### PRESENT COURSE DESCRIPTION

<table>
<thead>
<tr>
<th>Title</th>
<th>Tools in Electrical and Computer Engineering</th>
</tr>
</thead>
<tbody>
<tr>
<td>Semester Hours of Credit</td>
<td>2</td>
</tr>
<tr>
<td>If combination course type, # hrs. of credit for</td>
<td></td>
</tr>
<tr>
<td>Lecture:</td>
<td>Lab/Sem/Rec:</td>
</tr>
<tr>
<td>Repeat Credit Max. (if repeatable):</td>
<td>NA</td>
</tr>
<tr>
<td>Graduate Credit?</td>
<td>Yes</td>
</tr>
<tr>
<td>Credit will not be given for this course and:</td>
<td></td>
</tr>
<tr>
<td>Contact Hours Per Week: (Indicate hours in appropriate course type.)</td>
<td></td>
</tr>
<tr>
<td>Lecture</td>
<td>Lab</td>
</tr>
<tr>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Total Weekly Contact Hours:</td>
<td>2</td>
</tr>
<tr>
<td>Grading System:</td>
<td>Letter Grade</td>
</tr>
<tr>
<td>2810 Tools in Electrical and Computer Engineering (2)</td>
<td></td>
</tr>
</tbody>
</table>

**2810 Tools in Electrical and Computer Engineering**

(2) Prereq: CSC 1253 and credit or registration in EE 2130 and 2231. Contemporary tools in the area of electrical and computer engineering.

### PROPOSED COURSE DESCRIPTION

<table>
<thead>
<tr>
<th>Title</th>
<th>Tools in Electrical and Computer Engineering</th>
</tr>
</thead>
<tbody>
<tr>
<td>Short Title</td>
<td></td>
</tr>
<tr>
<td>Semester Hours of Credit</td>
<td>2</td>
</tr>
<tr>
<td>If combination course type, # hrs. of credit for</td>
<td></td>
</tr>
<tr>
<td>Lecture:</td>
<td>Lab/Sem/Rec:</td>
</tr>
<tr>
<td>Repeat Credit Max. (if repeatable):</td>
<td>NA</td>
</tr>
<tr>
<td>Graduate Credit?</td>
<td>Yes</td>
</tr>
<tr>
<td>Credit will not be given for this course and:</td>
<td></td>
</tr>
<tr>
<td>Contact Hours Per Week: (Indicate hours in appropriate course type.)</td>
<td></td>
</tr>
<tr>
<td>Lecture</td>
<td>Lab</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Total Weekly Contact Hours:</td>
<td>3</td>
</tr>
<tr>
<td>Grading System:</td>
<td>Letter Grade</td>
</tr>
<tr>
<td>2810 Tools in Electrical and Computer Engineering (2)</td>
<td></td>
</tr>
</tbody>
</table>

(2) 1 hr. lecture; 2 hrs. lab. Prereq: CSC 1253 and credit or registration in EE 2130 and 2231. Contemporary tools in the area of electrical and computer engineering.

---

**THESE QUESTIONS MUST BE ANSWERED COMPLETELY AND ACCURATELY OR PROPOSAL WILL BE RETURNED.**

Has this change been discussed with and approved by all departments/colleges affected? Yes | No |

Is this course included in any curricula, concentrations, or minors? Yes | X | No |

If yes, please list on a separate sheet.

Is this course a prerequisite or corequisite for other courses? Yes | X | No |

If yes, list courses; use separate sheet.

Is this course on the General Education list? Yes | No | X |

**JUSTIFICATION/EXPLANATION:** Use separate sheet.

**Note:** IF COURSE IS OR WILL BE CROSS-LISTED, SEPARATE FORMS MUST BE SUBMITTED BY EACH DEPARTMENT.

### APPROVALS

**Department Faculty Approval Date: 9-18-17**

[Signature]

Digitally signed by Jerry L. Trahan

Date: 2017.09.19 10:32:57 -06'00'

[Signature]

Graduate Dean Signature (date)

[Signature]

College Dean Signature (date)

[Signature]

Chair, FS C&C Committee (date)

[Signature]

