FORREST M. MIMS III

THE COMPUTER SCIENTIST

ANALOG COMPUTER TECHNIQUES FOR DIGITAL COMPUTERS

Now that incredibly powerful, low-cost *digital* computers have become commonplace, it's easy to overlook the importance of the venerable *analog* computer. Indeed, most users of today's personalized digital computers know little or nothing about analog computing.

As a longtime analog computer enthusiast, I'm happy to report that analog computers are alive and well in 1984. In this two-part series I'll cover analog computer basics and examine ways to apply simple analog techniques to any personal computer with a pair of analog joystick ports.

Why Analog Computers?

Since very few users of personal computers are even aware of analog computers, they stand to miss significant opportunities. Creative hardware and software users can benefit by adding to low-cost digital computers some of the operating techniques used with analog machines.

That's because some of today's analog machines incorporate digital logic and even entire digital computers. Such machines are sometimes called *hybrid* computers.

Because the basic principles of the electronic analog computer were developed as long as 50 years ago, it's only natural to wonder what advantages analog machines can possibly have over their digital counterparts. One major advantage is the ability of an analog machine to realistically *simulate* or *model* systems whose performance can only be described by interdependent sets of linear or non-linear differential equations having many variables. For example, analog computers can easily simulate the level and extent of flooding that results when a river's watershed receives an overabundance of rain. Likewise, analog machines can simulate the operation, over a wide range of conditions, of aircraft, ships, chemical manufacturing plants, nuclear reactors, and many other systems.

In all these examples, the analog computer can provide *real-time* solutions to the problem at hand because the computer functions as an electronic model whose parameters can be changed as fast as an operator can turn a dial.

For instance, assume you're the operator of an analog computer programmed to simulate the operation of a new sports car. Dials on the machine's control panel allow you to instantly specify the car's weight, dimensions and speed. Switches allow you to select such options as retractable headlights and various kinds of external rear-view mirrors.

Though our hypothetical analog computer incorporates an analog "front end," most of its output devices are familiar to users of digital personal computers. In addition to a printer they include a cathode-ray tube and an x-y pen plotter, both of which represent, graphically, the car's performance. Figure 1 summarizes the complete system in a block diagram.

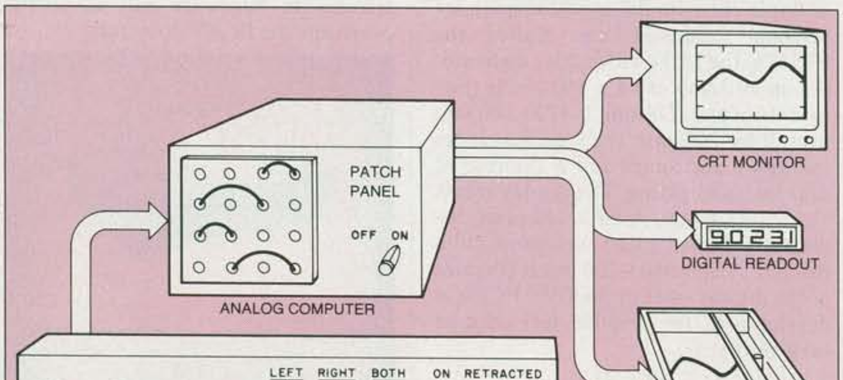


Fig. 1. Analog computer sports car simulator.

As you vary the car's design parameters, the various output devices almost immediately display the results. The printer lists the fuel efficiency (in miles per gallon) and aerodynamic drag coefficient that result from any combination of parameters you have selected on the control panel.

Yes, a digital computer can be programmed to provide the same results. One way would be to have the computer step through every conceivable variation in the car's parameters and provide listings and plots of the resulting performance possibilities. But this method would require more operating time andwould produce far more information than you need.

Furthermore, the all-digital approach removes the human-machine interface, the direct link between operator and computer that allows a human being to incorporate creativity, intuition, experience and knowledge in the design of a new sports car. In other words, the analog computer gives you all the advantages that result when you can casually change the parameters of the hypothetical sports car and instantly see the results.

