1.3 Computers — Simulation by Analog and Hybrid Systems

P. A. HOLST (1970, 1985) **B. G. LIPTÁK** (1995) **S. RENGANATHAN** (2005)

Types:	A. Analog B. Hybrid
Size:	Small — under 50 amplifiers Medium — from 50 to 150 amplifiers Large — over 150 amplifiers
Components:	Integrators, adders, multipliers, function generators, resolvers, diodes, switches, relays, digital attenuators, logic gates, flip-flops, counters, differentiators, pulse generators, and coefficient potentiometers
Interface Components:	Analog-to-digital and digital-to-analog converters, multiplexers, track-store amplifiers, and logic signal registers
Inaccuracy:	For static, linear analog units, ±0.005% to ±0.1% For static, nonlinear analog units. ±0.01& to ±2%
Frequency:	Clock cycle frequency, 100 kHz to 2 MHz. Conversion rate. 10 KHz to 1 MHz. Full power signal bandwidth, from DC to 25 to 500 kHz
Costs:	Computers with 20 to 100 amplifiers cost from $7500 to $30,000. Units with more than 100 amplifiers cost from $20,000 to $150,000. For 10 to 50 channels of hybrid interface, the cost ranges from $15,000 to $75,000.
Partial List of Suppliers:	ABB Kent-Taylor (B) (www.abb.com/us/instrumentation) Acromag Inc. (A) (www. acromag.com) Adtech Instrument (A,B) (www.adtech.info) AGM Electronics Inc. (A,B) (http://agmelectronics.com/AGMContact.asp) Ametek, Rochester (B) (www. ametekapt.com) ASC Computer Systems (A) (www.asclubbock.com) Celesco Transducer Products (B) (www.celesco.com) Compudas Corp. (B) (www.compudas. com) Computrol Inc. (B) (www.computrol.com) Devar Inc. (A) (www.devarinc.com) Electronic Associates Inc. (B) (http://dcoward.best.vwh.net/analog/eai.htm) Emerson Process, Daniel Div. (A) (www.easydeltav.com) Gould Inc. (B) (www. pumpsmar.net) ILC Data Device Corp. (B) (www.electrobase.com) Invensys, Foxboro (A, B) (www.foxboro.com) MTS Systems Corp. (B) (www.mtssensors.com) Siemens, Moore Products (A, B) (www.sea.siemens.com) Voice Computer (B) (www. webdesk. com/voice-computer-control/) Xycom Automation Inc. (B) (www. xycom. com).

This section first describes the components and the operation of both analog and hybrid computers and then covers some of the simulation techniques for first order, second order, PID, and other applications.

The computer is a versatile tool for many applications ranging from simple calculation to sophisticated control of large-scale process plants. Computers are classified as analog and digital. Analog computers accept continuous signals and perform many operations on them, such as addition, subtraction, multiplication, integration, and simulation of systems.

THE ANALOG COMPUTER

Analog computers work on continuous signals and consist of operational amplifiers, capacitors, resistors, potentiometers, diodes, switches, squarer cards, and patch cards. All the fundamental functions of computation, such as addition, subtraction, multiplication, integration, differentiation, and generation of different functions, can be carried out with an analog computer.

The heart of an analog computer is the operational amplifier. The operational amplifier is a single-ended, high-gain DC coupled wide bandwidth unit, which has a very high open-loop gain, on the order of 10^5 to 10^8. All the computer signals are referenced to a common ground. Offsets, drifts due to temperature variation, aging, and electronic noise are the main problems in the operational amplifier circuits. Hence special precautions are taken to reduce or eliminate them. Special care should be taken to minimize the electromagnetic and static coupling between different computing units and signal sources.

The basic analog computing units are the inverter, summer, integrator, and multiplier.

Operational Amplifier

The main component of an analog computer is the operational amplifier, popularly called the op-amp, which is represented by a symbol shown in Figure 1.3a.

Its static gain μ is of the order of 10^5 to 10^8 and is a single-ended input and output representation. Its other special features are:

- High input impedance
- High DC gain
- Low power requirement
- Low noise and drift
- Stability and ruggedness

With the advent of integrated circuits (IC), operational amplifiers are now available as chips, such as the units IC741, IC741C, and IC LM 308. The ICs are quite compact, and many problems such as offset, drift, and noise associated with discrete component models have been minimized or eliminated.

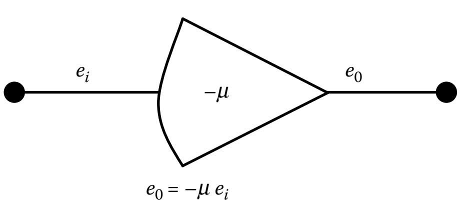


FIG. 1.3a
Symbol for the operational amplifier (OP-AMP).

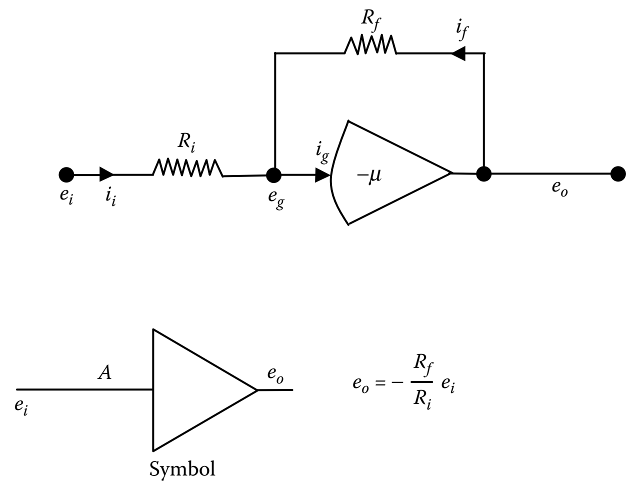


FIG. 1.3b
The circuit of and the symbol for an inverter.

In analog computers, operational amplifiers are used for three basic purposes:

- To generate the necessary computing functions
- To amplify the input signal level
- To provide isolation and unloading between the different input and output signals within the computing units

The Inverter

This operational amplifier can be used in conjunction with resistors to invert and multiply a signal. The circuit for and the symbol of the inverter are shown in Figure 1.3b. It may be noted that e_g is very near earth potential because of the large value of μ and a finite e_0 value. The current flowing into the op-amp grid is very small, on the order of micro amps.

According to Kirchoff's current law:

$$i_i + i_f = i_g \qquad \mathbf{1.3(1)}$$

when $i_g = 0$, $i_1 = -i_f$, $\frac{e_i}{R_i}$ and $i_f = \frac{e_0}{R_f}$.

Therefore

$$e_0 = -\frac{R_f}{R_i} e_i$$

If $R_f > R_i$, the input is inverted and amplified.

The Summer

The operational amplifier can be used as a summer by using the circuit shown in Figure 1.3c. Assuming μ to be large, i_g can

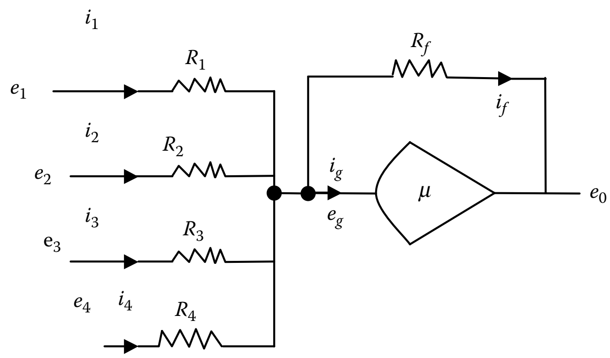


FIG. 1.3c
The circuit of a summer.

be neglected. Applying Kirchoff's equation, Equation 1.3(2) is obtained:

$$i_f = (i_1 + i_2 + i_3 + i_4) \qquad \mathbf{1.3(2)}$$

As e_g is near ground potential

$$i_f = e_0/R_f,\ i_1 = e_1/R_1,\ i_2 = e_2/R_2,\ i_3 = e_3/R_3 \quad \text{and} \quad i_4 = e_4/R_4 \qquad \mathbf{1.3(3)}$$

Then

$$e_0/R_f = -[e_1/R_1 + e_2/R_2 + e_3/R_3 + e_4/R_4] \qquad \mathbf{1.3(4)}$$

$$e_0 = -[R_f/R_1e_1 + R_f/R_2e_2 + R_f/R_3e_3 + R_f/R_4e_4] \qquad \mathbf{1.3(5)}$$

$$e_0 = -[g_1e_1 + g_2e_2 + g_3e_3 + g_4e_4] \qquad \mathbf{1.3(6)}$$

The symbol for a summer is given in Figure 1.3d.

