RAILWAY SIGNALING AND COMMUNICATIONS

MARCH, 1960

Computer Analyzes Train Operation

By Jerome O'Neill

In the June 1959 Railway Signaling and Communications an article described briefly how the General Railway Signal Co. had made use of an IBM 650 computer to predict train operation over a stretch of single track CTC then being designed for the CPR. The present report was delivered before the AAR Signal Section Convention in Washington, D.C., in October. It describes in greater detail the logic of the computer and how it may be of assistance in the design of track layouts and CTC signaling systems.

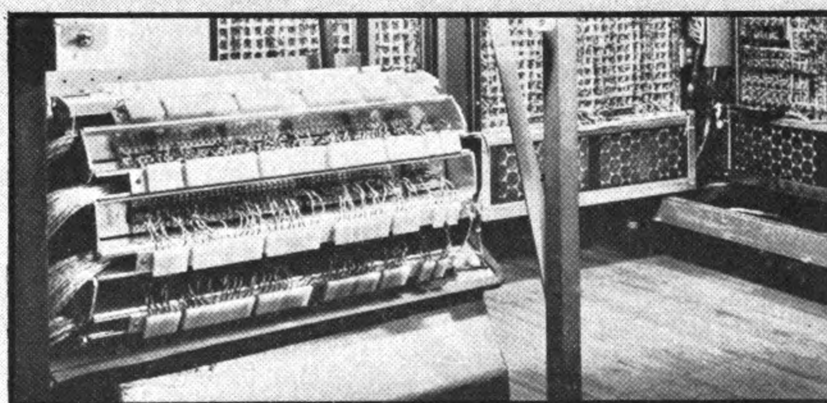
• In planning a CTC installation, many decisions must be made about the track and signal layout best suited to requirements. These are essentially a matter of engineering judgment, based on careful analysis of existing and prospective train operation.

A convenient method of analyzing train operation is to use available data, such as dispatchers' train sheets, delay reports, etc., to construct charts which show in graphic form how trains operated on days selected as typical. By redispatching trains on these charts—that is, by plotting the time-distance line for each train as it would have operated with CTC—it is possible to estimate savings that might be anticipated from a particular track and signal arrangement.

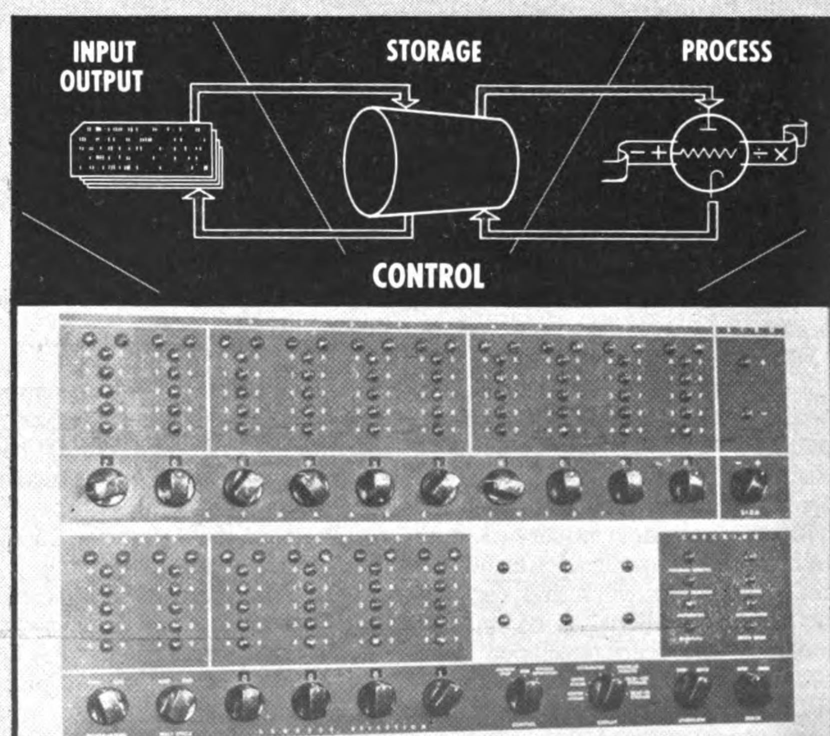
The train chart is a very useful aid to the planning engineer, but its preparation requires the accumulation of considerable data and many hours of drafting. Moreover, if contemplated changes in trains or track are substantial, data based on existing operations may not be directly applicable to the new condition, making accurate redispatching more difficult.

The thought naturally occurs—since redispatching is a prediction in any event—might it not be possible

Elements of the IBM 650 Computer



The 12,500 RPM magnetic drum can store 40,000 digits.



A representation of the computer's processing.

to predict train performances directly from physical data about trains and track? Given the basic facts about locomotives, train weights, grades and curves, can we use the formulas of physics and engineering to calculate the points on a train chart? If this could be done with sufficient accuracy, it would be a valuable supplement to graphic methods, and it might be possible to cut the time and work needed for a CTC study.

Complicated Problem

While easy enough to set forth as a concept, the problem turns out to be extremely complicated in practice. As a result, the necessary computations would take several years of a man's time working with a desk-type calculating machine.