I hope you're now convinced that analog computers have much to offer. So how can the owner of a *digital* computer can get into *analog* computing.

Fortunately, some basic analog computer techniques can be effectively used with any digital machine equipped with one or more analog-to-digital converters. In other words, if your computer has a pair of analog joystick inputs, chances are you can develop programs that will digitally simulate the ultimate simulator, the analog computer.

In this and next month's column we'll explore this topic in some detail. Let's begin by rediscovering some analog computer basics.

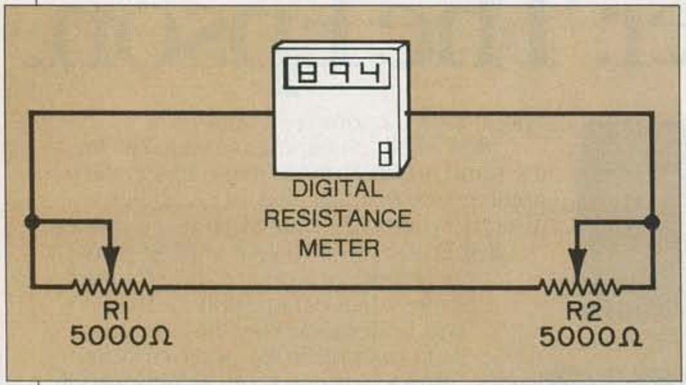


Fig. 2. Variable resistor analog addition circuit.

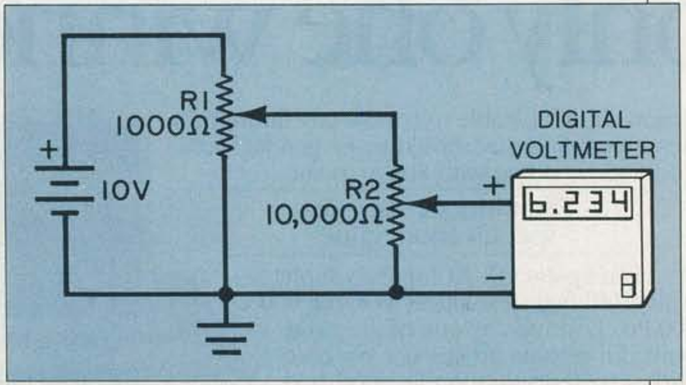


Fig. 3. Variable resistor analog multiplication circuit.

Analog Computer Basics

Mechanical analog computing machines have existed for more than a thousand years. One of the most widely used analog computing machines, the slide rule, was made obsolete only a decade ago by the portable scientific calculator, a digital microcomputer containing in its ROM stored programs dedicated to solving specific mathematical problems.

It's well-known that digital computer circuits process electrical signals that occupy one of two discrete levels. A floating or disconnected state, the so-called *third state*, may also be present. For example, when various circuits are connected to a common bus, to prevent confusion only *one* can be allowed to place a signal on the bus. The others are effectively isolated from the bus by circuits called three-state buffers or gates.

Analog computers process signals over a wide and continually variable voltage range. It's therefore much easier for analog computers to represent and perform mathematical operations upon numbers that have been represented by voltages.

On the other hand, digital computers are much more precise than analog machines and are therefore essential when paychecks, bank balances, loan statements, budgets and the like are being calculated.

Sometimes, however, the precise results offered by digital computers are superfluous. For example, would you prefer to read from a digital display that your car's fuel tank is 26.2 percent full or glance at an analog gauge that reveals you have about a quarter of a tankful remaining? Even if you prefer the digital readout, the reading is only as accurate as the *analog* sensor that measures the level of the fuel and the *analog*-to-digital circuit that drives the readout.

On a grander scale, an analog computer programmed to simulate a rain-drenched watershed might predict a flood crest of 15.2 feet with an error range of ±1 percent. A digital computer simulation might give a more precise prediction of 15.187568439731 feet. But since both predictions are subject to the same input errors (e.g., the uniformity and quantity of rainfall, condition of the soil, vegetation type and density), *both* provide a figure of *about* 15.2 feet. If you're not convinced, ask yourself which value the radio announcer will read over the air to an audience of potential flood victims.

