ENG 10: Lecture 6 Spring 2021

Engineering of the week

**1. Fernando Rico “Transparent solar panels”**

**2. Tiffany Soebroto “Drug delivery nanoparticles for cancer treatment”**

Please, complete this feedback form before next class

(also posted on Canvas - Modules – Links, Forms, Quizzes)

Week 6 Feedback Form:

● https://forms.gle/HGGZa5ZRJ8oYd3DY7

**Why Linear Algebra is important?**

• **How many variables** does this landing 

system need?

• **How many equations** does this landing system need?

• All the computations have to be done **in real-time** for a high accuracy of landing.

☺

Linear Equations

Start with a single linear equation:

*y* = *mx* + *b*

output slope input offset

3x3 System of Linear Equations

*a*1*,*1*x*1 + *a*1*,*2*x*2 + *a*1*,*3*x*3 = *b*1 *a*2*,*1*x*1 + *a*2*,*2*x*2 + *a*2*,*3*x*3 = *b*2 *a*3*,*1*x*1 + *a*3*,*2*x*2 + *a*3*,*3*x*3 = *b*3

3x3 System of Linear Equations Answer: YES!

2 4

3

*a*1*,*1 *a*1*,*2 *a*1*,*3 5

*a*2*,*1 *a*2*,*2 *a*2*,*3 *a*3*,*1 *a*3*,*2 *a*3*,*3

2 4

*x*1 *x*2 *x*3

3 5

2 4

*b*1 *b*2 *b*3

3 5

**A x b**

System of Linear Equations

What about bigger systems (e.g. 5x5, 10x10, 106x106)? **Same ideas apply:**

1) Group the components

2) Rewrite as Ax = b

3) Solve

System of Linear Equations

• But how do you solve it???

• We are unable to calculate the values in real time.

• … We can quickly compute Ax = b problems in Python with the following NumPy Module: ``x = np.linalg.solve (A, b)``

Math: Probability and Statistics

Probability

**② Chance of winning a lottery** Probability: Introduction ▪ What is probability? 

**- How likely something is to happen.**

**① Chance of rain in San Diego** ☺**Jackpot odds of average state lottery: 100,000,000 to 1** 

**③ Chance of being struck by lightning**

****

**The estimated chance of an average person living**

**Roughly 0 (usually)**☺

**in the US being struck by lightning a year: 960,000 to 1**

Probability: Introduction

▪ Tossing a coin: 

**Head Tail**

!

!

" **Probability**

"

✔ Probability of Head or Tail: 1/2

▪ Throwing a dice:



!

!

!

!

!

!

# **Probability**

#

#

#

#

#

✔ Probability of the any one of the faces: 1/6

Probability: Probability scale

▪ **Probabilities can be written as:**

**① Fractions from 0 to 1**

**② Decimals from 0 to 1**

**③ Percentages from 0 % to 100 %**

▪ **Probability Scale:**

1-in-6 chance 

Even chance

Impossible

Unlikely Likely

Certain

0 0.5 1 or 

**1/100,000,000 146 sunny days + 117 partly cloudy days in San Diego a year = 263 days (0.72)** 

Probability: Difference b/w theoretical value and actual result ▪ Interactive: Coin toss

http://www.shodor.org/interactivate/activities/Coin/ 

• Try tossing a coin **10** times

• Try tossing a coin **100** times

• Try tossing a coin **1000** times

Discussion about the results

• What is the difference b/w the theoretical probability value and the real result? For 10, 100, and 1000 times

• How does the result change with the number of trials?

• Why is the experimental result different from the theoretical value?

• How do we narrow the gap b/w theoretical value and experimental results?

