

Analog Calculator

6.101 Spring 2018

Aditya Gopalan

May 17, 2018

Analog computation is often quicker than digital computation but comes with the tradeoff of being less accurate. The goal of this work is to design an analog calculator capable of working within reasonable tolerances. All circuits were based on amplifiers using standard operational amplifier configurations and performed to within the specified tolerances. For robust applications requiring quick calculation, the ability to do instant computation seems to allow for greater flexibility and speed of computational algorithms.

1 Introduction

Numerical optimization is a powerful with wide-ranging applications from theory (algorithm development), robotics, traffic management, and other fields. Often applications are implemented in software, and can take upwards of minutes, or even hours to solve a single problem. For example, the robotics optimization in [1] takes about 12 minutes to run with the implementation in the paper. While [1] uses a powerful control method, the lack of speed means it can only be used for pre-planned open loop control and cannot be used for real-time feedback control. [2] discusses implementing analog circuits to solve linear and quadratic problems, with near instantaneous results (of course, the time spent building a large circuit is likely longer than that spent encoding a large optimization problem in software).

This work presents an analog calculator that performs the following binary operations:

- Addition
- Subtraction
- Multiplication
- Division
- Logarithms
- Exponentiation.

The rest of the work is organized as follows: in Section 2, circuit diagrams and transfer functions for the aforementioned binary operations are presented; in Section 3, a discussion of accuracy results compared against goals for the project and remarks about the process; and in Section 4, concluding remarks.

2 Circuit Diagrams and Transfer Functions

2.1 Addition

See Figure 1 for the addition circuit diagram. This circuit is comprised of an inverting adder and then an inverting buffer. The rails voltages for V^+ and V^- are $\pm 15\text{V}$, respectively. The transfer function is:

$$V_{\text{out}} = V_1 + V_2. \quad (1)$$

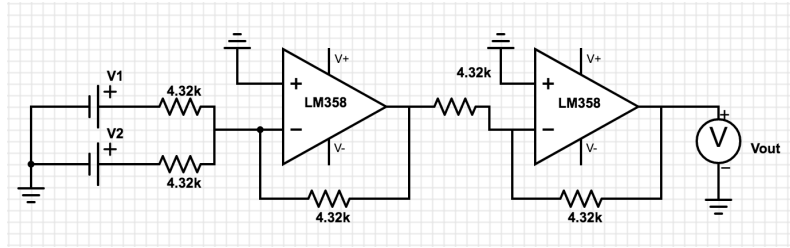


Figure 1: Circuit diagram for addition

2.2 Subtraction

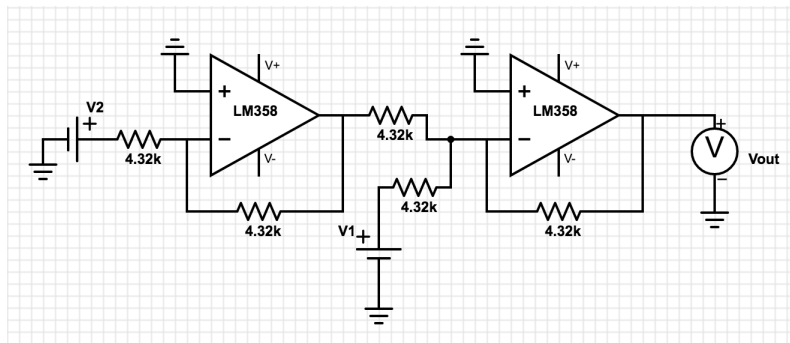


Figure 2: Circuit diagram for subtraction

See Figure 2 for the subtraction circuit diagram. This circuit consists of an inverting buffer followed by an inverting adder. The rails voltages for V^+ and V^- are $\pm 15V$, respectively. The transfer function is:

