ECE 445

SENIOR DESIGN LABORATORY

FINAL REPORT

Analog Computer ODE Solver

Team #26

DIANXING TANG
(dt12@illinois.edu)
HAIGE LIU
(haigel2@illinois.edu)
SHILONG SHEN
(shilong7@illinois.edu)
ZIXUAN QU
(zixuanq3@illinois.edu)

TA: Lumeng Xu
Sponsor: Said Mikki

May 17, 2025

Abstract

Analog computers have always been neglected in computer circuit design. Unlike digital computers, analog computers have very limited functions and are only used to solve mathematical problems. But in fact, the advantages of analog computers also lie in this. Especially when solving certain time-dependent problems, such as differential equations, analog computers can achieve low-latency solutions based on their physical circuit connectivity, which can help reduce delays in hardware design of many timing processes. In this project, as students of ECE and EE, we built a programmable simple analog computer based on simple circuits and Printed Circuit Board (PCB) design. This analog computer consists of two differential equation solvers, which can solve the first-order convergent ordinary differential equations and second-order convergent ordinary differential equations with variable parameters. Starting from the most basic operational amplifier chip, we explored the relationship between resistance and capacitance, and used Metal-Oxide-Semiconductor Field-Effect Transistor (MOSFET) and digital display mechanical potentiometer to realize the programmability of analog computers. Finally, we designed a user-friendly input and output system through Arduino. Users can use Arduino and Digital-to-Analog Converter (DAC) to output arbitrary waveforms (including common waveforms such as sine waves, cosine waves, constant signals, and double-wave coincidence waves) to solve any differential equations, and can present the results through an external four-channel oscilloscope. After testing, we compared the amplitude, frequency, and phase of the output of the analog computer we designed on some differential equations with the output of the Python digital differential equation solver we wrote, and finally concluded that our analog computer ordinary differential equation solver met our expected solution accuracy.

Contents

1 Introduction

In this final report of ECE 445 senior design, a complete design process of our project Analog Computer ODE Solver (Figure 1) is explained, including the overview of our design, the design procedure and functionality of each component, necessary verification of the requirement of each component and so on.

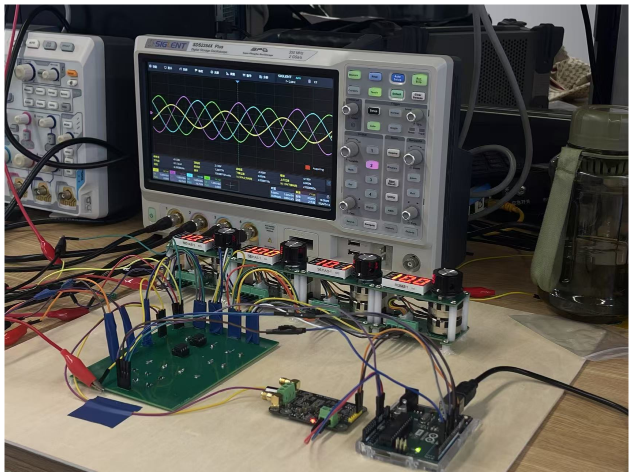

Figure 1: Visual Aid for Our Final Computer

We will conduct a comprehensive analysis of our achievement from theoretical support, previous research, feasibility analysis, and cost considerations of actually setting up the hardware equipment.

1.1 Background

Before the widespread use of digital computers, analog computers were among the first computing devices used to solve complex mathematical problems, especially ordinary differential equations (ODEs) [1]. Analog computers use continuous physical quantities, such as voltage, current, or mechanical motion, to model and solve equations in real time [2]. This approach offers inherent parallelism, high speed for certain types of problems, making analog computing an attractive alternative for certain applications [3].

In the ECE210 and ECE310 analog signal processing courses we have studied, we have learned about the relationship between physical circuits and differential equations. So we hope to design a simple Analog Computer ODE Solver through the courses we have learned and the PCB related knowledge learned in the ECE445 course.

1.2 Proposal Overview

As we have mentioned in our proposal and previous paragraphs, analog computers' property allows them to perform computations in a more natural way for some problems, especially those involving continuous-time systems like Ordinary Differential Equations

(ODEs) [4]. Therefore, we decided to build a programmable analog computer based on PCB. Besides the capabilities of solving different kinds of ODEs [5], providing a more versatile and powerful tool for scientific research and education. The innovative part of this design is the programmability in a hardware concept while the whole system is built on a PCB.

1.3 Component Units

Our whole system is composed of four major units: the **Power Supply Unit**, **User I/O Interface**, **Control Unit**, and **Computing Unit**.

- The **Power Supply Unit** provides all required voltages, including ±5V, 12V, and 5V, derived from a 220V AC source. The computer provides 5V exclusively to the Arduino through the USB port.
- The **User I/O Interface** includes an Arduino, knobs for resistance control, and an oscilloscope for real-time signal visualization.
- The **Control Unit** consists of a DAC, a variable resistor group, and a mode selector switch group. It converts digital signals into analog signals and sets parameters according to the selected ODE mode.
- The **Computing Unit** contains adders and a integrators composed of several operational amplifiers, and a differential equation solving circuit is formed through the differential relationship between the capacitors and the resistors.

By consisting these components, we built a a 3-mode ODE solving circuit that receives both the analog input signal and parameter settings, then computes the solution in real time.

The output analog signals—such as $u(t)$, $y(t)$, $y'(t)$, and $y''(t)$—can be observed on the oscilloscope for further analysis.

2 Design Details

2.1 Top-level Design

2.1.1 Block Diagram

The below figure show the top-level block diagram for signaling and powering between modules in the system.