Academic Affairs Approval (date)
Justification: Changing EE 2810 from a 2 hour lecture class (2 credits) to a 1 hour lecture 2 hr lab (still 2 credits)

EE 2810 is a course that teaches computer software. There are three main reasons why this class needs to be changed to a lecture/lab course:

1.) Large class sizes: The large class sizes make it difficult to teach in a large computer classroom. Breaking the students up into smaller groups to work on computer projects will give the department more flexibility to schedule the class given the computer classroom facilities available. Having a 1 hour lecture each week in a traditional classroom will also be useful to administer lectures and exams. It is difficult lecturing to 90+ students in a large computer classroom. The current computer room that holds 96 is not designed for lectures. It is designed for students to work on computer projects. It is also no longer available to ECE. With the completion of PFT, it is now controlled by ChemE. Only one new computer classroom is 70, and the rest are about 36 each.

2.) Access to software: Students need more time in a computer lab with access to the software. Currently, lectures take up half or more of the two hours available each week. Students have difficulty accessing the labs (software) outside of regular class time. The additional contact hour will be entirely devoted to computer lab access to work on projects, and get help when needed.

3.) If it were to continue strictly as a lecture course, the ECE division would require additional resources to teach multiple lecture sections in the smaller computer classrooms. The division does not have these resources.

This course is required for students majoring in:

Electrical Engineering
Computer Engineering

This course is a prerequisite for:

EE 3160, EE 3530, EE 3610, EE 4745, EE 4780

Registration in this class requires credit or registration in:

EE 2130 and EE 2231
Louisiana State University
Division of Electrical and Computer Engineering
EE 2810 Tools in Electrical and Computer Engineering
Fall 2018

Lecture 1 hr per week
Lab 2 hrs per week

Catalog Data: EE 2810 Tools in Electrical and Computer Engineering (2) 1 hr lecture, 2hrs. lab; Prereq.: CSC 1253 and credit or registration in EE 2130 and 2231. Contemporary tools in the area of electrical and computer engineering.

Instructor: John Scalzo, MSEE Georgia Tech, BSEE Virginia Tech
Patrick F. Taylor 3335
Phone: 578-5478 e-mail: lscalz1@lsu.edu
Experience: High Frequency (RF) Systems Design Engineer
Specialization: DSP, Analog and Digital Communications, RF Circuit design
Office Hours: This and other information can be found at my website:
http://www.ece.lsu.edu/scalzo/index.htm


For EE 2810, it is possible that the LSU Libraries’ collection includes an e-book of the required textbook: PSPICE and MATLAB for Electronics. This e-book is free for students (and faculty) and can be used by the entire class at once. All chapters can be downloaded, saved, and printed for use now or after the class. Access is available from this direct link:


or from the e-textbook webpage: www.lib.lsu.edu/ebooks.

Prerequisite by Topic: Linear circuits, Linear algebra, elements of programming

Goals/Instructional Objectives: This course is intended to provide students with a basic knowledge of contemporary software tools such as PSpice, Matlab, Simulink, and Eagle PCB to solve problems and for analysis and design in the field of electrical and computer engineering. Specifically, students will understand and learn how to apply these tools for different applications. The course will prepare students for more advanced courses in the EE and EEC programs that use these tools.

Course Learning Outcomes: At the end of the course, the students will be able to:
(a) Create Matlab based graphics, functions, time responses
(b) Write programs in Matlab in various settings - typical to electrical and computer engineering applications
(c) Create GUIs and Simulink simulations
(d) Analyze dc, steady - state ac and transient response of linear circuits using PSpice
(e) Analyze circuits in the time domain and frequency domain using PSpice

**Grading:**

- Homework/Classwork: 20%
- Projects (Three): 30%
- Tests (Three): 30%
- Final Exam: 20%

Grading Scale: Note the interval notation.

