

ANALOG CIRCUITS FOR COMPUTING

by

Lauren Chun, Dillon Nguyen, Nicole Pickett

Senior Project

ELECTRICAL ENGINEERING DEPARTMENT

California Polytechnic State University

San Luis Obispo

June 2023

© 2023 Lauren Chun, Dillon Nguyen, Nicole Pickett

TABLE OF CONTENTS

LIST OF TABLES AND FIGURES.....	ii
ACKNOWLEDGEMENTS.....	v
ABSTRACT.....	1
I. INTRODUCTION.....	2
II. BACKGROUND.....	3
III. DESIGN REQUIREMENTS.....	5
Functional Decompositions.....	5
Customer Requirements.....	6
Engineering Specifications.....	7
IV. DESIGN.....	8
Component Selection.....	8
LTspice Circuit Schematics.....	9
Printed Circuit Board Designs.....	12
V. SIMULATION AND HARDWARE TESTS AND RESULTS.....	13
LTspice Design, Testing, and Results.....	13
Monte-Carlo Sensitivity Analysis on an Improved Differentiator Op-Amp Circuit.....	13
LTspice Design Simulations.....	16
Breadboard Testing and Results.....	27
Printed Circuit Board Tests and Results.....	30
Breadboard and PCB Testing Summary.....	34
Execution Time Testing.....	35
Calculation Accuracy.....	39
VI. CONCLUSION.....	39
VII. BIBLIOGRAPHY.....	40
IX. APPENDICES.....	42
A. Analysis of Senior Project Design.....	42
B. Bill of Materials and Time Schedule Allocation.....	50
C. IC Location Diagram.....	52
D. Printed Circuit Board Artwork.....	52
E. Program Listing - MATLAB® Simulation Code.....	54
F. Hardware Configuration/Layout.....	63

LIST OF TABLES AND FIGURES

Tables

Table 3-1. Analog Circuits for Computing Customer Requirements and Engineering Specifications.....	7
Table 3-2. Analog Circuits for Computing Engineering Specifications with Target, Tolerance, and Compliance Values.....	7
Table 5-1. Calculations for the Improved Differentiator Op-Amp Circuit Input and Feedback Resistances.....	13
Table 5-2. Tabulated Breadboard Data for the Voltage Adder Circuit Test Cases.....	26
Table 5-3. Tabulated Breadboard Data for the Non-Inverting Voltage Multiplier Circuit Test Cases.....	28
Table 5-4. Tabulated Breadboard Data for the Resistive Divider Circuit Test Cases.....	29
Table 5-5. Tabulated Breadboard Data for the Differentiator Circuit Test Cases.....	29
Table 5-6. Tabulated Breadboard Data for the Integrator Circuit Test Cases.....	30
Table 5-7. Tabulated PCB Data for the Voltage Adder Circuit Test Cases.....	31
Table 5-8. Tabulated PCB Data for the Non-Inverting Voltage Multiplier Circuit Test Cases.....	32
Table 5-9. Tabulated PCB Data for the Resistive Divider Circuit Test Cases.....	33
Table 5-10. Tabulated PCB Data for the Differentiator Circuit Test Cases.....	33
Table 5-11. Tabulated PCB Data for the Integrator Circuit Test Cases.....	34
Table 5-12. MATLAB® Function Execution Time Results.....	37
Table A-1. Initial Cost Estimates for the Analog Circuits for Computing Project.....	44
Table B-1. Analog Circuits for Computing Project Bill of Materials.....	50
Table B-2. Timeline of Tasks and Milestones for the Analog Circuits for Computing Project.....	51

Figures

Figure 2-1. Conceptual Basic Arithmetic Circuits.....	3
Figure 2-2. Conceptual Basic Calculus Circuits.....	3
Figure 2-3. “The Analog Thing” Device by Anabrid.....	4
Figure 3-1. Level 0 Functional Decomposition Block Diagram.....	5
Figure 3-2. Level 1 Functional Decomposition Block Diagram.....	5
Figure 3-3. Level 2 Functional Decomposition Block Diagram.....	6
Figure 4-1. LTspice-Designed Voltage Adder Circuit Schematic.....	10
Figure 4-2. LTspice-Designed Non-Inverting Voltage Multiplier Circuit Schematic.....	10
Figure 4-3. LTspice-Designed Resistive Divider Circuit Schematic.....	11
Figure 4-4. LTspice-Designed Differentiator Circuit Schematic.....	11

Figure 4-5. LTspice-Designed Integrator Circuit Schematic.....	12
Figure 4-6. PCB Layout Differentiator and 2nd-Order Differential Equation.....	12
Figure 5-1. Block Diagram for Simulation and Hardware Testing.....	13
Figure 5-2. Differentiator Circuit with a Monte-Carlo Analysis Setup.....	14
Figure 5-3. Improved Differentiator Op-Amp Circuit Performance with Zero Tolerance at $T = 25^{\circ}\text{C}$	14
Figure 5-4. Gain Response of the Monte-Carlo Analysis of the Differentiator Circuit.....	15
Figure 5-5. Histogram of the Gain for the Improved Differentiator Op-Amp Circuit with Respect to Varying Temperatures from -25°C to 85°C in Intervals of 10°C	15
Figure 5-6. Histogram of the Frequency at 0dB for the Improved Differentiator Op-Amp Circuit with Respect to Varying Temperatures from -25°C to 85°C in Intervals of 10°C	16
Figure 5-7. Histogram of the Frequency at Maximum Gain for the Improved Differentiator Op-Amp Circuit with Respect to Varying Temperatures from -25°C to 85°C in Intervals of 10°C	16
Figure 5-8. LTspice Voltage Adder Simulation (Addition Test).....	17
Figure 5-9. LTspice Voltage Adder Simulation (Subtraction Test).....	18
Figure 5-10. LTspice Non-Inverting Multiplier Simulation (Gain of 2.2).....	18
Figure 5-11. LTspice Non-Inverting Multiplier Simulation (Gain of 6.45).....	19
Figure 5-12. LTspice Non-Inverting Multiplier Simulation (Gain of 9.3).....	19
Figure 5-13. LTspice Resistive Divider Simulation for DC Waveforms.....	20
Figure 5-14. LTspice Resistive Divider Simulation for Sinusoidal Waveforms.....	20
Figure 5-15. LTspice Resistive Divider Simulation for Square Waveforms.....	21
Figure 5-16. LTspice Resistive Divider Simulation for Triangle Waveforms.....	21
Figure 5-17. LTspice Differentiator Simulation for Sinusoidal Waveforms.....	22
Figure 5-18. LTspice Differentiator Simulation for Square Waveforms.....	23
Figure 5-19. LTspice Differentiator Simulation for Triangle Waveforms.....	23
Figure 5-20. LTspice Integrator Simulation for Sinusoidal Waveforms.....	24
Figure 5-21. LTspice Integrator Simulation for Square Waveforms.....	25
Figure 5-22. LTspice Integrator Simulation for Triangle Waveforms.....	25
Figure 5-23. Graphical Representation of the Breadboard Data Accuracy versus the PCB Data Accuracy.....	35
Figure 5-24. Oscilloscope Trace of the Execution Time for Test Equation, $y'' - 10y' - y$	36
Figure 5-25. Graphical Representation of the MATLAB® Execution Time and PCB Execution Time (Basic Arithmetic Circuits).....	38
Figure 5-26. Graphical Representation of the MATLAB® Execution Time and PCB Execution Time (Basic Calculus Circuits).....	38

Figure A-1. Gantt Chart for Analog Circuits for Computing Project (Project Planning Phase).....	45
Figure A-2. Gantt Chart for the Analog Circuits for Computing Project (Conceptual Design and Simulation Phase).....	45
Figure A-3. Gantt Chart for the Analog Circuits for Computing Project (Manufacturing and Testing Phase)..	46
Figure C-1. LM741 Pinout Diagram.....	52
Figure D-1. Voltage Adder PCB.....	52
Figure D-2. Non-Inverting Voltage Multiplier PCB.....	52
Figure D-3. Inverting Voltage Multiplier PCB.....	53
Figure D-4. Resistive Divider PCB.....	53
Figure D-5. Differentiator PCB.....	53
Figure D-6. Integrator PCB.....	53
Figure F-1. Hardware Layout for the Voltage Adder (Designed for a Gain of 1).....	63
Figure F-2. Hardware Layout for the Non-Inverting Multiplier (Designed for a Gain of 2.2).....	63
Figure F-3. Hardware Layout for the Resistive Divider.....	64
Figure F-4. Hardware Layout for the Differentiator.....	64
Figure F-5. Hardware Layout for the Integrator.....	65
Figure F-6. Hardware Layout for the Tested Second-Order Differential Equation.....	65

ACKNOWLEDGEMENTS

We would like to thank our advisor, Dr. Jason Poon, as well as Cal Poly EE Department faculty members Dr. Vladimir Prodanov, Dr. Dale Dolan, and Dr. Taufik for providing guidance and direction during the project development process. We would also like to thank the Cal Poly EE Department for providing the resources to perform extensive research and testing for the project.

ABSTRACT

This project entails designing, simulating, and verifying analog circuits that can perform essential computing functions for power systems applications. The project aims to remedy critical challenges associated with handling calculations digitally, namely, time and power. This project's scope includes creating a library of circuits in SPICE that can be used to model and simulate complex mathematical equations. From these SPICE models, the circuit can be constructed physically, where the solution can be generated in less time using less power than doing the computation digitally. The performance and efficiency of analog computing will be measured and compared to conventional digital methods.

I. INTRODUCTION

The basis of this project started in the 1960s as analog computers were first used to calculate complex algorithms. Over time, digital computation became more favorable due to its accuracy with less noise, distortion, and interference among its results [1]. Analog circuitry, however, is re-emerging as a solution for complex calculations. It can present more refined results while consuming less power and bandwidth, suggesting that analog computing can provide the same results as digital systems but more efficiently [2, 3].

With analog computing, physical circuits can be constructed to perform various mathematical operations; signals are sent through the circuits to generate results while measuring the speed and power consumption [5]. The goal is to create a collection of circuits that handle simple arithmetic, like addition, to complex math, like nonlinear differential equations [1]. The resulting analog computation can then be transmitted to software where it can be analyzed and applied to applications, particularly power grid emulation and testing.

II. BACKGROUND

Analog computation is defined as computations in which the result is derived via measurement from a continuous quality, such as voltage or current [1]. From this definition, analog computation is favored by continuous-time systems, which has been seen throughout history. Analog computation in the 1940s generated many special-purpose machines that handled continuous signals, such as harmonic and power-network analyzers [1]. The one thing in common with all of these devices is this: they all handle data in real time rather than using step functions to approximate the continuous data [1]. This makes analog computation a preferred choice for specific applications, such as power grid simulations or signal processing.

Analog computation ranges from basic to complex mathematical equations, all done with passive and active circuit configurations [1]. The basic arithmetic handled, such as those seen in Figure 2-1, uses passive circuits, such as a voltage divider, while active circuits handle addition, subtraction, and multiplication.

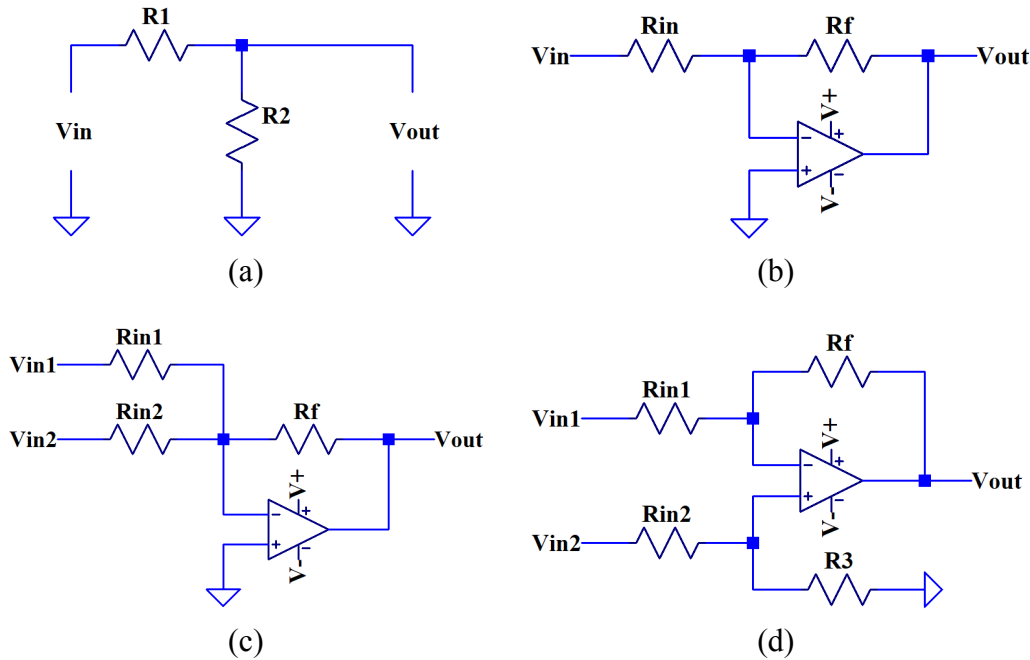


Figure 2-1. Conceptual Basic Arithmetic Circuits (a) Resistive Divider (b) Inverting Multiplier (c) Voltage Adder (d) Voltage Subtractor (Differential Amplifier)

In combination with calculus, these circuits can be used to represent complex systems such as linear differential equations such as the Mathieu, Bessel, and Legendre functions [2]. Figure 2-2 shows an integrator and differentiator. What is important to note about the differentiator is that the capacitor input configuration permits too much noise, the second configuration mitigates this issue.

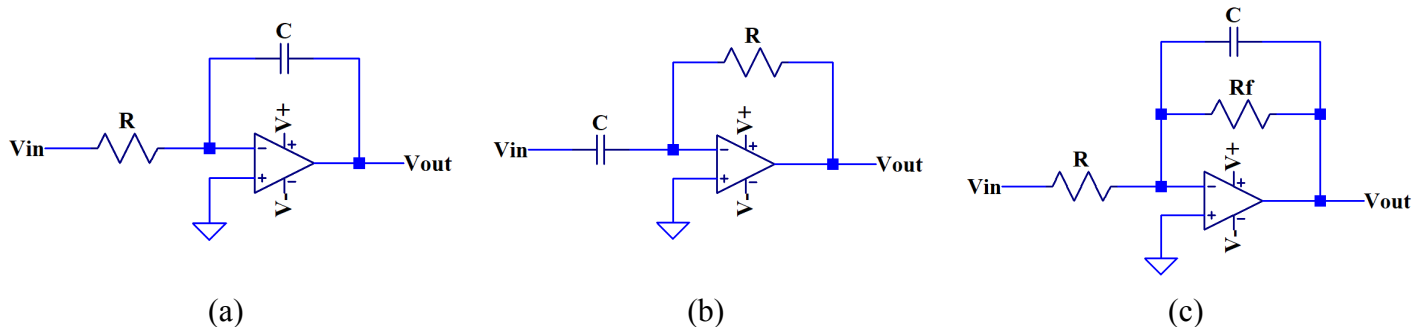


Figure 2-2. Conceptual Basic Calculus Circuits (a) Integrator (b) Differentiator (c) Noise-Reducing Differentiator

These circuits allow for complex analysis of larger-scale systems and provide high-speed solutions for systems such as power grids [3]. Traditional power grid simulations rely on numerical algorithms, which need to be faster to handle a real-time power system. Analog computation allows simulators like these to surpass the limits of numerical (discrete) algorithms as they use the measurement of power to extract data such as damping power and phase angle [3]. Circuits such as the integrator and differentiator are represented via a control system and then constructed accordingly to analyze the power grid. This results in simplification with simulation models.

Analog computation is also being furthered as it is used for power grid emulation. The difference is that analog microelectronics are used to handle these emulators [4]. In this case, it is being explored to create better emulations that will handle DC and AC emulations better. By using power as the continuous signal as the output, these emulators utilize CMOS microelectronic ASIC boards to simulate not only the power grid but each load attached to the grid as well [4]. This yields a much more comprehensive simulation, benefiting those who are optimizing and/or repairing the power grid.

The product titled "The Analog Thing" by the company Anabrid is another example of analog computation that is further expanded upon, shown in Figure 2-3. It is a low-cost, open-source analog computer with multiple inputs and connection ports to calculate mathematical functions [5]. This multipurpose analog computer can be used for applications from education to providing control to circuitry [5]. The computational analog circuits utilize the basic and calculus circuitry presented above, allowing this device to be multifaceted and provide solutions to specific problems.



Figure 2-3. "The Analog Thing" Device by Anabrid [5]

III. DESIGN REQUIREMENTS

Functional Decompositions

The level 0 functional decomposition displays the overall inputs and outputs of a system. For this project, the main input for the analog circuits system is the desired algorithm to be solved, supplied by the customer. The main output of the system is the PCB file representation of the desired algorithm so that the customer can manufacture their PCB-represented algorithm. Figure 3-1 illustrates the level 0 functional decomposition for this project.

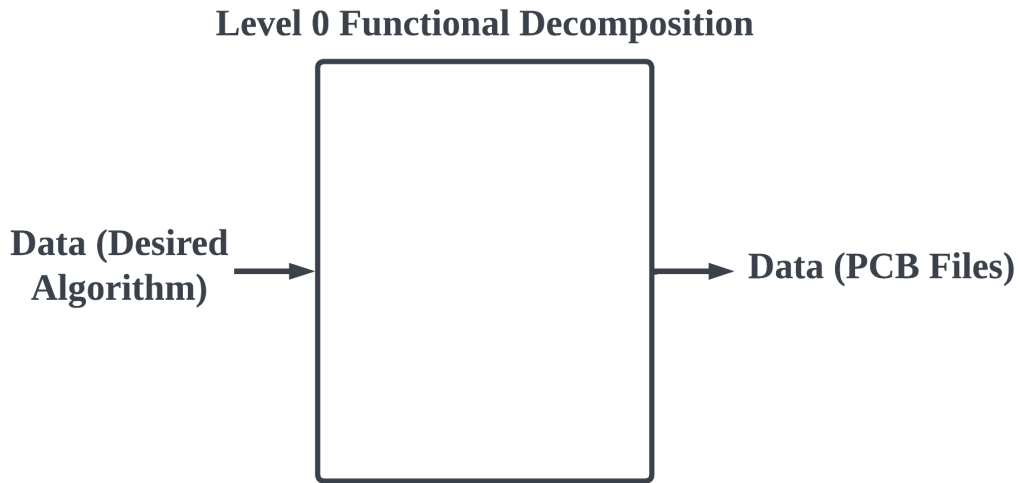


Figure 3-1. Level 0 Functional Decomposition Block Diagram

The level 1 functional decomposition elaborates on the level 0 functional decomposition, showing the internal modules within the main system as well as secondary inputs and outputs. For the analog circuits system, the internal modules include the decomposer, library of computational circuits, netlist synthesizer, and PCB layout generator. The decomposer, netlist synthesizer, and PCB layout generator modules must be done manually by the customer. The decomposer module entails the customer deconstructing their algorithm into smaller arithmetic and calculus functions that can be used to determine which circuits from the provided library are needed to solve the algorithm. The netlist synthesizer allows for open-source modeling and simulation of the customer's algorithm, providing a SPICE netlist for their algorithm. The PCB layout generator module entails customers transferring their modeled algorithm to a PCB layout format for future printing and solving via hardware testing and measurement. The library of computational circuits will be provided for the customer, which they can pull from to model their desired algorithm. Figure 3-2 provides a visual of this project's level 1 functional decomposition.

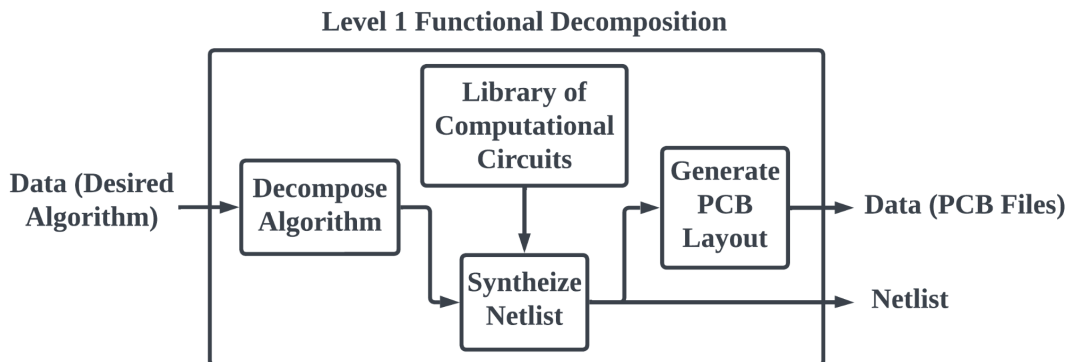


Figure 3-2. Level 1 Functional Decomposition Block Diagram

The level 2 functional decomposition elaborates on specific modules within the overall system specified in the level 1 functional decomposition. The decomposer, netlist synthesizer, and PCB layout generator are performed manually by the customer. Thus, the library of computational circuits is the only module that requires elaboration. The library of computational circuits comprises the following circuits: voltage adder, non-inverting multiplier, resistive divider, differentiator, integrator, and inverter. The library circuits selected by the customer will then be outputted as SPICE netlists to incorporate into their full algorithm model. The visual for the level 2 functional decomposition is shown in Figure 3-3.

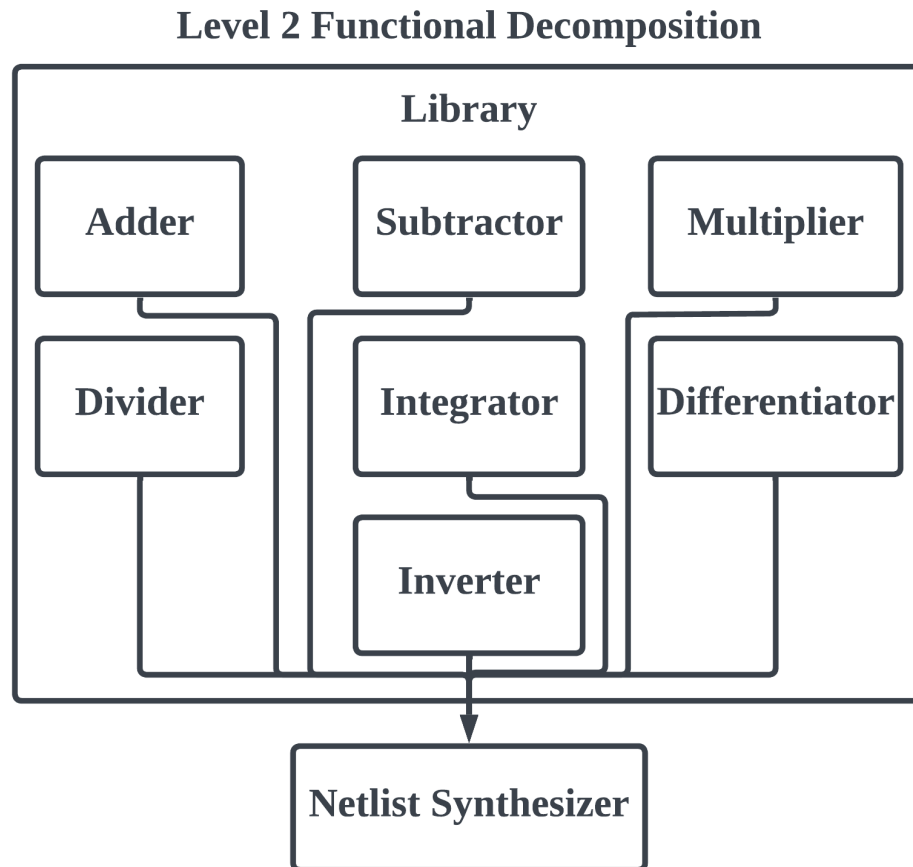


Figure 3-3. Level 2 Functional Decomposition Block Diagram

Customer Requirements

The customer requirements for this project are selected to allow for the most inclusivity possible while ensuring the computational circuits system is highly and properly functional as well as intuitive. The following are the selected customer requirements, detailing the constraints for the project tailored to the customer:

- a) Open-source simulation and hardware files
- b) Functions must include a variety of mathematical operations (e.g., arithmetic, calculus, differential equations)
- c) Library must be intuitive to use
- d) Circuits must provide results in a short amount of time for power grids
- e) Mathematical computations from circuits must be accurate

Table 3-1 elaborates on the chosen customer requirements as well as the engineering specification derived for each requirement.

Table 3-1. Analog Circuits for Computing Customer Requirements and Engineering Specifications.

Customer Requirements	Engineering Specification	Justification
a	Zero closed-source applications	An open-source library will ensure it is accessible to anyone without needing to pay for access and use [6].
b	A minimum of 5 mathematical operations completed	The core function of the computational analog circuits. Arithmetic and basic calculus will serve as the foundation for nonlinear computations and systems-of-equation solving. [2]
c	At least 2 hours to become familiar with the interface and resources	An intuitive interface will allow easy use of the analog circuits for the user to carry out their desired task. [7]
d	Computations will take equivocal or less time to digital systems (in msec)	A shortened time for computations will allow for improvement in power grid development, simulation, and maintenance for the customer. [4]
e	Computations are accurate with a maximum of 10% error	A maximum of 10% error will ensure the validity of the computations and that it will function correctly in any application. [1]

Engineering Specifications

Derived from the selected customer requirements, the engineering specifications detail the numerical target and tolerance values as well as risk and compliance assigned to each requirement. Each of the specifications is designed with the customer's needs and wants prioritized. Table 3-2 elaborates on these engineering specifications below.

Table 3-2. Analog Circuits for Computing Engineering Specifications with Target, Tolerance, and Compliance Values

	Parameter	Target	Tolerance	Risk (H,M,L)	Compliance (A,T,S,I)
1	Accuracy	10 percent error	Max	M	A, T
2	Number of mathematical operations	5 operations	Min	M	I
3	Closed-Source	0	Max	L	T
4	Intuitive	2 hrs	Min	M	I
5	Short Computation Time	<= to digital in msec	Max	H	A, T, S

IV. DESIGN

Component Selection

Individual passive component values differ between the various circuits, but the LM741 operational amplifier is the fundamental component for most of the mathematical operations. As the circuits are computational circuits, most require the use of some kind of active component. The LM741 was chosen due to its fairly large range of operational voltages ($\pm 18\text{V}$), commonality in the market, and low-cost use.

The circuit that is the exception to using the LM741 op-amp is the resistive divider circuit, which consists of only one potentiometer. The potentiometer selected for this circuit as well as other circuits that required variable resistance (e.g. integrator and non-inverting multiplier), is the 251B12T104A2NB 100k Ω -knob potentiometer. This selection is because 100k Ω provides the best ohmic range for the specifications of each circuit while allowing easy adjustment, unlike other potentiometers that use flat turning mechanisms.

All other passive component values, such as single resistors and capacitors, depend on each circuit and the tolerances specified. The process for selecting each of these components is discussed in the following sections.

Voltage Adder Component Selection

The voltage adder circuit uses an arbitrary number of components that depends on the desired number of summing input nodes. Each voltage adder has at least three resistors, two for the two summing input nodes and one for the feedback resistor. For testing purposes, a minimum of three resistors are used with the system designed for a gain of 1. The following equation describes the gain for the voltage adder used to determine the summing input node and feedback resistor values:

$$\frac{-V_{out}}{(V_{in,1} + V_{in,2} + V_{in,3} + \dots + V_{in,n})} = \frac{R_F}{R_{in}}$$

To ensure a gain of 1, the summing input and feedback resistors are all 10 k Ω . The input signals are arbitrary and do not affect the operation of the circuit as long as their sum does not exceed the $\pm 18\text{V}$ rails of the LM741 operational amplifier.

Non-Inverting Multiplier Component Selection

The non-inverting multiplier circuit uses two resistors to form an arbitrary multiplication constant represented by the gain of the LM741 op-amp. The gain of the amplifier is determined using the following equation:

$$\frac{V_{out}}{V_{in}} = 1 + \frac{R_2}{R_1}$$

The gain is dependent on the desired multiplication constant. For testing purposes, an arbitrary gain of 10 was selected; thus, the resistor values chosen are 100 k Ω and 10 k Ω .

Resistive Divider Component Selection

The components of the resistive divider circuit are two resistors. The two resistors are arbitrary and are dictated based on the following equation:

$$V_{out} = V_{in} \left(\frac{R_2}{R_1 + R_2} \right)$$

Depending on the desired divisor for the division operation, the resistors will be changed accordingly. For purposes of simulation, resistor values of 1 k Ω for both resistors were chosen to perform division with a divisor of 2.

Differentiator Component Selection

The differentiator circuit only uses a combination of two capacitors and two resistors. The component values are chosen based on the time constant equation shown below that dictates the charging characteristics.

$$\tau = RC$$

To determine the output voltage of the circuit, the following equation is used:

$$V_{out} = -R_F C \left(\frac{dV_{in}}{dt} \right)$$

V_{in} is arbitrary and dependent on the desired input signal. The component values and time constant vary depending on the type of input signal (i.e., sine wave, square wave, triangle wave, etc.) as well. For testing purposes, resistor values of 1.5 k Ω and 15 k Ω are used, and capacitor values of 1 pF and 10nF. The traditional configuration only uses a set of one resistor and capacitor (as represented in the equation). To mitigate gain increase and account for attenuation at higher frequencies, input resistor R_{in} and feedback capacitor C_F are added to the design.