Probability Review: Venn diagram

▪ Diagram that shows **all possible logical relations** between a finite collection of different sets using circles/closed curves

**Intersection of two sets A∩B**

**Union of two sets A**∪**B**

****

**Relative complement of A in B**

**Ac∩B = B\A Ac = U\A**

****

Probability: Mathematical treatment

▪ **JOINT probability for Independent events:**

**- When two events A and B are independently occur on a single experiment** 

**- P(A and B) = P (A ∩ B) = P(A)·P(B)**

**-** *e.g. if a pair of dice are rolled, the chance of both being 6 is: *and

"

"

"

#

$# x = #

▪ **Mutually EXCLUSIVE events:**

**- When either event A or event B occurs on a single experiment - P(A or B) = P(A**∪**B) = P(A) + P(B)**

**-** *e.g. the chance of rolling a 3 or 6 is P(3 or 6):*

**or% &

"

"

=

"

#

+ $ #

Probability: Mathematical treatment

▪ **NOT mutually EXCLUSIVE events**

**- When the events are not mutually exclusive**

**- P(A or B) = P(A) + P(B) – P(A and B)**

- *e.g. the chance of getting* ***a diamond or an alphabet card***

******

or

#$ += &#$ -

!" #$

!% #$

$#

Probability: Probability distribution

▪ **The distribution of a random variable is a collection of possible outcomes along with their probabilities:**

▪ **Binomial Distribution (symmetric): Discrete probability distribution**

****

▪ **Normal (or Gaussian) distribution (symmetric): Very common continuous distribution**

****Central limit theorem !

**Weibull** statistics for survival probability of a particular component as a function of its volume and the applied stress

**Probability of survival Probability of failure**

σ = applied stress

σu = stress below which there is a zero probability of failure

- this means there is an upper limit to flaw size

- for brittle materials σu = 0 (since any tensile stress might cause failure) σo = characteristic strength ~ **mean strength** of the material

m = **Weibull** modulus

- characterizes the variability in strength ~ **standard deviation** of material’s strength

**Weibull** plot

**Silicon carbide (SiC)** 

σo = characteristic strength

%

,

)

V

(

slope = m = 10

P

37%

σ = σo

Mean failure strength

**Weibull** analysis

• **Increasing m values reflect** more homogeneous material behavior with **strength levels** for a given component being **more predictable**.

• In the limit where m

approaches ∞, the probability of failure = 0 for all stress levels < #0.

• In the limit where m

approaches 0, the probability of failure 1, and failure occurs with equal certainty at any stress level.

Statistics

What is Statistics?

**Definition:** The mathematical and scientific methodologies for collecting, organizing, analyzing, and drawing conclusions from data or information to:

- **Make decisions, predictions, or plans**

- **Solve problems**

- **Generalize uncertain phenomenon and events**

****Salary Negotiation Plan for College Applications Global Warming

Data collection methods (**sampling**)

- **Surveys**

****

- **Interviews**

- **Observations** - **Experiments **

Importance of accuracy: Watch out for **Faulty data**!

What issues might we encounter

when collecting data?

Importance of accuracy: Watch out for **Faulty** data!

- **Lies in survey**

**- Samples do not represent the whole.**

- **Failed in controlling any independent variables that affect results**

****- **Not enough # of sampling**

▪ Accurate data collection is 

essential to minimize errors

▪ Otherwise, these unreliable

data bring wrong decisions

or distorted findings

How to represent data: Bar Chart and Histogram

**- Bar Chart (Discrete & Categories) - Histogram (Continuous & Ranges)**

Histograms are a great way to show results of continuous data, such as: Weight, height, time…

How to represent data: Line Graphs and Pie Charts

**- Line Graph:** Good to observe **trends**

**- Pie Chart:** Good to describe **numerical proportion**

****

Descriptive statistics

A way to provide simple and straightforward summaries of data from the sample and the measures in form of numbers or graphs.



The shooting percentage in basketball is a descriptive statistic

that summarizes the performance of a player or a team.

Inferential statistics

A procedure for inferring or predicting future or a population by using data from the population with some form of sampling (**Very Complicated**)

Population 

Sample

**Making**

**conclusions**

**Representative**

**Sample**

**Population**

Central tendency

**Definition:** A measure of central tendency is **a representative value** to describe a set of data by identifying the central position within that set of data.

Representative values: **Arithmetic Mean, Median, and Mode**

Central tendency: Arithmetic mean

**Arithmetic Mean:** The mean is equal to the sum of all the values in the data set divided by the number of values in the data set.

