REAL-TIME DIGITAL DIFFERENTIAL ANALYZER (DART)

Loren P. Meissner
U. S. Naval Ordnance Laboratory
Corona, California

Introduction

The solution requirements of differential analysis problems are increasing to the point where solution speed requirements are often beyond the capabilities of present-day conventional digital computers, while analog computers will not provide sufficient accuracy. For many of the problems of this class, a computing system is needed which possesses the desirable characteristics of both analog and digital computers - that is, one which has the inherent speed of an analog computer and the basic accuracy of a digital computer. One approach to the problem of designing a computer having these characteristics is described in this paper. It involves the use of a system design based on the electronic analog computer, with components which are digital in nature.

Basic Concept

The concept on which the DART system design is based involves the combined use of pulse-train information (such as is used in digital differential analyzers like MADDIDA or the CRC-105) and of information in the form of sets of d-c gating voltages. The relation between these two kinds of information can be readily shown. If a train of pulses is fed into a binary counter, consisting for example of a row of flip-flops, a set of d-c voltages, constant between pulses, is available at the plates of the flip-flops. This set of voltages may be used to control logical gates or switches of various types; hence, they are referred to as "gating voltages."

A study is in progress at the Naval Ordnance Laboratory, Corona, to investigate the design of computing elements using both these kinds of information, and to investigate the incorporation of these separate computing elements into computing loops for the solution of systems of differential equations.

The pulse-train units under consideration in this project depend upon essentially the same logic as do other digital differential analyzers. No magnetic drum or other central memory is used, however, since a separate computing unit is provided for each operation. For example, if a problem requires three integrations, three separate integrators are used. These are plugged together in the same basic manner as the integrators of an analog computer.

By eliminating the memory, and thus eliminating the time-sharing feature of magnetic-drum computers, a much higher basic pulse rate is made possible, although at the expense of equipment.

Present designs are aimed at a 10-kilocycle basic pulse rate, but much higher rates are possible without requiring the use of any radically new circuit techniques.

Gating Matrices

Once a binary number is available as a set of gating voltages, other related sets of gating voltages can be produced by using gating matrices. For example, an adding matrix would give the sum of two numbers available in gating-voltage form; a multiplying matrix would give the product. Other functions which could be performed by using gating matrices would include the extraction of digits, limiting the value of a variable, and other logical operations.
Combined System

In combining pulse-train computing elements with gating matrices, to produce a computing system for the solution of differential equations, it is possible to take advantage of the fact that a pulse-train integrator contains a register, from which a set of gating voltages may be obtained. This is the register in which the dy pulses are counted to produce y (see Figure 1).

The set of gating voltages representing y may be combined with other sets in gating matrices, as required to solve the given system of differential equations. Figure 2 shows an example of the use of this method, to solve the system:

\[
\begin{align*}
dy &= p \cdot dt \\
-dp &= (p + 1) \cdot y \cdot dt
\end{align*}
\]

In this example, a pulse train, dp, is fed into the Integrating Register to produce p as a set of gating voltage. This is combined with the pulse train, dt, in the Rate Scaler to give the pulse train, pdt.

Since, by equation (1), this is equal to dy, it is again integrated to give y. The two sets of gating voltages now available, y and p, are combined in the Multiplying Matrix to produce py. This is added to y in the Adding Matrix. At this point, py + y, or (p + 1)y, is available as a set of gating voltages. Another Rate Scaler is used to produce (p + 1)ydt. According to equation (2), this equals -dp which, except for a simple sign inversion, is the same pulse train used originally.

This example illustrates the combined use of pulse-train units and gating matrices for the solution of a simple system of differential equations.

Feasibility

It remains to be shown whether gating matrices can be produced which are simpler or faster than the elements which would be required to perform the corresponding operations with pulse trains. As yet, no specific detailed designs have been worked out for any of these gating matrices. Preliminary analyses indicate, however, that such units can be designed which would have desirable features as compared with pulse-train systems used alone.

Pulse-Train Operations

There are a number of possible choices involved in the detailed design of computing elements to perform pulse-train operations. There are a number of different ways of handling signs, both in the pulse trains and in the registers of pulse-train integrators. Also, at least two basically different methods of designing rate scalers (the "transfer" method and the "sieve" or "scaled-train" method) have been used.

In the design of pulse-train components for DART, these choices have been made in light of the fact that it must be possible to use these components along with gating matrices.
Integration

A pulse-train integrator consists of an Integrating Register and a Rate Scaler. The Register contains a row of flip-flops which count the incoming pulses to produce a set of gating voltages representing a binary number, \( y \). The rate at which the number in the Register, \( y \), is changing depends upon the pulse-train rate, \( dy \).

The Scaler has as one of its inputs the number \( y \), represented as a set of gating voltages (which are constant between inputs to the Register). The other input to the Scaler is a pulse train, \( dx \), which in many cases is a uniform train of pulses originating in the clock-pulse generator. This pulse train is scaled down by the factor \( y/k \) (where \( k \) is chosen so that \( y/k \) is always less than unity). The output of the Scaler is, then, a pulse train, \( (y/k)dx \); that is, the number of pulses at the output of the Scaler is less, by the factor \( y/k \), than the number of pulses at the input.

