ECE 445: Senior Design

Design Document

Analog Computer ODE Solver

Team Number	26
Team Members	Zixuan Qu, Haige Liu, Dianxing Tang, Shilong Shen
NetIDs	zixuanq3, haigel2, dt12, shilong7
Sponsor	Said Mikki
TA	Lumeng Xu
Majors	Computer Engineering & Electrical Engineering

Contents

1 Introduction

In this document, a detailed technical explanation is made to demonstrate how our design came out and how we will realize it in few weeks. We will conduct a comprehensive analysis of our goals from theoretical support, previous research, feasibility analysis, and cost considerations of actually setting up the hardware equipment.

1.1 Proposal Overview

As we have mentioned in our proposal, 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) [16]. Therefore, we decided to build a "programmable" analog computer based on PCB. Its portability and computational efficiency enable it to solve many real-time computing problems and be applied in various fields, such as unmanned driving and communication decoding. Besides the capabilities of solving different kinds of ODEs [[4], [5], [6]], this computer could be also used to handle other types of equations and perform operations such as Taylor series expansion, 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.

1.2 Challenges & Solutions

1.2.1 Switch Design

- **Challenge**: Designing switches that can accurately and reliably change (program) the hardware circuit structure is a major challenge. The switches need to have low resistance when closed to avoid affecting the analog signals in the circuit and high resistance when open to ensure that the unwanted parts of the circuit are effectively isolated.

- **Solution**: We will used changable resistors that can be controlled by voltage input.

In a hardware concept, we might use a digital potentiometer or MOSFET to be the variable resistor.

1.2.2 Accuracy and Precision

- **Challenge**: Analog computations are prone to errors due to component tolerances, such as the tolerance of resistors and capacitors [1]. Ensuring that the analog computer can provide accurate solutions for a wide range of equations, especially non-linear ones, is difficult.

- **Solution**: In order to ensure the consistency of each kind of components, we will use high accuracy chips instead of building all components using basic op-amp. For instance, AD633 could be one option for the multiplier we use.

1.2.3 Soldering

- **Challenge**: We need to solder most chips on PCB by ourselves, so our first requirement for chips is that the chip should better be plug-in version. However, some of chips we selected are surface mount (no plug-in version available) chips which are very tiny to do hand soldering. For example the AD5254 chips we have purchased are smaller than 1cm x 1cm size.

- **Solution**: We might consider to find alternatives for plug-in chips that we could do hand soldering. For another, we will also try to contact factories that could help us do machine soldering if we have to use those tiny chips.

2 Design

2.1 Block Diagram

This project presents a microcontroller-based analog computing system, in which a multi-voltage power module supports signal interactions between the user interface, signal generation components, and analog computation blocks such as ODE solvers and Taylor series calculators.

Figure1 and figure2 show the top-level block diagram for signaling and powering between modules in the system.

