Analog Computers

Background:

Analog computers are devices which solve a differential equation. Some of the first analog computes were used by the Royal Navy in 1902 for the fire control on battleships. These were mechanical devices that would take into account the movement of the firing ship and the target ship to generate range and deflection of the battleship's main guns. During WWII, mechanical analog computers were incorporated into U.S. battleship firing mechanisms. In 1942, this allow the U.S. Iowa to fire at and hit a target ship 23 miles away (!).

Analog computers were also used to model and solve calculation problems related to electrical utilities as far back as 1929. Here, dynamic models of large distributed utilities needed to be solved.

In the 1950's, electronic analog computers started replacing mechanical analog computers. Using operational amplifiers, circuits which solve differential equations can be built and operated for far less money and in less time than their mechanical counterparts. Initially, the electronic analog computers were still fairly large and expensive such as the RCA Typhoon analog computer which included 4000 electron tubes, 100 dials, and 6000 plug-in connectors

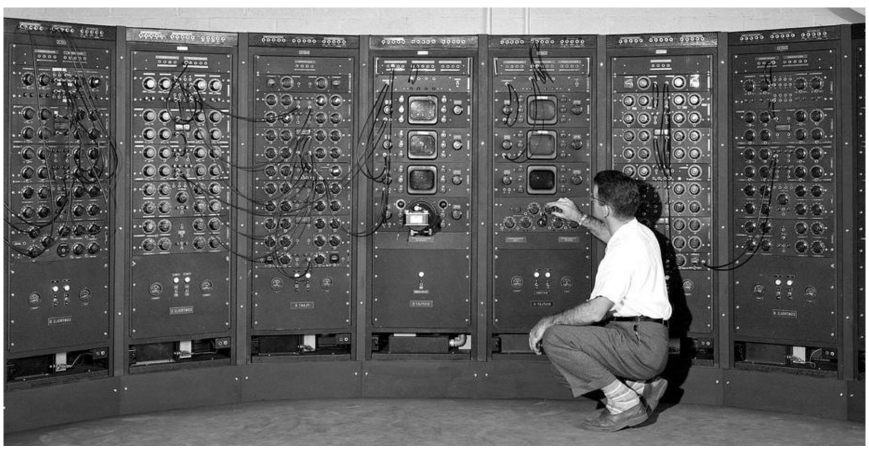

Figure 1: Analog computer from 1950.

Educational analog computers based on electronics came about in the 1960s with Heathkit, for example, selling an educational analog computer for $199 in 1960 ($2,980 in 2025 dollars). These analog computers had operational amplifiers which could be configured as integrators and summing amplifiers. The gains of these devices could be adjusted with potentiometers. Jumper wires then allowed you to change how the circuit was configured - similar to what is covered in this lecture.

In industry, analog computers were used to model jet engines in the 1980s. When running tests on an actual jet engine, the test cell costs $10,000 per minute to operate. If the test results in the destruction of a jet engine, the cost can exceed $4 million. Before you ever test your controller on an actual engine, it would be tested on an analog computer simulation of that jet engine. The advantage of the analog computer is that it costs just pennies to operate. If your controller fails, it only results in the voltage slamming to the rails and no harm is done. Once a controller has proven itself on the electronic model of the jet engine, then *maybe* you'd get permission to try it on an actual jet engine. Around 1990, the analog computers were replaced with laptop computers running numerical simulations of the jet engine. This was a far less expensive option - plus you could make multiple copies of the software and run tests in multiple labs.

One of the advantages of analog computers, is unlike their digital counterparts, they do not crash. You never get a divide-by-zero error. You never get stuck in an infinite loop. You never get an 'illegal operation' error. They just run. For example, when the first space shuttles were built, the flight controller relied on three computers in a tripply-redundant best of three voting scheme. Two of the computers were digital computers (similar to a laptop computer). The third was an analog computer solving differential equations using op-amps, capacitors, and resistors. By the time the third shuttle was built, confidence in the digital computers was enough that the analog computers were replaced with digital counterparts.