Integrator Component Selection

The integrator circuit design uses four resistors and one capacitor. The input resistor, feedback resistor, and feedback capacitor are determined by the equation:

$$V_{out} = -\frac{1}{R_{in} C} \int_0^t V_{in} dt$$

The additional two resistors are used as a resistive divider at the end of the integrator circuit as maintaining a gain of 1 became an issue during the design phase. The resistive divider allows for scaling of the output to obtain the true, desired integrated output. After trial and error, an input resistor value of 1 k Ω , feedback resistor value of 470 k Ω , and feedback capacitance value of 10 nF were chosen.

LTspice Circuit Schematics

Voltage Adder Design

The voltage adder circuit is designed based on the traditional voltage adder circuit configuration using a standard LM741 op-amp. The voltage adder circuit allows for as many summing nodes as desired. For the project's purposes, the number of summing nodes will not exceed four nodes. This constraint is not based on the design's operation, but rather it is not cost-effective and becomes redundant to add more than four summing nodes. The traditional configuration requires the input to be fed into the inverting terminal of the op-amp, resulting in inverted results. Therefore, when the output is probed and results are recorded for this circuit, the output will always be the inverted value of the real answer. This design can also be configured for subtraction and negative numbers by applying negative voltages to one or more of the summing inputs. The supply voltages are set to $\pm 18V$ to minimize clipping at the output; these supply voltage levels are maintained for all designed

active circuits. Figure 4-1 shows the finalized voltage adder circuit as simulated in LTspice, configured for only two summing nodes.

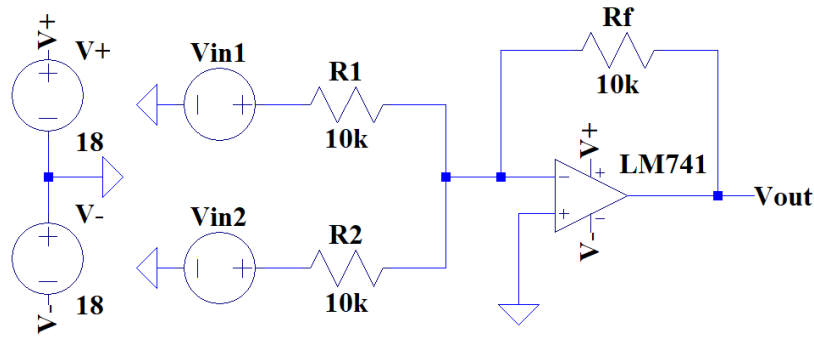


Figure 4-1. LTspice-Designed Voltage Adder Circuit Schematic

Non-Inverting Voltage Multiplier Design

The non-inverting multiplier circuit is designed based on the traditional non-inverting multiplier circuit configuration using a standard LM741 op-amp. The non-inverting multiplier circuit allows for multiplication between a constant (set by the gain of the op-amp) and an input signal. Multiplication can be executed using the inverting multiplier configuration and non-inverting multiplier configuration. For the project's purposes, the non-inverting configuration provided the desired results without the need to invert the measured output to obtain the true result. This design can also be configured for negative multiplication by applying negative voltages to one or more of the summing inputs. Figure 4-2 shows the finalized non-inverting multiplier circuit as simulated in LTspice, configured for a multiplication constant (gain) of 10.

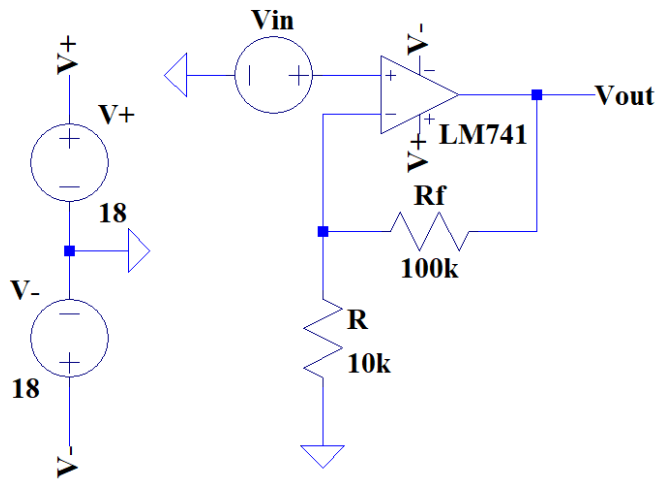


Figure 4-2. LTspice-Designed Voltage Multiplier Circuit Schematic

Resistive Divider Design

The resistive divider circuit is designed based on the resistive divider circuit configuration using standard resistors. The resistive divider circuit allows for division between a constant set by the two resistors and an input signal. For future purposes, a single 100 kΩ potentiometer can be used in place for the two resistors set up. Figure 4-3 shows the finalized resistive divider circuit as simulated in LTspice, configured for a divisor of 2.

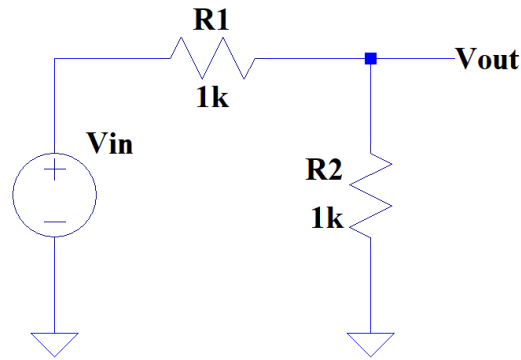


Figure 4-3. LTspice-Designed Resistive Divider Circuit Schematic

Differentiator Design

The differentiator circuit is designed based on the traditional differentiator circuit configuration using a standard LM741 op-amp with an added input resistor and feedback capacitor. The addition of the extra resistor and capacitor is to mitigate gain increase and account for attenuation at higher frequencies. The traditional configuration requires the input to be fed into the inverting terminal of the op-amp, resulting in inverted results. Therefore, when the output is probed, and results are recorded for this circuit, the output will always be the inverted value of the real answer. This design can also be configured for negative signals by applying negative voltages to the input. Figure 4-4 shows the finalized differentiator circuit as simulated in LTspice, configured for only sinusoids.

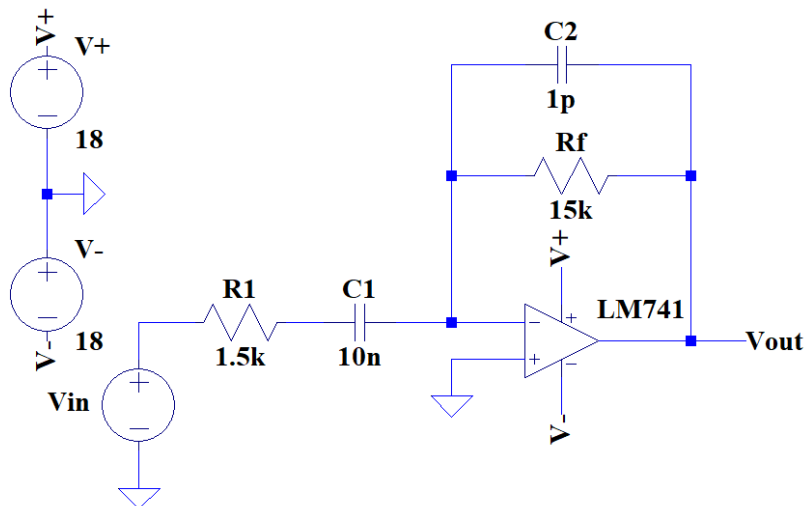


Figure 4-4. LTspice-Designed Differentiator Circuit Schematic

Integrator Design

The integrator circuit is designed based on the traditional integrator circuit configuration using a standard LM741 op-amp. An additional resistive divider combination is added at the end of the integrator system to maintain a gain of 1. The traditional configuration requires the input to be fed into the inverting terminal of the op-amp, resulting in inverted results. Therefore, when the output is probed, and results are recorded for this circuit, the output will always be the inverted value of the real answer. This design can also be configured for negative signals by applying negative voltages to one or more of the summing inputs. Figure 4-5 shows the finalized integrator circuit as simulated in LTspice.

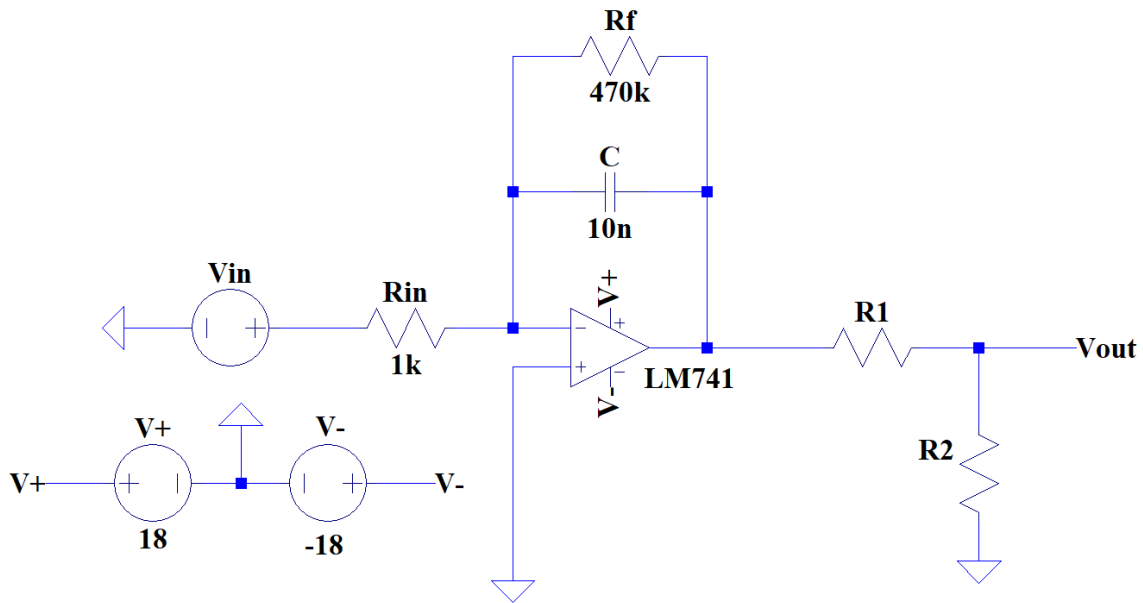
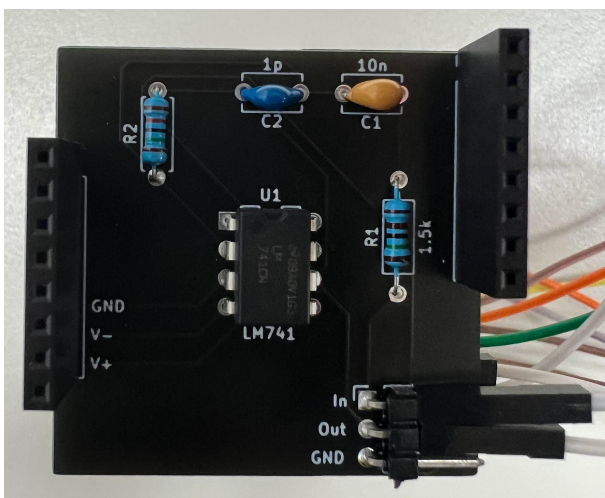


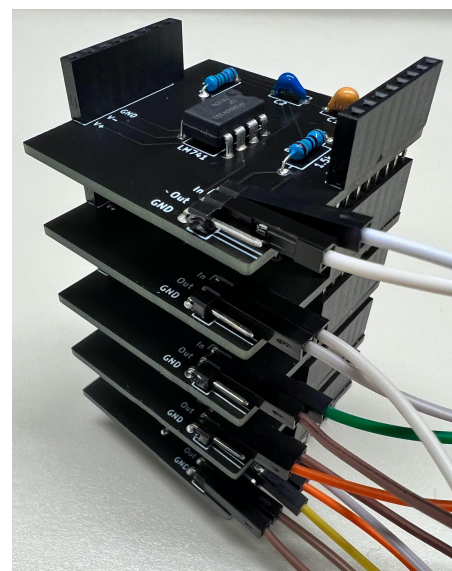
Figure 4-5. LTspice-Designed Integrator Circuit Schematic

Printed Circuit Board Designs

The printed circuit boards (PCBs) for the individual computational circuits were designed in the PCB software KiCad. The PCBs are designed to represent specific mathematical operations such as addition, multiplication, division, differentiation, and integration. The dimensions of each PCB are 19.05 mm x 19.05 mm, with the trace width chosen to be 0.25mm due to each PCB only needing low power and low frequency. In addition, header pins were placed on two sides of the board in order to make the PCBs stackable. This allows all of the positive and negative power supplies, along with the grounds, to be easily connected without additional wiring. With only one math operation on each PCB, it allows the customer to use the same set of PCBs to model various equations instead of needing to order a new PCB to represent a singular equation. Figures 4-6 a and b show an example of the manufactured differentiation PCB and how each of the individual PCBs cascade and stack together to form a more complex algorithm.



(a)



(b)

Figure 4-6. (a) Manufactured Differentiator PCB (b) Three Differentiators, One Inverting Multiplier (Gain of 10), and One Voltage Adder Manufactured and Cascaded to form a 2nd-Order Differential Equation

V. SIMULATION AND HARDWARE TESTS AND RESULTS

The project required both simulation and hardware testing as each circuit was designed via simulation first and then tested on breadboards. After confirming that each circuit fully operates, given the specified constraints and tolerances, the PCBs were designed and tested in the same manner. The testing setup used the same equipment in each setting: The system is powered by a function generator and DC power supply, where the outputs are observed using an oscilloscope. Figure 5-1 shows the block diagram that visualizes this process for all simulation and hardware testing.

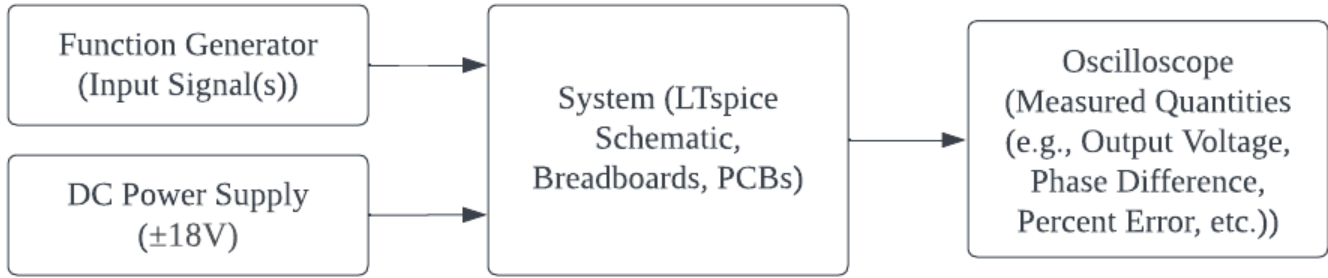


Figure 5-1. Block Diagram for Simulation and Hardware Testing

LTspice Design, Testing, and Results

Monte-Carlo Sensitivity Analysis on an Improved Differentiator Op-Amp Circuit

For LTSpice design and testing, each circuit was designed in LTSpice and then simulated. In addition, a Monte-Carlo analysis was performed to observe the effects of 10% on the entire system. Figure 5-2 shows a differentiator with an additional capacitor [1] to reduce noise at the sacrifice of bandwidth. This circuit is specifically designed for a bandwidth of 9kHz (from 1kHz and 10kHz). The following design equations are used where C_{in} was chosen as 10nF, and V_{in} is a 2kHz ramp function with an amplitude of 1V:

$$V_{out} = -R_F C_{in} \frac{dV_{in}}{dt} \quad (\text{Equation 1: Differentiator Gain})$$

Table 5-1: Calculations for the Improved Differentiator Op-Amp Circuit Input and Feedback Resistances

Feedback Resistance Calculation	Input Resistance Calculation
$f_L = \frac{1}{2\pi R_F C_{in}} \text{ (Equation 2: Lower Cutoff Frequency)}$ $R_F = \frac{1}{2\pi f_L C_1} = \frac{1}{2\pi * 1kHz * 10nF} = 15.9k\Omega$ $R_F = 15.9k\Omega \approx 16k\Omega \text{ (Standard Resistor Value)}$	$f_U = \frac{1}{2\pi R_{in} C_{in}} \text{ (Equation 3: Upper Cutoff Frequency)}$ $R_{in} = \frac{1}{2\pi f_U C_1} = \frac{1}{2\pi * 10kHz * 10nF} = 1.59k\Omega$ $R_{in} = 1.59k\Omega \approx 1.6k\Omega \text{ (Standard Resistor Value)}$

Assuming $R_{in} C_{in} = R_F C_F$, then C_F can be solved

$$C_F = \frac{R_{in} C_{in}}{R_F} = \frac{1.6k * 10nF}{16k} = 1nF$$

Due to the negative sign in Equation 1, an inverting amplifier with a gain of one has been cascaded to invert the output again to yield the positive derivative of the signal.

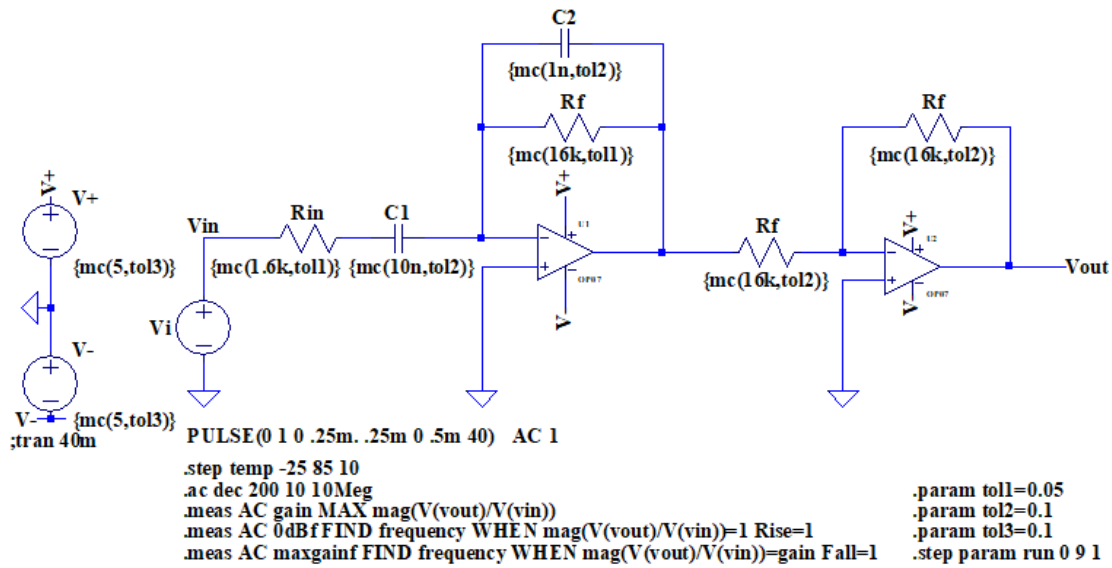


Figure 5-2. Differentiator Circuit with a Monte-Carlo Analysis Setup [8]

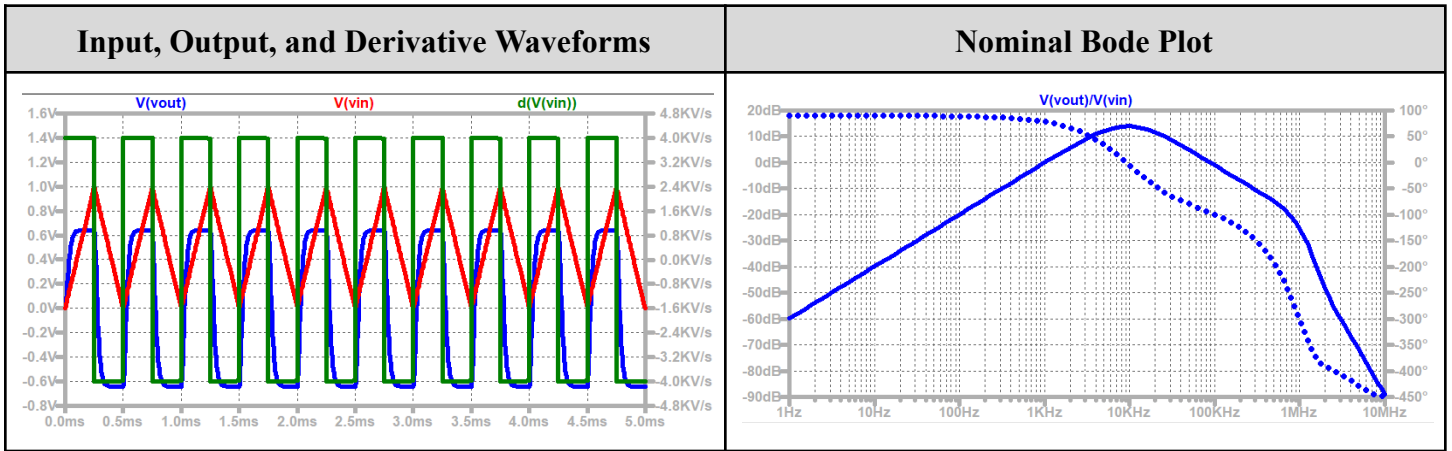


Figure 5-3. Improved Differentiator Op-Amp Circuit Performance with Zero Tolerance at T = 25°C

The plot above shows the input and output waveform of the differentiator. The input is a 1V 2kHz ramp function. Using cursor measurements, the output waveform amplitude is 640mV on the positive cycle. Equation 1 is used to verify the output waveform and functionality of the circuit. The derivative of the original waveform is measured to be 4kV/s (dark green).

$$V_{out} = -1 * R_F C_{in} \frac{dV_{in}}{dt} \rightarrow 640mV = 16k\Omega * 10nF * \frac{dV_{in}}{dt}$$

$$640mV = 1.6 * 10^{-4} * \frac{dV_{in}}{dt}$$

$$\frac{dV_{in}}{dt} = \frac{4kV}{s}$$

This equals the actual derivative of the input waveform, verifying the functionality of the circuit. This shows that the parameters of all passive components will be tested from a -10% to +10% tolerance. The simulation is run in AC mode to observe the effects of the gain response of the system. As shown, the effects of the tolerance correlate to the accuracy of the system as it shows a maximum of 10% at the output. Since each component tolerance varies between -10% to +10%, it is important to observe this behavior in order to characterize the impact it will have on the output.

Figure 5-4 exhibits the improved differentiator op-amp circuit's maximum magnitude response for temperatures from -25°C to 85°C. With this variation in temperature, the maximum gain has a range of

approximately 12.27 dB to 15.74 dB. The nominal operating temperature range of this circuit is -25°C to 125°C, which is proven as the maximum gain (peak magnitude response) of this circuit at approximately in the middle of this temperature range.

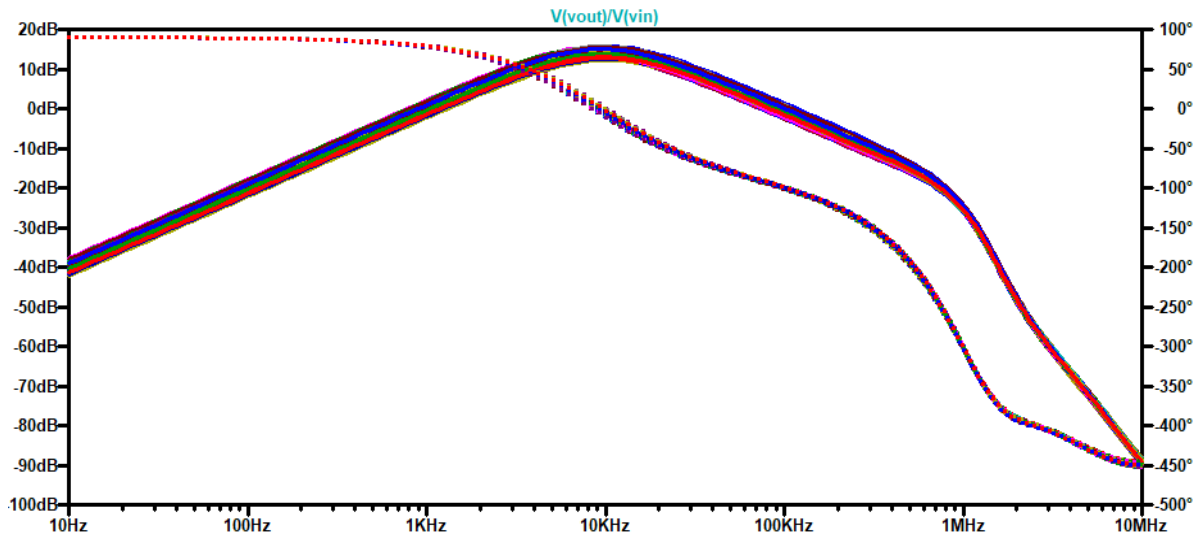


Figure 5-4. Gain Response of the Monte-Carlo Analysis of the Differentiator Circuit.

Figure 5-5 reflects the distribution for the improved differentiator op-amp circuit’s maximum gain with respect to temperatures from -25°C to 85°C. The largest bin in the histogram is a maximum gain between 13.77dB and 14.27dB with a count of 35 simulation steps from a total of 120 simulation steps. This indicates that the most common maximum gain from the circuit is within this dB range under temperature changes.

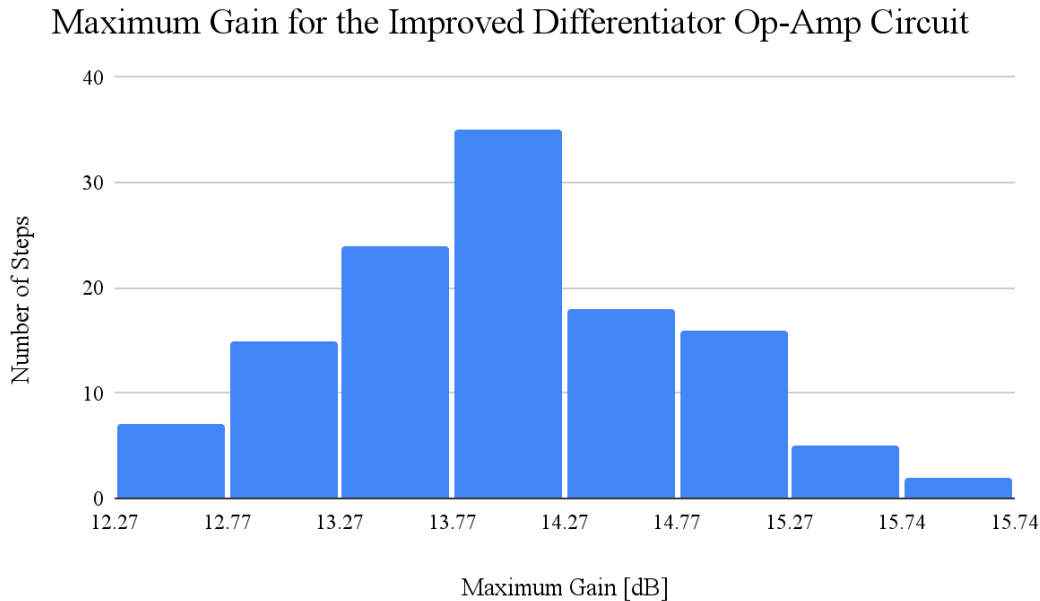


Figure 5-5. Histogram of the Gain for the Improved Differentiator Op-Amp Circuit with Respect to Varying Temperatures from -25°C to 85°C in Intervals of 10°C

Figure 5-6 reflects the distribution for the improved differentiator op-amp circuit’s frequency at 0 dB gain with respect to temperatures from -25°C to 85°C. The largest bins in the histogram are frequency ranges 930 Hz to 990 Hz and 990 Hz to 1050 Hz with counts of 30 simulation steps each from a total of 120 simulation steps. This indicates that the most common frequency values from the circuit are within these frequency ranges under temperature changes.

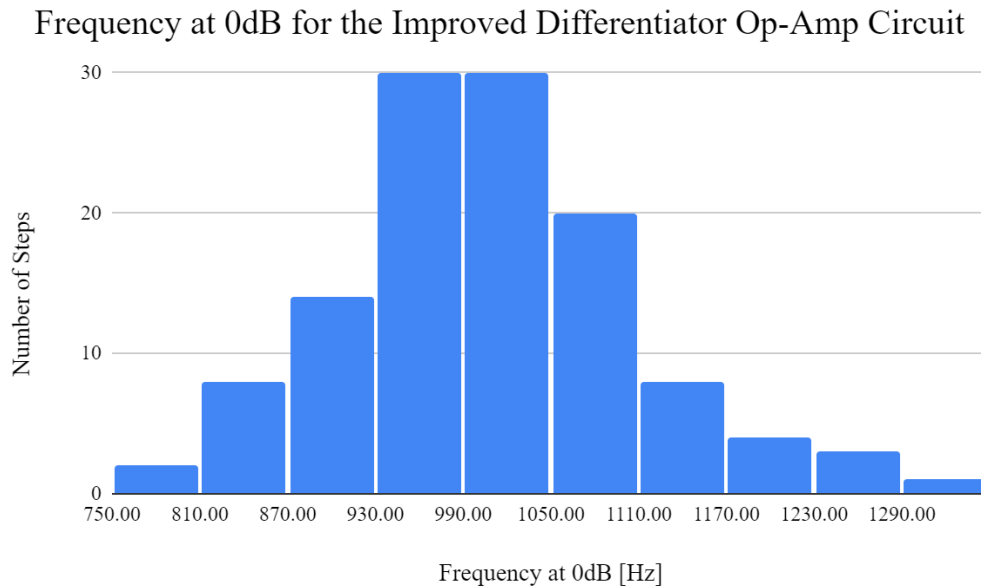


Figure 5-6. Histogram of the Frequency at 0dB for the Improved Differentiator Op-Amp Circuit with Respect to Varying Temperatures from -25°C to 85°C in Intervals of 10°C

Figure 5-7 reflects the distribution for the improved differentiator op-amp circuit's frequency at 0dB gain with respect to temperatures from -25°C to 85°C. The largest bins in the histogram are frequency ranges 9.625kHz to 9.9kHz with a count of 32 simulation steps from a total of 120 simulation steps. This indicates that the most common frequency values from the circuit are within this Hz range under temperature changes.