**Example:**

****

Staff 1 2 3 4 5 6 7 8 9 10 Salary 15k 18k 16k 14k 15k 15k 12k 17k 90k 95k

15,000 + 18,000 + 16,000 + 14,000 + 15,000 + 15,000 + 12,000 + 17,000 + 90,000 + 95,000 10

= $ 30,700

**Disadvantage of Mean**: Not the best way to reflect the population!!!

Central tendency: Median

**Definition:** The **middle score** for a set of data that has been arranged **in order of magnitude.**

For the set of salaries:

Salary 15k 18k 16k 14k 15k 15k 12k 17k 90k 95k Rearrange from smallest to largest salary:

Salary 12k 14k 15k 15k 15k 16k 17k 18k 90k 95k

**Median** = 12k, 14k, 15k, 15k, 15k, 16k, 17k, 18k, 90k, 95

Because there are an even number of data points,

Take the mean of the middle numbers…

Median = 15.5k

Central tendency: Mode

**Definition:** The mode is **the most frequent score** in our data set. It represents the highest bar in a bar chart or histogram. Gives an idea of the most dominant samples in the data set.

In our salary data set, the dominating salary amount is $15k, as it appears the most in the data set.

Salary 12k 14k 15k 15k 15k 16k 17k 18k 90k 95k

Central tendency:

Symmetric, positively and negatively skewed data **Mode < Median < Mean Mode = Median = Mean Mean > Median > Mode**

Variance

**Definition:** Variance is a measurement of the spread between numbers in a data set. The variance is a distance measure telling us **how far each number** in the set is from **the mean**.

Denoted as "2:σ12

"2 =∑ ( − ! 2

\*

X: individual data point ! : mean of data points N: total # of data points

σ22

σ12 < σ22

Standard deviation

**Definition:** A measure that is used to quantify the amount of variation or dispersion of a set of data values. This can also be considered as a distance measure.

Denoted as σ which is the square root

of its **variance**: 

! =∑ $ − & 2

(

X: individual data point

& : mean of data points

N: total # of data points

Homework 5 (probability and statistics)

**This week’s homework will use Python to explore probability concepts through simulated dice rolls and to calculate statistical values from known data**

****

1. Download the Homework 5 file from the Canvas website

2. Complete the homework by (1) filling in your own code where you see ??? and where asked to give a second example and (2) answering questions in “markdown” cells

3. Submit homework through Canvas using proper file names/types

39

HW 5

• **HW5 is due 11:59 pm on Wednesday (May 12th)**

• Complete all codes and answer questions (inside the notebook) • Rename the notebook file with your name

• Submit via Canvas

Design Proposal

(several helpful thoughts)

Design under constraints

• You never have **unlimited** time, money, etc.

• Some constraints **overlap** (e.g. small and lightweight)

• Some constraints **conflict** with each other (e.g. low cost and low weight)

Mission statement

• Describe the **overarching** goal(s)

• Doesn’t necessarily included quantified requirements

• Usually given (in part) by a client

Design requirements…

• Quantify your goals

• Guide you towards solutions

• Provide justifications for design decisions

Source of design requirements

• Sometimes provided by a client (directly or indirectly) or external source (e.g. government regulation)

• Usually requires work to get “hard” numbers

• Typically are “minimum” values (i.e. you would expect to do “better”)

Example 1: Sizing a pizza

• **Okay:** The pizza should be pizza size

• **Better:** The pizza should fit in the box

• **Better x2:** The pizza should have a diameter of 12” +/- 0.5” and height less than 2”

Example 2: Car frame

• **Okay:** The car frame should survive a crash

• **Better:** The car frame should not deform when hit by another car

• **Better x2:** The frame should deflect less than 1” when hit by a 1.5 ton vehicle traveling at 10 mph

Example 3: Storing medicine for shipping

• **Okay:** The medicine stays cool

• **Better:** The medicine stays below 50°F

• **Better x2**: The medicine’s internal temperature stays in the range of 20–50°F, with a median temperature of 40°F

Design Challenge

Example project ideas: Sound

• **Idea:** Provide a non-verbal warning to runners with headphones if a loud car/truck is approaching 

• **Input:** Sound sensor => Frequency data

• **Output:** Vibrating motor



Example project ideas: Accelerometer

• **Idea:** Increase visibility (and fun) for snowboarders at night by adding a motion-controlled light display