If the open loop gain μ is low, the summing junction potential e_g must be considered in Equation 1.3(2):

$$(e_0 - e_g)/R_f = -[(e_1 - e_g)/R_1 + (e_2 - e_g)/R_2 + (e_3 - e_g)/R_3 + (e_4 - e_g)/R_4] \qquad \mathbf{1.3(7)}$$

Since

$$e_0 = -\mu e_g \qquad \mathbf{1.3(8)}$$

Substituting 1.3(8) in 1.3(7), one obtains:

$$e_0(1/R_f + 1/\mu R_f) = -[e_1/R_1 + e_0/\mu R_1 + e_2/R_2 + e_0/\mu R_2 + e_3/R_3 + e_0/\mu R_3 + e_4/R_4 + e_0/\mu R_4] \qquad \mathbf{1.3(9)}$$

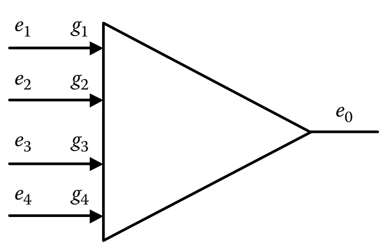


FIG. 1.3d
The symbol for a summer.

i.e.,

$$e_0[1/R_f + 1/\mu R_f + 1/\mu R_1 + 1/\mu R_2 + 1/\mu R_3 + 1/\mu R_4] = -[e_1/R_1 + e_2/R_2 + e_3/R_3 + e_4/R_4] \qquad \mathbf{1.3(10)}$$

$$e_0 = -f_u[g_1e_1 + g_2e_2 + g_3e_3 + g_4e_4] \qquad \mathbf{1.3(11)}$$

where

$$g_1 = R_f/R_1,\ g_2 = R_f/R_2,\ g_3 = R_f/R_3,\ g_4 = R_f/R_4$$
$$1/f_u = 1/\mu(1 + \mu + g_1 + g_2 + g_3 + g_4) \qquad \mathbf{1.3(12)}$$

f_u is the correction factor.

The analog computer circuits and their symbols, which are used for basic computation, are summarized in Figure 1.3e.

Frequency Response of the OP-AMP

It may be noted that the open loop amplification (open loop gain) of the OP-AMP decreases as the signal frequency increases. Hence the analog computation must be carried out within the corresponding frequency limits. For higher frequencies, the gain falls off. When the signal frequency is 100 kHz, the open loop gain becomes 10^2 as shown in Figure 1.3f. The correction factor for this gain is 101/100. That means the error in computation is 0.01%

Analog Circuits for Differential Equations

Consider the equation below, in which a is a positive constant and $f(t)$ is an arbitrary function of time:

$$dx/dt = ax + f(t) \qquad \mathbf{1.3(13)}$$

Let us assume that $-dx/dt$ is available as voltage from some amplifier. This voltage is connected to an integrator as shown in Figure 1.3g. The output of this integrator is x. Then this output is connected to the summer as also shown in Figure 1.3g. The output of this summer is $-[ax + f(t)]$, which is equal to $-dx/dt$. By connecting the output of the summer to the input of the integrator, the equality of the equation 1.3(13) has been electrically established.

Figure 1.3g can be redrawn as in Figure 1.3h. Once this circuit is made and the circuit is switched on, $x(t)$ is obtained as a function of time, which is the solution of the differential equation. It may be noted that the initial condition is assumed to be zero.

The detailed circuit for a first-order differential equation is described in Figure 1.3i.

There is no fixed procedure for constructing computer circuits to solve differential equations. There are many satisfactory ways to arrive at a computer circuit to solve a given differential equation.

The analog computer circuit shown in Figure 1.3h is for solving a first-order differential equation with a zero initial condition. But normally, differential equations will have boundary conditions or initial conditions. A first-order differential equation will have one initial condition, a second-order will have two initial conditions, and so on.

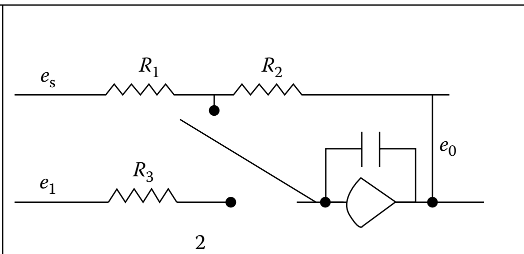
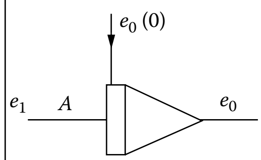
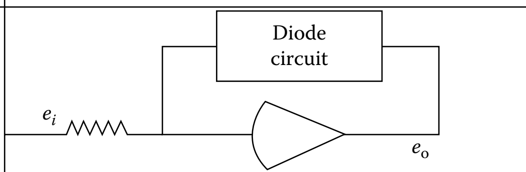
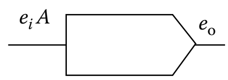
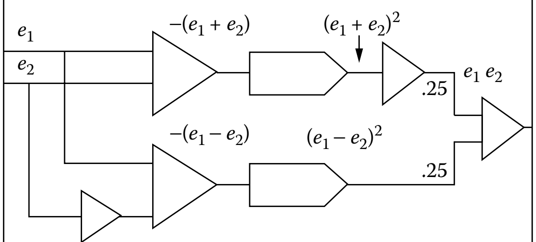
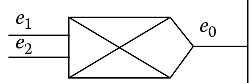
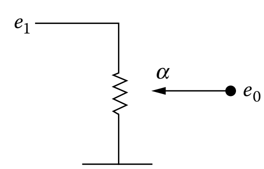
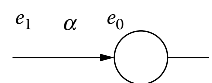
Integrator with initial condition	 $e_0 = -\left[\frac{1}{R_3C}\int_o^t e_i dt + \frac{R_2}{R_1}e_s\right]$	 $e_o = -A\int_o^t e_i\,dt + e_o$ $A = \frac{1}{R_3c}$; $e_o(o) = -\frac{R_2}{R_1}e_s$	To have an initial condition for integration the capacitor is charged initially to a required value by choosing R_2 & R_1. The switch is initially at point 1 for a sufficient time so that the capacitor C is charged to $R_2/R_1\,e_s$. When we want to integrate the signal e_s the switch is moved from position 1 to position 2 and then the integration starts.
Squarer	 $e_0 = -Ae_i^2$ Square function generation	 $e_0 = -Ae_i^2$	This device approximates a function $y = f(x)$ by a number of straight line segments. The electrical components used in a square function generator are diodes, resistors, and potentiometers.
Multiplier		 $e_0 = -\,e_1\,e_2$	This circuit is known as a quarter square multiplier $e_0 = \frac{1}{4}\left[(e_1+e_2)^2 - (e_1-e_2)^2\right]$ $= e_1e_2$ Now the circuit is available as a device which produces a current proportional to the product of two voltages.
Signal divider		 $e_0 = \alpha\,e_1$ $0 \le \alpha \le 1$	

FIG. 1.3e
Summary of analog computer circuits and their symbols used for basic computation.

For the first-order system described by Equation 1.3(13), assume the initial condition to be $x(0) = x_0$. This requires that the voltage from the integrator be held at x_0 until the integration starts. There are two ways of obtaining the initial condition. One way is to connect a battery of the required initial condition value between the grid and the output as shown in Figure 1.3j.

Initially, the switch is closed for a sufficiently long time for the capacitor to become charged to the initial condition value x_0. Two relay-operated switches are placed in the circuit as shown in Figure 1.3j. When the initial condition is to be established, switch 1 is closed and switch 2 is simultaneously opened. When the equation is to be solved, a suitable relay system opens switch 1 and closes switch 2 at the same time.

Another way of obtaining the initial condition voltage is shown in Figure 1.3k. In this case, when the computer is put in the initial condition (IC) mode, the switches S_1 and S_2 are put in position 1. This is done simultaneously by a master switch. A voltage V_s of desired magnitude and polarity is connected to the R_1 and R_2 circuit. Normally, R_1 and R_2 values will be 0.1 MΩ. When the switch S_1 is in position 1, $e_0 = -(R_2/R_1)V_s$, and when $R_1 = R_2$, e_0 becomes the negative of V_s.

Then, according to the time constant of the RC circuit, the capacitor is charged to $-e_0$. In this case the time constant $RC = 0.1$ s, and hence one can expect that in five times the time constant or say in 0.5 s the capacitor will be charged to V_s voltage, which becomes the initial condition for the integrator.