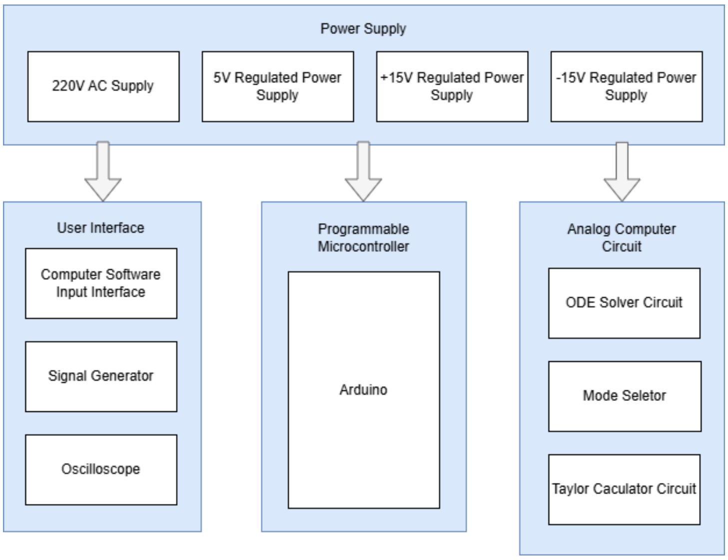

Figure 1: Top Level Power Layout

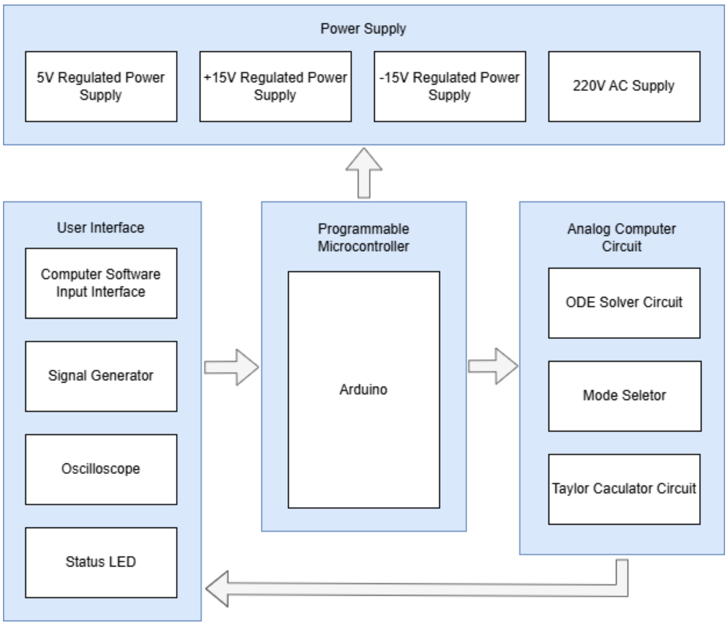

Figure 2: Top Level Signal Layout

2.2 Written Descriptions of each Block

2.2.1 Power Supply

The Power Supply module is designed to provide stable and reliable electrical energy to support the operation of all other modules within the analog computer system. It includes three primary regulated outputs derived from a standard 220V AC supply:

220V AC Supply This input directly supports the operation of user-interface equipment, specifically the signal generator and oscilloscope, enabling these instruments to perform their intended functions effectively.

±15V Regulated Power Supply This dual-output regulated power supply provides stable +15V and -15V voltages essential for the operational amplifiers and associated analog circuitry.

5V Regulated Power Supply A dedicated 5V regulated output is provided to power the Arduino programmable microcontroller. This stable power source is crucial for ensuring consistent operation, data processing, and interfacing tasks executed by the microcontroller.

2.2.2 User Interface

The User Interface module enables users to input computational problems and monitor the results effectively. Users initially provide the problem parameters—including problem type, equation coefficients, initial conditions, and the type of Taylor function—via the computer software input interface. These parameters are then transmitted to the Arduino microcontroller for further processing. Additionally, function signals such as the input function for ODEs (denoted as ”u” functions) or Taylor calculation inputs are generated through the signal generator and directly sent as analog signals to the Analog Computer circuit. Finally, the computed results from the analog circuits are returned and displayed on the oscilloscope, providing users with real-time visualization and verification of the computational outcomes.

2.2.3 Programmable Micro-controller

The programmable microcontroller module utilizes an Arduino, a versatile and widely-used open-source hardware platform, combined with an integrated development environment (IDE) for software programming. Users program the Arduino using a straightforward software interface, enabling automatic processing of computational inputs such as problem types, equation coefficients, initial conditions, and the selection of Taylor functions. Based on these inputs, the Arduino generates digital signals to execute precise control over the analog computer's circuit operations, including mode selection, adjusting variable resistances, setting initial voltage conditions, and initiating the operation of the analog circuits. By automating these crucial tasks, the programmable microcontroller enhances accuracy, repeatability, and ease of use within the analog computation system.

2.2.4 Analog Computer Circuit

The Analog Computer Circuit module is the core computational component responsible for performing real-time analog calculations based on the input parameters and control signals. This module includes multiple sub-circuits, such as the ODE solver and the Taylor series calculator, each capable of handling specific mathematical operations.