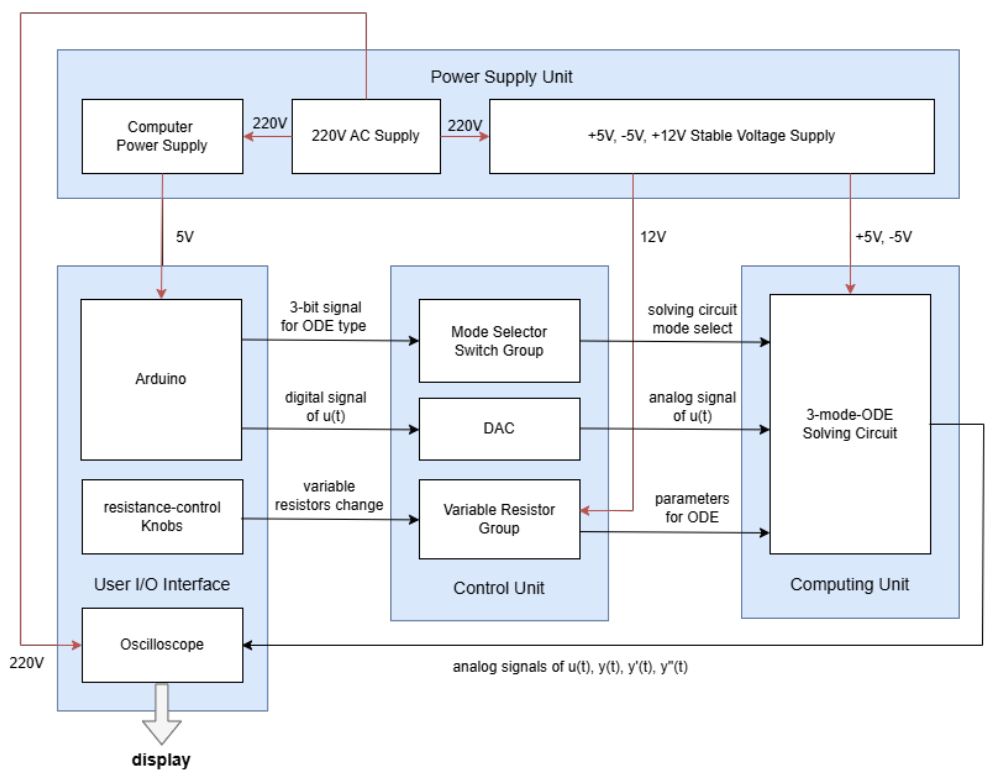

Figure 2: Top-level Block Diagram

2.1.2 Design Description

In our design, our analog computer consists of the following main modules:

- **Computing Unit** The computing unit in Figure 4 is the core unit of our analog computer, responsible for outputting the solution of ODE through analog circuit logic. The computation unit we designed supports 1st and 2nd order ordinary differential equations, and the order can be switched by the electrical signal of the control unit. We designed the circuit of the entire computing unit into a PCB, which is connected to other components through centralized power supply, multi-channel display, integrated control, etc.
- **Control Unit** The control unit is an important part of the programmability of our analog computer. Its control includes: using potentiometers (Figure 3) to realize

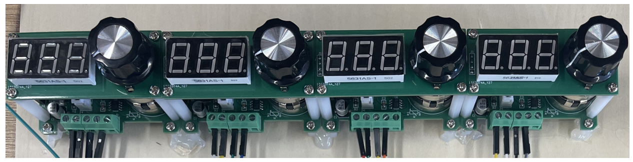

Figure 3: Potentiometers

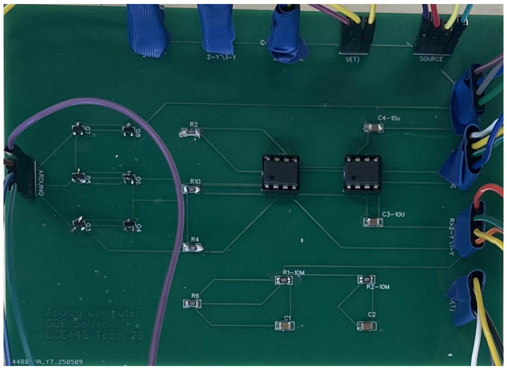

Figure 4: Computing Unit PCB

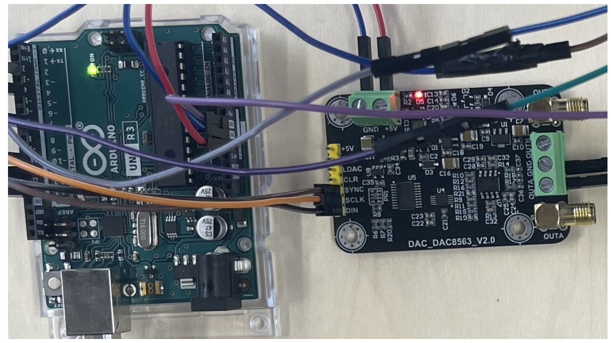

Figure 5: Arduino and Digital-Analog Converter

variable resistance to change the parameters of each ODE, and using electrical signals on MOSFET (Figure 4) to realize loop switching to switch the order of solver.

- **Power Supply Unit** The power supply unit is responsible for supplying power to the potentiometer digital display, operational amplifier chip, Arduino and other components. The selection of power supply unit components, especially the power supply for the chip, requires the use of a very stable constant voltage source to ensure the solver can solve stably and accurately.
- **User I/O Interface** For the design of the user Input /Output (I/O) interface (Figure 5), we need human-readable input and output. Hence, we integrate the signal generator and the control unit's level selector on the Arduino chip, and users can automatically control the Arduino and the simulated computer through the computer's serial port. As for output signal, since we need to read 4-channel signal in real time, the best way for us to do so is to use the oscilloscope in the laboratory.

In general, in our design, our PCB based analog computer can solve first-order and second-order ODEs with any variable parameters (and the solution must converge). Users can quickly solve the equations by inputting the signal to be solved to Arduino and changing the parameters of the ODE by changing the potentiometer.

2.2 Computing Unit

Figure 6 shows the overall block diagram of the Computing Unit, which is controlled by variable resistors and MOSFET switches. It supports three operation modes for solving different types of ODEs: first-order ODEs, second-order convergent ODEs, and second-order non-convergent ODEs. The red box highlights the specific block diagram for ODE solving computation.

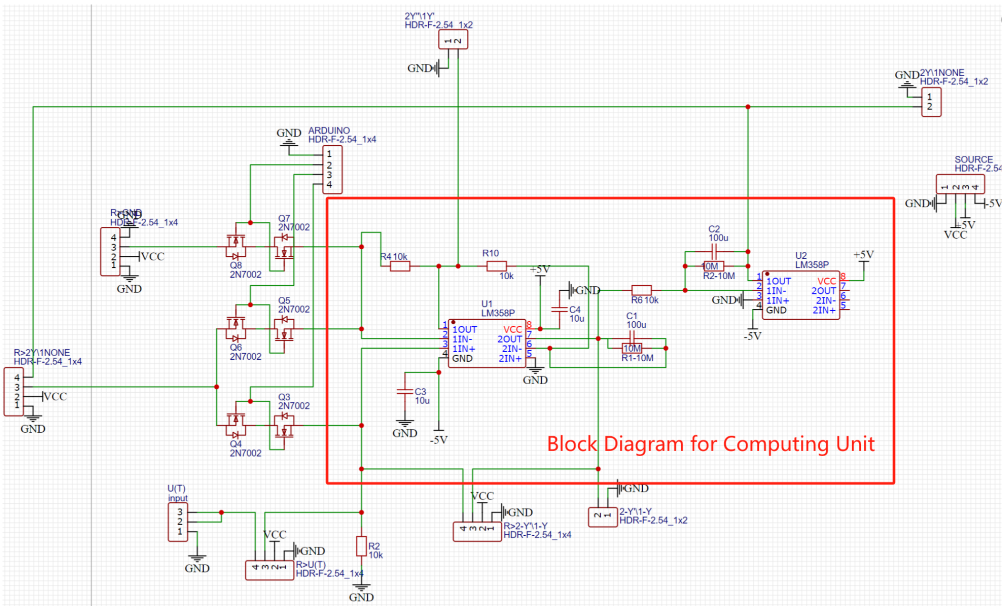

Figure 6: Computing Unit Block Diagram

2.2.1 Operational Amplifier Selection: LM358P

In our ODE solver circuit, we selected the **LM358P** operational amplifier or all computation modules.