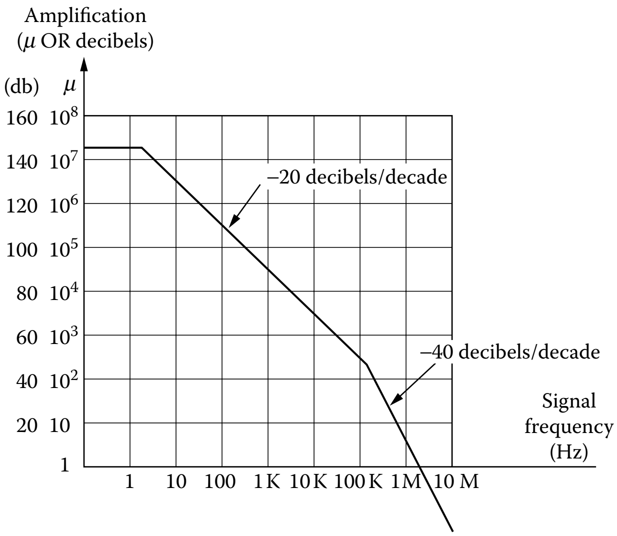


FIG. 1.3f
The open-loop gain characteristic is such that as the signal frequency increases, the open-loop gain (amplification) of the OP-AMP decreases.

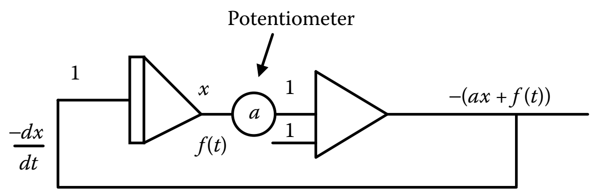


FIG. 1.3g
The circuit for solving Equation 1.3(13).

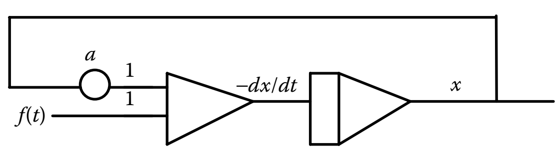


FIG. 1.3h
The computer circuit for solving first-order differential equations.

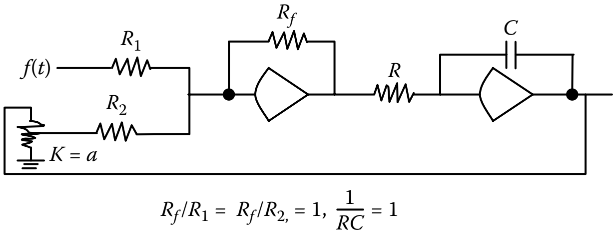


FIG. 1.3i
The detailed circuit for solving first-order differential equations.

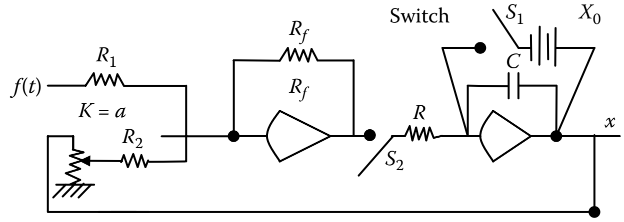


FIG. 1.3j
Analog computer circuit for solving the equation dx/dt = ax + f(t) with initial condition.

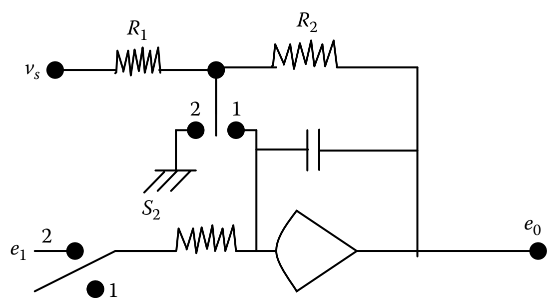


FIG. 1.3k
The circuit for an integrator with IC value.

When the computer is placed in the operate mode, switches S_1 and S_2 are switched to position 2 and the voltage is integrated. In this state, resistor R_1 provides a load for the initial condition voltage because one side of the resistor is grounded through switch S_1.

Figure 1.3l shows the symbol for an integrator with an IC value. It may be noted that the reference voltage is opposite in sign to the initial condition voltage because of the sign reversal at the amplifier.

Magnitude and Time Scaling

To get satisfactory results utilizing the analog computer, magnitude scaling and time scaling are needed.

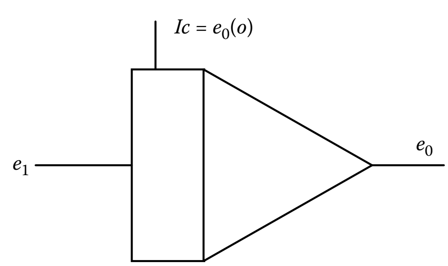


FIG. 1.3l
The symbol for an integrator with IC value.

Magnitude Scaling During the solution of a problem in an analog computer, the output voltage from any amplifier should not exceed the operating range of that amplifier. For many semiconductor analog computers the nominal voltage range is +15 to –15 V. If, at any point of the solution, the output voltage of any amplifier exceeds this value, the amplifier response may not be linear. Often, it is possible to estimate the maximum value of x, say x_m, which occurs during the solution. This maximum value should be reduced to the maximum output voltage of the amplifier, which is say 15 V, by selecting the suitable scaling factor of $k_x = 15/x_m$, so that the amplifier does not saturate.

For example, if $x_m = 5$ units, then $k_x = 15/5 = 3$.

By using magnitude scaling, full use is made of the voltage range available in an analog computer and also the overloading and saturation problems are taken care of. Scale factor for the derivates of the dependent variable

$$dx/dt, \qquad d^2x/dt^2$$

are also to be chosen by estimating their maximum values. To fully understand the concept of magnitude scaling, consider the second-order differential equation:

$$d^2x/dt^2 + a\,dx/dt + bx = f(t) \qquad \mathbf{1.3(14)}$$

where x is the dependent variable and t is the independent variable. The coefficients a and b are positive constants. The dependent variable and its derivates are available as voltages at the output of amplifiers on the computer. Magnitude scale factors are used to correlate the voltages from amplifiers to the dependent variable and its derivates.

In case of a vibration problem, x may represent displacement, dx/dt can stand for velocity, and d^2x/dt^2 for acceleration.

$$\text{Let } x1 = dx/dt \quad \text{and} \quad x2 = \frac{d^2x}{dt^2}$$

if the maximum anticipated values for x_m, $x1_m$, and $x2_m$. In that case the scale factors are:

$$k_1 = 15/x_m,\ k_2 = 15/x1_m, \quad \text{and} \quad k_3 = 15/x2_m \qquad \mathbf{1.3(15)}$$

Equation 1.3(14) can be rewritten as:

$$\frac{d^2x}{dt^2} = -a\frac{dx}{dt} - bx + f(t) \qquad \mathbf{1.3(16)}$$

Using the scale factors, the variables at the output of the amplifiers are $[k_1x]$, $[k_2x1]$, and $[k_3x2]$. Rewriting the equation using $[k_1x]$, $[k_2x1]$, and $[k_3x2]$, we get

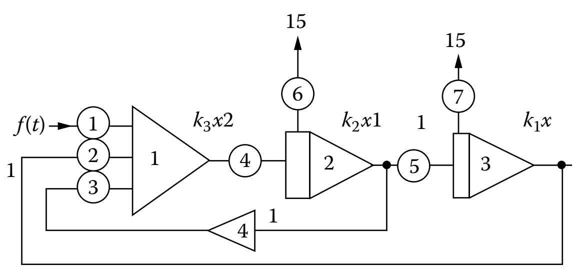
$$[k_3x2]/k_3 = -a[k_2x1]/k_2 - b[k_1x]k_1 + f(t) \qquad \mathbf{1.3(17)}$$

or

$$[k_3x2] = -a(k_3/k_2)[k_2x1] - b(k_3/k_1)[k_1x] + k_3f(t) \ \mathbf{1.3(18)}$$

The analog computer circuit for Equation 1.3(18) is shown in Figure 1.3m, which is a generalized circuit for solving a second-order differential equation with magnitude scaling.

Time Scaling It may be noted that the analog computer integrates with respect to time measured in seconds. But the independent variable in a differential equation representing a real time problem may be time in minutes or length in feet. The time needed for the solution of a particular problem may be 30 minutes, in which case 30 minutes of operators' time and the computer time would be needed, if time scaling was not used to speed up the solution.