Mode Selector To enable flexible configuration of computational modes (e.g., switching between ODE solving and Taylor function evaluation), we implement MOSFETs (Metal-Oxide-Semiconductor Field-Effect Transistors) as electronic switches. These MOSFETs are controlled by digital signals from the Arduino, allowing precise and fast switching between different computational paths without mechanical intervention.

ODE Solver Circuit The ODE Solver Circuit is designed to simulate and solve ordinary differential equations (ODEs) in analog form using a network of operational amplifier (op-amp) components. We construct custom analog computing modules using op-amps to fulfill different mathematical operations:

- **Inverting Amplifier**:Used to simulate multiplication of a signal by a negative constant.
- **Adder-Subtractor Circuit**: Combines multiple input signals to perform addition and subtraction as required in ODEs Figure3.
- **Integrator Circuit**: Performs the analog equivalent of integration to recover original signals from their derivatives, essential in reconstructing ODE solutions Figure4.

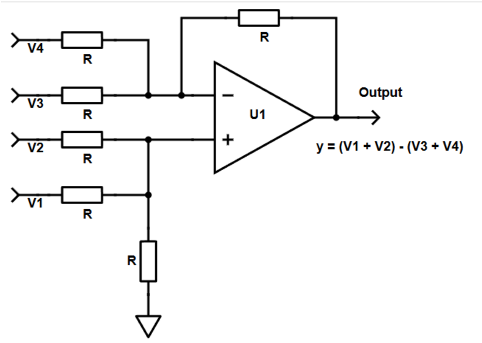

Figure 3: Op-amp Adder-Subtractor Component Circuit

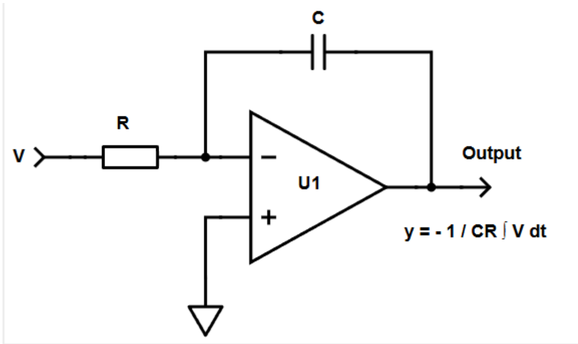

Figure 4: Op-amp Integrator Component Circuit

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, figure5 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.

To handle non-homogeneous equations, we incorporate analog multipliers in cases where signal multiplication is necessary.

Ultimately, we aim to implement a third-order ODE solver. Figure6 presents the complete schematic of our final third-order ODE solver circuit, integrating all key components and supporting high-order modeling capability.

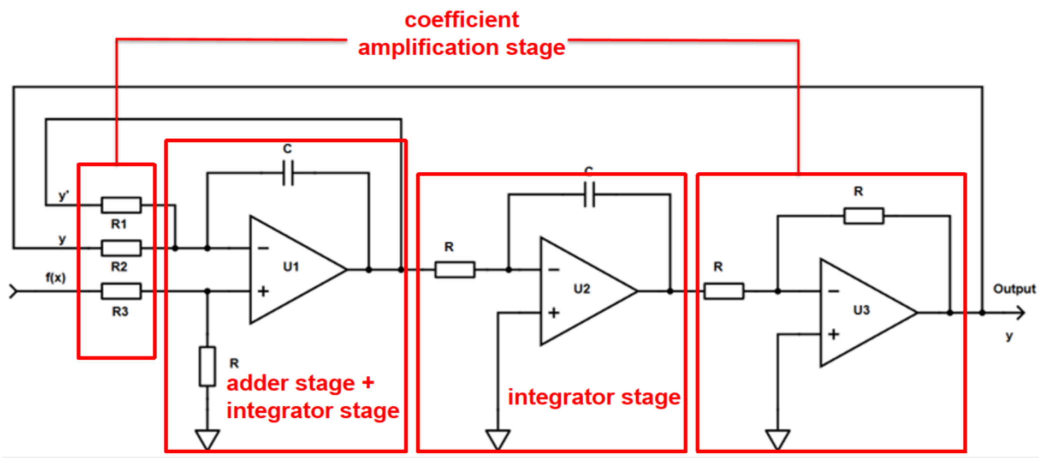

Figure 5: An example of circuit implementation solves the certain type of ODEs: $y'' = ay' + by - cu$