• **Input:** Accelerometer => Detect motion 

• **Output:** LEDs

50

1 Create the code 2

Make the connections



3 Upload the code

4 Test it





Arduino External Outputs

Arduino Microcontrollers

compute

Sensorread **Adafruit Circuit Playground**

react

Actuator

**Output** Options (Actuators)

• **Built-in devices on the Circuit Playground** • Speaker 

• LEDs

• **Today’s external actuator**

• Servo motor

Other Circuit Building **Tools**

• CPX 

• Breadboard

• Jumper wires and alligator clips



Breadboards

**Fundamental tool to build circuits by allowing multiple connections between devices **

Metal connects each row beneath the plastic which allows for connections between each element plugged into the ***terminal strips*** (rows)

Most breadboards also have ***power rails*** which are two columns on each end that are connected to each other

https://learn.sparkfun.com/tutorials/how-to-use-a-breadboard/all

Arduino Tutorial: External Actuators

We have to finish parts 2.3, 3.1 and 3.2 from this tutorial (**Lab 2**) begore we will move to **Lab 3**. **Main steps are:** 

**1. Download the library bundle (Bundle version 6.X, see next slide) and save it on your computer**

**2. Use certain files from this bundle when needed by**

**imported those files directly to your CIRCUITPY (library)**

Full Arduino tutorial is available on Canvas in Tutorials Module **“Arduino CPX Tutorials” zip** Folder

**2.3 (Lab 2)** Using a Buzzer via *SimpleIO (Optional)* We can also use **a different library** to accomplish the same buzzer actuation

1. Open Serial and hit CTRL+C to find the version of your software (**see figure below**) 2. Download the corresponding ***bundle*** *(***Bundle version 6.X***)* from https://circuitpython.org/libraries **3. Save bundle to desktop** then unzip folder (**extract all**)

Note: You can put this folder in other areas but may run into “pathway too long” errors

4. In unzipped folder > lib > search: **simple.io** → copy and paste in **CIRCUITPY > lib** 5. Add new code from tutorial using **simple.io** and save as ***code.py***

******

**3.1 (Lab 2)** Responding to Sensors on the CPX (Optional)

This example demonstrates how **the amount of light changes the position of servo**

**Our first several examples relied on set schedules for actuator behavior. Here we will learn how we can use inputs from sensors to control external actuators**

**1. First, we will need more software from the bundle—copy the full folder *adafruit\_motor* into CIRCUITPY > lib** 

2. Add **a fin** to the servo motor top

3. Connect the servo motor to CPX using the following:

**1. Brown** is ground, **connect to GND**

**2. Red** is power, connects to **3.3V**

**3. Orange** is PWM (Pulse-Width Modulation, controls the motor), **connect to A1**

4. Input the new code from the tutorial and save as *code.py* on your CPX

1. If not working, check Serial w/ Ctrl+D

2. If it says not working because safe mode, eject CPX and reload

**We can do this one without the breadboard!**

https://learn.adafruit.com/adafruit-circuit-playground-express/circuitpython-servo

**3.1 (Lab 2)** Responding to Sensors on the CPX (Optional)

This example demonstrates how **the amount of light changes the position of servo. There are minor changes into the Tutorial program. Please consider to check this program!**

import board

import pulseio

from adafruit\_circuitplayground.express import cpx

from adafruit\_motor import servo

from time import sleep

# Setup Section

pwm = pulseio.PWMOut(board.A1, duty\_cycle=2 \*\* 15, frequency=50)

servo = servo.Servo(pwm, min\_pulse=750, max\_pulse=2600)

# Function Section

def light\_to\_servo\_pos(light):

light\_max = 320

return 180 - ((light/light\_max) \* 180)

# Loop Section

while True:

servo.angle = light\_to\_servo\_pos(cpx.light)

print(servo.angle)

sleep(0.5)

**3.1 (Lab 2)** Responding to Sensors on the CPX (Optional)

This example demonstrates how **the amount of light changes the position of servo**