Potentiometer number	1	2	3	4	5	6	7	
Values of its setting	k_3	$b\,k_3/k_1$	$a\,k_3/k_2$	k_2/k_3	k_1/k_2	$k_2x/1(0)/15$	$k_1x/(0)/15$	← Magnitude scaling
Values of its setting	k_3	$b\,k_3/k_1$	$a\,k_3/k_2$	$k_2/b\,k_3$	$k_1/b\,k_2$	$k_2x1(0)/b15$	$k_1x/(0)/15$	← Magnitude and time scaling

FIG. 1.3m
A generalized analog computer circuit for solving a second-order differential equation with magnitude scaling (Equation 1.3[18]).

For some fast-acting processes, such as transients in circuit breakers, slowing down the problem is needed to capture and record the variation. Hence time scaling is needed whenever it is necessary to speed up or slow down the process when it is studied in an analog computer.

If one designates the computer time as t and the real time as τ, then the time scale factor can be defined as the ratio of $t/\tau = \beta$. The steps in the procedure that was adopted for time scaling are listed below:

Step 1. Assuming $\beta = 1$, magnitude scale the problem as described earlier.
Step 2. Choose β, $\beta > 1$ when speeding up the solution or $\beta < 1$ to slow down the solution. The actual values can be selected from knowing the real process.
Step 3. Multiply the gain of each input to an integrator by I/β
Step 4. Modify the time in the forcing function from $f(t)$ to $f(\tau/\beta)$.

The above procedure can be applied to Equation 1.3(18) to implement the time scaling:

$$\left[\beta^2 k_3 d^2x/d^2\tau\right] = -a(k_3/k_2)\left[\beta k_2 \frac{dx}{d\tau}\right] - b(k_3/k_1)[k_1 x] + k_3 f(\tau/\beta) \qquad 1.3(19)$$

The initial conditions for potentiometer settings are given in the table at the lower part of Figure 1.3m.

Nonlinear Components

For solving nonlinear equations using analog computers, one needs nonlinear components[1] such as function generators, multipliers, dividers, and switches. These nonlinear elements significantly extend the usefulness of a computer by allowing nonlinear equations to be solved with the same ease as linear ones.

Function Generator The most common function generator used is a diode function generator.[2] This circuit approximates a function $y = f(x)$ by a number of straight-line segments of different scopes. To generate a functional relationship $e_o = \sin(2e_i)$, a number of line segments with different slopes are needed, as shown in Figure 1.3n.

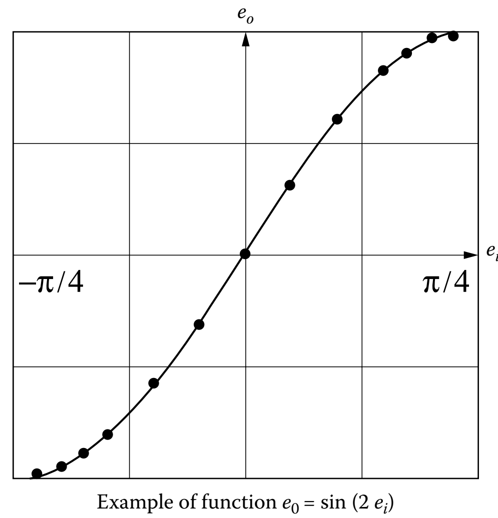


FIG. 1.3n
The response of a sine function generator.

Multiplier To obtain the instantaneous product of two time-varying voltages, a multiplier is needed. A convenient symbol for the multiplication block is shown in Figure 1.3e. Three types of multipliers are used in computers.[2] They are:

1. The quarter-square multiplier
2. The servo mechanical multiplier
3. The time division multiplier

Quarter-Square Multiplier The identity of the quarter-square multiplier can be described as:

$$xy = 1/4[(x+y)^2 - (x-y)^2] \qquad 1.3(20)$$

The analog computer circuit that corresponds to the above equation is shown in Figure 1.3o. Today, the quarter-square card is available from many manufacturers; this card gives a current output, which is proportional to the product of two voltages divided by 10.

The Analog Computer Unit

Commercially available analog computer equipment is box-type equipment with an orderly collection of amplifiers, power supplies, potentiometers, reference power supplies, diodes, switches, and some function generators, quarter-square cards, etc. The connections from the various components are brought to a panel called a patch board. The patch board is removable, and many spare patch boards are available. This facility is for wiring a computer circuit when the computer is to operate on another problem. The wired computer circuits can also be stored and used later.

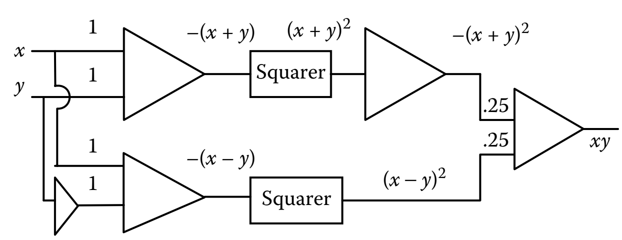


FIG. 1.3o
Analog computer circuit for a quarter-square multiplier.

In many analog computers, the resistors and capacitors used to construct summers and integrators are not accessible to the user. They are located inside the computer, and only their terminals are brought out to the patch board.

A reference power supply is provided for establishing initial conditions on the integrators. X-Y plotters, oscilloscopes, and strip chart recorders are used to record or observe the response or solution.

HYBRID COMPUTERS

A hybrid computer is a combination in hardware and software of one or more analog and digital computers. It aims at providing faster, more efficient, and more economical computational power than is available with computers of either type alone. The results depend to a large extent on the exchange of information between the analog and the digital computers and on the compatibility in operations and mutual interactions between the two parts.

A hybrid computer provides for the rapid exchange of information between the parallel and simultaneous computations and simulations within the analog computer and the serial and sequential operations of the digital computer. This information exchange links the two computational domains and offers the combined advantages of the fast and flexible analog computer with the precise and logic-controllable digital computer.

The extent of the information exchange between the two parts and the sophistication of the control structures and instruction repertoires determine the capability and the capacity of the hybrid computer. Best results are obtained when both computers are designed and developed with hybrid applications as the major purpose. If a hybrid computer is made up of general-purpose analog and digital computers, with an interface tailored to these, the resulting hybrid computer often poses severe limitations in equipment complement and operational features.

Hardware

The distinguishing feature of a hybrid computer is the interface that connects the analog and digital computers. The interface consists of data communication channels in which information is passed between the two computational parts. The interface does not carry out computations, but it may contain equipment that incorporates computational units, such as multiplying digital-to-analog converters.

The interface contains a number of conversion channels in which information is converted between an analog signal (voltage) and an encoded numerical (discrete) digital computer representation, according to programmed instructions and hardware executions. The number of conversion channels states the total parallel capacity of the interface, for conversions in both directions. In practice, the number of A/D (analog-to-digital) channels may differ from the number of D/A channels, depending on applications and implementations.

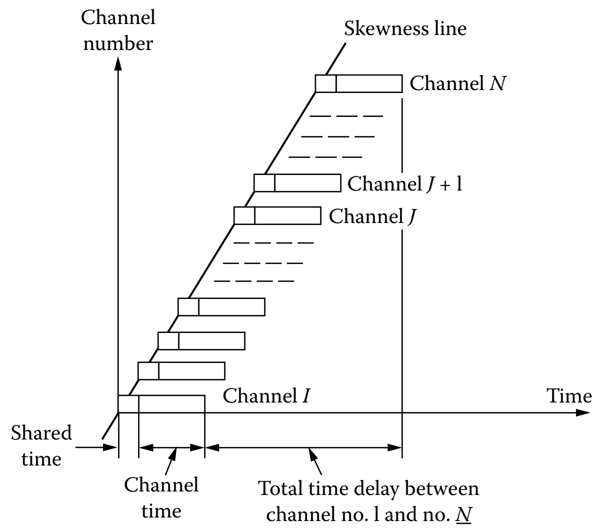


FIG. 1.3p
The skewness of converted data.

Since the conversion channels link parallel and concurrent analog computer variables with serial and sequential program steps in the digital computer, the interface must provide storage facilities, by which information (variables) can be stored, while all channels are being prepared (loaded). This way, all conversions can take place simultaneously, in terms of analog variables, and sequentially, in terms of digital variables.

If the converted information is not buffered, it reflects computational variables or conditions that are not concurrent or coreferenced in time. Such skewness is indicated in Figure 1.3p, showing the effect of a sequential conversion capability (among several time-shared units). To a degree, and at the cost of computer time, the skewness effects may be reduced by a common hold mode for all analog computing units, during which hold the information conversion may take place. Fast, accurate mode control facilities that allow rapid computer interruption and resumption of the analog program are necessary for this.