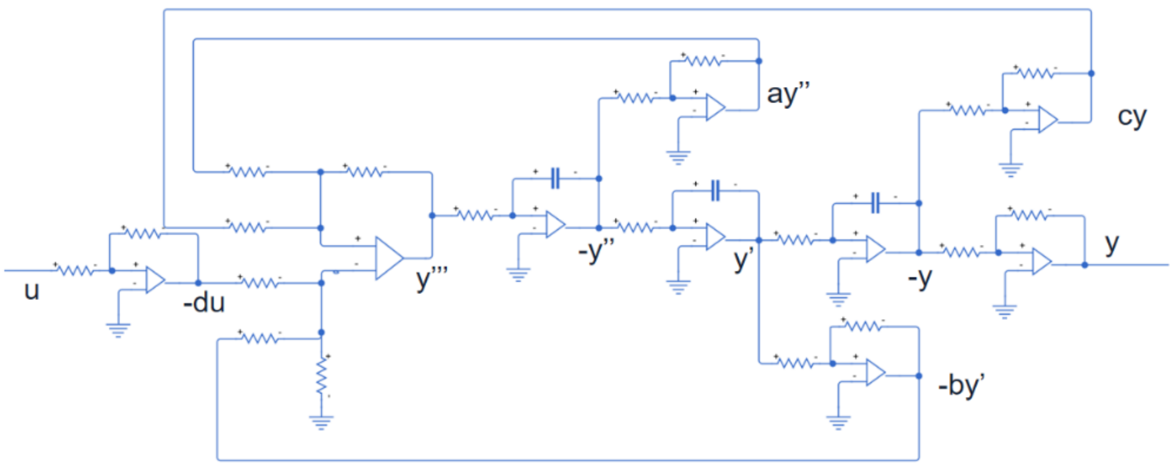

Figure 6: third-order ODE solver circuit

Taylor Calculator Circuit The Taylor Calculator Circuit is designed to approximate nonlinear functions using their fourth-order Taylor series expansions. This approach enables real-time analog computation of mathematical functions such as sin(x), cos(x), and exp(x), based on a given input signal.

In this circuit, we expand the target function f(x) around x = 0 as:

$$f(x) \approx f(0) + f'(0)x + \frac{f''(0)}{2!}x^2 + \frac{f^{(3)}(0)}{3!}x^3 + \frac{f^{(4)}(0)}{4!}x^4$$

We implement this approximation in hardware using operational amplifiers configured as multipliers, adders, and inverting amplifiers. Each term in the Taylor expansion corresponds to a separate analog stage that calculates powers of x (input signal), scales them by the precomputed coefficients (such as $\frac{1}{2}!$, $-\frac{1}{3}!$, etc.), and finally sums them to produce the approximated output signal.

As shown in Figure7, the input signal is first routed through a function selector (controlled digitally by the Arduino) which determines whether the function to approximate is sin(x), cos(x), or exp(x). Each branch includes its own set of coefficient multipliers corresponding to the Taylor expansion terms of the selected function. After computing and scaling entries, the terms are added together using a summing amplifier to generate the final analog output.