Analog computers provide real-time solutions to problems

The preceding example also illustrates the role analog sensors play in many kinds of dedicated (single function) digital computer systems. Though the digital portion of such systems may have ten- or twelve-place accuracy, the accuracy of the analog sensor is considerably less, probably no more than a few tenths of a percent. Moreover, the overall response of an individual sensor may *not* be uniform with other sensors.

All in all, miniature analog computers might well be at least as effective as their digital counterparts in dedicated applications involving analog sensors. Then why aren't they as popular?

In my opinion, a principal reason is that analog circuit design is fast becoming a lost art. Another reason is that the microprocessor arrived *before* sophisticated single-chip analog computer chips. This technology gap provides some interesting opportunities for creative computer enthusiasts, and I'll have more to say about it in Part 2 of this two-part column.

Do-It-Yourself Analog Computers

While large-scale analog computers can be very complex, small-scale machines can be surprisingly simple. For example, though practical analog computers are designed around precision operational amplifier circuits, simple analog computers can be constructed from a few potentiometers and a multimeter.

Figure 2 shows an *analog adding machine.* This ultra-simple circuit requires only two inexpensive potentiometers equipped with pointer knobs and scales. The circuit can display its results on a conventional analog meter or a digital multimeter.

If each potentiometer has a resistance of 5,000 ohms and if each ohm represents a unit, then this analog adder can sum any two numbers of up to 5,000 and provide a total of up to 10,000. Simply by changing the potentiometer scales, a form of programming, the machine can be set to add virtually any pair of numbers provided that an appropriate correction factor is applied to the value displayed by the multimeter.

Figure 3 shows how the two potentiometers in Fig. 2 can be rearranged or *reprogrammed* to provide an *analog multiplication machine.* Here the potentiometers are connected as *voltage dividers* across a 10-volt power supply.

How does the multipler work? Ignoring the value of the potentiometers, assume each is equipped with a scale having eleven equally spaced lines. The scale for R1 is marked 0 through 10 while that of R2 is marked 0 through 1 in increments of 0.1.

When R1's pointer is rotated to the 5 position on its scale, its wiper is at its center position and the voltage appearing across the wiper and ground is half the input value, or 5 volts. Likewise, when R2's pointer is rotated to the 0.5 position on its scale, its wiper is also at its center position, and the voltage across the wiper and ground is half the input value, or 2.5 volts. This output voltage is the product of the two potentiometer settings ($5 \times 0.5 = 2.5$).

The simple addition and multiplication circuits in Figs. 2 and 3 can be made more accurate by using multi-turn precision potentiometers. Ten-turn dial potentiometers are an excellent, albeit rather expensive, choice.

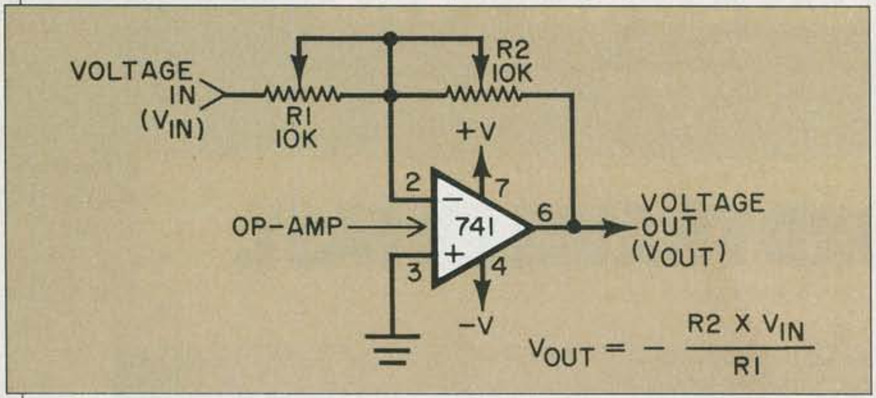


Fig. 4. Basic operational amplifier variable resistor multiplier.