$$V_{out} = V_1 - V_2. \quad (2)$$

2.3 Multiplication

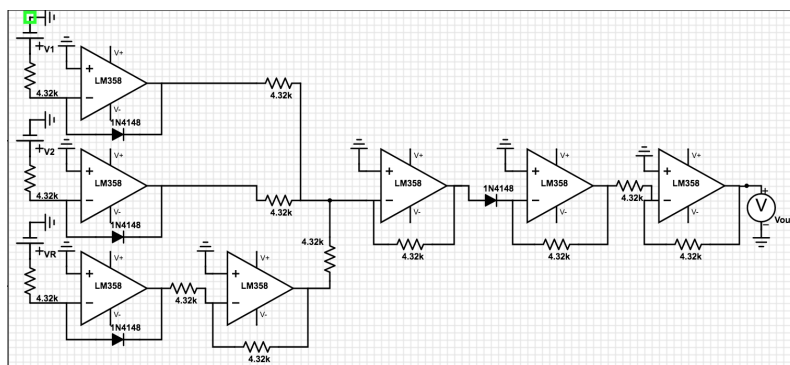


Figure 3: Circuit diagram for multiplication with a reference voltage

See Figure 3 for the multiplication circuit diagram. This circuit consists of an inverting logarithmic amplifiers of the two input voltages, and a non-inverting logarithmic amplifier on a reference voltage. The three voltages are then fed into an inverting adder and then a non-inverting exponential amplifier. **All reference voltages are to be tuned with a potentiometer until output readings are accurate.** The rails voltages for V^+ and V^- are $\pm 15V$, respectively. The transfer

function is:

$$V_{\text{out}} = \frac{V_1 V_2}{V_R}. \quad (3)$$

This project used the formulation of multiplication with a reference voltage, but multiplication was attempted with the below schematic which does not include a reference voltage.

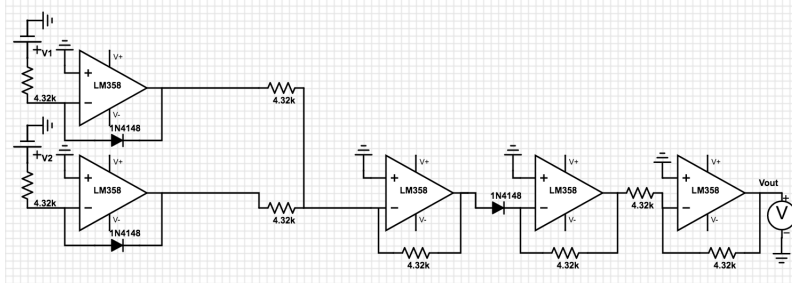


Figure 4: Circuit diagram for multiplication without a reference voltage.

See Figure 4 for the multiplication circuit diagram without a reference voltage. This circuit is identical to 3 except for the reference voltage. The rails voltages for V^+ and V^- are $\pm 15\text{V}$, respectively. The transfer function is:

$$V_{\text{out}} = \frac{V_1 V_2}{I_S R}. \quad (4)$$

2.4 Division

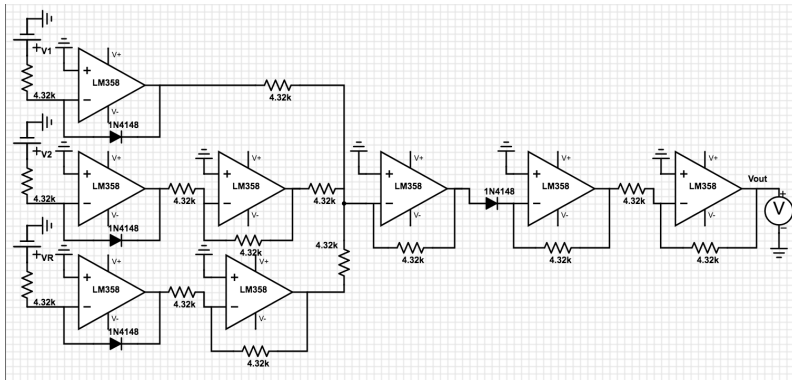


Figure 5: Circuit diagram for division

See Figure 5 for the subtraction circuit diagram. This circuit is identical to multiplying with a reference voltage except for an additional invert on V_2 to produce subtraction instead of addition inside the logarithmic calculation. The rails voltages for V^+ and V^- are $\pm 15\text{V}$, respectively. The transfer function is:

$$V_{\text{out}} = \frac{V_1}{V_2 V_R}. \quad (5)$$

2.5 Logarithm

For two inputs V_1, V_2 , pass each through the following conditioning circuit: This circuit takes an inverting logarithm on the input and subtracts from it the logarithm of the reference voltage (V_R should be the same as in division). After that, the two conditioned signals are passed into the division circuit in Section 2.4. The transfer function is:

$$V_{\text{out}} = \log_{V_1} V_2. \quad (6)$$

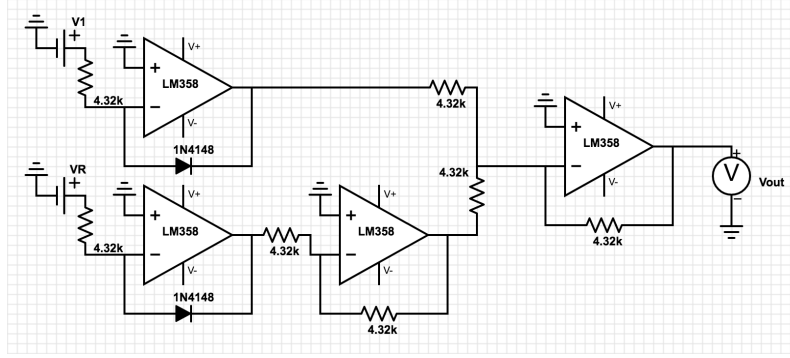


Figure 6: Circuit diagram for logarithm conditioning

3 Results

3.1 General Remarks

The datasheet for the LF356 operational amplifier suggests its use in logarithmic amplifiers. However, cascading a non-inverting logarithmic amplifier with a non-inverting exponential amplifier only seemed to consistently produce an error of around 30% between 0.1V and 10V. The LM6132 precision amplifier was also tested; while it had less than 1% error around 1V, it had about 50% error for 0.1V and 10V. It is not clear why the LM358 operational amplifier outperforms the other two amplifiers by such a significant margin, but performance with the LM358 is noticeably better. Cascading the non-inverting logarithmic and exponential amplifiers produced about a 1% error for the entire range of 0.1V-10V.

3.2 Addition

The goal error for addition was 5% with a stretch goal of 2%.

V_1 [V]	V_2 [V]	V_{out} [V]	Error[%]
0.0012mv	0.0013mv	0.0025	0
0	0.102	0.101	0.98
0	0.495	0.495	0
0	0.988	0.989	0.1
0.995	0.988	1.98	0.15
0.995	1.98	2.96	0.50
0.995	2.96	3.95	0.13
0.995	3.95	4.93	0.30
0.995	4.93	5.92	0.084
0.995	5.92	6.90	0.22
0.995	6.91	7.89	0.19
0.995	7.89	8.87	0.17
0.995	8.88	9.86	0.15
0.995	9.87	10.84	0.23

3.3 Subtraction

The goal error for subtraction was 5% with a stretch goal of 2%.

V_1 [V]	V_2 [V]	V_{out} [V]	Error[%]
0.0005	0.002	-0.0098	550
0.0992	0.002	0.089	8.4
0.494	0.002	0.483	1.8
0.987	1.00	-0.027	110
1.97	1.00	0.956	1.4
2.95	1.00	1.94	0.51
3.94	1.00	2.93	0.34
4.93	1.00	3.91	0.51
5.91	1.00	4.90	0.20
6.90	1.00	5.88	0.34
7.89	1.00	6.87	0.29
8.88	1.00	7.85	0.38
9.86	1.00	8.83	0.34

The errors in the first few subtractions seem alarmingly large; however they are computed on values near zero and any error at all around zero is guaranteed to be large. The values look qualitatively accurate to the same absolute error as in the rest of the computations.

3.4 Multiplication

The goal error for multiplication was 10% with a stretch goal of 5%.

V_1 [V]	V_2 [V]	V_{out} [V]	Error[%]
0.002	0.987	0.0048	140
2.00	0.99	1.94	2.0
2.00	0.494	0.975	1.3
2.00	0.986	1.95	1.1
2.00	1.97	3.87	1.8
2.00	2.96	5.79	2.2
2.00	3.92	7.69	1.9
2.00	4.93	9.59	2.7
2.00	5.92	11.44	3.4

Implementation of the multiplication circuit required careful analysis. Primary among the challenges involved was that the resultant output displayed unwanted oscillatory behavior. This was resolved by adding bypass capacitors to both the positive and negative supply rails of each op-amp involved in the computation. In addition, despite the current draw being less than 0.01A, unity-gain voltage buffers were added between stages for increased resilience to voltage swings. All voltage buffers were LM358N operational amplifiers.

