantly share this information with each other and will not have the same barriers to communication and understanding that humans do. This could lead to a Borg-like hive mind, which, of course, is nothing like our own.

Chapter 8

Why might you prefer a passive to an active sensor?

A consideration when deciding between an active or passive sensor is the power usage. An active sensor typically requires power to be able to use the sensor. If you are building a robot that requires low power usage, a passive sensor might be preferred. Another consideration is the property to be measured. In some cases, an active sensor may be more accurate, or may be the only type of sensor able to measure the property. For example, we use ultrasound to detect a fetus in the womb. Ultrasonic sensors are active sensors that send out an ultrasonic pulse from the emitter, and the detector receives the echo of these sounds from objects. This is one of the best devices we have to detect unborn fetuses, and using a passive sensor in this case would probably be less accurate.

Are potentiometers active or passive sensors?

Potentiometers are passive sensors. An active sensor consists of an emitter and a detector, but potentiometers do not have an emitter. The current passes through the potentiometer and the tab changes the resistance between the two endpoints. The detector detects this variable resistance.

What robot sensors would you say are most similar to such stretch receptors? Are they similar in form (mechanism of how they detect) or function (what they detect)? Why might stretch receptors be useful to robots, even without stomachs and eating?

It seems to me that the sensor most similar to a stomach stretch receptor is a potentiometer. The potentiometer measures tension, and as the stomach expands with food, the muscles stretch and become more tense. Additionally, the stomach muscles provide an analog signal to the brain, and potentiometers provide analog signals as well. The measuring of tension is similar in function to the stretch receptor, but not similar in form. The potentiometer has a middle tab that can move between two endpoints, whereas in the stretch receptor, the end points move away from each other.

Stretch receptors can have a number of useful applications in robotics. An example might be a robot returning to earth from space, using parachutes to slow its descent. The stretch receptors could be placed in the parachute to ensure that it is fully deployed. If not, the robot may decide to open a reserve chute to properly control its descent speed. Another application could be in a robot that uses a balloon to travel. The stretch receptors could detect the level of balloon inflation, giving the robot sensory information to control the balloon or detect the presence of a leak.

Chapter 9

What is the speed of sound in metric units?

From the book, the speed of sound in imperial units is 1.12 feet per millisecond. This translates to the metric version of 341.376 meters per second, or 0.341376 meters per millisecond.

How much greater is the speed of light than the speed of sound? What does this tell you about sensors that use one or the other?

As mentioned above, the speed of sound is about 341 m/s. The speed of light is several orders of magnitude faster at about 299 792 458 m/s. This tells me that the sensors that use light detection must be very fast in order to process the incoming information. Sensors that use sound do not need to be as fast, because the speed of sound is slower.

What happens when multiple robots need to work together and all have sonar sensors? How might you deal with their sensor interference?

If we have a group of robots working together to achieve a goal and all using sonar sensors, there will probably be interference among all the signals. For example, a robot that sends out a frequency pulse may have that pulse return to a different robot. The receiving robot may interpret that signal as its own and will product a false reading. This type of interference will probably cause many errors.

To deal with this issue, each robot can be set to use their individual sensors for a particular time slice. A single robot can be alloted a period of time in which it alone is using sonar to detect its environment. A different robot can use the next time slice, and so on. Each robot will be able to ensure that it is detecting only its own signal during a particular period of time, and can ignore any other signals received at any other period of time. However, the more robots in the group, the greater the time in between each sensor reading for an individual robot.

Another method that can be used is to have each robot's sonar operate within a narrow frequency range. Each robot can be assigned a particular range to use for its sensors. This allows all robots to be able to use sonar at the same time. This is the system that radio stations use. Each station is broadcasting at the same time. To receive a particular station, the detector (our radio) must be set to receive a particular frequency.

Besides using time-of-flight, the other way to use sonars is to employ the Doppler shift. This involves examining the shift in frequency between the sent and reflected sound waves. By examining this shift, one can very accurately estimate the velocity of an object. In medical applications, sonars are used in this way to measure blood flow, among other things. Why don't we use this in robotics?

We don't use the Doppler shift in robotics because there are not many applications where this is very useful to a robot. In the medical imaging example of this question, we are looking inside the human body to measure the blood flow. A robot rarely needs to know how fast something is moving inside something else. Robots are more often concerned with the surface of moving objects. This can be accomplished by comparing multiple sonar readings over time, or using another type of sensor. Additionally, the computation required to accomplish the use of the Doppler shift is much more computationally intensive. We want robots to be able to do things quickly, and this type of calculation will take up a significant amount of time.

Since two eyes are much better than one, are three eyes much better or even any better, than two?