The LM358P op-amp is ideal for our system, supporting dual-supply (±5V) operation with sufficient bandwidth (1MHz) and slew rate (0.3Vμs) for low-frequency (<10Hz) signal processing. Other considerations such as safety, cost-effectiveness, and ease of acquisition are also important.

2.2.2 Weighted Summing Circuit

With an op-amp configured in differential mode, signals can be applied simultaneously to both the inverting and non-inverting inputs of the op-amp, to perform addition and subtraction operations. Extra adding or subtracting terms can added by simply adding more input resistors to the inputs of the op-amp.

$$y = (V_1 + V_2) - (V_3 + V_4) \tag{1}$$

When the resistors in series with V1,V2,V3,V4 in the circuit are of different resistance values, the output voltage of the circuit is the sum of the voltages with weights. We will use the Weighted Summing Circuit in Figure 7 later for simulating the differential equation circuit.

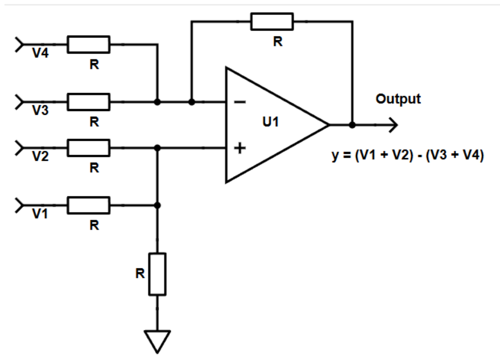

Figure 7: Weighted Summing Circuit

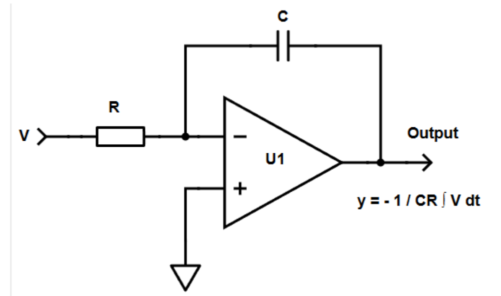

Figure 8: Op-amp Integrator Component Circuit

Requirement Description	**Verification Method**
- The weighted summing circuit must support analog signal inputs of both DC and AC with a frequency less than 10 Hz, and support extremum signal within the range of -5V to 5V.	- - Connect the circuit as shown in the Figure 7. Use analog signal voltage source to apply different input voltages from -5V to 5V with a frequency less than 10 Hz. Verify the output voltage will correspondingly change with the input voltages changing.
- It must respond to the input voltage within 0.1 seconds and provide a stable output.	- Use analog voltage source to apply input voltages that vary with time. Simultaneously measure the input and output voltage waveforms to verify that the response time is less than 0.1 seconds.
- It must perform the weighted summing operation with an error less than 10% of the theoretical value.	- Use analog signal voltage source to apply input voltages. Change resistance of R1, R2, R3. Verify the difference between the output voltage and theoretical voltage is less than 10%.

2.2.3 Integrator Circuit

Figure 8 shows a simple op-amp integrator requiring minimal components. All input current flows through the feedback capacitor, generating an output voltage proportional to the integral of the input signal. This fundamental circuit will serve as a building block for differential equation simulation.

$$y = -\frac{1}{CR}\int V\,dt \tag{2}$$

To precisely represent and solve ODEs, we chain these components together:

- Integration stages convert higher-order derivatives to lower-order terms.

- Amplifier stages apply coefficients to intermediate results.
- Adder stages combine these scaled signals to construct the final equation.

For example, Figure 9 illustrates the complete circuit for solving a second-order ODE of the form: $y'' = ay' + by - cu$,where each term is realized using a combination of integrators, inverters, and adders.

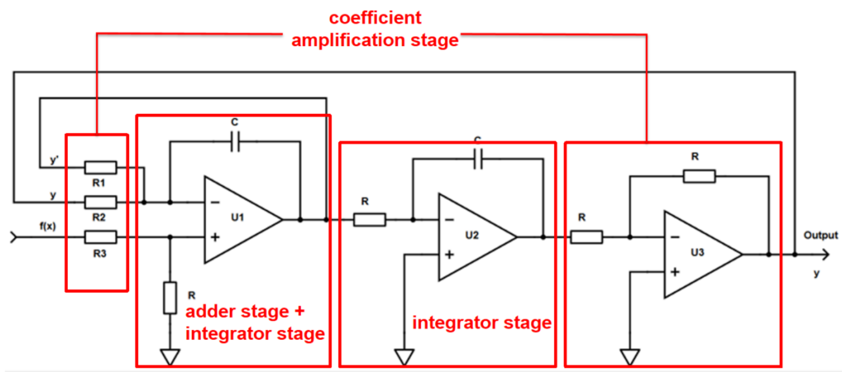

Figure 9: An example of circuit implementation solves the certain type of ODEs

Requirement Description	Verification Method
- The integrator circuit must support analog signal inputs of AC with a frequency less than 10HZ, and support extremum signal within the range of -5V to 5V.	- Connect the circuit as shown in the Figure 8. Use analog signal voltage source to apply different input voltages from -5V to 5V with a frequency less than 10HZ. Verify that the output voltage will correspondingly change with the input voltages changing.
- It must respond to the input voltage within 0.1 seconds and provide a stable output.	- Use analog signal voltage source to apply input voltages that vary with time. Simultaneously measure the input and output voltage waveforms to verify that the response time is less than 0.1 seconds.
- It must perform the integration operation with an error less than 10% of the theoretical value.	- Use analog signal voltage source to apply input voltages that vary with time. Verify the difference between the output voltage and theoretical voltage is less than 10%.
- It must include the function of filtering out noise interference.	- Intentionally apply noise signal. Verify the output voltage remains stable.

2.3 Control Unit

The control unit uses MOSFETs and potentiometers to enable hardware programmability: adjusting ODE parameters via digital potentiometers while configuring circuit topology through MOSFET switching, allowing dynamic reconfiguration of the analog computer's solver order and parameters.

2.3.1 Switching System

In our ECE340 course, we studied the output characteristic curves of MOSFETs. A MOSFET has three terminals: source (S), gate (G), and drain (D). Its operation relies on controlling the conductive channel between source and drain through the gate-to-source voltage (V_{GS}):

Cut-off region: $V_{GS} < V_{TH}$; no channel is formed, $I_{DS} \simeq 0$, the MOSFET is off.
Triode region: $V_{GS} > V_{TH}$ and low V_{DS}; it behaves like a variable resistor.
Saturation region: $V_{GS} > V_{TH}$ and $V_{DS} \geq V_{GS} - V_{TH}$; I_{DS} saturates and depends only on V_{GS}.

Based on this principle, we initially used single MOSFETs as voltage-controlled switches. Later, building on knowledge from ECE464, we adopted a more robust configuration using two N-channel MOSFETs in reverse series, which allows bidirectional current blocking and enhances circuit reliability.