This is accomplished by feeding the input pulse train, \( dx \), into a row of binary scalers to produce scaled trains \( dx/2, dx/4, \ldots, dx/2^n \). Each of these scaled trains corresponds to one of the binary digits of the number \( y \), with the most significant digit of \( y \) corresponding to the first scaled train, which has the largest number of pulses. The gating voltages representing "ones" in the binary number \( y \) allow their corresponding pulse trains to pass; those representing "zeros" do not. The scaled and gated pulse trains are then recombined to form the output pulse train, \( dz \).

The number of pulses in the output pulse train, \( dz \), is proportional to \( dx \sum dy \). This is a numerical approximation (according to the Rectangular integration formula) to the integral, \( Kydx \); \( K \) is a constant of proportionality which depends simply upon the number of binary digits in the number \( y \).

The integrator, then, has two input pulse trains, \( dy \) and \( dx \), and an output pulse train, \( dz = Kdy \). In addition, it should be remembered that the number \( y \) exists as a set of d-c gating voltages (constant between \( dy \) inputs) at the output of the Register.

Signs

As shown in Figure 3, each of the inputs and outputs consists of a pair of pulse trains, one (labeled "\( \Delta \)") carrying magnitude information, the other (labeled "\( -- \)") carrying sign information. A pulse on the "\( \Delta \)" line of the \( dy \) input represents an increment of \( y \) (a change in the lowest digit of the Register). If the "\( \Delta \)" pulse is accompanied by a pulse on the "\(--\)" line of the \( dy \) input, the increment is negative; if it is accompanied by no pulse on the "\(--\)" line, the increment is positive. If there is no pulse on the "\( \Delta \)" (magnitude) line, the Register is unchanged, whether or not there is a pulse on the "\(--\)" (sign) line.

Count Control

The number in the Register represents the absolute value of \( \int dy \); a separate flip-flop is used for the sign of \( y \). A positive increment must cause the absolute value, \( |y| \), to increase if the Register sign is positive and to decrease if it is negative; a negative increment must cause \( |y| \) to decrease if the Register sign is positive and to increase if it is negative. In other words, if the incoming increment has the same sign as the Register, the number in the Register must be made to increase in absolute value ("count up"). If the signs are opposite, the number in the Register must be made to decrease in absolute value ("count down"). The Count Control, shown in Figure 3, accomplishes...
this. The pulse on the "—" line is made available before the "Δ" pulse; a logical network is used to sense whether the Register and input signs are the same or opposite, and to produce a gating voltage on one of a pair of lines, ordering the Register to "count up" or "count down." When the "Δ" pulse arrives, the count order has been established, and the magnitude of the number in the Register increases or decreases as required.

Figure 4, showing in detail a single digit of the integrator, illustrates the manner in which the Register is made to count up or to count down. Essentially, the method simply requires that the pulse which causes a flip-flop to change state (the "carry pulse") come from either plate of the preceding flip-flop, depending upon whether it is desired to "count up" or "count down."

In the binary number system, "counting up" proceeds as follows: The lowest digit is changed (from 1 to 0 or from 0 to 1). If it is changed from 0 to 1, the process ends; if it is changed from 1 to 0, a "carry" occurs: i.e., the second digit is changed. If this digit is changed from 0 to 1, the process is finished; if from 1 to 0, the third digit is changed. This process continues until the first digit is reached which was initially 0. At this point, a change from 0 to 1 will occur, and nothing will happen to change any of the higher digits. Electronically, this is done by producing a "carry" pulse whenever a flip-flop changes from the "1" state to the "0" state; this carry pulse is used to flip the next higher flip-flop.

When "counting down," the process is exactly the same except that the roles of the 1's and 0's are reversed: a carry occurs from every digit until the first digit is reached which was initially 1. This requires a carry pulse whenever a flip-flop changes from "0" to "1." This simply means that the carry pulse must be derived from the opposite plate of the flip-flop when counting down. Outputs are available from both plates of every flip-flop; when counting up, carry pulses are taken from the "RO" plate of each flip-flop, and when counting down they are taken from the "RC" plate. The "RO" plate is the one which is down ("open") when the Register is reset to zero; the "RC" plate is up ("closed") when reset. The gating voltage is also taken from the "RC" plate, since this is the plate which is down ("open") when the flip-flop is in the "1" state.

Scaler

The Scaler consists simply of a row of flip-flops, with the number of pulses at the output of each flip-flop equal to half the number of pulses at its input. One somewhat unconventional feature, however, is that the pulses which are used to flip the next flip-flop are taken from one plate, and the pulses which are combined to form the dz output are taken from the other plate. This insures that no two flip-flops will feed pulses into the dz line on the same input pulse, and allows these pulses to feed directly into the output line from all the flip-flops without danger of coincidence.

Acknowledgment

The work described in this paper was sponsored in part under I.P.R. No. AF(33-00)-51-4345-E, between the U. S.-Air Force and the National Bureau of Standards, Corona (now the U. S. Naval Ordnance Laboratory, Corona), and in part by Foundational Research Funds allocated to the Naval Ordnance Laboratory, Corona.

The electronic design of the components described is being developed by Mr. E. O. Codier, Naval Ordnance Laboratory.
Bibliography


Figure 1. Pulse-Train Integrator (Simplified)

Figure 2. Use of Combined System
Figure 3. Pulse-Train Integrator, Showing Signs

Figure 4. Single Digit of Integrator