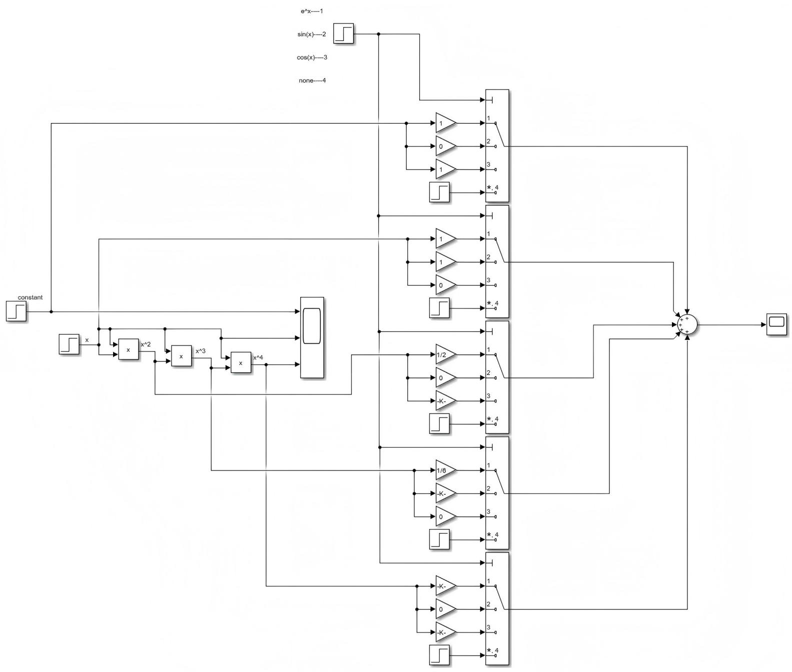

Figure 7: Taylor calculator circuit

2.3 Flowchart

The overall operation process of the analog computer system is illustrated in Figure8 as a flowchart, detailing how mode selection and signal processing are managed step-by-step.

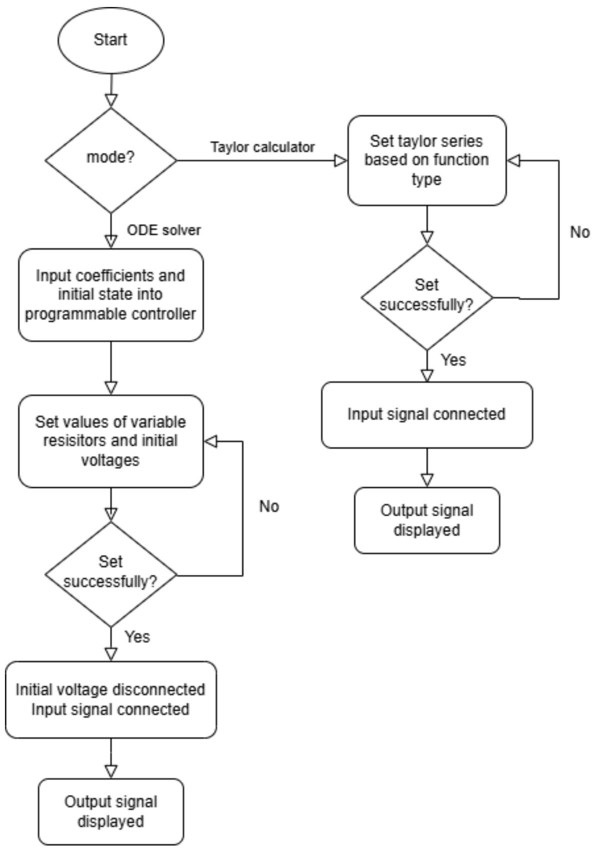

Figure 8: Flowchart for the Analog Computer

2.4 Calculations and Simulations

To validate the behavior of our second-order ODE solver, we conducted a SPICE-based simulation using Multisim. The simulation waveform is shown in Figure9, where the blue curve represents the input sine signal and the yellow curve shows the output response.

As expected from theoretical analysis, the output is also a sinusoidal waveform but with a clear phase shift and a reduced amplitude compared to the input. This matches the typical behavior of second-order linear systems governed by differential equations of the form:

$$y'' + a \cdot y' + b \cdot y = c \cdot u(t) \tag{1}$$

The results confirm that our analog implementation correctly simulates the desired differential relationship, verifying the effectiveness of the integrator, inverter, and adder components in the circuit.