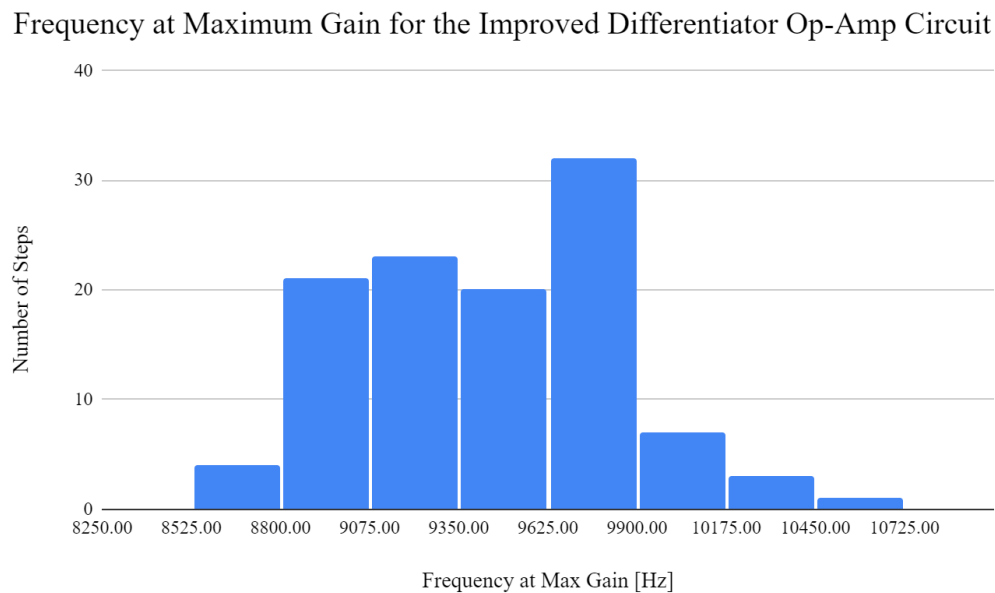


Figure 5-7. Histogram of the Frequency at Maximum Gain for the Improved Differentiator Op-Amp Circuit with Respect to Varying Temperatures from -25°C to 85°C in Intervals of 10°C

LTspice Design Simulations

Each of the designed computational circuits is tested with at least one type of input waveform (DC, sine, square, and/or triangle) with varying amplitudes, frequencies, and phase shifts. This comprehensive simulation testing is to ensure the designs' full functionality for a wide range of inputs as well as cascading compatibility with

other computational circuits. Figure 5-1 reflects the testing setup for each circuit, and the following steps describe the procedure for simulation testing:

1. Connect +18V, Ground, and -18V to each respective power rail input.
2. Configure a waveform generator (voltage source set to DC, PULSE, or SINE) to the desired input waveform.
3. Configure the appropriate simulation and time-axis settings, and run the simulation.
4. Probe the input and output waveforms and record measurements.

Voltage Adder Simulation

The voltage adder is capable of performing addition and subtraction by simply changing the phase delay of one or more of the inputs. The voltage adder circuit is also capable of summing multiple inputs at the same time, but for simplicity in testing and to prove design functionality, two inputs are added instead. Figure 5-8 demonstrates the addition function of the voltage adder design adding input waves $V_{in,1} = -0.05\sin(2\pi 1000t)$ (green waveform) and $V_{in,2} = 0.35\sin(2\pi 1000t)$ (blue waveform). The resulting red waveform, $V_{out} = 0.3\sin(2\pi 1000t)$ is shifted 180° due to the adder's inverting nature, thus the final output is represented by $V_{out} = -0.3\sin(2\pi 1000t + 180^\circ)$.

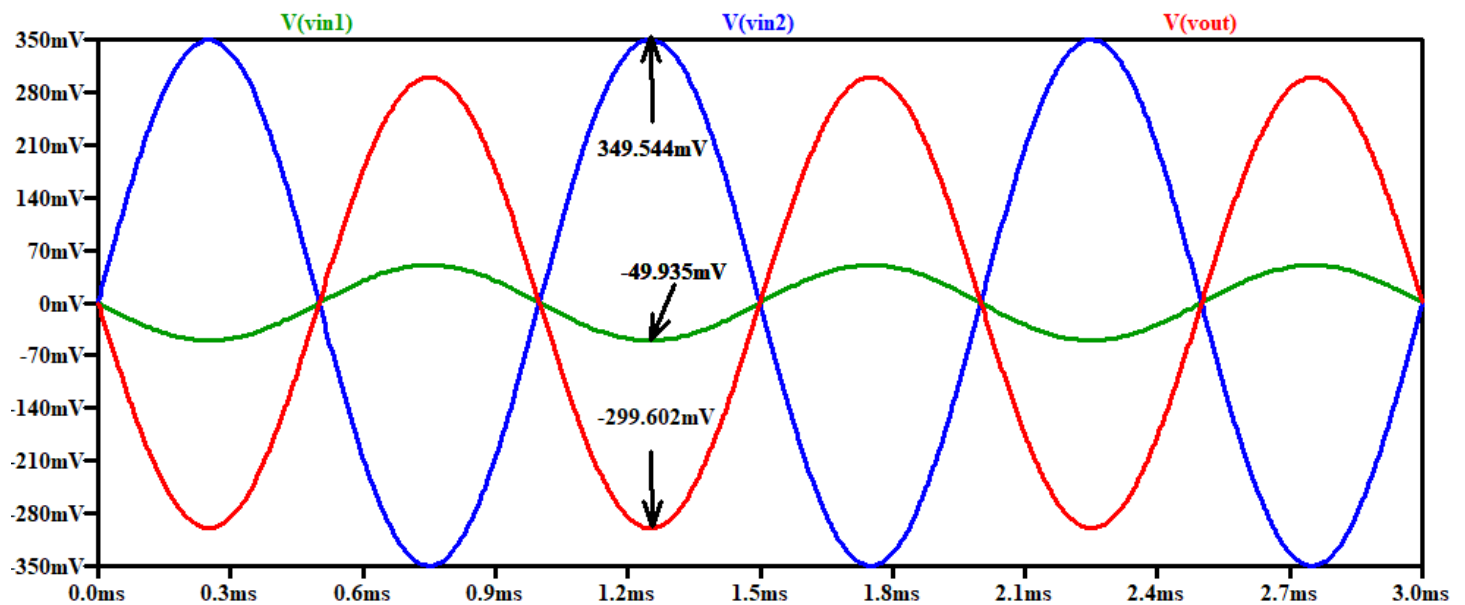


Figure 5-8. LTspice Voltage Adder Simulation (Addition Test)

The voltage adder's subtracting capability is demonstrated in Figure 5-9. For the following test performed, the second adder input is shifted by 180° . The first input (green waveform) is represented by the equation $V_{in,1} = -0.1\sin(2\pi 1000t + 180^\circ)$, and the second input (blue waveform) as $V_{in,2} = 0.7\sin(2\pi 1000t)$. The resulting output (red waveform) is $V_{out} = -0.6\sin(2\pi 1000t + 180^\circ)$.

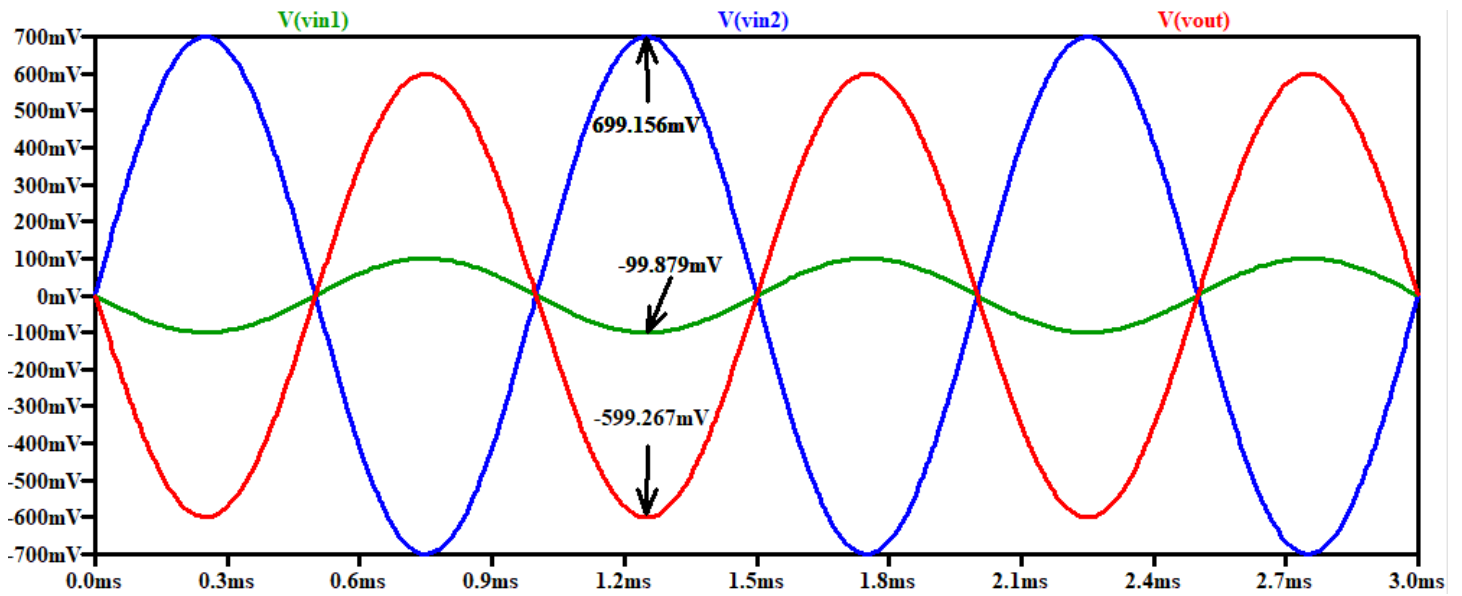


Figure 5-9. LTspice Voltage Adder Simulation (Subtraction Test)

Non-Inverting Multiplier Simulation

The non-inverting multiplier design allows for adjustable gain through the input and feedback resistors. Different combinations of resistors were used to test different gains. Figure 5-10 shows the input and output for a non-inverting multiplier with a gain of 2.2. The input (green) waveform is represented by $V_{in} = \sin(2\pi 1000t)$, and the resulting wave is approximately represented as $V_{out} = 2.2\sin(2\pi 1000t)$. When performing traditional multiplication, $1 \times 2.2 = 2.2$, thus the result is as expected and accurate. The calculation below shows the percent error for this simulation. As this configuration is a non-inverting configuration, the output is the true result without the extra inversion seen with the voltage adder or an inverting multiplier.

$$\text{Percent Error} = \left| \frac{2.196 - 2.2}{2.2} \right| \times 100 = 0.181\%$$

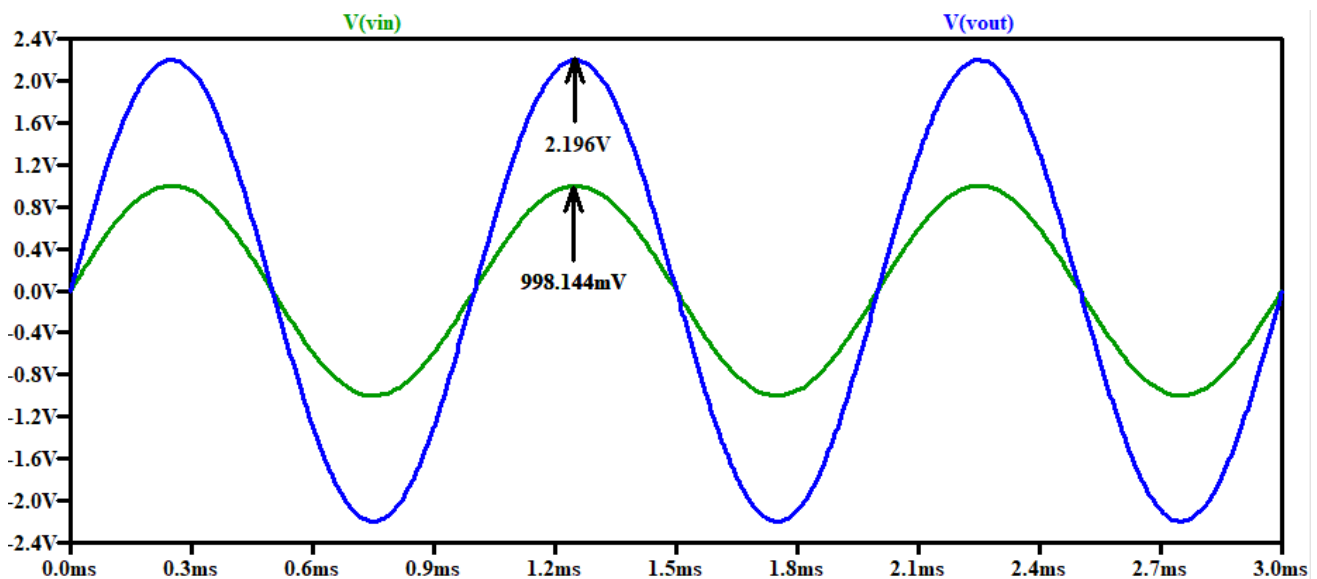


Figure 5-10. LTspice Non-Inverting Multiplier Simulation (Gain of 2.2)

Figure 5-11 shows the input and output for a non-inverting multiplier with a gain of 2.2. The input (green) waveform is represented by $V_{in} = 0.5\sin(2\pi 1000t)$, and the resulting wave is approximately represented as $V_{out} = 3.22\sin(2\pi 1000t)$. When performing traditional multiplication, $0.5 \times 6.45 = 3.225$, thus the result is as expected and accurate. The calculation below shows the percent error for this simulation.

$$\text{Percent Error} = \left| \frac{3.221 - 3.225}{3.225} \right| \times 100 = 0.124\%$$

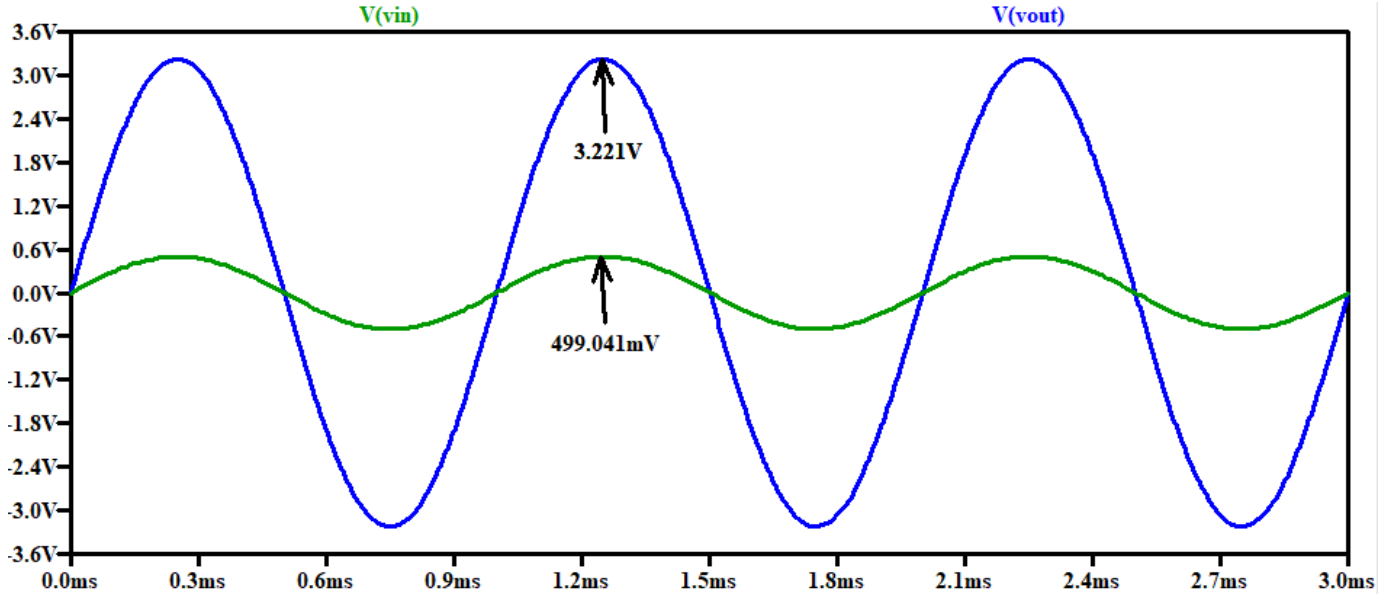


Figure 5-11. LTspice Non-Inverting Multiplier Simulation (Gain of 6.45)

Figure 5-12 shows the input and output for a non-inverting multiplier with a gain of 9.3. The input (green) waveform is represented by $V_{in} = 0.5\sin(2\pi 1000t)$, and the resulting wave is approximately represented as $V_{out} = 4.65\sin(2\pi 1000t)$. When performing traditional multiplication, $0.5 \times 9.3 = 4.65$, thus the result is as expected and accurate. The calculation below shows the percent error for this simulation.

$$\text{Percent Error} = \left| \frac{4.657 - 4.65}{4.65} \right| \times 100 = 0.151\%$$

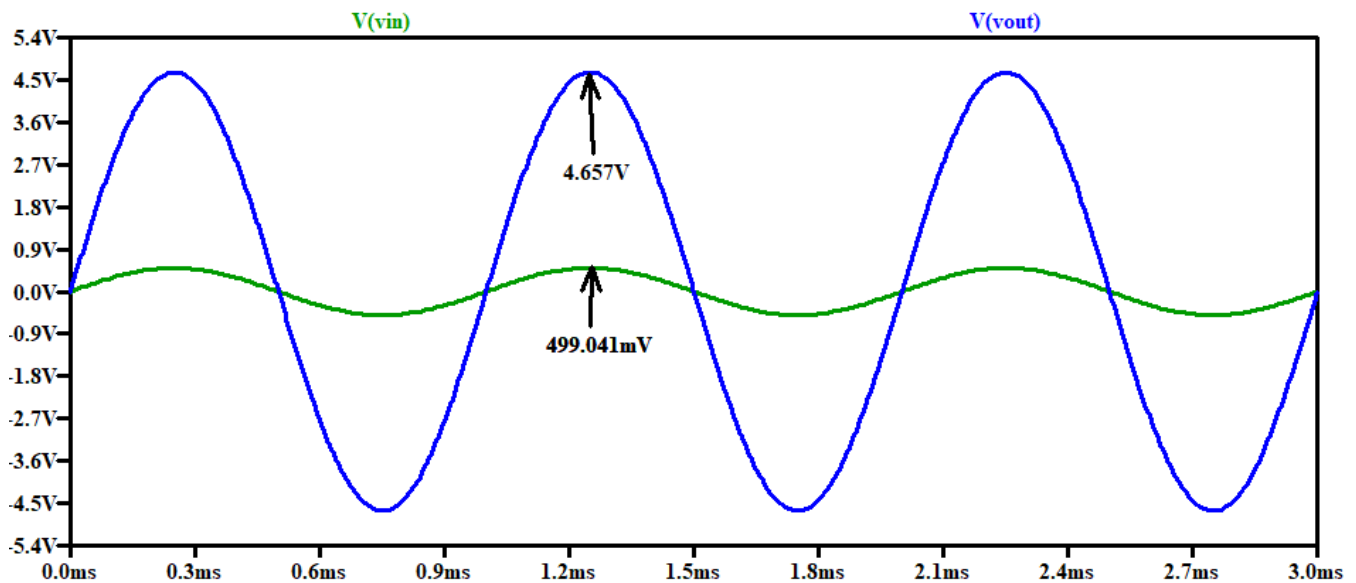


Figure 5-12. LTspice Non-Inverting Multiplier Simulation (Gain of 9.3)

Resistive Divider Simulation

The resistive divider is capable of dividing any input signal by a divisor designated by two resistors. Four types of waveforms (DC, sine, square, and triangle) were tested to verify the circuit design's functionality. All of the following resistive divider tests use a divisor of 2 by using two 1 k Ω resistors. Figure 5-13 shows the input (green) and output (blue) waveforms for the DC waveform test. The input DC wave has a 10V offset, resulting in an output of 5V. The percent error for this test case is shown in the calculation below.

$$\text{Percent Error} = \left| \frac{5-5}{5} \right| \times 100 = 0\%$$

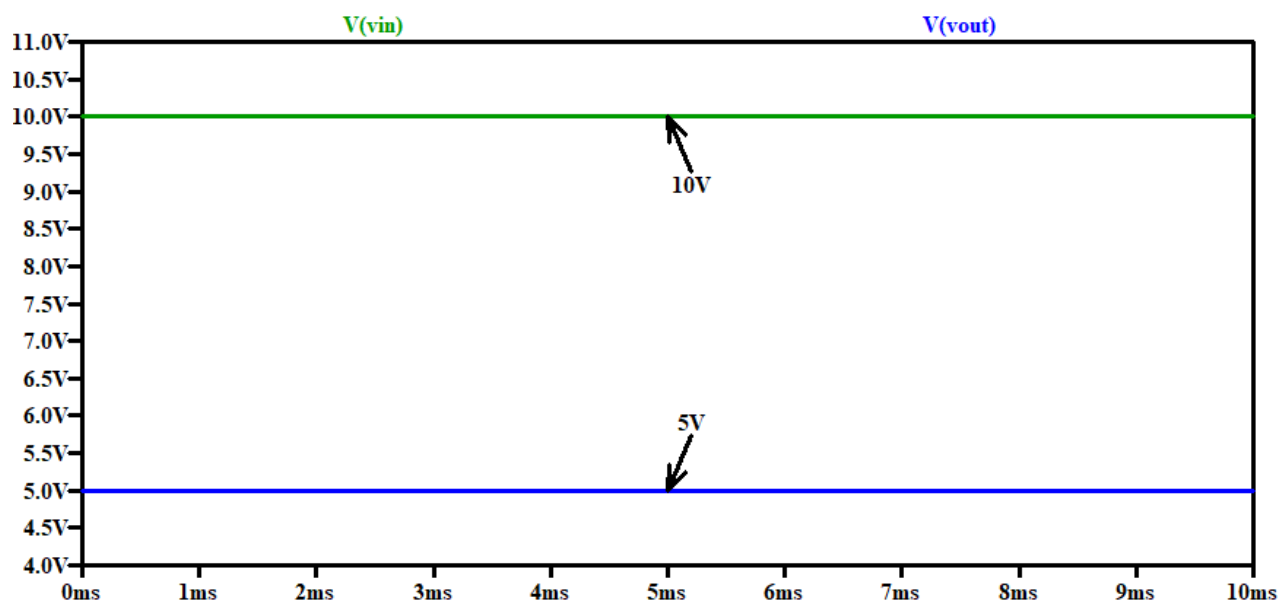


Figure 5-13. LTspice Resistive Divider Simulation for DC Waveforms

Figure 5-14 shows the input (green) and output (blue) waveforms for the sinusoidal waveform test. The input sinusoid is represented as approximately $V_{in} = 5\sin(2\pi 1000t)$, and with a divisor of 2, the output sinusoid is represented as approximately $V_{out} = 2.5\sin(2\pi 1000t)$. The percent error for this test case is shown below.

$$\text{Percent Error} = \left| \frac{2.496 - 2.5}{2.5} \right| \times 100 = 0.160\%$$

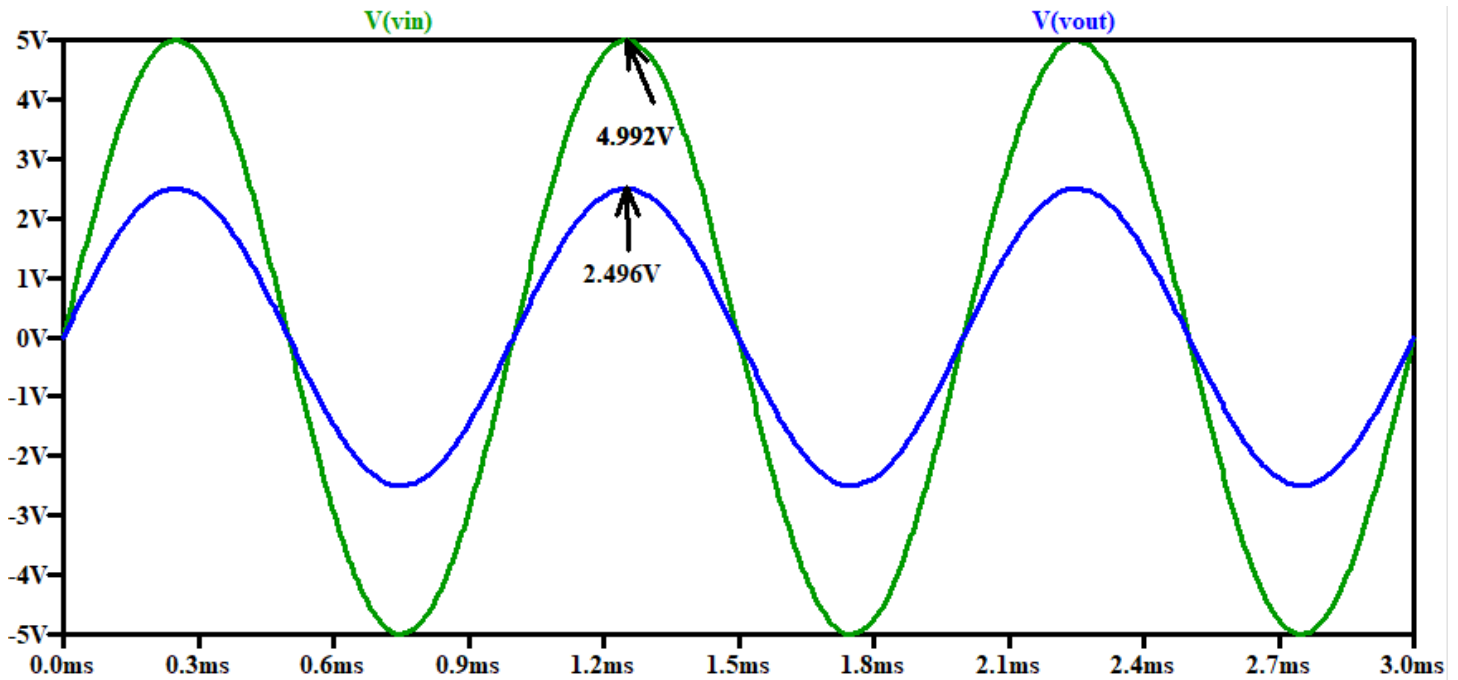


Figure 5-14. LTspice Resistive Divider Simulation for Sinusoidal Waveforms

Figure 5-15 shows the input (green) and output (blue) waveforms for the sinusoidal waveform test. The input is a square wave of $10V_{pp}$, 1 kHz, DC offset of 2.5V, and with a divisor of 2, the output square wave is $5V_{pp}$, 1 kHz, DC offset of 1.25V. The percent error for this test case is shown below.

$$\text{Percent Error} = \left| \frac{2.5 - 2.5}{2.5} \right| \times 100 = 0\%$$

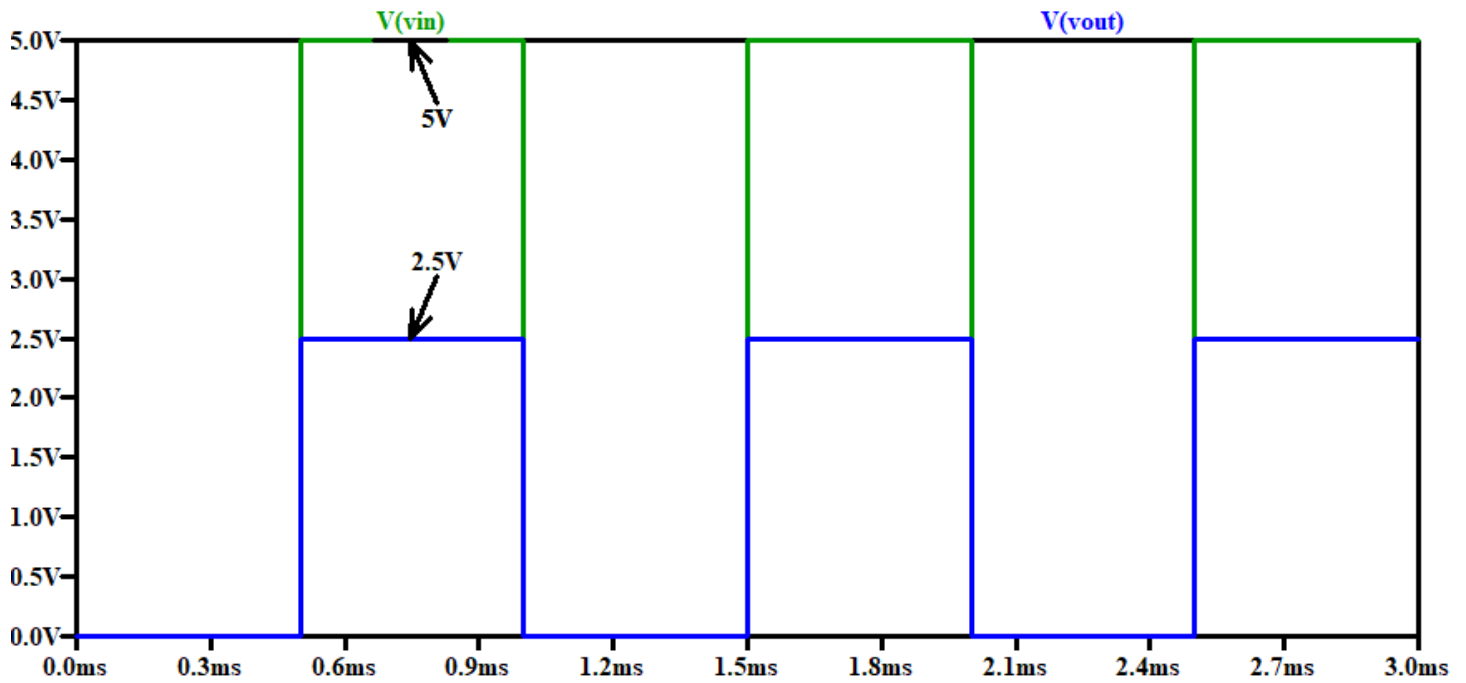


Figure 5-15. LTspice Resistive Divider Simulation for Square Waveforms

Figure 5-16 shows the input (green) and output (blue) waveforms for the sinusoidal waveform test. The input is a triangle wave of $10V_{pp}$, 1 kHz, DC offset of 2.5V, and with a divisor of 2, the output triangle wave is $5V_{pp}$, 1 kHz, DC offset of 1.25V. The percent error for this test case is shown below.