In our design, three such reverse-series MOSFET pairs were used to form switches. These were controlled by Arduino output signals connected to pins 2, 3, and 4 (Figure 10). For example, when a high signal was sent to pin 2, the switch between terminals A and B closed; otherwise, it remained open. The same logic applied to pins 3 and 4, controlling switches between terminals C–D and E–F respectively.

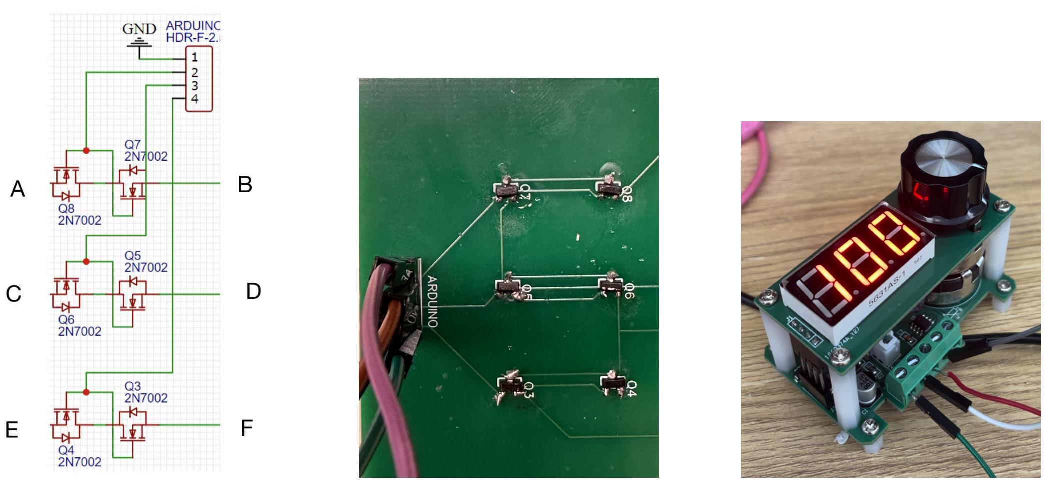

Figure 10: Switching Design with MOSFET & SX02 Potentiometer (Right)

Requirement Description	Verification Method

- Each switch of this switching system must close (the resistance of the switch is less than 100Ω) when receiving 5V control voltage.	- Use Arduino to provide a 5V control voltage to one of the switches. Verify the resistance of the switch is less than 100Ω. Repeat the operation to each switches.
- Each switch of this switching system must open (the resistance of the switch is larger than $10G\Omega$) when receiving 0V control voltage.	- Use Arduino a 0V control voltage to one of the switches. Verify the resistance of the switch is larger than $10G\Omega$. Repeat the operation to each switches.
- Both its turn-on response time and turn-off response time must be less than 1 second.	- Use Arduino to quickly switch 0V→5V and 5V→0V, and test the opening and closing response times to ensure that they are both less than 1 second.

2.3.2 Parameter Tuner

The idea of building a programmable analog computer came from [6], which introduced a method of tuning parameters in a digital way. In order to adjust the parameters of the ODE to be solved, according to our previous research[7] and the designed analog circuit logic, we need to adjust the ratio of specific resistors to achieve this. Therefore, we need to introduce variable resistor. The variable resistor we apply here is SX02 mechanial potentiometer (Figure 10). This allow us to adjust the resistor by hand and shows corresponding resistor on terminal.

Requirement Description	**Verification Method**
- The mechanical potentiometer must provide a variable resistance range from 1kΩ to 50kΩ.	- Connect it with ohmmeter and rotate the knob to change the resistance value. Verify the range of the resistance range.
- It must display the resistance value within an error of 10%.	- Use a variable power supply to power its display screen and rotate the knob to change the resistance value. Verify the difference of the actual resistance and the displayed resistance is less than 10%.

2.4 Power Supply Unit

2.4.1 220V AC Supply

The 220V AC supply provides power to the computer, oscilloscope, and the DC voltage converter, which generates the ±5V and ±12V required by the analog circuits.

2.4.2 ±5V, +12V Stable DC Voltage Supply

This stable ±5V/+12V DC supply powers the system's analog components: the ±5V enables LM358P op-amps to achieve symmetrical voltage swing, while 12V drives variable resistors for reliable control. The isolated design ensures precise analog processing, enhanced linearity, and lab-safe operation.

DC Voltage Supply Setup

The ±5V and +12V DC voltages used in the system are provided by a programmable DC power supply (RIGOL DP832). This instrument features three independent output channels:

- **Channel 1** is configured to output +5V.
- **Channel 2** is configured to output +12V.
- **Channel 3** is configured to output -5V, by connecting the positive terminal to the ground reference and taking the negative terminal as the -5V output.

To realize the -5V output, a virtual ground is established by connecting the positive terminal of Channel 3 to the system ground (green terminal), while the negative terminal is used as the -5V supply.

Requirement Description	Verification Method
- It must provide stable voltages ±5V, +12V, which means the ripple of the voltage should be less than 0.1V.	- Measure the output voltage of the voltage source with an oscilloscope. Verify the ripple of the output voltage is less than 0.1V.
- It must include short-circuit protection and thermal protection.	- Intentionally short or overheat this voltage supply under supervision. Verify that it will automatically limit the output power.

2.4.3 USB-A Power Supply

The Arduino is connected to the computer via a USB-A portal, which not only provides a stable 5V power supply but also enables serial communication between the two devices. Through this connection, the computer can transmit control instructions to the Arduino and receive data if needed, allowing real-time configuration of the ODE mode, input signal type, and parameter adjustments during the experiment.

2.5 User I/O Interface

2.5.1 Waveform Generator

In the initial design and experimental verification, an important device we need to use is a function waveform generator. Because we need to use different sin waves to verify that our analog output has the same amplitude and phase as the ideal digital output to ensure that our design is correct. However, in actual use, for lightweight display and integration, we choose to use a homemade function generator. Our function generator is implemented by Arduino and a digital-to-analog converter (DAC).

Arduino - DAC8563 System

Arduino Uno itself cannot output analog signals, but a DAC controlled by a serial port on the basis of Arduino can convert digital signals to analog signals. The specific operation is as follows:

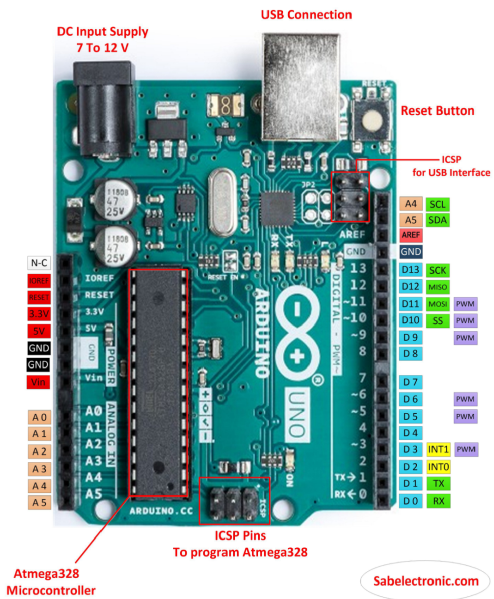

Figure 11: Arduino Uno Layout

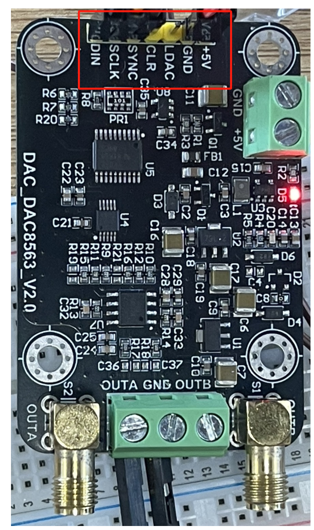

Figure 12: DAC8563 Layout

As shown in the given layout instruction of Arduino Uno in Figure 11, the green labels are used for serial connection. We need to use port 1̃1 Master Output Slave Input (MOSI), 1̃3 Serial Clock (SCK) and 1̃0 Slave Select (SS).

