A virtual analog computer for your desktop

Arthur Glazar - June 29, 2012

Editor's Note: This article came about from another article called "Circuits without wires" in which a comment by Arthur Glazar led to my discovery of this gem of an article, thanks to my colleague Margery Conner.
— Steve Taranovich

If you "do" circuit design, then you probably use PSPICE or one of its derivatives in your design process. If that's the case, you can easily add a powerful "virtual analog computer" (VAC) to your desktop.

Why would anyone want an analog computer, since they became extinct, for good reasons, back in the 1970s? The answer is first, analog computers are fun to use. Second, you can't beat an analog computer for solving, analyzing, manipulating and teaching differential equations. Finally, the limitations that led to the demise of traditional analog computers are eliminated in a PSPICE virtual implementation.

Traditional analog computers utilize operational amplifiers (op-amps) that are configured as functional blocks such as integrators and summers. But op-amp signals can only swing between the supply-voltage rails. As a consequence, traditional analog computer simulations must include "amplitude scaling" to keep all signals within that dynamic range. Similarly, "time scaling" is needed in a traditional analog computer to account for the frequency response of the functional blocks. These considerations vanish in the VAC, so that kilovolts and microvolts, hours and microseconds peacefully coexist in a simulation. Also, there are no leakage currents (and hence no integrator "drift"), no potentiometer loading effects, no limits on the number of integrators, multipliers, etc., etc. Examples will follow.

A final observation before getting into the implementation of the VAC: To avoid ambiguity, it is recommended that the International System of Units (abbreviated SI from French: Système international d'unités) of measure be used exclusively; that is, kilograms, meters, seconds, newtons, amperes, etc. By doing so, the voltage at any node of an analog simulation will be a one-to-one analog of kilograms, meters, newtons, etc. For convenience, various function blocks may be included to perform input/output conversions to and from SI. In short, although SI may be the "native" system of units, VAC can be customized to handle other systems of units as well.

I use LTSPICE, a product of Linear Technology Corporation. This is a world-class circuit simulator available from LTC as a free download from LTC.com. If you prefer a different "Alphabet-
SPICE", that's ok as long as it includes the features that will be discussed.

In "building" a VAC the first step is to create a library of commonly-used functions. Other functions can be created as needed. Here is a basic list:

- **INTEGRATOR**
- **DIFFERENTIATOR**
- **SUMMER**
- **INVERTER**
- **MULTIPLIER**
- **DIVIDER**

**Figure 1** shows the symbols that I use for these functions. And of course, any of LTSPICE's existing library components can be used within a VAC simulation as we will show. For very complex simulations, the LTSPICE hierarchy methods can keep things organized.
Let's now look at a simple example of a differential equation and how to set up and run it in LTSPICE. We'll use an electrical problem, but it could be a mechanical or mechatronic problem just as well.

**Figure 2** shows a simple R-L-C series circuit together with the 2nd-order differential equation that relates the current, i, to the other elements.

![Figure 2](image)

Figure 2 A simple R-L-C series circuit together with the 2nd-order differential equation

\[ R \frac{di}{dt} + L \frac{d^2i}{dt^2} + \frac{i}{C} = \frac{dV}{dt} \]

In setting up a VAC simulation, the first step is to apply the following general rule:

**RULE:** Rearrange the differential equation so that the highest-order derivative is isolated on the right side of the equation.

Applying this Rule to the previous equation gives the following:

\[ -(R/L)(di/dt) - (1/LC)i + (1/L)(dv/dt) = \frac{d^2i}{dt^2} \]

The notation in the above equations is not amenable to keyboard entry in LTSPICE, so instead, we will use "primes" to denote time derivatives and rewrite the equation as follows:

\[-(R/L)i' - (1/LC)i + (1/L)V' = i'' \]

\[(1)\]

The three terms on the left side of equation (1) represent inputs to a summing amplifier, and the single term on the right side represents the output of the summing amplifier as shown in **Figure 3**.
Figure 3 The summing amplifier with inputs representing the left side of equation (1) and the output representing the right side of equation (1).