3 eyes are not that much better than 2. As mentioned in the textbook, 2 eyes are much better than 1 because it allows animals/robots to interpret the world in 3D. However, adding a third eye does not expand sensing into the fourth dimension! However, having a third eye is not completely useless. For example, a robot with a third eye mounted on the back of the robot will be able to see behind itself. This could be useful in a robotic car, where we need to back up into a parking space, change lanes, or check behind the car for other reasons. A third eye can also be useful if it senses light outside the visual range. A robot with a third eye that can see in the infra red spectrum can use this information to identify objects in the image created by the other two eyes, or perhaps see in the dark, like using night vision goggles.

U4 - Exercises

I read through this section about sensors and found the information corresponded well to the concepts learned in the corresponding chapters. The use of IR and sonar sensors was particularly relevant as the functions and limitations of these types of sensors were covered in-depth in the textbook.

The Robotics Primer Workbook
Sensors

Exercise 1: Sensors & Levels of Processing

1. Describe the difference between an exteroceptive and proprioceptive sensor.

A proprioceptive sensor is used to provide information to the robot about its internal state. This can include positions of its effectors, the temperature of the processor, etc.. On the other hand, an exteroceptive sensor provides information about the robot's external environment. This can include the position of objects in the environment, the ambient light/sound levels, etc..

2. Create a table listing the SIK's sensors and if it is an exteroceptive of proprioceptive sensor.

The SparkFun Inventor's Kit has several sensors. Many can be classified as exteroceptive. However, in many cases, the determination of exteroceptive vs. proprioceptive sensor often depends on how that sensor is employed within the robot being built. Here is a table:

Sensor	Measures	Type
temperature sensor	temperature	both
photo resisitor	light level	exteroceptive
flex sensor	amount of "ben"	both
push button switches	switch state	exteroceptive
potentiometer	knob position	exteroceptive
soft potentiometer	position of pressure	exteroceptive
sealed relay	voltage spike	proprioceptive

Here is a comparison wiht the iRobot Create:

Sensor	Measures	Type
wheel encoders	odometry	Proprioceptive
IR cliff detector	binary cliff indicator - measures distance	Exteroceptive
passive IR beam detector	IR beam detection	Exteroceptive
battery capacity	voltage	Proprioceptive
wheel drop	wheels not touching the ground	Exteroceptive
bump sensors	robot collision with an object	Exteroceptive

3. Organize the sensors into a list of increasing level of processing.

The sensors included with the SparkFun Inventor's Kit are not very complex, therefore do not require much processing. I have ordered them as best I can:

Sensor	Level of Processing
push button switches	low
photo resisitor	low
temperature sensor	low
sealed relay	low
potentiometer	low
flex sensor	low
soft potentiometer	low

Comparison with the iRobot Create:

Sensor	Level of Processing
battery capacity	low
wheel drop	low
bump sensors	low
IR cliff detector	medium
passive IR beam detector	medium
wheel encoders	medium-high

Questions from the Instructors Notebook

Compare what they are doing with the iRoomba to robot kits like Lego Mindstorms. How do they compare?

To find out what kinds of exercises they are doing with the Lego Mindstorms, I found this website . These exercises involve creating robots to perform a number of tasks using the various sensors, and a lot of them concentrate on robotic movement. The focus of the exercises for the iRobot in this section also involve movement of the robot, but there is more of a reliance on the sensors to control the movement. For example, the IR and sonar sensors are used to make the robot follow a wall. These exercises make the robot more autonomous compared with the pre-planned movements used in many of the Lego exercises.

How do the sensors used in the iRoomba platform compare to the sensors in your Inventor's Kit?

The sensors of the iRoomba platform seem much more complex than the simple sensors included with the SIK. The SIK sensors only measure simple values, can be read easily by measuring the voltage, and do not need any calibration. As such the type of tasks they can be used for are relatively simple, unless used in a very clever design. The iRobot sensors are much more advanced, specifically, the IR and sonar sensors are much more advanced, as well as the wheel encoders. This also requires much more processing than is needed with the SIK sensors.

Unit 5: Robotic Control 1 - Feedback and Architectures

Unit 5 is concerned with how we control robots. It covers the concepts of feedback, or closed loop, control, as well as the how and why of control architectures.

Chapter 10
Stay in Control
Feedback Control

This chapter deals primarily with feedback control with a brief mention of feedforward control, which is not used nearly as often in robotics. It discusses the concept of state and the complexities of error within a control loop, as well as the basic forms of feedback control. For me, this is where robotics starts to get very interesting. I enjoy programming a lot, so this type of chapter begins to touch on the aspects of robotics I like the most. Getting the robot to do what you want may start with the sensors, effectors, and actuators in use, but the key to creating a good robot lies in its controller. Feedback control is an excellent concept to use in my future robotics project(s).