Respectively, the ports we use in the Arduino have to connect to specific ports in DAC shown in Figure 12. By using SPI library in Arduino IDE, we can connect MOSI to DIN (main data transfer), SS to SYNC (device choice by default) and SCK to SCLK (using the same clock).

By applying the necessary transformations (such as calibrating the reference voltage to 5V or 3.3V) and converting units, we have achieved the generation of a single wave to a dual wave complex using the interface of our personal laptop.

We can adjust the amplitude and frequency of each individual waveform so that we can customize all functions we need to be the $u(t)$ of our input.

Requirement Description	Verification Method
- Arduino-DAC8563 System must provide an analog signal. And the amplitude range of the analog signal is from -5V to 5V, and the frequency range is from 0 to 10Hz.	- Connect the Arduino to the computer via the USB interface. Adjust the relevant codes for analog signal generation. Verify the amplitude range and frequency range of the analog signal.

Switching System Input Integration

Besides, we have mentioned that we used MOSFET as the switching system design, and we need to use three 0V or 5V voltage to be the activate signal. When we use the Arduino, we can use the digital output port 2, 4, 7 of Arduino, which can produce a 5V constant signal (reference voltage) and we can modify it also in the interface of serial monitor.

Requirement Description	Verification Method
- The switching system input integration must be able to generate control voltages signals of 0V or 5V.	- Connect the Arduino to the computer via the USB interface. Adjust the relevant codes for switching control. Verify the control voltage can switches.

2.6 Waveform Display Device

To verify the correctness of our analog ODE solver, we needed to visualize its output and compare it with digital signal computing results from Python. We initially considered using Arduino as a simple oscilloscope, but encountered critical limitations: 0–5V input range, limited sampling rate, low frequency support, and poor multi-channel display capability as shown in Figure 13. These constraints made Arduino unsuitable for visualizing our second-order ODEs, especially since negative derivatives and higher frequencies are inevitable in general solutions. After discussion with our sponsor and TA, we adopted a lab oscilloscope instead. It provides real-time, high-resolution analog waveform display, aligns with the nature of our analog computing design, and avoids the shortcomings of low-cost digital alternatives.

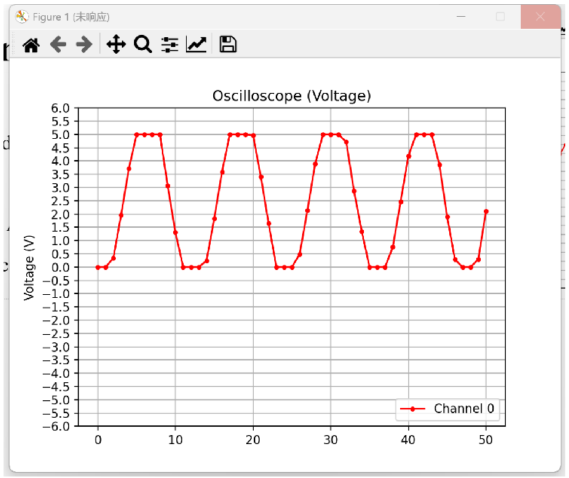

Figure 13: Arduino Analog Signal Output With Error

Figure 14: Oscilloscope Display

An example of our output signal display in oscilloscope is shown in Figure 14.

Requirement Description	**Verification Method**
- The waveform display device must support the 4-channel detection of the analog signals. And the amplitude range of each analog signal is from -5V to 5V, and their frequency range is from 0 to 10Hz.	- Use analog signal voltage source to apply 4-channel different analog signal from -5V to 5V with a frequency less than 10 HZ. Verify that it can display input analog signal completely.

2.7 Analog Computer

2.7.1 Analog Computer Circuit Integration

To construct a complete and functional analog computer, we integrate the four essential modules previously described: Computing Unit, Control Unit, Power Supply Unit and User I/O Interface.

These modules are physically wired and logically organized into a unified analog computing platform. The Arduino provides a digital 3-bit signal to control the MOSFET-based mode selection switch, an analog waveform $u(t)$ via DAC, and control signals for resistance adjustment. The user can then observe the output of the selected ODE-solving circuit in real time through the oscilloscope.

2.7.2 Circuit Functionality and Mode Switching

The analog computer solves different ODE types by selecting one of three embedded circuits via a 3-bit digital control signal (see Figure 15). Each mode solves a specific form of ODE:

- **Mode 1: 1st-ODE ($ay' + by = u(t)$)**
  Control signal = 500, diagram is 16. A single integrator circuit combines $u(t)$ and $y(t)$ via an adder-inverter stage to generate $y(t)$, solving stable first-order dynamics.
- **Mode 2: 2nd-Order Convergent ODE ($ay'' + by' + cy = u(t)$)**
  Control signal = 050, diagram is 17. A two-stage integrator chain processes weighted $u(t)$, $y(t)$, and $y'(t)$ to compute $y(t)$, solving stable, damped second-order systems.
- **Mode 3: 2nd-Order Non-Convergent ODE ($ay'' - by' + cy = u(t)$)**
  Control signal = 505, diagram is 18. Similar to Mode 2, but $y'(t)$ is inverted to simulate negative damping, enabling unstable or oscillatory responses.