Figure 1.3q shows an example of an A/D conversion channel in which the analog signal is tracked and stored in analog form. The conversion channel utilizes a multiplexer and converter that are shared among a number of channels, typically 24 to 36, and it reads the converted information into a programmed (controlled) memory location in the digital computer. From this memory location the digital program can then obtain the converted information.

An example of a D/A conversion channel is shown in Figure 1.3r. It consists of a buffer register for the digital information and converter unit. The buffer register holds the digital information until the moment of conversion, when it is loaded into the converter with other D/A channels. For single-channel or continuous D/A conversion, the buffer register is sometimes bypassed, and the digital information read directly (jammed) into the converter.

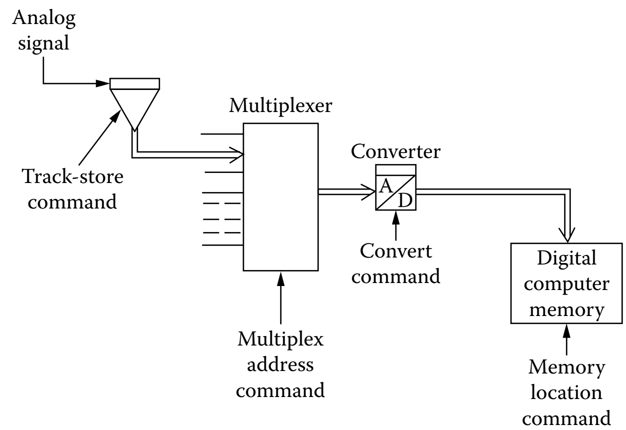


FIG. 1.3q
The components of an A/D channel.

The analog conversion output is a direct signal output or, as in a multiplying D/A, the product of an analog signal input and the converted D/A information. The conversion moment is only triggered by digital computer instructions, or indirectly, by interrupts form the analog computer through the digital computer program.

A number of status channels in the interface provide binary information exchange between the analog and digital computers. This information relates to the status of particular computational variables or computing units and to the condition and progress of the programmed functions. Status channels are used to ensure proper relationships in terms of operation sequences, timing of program events, steps, and coordination of the two hybrid computer parts.

There are three main types of status channels, depending on the importance or immediacy of the reported conditions:

1. Status indicating, which at all times corresponds in status to the condition of its input variable
2. Status retaining, which will retain the predetermined status of the input variable from the first time it is set or activated until the status channel is interrogated by the receiving computer and the channel is reset or deactivated
3. Status responding, which will respond immediately to the condition of the input variable when this is set or activated and will cause the receiving computer to interrupt its task or change its modes or functions according to the programmed actions

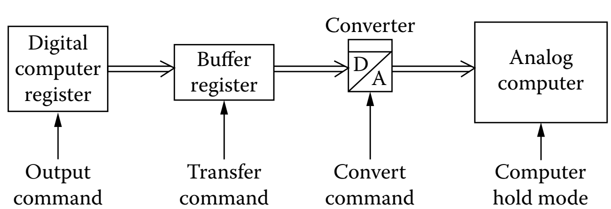


FIG. 1.3r
The components of a D/A channel.

Status channels are generally furnished in the interface as one-way signal lines, which must be assigned to the conditions or computing units that are to be reported during the programming.

The interaction and control of operations between the two computers are handled by binary command channels. The command channels represent direct, fixed interactive links between the control systems or command sources of the two computers. They control the executions of programs, such as start and stop computations and iterations, and the initialization or reset of programmed functions and routines.

Operation

The operational efficiency of a hybrid computer depends on the command and control (instruction) structures of the two computers. Analog and digital computers are sufficiently different in organization and operation to present profound problems in terms of command characteristics and functional orientation. In a hybrid computer, the analog and digital computers are linked together on the fundamental control level, providing facilities for mutual interruption and interaction of tasks.

In terms of analog computer control, the hybrid computer permits the digital computer program to carry out the control system functions previously outlined. These functions span the setup and checkout of analog and hybrid programs, the initialization and presetting of conditions for the computations and simulations, and the measurement, recording, and monitoring of the analog computer variables and functions during productive runs. Of prime importance in a hybrid computer is the ability of the computer programs to govern the progress of the computations and to take the appropriate control actions, depending on the obtained results and responses.

The digital computer instruction repertoire normally includes many bit-handling instructions designed to facilitate the exchange of information through the status channels, command channels, and direct mode control and command channels. These instructions permit multiple-level priority assignments and convenient handling of the exchanged information, which can be either in a converted data format or in single-bit format.

The fast access to information in the digital computer memory through direct access or cycle stealing is important for high-speed conversion channel utilization and efficient hybrid computations.

Software

Good and complete software is required to attain the optimum operational efficiency and utilization of the hybrid computer. This is especially important in hybrid computer applications since the complexity and extent of many hybrid programs and the sophistication of the instruction repertoire and control routines in the computers otherwise limit the usefulness and understanding of the computer capabilities.

The software is designed with practical problem-solving objectives in mind, such as handling the system's functions and presenting conditions and variables in concise and efficient formats. For best results, the software is written for a particular hybrid computer and is defined and specified according to the characteristics of the hardware configuration. The software consists of three types of system programs:

1. Batch-oriented hybrid computer operation programs, organized and utilized for large, complex hybrid problems, with extensive program setup and checkout demands
2. Conversational computer operation programs, designed for experimental and development-oriented type hybrid computations and simulations
3. Utility routines designed for efficient and convenient setup, checkout, and documentation of intermediate or limited hybrid computer programs, such as test programs, experimental circuit evaluations, and hybrid program developments

The software enables the operator to carry out a hybrid computation or simulation of the specific problem with a minimum of effort. In a hybrid application, this includes determining the signal connections and tie-lines, calculating the scale factors and coefficients of analog variables, adjusting coefficient units to the appropriate values prior to or during computations, and selecting the states or modes of the analog computing units.

An important aspect of the software is its capabilities for readout and documentation. It must be able to obtain the values within the hybrid program and to decode or interpret these in the language of the problem (such as in engineering units or mathematical terms). Finally, it must be able to make this information available to the operator, either on CRT, graphic display, trend curve, or in other forms.

PROCESS SIMULATION

The simulation of most processes, reactions, and plants requires the developing of a representation or building a model of common elements. For this model, each element is adapted to the particular functions it must simulate. When all the parts have been defined and assembled, the overall effects and constraints may be imposed, such as material and energy balances, operating modes, limitations, instrumentation, and control system characteristics.

Laplace Transforms

For linear control systems, Laplace transforms can be used to obtain their transfer functions. Laplace impedances can be used when working with Laplace transfer functions for system simulation. Information on Laplace impedance is available in most books on circuit theory or network analysis,[3] while these Laplace impedances are referred to as complex impedances in control system books.[4]

For the two terminal elements like resistance, capacitance, and inductance the impedance are given by R, $1/Cs$ and Ls, respectively. If these complex impedances are connected in series, the total impedance is the sum of the individual complex impedances, and if these are connected in parallel, then the parallel resistance rule is applied. It may be noted that complex impedances are valid only for such transfer function approaches where the initial conditions are assumed to be zero. For example, a series combination of R and C has a total impedance of

$$Z = R + 1/Cs = (RCs + 1)/Cs \qquad \textbf{1.3(21)}$$

The impedance of R and C connected in parallel is obtained by equating the reciprocal of the parallel impedance, i.e.,

$$1/Z = 1/R + Cs = (1 + Rcs)/R$$

Then

$$Z = R/(1 + RCs) \qquad \textbf{1.3(22)}$$

First-Order System Simulation The first-order transfer function $A/(1 + sT)$ can be simulated by the circuit shown in Figure 1.3s:

$$Y/X = Z_f/Z_i = R/R_1(1 + Rcs) = A/(1 + Ts) \qquad \textbf{1.3(23)}$$

where $A = R/R_1$ and $T = RC$.

Another way of obtaining a simulation circuit is as follows: First convert the Laplace function into a time function:

$$(1 + Ts)Y = AX \qquad \textbf{1.3(24)}$$

Then, in order to get the time domain equation, replace s and d/dt in Equation 1.3(24):

$$\begin{aligned} y + T d/dt\ y &= Ax \\ d/dt\ y &= -y/T + A/Tx \end{aligned} \qquad \textbf{1.3(25)}$$

The computer circuit for this given in Figure 1.3t.