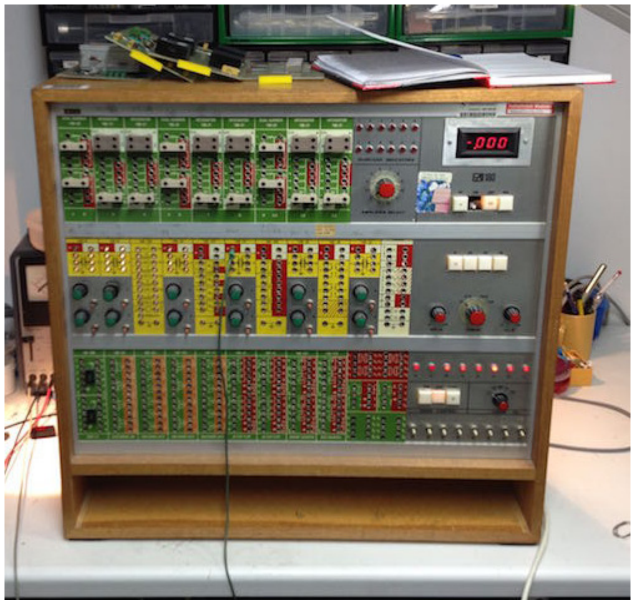

Figure 2: educational analog computer - circa 1970

In this lecture, a procedure to solve a generic differential using an op-amp circuit is presented. This ability to solve differential equations is the heart of an analog computer.

Building Blocks of Analog Computers

Let's start with basic circuits used in analog computers. These are

- Summing Inverting Amplifiers, and
- Summing Integrators

The summing inverting amplifier usually used with analog computers is the inverting amplifier shown in Figure 3.

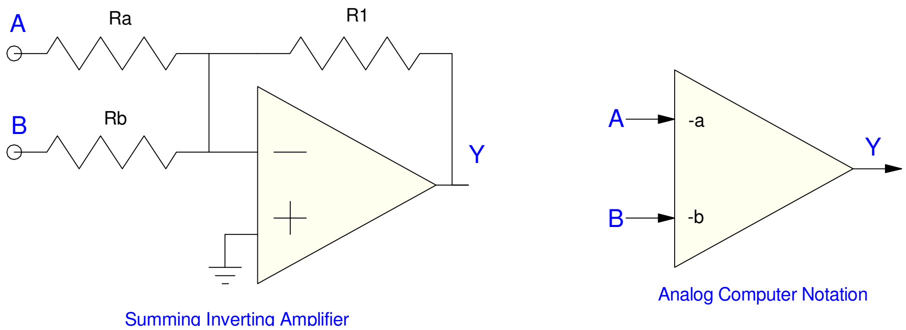

Figure 3: Summing inverting amplifier and its analog computer symbol

For this circuit, the output is the sum of the inputs (with a negative gain)

$$-Y = aA + bB$$

where

$$a = \left(\frac{R_1}{R_a}\right)$$

$$b = \left(\frac{R_1}{R_b}\right)$$

The second circuit used with analog computers is the summing inverting integrator shown in Figure 4.