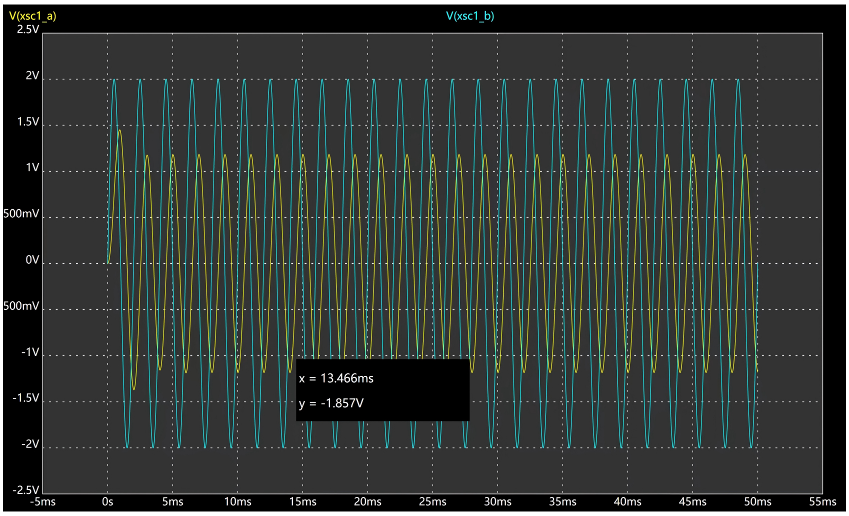

Figure 9: Simulation of Second Order ODE Solver

3 Requirements & Verifications

3.1 Adder-subtractor Circuit

3.1.1 Requirements

- The circuit will generate the sum of input signals when receiving high control voltage (4.5V-5.5V) from the microcontroller, which is allowed an error within 10
- The circuit will generate the difference of input signals when receiving low control voltage (0V-0.5V) from the microcontroller, which is allowed an error within 10

3.1.2 Verifications

- Connect the circuit as shown in Figure 3. Provide high control voltage and several groups of constant signals. Record the output signal.
- Provide a low control voltage and several groups of test signals. Record the output signal.
- If the output signal is out of the desired range, adjust the resistor value.

3.2 Integrator Circuit

3.2.1 Requirements

- The circuit will compute the integrator of the input signal in real time, which is allowed an error within 10

3.2.2 Verifications

- Connect the circuit as shown in figure 4. Provide several types of input signal, including sin function and square function. Record the input and output signals over a certain period of time simultaneously.
- If the output signal is out of the desired range, adjust the value of the resistor and capacitor.

3.3 Multiplier Circuit

3.3.1 Requirements

- The circuit will calculate the product of the input signal, which is allowed an error within 10

3.3.2 Verifications

- Connect the circuit and provide several groups of constant inputs (from 0.5V to 10V). Record the output signals.
- If the output signal is out of the desired range, adjust the value of the resistors that determine the proportion.

3.4 Mode selector circuit

3.4.1 Requirements

- The circuit will determine the mode we will calculate. Assume 1 is high voltage (4.5V-5.5V), 0 is low voltage (0V). The circuit will receive a 3-bit digital signal from the microcontroller. The first 2-bit controls the order number (00-first-order ODE, 01- second-order ODE, 10-third-order ODE).
- Last 2-bit controls Taylor use. 00 means not connecting Taylor circuit. 01 means calculating sin function, 10 means calculating cos function, 11 means calculating exp function.

3.4.2 Verifications

- Connect the circuit of thet of the mode selector. Replace the ODE solver circuit with constant voltage (first-order ODE circuit output - 1V; second-order ODE circuit output - 2V; third-order ODE circuit output - 3V). Provide these first 2-bit signals 00, 01, 10 successively. Record output voltages.
- Provide last 2-bit signal 00, 01, 10. 11 successively. Record the output voltages.
- If the output voltage is out of the desired range, check the corresponding switch and its connection.

3.5 Taylor calculator circuit

3.5.1 Requirements

- The circuit will calculate the Taylor expansion of up to five terms. Ensure there are no errors in the calculation of x^2, x^3, x^4 and products of their coefficients. Each of them are allowed an error within 10

3.5.2 Verifications

- Connect the circuit as shown in figure 7. Provide different values of x and coefficients. Record x^2, x^3, x^4 and products of their coefficients. If the output voltage is out of the desired range, check for any connection problems.