<table>
<thead>
<tr>
<th>Grade</th>
<th>Range</th>
<th>Grade</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>A+</td>
<td>$[\infty,97]$</td>
<td>C+</td>
<td>$[80,77]$</td>
</tr>
<tr>
<td>A</td>
<td>$[97,93]$</td>
<td>C</td>
<td>$[77,73]$</td>
</tr>
<tr>
<td>A-</td>
<td>$[93,90]$</td>
<td>C-</td>
<td>$[73,70]$</td>
</tr>
<tr>
<td>B+</td>
<td>$[90,87]$</td>
<td>D+</td>
<td>$[70,67]$</td>
</tr>
<tr>
<td>B</td>
<td>$[87,83]$</td>
<td>D</td>
<td>$[67,63]$</td>
</tr>
<tr>
<td>B-</td>
<td>$[83,80]$</td>
<td>D-</td>
<td>$[63,60]$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>F</td>
<td>$[60,0]$</td>
</tr>
</tbody>
</table>

**14 Week Course Outline**

**Week 1**

**Lecture:** Introduction of the syllabus, explanation of the software, its availability in the different labs on campus and the instructions or cost to purchase for personal use. Printed Circuit Board Introduction.

**Lab:** lab will not meet during week 1

**Week 2**

**Lecture:** Eagle PCB, the schematic, Present the Eagle Project, How to create a self-sustaining document

**Lab:** Create an Eagle Schematic

**Week 3**

**Lecture:** Eagle PCB, the board file, ground planes

**Lab:** Create an Eagle board file, begin working on the project

**Week 4**

**Lecture:** Processing a CAM job, creating the gerber files

**Lab:** Continue working on the Eagle Project
Week 5

Lecture: Eagle Test

Lab: Finish working on the Eagle Project

Week 6

Eagle Project Due

Lecture: PSPICE Fundamentals using Orcad Capture (DC, transient, AC analysis)

Lab: Work on Chapter 1 problems using (Attia)

Week 7

Lecture: PSPICE Fundamentals using PSPICE AD (DC, transient, AC analysis), PSPICE Project presentation

Lab: Work on Chapter 2 problems, begin PSPICE project

Week 8

Lecture: Advanced PSPICE (Part 1) – Device models

Lab: Chapter 3, 6, 7,8 ,PSPICE Project

Week 9

Lecture: PSPICE Test

Lab: Chapter 7, 8, PSPICE Project

Week 10

PSPICE PROJECT DUE

Lecture: Introduction to Matlab (Chapter 4, Attia)

Lab: Matlab Lab 1 – problems from Chapter 4

Week 11

Lecture: Matlab – Scripts, functions; Introduction to the Matlab Project
Lab: Matlab Lab 2 – problems from Chapter 5, begin Matlab Project

Week 12

Lecture: Importing and Exporting Data using Matlab, Introduction SIMULINK

Lab: Matlab Project

Week 13

Lecture: Creating GUI’s using Matlab

Lab: Matlab Project

Week 14

Lecture: Matlab Test

Lab: Matlab Project, Matlab Project Due

Documentation requirements:
All project reports submitted for grading must be self-sustaining. In this context, the document should contain any and all information necessary so that no other materials are necessary for completely understanding the document’s contents. Points will be deducted for improper documentation. Reports must include all proper aspects of a technical document.

Analysis:
This course requires a significant amount of analysis. Students will be introduced to many tools for analyzing content. However, in the reports quantitative and qualitative analysis is required.

Here is a useful resource for writing reports. Using simulation software is similar to executing a laboratory experiment.

http://www.sussex.ac.uk/ei/internal/forstudents/engineeringdesign/studyguides/labwriting

Missed exams and assignments:
Make-up tests will be given only for excused absences (according to PS-22). Students must notify me PRIOR to a test via email or call to the ECE main office 578-5241. Test 1 must be made up before test 2. Test 2 must be made up before the final exam. No late work will be accepted.

Academic Dishonesty:
Academic dishonesty of any kind will not be tolerated. Anyone suspected of cheating will be reported to the Dean of Students. Homework done outside of class can be done in small groups (less than 5), but borrowing or copying work is not allowed. Projects CANNOT be done in groups. Students can ask questions to each other how to use the software, but no concepts or work related to the project can be discussed. Students cannot seek help from anyone not affiliated with this course outside of the class for any projects unless approved by the course
instructor BEFORE an appointment is made.

**Attendance and Tardiness:**
There will be no attendance or tardiness policies for this class. However, some in class work will be assigned for a grade. There will be no opportunities for make up if in class work is missed. Opportunities will arise for one or more of these assignments to be dropped to accommodate the occasional excused absence.