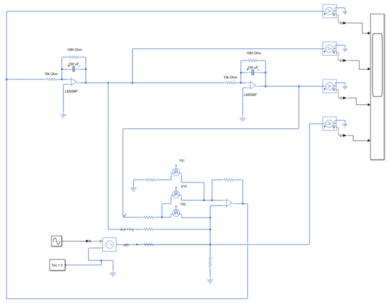

Figure 15: ODE-Solving Circuit

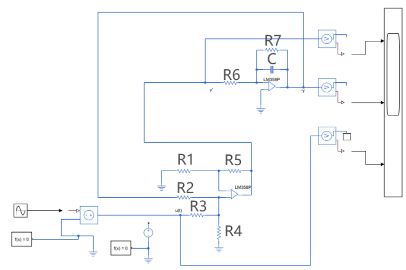

Figure 16: Mode 1

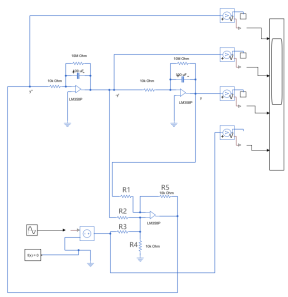

Figure 17: Mode 2

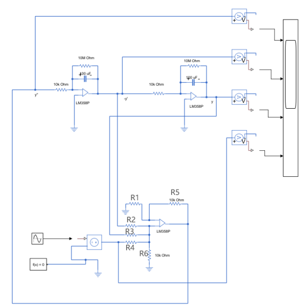

Figure 18: Mode 3

3 Tolerance Analysis

3.1 First Order ODE

To find the error source of the first-order ode, let us reconsider the computing data flow. The equation to be solved is

$$ay'(t) + by(t) = u(t) \tag{3}$$

According to the circuit structure,

$$a = \frac{1 + \frac{R_3+\Delta_3}{R_4+\Delta_4} + \frac{R_3+\Delta_3}{R_2+\Delta_2}}{1 + \frac{R_5+\Delta_5}{R_1+\Delta_1}} \qquad b = \frac{R_3+\Delta_3}{R_2+\Delta_2} \tag{4}$$

For the integration circuit, the additional ratio is not 1

$$(R_5+\Delta_5)(C+\Delta)! = 1 \tag{5}$$

Thus the real solving equation would be the following:

$$\left(1 + \frac{R_3+\Delta_3}{R_4+\Delta_4} + \frac{R_3+\Delta_3}{R_2+\Delta_2}\right)\frac{1}{\left(1+\frac{R_3+\Delta_3}{R_1+\Delta_1}\right)}y'(t) + \frac{R_3+\Delta_3}{(R_2+\Delta_2)(R_3+\Delta_3)(C+\Delta)}y(t)$$
$$= \frac{1}{(R_3+\Delta_3)(C+\Delta)}u(t)$$

By further calculating the partial derivatives of solution with respect to other parameter, we could eliminate some subtle variables which have little impact on the result. When $\Delta_i \ll R_i$ and $\Delta \ll C$, we could get the error of output:

$$\Delta = y_p \cdot \left[-\frac{\Delta_1}{R_1}\cdot\frac{R_3}{R_1+R_3} - \frac{\Delta_2}{R_2}\cdot\left(1+\frac{R_3(R_1+R_3)}{R_2(R_1+R_3+R_4)}\right) + \frac{\Delta_3}{R_3}\cdot\left(\frac{R_3(R_2+R_4)}{R_2R_4(R_1+R_3)} - 1\right) - 2\frac{\Delta}{C}\right].$$

To make sure the error of output is within 10 percentage, further estimate other error source error range. By assuming only one error on site and other parameters are ideal, we could have the following maximum error. When they have the following value R1=10kΩ, R2=20kΩ, R3=30kΩ, R4=40kΩ, ω=100rad/s. (also applies to other values)

$$\left|\frac{\Delta y}{y_p}\right| \le 10\% \quad \text{Which is} \quad |S_1\delta_1 + S_2\delta_2 + S_3\delta_3 + S_4\delta_4 + S_C\delta_C| \le 0.1 \tag{6}$$

1. R_1 allowance error: $0.75 \cdot \frac{|\Delta_1|}{10k} \le 0.1 \quad \Rightarrow \quad |\Delta_1| \le \frac{0.1\times 10k}{0.75} \approx 1.33k\Omega\,(13.3\%)$.
2. R_2 allowance error: $1.75 \cdot \frac{|\Delta_2|}{20k} \le 0.1 \quad \Rightarrow \quad |\Delta_2| \le \frac{0.1\times 20k}{1.75} \approx 1.14k\Omega\,(5.7\%)$.
3. R_3 allowance error: $0.94375 \cdot \frac{|\Delta_3|}{30k} \le 0.1 \quad \Rightarrow \quad |\Delta_3| \le \frac{0.1\times 30k}{0.94375} \approx 3.18k\Omega\,(10.6\%)$.

4. Capacity C allowance error: $2 \cdot \frac{|\Delta|}{100\mu F} \leq 0.1 \quad \Rightarrow \quad |\Delta| \leq \frac{0.1\times 100\mu F}{2} = 5\mu F\,(5\%)$.

The error of variable resistors would be below 5 percent according to the product manual, thus the error of our output would be below 10 percent. For our capacity, it is constant and thus always satisfy our accuracy requirement.

3.2 Second Order ODE

Following the same analysis step, we have no necessity to reintroduce. The real solved equation in our circuit is (note that due to constant capacity and its corresponding resistor of integrating circuit, its error could be ignored here)

$$\frac{1+\frac{R_3+\Delta_3}{R_2+\Delta_2}+\frac{R_3+\Delta_3}{R_4+\Delta_4}}{1+\frac{R_5+\Delta_5}{R_1+\Delta_1}}y'' + \frac{R_3+\Delta_3}{R_2+\Delta_2}y' + \frac{1+\frac{R_3+\Delta_3}{R_2+\Delta_2}+\frac{R_3+\Delta_3}{R_4+\Delta_4}}{1+\frac{R_1+\Delta_1}{R_5+\Delta_5}}y = u$$

Given initial value zeroing, use sine function input and choose one timestamp when sine and cosine signals are the same $\frac{\sqrt{2}}{2}$.

$$y(t) = \frac{\sqrt{2}}{2} \cdot \frac{\left(1+\frac{R_3+\Delta_3}{R_2+\Delta_2}+\frac{R_3+\Delta_3}{R_4+\Delta_4}\right)\left(\frac{R_5+\Delta_5}{R_1+\Delta_1}-\omega^2\right) - \left(1+\frac{R_5+\Delta_5}{R_1+\Delta_1}\right)^2\left(\frac{R_3+\Delta_3}{R_2+\Delta_2}\right)\omega}{\left(1+\frac{R_3+\Delta_3}{R_2+\Delta_2}+\frac{R_3+\Delta_3}{R_4+\Delta_4}\right)^2\left(\frac{R_5+\Delta_5}{R_1+\Delta_1}-\omega^2\right)^2 + \left(1+\frac{R_5+\Delta_5}{R_1+\Delta_1}\right)^2\left(\frac{R_3+\Delta_3}{R_2+\Delta_2}\right)^2\omega^2} \tag{7}$$

Further set parameters to our common value R1=10kΩ, R2=10kΩ, R3=10kΩ, R4=10kΩ, R5=10kΩ, ω=100rad/s, calculate the partial derivatives of each parameters with respect to the output, we have the following inequalities.

$$\left|\frac{\Delta y}{y_p}\right| \leq 10\%\Omega \quad \text{Which is} \quad |S_1\delta_1 + S_2\delta_2 + S_3\delta_3 + S_4\delta_4 + S_5\delta_5 + S_C\delta_C| \leq 0.1 \tag{8}$$

Resistor Parameters Table: R_1: 10 kΩ ±7.2% — R_2: 10 kΩ ±7.2% — R_3: 10 kΩ ±10.5% — R_4: 10 kΩ ±11.7% — R_5: 10 kΩ ±10.4% According to the test result in the last section, our resistor error would still be acceptable range.