Two methods were considered for computing multiplications. The first was used to collect data and used the reference voltage. However, without that voltage, the overall transfer function from V_1, V_2 to V_{out} is

$$V_{\text{out}} = \frac{V_1 V_2}{I_S R}. \quad (7)$$

Because I_S is presumed to be a constant, a reasonable approach to implementing this transfer function is to choose

$$R = \frac{1}{I_S}. \quad (8)$$

For this solution, 1N43A (Germanium) diodes were used in place of the 1N4148 diodes in all computations involving logarithms and exponentials. When measured in an *M3* device, the reverse saturation currents of the 1N43A diodes ranged from 3.6mA to 7.4mA. The inconsistency from device to device seems to be the reason that cancelling I_S with an appropriate resistor did not work; however it is worth noting that cascading a non-inverting logarithmic amplifier with a non-inverting exponential amplifier seemed to produce an accurate (unity) gain despite those inconsistencies.

3.5 Division

The goal error for division was 10% with a stretch goal of 5%.

$V_1[V]$	$V_2[V]$	$V_{out}[V]$	Error[%]
0.0003	1.00	0.001	67
0.987	1.00	0.986	0.1
1.97	1.00	1.95	1.0
2.96	1.00	2.91	1.7
3.94	1.00	3.87	1.8
4.93	1.00	4.90	0.61
5.91	1.00	5.88	0.51
6.91	1.00	6.93	0.29
7.89	1.00	7.94	0.63
8.88	1.00	9.00	1.4
9.86	1.00	9.95	0.91
0.493	0.990	0.512	2.8
0.0998	0.990	0.105	4.2

3.6 Logarithm

The goal error for logarithms was 20% with a stretch goal of 10%.

$V_1[V]$	$V_2[V]$	$V_{out}[V]$	Error[%]
1.97	1.99	1.19	21
3.95	1.99	2.06	3.2
7.89	1.99	2.83	5.7
8.88	3.01	2.07	4.4

It is worth noting here the discrepancy in the errors – namely that $\log_{1.99} 1.97$ has a significantly larger error than the other three computations. This is due to the fact that we subtract a reference voltage of approximately 1V from each of the two inputs, resulting in the computation of $\log 1$, which is approximately 0V. Proceeding to divide two values that are both approximately 0V as in the circuit diagrams is difficult to do without noise, hence the larger error in this computation.

3.7 Exponentiation

The goal error for exponentiation was 20% with a stretch goal of 10%.

A proposed exponentiation function works as follows for inputs V_x, V_y .

1. Pass V_x through a non-inverting logarithmic amplifier (output $V_{\log,x}$)
2. Multiply $V_{\log,x}$ by V_y (output $V_y V_{\log,x}$)
3. Pass 1V through a non-inverting logarithmic amplifier (output 1_{\log})
4. Multiply 1_{\log} by $V_y - 1$ (output $(V_y - 1)1_{\log}$)
5. Subtract the result of Step 4 from the result of Step 2
6. Pass the result of Step 5 into a non-inverting exponential amplifier.

When calculating the expression 2^2 , the result of the first multiplication in Step 2 above, $(2 \log 2)$ should have been 1.16V. Tuning the other voltage (Step 4) with a power supply, a value of 1.17V outputted more than 5V (the output was not steady), and a value of 1.15V outputted less than 3V (the output was not steady). Because an 0.01V error is well within a 1% tolerance for that calculation, it seems that an exponentiation function is not achievable with this method.

4 Conclusion

The difficulty of analog computation is the trade-off between speed, accuracy, and sensitivity. For applications requiring approximate (but consistent over a wide range of values) computations, it seems that this approach to an analog calculator is a reasonable solution as long as arbitrary exponentiation is not needed. For example, should applications using the results of [1] have perfect sensor readings, a functional analog calculator may allow for real-time optimal control (since any

robustness to account for sensor inaccuracies could then be used to account for inaccuracies in the analog controller). However, despite the speed of these calculations, accuracy is indeed often necessary and the implementation of this calculator is significantly less accurate than a simple calculator obtainable at an office supply store, while also requiring parts that cost significantly more.

References

- [1] Aditya Gopalan, Eric Cousineau, and Russ Tedrake. An implicit trajectory optimization formulation for rigid bodies undergoing elastic collision. *International Conference on Intelligent Robots*, 2018. Under Review.
- [2] Sergey Vichik and Francesco Borrelli. Solving linear and quadratic programs with an analog circuit. *Computers & Chemical Engineering*, 70:160–171, 2014.