Second-Order System Simulation In order to simulate a second-order system, which is in the form of $K/(1 + sT_1)(1 + sT_2)$,

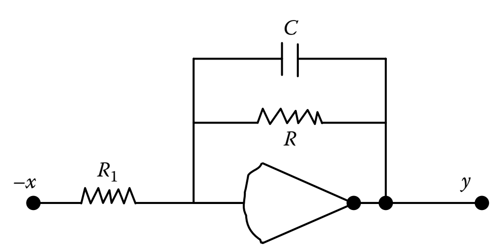


FIG. 1.3s
This circuit simulates a first-order transfer function A/(1 + sT).

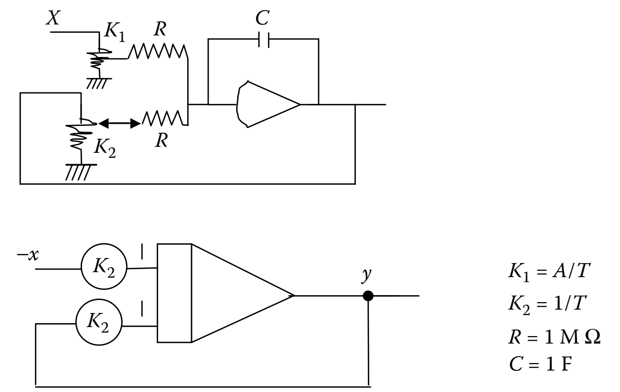


FIG. 1.3t
This alternate circuit also simulates a first-order transfer function A/(1 + sT).

one can connect two first-order system circuits (which are shown in Figure 1.3s) in cascade, but such a circuit cannot be used to simulate underdamped systems.

In order to simulate a second-order system of the form

$$G(s) = Y/X = k\left(s^2 + 2\xi w_n s + w_n^2\right) \qquad \textbf{1.3(26)}$$

where $\xi < 1$, then the circuit shown in Figure 1.3u can be used. To obtain this circuit, one can first rewrite the transfer function into a differential equation form, using the following steps.

Cross-multiplying Equation 1.3(26)

$$s^2Y + 2\xi w_n sY + w_n^2 Y = kX \qquad \textbf{1.3(27)}$$

and replacing s by d/dt, one obtains:

$$\frac{d^2}{dt^2}y + 2\xi w_n \frac{dy}{dt} + w_n^2 y = kx \qquad \textbf{1.3(28)}$$

Keeping d^2/dt^2

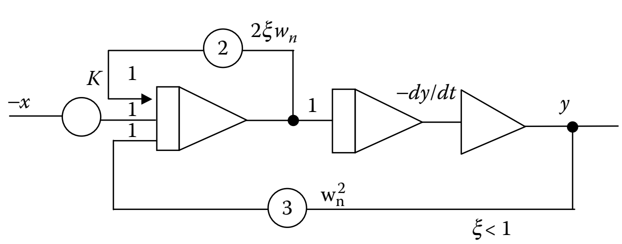


FIG. 1.3u
This circuit simulates an underdamped second-order system.

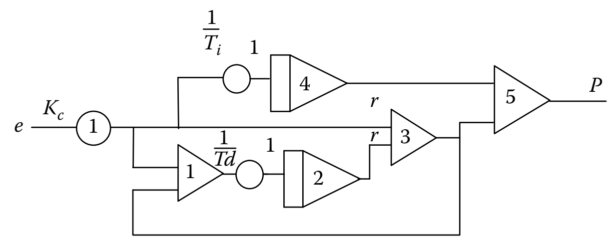


FIG. 1.3v
Analog simulation of a PID controller.

on the left-hand side and shifting all other terms to the right and multiplying the whole equation by −1, we obtain:

$$-\frac{d^2}{dt^2}y = 2\xi w_n \frac{dy}{dt} + w_n^2 y - kx \qquad \textbf{1.3(29)}$$

For Equation 1.3(29), the circuit is shown in Figure 1.3u.

Simulation of PID Controller The transfer function that represents a proportional, integral, and derivative (PID) controller is:

$$\frac{P(s)}{E(s)} = K_c\left[\frac{1 + T_D s}{1 + T_D s/r} + \frac{1}{T_1 s}\right] \qquad \textbf{1.3(30)}$$

When r is made very large, the transfer function reduces to

$$\frac{P(s)}{E(s)} - = k_c\left[1 + T_D s + \frac{1}{T_1 s}\right] \qquad \textbf{1.3(31)}$$

This is the transfer function of an ideal PID controller.[4] The analog computer circuit that simulates it is shown in Figure 1.3v.

Lag Functions

A common element in the process and plant simulations is the lag, defined by the transfer function (in Laplace notation) as:

$$H(s) = \frac{Y(s)}{X(s)} = \frac{K}{1 + \tau s} \qquad \textbf{1.3(32)}$$

where K is the steady-state gain and τ is the time constant. In general, the gain and time constant will vary with other simulation variables, such as flows, temperatures, and pressures. A generalized circuit for the lag function is shown at the top of Figure 1.3w. Other versions of the lag are also shown in the figure with different aspects and advantages, such as independent adjustment of gain and time constant, bipolar output, and simplicity and independent adjustment (but poorly scaled in many cases).

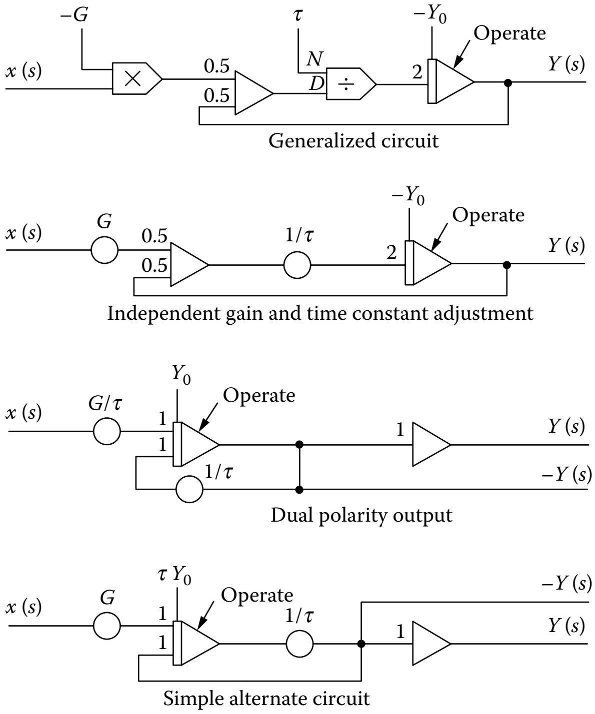


FIG. 1.3w
The lag function.

Lead-Lag Element Another commonly used element is the lead-lag, for which the transfer function is:

$$H(s) = K\frac{s}{1+\tau s} \qquad \textbf{1.3(33)}$$

or

$$H(s) = K\frac{1+\tau_1 s}{1+\tau_2 s} \qquad \textbf{1.3(34)}$$

Figure 1.3x shows a generalized circuit for the transfer function described in Equation 1.3(33), with variable gain K and time constant τ.

The class of lead-lag transfer functions described by Equation 1.3(33) may be simulated with circuits of the type shown in Figure 1.3y. The generalized arrangement used for

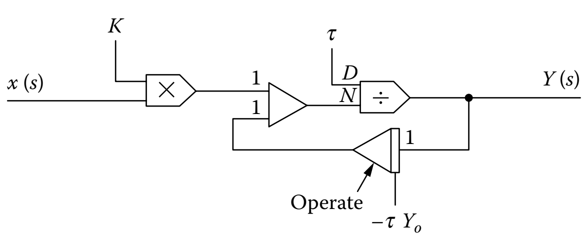


FIG. 1.3x
The lead-lag function for Equation 1.3(33).

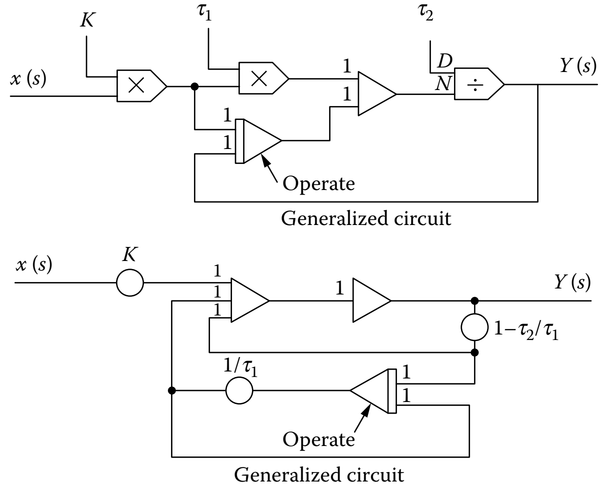


FIG. 1.3y
The lead-lag function for Equation 1.3(34).

individual time constants and gain variations by signal inputs often results in poor integrator scaling, especially for large τ_2 values. The common circuit permits the ratio between τ_1 and τ_2 to be adjusted over a wide range.