**Classroom Decorum (Professionalism):** The rationale behind this classroom decorum is to teach skills and prepare you for the requirements of the engineering profession.

Students are expected to conduct themselves in accordance with accepted standards of classroom behavior. This means arriving in class on time, being respectful to your fellow students, remaining in class for the full time, listening attentively to class instructions and discussion, and refraining from carrying on personal conversations, unless specified to do so. Cell phones and other electronic devices should be silenced while in the classroom. Laptops are welcome in the class provided they are used for class-related purposes. Class-related purposes DO NOT include assignments, unless specified permission is given. Their misuse, like the use of cell phones and other electronic devices, will hurt your CLE grade. Failure to adhere to good classroom decorum, as described above, disrupts class and limits everyone’s participation.

**Disabilities:**
The University is committed to making reasonable efforts to assist individuals with disabilities in their efforts to avail themselves of services and programs offered by the University. To this end, Louisiana State University will provide reasonable accommodations for persons with documented qualifying disabilities. If you have a disability and feel you need accommodations in this course, you must present a letter to me from Disability Services in 115 Johnston Hall, indicating the existence of a disability and the suggested accommodations.
Topics:

(a) *Topics in Matlab*

(i) **Basic Topics**
- Matrices, Arrays and their mathematics
- Graphics – Plotting 2D, 3D; Plot labels, log, semilog; sub plots
- Programming -.m files; Matlab functions
- Vectors – vector algebra (convolution)
- Creating Functions – Traditional functions, Inline functions
- Linear algebra – matrices
- Complex numbers – Complex vectors, Mag(), angle() functions
- Complex models – Tf, zpk, Itimodels
- Time responses – Step, impulse, initial, lsim, ltvview
- Solving transforms using Matlab – Residue (partial function exapansion); Tf2pz should be tf2zpf; P2xtf should be pz2tf; Bode plots, bode, sisotool
- Importing data into Matlab using .csv files and the csvread() function

(ii) **Advanced Topics**
- GUIs, Simulink (Examples from Electrical and Computer Engineering)

(iii) **(Test - 1)**

(b) *PSpice*

- Introduction: Description of SPICE, Types of SPICE, Types of Analysis, Limitations of SPICE
  - Circuit Description: Introduction, Element Values, Nodes, Circuit Elements, Sources, Types of Analysis, Output Variables, PSpice Output Commands, Format of Circuit Files, format of Output Files, Examples of PSpICE simulations, Graphical Input Files
- D.C. Circuit Analysis: Introduction, Resistors, Modeling of Elements, Operating Temperature, Independent Sources, Dependent Sources, DC Output Variables, Types of Output, Types of DC Analysis, Examples of PSpICE Simulations
- Transient Analysis: Introduction, Capacitors, Inductors, Diodes, and Transistors, Modeling of Transient Sources, Transient Sources; transient Output Variables, Transient Output Commands, Transient response, Switches (Voltage-Controlled Switch, Current-Controlled Switch), Examples of PSpICE Simulations
- Circuit Simulations Using PSpice for DC and Transient Analysis during class hours with remote login to PSpice.
- Circuit Simulations Using PSPICE for AC Circuit Analysis during class hours with remote login to PSPICE. Interfacing Spice to Matlab

(c) Printed Circuit Board Design (4 lectures, 2 lecture/project, 1 Test)
- Getting started with OrCAD Capture, Building a simple schematic, Processing a design
- Building a multi-sheet schematic, Editing properties
- Creating parts and symbols, Building a hierarchical design
- Preparing a design for Layout
- (lecture/project) Circuit assembly on a PCB design using PCB boards and soldering iron
- (lecture/project) Circuit assembly on a PCB design using PCB boards and soldering iron.