Chapter 11
The Building Blocks of Control
Control Architectures

Chapter 11 explains the concept of control architectures, a concept I was completely unaware of but very eager to learn about. These architectures are the guiding principles for designing the software that controls the robot. The chapter also has a section on language, but does not make a specific recommendation because there is no one language that is suited to all aspects of robotics. However, there are languages that can better work with specific control architectures. This chapter is basically an introduction the 4 basic control architectures: deliberative, reactive, hybrid, and behaviour-based. I am eager to learn more about each type in the upcoming units/chapters.

Questions to Ponder

Food for Thought - Chapters 10 & 11

Exercises

Exercises - SparkFun Inventor's Kit

U5 - Food for Thought

Chapter 10

What happens when you have sensor error in your system? What if your sensor incorrectly tells you that the robot is far from a wall but in fact it is not? What about vice versa? How might you address these issues?

In this case, I assume that a sensor error means some sort of inaccurate reading given by the sensor, not that the sensor is non-functioning. Therefore, if I have a sensor error in my system, that means my robot will probably not be able to function as intended. For example, when the sensor suggests that the robot is far from the wall but it is not, the robot will continue to move as if the wall is far away and eventually run into the wall because it does not expect it to be there. In the reverse situation, the robot may not be able to move past a certain point because it falsely believes that there is a wall in place, when in reality it has much more freedom to move

What can you do with open loop control in your robot? When might it be useful?

As mentioned in the textbook, open loop control works well if the environment is predictable and there are no significant changes that will affect the robot's performance. This makes it well suited to repetitive, state-independent tasks. A useful robot with open loop control could be useful in a factory. Many factories run using automated processes, the conditions in a smoothly running factory should be fairly consistent. For example, if we are building cars, we can have a robot that puts on the tires and tightens the nuts. Each car coming down the line should be essentially the same. The robot grabs a tire and puts it on. Each cycle should have little variation between them. If there is a large variation, there is likely something wrong with the process, so the robot can't do it's job, and probably shouldn't anyway. This situation is ideal for open loop control.

Chapter 11

How important is the programming language? Could it make or break a particular gadget, device, or robot?

The programming language is very important in the view that it is the tool we use to program the robot controller. Without a well-implemented controller, our robot will not be very sophisticated, and probably not very useful. In terms of the individual language used to program a robot, the choice of programming language becomes less important. The textbook mentions that robots can be programmed in basically any language. However, some languages are better suited to certain applications than others. Therefore, the more that the application lines up with a specific language, the more important the choice of language becomes.

Since we can program the robot in any language, it is unlikely to be the biggest factor that will make or break a gadget, device or robot. However, there are situations where this can be the case. For example, if we are dealing with a real-time system like jet-fighter flight controls, the choice of language can have a large impact on critical response times. A language like Java is known to run a little slower because of its automatic garbage collection routines. If the system is performing garbage clean up while the pilot is trying to perform an evasive maneuver to avoid a missile, then the device will fit decidedly into the break category.

With the constant development of new technologies that use computation, do you think there will be increasingly more or increasingly fewer programming languages?

My view is that there will always be a variety of programming languages available. The development of new technologies creates new niches where a language can specialize. No one all-encompassing language is suited to all tasks perfectly, and it seems unlikely that one will be developed. The expansion of computation into new areas of technology only creates areas that will be well suited to a particular language that handles that specific area very well. Additionally, the time, training, cost, and effort required to learn a new language is a significant investment. This can make it difficult to abandon older languages, so they will still exist and be used, often when there may be a "better" language available. Therefore, I predict that there will be increasingly more languages available, unless there is some sort of massive technological or economic breakthrough to drive people toward an individual language.

U5 - Exercises

While there are no exercises laid out in the Instructor's Notebook, I have taken it upon myself to complete the rest of the SparkFun Inventor's Kit series of circuits.

SIK Guide - Circuits #13 - #16

I've finished the series of circuits included with the SparkFun Inventor's Kit. The only issue I've had the entire time is that I broke my integrated circuit when removing it from the board
. Other than that, I've definitely learned a lot about electronics, circuits, and working with the Arduino IDE. My favourite circuit is the final one, the game Simon Says. It's fun to play, and I've been doing it on and off for the past few days. More than I've used any other circuit created so far. I hope I can put all this information to good use when designing my own project.

I have made videos of circuits 13 through 16 which can be viewed below:

YouTube