4 Cost Analysis

4.1 Bill of Materials (BOM)

The main cost of this project is on circuit components. It includes PCB board, resistors, amplifier chips and so on. The detailed cost is listed below.

Device Name	Quantity	Cost
AD633JRZ	3	44.4 RMB
AD5254BRUZ1	3	56.4 RMB
AD835 Multiplier	1	118 RMB
arduino uno r3	1	104 RMB
DAC8563	1	126 RMB
1000UF Capacity	5	20.5 RMB
SX02 Digital Potentiometer	4	220 RMB
0-10V Analog-to-Digital Potentiometer Converter Module	1	45.8 RMB
SMA Breadboard Wire	4	27.2 RMB
PCB and relative materials	4	429.21 RMB
Packaging Material	1	50 RMB
UI element and Power Supply	1	0 RMB

Table 9: A simple model device summary

The BOM of this project is totally 1241.51 RMB.

4.2 Labor Costs

We normally work together on this project on site and have a fair work distribution. Including 1 hour meeting our professor one week, 10 hours on laboratory to build and test our circuit. The estimated salary per hour could be 50 RMB and the lab cost hours are 440 hours at least. The labor costs would be $2.5 \times 50 \times 440 = 55000$ RMB.

5 Work Distribution

We have a relatively balanced division of labor, with each person leading a major role while participating in all the work. The followings show the main work (not all work) everyone has done.

Zixuan Qu (25%)

- Finished all Arduino coding stuff, including the Arduino-AD8563 waveform generator of the I/O system and the Integrated switching with codes.
- Designed the schematic and PCB layout of 2nd order ODE solver in ICEDA [8] and Parameter Tuner with potentiometer of control unit.

Shilong Shen (25%)

- Designed the ultimate Analog Computer with integrated first-order and second-order ODE solvers.
- Implemented the software simulation of the 1st-2nd ODE Solver in Simulink, and tested the physical circuit based on the simulation.

Haige Liu (25%)

- Provided theoretical support and mathematical analysis for the feasibility of ODE solver circuits for analog computers.
- A test data set was created to verify the correctness of the final design, and a Python-based ODE solver was designed to assist in verifying the results.

Dianxing Tang (25%)

- The PCB board and some components were welded and the functionality of the PCB was tested.
- Designed the overall packaging appearance, including the cutting of acrylic sheets and the relative positions of the components.

6 Ethics and Safety

In general, our design strictly obeys the code of ethics of IEEE [9], and we will also focus on the safety of our working procedure, detailed explanations are made in the following subsections.

6.1 Ethics Consideration

- **Intellectual Property**: Ensure that all the design concepts, circuit schematics, and algorithms used in the project are either original or properly cited if based on existing works. Avoid any form of plagiarism.
- **Open - Source and Collaboration**: Consider making the design of the analog computer open source, allowing other researchers and students to build upon and improve the work. This promotes collaboration and the sharing of knowledge within the scientific community.

6.2 Safety Consideration

- **Electrical Safety**: Since the analog computer is based on PCB boards with electrical components, proper insulation and grounding must be ensured. All electrical connections should be made according to safety standards to prevent electric shocks and short circuits.
- **Laboratory Safety**: Most of the work for the later half of our job will be conducted in laboratory, therefore, we must strictly follow the rules and keep our self safe from all equipment we use. For example, when we do hand soldering with soldering pencil, we need to be careful with the heat iron and not to be scald.
- **Component Handling**: When handling components such as resistors, capacitors, and integrated circuits, proper anti-static measures should be taken to avoid damaging the components. Also, ensure that the components are rated for the voltage and current levels in the circuit to prevent overheating and potential fire hazards.

7 Conclusion

In our design, we successfully used the characteristics of analog circuits to solve first-order and second-order differential equations. We controlled the circuits through a laptop and knobs, just like an engineer operating a professional computer device, and then obtained the desired output from the oscilloscope.

Although the idea of creating and applying analog computers to solve practical problems was proposed as early as the end of the 20th century, the application of analog computers was limited to solving certain types of theoretical equations, and few people have realized this idea. In this project, by building a general-purpose analog computer to quickly solve differential equations, we also hope to realize a realistic idea through this: equip some practical devices with analog computers to efficiently process analog signals without converting the signals to digital signals and then converting them back, reducing the delay of device communication.

In this senior design, we really worked with some engineers from proposal, planning, design to construction implementation. We learned knowledge that we could not learn from books through actual operation and experiments. We turned abstract content into concrete growth, which was very helpful for our future engineering career.

References

[1] *Differential equation — Solving, Applications & Examples — Britannica*. [Online]. Available: https://www.britannica.com/science/differential-equation (visited on 02/23/2025).

[2] B. J. MacLennan, "A review of analog computing," *Department of Electrical Engineering & Computer Science, University of Tennessee, Technical Report UT-CS-07-601 (September)*, pp. 19 798–19 807, 2007.

[3] E. R. S. Banger, *Analog Computer: Examples, Types, Characteristics, & Advantages*, en-US, Oct. 2024. [Online]. Available: https://computertechinfo.com/analog-computer-definition-examples/ (visited on 02/23/2025).

[4] C. E. Shannon, "Mathematical theory of the differential analyzer," *Journal of Mathematics and Physics*, vol. 20, no. 1-4, pp. 337–354, 1941.

[5] T. ElAli, S. Jones, F. Arammash, *et al.*, "An Analog Computer To Solve Any Second Order Linear Differential Equation With Arbitrary Coefficients," en, in *Innovative Algorithms and Techniques in Automation, Industrial Electronics and Telecommunications*, T. Sobh, K. Elleithy, A. Mahmood, and M. Karim, Eds., Dordrecht: Springer Netherlands, 2007, pp. 449–451, ISBN: 978-1-4020-6266-7. DOI: 10.1007/978-1-4020-6266-7_81.

[6] J. Wu, B. Liu, J. Peng, *et al.*, "On-Chip Tunable Second-Order Differential-Equation Solver Based on a Silicon Photonic Mode-Split Microresonator," *Journal of Lightwave Technology*, vol. 33, no. 17, pp. 3542–3549, Sep. 2015, Conference Name: Journal of Lightwave Technology, ISSN: 1558-2213. DOI: 10.1109/JLT.2015.2442911. [Online]. Available: https://ieeexplore.ieee.org/document/7122249/?arnumber=7122249 (visited on 02/23/2025).

[7] G. Fernando, *Analogue Computing*, en, Section: Analogue Computing, Oct. 2021. [Online]. Available: https://www.i4cy.com/analog_computing (visited on 02/23/2025).

[8] JLCPCB, *JLCPCB - PCB prototype & SMT assembly service*, https://www.jlc.com/newOrder/?msclkid=880fa4654b5d181b3eb9cc4c2532eb53#/newBrand?from=YL3W0Q14-gPU3&s=TD, Accessed: 2025-04-28, 2025.

[9] *Ieee code of ethics*, IEEE Policies, Section 7 – Professional Activities (Part A – IEEE Policies). [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html (visited on 03/13/2025).