Moreover, both the addition and multiplication circuits can be combined into a single circuit that can be reconfigured by means of a *patch panel* equipped with switches or removable plugs and wires. By doing so, a *programmable* analog computer can be created.

Experienced circuit designers can replace the requisite programming wires, plugs and switches with *analog gates.* A digital word (i.e., binary number) much like a digital computer's machine language instruction, can then be used to tell the computer to add or multiply. The analog computer can be greatly expanded using such techniques. And it can be controlled using familiar but, by comparison, more easily programmed digital computer techniques.

Activating the Analog Computer

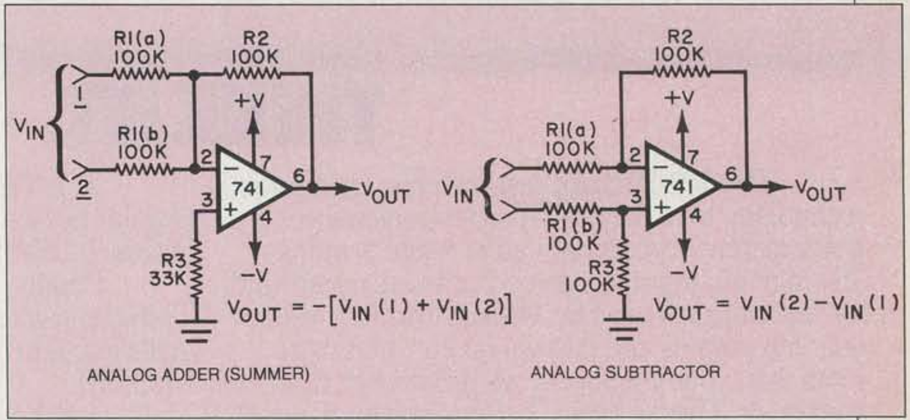
The basic variable resistor analog computer discussed thus far is a *passive* device. Far more versatile performance can be had by adding *active* electronic circuits, the most important being the *operational amplifier.* First developed by George H. Philbrick in the mid-1930s for use in an early all-electronic analog computer, the op-amp, as it is usually called, uses feedback from its output to one of its two inputs to control precisely its amplification factor or *gain.*

Figure 4 shows a very basic op-amp *function block.* The output voltage of this circuit is the product of the feedback resistance (R2) and the input voltage (V_{in}) divided by the input resistance (R1). Therefore, the basic circuit in Fig. 4 can perform multiplication and division.

Figure 5 shows how the circuit in Fig. 4 can be slightly modified to enable it to perform addition and subtraction. More than two voltages can be combined by adding additional input resistors to the adder (summer) circuit.

The summer in Fig. 5 can average the sum of two or more input voltages by making the ratio R2/R1 equal to the number of input voltages. For instance, if each of two input resistors [R1(a) and R1(b)] has a resistance of 100,000 ohms, then changing R2's resistance to 50,000 ohms will cause the output voltage to equal the average of the two input voltages.

Fig. 5. Basic operational amplifer variable resistor multiplier.



More precise multiplication, division and square root extraction can be performed by means of an op-amp operated as a logarithmic amplifier. This is accomplished by substituting a diode or transistor in the feedback loop of an op-amp. The natural logarithmic relationship between the voltage drop across a *pn* junction and the current flowing through the junction makes possible this important analog computer function block.

Op-amps can provide many other functions useful in analog computers, including integration, differentiation, raising exponents and function generation.

As you can see, the op-amp is to analog computers what the logic gate is to digital computers. Op-amps, however, are much more temperamental than logic gates. Sensitive to temperature changes, high-accuracy op-amp function blocks require precision resistors having a tolerance of 0.1 percent.