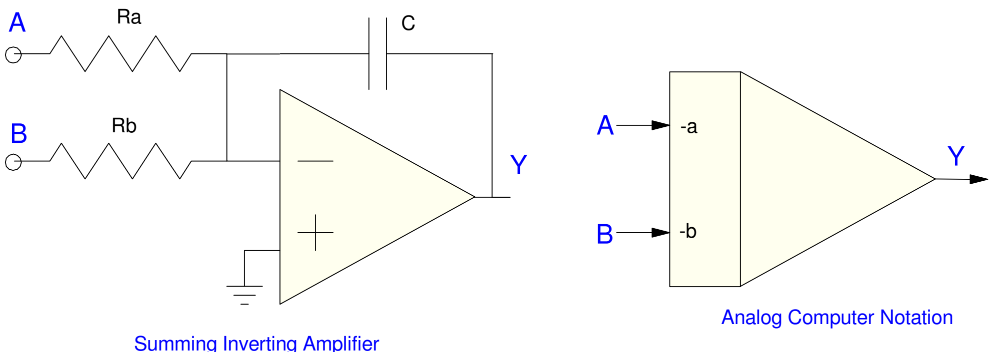

Figure 4: Summing inverting integrator and its analog computer symbol

For this circuit

$$-sY = aA + bB$$

$$a = \left(\frac{1}{R_a C}\right)$$

$$b = \left(\frac{1}{R_c C}\right)$$

The net result is that

- If you can express a given differential equation in terms of inverting summing amplifiers and inverting summing integrators,

- Then you can implement that differential equation using op-amps.

The procedure for doing so follows.

Note: In theory, you *could* implement a given differential equation using either integrators or differentiators. In practice, differentiators should be avoided. Differentiators amplify noise whereas integrators filter out noise. Likewise, the following procedure will only use integrators and amplifiers (no differentiators).

Procedure to Implement a Differential Equation with an Analog Computer

Assume y(t) and u(t) are related by the following differential equation:

$$y''' + a_2 y'' + a_1 y' + a_0 y = b_2 u'' + b_1 u' + b_0 u$$

Given u(t), design a circuit to find y(t) (i.e. design an analog commuter to solve this differential equation).

Step 1: Create a dummy variable, x(t). Rewrite this differential equation as

$$x = b_2 u'' + b_1 u' + b_0 u$$

$$y''' + a_2 y'' + a_1 y' + a_0 y = x$$

Now use the property of linearity and filter first then differentiate

$$x''' + a_2 x'' + a_1 x' + a_0 x = u$$

$$y = b_2 x'' + b_1 x' + b_0 x$$

It's not clear why to do this yet. Later on, this trick avoids having to use differentiators.

Now, given x''', find x(t) and its derivatives using integrators

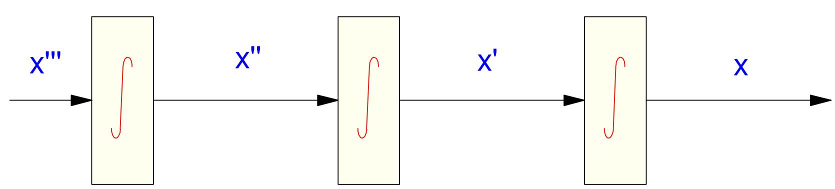

Figure 5: Given x''', find x(t) and its derivatives using three integrators.

Step 2: Now that you have x(t) and its derivatives, implement the differential equation

$$x''' + a_2 x'' + a_1 x' + a_0 x = u$$

Do this by rewriting it as

$$x''' = u - a_2 x'' - a_1 x' - a_0 x$$

Figure 6 shows a block diagram of this function

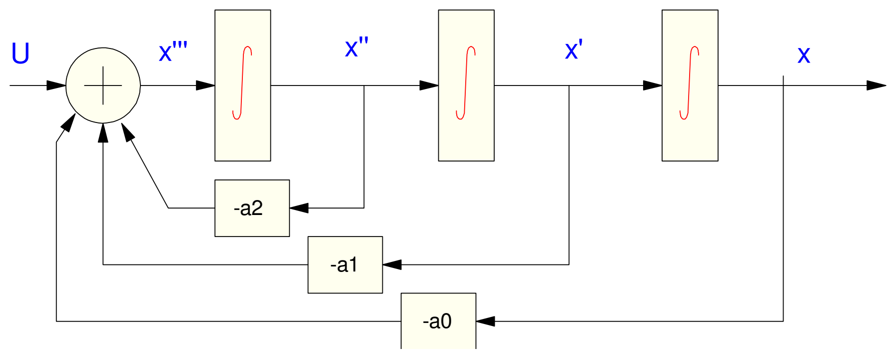

Figure 6: Block diagram representation for the differential equation for x(t)