**Relationship of Course to ABET Student Outcomes:**

<table>
<thead>
<tr>
<th>The course contributes to these ABET outcomes:</th>
<th>How?</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1. An ability to apply knowledge of mathematics, science and engineering</td>
<td>Students will use engineering specific software to obtain solutions to electrical and computer systems</td>
</tr>
<tr>
<td>#5 an ability to identify, formulate and solve engineering problems</td>
<td>The course provides a basic understanding of the operation of different software packages and their applications so that students are prepared for advanced courses and engineering practice</td>
</tr>
<tr>
<td>#9 a recognition for the need for adequate preparation for continued professional growth and life-long learning</td>
<td>Software packages are update often. Industry leaders also change. It is necessary for engineers keep up with these packages. Engineers must also be prepared to use industry specific or proprietary software once they enter the workplace</td>
</tr>
<tr>
<td>#11 an ability to use techniques, skills, and modern engineering tools necessary for engineering practice</td>
<td>Students learn to use software for analysis simulation and design of electrical and computer systems</td>
</tr>
</tbody>
</table>
Most of you took EE 2120 in the spring of 2017. In that class, you built and analyzed a Wheatstone bridge project using Multisim. For reference, I posted that project on the course moodle page for EE 2810. The circuit is below.

The details of the circuit are not important for this project. Many of you already did an extensive analysis. The circuit has many applications, one of which is to monitor the structural integrity of a road bridge. In reality, the output voltage of the Wheatstone bridge is quite small, and must be amplified. The circuit below is an excellent amplifier for this application. You will learn to design this circuit in EE 3220, if you decide to take that class. I will demonstrate this circuit in class, but understanding how the circuit works (known as a differential amplifier) is not important. Here is the circuit below:
For this project, you are going to design a single PCB that will contain connectors and components for both the Wheatstone bridge and the differential amplifier. The complete circuit, modeled on Multisim looks like this:

For the Wheatstone Bridge:

- R11 is a potentiometer. The maximum value of the potentiometer doesn’t matter exactly. It could anywhere from 25 kΩ to 50 kΩ. The purpose of this component was to allow tuning after installation to set $V_x$ (Figure 1) to 5 volts. This component will not need to dissipate more than $\frac{1}{4}$ watt.
- R7, R8 and R10 are 25kΩ, $\frac{1}{4}$ watt resistors.
- R9 is not on a resistor on the PCB. It represents a flex sensor: SEN_FLEX2P2_DE3 from CuteDigi.com found here: http://store.cutedigi.com/flex-sensor-2-2/?gclid=Cl7Okv6h79ECFY-1wAodZawAyg. You need to design a connector for the wires of the sensor to connect to the PCB. The other end of the sensor will be placed on a beam of a bridge.
- Instead of a 12 V supply, we will use a 9 volt battery placed on the PCB in a battery socket. I purchased a breakout board for the myDAQ device that has one of these 9 volt sockets built into the board. See below.
For the differential amplifier:

- IO1 – inverting output of the differential amplifier, needs to leave the PCB, paired with IO2
- IO2 – non-inverting output of the differential amplifier, needs to leave the PCB, paired with IO1
- The positive terminal of a 9 volt battery will be the same positive terminal used to power the Wheatstone bridge.
- A second 9 volt battery will power the “bottom” of the differential amplifier. We cannot use the same 9 volt battery mentioned earlier because the connection is reversed. For this battery, the positive terminal is grounded, and the negative terminal is connected to R5, the 500 Ω resistor on the differential amplifier.
General

Terminals A and B are simply used to show a connection from the Wheatstone bridge to the differential amplifier. I just didn’t want to clutter the Multisim circuit with long wire connections. There is no way around this on the PCB; you have to find a way to make these connections. IO1 and IO2 can connect to a microprocessor. From the microprocessor we can record or transmit data for analysis.

An additional ground wire must be connected to the PCB. The connector must be capable of receiving a wire size of 14 gauge. I prefer a PCB terminal block that we can screw clamp the wire into the PCB. The other end of the wire will connect to a large piece of metal on the bridge, or stuck into the actual earth like this:

We could choose not to ground the PCB, but this is not very smart. Autonomous vehicles (drones, robots) cannot be tethered by a ground wire. This makes grounding problematic. Since this PCB will be mounted somewhere on a bridge, we will take advantage of earth grounding.
For this assignment I need you to:

- Design a PCB for the entire circuit
- Create the requisitions to order all of the parts we need to build the entire circuit on the PCB, including connectors and batteries, and the board itself from a vendor.
- Document the process in a report, for future reference. The document should be self-sustaining.