These drawbacks are a major reason why analog computers were soon surpassed by more dependable and predictable digital computers. In recent years, however, analog computer function modules have been fabricated with on-chip resistors having unprecedented accuracies. One such circuit, the Analog Devices AD534, includes a dozen on-chip resistors trimmed to a high degree of accuracy by a pulsed laser.

When first introduced five years ago, its manufacturer called the AD534 the first single-chip analog computer. Since this chip can multiply, divide or square two input voltages or take the square root of a single input voltage—all with an accuracy of ± 0.25 percent—the AD534 certainly qualifies as a powerful analog processing chip. I'll have more to say about it and similar chips in Part 2.

Simulating an Analog Computer with BASIC

If you enjoy experimenting with hardware, you can assemble a surprisingly powerful analog computer from a few op-amps, an AD534, a patch panel and a digital multimeter. Alternatively, you

(Continued on page 113)

Computer Scientist
(Continued from page 31)

can learn a great deal about the flexibility and power of analog computers by simulating a simple machine with the help of any personal computer having analog joystick ports.

Suitable computers include Radio Shack's TRS-80 Color Computer, the Apple IIe and the IBM PC family. Consider, for example, the PC*jr*, a machine having two analog joystick ports. Each port supports two potentiometers and two on-off switches. Therefore, a simple analog control panel having four potentiometers and four switches can be constructed.

In Part 2 of this column, I'll describe how to assemble such a panel from readily available parts. In the meantime, you can easily experiment with an analog input for your computer by using its joysticks as analog input devices.

For example, the listing that follows converts the PC*jr* into an analog input machine that provides the sum, difference, product and quotient of any two joystick values:

```
10  'PCjr ANALOG COMPUTER
    SIMULATOR
20  CLS
30  X=STICK (0):Y=STICK(1)
40  LOCATE 6,15
50  PRINT "X ="X;"Y =";Y
60  LOCATE 8,15
70  PRINT "X*Y =";X*Y
80  LOCATE 10,15
90  PRINT "X/Y =";X/Y
100 LOCATE 12,15
110 PRINT "X+Y =";X+Y
120 LOCATE 14,15
130 PRINT "X-Y =";X-Y
140 GOTO 30
```

Though this program is specifically designed for the PC*jr*, it can be easily adapted for use with any other machine having joystick inputs. When the program is run, the monitor displays a table showing the current joystick values (X and Y) and the results of the four basic arithmetic operations. Here's the table that was displayed when the joystick handle was moved to a random position:

```
X = 110   Y = 78
X*Y = 8580
X/Y = 1.410256 -02
X+Y = 188
X-Y = 32
```

Many other arithmetic operations and other enhancements can be added to this basic program: The X and Y values can be variables in an equation that is plotted on the display. Moving the joystick to change the variables would then almost simultaneously alter the shape of the plotted curve. Similarly, the X and Y values can be used to set, via the joystick, the aspect ratio of an ellipse having a fixed diameter, as in this PC*jr* listing:

```
05 'PCjr VARIABLE ASPECT
   RATIO ELLIPSE
10 CLS
20 SCREEN 1,0
30 X=STICK(0):Y=STICK(1)
40 CIRCLE (160,94),
   20,,,,X/Y
50 CLS
60 GOTO 30
```

When this routine is entered and run, a flashing blue ellipse or circle appears. Moving the joystick's handle changes the aspect ratio of the ellipse.

Finally, a second joystick can be add-

(Continued on page 114)

Computer Scientist

(Continued from page 113)

ed to provide a total of four variables. If the joysticks are equipped with trigger buttons, from two to four user-selectable status conditions or preprogrammed numerical values can be inserted into the mix of available options.

In Part 2 I'll describe various ways to make and use a simple analog control panel for a personal computer. I'll also describe the surprisingly simple architecture of a futuristic, single-chip analog/digital microcomputer.

For more details about do-it-yourself simple analog computer circuits, see the "Experimenter's Corner" columns in the January and February 1979 issues of *Popular Electronics*. Both have been reprinted in *The Forrest Mims Circuit Scrapbook* (McGraw-Hill, 1983). ◇