****63

**3.2 (Lab 2**) Another example that shows how **servo rotates between 0 and 90 degrees**

**Our first examples all relied on set schedules for actuator behavior. Here we will learn how we can use inputs from sensors to control external actuators**

**1. First, we will need more software from the *bundle*—copy the full folder *adafruit\_motor* into CIRCUITPY > lib**

2. Add **a fin** to the servo motor top

3. Connect the servo motor to CPX using the following:

**1. Brown** is ground, **connect to GND**

**2. Red** is power, connects to **Vout**

**3. Orange** is PWM (Pulse-Width Modulation, controls the motor), **connect to A2** 4. Input the new code (**that Fabian sent, see next slide**) and save as *code.py on your CPX* 1. If not working, check Serial w/ Ctrl+D

2. If it says not working because safe mode, eject CPX and reload

**We can do this one without the breadboard!**

https://learn.adafruit.com/adafruit-circuit-playground-express/circuitpython-servo

**3.2 (Lab 2)** Program for Servo Motor to show **how servo rotates between 0 and 90 degrees**

import time

import board

import pulseio

from adafruit\_motor import servo

# create a PWMOut object on Pin A2.

pwm = pulseio.PWMOut(board.A2, duty\_cycle=2 \*\* 15, frequency=50)

# Create a servo object, my\_servo.

my\_servo = servo.Servo(pwm)

while True:

for angle in range(0, 90, 5): # 0 - 180 degrees, 5 degrees at a time.

my\_servo.angle = angle

time.sleep(0.05)

for angle in range(90, 0, -5): # 180 - 0 degrees, 5 degrees at a time.

my\_servo.angle = angle

time.sleep(0.05)

**Input** Options (Sensors) 

**Built-in sensors on the Circuit Playground**: Sound: Noise levels, sound signals, frequencies Temperature

Light: Brightness, color (using LEDs)

Accelerometer: Acceleration, tilt

**Today’s external sensor**

Proximity: Measures distance of object from sensor by detecting ultrasonic wave reflection

Libraries for different sensors (example) • In order to figure out everything regarding **proximity sensor (for example)** please go to: • https://learn.adafruit.com/using-vcnl4010-proximity-sensor/

• Read everything carefully, **press "Pinouts"** on the right bottom corner and again read everything carefully. • Next, **press "Arduino"** on the right bottom corner, and check what kind of steps you have to do in order to install Adafruit\_VCNL4010 library and a demo.

Arduino Tutorial: External Sensors

Lab 3: External Sensors for Arduino with 

CircuitPython

Introduction

At this point, we have covered most of the important concepts

and setup for designing and developing on Arduino with Circuit

Python. This lab focuses on using external sensors (input devices),

which is very similar to using external actuators. As with

actuators, the CPX has many built-in sensors, but there may be

additional sensors (like ow meters) that need to be attached for a

given project. In addition, most other Arduino devices don't have

as many built-in sensors, so all sensors for a project must be

attached externally. We will then provide examples combining

internal and external sensors and actuators.

Full Arduino tutorial is available on Canvas in **Arduino CPX Tutorials.zip** folder

**. a** onnectng an xterna roxmty ensor (**shows the distance from the external proximity sensor in cm**)

Page 3

5. Enter the code below in Mu, and save to the CPX as **code.py**

**Import** the external library 

“*adafruit hcsr04.mpy*” into

the **lib** folder in **your CPX**

70

**1. (Lab 3)** Connecting an External Proximity Sensor to CPX

71

**2.1 (Lab 3).** Combining Ext. Sensors and Actuators “**Add a needle gauge to the proximity sensor” for mapping proximity to servo position**

**No need to disassemble the proximity sensor setup!**

Just add the servo motor connections!

**1. Brown is ground**, connect to GND

**2. Red is power**, connects to 3.3V

**3. Yellow is control** (Pulse-Width Modulation (PWM)), connect to A1

**“Add a needle gauge to the proximity sensor”**

73

**2.2 (Lab. 3).** Multiple Actuation

**“Add warning lights/sounds to the proximity sensor”**

**No need to disassemble the proximity sensor/servo motor setup!** Just change the code (**from pp. 6-7**) to add additional actuation of warning lights and sounds!