Deliverables:

- Report (PDF or Word doc) – self sustaining document that includes
  - Written explanation of the design process
  - Explanation of the PCB vendor selection process
  - Pictures of the schematic and board files
  - A line item budget
  - Copies of the order forms
- Eagle Schematic Created in Eagle (*.sch)
- Eagle Board file (*.brd)
- Cam file required by the vendor (*.cam)
- Design Rules file from the Vendor (*.dru)
- Gerber files generated by processing the CAM job. (many files determined by the vendor)
- Requisition forms (1 form per vendor)

Due Date – This project is due Monday September 25 by midnight.
A heart pacemaker circuit is shown below. The SCR (silicon-controlled rectifier) is a device that has two modes of operation. When the voltage across the SCR is increasing but less than 5 volts, the SCR behaves as an open circuit. Once the voltage across the SCR reaches 5 volts, the device functions like a constant current source equal to 50μA. The voltage will begin to drop when the SCR produces current, but the behavior of the SCR will remain like a constant current source until the voltage drops to 0.2 volts. At this point the SCR will shut off and again become an open circuit.

\[
\begin{align*}
\text{SCR} & := \begin{cases} 
\text{OPEN CIRCUIT IF } v_c(t) < 5 \text{ V} \\
+ & 50 \mu \text{A CONSTANT CURRENT SOURCE UNTIL VOLTAGE LE } 0.2 \text{ V} 
\end{cases}
\end{align*}
\]

Assume at \( t=0 \), \( v_c(t) = 0 \) and the 1 μF capacitor begins to charge toward the 6 V source voltage. \( R = 570k\Omega \).
1. Find \( v_c(t) \) (by hand) and plot using Matlab for about 3-4 seconds. If you cannot solve for the this voltage, create the following waveform to use for the remainder of the project (over 3 seconds) (for minus points).

2. Now simulate external interference to the pacemaker by adding a voltage to \( v_c(t) \). Since the noise can be written as a sum of trig functions, it is called \textit{deterministic noise}.

\[
n1(t) = 0.5\cos(2\pi 60t) + 0.2\cos(2\pi 100,000t)
\]

\[
\xrightarrow{x(t)} \Sigma \xrightarrow{\text{Signal with added noise}} \xrightarrow{n(t)}
\]

3. Now also add \textit{random noise} to the signal. \textit{Write a Matlab function to do this.} Allow the user of the script to select the amplitude of the noise signal to be between 100 mV to 1 V peak to peak. The noise should not have any DC value (centered around 0V). Use the rand() function to create this noise signal and add it to the signals \( v_c(t) + n1(t) \)

\textbf{Noise removal}

Now that our pacemaker signal has been corrupted by all kinds of noise (deterministic and random) we need to create some algorithms to remove it. You should be learning about bode plots now. Refer to section 9-5 in the Ulaby text for reference.

4. Create a first order transfer function that has a breakpoint frequency well below \( \omega = 2\pi 60 \) so that the filter will pass the desired pacemaker signal and attenuate the deterministic noise.

\[
H(s) = \frac{1}{1 + sRC}
\]
5. Create the impulse response $h(t)$ for the filter created in part 4 and use the convolution function to simulate the process our corrupted signal through this filter. The signal flow graph representation is below:

$$x(t) + n(t) = x_{\text{noise}}(t)$$

Signal with noise \hspace{2cm} \text{Noise removal filter with an impulse response:} \hspace{2cm} y_1(t) = \text{conv}(h_1, x_{\text{noise}})$$

Desired signal

NOTE: The output of the conv() function has more points than original signal. So if $x(t)$ has 100 points and $h(t)$ has 5 points the output $y(t)$ will have $100+5-1 = 104$ points. I would suggest simply cropping off 4 points \textit{from the back end} of the output $y(t)$ so it has the same number of points as the original independent variable $t$.