$$\text{Percent Error} = \left| \frac{2.494 - 2.5}{2.5} \right| \times 100 = 0.240\%$$

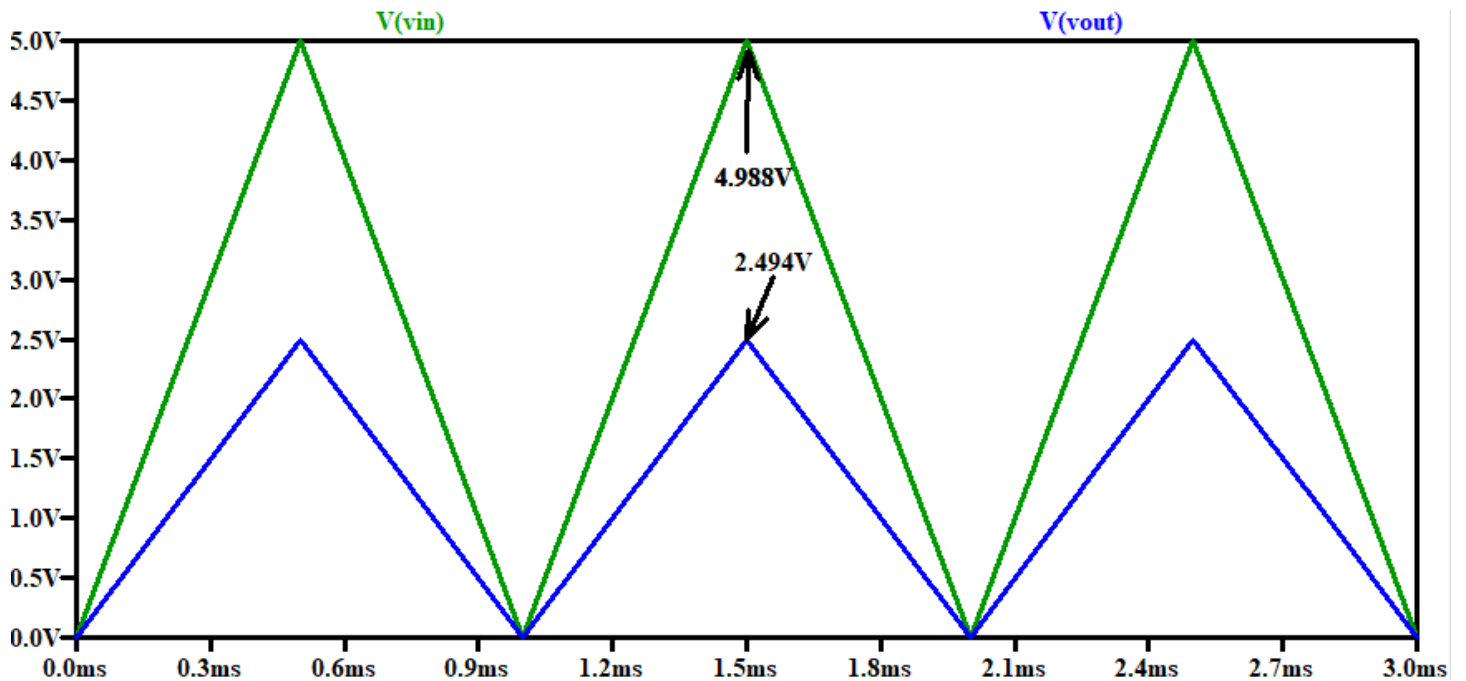


Figure 5-16. LTspice Resistive Divider Simulation for Triangle Waveforms

Differentiator Simulation

The differentiator circuit is also capable of performing the derivative of any input signal with DC waveforms as the exception. The following tests are different types of waveforms (sine, square, and triangle) inputted into the differentiator design. The output should have a 90° phase shift to the left. Figure 5-17 shows the differentiation of an input (green) sine wave of $V_{in} = 0.5\sin(2\pi 1000t)$. The resulting (blue) waveform is approximately represented by $V_{out} = 0.47\sin(2\pi 1000t + 90^\circ)$. The phase difference in degrees is calculated below as well as the percent error for this test case.

$$\phi = \frac{360^\circ(\text{time difference})}{\text{frequency}} = \frac{360^\circ((1.512-1.250)\times 10^3)}{1000 \text{ Hz}} = 94.32^\circ$$

$$\text{Percent Error}_\phi = \left| \frac{94.32-90}{90} \right| \times 100 = 4.80\%$$

$$\text{Percent Error}_{\text{Amplitude}} = \left| \frac{2.494-2.5}{2.5} \right| \times 100 = 0.240\%$$

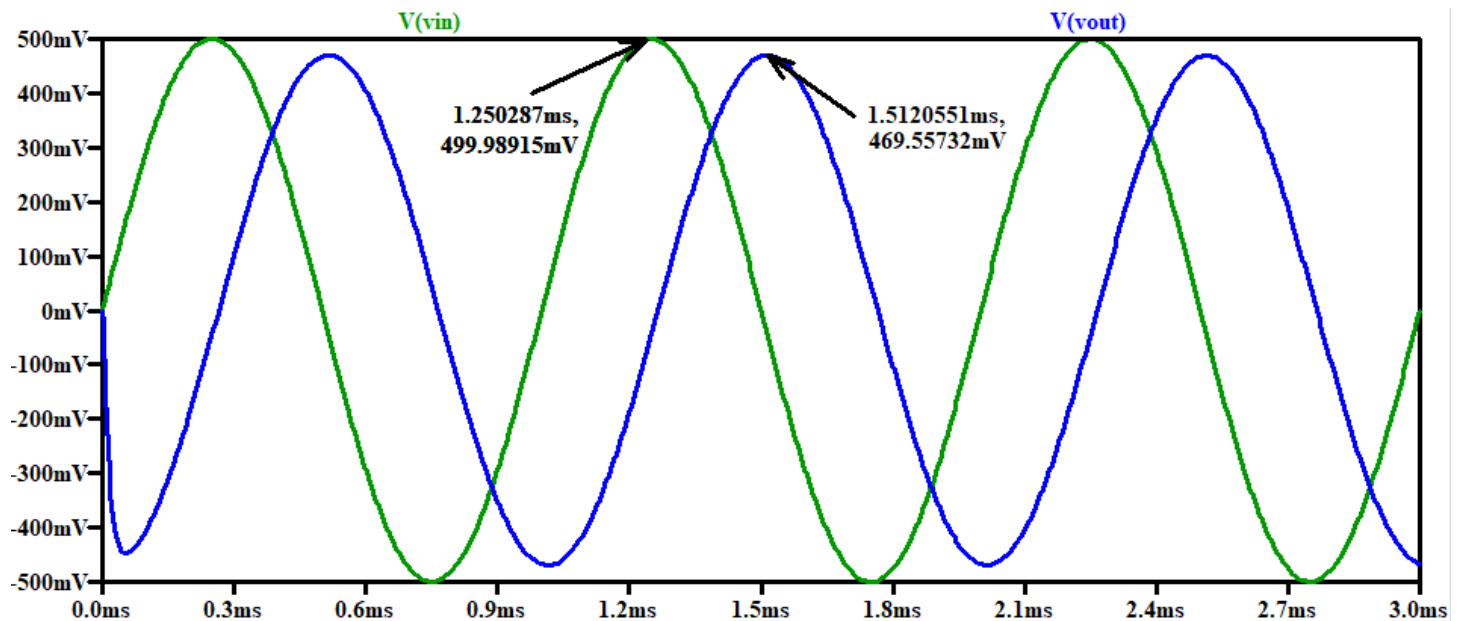


Figure 5-17. LTspice Differentiator Simulation for Sinusoidal Waveforms

For square wave differentiation, the output shifts 180° to the left, with the waveform resembling impulses of alternating sign (positive or negative). Figure 5-18 shows the differentiation of an input (green) square wave of $2V_{pp}$, 1 kHz, and a DC offset of 0.5V. The resulting (blue) waveform has approximately $1.2V_{pp}$, 1 kHz, and a DC offset of 0V. The phase difference in degrees is calculated below as well as the percent error for this test case.

$$\phi = \frac{360^\circ(\text{time difference})}{\text{frequency}} = \frac{360^\circ((2.003-1.502)\times 10^3)}{1000 \text{ Hz}} = 180.63^\circ$$

$$\text{Percent Error}_\phi = \left| \frac{180.63-180}{180} \right| \times 100 = 0.2\%$$

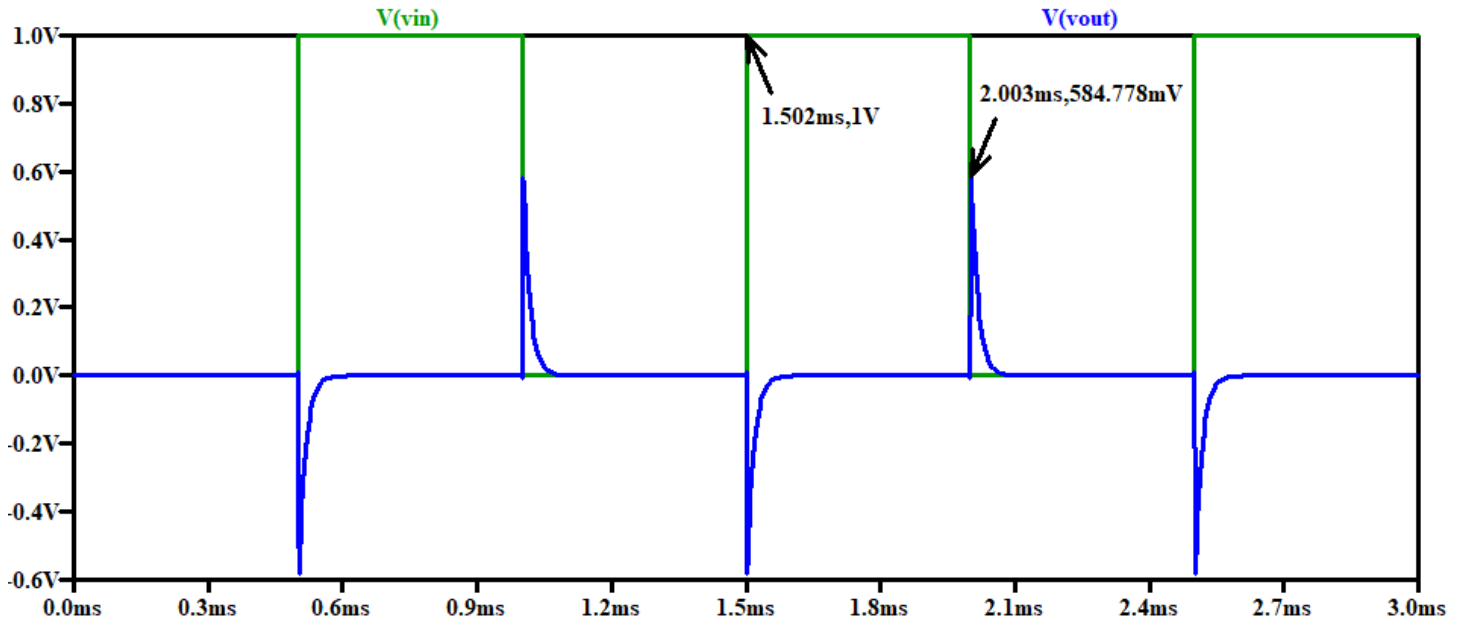


Figure 5-18. LTspice Differentiator Simulation for Square Waveforms

For triangle wave differentiation, the output shifts 180° to the left, with the waveform resembling a square wave. Figure 5-19 shows the differentiation of an input (green) triangle wave of 2V_{pp}, 1 kHz, and a DC offset of 0.5V. The resulting (blue) waveform has approximately 0.88V_{pp}, 1 kHz, and a DC offset of 0V. The phase difference in degrees is calculated below as well as the percent error for this test case.

$$\phi = \frac{360^\circ(\text{time difference})}{\text{frequency}} = \frac{360^\circ((1.762-1.499)\times 10^3)}{1000 \text{ Hz}} = 94.68^\circ$$

$$\text{Percent Error}_\phi = \left| \frac{94.68-90}{90} \right| \times 100 = 5.2\%$$

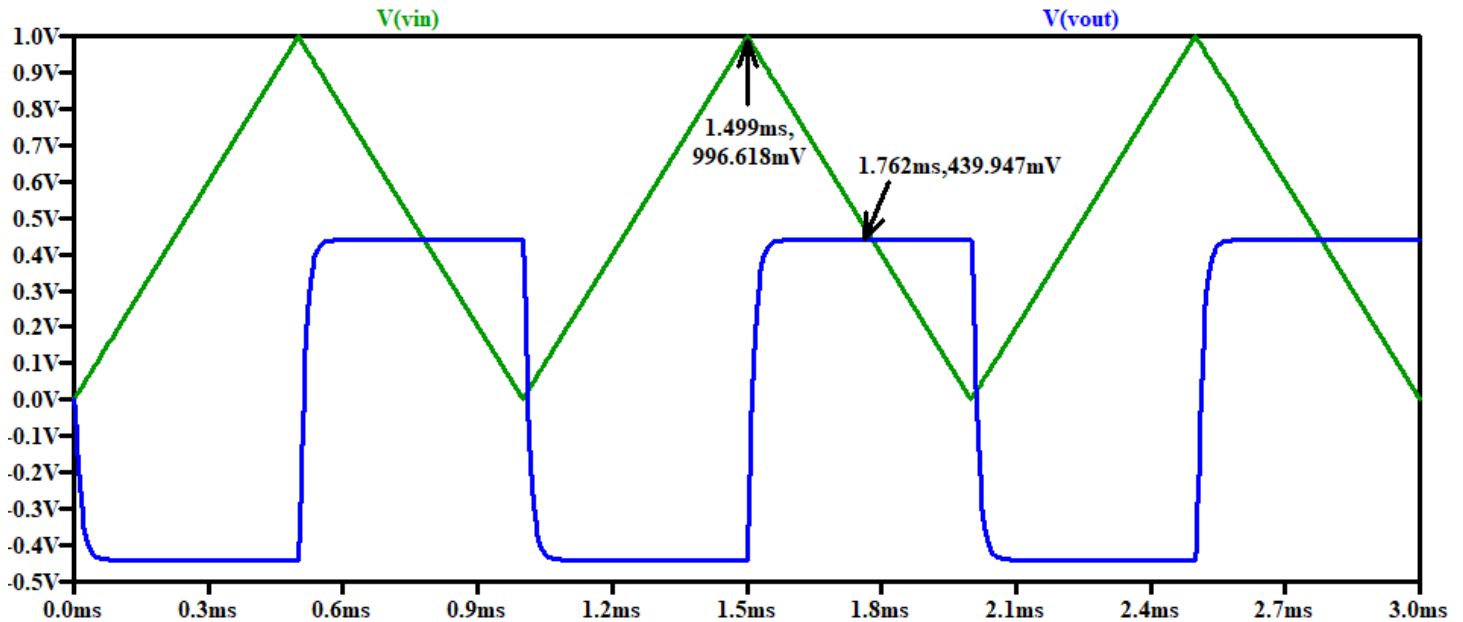


Figure 5-19. LTspice Differentiator Simulation for Triangle Waveforms

Integrator Simulation

The integrator circuit is able to perform the integral of most inputs, with DC waveforms being the exception. The following integration tests include sine, square, and triangle wave inputs. The output waveform should have an approximate 90° phase shift to the right (-90°). Figure 5-20 shows the simulation for an integrated sine wave. The input (green) wave is represented by $V_{in} = 0.5\sin(2\pi 1000t)$, and the output is approximately represented by $V_{out} = 0.5\sin(2\pi 1000t - 90^\circ)$. The phase difference for this test case is calculated below as well as its percent error.

$$\phi = \frac{360^\circ(\text{time difference})}{\text{frequency}} = \frac{360^\circ((1.995 - 2.248) \times 10^3)}{1000 \text{ Hz}} = -91.08^\circ$$

$$\text{Percent Error}_\phi = \left| \frac{-91.08 - (-90)}{(-90)} \right| \times 100 = 1.2\%$$

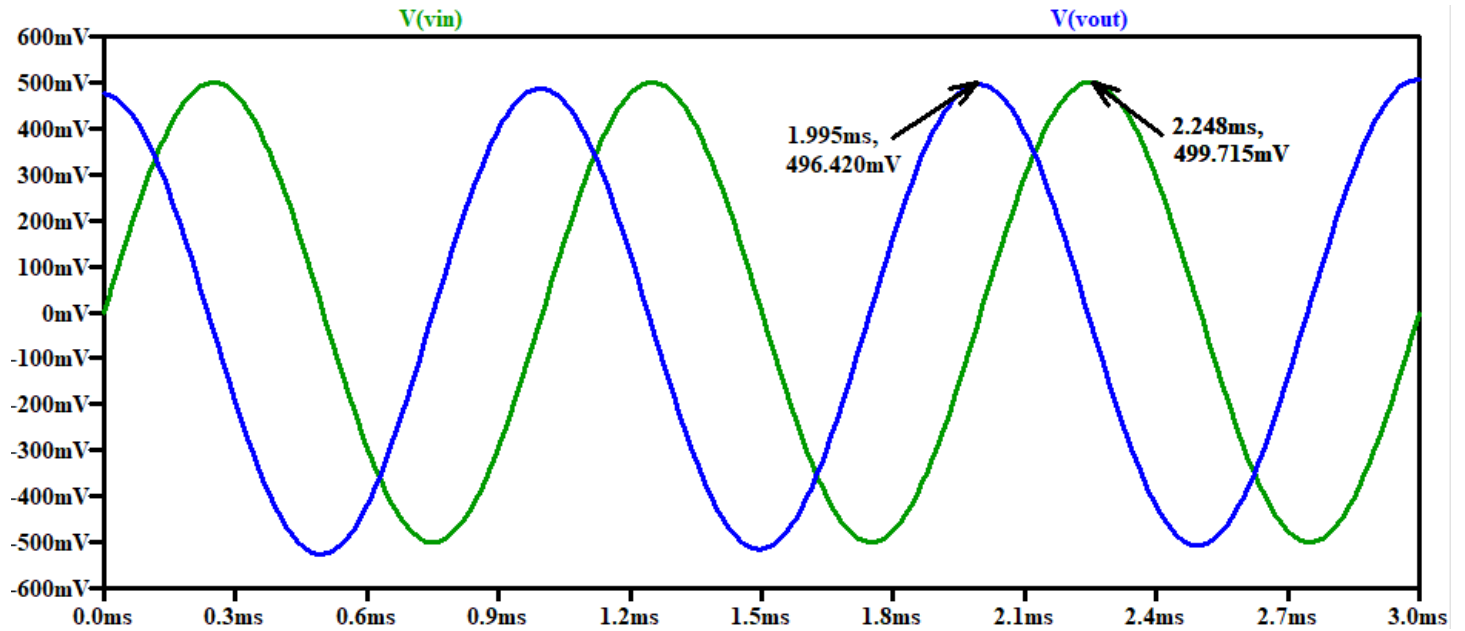


Figure 5-20. LTspice Integrator Simulation for Sinusoidal Waveforms

Figure 5-21 shows the simulation for an integrated square wave. The input (green) wave is $0.5V_{pp}$ and 1 kHz with DC offset of 0V. The triangle wave output is approximately $0.497V_{pp}$ and 1 kHz with DC offset of -4.9V. The phase difference for this test case is calculated below as well as its percent error.

$$\phi = \frac{360^\circ(\text{time difference})}{\text{frequency}} = \frac{360^\circ((0.519329 - 1.001) \times 10^3)}{1000 \text{ Hz}} = -173.402^\circ$$

$$\text{Percent Error}_\phi = \left| \frac{-173.402 - (-180)}{(-180)} \right| \times 100 = 3.666\%$$

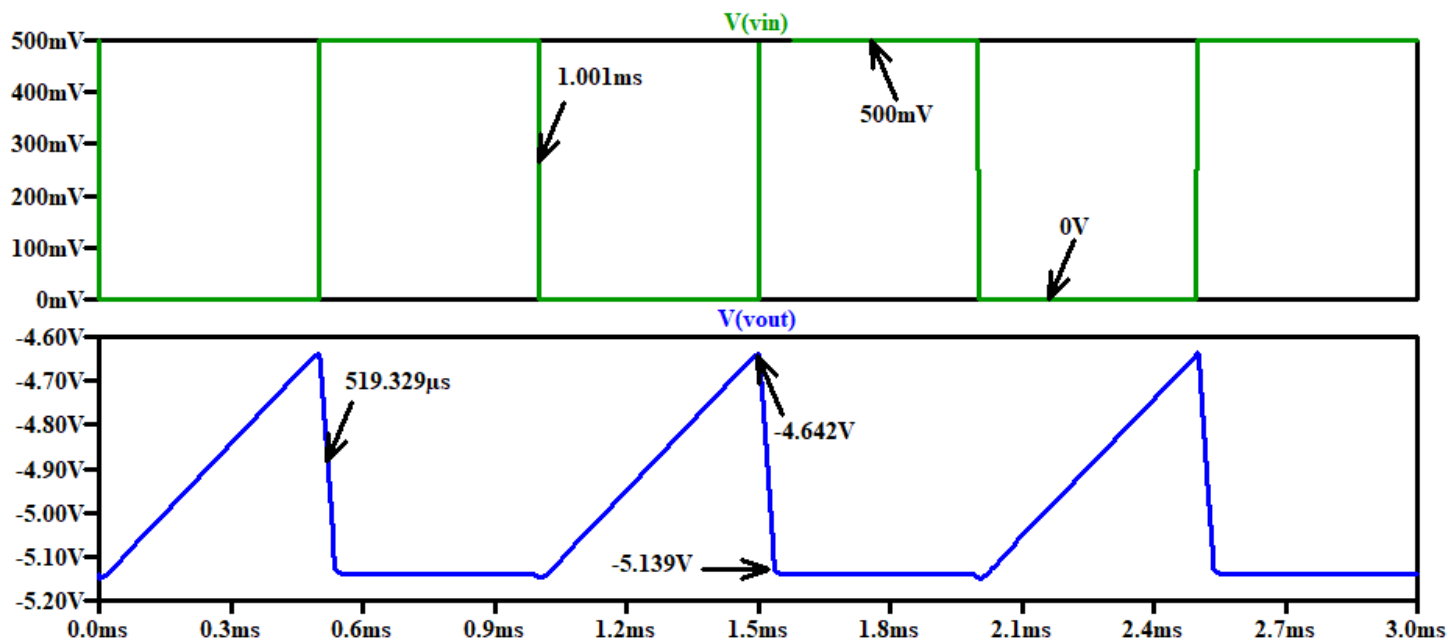


Figure 5-21. LTspice Integrator Simulation for Square Waveforms

Figure 5-22 shows the simulation for an integrated triangle wave. The input (green) wave is $0.5V_{pp}$ and 1 kHz with DC offset of 0.25V. The triangle wave output is approximately $0.494V_{pp}$ and 1 kHz with DC offset of -1.15V. The phase difference for this test case is calculated below as well as its percent error.

$$\phi = \frac{360^\circ(\text{time difference})}{\text{frequency}} = \frac{360^\circ((1.235-1.5) \times 10^{-3})}{1000 \text{ Hz}} = -95.4^\circ$$

$$\text{Percent Error}_\phi = \left| \frac{-95.4 - (-90)}{(-90)} \right| \times 100 = 6\%$$

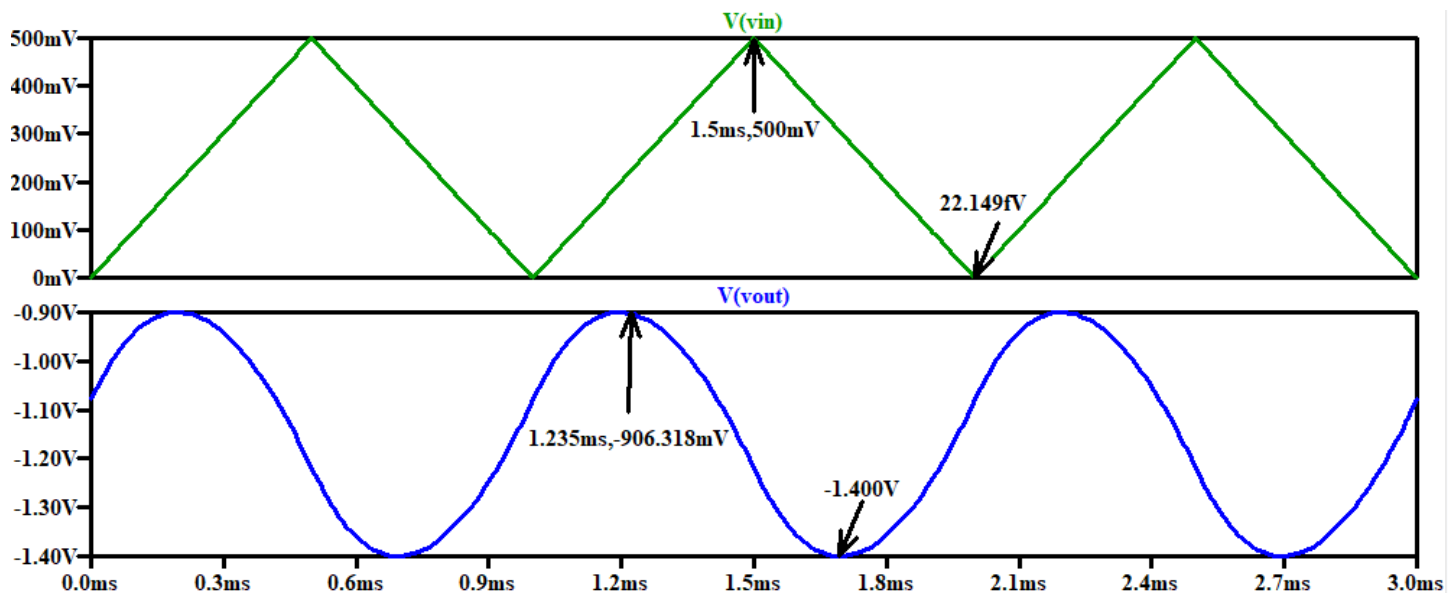


Figure 5-22. LTspice Integrator Simulation for Triangle Waveforms

Breadboard Testing and Results

After thoroughly performing simulation testing on the LTspice circuits, the designs were constructed on breadboards to perform hardware (physical) tests. To perform the breadboard tests, the Rigol Technologies DP832 power supply, Keysight Technologies EDUX1052A function generator, and Keysight Technologies DSOX1204A oscilloscope were used to power the circuit and record measurements. The block diagram shown in Figure 5-1 shows the testing configuration for the hardware testing. The following describes the procedure for breadboard testing:

1. Construct the circuit based on the simulation-based design schematic.
2. Connect +18V, Ground, and -18V to each respective power rail input.
3. Set the waveform generator to the desired input waveform and set it to a high-impedance output.
4. Turn the power source on.
5. Connect the waveform generator to the circuit and turn it on.
5. Probe the input and output waveforms and record measurements.

Voltage Adder Breadboard Testing Results

Table 5-2 shows the breadboard testing results for the voltage adder circuit. For reference, the cells highlighted in blue represent DC voltage testing while white cells denote sinusoidal testing unless otherwise noted by signal type. As expected, DC voltage operations gave accurate output as the highest percent error is 4.17%, which is well below the design requirement of 10%. However, sinusoidal testing displays high inaccuracy as the highest percent error is 22%. The explanation that comes to mind is simply loose connections to the breadboard as an old breadboard used alongside jumper wires that sometimes broke during testing.