4 Tolerance Analysis

In the design of the analog computer for solving ODEs and performing Taylor series calculations, tolerance analysis plays a vital role. Component tolerances, particularly those of resistors and capacitors, can have a profound impact on the overall performance of the system.

Resistors are used extensively in various circuit components such as amplifiers, adders, and integrators. Their tolerance values can directly affect the gain of amplifiers. For instance, in an inverting amplifier used in the ODE solver circuit, if the resistor tolerance causes a deviation in its resistance value, the amplification factor will change. This alteration can lead to incorrect scaling of signals, which in turn distorts the results of ODE solutions. Similarly, capacitors' tolerance can affect the time constants in integrator circuits. Inaccurate time constants will result in improper integration, causing significant errors in reconstructing the original signals in ODE solving.

To ensure the system's accuracy, a meticulous analysis of each component's tolerance is essential. This involves studying the datasheets of components to understand their specified tolerance ranges. After that, we must calculate the acceptable deviation range for signal inputs and component connections. For example, we need to determine how much variation in input signals can be tolerated without sacrificing the accuracy of the computational results.

Finally, it is crucial to verify the system's performance at the tolerance limits. By testing the analog computer with components at their maximum and minimum tolerance values, we can identify potential issues. If the system fails to meet the accuracy requirements under these extreme conditions, we may need to either select higher - precision components or modify the circuit design. This comprehensive approach to tolerance analysis will help us build a reliable and accurate analog computer.

5 Cost Analysis

The main cost of this project is on circuit components. It including PCB board, resistors, multipliers, adders, integrators and so on. And a small part example is listed below (which means it could be more if we want to make a larger one with more functionalities).

Device Name	Quantity	Cost
Multiplier	3	100 RMB
PCB	1	50 RMB
Inverting Amplifier	20	100 RMB
Adder	1	20 RMB

Table 1: A simple model device summary

The approximately total cost of a simple version of design will be 300 RMB. This could be added if we encounter some problems like broken chips and upgraded chips.

6 Schedule

So far, we have done software level simulations and decided the exact type of ODEs and other related functions we will realize. Our schedule for the whole project is listed as following:

6.1 Preliminary Work

- **Mar.10 - Mar.24** A basic understanding and research about the related theorem of the project: ODE, Analog Computer, ODE solver and etc. Making a general goal for our design and finishing the proposal.
- **Mar.25 - Mar.31** Theoretical analysis about the top-level design of Analog Computer, dividing the system into a different modules and build software level (both

mathematical and chip-level software simulation using Simulink and ICEDA) modular simulation.

- **Apr.1 - Apr.7** Complete the simulation testing and determine the type of ODE we need to solve. Do research about the physical chips, select and test purchased chips. Start to draft design document.

- **Apr.8 - Apr.14** Debug the hardware chips and do modular test for physical circuit. Searching for alternatives of current SMT-need chips. Finish the design document.

6.2 Future Plan

- **Apr.15 - Apr.21** Contacting the factory for SMT service if necessary to solder SMT-need chips on PCB and hand-solder plug-in chips to test modular circuit in labs.

- **Apr.22 - May. 4** Testing physical modules like integrator, multiplier, op-amp, changeable resistors like MOSFET or potentiometer, switching systems and so on. Integrate the modules to be a final version PCB, an first version analog computer will be built.

- **May.5 - May.11** If there exists bug in the circuit, debug and refine the circuit. Preparing datasheet and related documents of our analog computer. Preparing to do the mock demo and final demo.

- **May.12 - May.26** Writing final report and preparing for final presentation. The design part should be all done.

It should be noted that this schedule is a ideal one if every step goes fluently. Some real details may be slightly different from this ideal assumption.

7 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.

7.1 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.

7.2 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.