6. Removing the random noise requires passing the signal through another filter, but this one is a little easier to create, with a little care. All we have to do is pass the corrupted signal through an $n$-point averaging filter. The filter has the form:

$$h_2(t) = \frac{1}{n} \text{ones}(1:n)$$

You choose $n$, the number of points. But be careful – too small and the noise won’t be removed very much. Too large and the desired signal will be removed. Test out a few values for $n$ and see what happens. \textit{Write a Matlab function to do this.} Allow the user to select $n$. Then use the convolution function to simulate passing the corrupted signal through the noise removal filter. A similar signal flow graph representation is below:

$$y_1(t)$$

Signal with noise \hspace{2cm} \text{n-point averaging filter} \hspace{2cm} y_2(t) = \text{conv}(h_2, y_1)$$

Desired signal
7. Once the signal has been passed through these two filters, it should look more like the original signal created in part 1. Analyze the signals and see how they compare.

Write a Matlab script to execute this project. Comment the code appropriately. Prepare a word document that communicates your attempt to complete this project, showing results (graphs) and analysis. Also upload the script and any functions you created. Prepare plots to insert into your word document that are clearly labeled and enhance your written text to communicate the work you did to complete this project.

Here is a block diagram summary of what you need to do:

\[ x(t) \xrightarrow{\Sigma} x_{\text{noise}}(t) \]

Signal with added random and Deterministic noise

\[ x_{\text{noise}}(t) \xrightarrow{\text{Noise removal filter with impulse response:}} y_1(t) = \text{conv}(h_1, x_{\text{noise}}) \]

Desired signal (deterministic noise removed)

\[ y_1(t) \xrightarrow{\text{n-point averaging filter (removes random noise)}} y_2(t) = \text{conv}(h_2, y_1) \]

Desired signal
The goal of this project is to simulate the circuit you built in the PCB project. You need to simulate the Wheatstone Bridge and the op-amp difference amplifier using BOTH Capture AND PSPICE AD.

Use the LM741 Op-Amp found in the “opamp” library. Below is what the circuit looks like as we left it in the PCB project. The difference amplifier has a gain of 5. \( V_{\text{OUT}} = \frac{R_4}{R_3} (V_3 - V_2) \). Where pin 6 of the op-amp \( V_{\text{OUT}} \). It turns out that a gain of 5 is not enough and we will fix that later.

1. Use PSPICE to model the Wheatstone Bridge portion of the circuit. Replace the weight sensor with a resistor (as you see above). To mimic the weight sensor, vary the resistor by simply changing \( R \) from 300-400 in increments of 25 and measuring \( V_{\text{out}} \). Show the results of the simulation in table form, but also include one output file and a picture of the circuit showing the DC node voltages. In reality, the weight sensor is 350 ohms under no load and the resistance drops when a load is applied. Its resistance will never go above 350 ohms. Let’s assume for us that the smallest value the resistor (weight sensor) will be is 300 ohms.
2. Use PSPICE to model the difference amplifier used on the PCB design project. As an input to the difference amplifier, use a triangle waveform that ranges from \(-V\) to \(V\) where \(V\) is the range of voltages discovered in #1. Attach the triangle waveform to the non inverting input of terminal, and ground the inverting input terminal. Plot the triangle wave and the output of the op-amp on the same plot.

3. Put the two circuits together and simulate the entire circuit using the Wheatstone bridge driving the op-amp circuit with a weight sensor resistance of 300 ohms. But now you will need to make some changes to the design, because it doesn’t work very well as it is. The goal of the op-amp circuit is to design it so that the output of the op-amp is close to \(+5\) V when the weight sensor is under maximum weight (300 ohms), and \(0\) V when the sensor has no weight bearing load applied (350 ohms). Even after improving the design, you won’t hit these output voltages exactly. So let’s create an acceptable range. Redesign the op-amp circuit to meet these conditions.

Under No load: \(|V_{OUT}| < |0.01| V\)
Under Maximum Load \(V_{OUT} > 4.6\ V\)

If you can come close to these values, this design will actually work quite well if you connect pin 6 of the op-amp to an analog pin on a microcontroller. We can program the microcontroller to interpret voltages from pin 6 to corresponding weights, time stamp and store the data for later. This is for work left for another course (EE3752, or EE 4820).

Deliverables:

A word document containing the following

- text (intro, procedure, analysis, conclusion)
- graphics (schematics with node voltages)
- plots
- output files