Table 5-2. Tabulated Breadboard Data for the Voltage Adder Circuit Test Cases

Operation	Resistor Combination	Input 1 [Vpp]	Input 2 [Vpp]	Output [Vpp]	Expected [Vpp]	Error [%]
Addition	100k Ω , 100k Ω , 100k Ω	0.1	0.7	-0.8	-0.8	0.00%
	100k Ω , 100k Ω , 100k Ω	2.0	3	-4.925	-5.0	1.50%
	100k Ω , 100k Ω , 100k Ω	10.0	5	-14.825	-15.0	1.17%
	100k Ω , 100k Ω , 100k Ω	0.1	0.7	0.72	0.8	10.00%
	100k Ω , 100k Ω , 100k Ω	2.0	3	3.9	5.0	22.00%
	100k Ω , 100k Ω , 100k Ω	10.0	5	11.7	15.0	22.00%
Subtraction	100k Ω , 100k Ω , 100k Ω	-0.1	0.7	-0.575	-0.6	4.17%
	100k Ω , 100k Ω , 100k Ω	-2.0	3	-1	-1.0	0.00%
	100k Ω , 100k Ω , 100k Ω	-10.0	5	4.975	5.0	0.50%
	100k Ω , 100k Ω , 100k Ω	-0.1	0.7	-0.68	-0.6	13.33%
	100k Ω , 100k Ω , 100k Ω	-2.0	3	-1.12	-1.0	12.00%
	100k Ω , 100k Ω , 100k Ω	-10.0	5	-4.25	-5.0	15.00%

Non-Inverting Multiplier Breadboard Testing Results

Table 5-3 displays the breadboard testing results for the non-inverting voltage multiplier circuit. The blue cells indicate the best frequency of operations. While the white cells denote different maximum frequencies of operation within the 10% error engineering specification mentioned in Chapter 3. DC testing shows the most accurate results due to the signal lacking frequency. The frequency of the signal plays a major role in accuracy

due to the LM741's gain bandwidth product. Since DC does not have a frequency, it generates the most accurate results. However, the sinusoidal signal used for testing had a frequency of 1kHz, which impacts the gain bandwidth product and in turn, increases error. In addition, this circuit amplifies the signal and it can also amplify noise generated by loose connections which also increases error.

Table 5-3. Tabulated Breadboard Data for the Non-Inverting Voltage Multiplier Circuit Test Cases

Gain	Resistor Combination	Frequency [kHz]	Phase Difference [ns]	Phase Difference [°]	Input [Vpp]	Output [Vpp]	Expected [Vpp]	Error [%]
2.2	10kΩ, 12kΩ	40	-357	-5.14080	2	4.42	4.4	0.45%
2.2	10kΩ, 12kΩ	10	-105	-0.37800	10	22.2	22	0.91%
2.2	10kΩ, 12kΩ	8	-256	-0.73728	12	26.9	26.4	1.89%
2.2	10kΩ, 12kΩ	80	-1000	-28.80000	2	4.02	4.4	8.64%
2.2	10kΩ, 12kΩ	16	-3297	-18.99072	10	21.4	22	2.73%
2.2	10kΩ, 12kΩ	14.75	-5300	-28.14300	12	23.7	26.4	10.23%
6.45	3.3kΩ, 18kΩ	23.8	-1087	-9.31342	1	6.47	6.45	0.31%
6.45	3.3kΩ, 18kΩ	6	-165	-0.35640	3	19.5	19.35	0.78%
6.45	3.3kΩ, 18kΩ	6	-1392	-3.00672	5	32.4	32.25	0.47%
6.45	3.3kΩ, 18kΩ	4.685	-1650	-2.78289	1	5.87	6.45	8.99%
6.45	3.3kΩ, 18kΩ	19	-4200	-28.72800	3	17.5	19.35	9.56%
6.45	3.3kΩ, 18kΩ	11.58	-7256	-30.24881	5	29.1	32.25	9.77%
9.3	1.2kΩ, 10kΩ	11.58	-1757	-7.32458	1	9.5	9.3	2.15%
9.3	1.2kΩ, 10kΩ	10	-3150	-11.34000	2	18.8	18.6	1.08%
9.3	1.2kΩ, 10kΩ	8	-3650	-10.51200	3	28.87	27.9	3.48%
9.3	1.2kΩ, 10kΩ	34	-2684	-32.85216	1	8.4	9.3	9.68%
9.3	1.2kΩ, 10kΩ	18.8	-5050	-34.17840	2	16.9	18.6	9.14%
9.3	1.2kΩ, 10kΩ	12.9	-7562	-35.11793	3	25.4	27.9	8.96%

Resistive Divider Breadboard Testing Results

As shown in Table 5-4, the resistive divider is verified to work across various waveforms. Accuracy is well within design requirements. However, the issue with using a potentiometer is that it is very sensitive to adjustments and the resistance always varies when tuning to the right divisor. This causes errors in the output due to tolerance and the sensitivity of the potentiometer, which may explain why there are varying levels of error among the results.

Table 5-4 Tabulated Breadboard Data for the Resistive Divider Circuit Test Cases

Wave Type	Resistor Combination	Input [Vpp]	Divisor	Output [Vpp]	Expected [Vpp]	Error [%]
DC	1k Ω , 1k Ω	10	2	5.1	5.000	2.00%
	1k Ω , 3.9k Ω	10	4.9	2.1	2.041	2.90%
	1k Ω , 10k Ω	10	11	0.979	0.909	7.69%
Sine	1k Ω , 1k Ω	10	2	5.3	5.000	6.00%
	1k Ω , 3.9k Ω	10	4.9	2.17	2.041	6.33%
	1k Ω , 10k Ω	10	11	0.96	0.909	5.60%
Square	1k Ω , 1k Ω	10	2	5.3	5.000	6.00%
	1k Ω , 3.9k Ω	10	4.9	2.21	2.041	8.29%
	1k Ω , 10k Ω	10	11	0.96	0.909	5.60%
Triangle	1k Ω , 1k Ω	10	2	5.1	5.000	2.00%
	1k Ω , 3.9k Ω	10	4.9	2.13	2.041	4.37%
	1k Ω , 10k Ω	10	11	0.92	0.909	1.20%

Differentiator Breadboard Testing Results

As shown in Table 5-5, the differentiator operated as expected with the output and phase difference error never exceeding the requirement of $\pm 10\%$ tolerance. The frequency also shows that this circuit (and all others tested) will be in a low-frequency range. This also satisfies the requirement for the customer as the frequency of a standard US power grid of 50 Hz. This ensures that these circuits will be suitable for power grid computations.

Table 5-5. Tabulated Breadboard Data for the Differentiator Circuit Test Cases

Input	Resistor Combination	Frequency [kHz]	Phase Difference [°]	Expected	Phase Difference Error [%]	Input [Vpp]	Output [Vpp]	Expected Output [Vpp]	Output Error [%]
Sine	1.5k Ω , 15k Ω	1	97.79	90	8.66%	1	1.05	1	5.00%
	1.5k Ω , 15k Ω	0.93	97.69	90	8.54%	6	6.3	6	5.00%
	1.5k Ω , 15k Ω	0.89	98.85	90	9.83%	12	12.7	12	5.83%
Square	1.5k Ω , 1k Ω	1	179.79	180	0.12%	1	1.27	1.21	4.96%
	1.5k Ω , 820 Ω	1	178.21	180	0.99%	6	7	7.2	2.78%
	1.5k Ω , 1.2k Ω	1	176.51	180	1.94%	12	13.1	14.5	9.66%
Triangle	1.5k Ω , 22k Ω	1	94.42	90	4.91%	1	1.01	1	1.00%
	1.5k Ω , 22k Ω	1	95.62	90	6.24%	6	6.5	6	8.33%
	1.5k Ω , 18k Ω	1	95.75	90	6.39%	12	11.3	12	5.83%

Integrator Testing Results

Table 5-6 below displays the collected data for the various test cases performed on the integrator circuit. When testing, the potentiometer could not be probed when the circuit was active as this caused distortion on the output. To solve this issue, the potentiometer was tuned first and then the circuit was activated. Additionally, the potentiometer needed to be adjusted for different sine wave amplitudes due to the gain of the circuit. Another issue is that any resistance value of R_F below 470 k Ω would cause the phase difference to increase. However, when comparing the input and output amplitudes, the design requirement of a $\pm 10\%$ tolerance was met successfully.

Table 5-6. Tabulated Breadboard Data for the Integrator Circuit Test Cases

Wave Type	Resistor Combination	Potentiometer Resistance [k Ω]	Frequency [kHz]	Expected Phase Difference [$^\circ$]	Actual Phase Difference [$^\circ$]	Error [%]	Expected Input [Vpp]	Input [Vpp]	Output [Vpp]	Error [%]
Sine	1k Ω , 470k Ω	97.1	1	-90.000	-91.090	1.21%	0.5	0.51	0.51	0.00%
	1k Ω , 470k Ω	97.1	1	-90.000	-93.400	3.78%	1	1.01	0.96	4.95%
	1k Ω , 470k Ω	95.6	1	-90.000	-92.470	2.74%	2	2.01	2.01	0.00%
Square	1k Ω , 470k Ω	97.47	1	-90.000	-93.200	3.56%	0.5	0.59	0.59	0.00%
	1k Ω , 470k Ω	97.47	1	-90.000	-94.580	5.09%	1	1.15	1.15	0.00%
	1k Ω , 470k Ω	97.47	1	-90.000	-93.990	4.43%	1.8	2.09	2.09	0.00%
Triangle	1k Ω , 470k Ω	94.5	1	-90.000	-91.770	1.97%	0.5	0.5	0.54	8.00%
	1k Ω , 470k Ω	94.5	1	-90.000	-90.940	1.04%	1	0.98	1.05	7.14%
	1k Ω , 470k Ω	94.5	1	-90.000	-94.570	5.08%	2	2.01	2.01	0.00%

Printed Circuit Board Tests and Results

After testing the designed computational circuits on breadboards and confirming their functionality and compatibility with each other, the circuits were manufactured as PCBs. The appropriate components, both passive and active previously used in the breadboard testing, were soldered onto the PCBs. Once conductivity was confirmed among each of the circuits, each were tested using the same test cases used in the breadboard testing. The same aforementioned Rigol Technologies power supply, Keysight Technologies function generator, and Keysight Technologies oscilloscope were used to power the circuits and obtain data. The PCB testing apparatus is visualized in Figure 5-1, and the following describes the procedure for PCB testing:

1. Connect +18V, Ground, and -18V to each respective power rail input.
2. Set the waveform generator to the desired input waveform and set it to a high-impedance output.
3. Turn the power source on.
4. Connect the waveform generator to the circuit and turn it on.
5. Probe the input and output waveforms and record measurements.

Voltage Adder PCB Testing Results

Table 5-7 shows the test cases for the PCB voltage adder. For reference, the cells highlighted in blue represent DC voltage testing while white cells denote sinusoidal testing unless otherwise noted by signal type. Comparing these results to the breadboard testing shows massive improvements in accuracy as operations on sinusoids are well below the requirement. This also confirms the explanation for the source of error as soldering components

provide a rigid connection between components. However, the accuracy of operations with DC signals decreased as the breadboard testing showed percent errors below 5%. This displays a tradeoff as in order to improve accuracy with sinusoids, the PCB provides the best result at the cost of accuracy with DC signals.

Table 5-7. Tabulated PCB Data for the Voltage Adder Circuit Test Cases

Operation	Resistor Combination	Input 1 [Vpp]	Input 2 [Vpp]	Output [Vpp]	Expected [Vpp]	Error [%]
Addition	100k Ω , 100k Ω , 100k Ω	0.1	0.7	-0.75	-0.8	6.25%
	100k Ω , 100k Ω , 100k Ω	2.0	3	-4.72	-5.0	5.60%
	100k Ω , 100k Ω , 100k Ω	10.0	5	-14.5	-15.0	3.33%
	100k Ω , 100k Ω , 100k Ω	0.1	0.7	0.76	0.8	5.00%
	100k Ω , 100k Ω , 100k Ω	2.0	3	4.78	5.0	4.40%
	100k Ω , 100k Ω , 100k Ω	10.0	5	15.3	15.0	2.00%
Subtraction	100k Ω , 100k Ω , 100k Ω	-0.5	1	-0.473	-0.5	5.40%
	100k Ω , 100k Ω , 100k Ω	-2.0	3	-0.95	-1.0	5.00%
	100k Ω , 100k Ω , 100k Ω	-10.0	5	5.075	5.0	1.50%
	100k Ω , 100k Ω , 100k Ω	-0.1	0.7	-0.61	-0.6	1.67%
	100k Ω , 100k Ω , 100k Ω	-2.0	3	-0.95	-1.0	5.00%
	100k Ω , 100k Ω , 100k Ω	-10.0	5	-5.075	-5.0	1.50%

Non-Inverting Multiplier PCB Test Results

Table 5-8 shows the PCB Testing results for the voltage multiplier. The blue cells indicate the best frequency of operations. While the white cells denote different maximum frequencies of operation within the 10% error engineering specification mentioned in Chapter 3. The tests show that design requirements are still met, but barely so in the case of maximum operating frequencies. This is due to the gain bandwidth product of the LM741 as at higher frequencies, there is a loss of available gain. This causes inaccuracies in the output, which is clearly demonstrated in the maximum operating frequencies.

Table 5-8. Tabulated PCB Data for the Non-Inverting Voltage Multiplier Circuit Test Cases

Gain	Resistor Combination	Frequency [kHz]	Phase Difference [ns]	Phase Difference [°]	Input [Vpp]	Output [Vpp]	Expected [Vpp]	Error [%]
2.2	10kΩ, 12kΩ	81	-837	-24.36	2	4.4	4.4	0.00%
2.2	10kΩ, 12kΩ	17.5	-3484	-21.93	10	22.1	22	0.45%
2.2	10kΩ, 12kΩ	15	-3520	-19.01	12	26.5	26.4	0.38%
2.2	10kΩ, 12kΩ	99.5	-1011	-36.31	2	3.9	4.4	11.36%
2.2	10kΩ, 12kΩ	20.6	-3968	-29.43	10	20.1	22	8.64%
2.2	10kΩ, 12kΩ	17.6	-5200	-32.90	12	23.7	26.4	10.23%
6.45	3.3kΩ, 18kΩ	24	-1304	-11.27	1	6.5	6.45	0.78%
6.45	3.3kΩ, 18kΩ	19	-2848	-19.50	3	19.3	19.35	0.26%
6.45	3.3kΩ, 18kΩ	6	-1696	-3.66	5	32.6	32.25	1.09%
6.45	3.3kΩ, 18kΩ	56.5	-1714	-34.82	1	5.8	6.45	10.08%
6.45	3.3kΩ, 18kΩ	22.5	-3928	-31.90	3	17.7	19.35	8.53%
6.45	3.3kΩ, 18kΩ	13.5	-6752	-32.87	5	29.3	32.25	9.15%
9.3	1.2kΩ, 10kΩ	24	-2296	-19.84	1	9.4	9.3	1.08%
9.3	1.2kΩ, 10kΩ	17.5	-4512	-28.45	2	18.5	18.6	0.54%
9.3	1.2kΩ, 10kΩ	11.9	-6712	-28.71	3	27.7	27.9	0.72%
9.3	1.2kΩ, 10kΩ	36	-2996	-38.90	1	8.4	9.3	9.68%
9.3	1.2kΩ, 10kΩ	20	-5084	-36.60	2	16.9	18.6	9.14%
9.3	1.2kΩ, 10kΩ	14	-7960	-40.12	3	25.3	27.9	9.32%

Resistive Divider PCB Testing Results

Table 5-9 shows the PCB testing for the resistive divider. Comparing these results to the breadboard testing, the percent error decreased drastically as the percent error never goes above 1%. This further confirms the hypothesis of the breadboard connections being faulty as the percent error improved with the PCB circuits. However, the issue with the sensitivity of the potentiometers remains as the potentiometers needed to be tuned precisely to generate the results desired.

Table 5-9. Tabulated PCB Data for the Resistive Divider Circuit Test Cases

Wave Type	Resistor Combination	Input [Vpp]	Divisor	Output [Vpp]	Expected [Vpp]	Error [%]
DC	1k Ω , 1k Ω	10	2	5	5.000	0.00%
	1k Ω , 3.9k Ω	10	4.9	2.05	2.041	0.45%
	1k Ω , 10k Ω	10	11	0.91	0.909	0.10%
Sine	1k Ω , 1k Ω	10	2	5.03	5.000	0.60%
	1k Ω , 3.9k Ω	10	4.9	2.05	2.041	0.45%
	1k Ω , 10k Ω	10	11	0.9	0.909	1.00%
Square	1k Ω , 1k Ω	10	2	5	5.000	0.00%
	1k Ω , 3.9k Ω	10	4.9	2.03	2.041	0.53%
	1k Ω , 10k Ω	10	11	0.91	0.909	0.10%
Triangle	1k Ω , 1k Ω	10	2	5.03	5.000	0.60%
	1k Ω , 3.9k Ω	10	4.9	2.03	2.041	0.53%
	1k Ω , 10k Ω	10	11	0.91	0.909	0.10%

Differentiator PCB Testing Results

Table 5-10 shows the PCB differentiator test cases. Compared to the breadboard testing, only sine waves were tested due to hardware limitations. The PCB design was only compatible with a sine wave, whereas square and triangle waves required different resistor values to work properly. Besides this, the PCB differentiator works with high accuracy for the output as the percent error never goes above 4%. However, the phase difference error is quite high as the percent error is close to the maximum 10% imposed by design requirements.

Table 5-10. Tabulated PCB Data for the Differentiator Circuit Test Cases

Type of Sine Derivative	Resistor Combination	Frequency [kHz]	Actual Phase Difference [°]	Expected Phase Difference [°]	Error [%]	Input [Vpp]	Output [Vpp]	Expected Output [Vpp]	Output Error [%]
First-Order	1.5k Ω , 15k Ω	1	97.60	90	8.44%	1	1.01	1	1.00%
	1.5k Ω , 15k Ω	0.93	97.41	90	8.23%	6	5.9	6	1.67%
	1.5k Ω , 15k Ω	0.89	98.53	90	9.48%	12	11.9	12	0.83%
Second-Order	2 x (1.5k Ω , 15k Ω)	1	165.52	180.00	8.04%	1	0.98	1	2.00%
	2 x (1.5k Ω , 15k Ω)	0.93	164.63	180.00	8.54%	6	5.9	6	1.67%
	2 x (1.5k Ω , 15k Ω)	0.89	162.64	180.00	9.64%	12	11.9	12	0.83%
Third-Order	3 x (1.5k Ω , 15k Ω)	1	251.77	270.00	6.75%	1	1.01	1	1.00%
	3 x (1.5k Ω , 15k Ω)	0.93	255.60	270.00	5.33%	6	6.2	6	3.33%
	3 x (1.5k Ω , 15k Ω)	0.89	255.85	270.00	5.24%	12	11.9	12	0.83%

Integrator PCB Testing Results

Table 5-11 shows the testing data for the PCB integrator. From the results, the accuracy of this design is well within design requirements as percent errors rarely go above 5%. In addition, additional test cases of double and triple integration were added to observe the effects of cascading. The error does increase when the integrators are cascaded as the error is amplified with each cascade.

Table 5-11. Tabulated PCB Data for the Integrator Circuit Test Cases

Input	Resistor Combination	Frequency [kHz]	Expected Phase Difference [°]	Actual Phase Difference [°]	Error [%]	Expected Input [Vpp]	Input [Vpp]	Output [Vpp]	Error [%]
Sine	1k Ω , 470k Ω	1	-90.000	-91.610	1.79%	0.5	0.5	0.48	4.00%
	1k Ω , 470k Ω	1	-90.000	-90.940	1.04%	1	1.01	0.98	2.97%
	1k Ω , 470k Ω	1	-90.000	-92.100	2.33%	2	2.09	2.09	0.00%
Sine (Double Integration)	1k Ω , 470k Ω	1	180.000	173.660	3.52%	0.5	0.535	0.531	0.75%
	1k Ω , 470k Ω	1	180.000	174.450	3.08%	1	1.05	1.09	3.81%
	1k Ω , 470k Ω	1	180.000	172.640	4.09%	2	2.05	2.07	0.98%
Sine (Triple Integration)	1k Ω , 470k Ω	1	270.000	286.090	5.96%	0.5	0.56	0.52	7.14%
	1k Ω , 470k Ω	1	270.000	262.960	2.61%	1	1.07	1.07	0.00%
	1k Ω , 470k Ω	1	270.000	261.490	3.15%	2	2.07	2.03	1.93%
Square	1k Ω , 470k Ω	1	-90.000	-93.580	3.98%	0.5	0.52	0.52	0.00%
	1k Ω , 470k Ω	1	-90.000	-90.400	0.44%	1	1.05	1.03	1.90%
	1k Ω , 470k Ω	1	-90.000	-90.550	0.61%	2	2.13	2.05	3.76%
Triangle	1k Ω , 470k Ω	1	-90.000	-89.440	0.62%	0.5	0.52	0.52	0.00%
	1k Ω , 470k Ω	1	-90.000	-91.800	2.00%	1	1.05	1.03	1.90%
	1k Ω , 470k Ω	1	-90.000	-91.090	1.21%	2	2.09	2.05	1.91%

Breadboard and PCB Testing Summary

As shown in Figure 5-23, the average accuracy of each test performed on the individual breadboards and PCBs did not exceed 10.00%, which conforms to the tolerances designated for the project. Another inferred conclusion is that accuracy overall decreased when testing the PCB circuits compared to the breadboard circuits. The noticeable difference in error between the breadboard circuits and PCB circuits can be attributed to the metal within the breadboards. As the long pieces of metal are aligned similarly to that of a capacitor, breadboards tend to have parasitic capacitances and higher probabilities of crosstalk when testing circuits. Furthermore, breadboards form less reliable connections as they rely on contact between the internal metal and the component, whereas PCB connections are soldered and form complete connections.

In addition to the aforementioned potentials for error, the mathematical circuits with the highest percent error (voltage adder and differentiator) tend to be slightly more inaccurate as they do not have potentiometers that help to “tune” the output so that the circuit works for almost all inputs. Circuits that use these potentiometers like the resistive divider and integrator have significantly lower average percent errors (0.37% and 1.56% respectively).

Breadboard Percent Error vs. PCB Percent Error

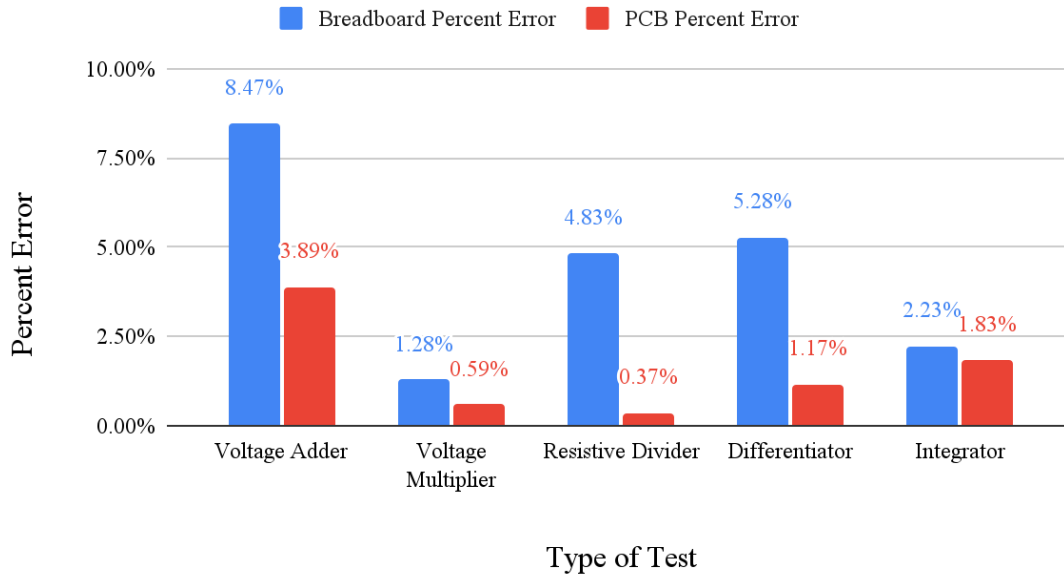


Figure 5-23. Graphical Representation of the Breadboard Data Accuracy versus the PCB Data Accuracy

Execution Time Testing

To measure the execution, or performance, time of the analog circuits, the cursors are placed at the start of the output waveform and at the end of the transient response. This measurement reveals the amount of time it takes to begin generating the output wave that gives the solution(s) to the input algorithm. The testing methodology for the analog execution was the following:

1. Connect the power rails accordingly.
2. Assemble PCB to represent desired equation.
3. Connect inputs and outputs of each PCB to each other to properly represent the equation.
4. Connect a scope probe at the input and output. Ensure that the waveform generator is not turned on for the input.
5. Set the trigger of the scope to trigger off of source one (the input).
6. Turn on the input and capture the transient response.
7. Measure the time between the input turning on and the output waveform to appear.

The oscilloscope trace shown in Figure 5-24 shows the transient response of the second-order differential equation that was measured to obtain the performance time of the cascaded analog circuits. For this tested second-order differential equation, $y'' - 10y' - y$, the execution time was measured as 82.000 μs .

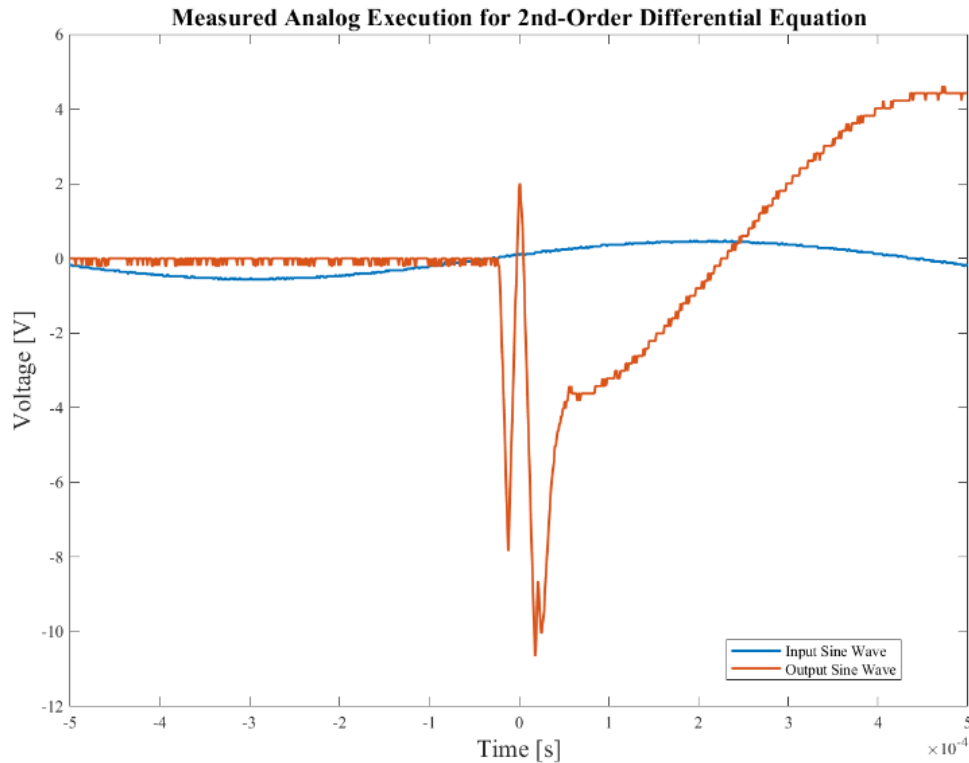


Figure 5-24. Oscilloscope Trace of the Execution Time for Test Equation, $y'' - 10y' - y$

Each of the execution times for the individual computational circuits was measured, and the measured times can be seen in Table 5-12. The table also displays the digitally measured execution times for mathematical operations and functions executed in MATLAB®. The selected mathematical operations were tested in a digital medium like MATLAB® to use as a benchmark for the analog circuits' execution times. To obtain the digital execution times, the “tic toc” function was used, measuring the time it takes to generate the output or output waveform. Each test case's execution time was measured 100 times and averaged to obtain the most accurate time measurement as there is typically variability with digital execution times.

Each of the measured digital execution times was in the microsecond range, however, functions involving calculus like integral(), diff(), trapz(), cumtrapz(), etc. have much longer execution times, entering the thousands of microseconds, due to the complexity of those operations. As expected, the analog circuits' execution times were much smaller compared to the digital execution times, demonstrating that these analog circuits can perform the same mathematical functions and operations with more efficiency.

Table 5-12. MATLAB® Function Execution Time Results

Function Tested	Average Digital Execution Time [μs]	Analog Execution Time [μs]	Digital to Analog Execution Time Ratio
$0.5\sin(2\pi 1000t) + 0.35\sin(2\pi 1000t)$	4.712	3.200	1.473
$0.05\sin(2\pi 1000t) + 0.035\sin(2\pi 1000t)$	10.160	7.500	1.355
$-1.5\sin(2\pi 1000t) + 1\sin(2\pi 1000t)$	33.784	5.400	6.256
$\sin(2\pi 81000t) \times 2.2$	11.060	5.240	2.111
$0.5\sin(2\pi 24000t) \times 6.45$	12.132	12.000	1.011
$0.5\sin(2\pi 24000t) \times 9.3$	10.793	8.650	1.248
$\frac{5\sin(2\pi 1000t)}{4.9}$	8.889	4.000	2.222
$\frac{10\operatorname{sgn}(\sin(2\pi 1000t))}{2}$	10.332	3.800	2.719
$\frac{10\left(2\left 2\left(\frac{t}{p}-\lfloor\frac{t}{p}+\frac{1}{2}\rfloor\right)\right -1\right)}{11}$	8.702	4.200	2.072
$\frac{d}{dt}(0.5\sin(t))$	6964.116	98.382	70.786
$\frac{d}{dt}(\operatorname{sgn}(\sin(2\pi 1000t)))$	6497.709	342.382	18.978
$\frac{d}{dt}\left(2\left 2\left(\frac{t}{p}-\lfloor\frac{t}{p}+\frac{1}{2}\rfloor\right)\right -1\right)$	6143.135	113.500	54.125
$\int_{-\infty}^{+\infty} \sin(t)dt$	4577.318	99.000	46.236
$\int_{-\infty}^{+\infty} \operatorname{sgn}(\sin(2\pi 1000t))dt$	5824.736	107.000	54.437
$\int_{-\infty}^{+\infty} \left(2\left 2\left(\frac{t}{p}-\lfloor\frac{t}{p}+\frac{1}{2}\rfloor\right)\right -1\right)dt$	6917.272	106.000	65.257
$-1 \times \sin(t)$	736.565	107.600	6.845
$-1.5 \times \operatorname{sgn}(\sin(2\pi 1000t))$	1784.996	167.000	10.689
$-2 \times \left(2\left 2\left(\frac{t}{p}-\lfloor\frac{t}{p}+\frac{1}{2}\rfloor\right)\right -1\right)$	2356.305	108.000	21.818
$y'' + 10y' + y$	2283.542	82.000	27.8481

The aforementioned second-order differential equation was also tested in MATLAB®. As the execution time resulted in approximately 2283.542 μs, the averaged digital execution time was approximately 27.8481 times longer than that of the measured analog execution time for the same equation.