References

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

[2] G.E.R. Cowan, R.C. Melville, and Y.P. Tsividis. "A VLSI analog computer/digital computer accelerator". In: *IEEE Journal of Solid-State Circuits* 41.1 (Jan. 2006). Conference Name: IEEE Journal of Solid-State Circuits, pp. 42–53. ISSN: 1558-173X. DOI: 10.1109/JSSC.2005.858618. URL: https://ieeexplore.ieee.org/document/1564344/?arnumber=1564344 (visited on 02/23/2025).

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

[4] T. ElAli et al. "An Analog Computer to Solve any First Order Differential Equation". en. In: *Advances in Computer, Information, and Systems Sciences, and Engineering*. Ed. by Khaled Elleithy et al. Dordrecht: Springer Netherlands, 2006, pp. 43–46. ISBN: 978-1-4020-5261-3. DOI: 10.1007/1-4020-5261-8_8.

[5] T. ElAli 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*. Ed. by Tarek Sobh et al. Dordrecht: Springer Netherlands, 2007, pp. 449–451. ISBN: 978-1-4020-6266-7. DOI: 10.1007/978-1-4020-6266-7_81.

[6] T. ElAli et al. "Analog Computer to Solve Third-Order Linear Differential Equation". en. In: *Novel Algorithms and Techniques In Telecommunications, Automation and Industrial Electronics*. Ed. by Tarek Sobh et al. Dordrecht: Springer Netherlands, 2008, pp. 248–250. ISBN: 978-1-4020-8737-0. DOI: 10.1007/978-1-4020-8737-0_44.

[7] Alireza Fasih et al. "New computational modeling for solving higher order ODE based on FPGA". In: *2009 2nd International Workshop on Nonlinear Dynamics and Synchronization*. ISSN: 2324-8335. July 2009, pp. 49–53. DOI: `10.1109/INDS.2009.5227969`. URL: `https://ieeexplore.ieee.org/document/5227969/?arnumber=5227969` (visited on 02/23/2025).

[8] Guy Fernando. *Analogue Computing*. en. Section: Analogue Computing. Oct. 2021. URL: `https://www.i4cy.com/analog_computing` (visited on 02/23/2025).

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

[10] Claude Leibovici. *Answer to "Numerical methods (for ODE/PDE) that could take approximate solutions/good initial guesses, and further refine it to an certain accuracy"*. July 2015. URL: `https://math.stackexchange.com/a/1360379` (visited on 02/23/2025).

[11] Bruce J MacLennan. "A review of analog computing". In: *Department of Electrical Engineering & Computer Science, University of Tennessee, Technical Report UT-CS-07-601 (September)* (2007), pp. 19798–19807.

[12] Ross Millikan. *Answer to "Numerical methods (for ODE/PDE) that could take approximate solutions/good initial guesses, and further refine it to an certain accuracy"*. July 2015. URL: `https://math.stackexchange.com/a/1360211` (visited on 02/23/2025).

[13] Nate. *Numerical methods (for ODE/PDE) that could take approximate solutions/good initial guesses, and further refine it to an certain accuracy*. Forum post. July 2015. URL: `https://math.stackexchange.com/q/1359923` (visited on 02/23/2025).

[14] Transistor Overlord. *Answer to "Numerical methods (for ODE/PDE) that could take approximate solutions/good initial guesses, and further refine it to an certain accuracy"*. Mar. 2017. URL: `https://math.stackexchange.com/a/2211605` (visited on 02/23/2025).

[15] Dhruv Shah et al. "DPAC: Digitally Programmable Analog Computer". In: URL: https://api.semanticscholar.org/CorpusID:212619098.

[16] Claude E Shannon. "Mathematical theory of the differential analyzer". In: *Journal of Mathematics and Physics* 20.1-4 (1941), pp. 337–354.

[17] Jiayang Wu et al. "On-Chip Tunable Second-Order Differential-Equation Solver Based on a Silicon Photonic Mode-Split Microresonator". In: *Journal of Lightwave Technology* 33.17 (Sept. 2015). Conference Name: Journal of Lightwave Technology, pp. 3542–3549. ISSN: 1558-2213. DOI: 10.1109/JLT.2015.2442911. URL: https://ieeexplore.ieee.org/document/7122249/?arnumber=7122249 (visited on 02/23/2025).