Step 3: Now generate y(t) by summing x(t) and its derivatives

$$x''' + a_2 x'' + a_1 x' + a_0 x = u$$

$$y = b_2 x'' + b_1 x' + b_0 x$$

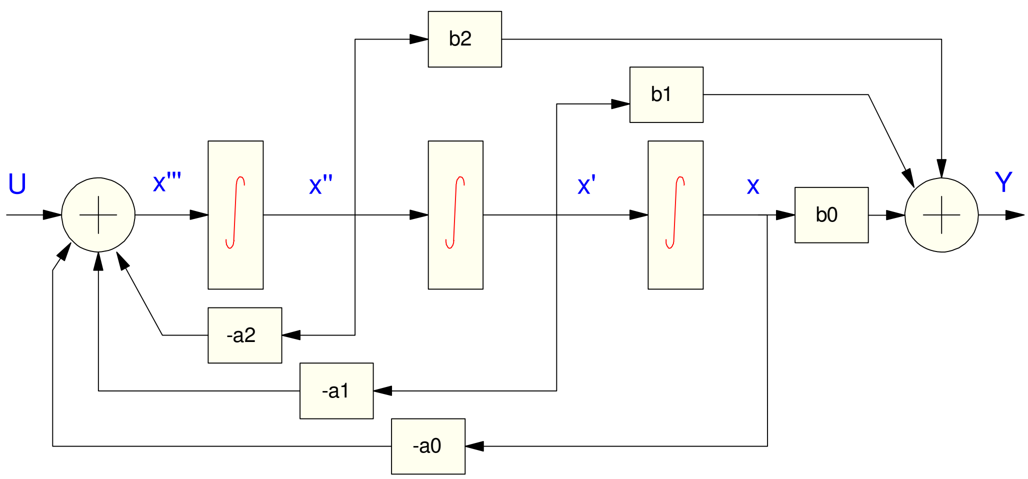

Figure 7: Block diagram for implementing the full differential equations

Note that this circuit has no differentiators. Instead of differentiating the output of the integral, simply use the input to the integrator (differentiation undoes integration).

Step 4: Now, replace each block with its analog computer symbol as presented in Figure 8.

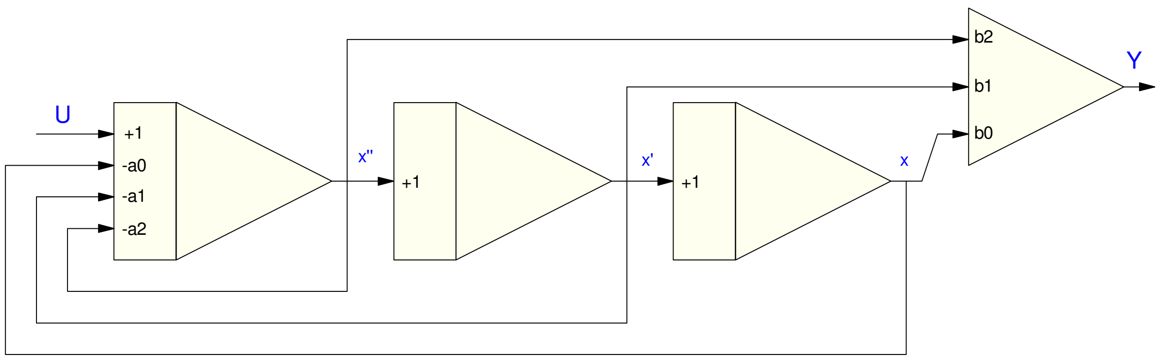

Figure 8: Analog computer for solving a differential equation

Step 5: Note with the two circuits given in Figures 3 and 4, the gains are all negative. Adjust the gains of the above circuit so that

- They are all negative, and
- They are all close to 1.000

Do this in a way that the overall differential equation is unchanged, however.

The trick to doing this is that

- The loop gains must remain the same, and
- The gain from U to Y through each path must be the same.

For the sign of the gains, this simply means

- For positive gains, there must be an even number of op-amps in each path.
- For negative gains, there must be an odd number of op-amps in each path.