Figure 5-25 shows a graphical representation of the analog execution times versus the averaged digital execution times for the basic arithmetic test cases that include operations like addition, subtraction, multiplication, and division. As expected, the execution times for these test cases, both digital and analog, were on the tens of microseconds. However, analog was faster for each computational test case, though much closer in value to digital compared to that of the more complex calculus functions.

Analog vs. Digital Execution Time (Basic Arithmetic)

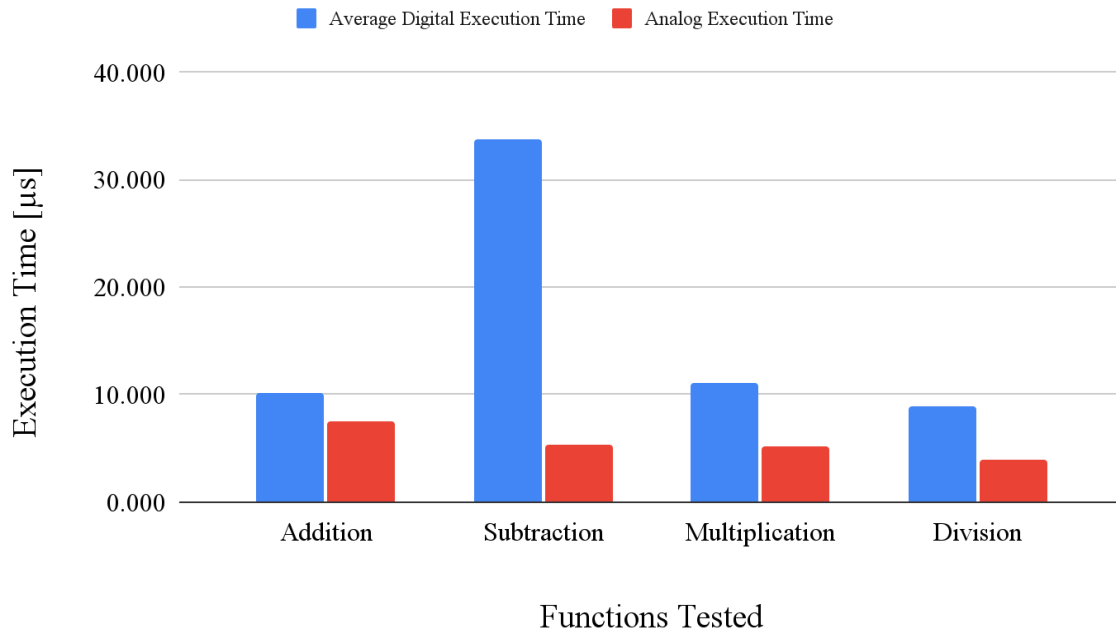
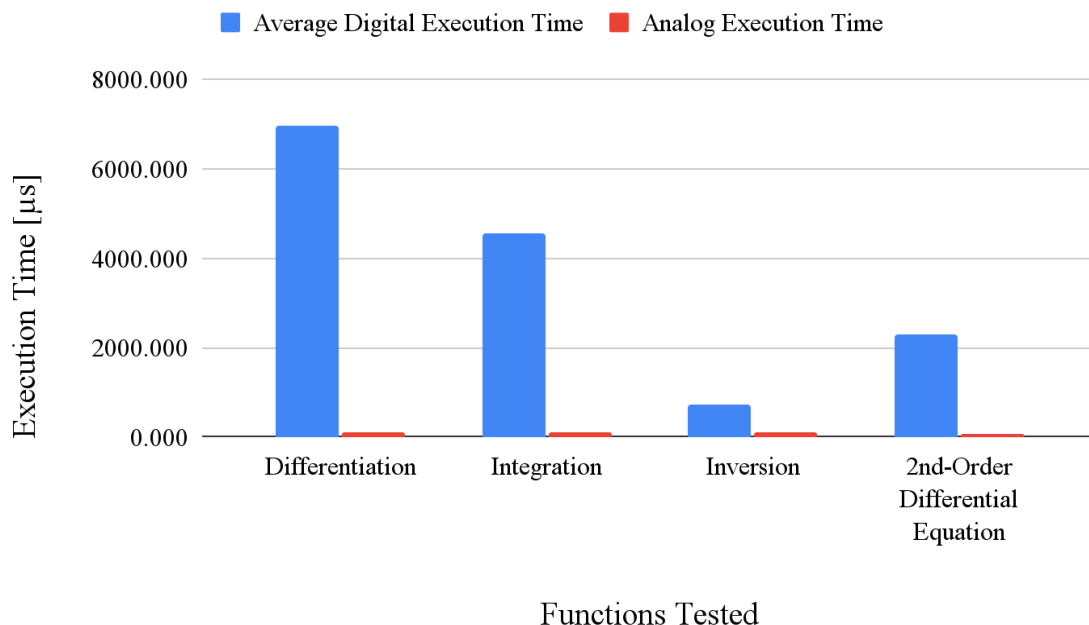


Figure 5-25. Graphical Representation of the MATLAB® Execution Time and PCB Execution Time (Basic Arithmetic Circuits)

Figure 5-26 shows a graphical representation of the analog versus averaged digital execution times for the more complex, calculus-based functions such as differentiation, integration, and inversion as well as differential equations. Contrasting the performance of the basic arithmetic computations, the averaged digital execution times were each on the scale of thousands of microseconds. The analog computations, however, were much faster, none of them exceeding 100 μ s.

Analog vs. Digital Execution Time (Calculus)



Function 5-26. Graphical Representation of the MATLAB® Execution Time and PCB Execution Time (Basic Calculus Circuits)

From the analog versus averaged digital execution timing results, it can be concluded that for basic arithmetic computations, the performance time is on a similar scale with the analog systems executing slightly faster than a digital system for each operation. However, for more complex, calculus-based operations, and thus algorithms, analog systems can perform much faster than that of digital systems while maintaining fairly low error in the output (solution).

Calculation Accuracy

The PCB circuitry represents the output of the equation $y'' - 10y' - y$ as a real-time signal. To find a solution, one must simply set the equation to the desired value. For this example, let $y'' - 10y' - y = 3$, and relating this to the PCB output means that at every instance 3V occurs, which is a solution for this differential equation. Placing a y-cursor at 3V in the waveform will give every instance of the time it occurs. The PCB testing showed a time value of 392 μs for the first instance of 3V. This value then has 82 μs subtracted from it to account for execution time. From this, the final result is 310 μs .

Using MATLAB®, the X value of the sinusoidal signals used to represent $y(x)$ is solved to be at around 1.875 when the equation is set to 3. The following relation is then used to calculate the time value

$$\begin{aligned} \text{Let } x &= 2000\pi t \\ 1.875 &= 2000\pi t \\ \frac{1.875}{2000\pi} &= t \\ t &= 294.52 \mu\text{s} \end{aligned}$$

The true solution to this differential equation is 294.52 μs while the PCB gives a value of 310 μs . The percent error is 4.99%, which verifies that the design and implementation of this project are working as intended.

VI. CONCLUSION

Analog computation is shown to be a viable option for computing algorithms due to the speed and power consumption. When the PCBs and test algorithm were constructed, they were shown to be around 27 times faster than MATLAB® while only drawing 0.180 W of power as measured for the power source. As shown throughout testing, the accuracy is mainly determined by the component tolerances and hardware. Once the proper hardware is designed and constructed, any algorithm can be constructed as a circuit. This saves on time and power when compared to the speed and power draw of digital computational programs, such as MATLAB®, as they can take long periods of time to solve. To further improve the design of this project, smaller component tolerances such as 1% can be used in order to achieve higher accuracy. Another improvement that can be made is choosing higher-quality op-amps so a large range of frequencies can be handled due to a better gain bandwidth product.

VII. BIBLIOGRAPHY

- [1] A. S. Jackson, *Analog Computation*. McGraw-Hill Book Company, INC., 1960.
- [2] J. N. Warfield, *Introduction to Electronic Analog Computers*. Prentice-Hall, INC., 1959.
- [3] I. Nagel, L. Fabre, R. Cherkaoui, and M. Kayal, "High-speed power system stability simulation using analog computation: Systematic error analysis," in *Proceedings of the 17th International Conference Mixed Design of Integrated Circuits and Systems - MIXDES 2010*, 2010, pp. 514– 518.
- [4] I. Nagel, L. Fabre, R. Cherkaoui, and M. Kayal, "Microelectronic, high-speed data processing calculators for power system analysis: Comparison," in *Proceedings of the 8th IEEE International NEWCAS Conference 2010*, 2010, pp. 137–140. doi: 10.1109/NEWCAS.2010.5603732.
- [5] *Analog computing for the future*. [Online]. Available: <https://the-analog-thing.org/>.
- [6] *Open and closed source software*. [Online]. Available: https://isaaccomputerscience.org/concepts/sys_os_open_closed_source?examBoard=all&stage=all.
- [7] *Pspice user's guide*, 60-30-632, Second Edition, Cadence PCB Systems Division, 2000. [Online]. Available: https://www.seas.upenn.edu/~jan/spice/PSpice_UserguideOrCAD.pdf.
- [8] V. Prodanov, "Statistical Exploration of Circuit Sensitivities using LTSpice," in EE 460 Lecture.
- [9] S. Clowers, "Understanding how solder and coatings impact PCB shelf life," Epec's Blog, <https://blog.epectec.com/understanding-how-solder-and-coatings-impact-pcb-shelf-life>.
- [10] *Markup Calculator*. FreshBooks. [Online]. Available: <https://www.freshbooks.com/tools/markup-calculator>.
- [11] *LM741 Operational Amplifier Datasheet*, Texas Instruments. [Online]. Available: <https://www.ti.com/lit/ds/symlink/lm741.pdf>.
- [12] A. S. Deese and C. O. Nwankpa, "Emulation of power system load dynamic behavior through reconfigurable analog circuits," 2006 IEEE International Symposium on Circuits and Systems, 2006, pp. 4 pp.-1694, doi: 10.1109/ISCAS.2006.1692929.
- [13] A. S. Deese and C. O. Nwankpa, "Circuit Theoretical Analysis of Reconfigurable Analog Load Emulation Circuit," 2007 IEEE Lausanne Power Tech, 2007, pp. 737-742, doi: 10.1109/PCT.2007.4538407.
- [14] A. Deese and C. O. Nwankpa, "Utilization of FPAA technology for emulation of multi-scale power system dynamics in smart grids," 2012 IEEE Power and Energy Society General Meeting, 2012, pp. 1-1, doi: 10.1109/PESGM.2012.6345751.
- [15] A. Deese, J. C. Jiménez, J. Berardino and C. O. Nwankpa, "Hardware prototype to emulate the dynamics of power system generators with field programmable analog arrays," Proceedings of 2010 IEEE International Symposium on Circuits and Systems, 2010, pp. 2314-2317, doi: 10.1109/ISCAS.2010.5537072.
- [16] C. Nwankpa, A. Deese, Qingyan Liu, A. S. Leger, J. Yakaski and N. York, "Power system on a chip (PSoC): analog emulation for power system applications," 2006 IEEE Power Engineering Society General Meeting, 2006, pp. 13 pp.-, doi: 10.1109/PES.2006.1709363.
- [17] L. Fabre, I. Nagel, R. Cherkaoui and M. Kayal, "High-speed, mixed-signal emulation for power system dynamic analysis," 2009 IEEE Bucharest PowerTech, 2009, pp. 1-8, doi: 10.1109/PTC.2009.5281884.
- [18] L. Fabre, I. Nagel, C. Meinen, R. Cherkaoui and M. Kayal, "A mixed-signal platform dedicated to power system dynamic computation," 2011 IEEE International Symposium of Circuits and Systems (ISCAS), 2011, pp. 1860-1863, doi: 10.1109/ISCAS.2011.5937949.
- [19] L. Fabre, G. Lanz, I. Nagel, F. L. Conte, R. Cherkaoui and M. Kayal, "An ultra high-speed, mixed-signal emulator for solving power system dynamic equations," 2011 IEEE 9th International New Circuits and systems conference, 2011, pp. 374-377, doi: 10.1109/NEWCAS.2011.5981248.

- [20] R. Billmeyer, M. Lu, B. Johnson and S. Dhople, "Modeling and Simulation of Power-Electronic Inverters in Analog Electronic Circuit Simulators," 2021 IEEE International Symposium on Circuits and Systems (ISCAS), 2021, pp. 1-5, doi: 10.1109/ISCAS51556.2021.9401268.
- [21] L. Fabre et al., "A 3D architecture platform dedicated to high-speed computation for power system," 2013 IEEE Grenoble Conference, 2013, pp. 1-6, doi: 10.1109/PTC.2013.6652419.
- [22] L. Fabre et al., "An Ultra-High Speed Emulator Dedicated to Power System Dynamics Computation Based on a Mixed-Signal Hardware Platform," in IEEE Transactions on Power Systems, vol. 28, no. 4, pp. 4228-4236, Nov. 2013, doi: 10.1109/TPWRS.2013.2259186.
- [23] B. Chen, G. Pin, W. M. Ng, P. Li, T. Parisini and S. -Y. R. Hui, "Online Detection of Fundamental and Interharmonics in AC Mains for Parallel Operation of Multiple Grid-Connected Power Converters," in IEEE Transactions on Power Electronics, vol. 33, no. 11, pp. 9318-9330, Nov. 2018, doi: 10.1109/TPEL.2017.2789303.
- [24] R. Fried, R. S. Cherkaoui, C. C. Enz, A. Germond and E. A. Vittoz, "Approaches for analog VLSI simulation of the transient stability of large power networks," in IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, vol. 46, no. 10, pp. 1249-1263, Oct. 1999, doi: 10.1109/81.795838.
- [25] J. Cohn, "Transitions from Analog to Digital Computing in Electric Power Systems," in IEEE Annals of the History of Computing, vol. 37, no. 3, pp. 32-43, July-Sept. 2015, doi: 10.1109/MAHC.2015.37.
- [26] M. Kayal, R. Cherkaoui, I. Nagel, L. Fabre, F. Emery and B. Rey, "Toward a Power System Emulation using Analog Microelectronics Solid State Circuits," 2007 IEEE Lausanne Power Tech, 2007, pp. 726-730, doi: 10.1109/PCT.2007.4538405.
- [27] I. Nagel, L. Fabre, R. Cherkaoui and M. Kayal, "Microelectronic, high-speed data processing calculators for power system analysis: Comparison," Proceedings of the 8th IEEE International NEWCAS Conference 2010, 2010, pp. 137-140, doi: 10.1109/NEWCAS.2010.5603732.
- [28] Y. Huang, N. Guo, M. Seok, Y. Tsividis, K. Mandli and S. Sethumadhavan, "Hybrid Analog-Digital Solution of Nonlinear Partial Differential Equations," 2017 50th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO), 2017, pp. 665-678.
- [29] I. Nagel, L. Fabre, M. Pastre, F. ç. o. Krummenacher, R. Cherkaoui and M. Kayal, "High-Speed Power System Transient Stability Simulation Using Highly Dedicated Hardware," in IEEE Transactions on Power Systems, vol. 28, no. 4, pp. 4218-4227, Nov. 2013, doi: 10.1109/TPWRS.2013.2259185.
- [30] I. Nagel, L. Fabre, R. Cherkaoui and M. Kayal, "High-speed power system stability simulation using analog computation: Systematic error analysis," Proceedings of the 17th International Conference Mixed Design of Integrated Circuits and Systems - MIXDES 2010, 2010, pp. 514-518.
- [31] J. Hasler and A. Natarajan, "Continuous-Time, Configurable Analog Linear System Solutions With Transconductance Amplifiers," in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 68, no. 2, pp. 765-775, Feb. 2021, doi: 10.1109/TCSI.2020.3039763.

IX. APPENDICES

A. Analysis of Senior Project Design

Project Title: Analog Circuits for Computing

Student's Name: Lauren Chun

Student's Signature:



Dillon Nguyen

Student's Signature:



Nicole Pickett

Student's Signature:



Advisor's Name: Dr. Jason Poon

Advisor's Initials:



Date: 6/5/23

Summary of Functional Requirements

The analog circuits for computing are able to perform mathematical operations such as addition, subtraction, multiplication, division, differentiation, and integration, which can also be applied to solve systems of equations and differential equations. To execute the mathematical computations, the desired algorithms are decomposed and the appropriate mathematical operations needed to solve are applied. From the required mathematical operation circuits, a SPICE netlist and corresponding PCB layout needed to derive the solution to the algorithm can be generated based on the modeled algorithm. After manufacturing the appropriate PCBs, the results of the mathematical computations are produced with no greater than 10% error and a computation time less than that of the same operations performed using an equivalent digital system such as MATLAB®.

Primary Constraints

The primary constraint for planning the design of the computational analog circuits system typical for most design projects is the maximum 10% error constraint. When actually designing each computational circuit, the differentiator and integrator systems created the most challenges in implementation. During the LTspice design phase, it was difficult to find a resistor-capacitor combination that would result in an appropriate time constant for the circuit while maintaining a gain that would not significantly amplify the output and result in clipping. Switching to the improved differentiator configuration, which involves adding an input resistor and a feedback capacitor to mitigate gain increase and account for attenuation at higher frequencies. The integrator also induced challenges due to the limitation that unity gain cannot be maintained while obtaining the output. Adding a potentiometer (resistive divider) at the output of the integrator helped provide the desired gain of the system while maintaining the integrity of the output's phase shift.

One of the final major challenges was determining the most efficient and effective method of measuring and comparing the execution times of the analog systems to a comparable digital computational system. After much experimentation with microcontrollers and oscilloscope capabilities, the analog execution times were obtained using the "Single" feature on the oscilloscope, allowing for easy measurement of the elapsed time between the signal entering the input of the PCB to the output signal leaving the output of the PCB. Resolving the challenge of obtaining digital computation times, MATLAB® script was generated and measured execution times using the "tic toc" function.

Economic

There is power consumption that is used towards all phases of the project's lifecycle. The design and simulation phase requires power consumption as design and simulation software via computers was used to research and design each of the computational circuits available in the library. The testing and manufacturing phase also had slight power consumption as that phase required a power source to power the circuits and perform the extensive breadboard and PCB testing necessary to validate each circuit's functionality and compatibility with each other. This power consumption provides demand to power providers.

The physical instruments used during the project's lifecycle were used primarily in the testing and manufacturing phase of the project. These aforementioned instruments include the Rigol Technologies power supply, Keysight Technologies function generator, and Keysight Technologies oscilloscope as well as various soldering equipment used to attach the components to the PCBs. The use of these equipment supplies the demand for equipment like these and economically benefits the companies that provide these instruments.

The external resources used in the project include the use of design and simulation programs such as LTspice, KiCAD, and MATLAB®, which provide business to them and further fuels the demand for design and simulation programs such as these. Furthermore, to construct the physical circuits and manufacture the PCBs, companies such as DigiKey and JLCPCB were used as component and PCB providers. Providing business to their companies as well as the component manufacturers that use DigiKey as a platform for their products also fuels the demand for circuit components and the PCB manufacturing industry.

The use of physical components throughout the manufacturing and testing phase of the project contributes to the need for components and PCBs. As these components are produced from natural materials such as metals, metal alloys, ceramics, polymers, and other materials that come from natural harvesting, the demand for those resources is sustained.

The overall materials cost for the product is minimal as most of the main circuit components such as resistors, capacitors, and ICs are already previously owned by the product designers. The full cost breakdown for the project can be seen in Table B-1 which shows the Bill of Materials. Factors such as materials cost and human labor can be considered with the assumption that the starting salary for an electrical engineer is approximately \$90,000. The manufacturing cost for the PCBs totaled \$115.09, and the total cost of the project materials, manufacturing, soldering, and testing, was \$363.15. The original estimation of project costs is in Table A-1 below. The customer would be responsible for covering the manufacturing costs once they have created the model for their algorithm.

Table A-1. Initial Cost Estimates for the Analog Circuits for Computing Project

Parts	Part Cost	Comments	Starting Engineer Salary	Factors	
Resistors (±5%)	\$0.10	Students have a variety of resistors and capacitor, but for additional or specific components they must be purchased	From research and personal references, the starting salary is \$90,000 . 3 Group Members, 10 hrs / week per person	People	3
Capacitors (±10%)	\$0.50 to \$2.00			Hours	10
OpAmp (LM741)	\$0.90	Both op-amp models are needed to either save on space or computation time when necessary		Wage	\$43.27
OpAmp (LM358P)	\$0.44			Duration (Weeks)	20
PCB Manufacturing	\$250	This is a maximum provided range for PCB manufacturing.		COST	\$25,962.00
IC Chips	\$0.50 to \$30	Ranges from \$0.50 for simple IC's to \$30 for complex IC's like an analog multiplier			
Solder	\$20	A spool of soldering contains enough for applications			
Soldering Iron	\$0	Students have a soldering iron			
Solder Paste	\$20	A tub of solder paste is enough for applications			
Total Parts Cost	\$291.44				
TOTAL COST	\$26,285.44				

One of the main benefits of the project is its use of open-source software for circuit design and simulation. By using these open-source materials, any cost towards circuit design and simulation was mitigated with the only remaining costs towards manufacturing. The manufacturing costs include the cost of components used to construct the physical computational circuits and manufacture the actual PCBs (if the customer is outsourcing). The actual costs of the manufacturing process are discussed in the following section “If Manufactured on a Commercial Basis”.

The total design time for the customer’s desired algorithm depends on the complexity of their algorithm and what library circuits they decide to include in their system. Once at the manufacturing stage, should they choose to outsource, the manufacturing time is dependent on the company they choose to use for manufacturing. This manufacturing time can take anywhere from approximately a week to two weeks or more. Once the customer manufactures their PCBs, if they chose not to have the manufacturer also solder the corresponding components, the time it takes to solder the components on will need to be incorporated into the final production time. When the PCBs have been completely manufactured and soldered, they can last, typically for 50-70 years if kept in good condition properly [9]. Little to no maintenance is required after manufacturing as long as only the proper, low-voltage inputs are sent through the circuit and the PCBs are kept in an environment that will not harm the electronics.

The initial project timeline is visualized in Figures A-1 through A-3 below with the project lifespan as 30 weeks. The project timeline spans phases such as the initial project planning, conceptual design, and manufacturing. The final list of milestones and task completions can be seen in Table B-2 of Appendix B.

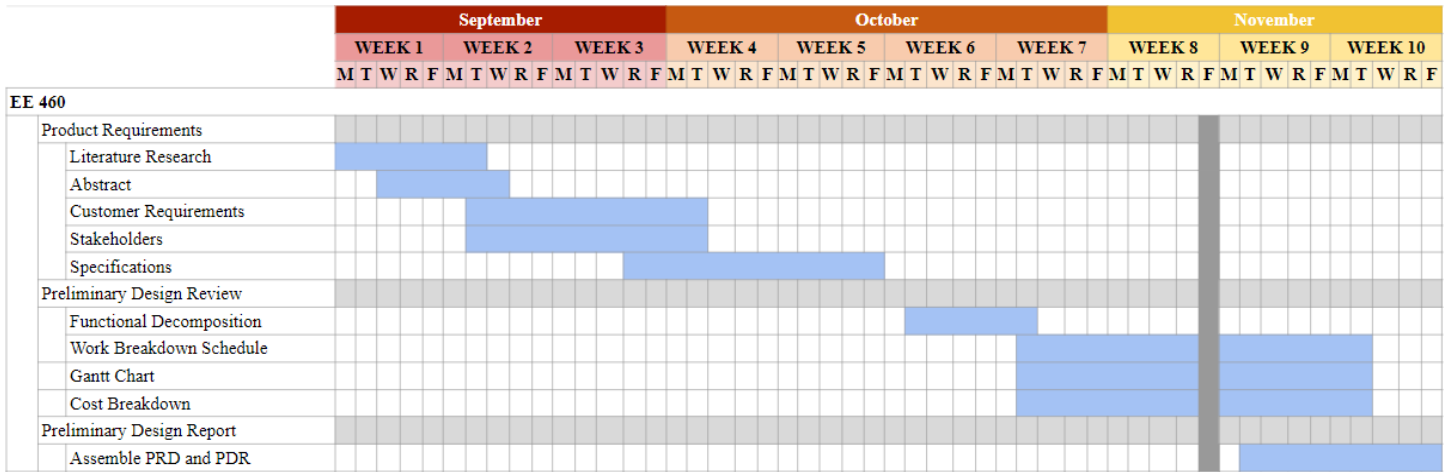


Figure A-1. Gantt Chart for Analog Circuits for Computing Project (Project Planning Phase)

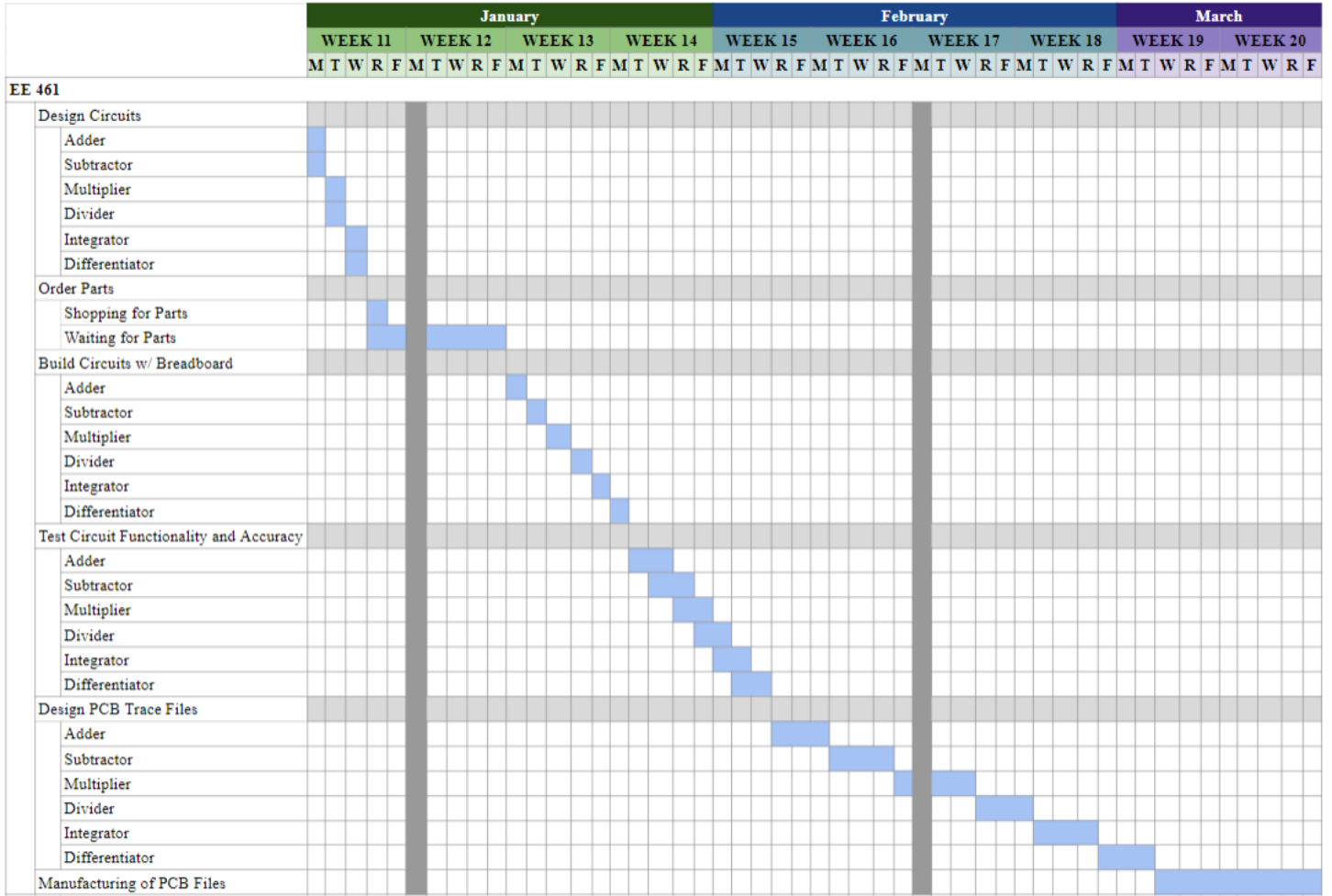


Figure A-2. Gantt Chart for the Analog Circuits for Computing Project (Conceptual Design and Simulation Phase)

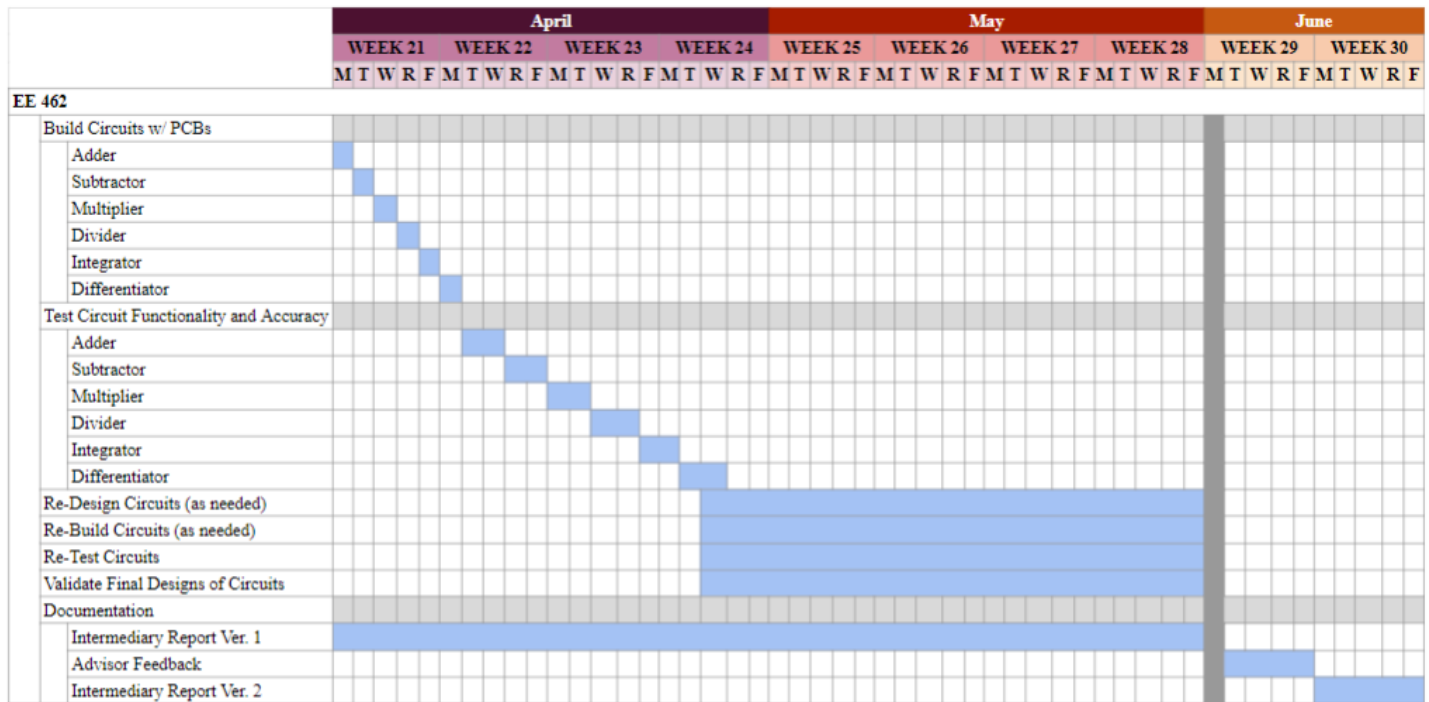


Figure A-3. Gantt Chart for the Analog Circuits for Computing Project (Manufacturing and Testing Phase)

There is room for improvement and expansion on this project that can be implemented, which includes using components of higher quality and tolerance to improve the accuracy of the analog circuits. Another area of expansion is adding more mathematical functions to the library of circuits such as analog multiplication, analog division, exponents, square roots, matrices, and many other operations. Having more available mathematical functions in the computational circuits library allows for a wider range of possible, solvable algorithms that can be applied to the system.