Delays, Transport Lags In many analog computer simulations, a signal delay effect, or simulation of a transport lag, such as the flow of fluids through pipes or the movement of material on a conveyor, is desired. Delays or dead times such as this often represent important dynamic factors and influence the stability and operation of the plant or process. In general, there is no true analog computer "model" for a transportation delay, and for most applications some form of approximation of the delay function must suffice.

The most commonly known approximations are the Padé functions. These are often unsatisfactory in process and plant simulations, and other empirically determined functions are used in their place. Figure 1.3z shows a delay approximation, for which the time domain responses are better than those of the Padé functions and give more stable, damped, and consistent performance. The delay approximation is the low-pass type, with the transfer function:

$$H_2(s) = \frac{3.45 - 0.345(\tau s)}{3.45 + 3.18(\tau s) + (\tau s)^2} \qquad \textbf{1.3(35)}$$

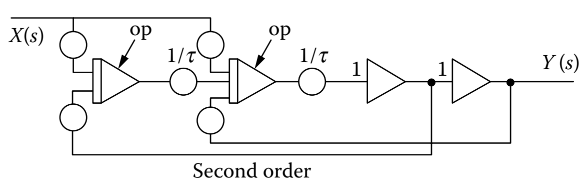


FIG. 1.3z
Approximation of the transport lag.

In Figure 1.3z, the transportation delay time (τ) can be varied by adjusting the coefficient potentiometers. If the delay time is a function of a simulation variable, such as the flow rate in a pipe, multipliers must replace the potentiometers. For the circuit described, the range of variation is generally limited to less than 100 : 1, without rescaling or rearranging the gain distributions. The shortest delay is determined by the highest input gains available (or by gain–integrator speed combinations).

Control System Simulation

The simulation of instrumentation control systems includes measuring or detecting devices, the controllers, and the actuating or manipulating output elements. Most measuring devices may be simulated by one or more simple lags for dynamic representation, with added nonlinearities for device characteristics, sensitivity, or operating point modeling.

For special functions, such as logarithmic input–output relationships (such as pH measurements), the general-purpose function-generating units may be used or the functions generated by implicit techniques. When the measuring device puts out a discontinuous (pulse-type) signal, signal-generating circuits may be used, as discussed later in this section.

Interacting PID Controller A direct, three-mode (PID) controller has the transfer function:

$$M(s) = \frac{100}{\text{PB}}\left[1+\frac{1}{T_i s}\right]\left[1+\frac{T_d s}{1+T_0 s}\right][R(s)-C(s)] \qquad \textbf{1.3(36)}$$

where $M(s)$ = controller output in Laplace representation; $C(s)$ = measured (controlled) input variable; $R(s)$ = reference (set point); PB = proportional band (in percent); T_i = reset (integration) time; T_d = derivative (rate) time; and T_0 = stabilizing (filtering) time constant.

The filter time constant is usually made as small as possible and is often a fraction of the derivative time T_d (such as $T_d/16$).

Figure 1.3aa shows simulation circuits for direct and reverse acting three-mode controllers, with the initial value M_0 corresponding to the controller output when it was in "manual."

The controller transfer function in Equation 1.3(36) is of the interacting type, which is the common case for most industrial applications.

Noninteracting PID Controller A mathematically noninteracting controller is expressed by:

$$M(s) = \left[\frac{100}{\text{PB}}+\frac{1}{T_i s}+\frac{T_d s}{1+T_0 s}\right][R(s)-C(s)] \qquad \textbf{1.3(37)}$$

Here all the three modes can be independently adjusted.

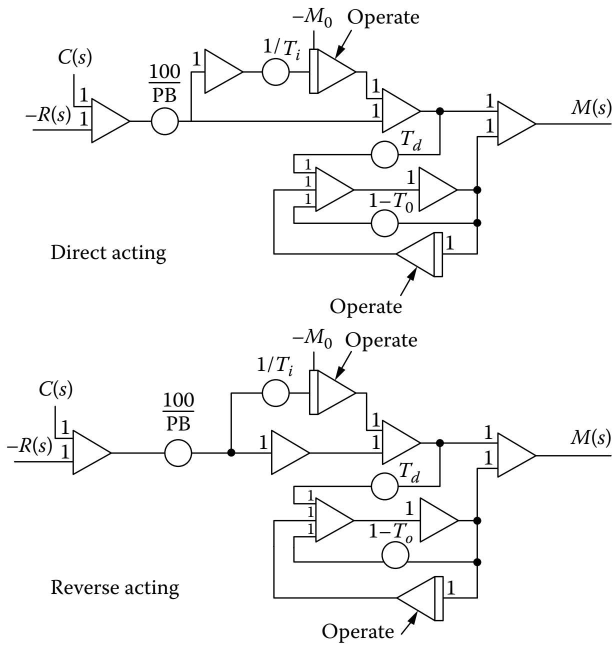


FIG. 1.3aa
Interacting PID controller described by Equation 1.3(36).

In general, the proportional band should have an *overall* adjusting effect, which leads to the practical noninteracting controller with the transfer function:

$$M(s) = \frac{100}{\text{PB}}\left[1+\frac{1}{T_i s}+\frac{T_d s}{1+T_0 s}\right][R(s)-C(s)] \qquad \textbf{1.3(38)}$$

This circuit arrangement is shown in Figure 1.3bb.

More modern controllers often limit the derivative action so that it will respond only to changes in the measured variable and not to rapid set point changes, which could upset the process. This characteristic is described in the modified transfer function:

$$M(s) = \frac{100}{\text{PB}}\left[1+\frac{1}{T_i s}\right]\left[R(s)-\left(1+\frac{T_d s}{1+T_0 s}\right)C(s)\right] \qquad \textbf{1.3(39)}$$

which is of the interacting kind.

Hybrid Simulation

A hybrid controller can simulate direct digital control of a process. In that case the individual PID control algorithms can be simulated by the digital computer, while the process is simulated using an analog computer, and the two are interconnected for system analysis and design.

Simulation of Direct Digital Control Direct digital control may be simulated with a controller circuit as shown in Figure 1.3cc, where the measured variable $C(s)$ and the set point $R(s)$ are sampled at fixed intervals (Δt) and held in

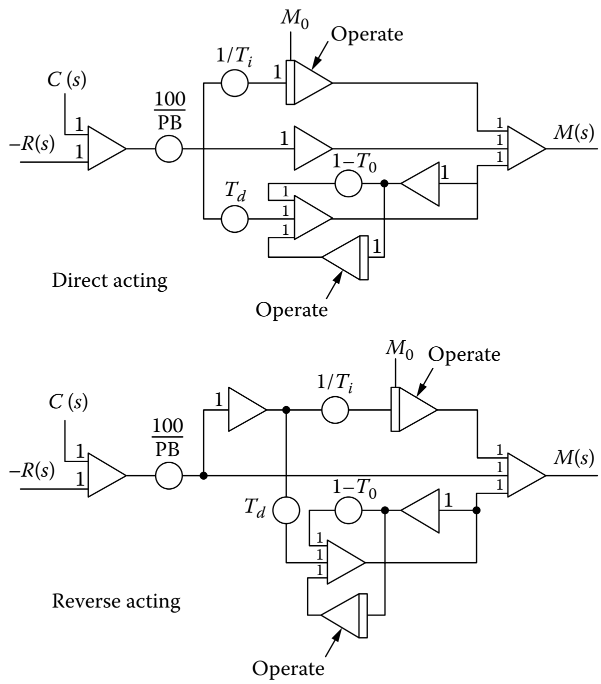


FIG. 1.3bb
Practical noninteracting controller described by Equation 1.3(38).

track-store units. The controller algorithm for this case is given by Equation 1.3(40):

$$\Delta m = \frac{100}{\text{PB}}\left[\left(1+\frac{\Delta t}{T_i}\right)(r_n - c_n) - r_{n-1} + c_{n-1} - \frac{T_d}{\Delta t}(c_n - 2c_{n-1} + c_{n-1})\right] \quad \textbf{1.3(40)}$$

with Δm being the change in the output, which is computed in the time interval. This control corresponds to the one given in Equation 1.3(39), with derivative action responding to the measured variable only. The track-store operation is controlled by the pulse P occurring once during each time interval

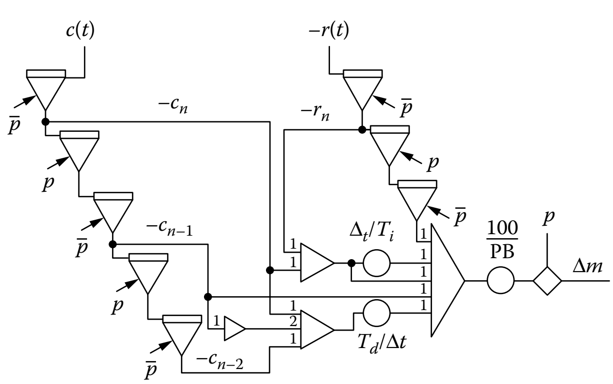


FIG. 1.3cc
Analog simulation of direct digital control based on Equation 1.3(40).