Add an inverter (gain = -1) as needed to do so.

As for keeping the gains close to 1.000, this is just to avoid saturating one of the stages. If you have three gains:

$$(-1000) \cdot (-1) \cdot (-1)$$

it works better in practice if you write this as

$$(-10) \cdot (-10) \cdot (-10)$$

This results in the analog computer circuit of Figure 9

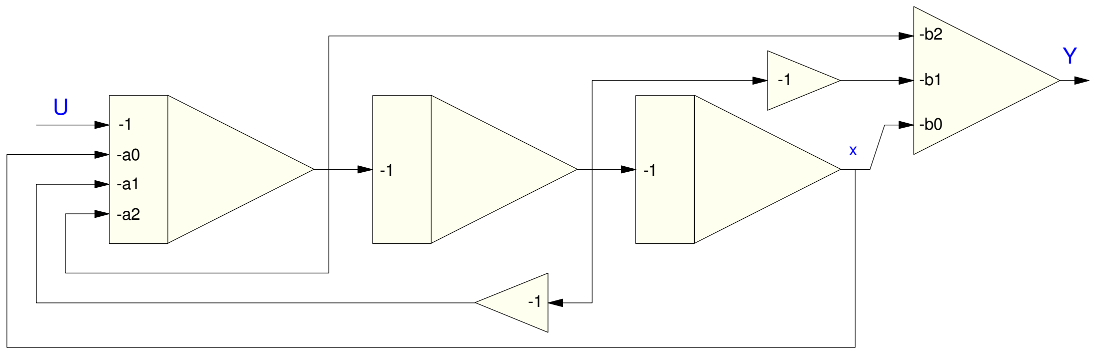

Figure 9: Final analog computer to solve a 3rd-order differential equation.

Step 6: Finally, replace each op-amp with its op-amp circuit. This is shown in Figure 10.