If Manufactured on a Commercial Basis

The customer population has much variation as it includes anyone who has the need to perform large-scale and/or time-consuming computations. The size of this population has a direct correlation to the number of analog computational systems that would be sold per year. Assuming that the analog computational system will be primarily used by the power industries, it is appropriate to estimate 1,000 units sold per year.

The manufacturing cost will vary per customer as it is dependent on the algorithm they plan to model. The manufacturing cost depends on which computational circuits the customer decides to use from the provided library as well as how many of each computational circuit are required to model their desired algorithm. The total manufacturing cost for each PCB will also vary depending on if they outsource their PCBs and soldering for components or if they manufacture everything themselves. If they decide to solder manually, the manufacturing cost will also depend on which components they use in their final system.

If using the exact materials, designs, components, and manufacturers used to produce the example PCBs for this project, the approximate production cost for each PCB will be as follows:

Voltage Adder:	\$5.04
Inverting Voltage Multiplier:	\$5.01
Non-Inverting Multiplier:	\$5.01
Resistive Divider:	\$13.08
Differentiator:	\$3.08
Integrator:	\$7.60

The following production costs listed for each computational circuit do not include the shipping and tax fees that come with outsourcing PCB manufacturing and will be dependent on the customer's location and shipping requirements. Assuming the starting salary for an electrical engineer is \$90,000 and the project duration is 30 weeks, the total labor cost is estimated at around \$25,962.

To create a profit, especially in large-scale production, retail markup is essential. Marking up the value of the product ensures fair wages and other essential services such as social security and any provided insurance to those designing and manufacturing the product. Retail markup also ensures the success of the overall project, product, and its manufacturers in the long run. A typical retail markup for businesses is 50% [10], and thus, would be an appropriate price indicator for the analog computational circuits system. While the total manufacturing cost is heavily dependent on the algorithm the customer chooses to model, assuming that the customer will have all circuits in the library at their disposal and that the 50% markup applies, an appropriate market price would be \$60.00.

Again, the customer base is very broad as well as diverse and the production cost is very dependent on the customer's desired algorithm. Given the estimated market price of \$60.00 and the estimated 1,000 units sold per year, the estimated profit per year would be \$60,000, not including other fixed and variable costs involved in the manufacturing process. Once the user obtains access to the analog computational circuits library, the modeling of their desired algorithm is free through the use of open-source circuit design and simulation programs. The only cost to operate the analog computational circuits system will come from manufacturing should they choose to proceed with that step, which the costs are elaborated on earlier.

Environmental

The product life cycle begins at the production stage where the required materials such as metals, metal alloys, and plastics are harvested from the earth and developed. This harvesting and developing process come with a carbon footprint that negatively impacts the air, the earth they harvest, and the health of the people, directly and indirectly, involved in the process. There is also a carbon footprint involved in the manufacturing of the individual circuit components needed to build and fabricate the circuits themselves like the air pollution and waste emitted and produced by factories. The carbon emissions and waste produced by the PCB manufacturing process further contribute to these negative environmental impacts. Furthermore, the energy required to run the harvesting and developing process, the manufacturing factories, and the computers and programs to design the product may be produced by nonrenewable and unclean energy sources that contribute to environmental consequences as well. The negative impacts of this manufacturing process include health issues for harvesters and manufacturers, air pollution from the use of toxic chemicals, and ground and water pollution from the toxic chemicals released during the harvesting and production process. Air, ground, and water pollution, in turn, can cause harm to other animal species and people that use those resources for consumption and daily living.

The analog circuits for computing will be designed to mitigate these negative environmental impacts by using variable passive components if possible. This will promote reusable circuits as PCBs and other circuit components do not recycle well and can be detrimental to the environment. Designing and building circuits with these consequences in mind is vital as every part of the project and product life cycle has some environmental effect; sometimes even the smallest actions have the greatest impacts. Promoting sustainability and environmental preservation is essential in the electrical engineering field as resources are becoming more limited and technology and society are advancing faster than ever before. The analog circuits for computing project aims to contribute to this advancing society, but also prioritize prolonging the environment as much as possible.

Manufacturability

From the computational analog circuits system, the customer will receive an open-source SPICE netlist that will help generate PCB files to manufacture a model for and solve their desired algorithm. The netlist and PCB files will be generated using computer simulation programs, thus only a computer is required for them during the design process. Should they decide to manually manufacture the PCBs and solder the necessary

components, the customer will be responsible for obtaining the components and equipment necessary to do so. The materials required to manufacture and construct the analog circuits system are metals like copper, aluminum, iron, and metal alloys as well as other nonmetals like silicon, fiberglass, epoxy laminate, and Teflon. These metals and nonmetals are the materials typically found in circuit components like resistors, capacitors, and ICs as well as PCBs. Component tolerances are important in the computational analog circuits as they impact the behavior and accuracy of the system output. As accuracy is key for the anticipated power grid simulations application. It is vital that components with high tolerance and quality are selected to ensure proper functionality that meets the customer's standards and expectations. Since the analog circuits should be designed with higher-quality components, if it is not an offered component and/or option by the manufacturer, the product designers will order the necessary components and solder the PCBs manually.

Sustainability

As the final product, if the customer decides to continue with manufacturing, is a system of PCBs, the only challenge with maintaining them would be to ensure that the circuits are kept in a safe place as they can be delicate. Another measure to maintain the completed PCB system to ensure only the appropriate inputs are sent through the circuits; the current computational circuits are designed for low-voltage inputs, thus any high-voltage inputs would cause damage to the components as well as the PCB itself.

The open-source aspect of the product design process promotes accessibility and ease of testing without the need to manufacture circuits (breadboard or PCBs), thus eliminating the carbon footprint of using physical circuits to simulate designs. The power consumption from running the open-source simulations, however, is a considered negative environmental impact. The manufacturing of PCBs needed to create the analog circuits contains non-recyclable material and will be manufactured using toxic chemicals, which negatively impact the environment. The carbon footprint from shipping the PCBs as well as the solder waste and fumes from soldering components onto the PCBs (manufactured or at home) further contribute to the negative impacts on the environment from the product manufacturing process. Furthermore, if variable passive components are used in the design, the library of circuits will allow the PCBs manufactured to be reused. The only positive economic impacts include providing business for the circuit design and simulation programs used as well as the PCB manufacturing and component distribution industry. Having an analog system that can perform mathematical computations more efficiently than that of an equivalent digital system, especially in the power grid simulations field, can have great positive societal impacts depending on the product's application.

Some upgrades that can improve the design of the final product would be the use of higher-quality components that use much lower tolerances. Using components with lower tolerances will reduce the amount of error present in the output (solution), thus "upgrading" the final system the customer models. The only challenges posed to performing upgrades like these are the cost of the higher-quality, lower-tolerance components. If customers are able to afford resistors, capacitors, potentiometers, and op-amps with higher performance capabilities and tolerance, then it can serve as a significant improvement to their overall final system.

Ethical

The moral dilemma lies in that the product designers want the PCBs manufactured at the lowest expense, which means manufacturing PCBs internationally. Another moral dilemma lies in that time can be saved and quality can be improved if the PCB manufacturers solder on the components as well, but more risks and health implications come with that as the solder fumes and waste can be dangerous to the human body, especially to those who are constantly exposed to it.

Despite the negative environmental and health implications that are involved with these dilemmas and the manufacturing process, all cannons within the IEEE Code of Ethics are abided by. This code of ethics is highly considered to ensure that, not only is the product fabricated with high-quality materials for accurate results, but with as many negative sustainability impacts mitigated as possible as well. Furthermore, the standards held by the IEEE Code of Ethics also help to ensure the safety of those involved in the design and

manufacturing process, thus following these cannons can mitigate the safety risks as well. The computational analog circuits will be designed intending to improve large-scale, time-consuming simulation systems that provide a better alternative to current digital systems; however, the application of such a system will, ultimately, be decided by the customer regardless of if the application is ethical or unethical.

Health and Safety

The computational analog circuits should not require or generate any input or output within the high voltage range, thus there is minimal risk of injury. The system was designed through simulation tools and thoroughly tested before implementation on breadboards and PCBs. These tests included checks to ensure the power required does not enter the high-voltage range, the current distribution is regulated, and each circuit fully operates as desired.

The disposal of PCBs raises health concerns, however, regarding the toxic fumes that arise from said disposal. Those directly involved with the disposal process are constantly exposed to those toxic fumes resulting in many health implications. Understanding these consequences of the analog circuits system's use of PCBs, the customer's final, modeled algorithm should be designed to minimize the number of PCBs required for its full operation.

Social and Political

The computational analog circuits will have an emphasis on industries that require complex and large-scale simulation applications, thus stakeholders include, not only those investing in the project's success but the customers who use the product in the future. There are social implications involved with manufacturing the final product; however, the social implications that come with PCB manufacturing are not new and, unfortunately, a societal norm. Therefore, these social implications would not pose much of a risk to the stakeholders and customers.

PCB manufacturing uses non-recyclable materials, resulting in pollution emissions during the manufacturing process and during shipping. The PCB manufacturers that are used for the project are all located internationally, which implies that the pollution from the manufacturing will create health, social, and potentially even political, implications for them in the long run. Regardless of if it is small- or large-scale manufacturing for this product, there will be a carbon footprint that contributes to the degradation of the environment and the people, directly and indirectly, involved with the manufacturing process.

The project and product promote equality as all the analog circuits will be designed using open-source software. The customer will also be provided with the netlist from the open-source design and the generated PCB files so that they may redesign and simulate the analog circuits themselves as needed. The open-source aspect promotes equity among customers with the only requirement being access to technology that can run the open-source and PCB design and simulation tools. As the computational analog circuits aim to match or be more efficient than equivalent digital systems, the product allows for social improvement in areas of industries such as power grid simulations.

Development

Background knowledge in calculus, linear algebra, and circuit analysis is essential as the project involves designing circuits that carry out mathematical operations of varying complexity. These computational circuits were also designed primarily in LTspice, thus experience with this open-source software is vital. As the customer's final modeled algorithm using the computational circuits will need to be manufactured as a PCB, knowledge of PCB layout design is also required. Should the customer choose to add the components onto their PCBs manually, soldering skills are also necessary to implement their manufactured, modeled algorithm. Prior knowledge of these fundamental electrical engineering and mathematical concepts as well as experience with

circuit design, simulation, and manufacturing should have been learned from previous courses in their Electrical Engineering curriculum but can be self-taught with enough discipline and maturity.

B. Bill of Materials and Time Schedule Allocation

Table B-1 shows the bill of materials for the Analog Circuits for Computing project. Not all materials were used in the final product as some were damaged during the production process or were extra parts after the final product was manufactured. While not all materials were used in the end, the number of materials shown is the amount of each order for the project.

Table B-1. Analog Circuits for Computing Project Bill of Materials

Count	Value	Description	Size	Part Number	Manufacturer	Per Unit Cost (\$)
20	100 k Ω	Potentiometer, 10% Tolerance	12mm Shaft, 4mm Shaft Diameter 10.1mm Height, 10mm Length	251B12T104A2NB	CTS Electronic Components	\$2.56
20		LM741 Operational Amplifier	3.3mm x 9.27mm x 6.35mm	LM741CN/NOPB	Texas Instruments	\$0.90
128		8-pin Wire Wrap Female Header Pins	2.54mm	8Fx1L-254mm	Gravitech	\$0.50
20		10 SIL Horizontal Pin Header Tin	2.54mm	M20-9751046	Harwin	\$0.47
5		Voltage Adder Printed Circuit Board	1.5in x 1.5in		JLCPCB	\$2.22
5		Voltage Multiplier Printed Circuit Board	1.5in x 1.5in		JLCPCB	\$2.62
5		Resistive Divider Printed Circuit Board	1.5in x 1.5in		JLCPCB	\$2.62
5		Differentiator Printed Circuit Board	1.5in x 1.5in		JLCPCB	\$2.62
5		Integrator Printed Circuit Board	1.5in x 1.5in		JLCPCB	\$2.62
5		Inverter Printed Circuit Board	1.5in x 1.5in		JLCPCB	\$2.62

Table B-2 shows the timeline of tasks and milestones for the Analog Circuits for Computing Project. This timeline marks tasks performed from the initial project planning to the conceptual design process to the manufacturing of the final product.

Table B-2. Timeline of Tasks and Milestones for the Analog Circuits for Computing Project

Date of Completion	Task
09/27/22	Literature Review
09/28/22	Abstract
10/11/22	Customer Requirements
10/11/22	Stakeholders
10/21/22	Specifications
11/01/22	Functional Decomposition
11/29/22	Work Breakdown Schedule
11/29/22	Gantt Chart
11/29/22	Cost Breakdown
12/02/22	Assemble PRD and PDR
01/13/23	Adder/Subtractor Simulation(LTspice) Testing and Verification
01/20/23	Non-Inverting Multiplier Simulation(LTspice) Testing and Verification
01/27/23	Divider Simulation(LTspice) Testing and Verification
02/03/23	Assembled Parts List
02/10/23	Arrival of Ordered Parts
02/15/23	Differentiator Simulation(LTspice) Testing and Verification
02/22/23	Integrator Simulation(LTspice) Testing and Verification
3/9/23	Adder/Subtractor Breadboard Testing and Verification
3/9/23	Non-Inverting Multiplier Breadboard Testing and Verification
3/9/23	Divider Breadboard Testing and Verification
3/9/23	Differentiator Breadboard Testing and Verification
3/9/23	Differentiator PCB Layout
3/10/23	Connections Compatibility Simulations (Fundamental Circuits)
4/7/23	Adder/Subtractor PCB Layout
4/8/23	Non-Inverting Multiplier PCB Layout
4/9/23	Integrator Breadboard Testing and Verification
4/9/23	Divider PCB Layout
4/19/23	Integrator PCB Layout
4/19/23	PCBs Ordered
5/10/23	PCBs Soldered
5/14/23	PCBs Tested
5/19/23	Test Cases/Equations to Use for Breadboard and PCB Testing Chosen
5/24/23	PCB Execution Times Measured
5/24/23	MATLAB Script for Digital Execution Times
5/26/23	First Draft of Final Report
5/27/23	Senior Project Expo Board PDF
6/2/23	Senior Project Expo
6/5/23	Final Draft of Final Report

C. IC Location Diagram

The only IC used throughout each of the analog computational circuits is the LM741 operational amplifier. The pinout for this IC is shown in Figure C-1 below.

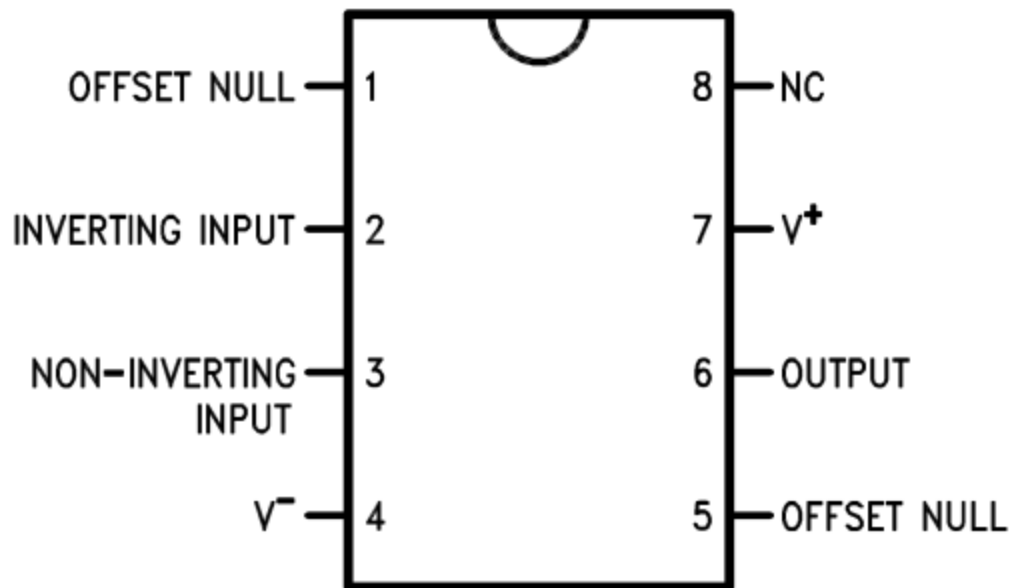


Figure C-1: LM741 Pinout Diagram [11]

D. Printed Circuit Board Artwork

Figures D-1 through D-6 show the PCB layouts generated for each individual computational circuit. Each of these PCB layouts is the final design sent out for manufacturing with JLCPCB.

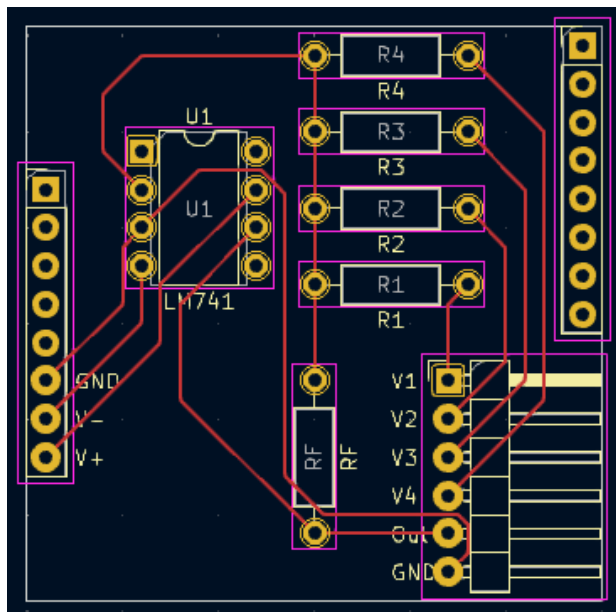


Figure D-1: Voltage Adder PCB

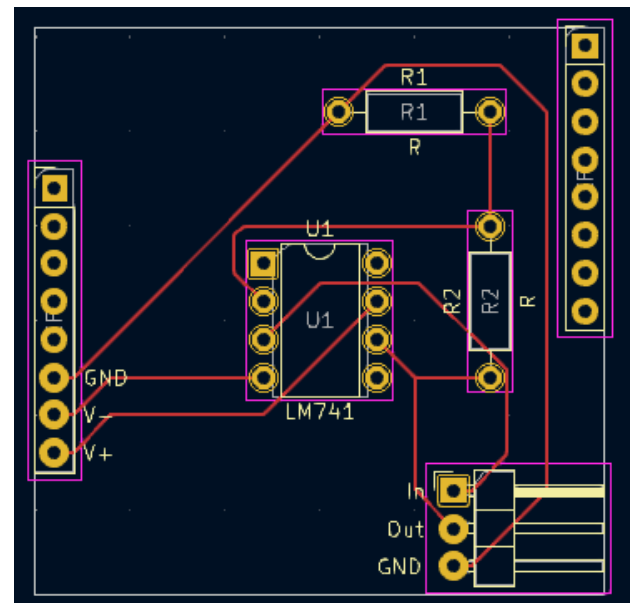


Figure D-2: Non-Inverting Voltage Multiplier PCB

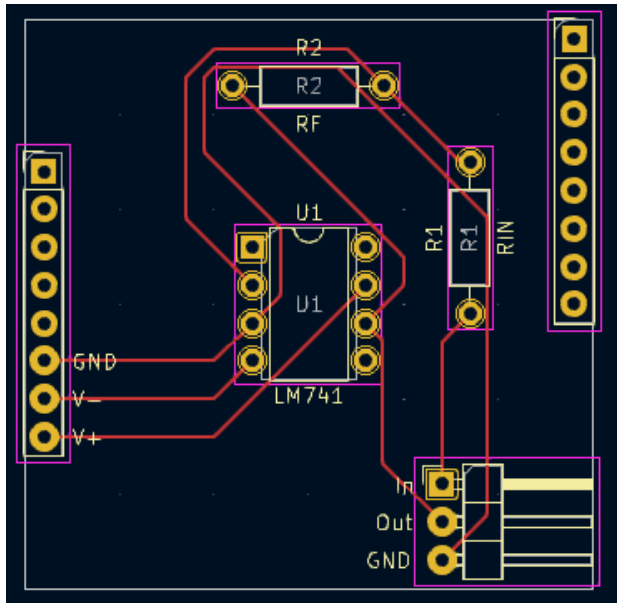


Figure D-3: Inverting Voltage Multiplier PCB

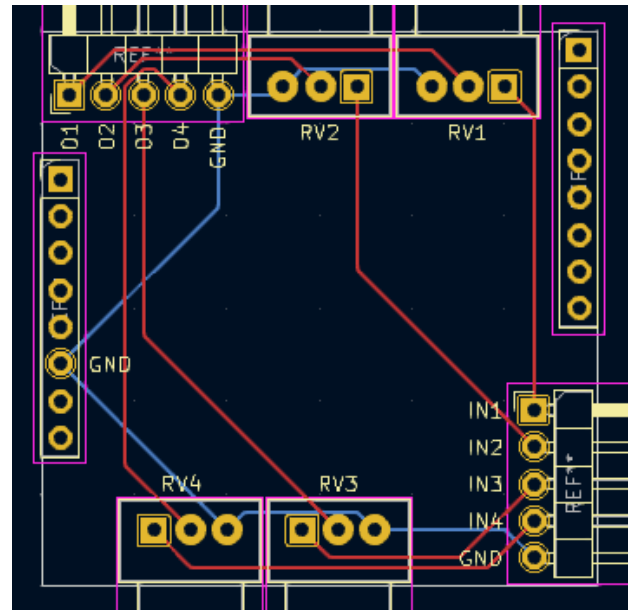


Figure D-4: Resistive Divider PCB

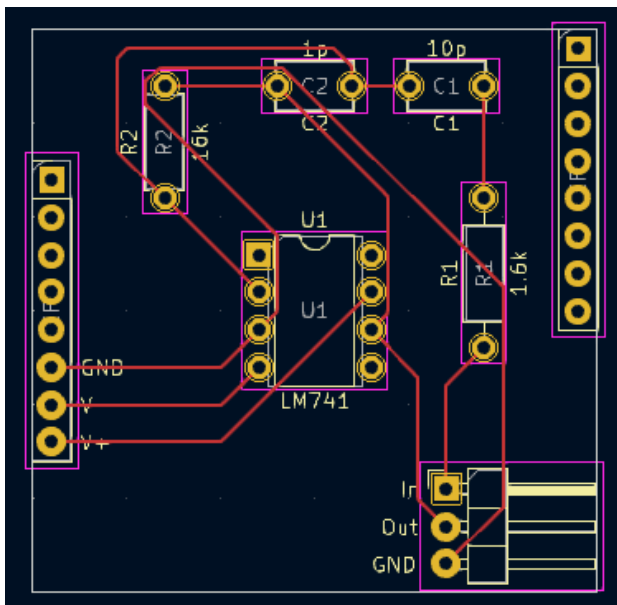


Figure D-5: Differentiator PCB

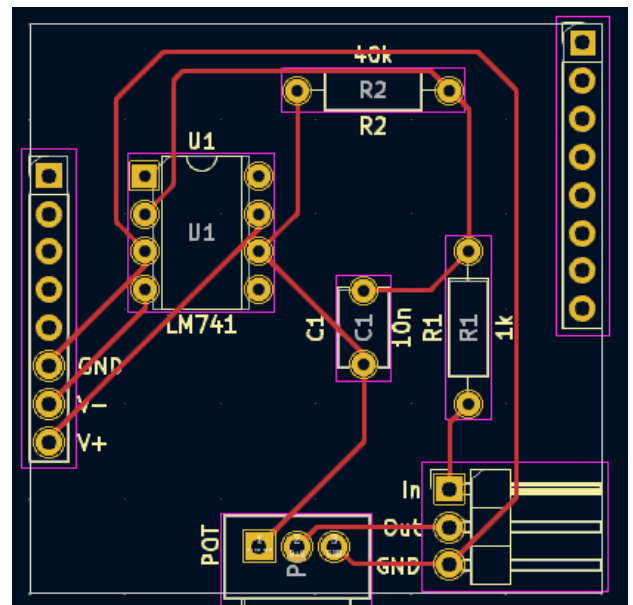


Figure D-6: Integrator PCB

E. Program Listing - MATLAB® Simulation Code

```
% -----  
% MATLAB Script for Testing Digital Computational Timing  
% EE 462 - Analog Circuits for Computing  
% Project Members: Lauren Chun, Dillon Nguyen, Nicole Pickett  
%  
% Script written by Lauren Chun  
% -----  
  
clear;  
clc;  
close;  
  
%% ADDITION AND SUBTRACTION TESTS  
  
temp_times = zeros(1,100);      % Array to store temporary execution time samples  
times_add_sub = zeros(1,3);     % Array to store final, averaged execution times for each  
test case  
  
% Basic Addition Test (DC Signal Addition)  
  
% Perform the test case 100 times, and take average execution time of the  
% 100 samples  
for i = 1:100  
    tic  
    x = 1 + 7;  
    temp_times(i) = toc;  
end  
  
add_sub_result = x;              % Save test case result  
times_add_sub(1) = mean(temp_times); % Average 100 execution time samples  
temp_times = zeros(1,100);      % Clear the temporary execution times for the next  
test case  
  
% Sinusoidal Addition (Sine Wave Addition)  
  
t = 0:(1/50000):0.005; % Make 0.1 seconds sampled every 1/1000 of a second  
sine1 = (0.05 * sin(2*pi*1000*t)); % 0.1Vpp, 1 kHz Sine Wave  
sine2 = (0.035 * sin(2*pi*1000*t)); % 0.7Vpp, 1 kHz Sine Wave  
  
% Perform the test case 100 times, and take average execution time of the  
% 100 samples  
for i = 1:100  
    tic  
    x = sine1 + sine2;  
    temp_times(i) = toc;  
end  
  
% Plot the original sine wave against result of the operation  
figure(1);  
plot(t,sine1);  
hold on  
plot(t,sine2);  
hold on  
plot(t,x);  
title("Sinusoidal Addition Waveforms");  
xlabel('Time [s]');  
ylabel('Voltage [V]');
```

```

legend("Sinusoid 1","Sinusoid 2","Sinusoid 1 + Sinusoid 2");

times_add_sub(2) = mean(temp_times);    % Average 100 execution time samples
temp_times = zeros(1,100);             % Clear the temporary execution times for the next
test case

% -----
% Basic Subtraction (DC Signal Subtraction)

% Perform the test case 100 times, and take average execution time of the
% 100 samples
for i = 1:100
    tic
    x = (-1.5 * sin(2*pi*1000*t)) + (1 * sin(2*pi*1000*t));
    temp_times(i) = toc;
end

times_add_sub(3) = mean(temp_times);    % Average 100 execution time samples
temp_times = zeros(1,100);             % Clear the temporary execution times for the next
test case

%% MULTIPLICATION TESTS

% Multiplier = 2.2

times_mult = zeros(1,3);    % Array to store final, averaged execution times

t = 0:(1/1000000):0.0001; % Make 0.1 seconds sampled every 1/1000000 of a second

% Perform the test case 100 times, and take average execution time of the
% 100 samples
for i = 1:100
    tic
    x = (sin(2*pi*81000*t)) * 2.2;
    temp_times(i) = toc;
end

times_mult(1) = mean(temp_times);    % Average 100 execution time samples
temp_times = zeros(1,100);          % Clear the temporary execution times for the next test
case

% Plot the original sine wave against result of the operation
figure(2);
plot(t,sin(2*pi*81000*t));
hold on
plot(t,x);
hold on
title("Sinusoidal Multiplication Waveforms");
xlabel('Time [s]');
ylabel('Voltage [V]');
legend("2Vpp, 81 kHz Sinusoid","(2Vpp, 81 kHz Sinusoid) x 2.2");

% Multiplier = 6.45

% Perform the test case 100 times, and take average execution time of the
% 100 samples
for i = 1:100
    tic
    x = (sin(2*pi*24000*t)) * 6.45;
    temp_times(i) = toc;