Since the output of the summing amplifier is $i''$ (the second derivative of current, $i$), it can be integrated once to obtain $i'$ and twice to obtain $i$ as shown in Figure 4. Then $i'$ and $i$ can be multiplied by $-(R/L)$ and $-(1/LC)$, respectively, and fed back to the appropriate input as shown in Figure 5.

Figure 4 Showing the inputs being integrated twice to give $i$ as an output.

Figure 5 Showing $i'$ and $i$ being multiplied by $-(R/L)$ and $-(1/LC)$, respectively, and fed back to the appropriate input.

Completing the VAC simulation diagram.
Completing the VAC simulation diagram

In Figure 6 the third input has been added to the summing amplifier to complete the VAC simulation diagram. The lower section of the figure, the "Control Panel", illustrates one way to organize the problem setup. The Control Panel groups together the problem parameters, constants and coefficients for easy editing. Text notes have also been included in the Control Panel area describing the problem parameters for this particular example.

![Diagram](image.png)

**Figure 6** The upper diagram shows the third input has been added to the summing amplifier to complete the VAC simulation diagram; while the lower section, the "Control Panel", illustrates one way to organize the problem setup.

It was mentioned earlier that standard SPICE circuitry could be mixed with VAC components. This is illustrated in Figure 6: in addition to the VAC simulation at the upper right of the figure, the original series RLC is also modeled at the upper left. Both models can be run simultaneously, and identical solutions for the current, i, can be plotted as shown in Figure 7, in which the upper panel displays the RLC circuit solution and the lower pane displays the VAC solution. The upper plot was obtained using the LTSPICE current probe, while the lower trace was obtained using the voltage probe.
Figure 7 Showing both models can be run simultaneously, and identical solutions for the current, i, can be plotted

VAC functional blocks (Integrator, Multiplier, etc.) are created using LTSPICE subcircuits, and all of the subcircuits use the same basic circuit element, the arbitrary behavioral voltage source, BV. Detailed information on how to generate subcircuits and their symbols is provided in the LTSPICE help files. However, we will briefly examine the makeup of the Integrator subcircuit, since it is typical of all VAC function blocks.

Figure 8 shows the Integrator subcircuit schematic (Integrator.asc) and its netlist (Integrator.sub). The integrator requires two input nodes and one output node. One of the inputs (Node A) is for the signal to be integrated. The second input (Node B) is for setting the initial condition that will appear at the integrator's output (Node C) at time t=0. The input resistors are only needed to satisfy LTSPICE circuit check rules.

![Integrator Subcircuit Schematic](image)

Figure 8A shows the Integrator subcircuit schematic (Integrator.asc) and its netlist (Integrator.sub).

The output of the behavioral voltage source, B1, is applied to Node C via a 10 ohm resistor. The resistor’s function is simply to prevent unintentional circuit check errors.

The output at Node C is defined by the statement shown in Figure 8A:

\[ V = \int dt(V(A), V(B)) \]
V=\text{idt}(V(A), V(B)) \text{ which instructs LTSPICE to perform a time integration on the signal at node A, and to apply the initial condition appearing at node B. The Help files in LTSPICE contain syntax descriptions for all of the mathematical functions available for the BV.}

An \textit{analog computer design tool} "starter" library of VAC functions and instructions can be obtained on EDN's Tool section.

\textbf{About the author}

Arthur Glazar received his B.E.E. degree from Pratt Institute in 1960. He immediately went to work for the Marine Division of Sperry Gyroscope Company in Syosset, Long Island. At Sperry, he helped to commission and operate a state-of-the-art analog computing facility used in the Polaris Submarine program. After leaving Sperry he continued to work in the Long Island electronics industry until retiring in 1991. Along the way he obtained N.Y. State P.E. certification. He is an I.E.E.E. Senior Life Member, and a member of Tau Beta Pi and Eta Kappa Nu. Glazar presently spends his spare time restoring vintage (analog) avionics at the Cradle of Aviation Museum in Garden City, Long Island.

\textbf{Related articles}:

- \textit{Analog: back to the future}
- \textit{Looking back at early computers}
- \textit{Who Remembers Analog Computers?}
- \textit{Analog design in the 21st century: challenges, tools, and IC advances}