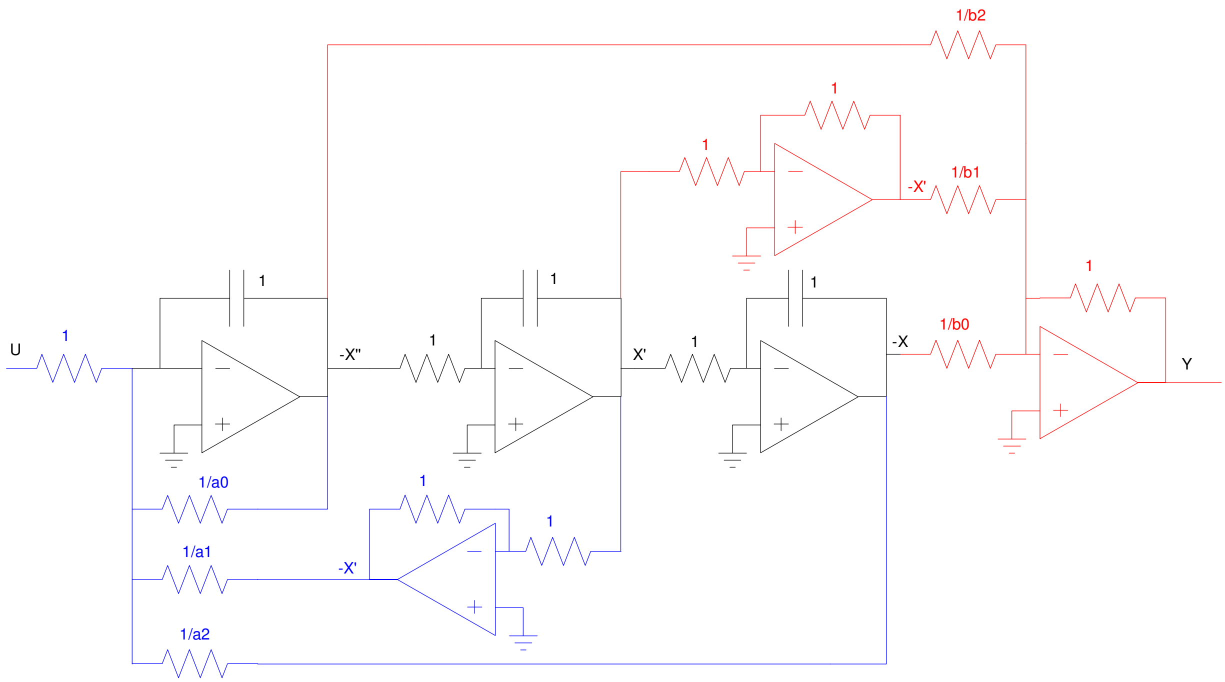

Figure 10: Op-Amp circuit to implement a 3rd-order differential equation (analog computer)

Note that the units of R and C can be adjusted as long as RC = 1. You *could* implement this circuit using

- R = 1 means 1 Ohm
- C = 1 means 1 Farad

You could also interpret this to be

- R = 1 means 1M Ohm
- C = 1 means 1 uF

As long as RC = 1, the units you choose for R and C are valid.

Modern Analog Computers

There's a saying:

Anything you can do in hardware you can do in software.
Anything you can do in software, you can also do in hardware.

This applies to analog computers as well.

- In the 1950s, analog computers were implemented using mechanical devices or vacuum tubes.
- In the 1980s, analog computers were implemented using op-amp circuits
- Today, analog computers are implemented using a microcontroller using numerical integration

Matlab's *Simulink* is one way to implement an analog commuter using software. There are other ways as well.

With software

- You don't have wires falling out
- You don't have DC offsets for integrators
- Your multiply functions are accurate to 16 decimal places
- You have an analog computer you can change with a simple double-click

Likewise, most analog computers are implemented in software today.

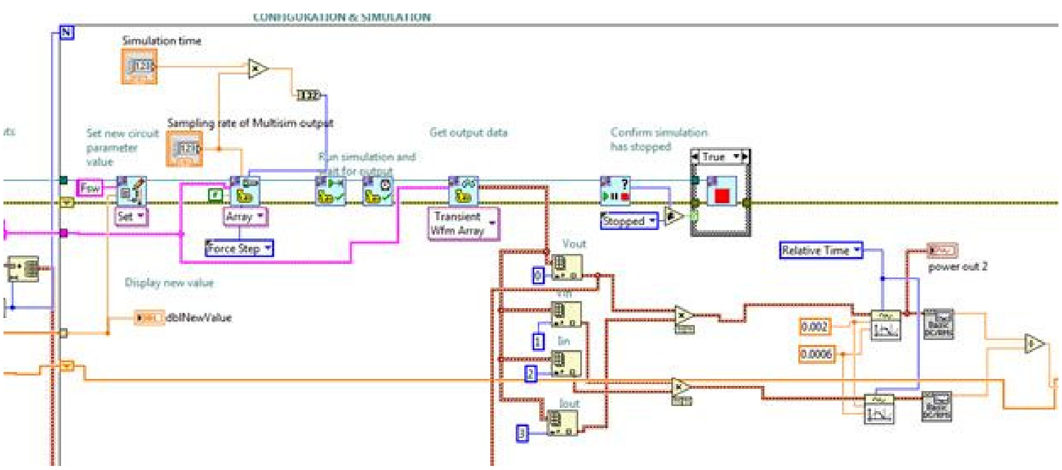

Figure 11: Software implementation of an analog computer with Simulink

Summary:

Analog computers are circuits which solve differential equations. The heart of analog computers is the inverting integrating circuit. If you are trying to solve an Nth-order differential equation, you need to include N integrators in your circuit. Plus inverting amplifiers and summing junctions

Op-amp circuits are one way of building an analog computer. Other methods exist as well:

- Analog computers can be implemented using mechanics (battleship fire-control computes in WWII)
- They can be implemented using op-amps (this lecture), and
- They can be implemented using software (such as Matlab's Simulink).