Δt to produce an output pulse Δm of amplitude (height) corresponding to the desired change in the control variable.

Simulation of Control Valves The most important manipulating output element is the control valve, which can have several distinct functional aspects in an analog computer simulation. The flow characteristics can be linear, equal percentage, quick opening, or butterfly type.

Except for the linear valve, the flow characteristics must be generated by a special analog computer circuit (by implicit techniques), or programmed in a function-generating unit, for a true representation (see Chapter 3 for details). Additional effects such as velocity limiting or backlash in valve stem movements must be included in the model.

The dynamic performance of the control valve may be represented by a time lag of first or second order, with a limited velocity in stem movement. The time lags are simulated by circuits that were described under transfer functions, lags, and lead-lags. Velocity limiting may be expressed by the equation

$$\frac{dm_0}{dt} = \text{LOW}\left(m_L; \frac{dm_i}{dt}\right) \quad \textbf{1.3(41)}$$

where m_0 is the stem position, m_L the velocity limit, and m_i the input stem position of an ideal, unconstrained valve. The simulation circuit for velocity limiting is shown in Figure 1.3dd.

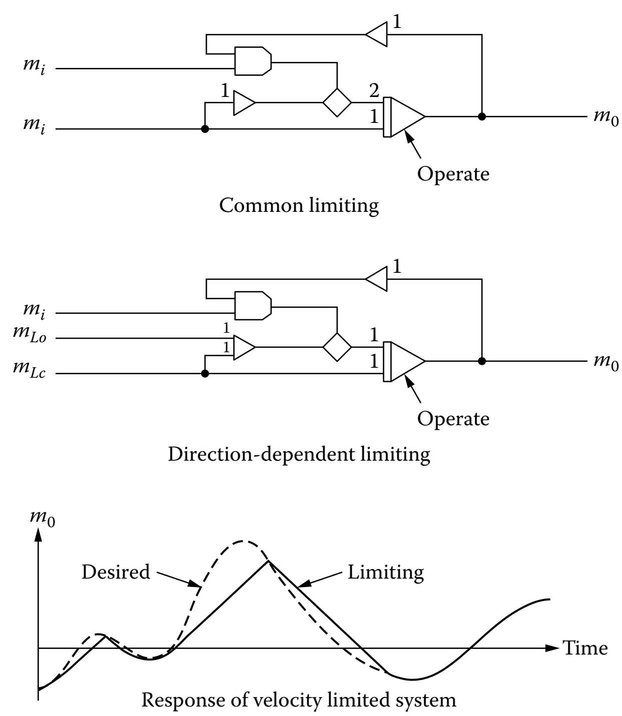


FIG. 1.3dd
The analog simulation circuit and response of a velocity-limited system.

References

1. Coughanour, D. R., and Koppel, L.B., *Process Systems Analysis and Control,* New York: McGraw-Hill, 1965.
2. Jackson, A. S., *Analog Computation,* New York: McGraw-Hill, 1960.
3. Van Valkanburg, M. E., *Network Analysis,* Eaglewood Cliffs, NJ: Prentice Hall, 2000.
4. Ogata, K., *Modern Control Engineering,* New Dehli: Prentice Hall of India, 2002.

Bibliography

Adams, J., et al., ''Mathematical Modeling of Once-Through Boiler Dynamics,'' IEEE Winter Power Meeting, Paper 31 TP65-177, 1965.

Ashley, J. R., *Introduction to Analog Computation,* New York: John Wiley & Sons, 1963.

Box, G., and Draper, N., *Empirical Model-Building and Response Surface,* New York: Wiley, 1987.

Brignell, J., and White, N., *Intelligent Sensor Systems,* Rev. ed., Bristol, UK: Institute of Physics Pub., 1996.

Cavicchi, T. J., *Digital Signal Processing,* New York: Wiley, 2000.

Dyer, S. A. (Ed.), *Survey of Instrumentation and Measurement,* New York: Wiley, 2001.

Ebert, K., et al., *Computer Applications in Chemistry,* Weinheim, Germany: VCH, 1989.

Eren. H., "Instruments," *Survey of Instrumentation and Measurement,* S. A. Dyer, 2001.

Fowler, K.R., *Electronic Instrument Design — Architecting for the Life Cycle,* New York: Oxford University Press, 1996.

Garrett, P. H., *Advanced Instrumentation and Computer I/O Design: Real-Time System Computer Interface Engineering,* New York: IEEE Press, 1994.

Grover, D., and Deller, R. J., *Digital Signal Processing and the Microcontroller,* Upper Saddle River, NJ: Prentice Hall PTR, 1999.

Hamell, E., "Modeling and Simulation," *InTech,* November 1982.

Haykin, S., and Kosko, B., (Eds.), *Intelligent Signal Processing,* New York: IEEE Press. 2001.

Howe, R. M., *Design Fundamentals of Analog Computer Components,* New York: Van Nostrand, 1961.

James, M. L., Smith, G. M., and Wolford, J. C., *Analog Computer Simulation of Engineering System,* Scranton, PA: International Textbook Company, 1966.

Jenness, R. R., *Analog Computation and Simulation: Laboratory Approach,* Boston: Allyn and Bacon, 1965.

Kasparek, S., *Computer Graphics and Chemical Structures,* New York: Wiley, 1990.

Korn, G., *Interactive Dynamic Systems Simulation,* New York: McGraw-Hill, 1989.

Lupfer, D. E., "Analog Computer Cuts Distillation Costs," *ISA Journal,* August 1966.

MacKay, D. M., and Fisher, M. E., *Analog Computing at Ultra-High Speed,* New York: John Wiley & Sons, 1962.

Madisetti, V. K., and Williams, D. B., *The Digital Signal Processing Handbook,* Boca Raton, FL: CRC Press, 1998.

McDonald, J. P., and Kwatny, H. G., "Design and Analysis of Boiled Turbine-Generator Controls Using Optimal Linear Regulatory Theory," *JACC*, 1972.

McDonald, J. P., and Kwatny, H. G., and Space, J. H., "A Non-Linear Model For Reheat Boiler–Turbine–Generator Systems, Parts I and II," *JACC,* 1971.

Mostafa, E. S., "An Operator Training Simulator," *Instrumentation Technology,* January 1982.

Pallás-Areny, R., and Webster, J. G., *Analogue Signal Processing,* New York: John Wiley & Sons, 1999.

Pallás-Areny, R., and Webster, J. G., *Sensors and Signal Conditioning,* 2nd ed., New York: John Wiley & Sons, 2001.

Paton, B. E., *Sensors, Transducers, and LabView,* Upper Saddle River, NJ: Prentice Hall PTR, 1999.

Peterson, G. R., *Basic Analog Computation,* New York: Macmillan, 1967.

Randy, F., *Understanding Smart Sensors,* 2nd ed., Boston: Artech House, 2000.

Rogers, A. E., and Connolly, T. W., *Analog Computation in Engineering Design,* New York: McGraw-Hill Book Co., 1960.

Scrivers, D. B., "Special Purpose Analog Computers," *ISA Journal,* July 1966.

Simpkin, L. J., and Wooden, L. F., "Use of Analog Computations of Power Plant Automatic Control Systems," IS, The Ninth National Power Instrumentation Symposium, Denver, CO, May 1964.

Tomovic, R. J., and Karplus, W. J., *High Speed Analog,* New York: John Wiley & Sons, 1962.

Vassos, B., and Ewing, G., *Analog and Digital Electronics,* New York: Wiley, 1985.

Webster, J. G. (Ed.), *The Measurement, Instrumentation, and Sensors Handbook,* Boca Raton, FL: CRC Press, 1999.