```

```

end

times_mult(2) = mean(temp_times); % Average 100 execution time samples
temp_times = zeros(1,100); % Clear the temporary execution times for the next test
case

% Plot the original sine wave against result of the operation
plot(t, sin(2*pi*24000*t));
hold on
plot(t, x);
hold on
legend("2Vpp, 81 kHz Sinusoid", "(2Vpp, 81 kHz Sinusoid) x 2.2", "1Vpp, 24 kHz Sinusoid", "(1Vpp, 24 kHz Sinusoid) x 6.45");

% Multiplier = 9.3

% Perform the test case 100 times, and take average execution time of the
% 100 samples
for i = 1:100
    tic
    x = (sin(2*pi*24000*t)) * 9.3;
    temp_times(i) = toc;
end

times_mult(3) = mean(temp_times); % Average 100 execution time samples
temp_times = zeros(1,100); % Clear the temporary execution times for the next test
case

% Plot the original sine wave against result of the operation
plot(t, x);
legend("2Vpp, 81 kHz Sinusoid", "(2Vpp, 81 kHz Sinusoid) x 2.2", "1Vpp, 24 kHz Sinusoid", "(1Vpp, 24 kHz Sinusoid) x 6.45", "(1Vpp, 24 kHz Sinusoid) x 9.3");

%% DIVISION TESTS

times_div = zeros(1,4); % Array to store final, averaged execution times

% DC Signal Division

% Perform the test case 100 times, and take average execution time of the
% 100 samples
for i = 1:100
    tic
    x = 10 / 2;
    temp_times(i) = toc;
end

div_result = x; % Store the operation result
times_div(1) = mean(temp_times); % Average 100 execution time samples
temp_times = zeros(1,100); % Clear the temporary execution times for the next test
case

% Sinusoidal Division (Sine Wave Division)

t = 0:(1/50000):0.005; % Make 0.1 seconds sampled every 1/1000 of a second

sine = 5 * sin(2*pi*1000*t); % 10Vpp, 1 kHz, 0V Offset

% Perform the test case 100 times, and take average execution time of the
% 100 samples

```

```

for i = 1:100
    tic
    x = sine / 4.9;
    temp_times(i) = toc;
end

% Plot the original sine wave against result of the operation
figure(3);
plot(t,sine);
hold on
plot(t,x);
title("Sinusoidal Division Waveforms");
xlabel('Time [s]');
ylabel('Voltage [V]');
legend("Sinusoid","Sinusoid/2");

times_div(2) = mean(temp_times);    % Average 100 execution time samples
temp_times = zeros(1,100);        % Clear the temporary execution times for the next test
case

% Square Wave Division

t = linspace(0,3*pi)';
square_wave = 10 * square(t);

% Perform the test case 100 times, and take average execution time of the
% 100 samples
for i = 1:100
    tic
    x = square_wave / 2;
    temp_times(i) = toc;
end

% Plot the original square wave against result of the operation
figure(4);
plot(t/pi,square_wave);
hold on
plot(t/pi,x);
title("Square Wave Division Waveforms");
xlabel('Time [s]');
ylabel('Voltage [V]');
legend("Square Wave","Square Wave/2");

times_div(3) = mean(temp_times);    % Average 100 execution time samples
temp_times = zeros(1,100);        % Clear the temporary execution times for the next test
case

% Triangle Wave Division

t = 0:(1/50000):0.005; % Make 0.1 seconds sampled every 1/1000 of a second

triangle = 10 * sawtooth(2*pi*1000*t, 1/2);

% Perform the test case 100 times, and take average execution time of the
% 100 samples
for i = 1:100
    tic
    x = triangle / 11;
    temp_times(i) = toc;
end

```

```

% Plot the original triangle wave against result of the operation
figure(5);
plot(t,triangle);
hold on
plot(t,x);
title("Triangle Wave Division Waveforms");
xlabel('Time [s]');
ylabel('Voltage [V]');
legend("Triangle Wave","Triangle Wave/2");

times_div(4) = mean(temp_times);    % Average 100 execution time samples
temp_times = zeros(1,100);         % Clear the temporary execution times for the next test
case

%% DERIVATIVE TESTS

times_diff = zeros(1,3);           % Array for final, averaged execution times
t = 0:(1/50000):15;                % Make 0.1 seconds sampled every 1/1000 of a second

% Sinusoid Differentiation (Sine Wave Derivative)

sine = 0.5 * sin(t);               % 1Vpp Sine Wave

% Perform the test case 100 times, and take average execution time of the
% 100 samples
for i = 1:100
    tic
    x = diff(sine)./diff(t);
    temp_times(i) = toc;
end

times_diff(1) = mean(temp_times);   % Average 100 execution time samples
temp_times = zeros(1,100);         % Clear the temporary execution times for the next test
case

% Plot the original sine wave against result of the operation
figure(6);
plot(t(2:end),sine(2:end));
hold on
plot(t(2:end),x);
title("Sinusoid Differentiation Waveforms");
xlabel("Time [s]");
ylabel("Voltage [V]");
legend("Sinusoid","First Derivative of Sinusoid");

% Square Wave Diferentiation

square_wave = square(t);

% Perform the test case 100 times, and take average execution time of the
% 100 samples
for i = 1:100
    tic
    x = diff(square_wave)./diff(t);
    temp_times(i) = toc;
end

times_diff(2) = mean(temp_times);   % Average 100 execution time samples
temp_times = zeros(1,100);         % Clear the temporary execution times for the next test

```



```

case

% Plot the original square wave against result of the operation
figure(7);
yyaxis left
plot(t,square_wave);
hold on
yyaxis right
plot(t(2:end),x);
title("Square Wave Differentiation Waveforms");
xlabel("Time [s]");
ylabel("Voltage [V]");
legend("Square Wave","First Derivative of Square Wave");

% Triangle Wave Differentiation

% t = 0:(1/50000):0.005; % Make 0.1 seconds sampled every 1/1000 of a second
triangle = 10 * sawtooth(t, 1/2);

% Perform the test case 100 times, and take average execution time of the
% 100 samples
for i = 1:100
    tic
    x = diff(triangle) ./ diff(t);
    temp_times(i) = toc;
end

% Plot the original triangle wave against result of the operation
figure(8);
yyaxis left
plot(t,triangle);
hold on
yyaxis right
plot(t(2:end),x);
title("Triangle Wave Differentiation Waveforms");
xlabel('Time [s]');
ylabel('Voltage [V]');
legend("Triangle Wave","First Derivative of the Triangle Wave");

times_diff(3) = mean(temp_times); % Average 100 execution time samples
temp_times = zeros(1,100); % Clear the temporary execution times for the next test
case

%% INTEGRATION TESTS

% Sinusoid Integration (Sine Wave Integral)

times_int = zeros(1,3); % Array for storing final, averaged execution times
syms t

% Perform the test case 100 times, and take average execution time of the
% 100 samples
for i = 1:100
    tic
    y = int(sin(t),t);
    temp_times(i) = toc;
end

times_int(1) = mean(temp_times); % Average 100 execution time samples

```

```

temp_times = zeros(1,100);           % Clear the temporary execution times for the next test
case

% Plot the original sine wave against result of the operation
figure(9);
t = 0:(1/10000):15; % Make 0.1 seconds sampled every 1/1000 of a second
plot(t, sin(t));
hold on
plot(t, -cos(t));
title("Sinusoidal Integration Waveforms");
xlabel('Time [s]');
ylabel('Voltage [V]');
legend("Sinusoid", "First Integral of the Sinusoid");

% Square Wave Integration

t = 0:(1/10000):15; % Make 0.1 seconds sampled every 1/1000 of a second

% Perform the test case 100 times, and take average execution time of the
% 100 samples
for i = 1:100
    tic
    y = cumtrapz(t, square(t));
    temp_times(i) = toc;
end

times_int(2) = mean(temp_times); % Average 100 execution time samples
temp_times = zeros(1,100);      % Clear the temporary execution times for the next test
case

% Plot the original square wave against result of the operation
figure(10);
plot(t, square(t));
hold on
plot(t, y);
title("Square Wave Integration Waveforms");
xlabel('Time [s]');
ylabel('Voltage [V]');
legend("Square Wave", "First Integral of the Square Wave");

% Triangle Wave Integration

% Perform the test case 100 times, and take average execution time of the
% 100 samples
for i = 1:100
    tic
    y = cumtrapz(t, sawtooth(t, 1/2));
    temp_times(i) = toc;
end

times_int(3) = mean(temp_times); % Average 100 execution time samples
temp_times = zeros(1,100);      % Clear the temporary execution times for the next test
case

% Plot the original triangle wave against result of the operation
figure(11);
plot(t, sawtooth(t, 1/2));
hold on
plot(t, y);
title("Triangle Wave Integration Waveforms");

```

```

xlabel('Time [s]');
ylabel('Voltage [V]');
legend("Triangle Wave","First Integral of the Triangle Wave");

%% INVERSION

times_inversion = zeros(1,3); % Array for storing final, averaged execution times
t = 0:(1/10000):15; % Make 0.1 seconds sampled every 1/1000 of a second

% Sinusoid Inversion

% Perform the test case 100 times, and take average execution time of the
% 100 samples
for i = 1:100
    tic
    x = (sin(t)) * -1;
    temp_times(i) = toc;
end

times_inversion(1) = mean(temp_times); % Average 100 execution time samples
temp_times = zeros(1,100); % Clear the temporary execution times for the next
test case

% Plot the original sine wave against result of the operation
figure(12);
plot(t,sin(t));
hold on
plot(t,x);
title("Inversion on Sinusoids");
xlabel('Time [s]');
ylabel('Voltage [V]');
legend("Original Sinusoid","Inverted Sinusoid");

% Square Wave Inversion

% Perform the test case 100 times, and take average execution time of the
% 100 samples
for i = 1:100
    tic
    x = (1.5 * square(t)) * -1;
    temp_times(i) = toc;
end

times_inversion(2) = mean(temp_times); % Average 100 execution time samples
temp_times = zeros(1,100); % Clear the temporary execution times for the next
test case

% Plot the original square wave against result of the operation
figure(13);
plot(t,(1.5 * square(t)));
hold on
plot(t,x);
title("Inversion on Square Waves");
xlabel('Time [s]');
ylabel('Voltage [V]');
legend("Original Square Wave","Inverted Square Wave");

% Triangle Wave Inversion

% Perform the test case 100 times, and take average execution time of the

```

```

% 100 samples
for i = 1:100
    tic
    x = (2 * sawtooth(t,1/2)) * -1;
    temp_times(i) = toc;
end

times_inversion(3) = mean(temp_times); % Average 100 execution time samples
temp_times = zeros(1,100); % Clear the temporary execution times for the
next test case

% Plot the original triangle wave against result of the operation
figure(14);
plot(t, (2 * sawtooth(t,1/2)));
hold on
plot(t,x);
title("Inversion on Triangle Waves");
xlabel('Time [s]');
ylabel('Voltage [V]');
legend("Original Triangle Wave","Inverted Triangle Wave");

%% EQUATIONS

% Equation:  $y'' + 10y' + y = 3$ 

t = 0:(1/10000000):0.01; % Make 0.1 seconds sampled every 1/1000 of a second
y = sin(2000*pi*t); % 2Vpp Sine Wave

% Perform the test case 100 times, and take average execution time of the
% 100 samples
for i = 1:100
    tic

    a = diff(diff(y)./diff(t))./diff(t(2:length(t)));
    a(100000) = 0; a(100001) = 0;

    b = 10 * (diff(y)./diff(t));
    b(100001) = 0;

    z = a + b + y;

    temp_times(i) = toc;
end

times_equations = mean(temp_times); % Average 100 execution time samples

% Plot the result of the operation
figure(15);
plot(t,z);
title("Second-Order Differential Equation");
xlabel('Time [s]');
ylabel('Voltage [V]')

```

F. Hardware Configuration/Layout

Figure F-1 shows the hardware layout for the voltage adder circuit. This setup applies to the LTspice simulation, breadboard, and PCB testing.

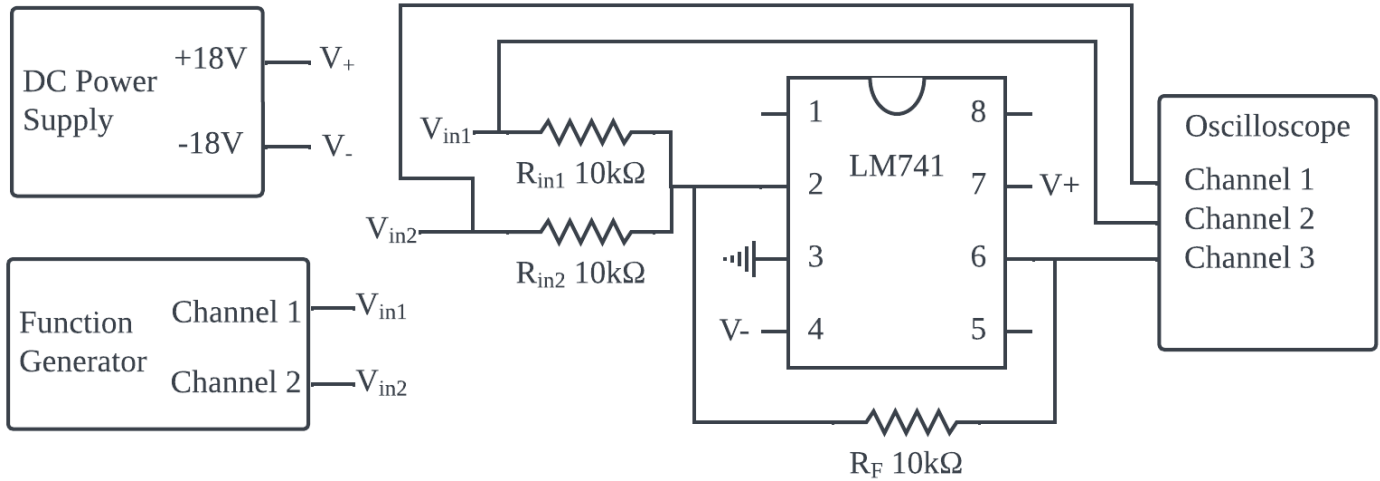


Figure F-1. Hardware Layout for the Voltage Adder (Designed for a Gain of 1)

Figure F-2 shows the hardware layout for the non-inverting voltage multiplier circuit. For this configuration, the multiplier is set to a gain of 2.2 via the combination of resistors. This setup applies to the LTspice simulation, breadboard, and PCB testing.

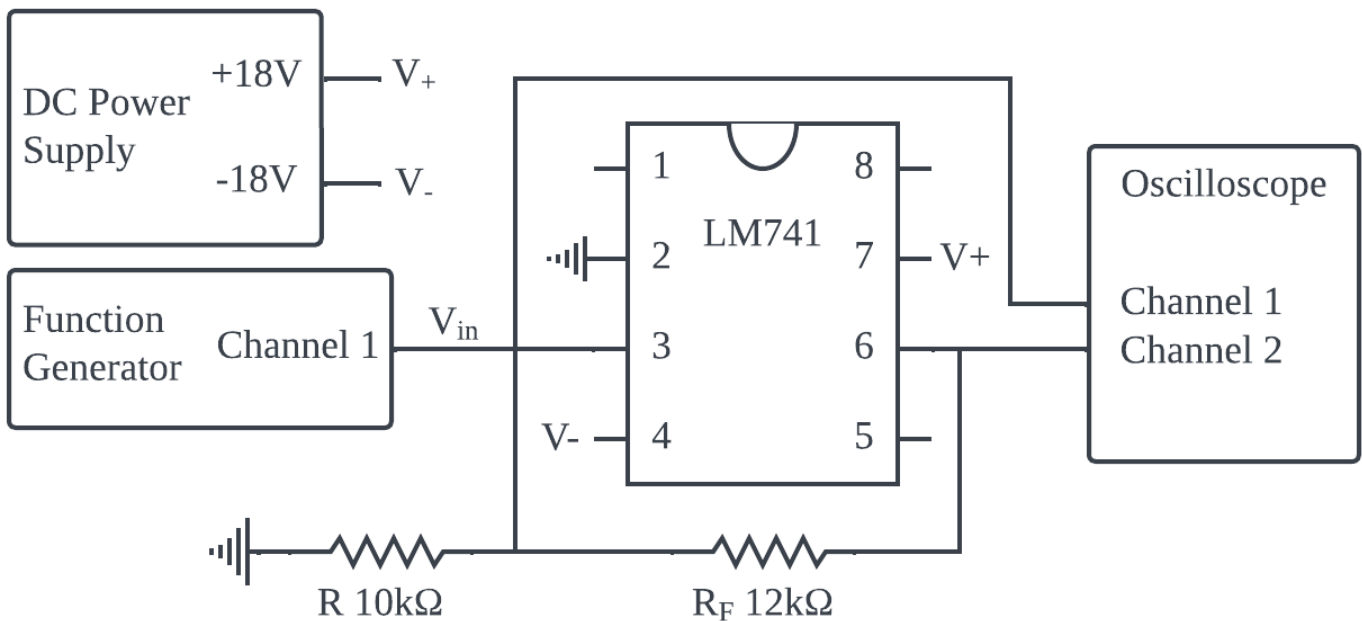


Figure F-2. Hardware Layout for the Non-Inverting Voltage Multiplier (Designed for a Gain of 2.2)

Figure F-3 shows the hardware layout for the resistive divider circuit. This setup applies to the breadboard and PCB testing. This configuration does not apply to the LTspice simulation configuration as two resistors were used rather than one potentiometer in LTspice.

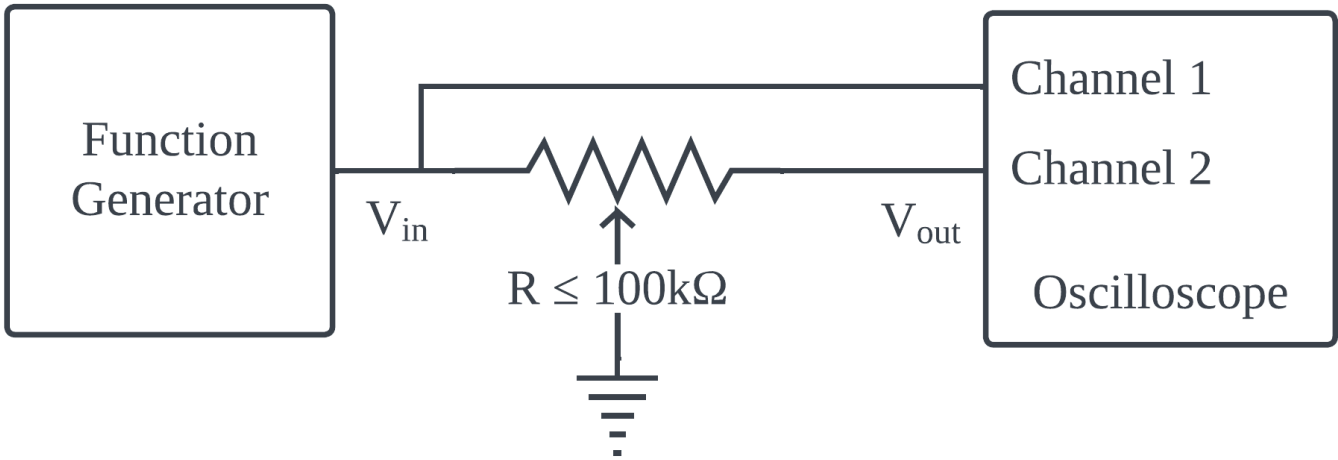


Figure F-3. Hardware Layout for the Resistive Divider

Figure F-4 shows the hardware layout for the differentiator circuit. This setup applies to the LTspice simulation, breadboard, and PCB testing.

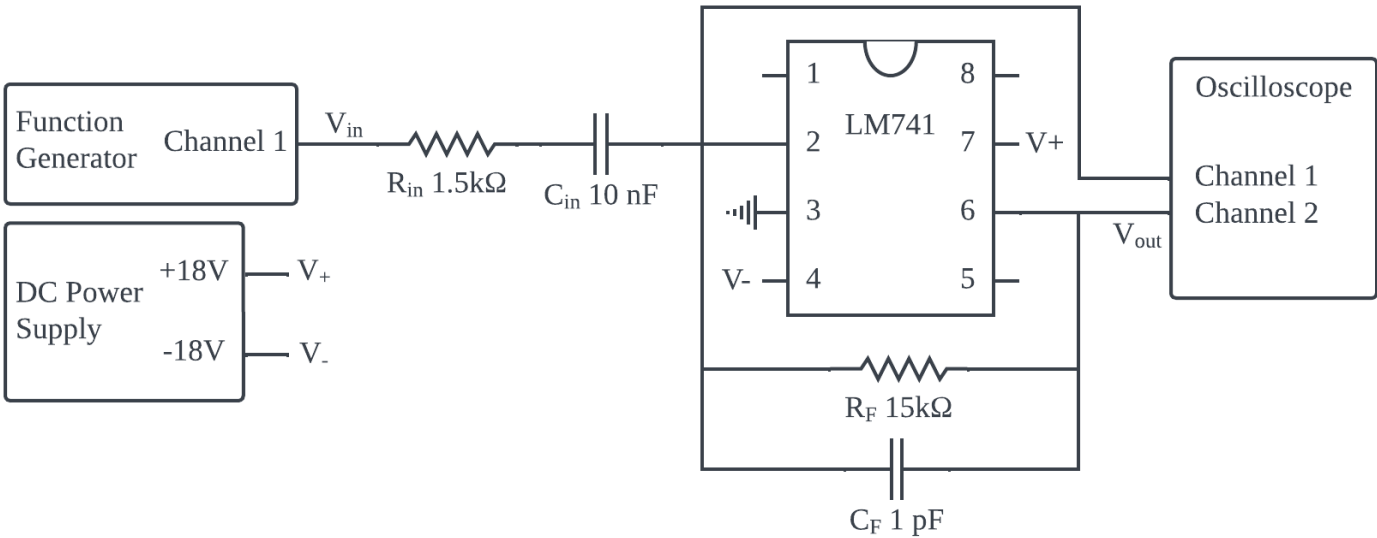


Figure F-4. Hardware Layout for the Differentiator

Figure F-1 shows the hardware layout for the integrator circuit. This setup applies to the breadboard and PCB testing layouts, but not the LTspice layout. This is because the LTspice configuration uses two resistors rather than one potentiometer for the resistive divider setup.

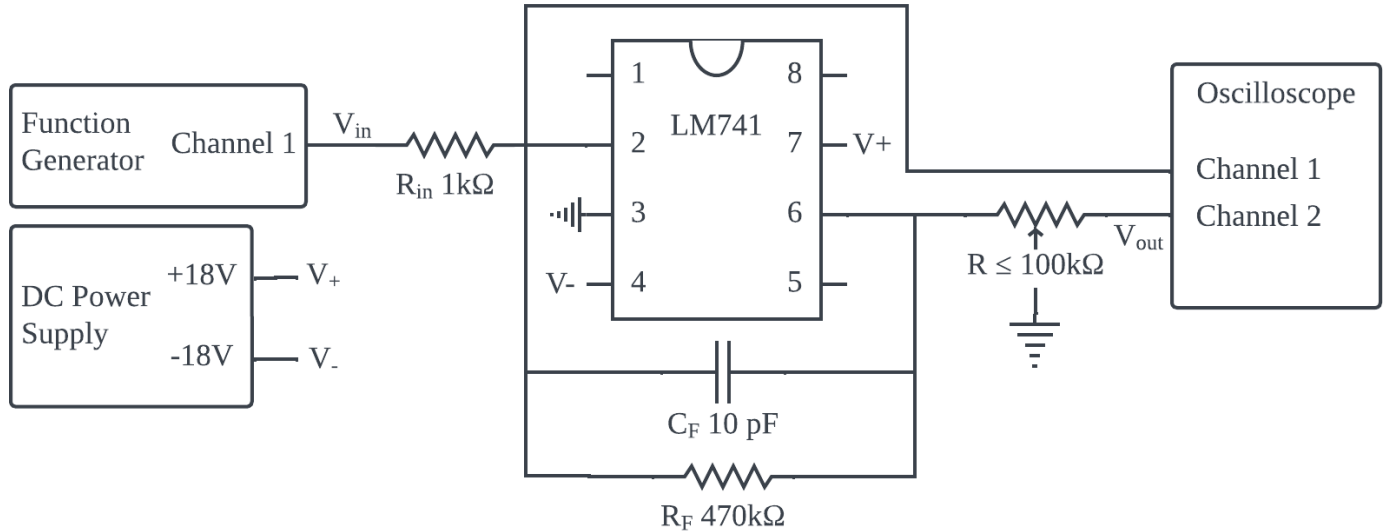


Figure F-5. Hardware Layout for the Integrator

Figure F-6 shows the hardware layout for the second-order differential equation circuit. This setup applies to breadboard and PCB testing configurations.

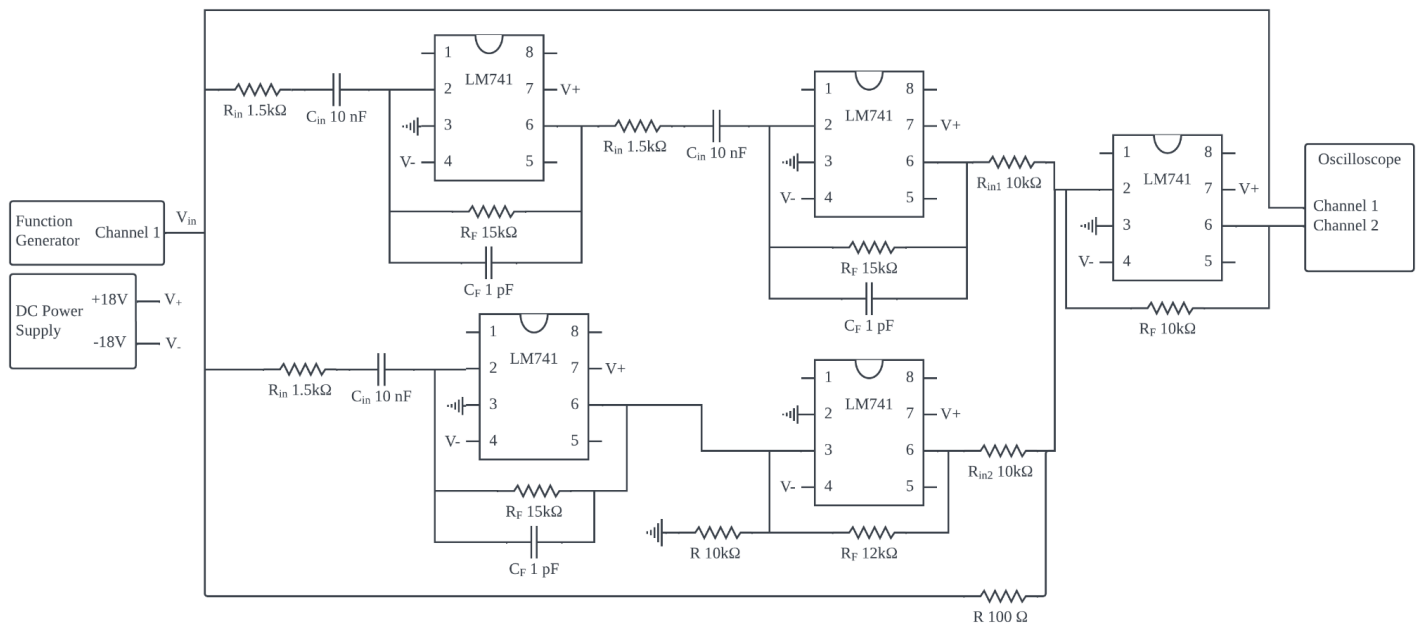


Figure F-6. Hardware Layout for the Tested Second-Order Differential Equation