Appendix A Working Example

In Appendix A, we will use a simple example of solving a second order ODE to show you how our design work in holistic and brief description.

If we want to solve:

$$8y'' + 3y' + 4y = 3u(t) \qquad\qquad u(t) = 2sin(t) + 3sin(\pi \cdot t) \tag{9}$$

According to the simplified theoretical relation between parameters and the resistance of potentiometers (Figure 19).

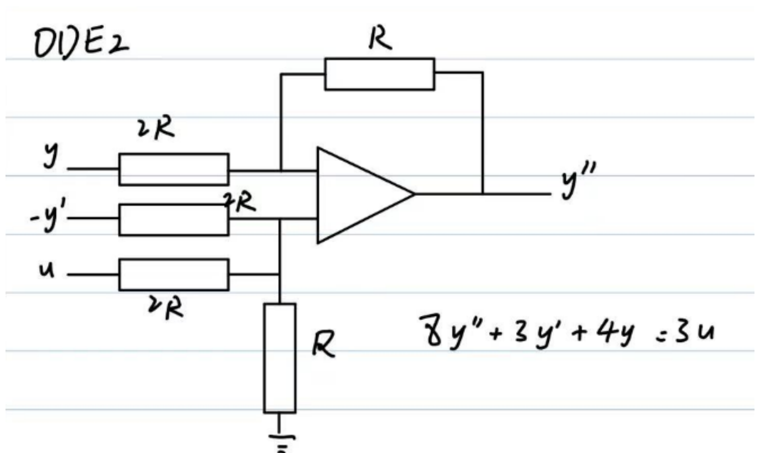

Figure 19: Relation between parameters and the resistance of potentiometers

We can change the corresponding potentiometers to 10, 20, 20, 20 using the knobs shown in Figure 3. Then, set the $u(t)$ to be a composite wave of $2sin(t)$ and $3sin(\pi \cdot t)$ using the computer and Arduino.

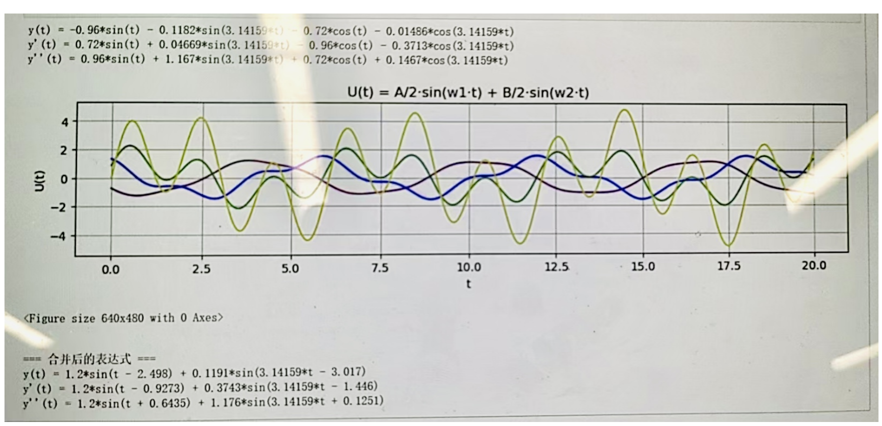

Figure 20: Digital Solution using Python

Figure 21: Analog Solution using Our Design

Finally, we can compare the result produced by both digital solver (Figure 20), which is based on Python script, and the analog signal result (Figure 21), which is solved by our analog computer, on oscilloscope to justify the correctness of our solution. For example, in this question, both green plot results reflect the equation of $y = 1.2 \cdot sin(t - 2.498) + 0.1191 \cdot sin(\pi \cdot t - 3.017)$.

Appendix B Checklist for ECE 445 Final Report Authors

Note: Be sure to check even "automatic" features!

Pagination and margins

__✓__ Title page unnumbered (counts as i)

__✓__ Preliminary pages in lower case roman numerals

__✓__ Chapter 1 starts on Arabic page 1; all pages numbered consecutively after that; each chapter begins on new page

__✓__ Minimum one-inch margin on all sides of every page (page number falls slightly outside, which is OK)

Abstract

__✓__ On page ii

__✓__ Title same style as chapter titles, but unnumbered

__✓__ Presents main findings concisely and that is all

Table of contents

Format

__✓__ Preliminary material (abstract) *not* included

__✓__ Consistent capitalization

__✓__ Leader dots appear and page numbers aligned (automatic)

Agreement with text

__✓__ Wording of chapter titles and subheadings matches text exactly (automatic)

__✓__ Page numbers correct (automatic)

__✓__ Update your automatically generated content!

Figures

Placement

__✓__ Same page as first citation in text or first possible page after that

or

_____ Separate section (with tables) at end of each chapter

or

_____ Separate chapter (with tables) after Conclusion

__✓__ **Not** scattered among short passages of text

Figures (continued)

Numbering and citations

✓ ____ Every figure cited directly in text (e.g., "Figure 1 shows …")

✓ ____ Figures numbered in order of their citation in text

Quality

✓ ____ Information conveyed economically

✓ ____ Neat, legible, and within margins

✓ ____ Axes labeled

Captions

✓ ____ Every figure has descriptive caption (not just "Figure 1")

✓ ____ Caption below figure, use "Figure X (continued)" for multipage figures

Tables

Placement

✓ ____ Same page as first cited in text or first possible page after that

or

____ Separate section (with figures) at end of each chapter

or

____ Separate chapter (with figures) after Conclusion

Numbering and citations

✓ ____ Every table cited directly in text (e.g., "Table 1 shows …")

✓ ____ Tables numbered in order of their citation in text

Quality

✓ ____ Neat and legible

✓ ____ Decimals aligned

✓ ____ Column and row headers labeled, with unit symbols, if necessary

Titles

✓ ____ All tables have descriptive title (not just "Table 1")

✓ ____ Title *above* table, use "Table X (continued) for multipage figures

Equations

- ✓ Neat and legible, with proper use of italics and bold
- ✓ Centered or indented consistently
- ✓ Numbered in sequence and according to same scheme (whole number or single-decimal) as figures and tables, but in a sequence independent of figures and tables
- ✓ Use of parentheses both in display and in text citation
- ✓ Numbers are flush right

Appendices

- ✓ Appear before References *if* they contain reference citations
- ✓ Figures and tables numbered, with captions/titles, and cited in the text

References

- ✓ All references cited in the text, and every citation corresponds to an entry in References
- ✓ Numbered in order of citation in text
- ✓ Use of brackets and other IEEE style
- ✓ Use the template, and proofread!

Writing and style

- ✓ Quantities expressed with number, space, and correct unit symbol
- ✓ Abbreviations defined at first use and used consistently afterward
- ✓ Writing is neutral in tone, formal in style, and consistent from writer to writer
- ✓ Active voice used as much as possible
- ✓ Needless words omitted
- ✓ Every sentence clear and readable
- ✓ Read the paper aloud
- ✓ Ask a friend unfamiliar with the subject matter to read and comment