But the idea still has great appeal. Concerned with a somewhat similar problem—the prediction of locomotive performance under varying conditions of service—the Pennsylvania Railroad developed and placed in service in 1950 a train performance calculator of the analog type which gave accurate results in a small fraction of the time required for desk computation. In the problems solved by the Pennsylvania machine, however, it is necessary to handle only one train at a time. The CTC problem requires that many trains be handled simultaneously. In addition, the solution must not only provide a reliable prediction of the progress of each train based on the physical formulas, it must also take account of the effects of trains on each other through the signal system, and the need for trains to meet and pass.

Until the recent past, this problem was too complex to be handled economically by available means of computation. But the development of electronic digital computers has changed the picture. Using the IBM 650 computer now in service at the General Railway Signal Co. for general accounting work and for engineering calculations, a computation program has been developed which provides an encouragingly accurate simulation of train operation on a single-track railroad.

It may be helpful at this point to visualize in simplified form the elements of the IBM 650 computer. First, there is an input-output section through which information is fed into the computer, and out of which come the answers to problems.

Next, there is a storage section. This is used for storing data, instructions, tables, intermediate results, final results, and any other desired information. Heart of the storage section is a rotating drum with a magnetic surface on which information is recorded in patterns of magnetized spots. There are 2,000 storage locations on the drum, each identified by a four-digit number called its address. These run from 0000 to 1999. Each location can store a 10-digit number, together with a plus or minus sign. Each 10-digit number is called a "word," so that there is storage capacity for 2,000 words.

The drum rotates continuously at 12,500 rpm. Any address on the drum can be located and data read or entered in less than five thousandths of a second. Reading information from the drum has no effect on its permanency; the magnetized spots remain until erased.

Third, there is the process section. Here is where the computer acts like a super desk calculator, but using electronic circuits instead of gears, dials, and a carriage. Here, among other things, the computer does addition, subtraction, multiplication, and division; shifts numbers left or right; and lines up decimal points. To illustrate its capacity and speed, this section of the computer can divide a 20-digit factor by a 10-digit factor to give a 10-digit quotient and a 10-digit remainder in less than 25 thousandths of a second.

Control Section Coordinates

Finally, we have the control section. This is the electronic robot that coordinates the operation. It can:

1. Locate any specified address on the drum.
2. Read the data or instructions at the address.
3. Direct the process section to carry out arithmetical operations specified in the instructions, using specified numerical data.
4. Store results at any specified address.
5. Look up numbers in tables stored on the drum.
6. Branch—that is, choose between two different instructions, depending upon whether a specified number is plus or minus, zero or non-zero, or if the digit 8 or 9 occurs at a specified position in the number.

The ability to branch, to make a logical choice between two courses of action, is an extremely important feature of electronic data processing computers. It is an essential element in the computer's ability to simulate train operation.

Before the computer can work out a problem, it must have two kinds of information: the basic physical data, and the step-by-step instructions on how to handle this data. These instructions are known as the program. Preparing the program is the most important element in using a computer to solve a problem. The great bulk of the work of preparing this program was done by two GRS engineers, Willis R. Smith and Robert T. Coupal.

Two Phases to Programming

There are two phases to programming. First the programmer analyzes the problem, breaks it down into fundamental logical steps. To help visualize these steps, flow diagrams are usually prepared which show the order in which the operations occur, and the branching that may be necessary to handle different conditions. Each step must be based on the computer's ability to add, subtract, multiply, and divide, to look up tables, and to branch. No matter what the nature of the problem, the programmer must devise ways of expressing conditions so that they can be handled by these operations.

The second thing the programmer does, after the logic of the program is decided, is to prepare the specific instructions, step by step, for the operations necessary to carry out the planned procedure. There are various ways of doing this, but whatever the method, the instructions must eventually be written in the only language the computer understands—the language in which the words are the 10-digit numbers mentioned earlier.

Although the program we are concerned with comprises many subprograms, it may be conveniently considered as made up of two major sections: a calculation program and a route-logic program.

The calculation program determines how trains would move, considering only the physical effects of locomotive power, train weight, rolling resistance, grades, and curves, but taking account of speed limits set by operating and signal rules. This is done by computing train acceleration, and then integrating twice by arithmetical means to get speed and distance.*

The information fed to the computer to compute acceleration values from this formula is: train tonnage, locomotive horsepower, track grade and curvature, train weight, and certain constants related to rolling resistance. From this raw data, the computer determines: effective mass, locomotive tractive effort for any given speed, positive or negative gravity component resulting from grade, curve resistance, and rolling resistance.

The calculation program computes the values of acceleration for each train at intervals of six seconds, determines speed at the end of each interval from this, and then the actual train position from the average train speed for each six-second interval. As the position of the train changes, the computer continuously modifies the grade and curvature data to agree with the track section on which the train is located.

Before a computed speed is used for moving the train, however, it is checked against any speed limits which may exist. These may be a general limit, local limits imposed on a particular section of track, or a limit resulting from a signal restriction developed by the route-logic program. In any case, if the train speed is excessive, a braking factor is applied which brings the speed down at a normal braking rate.

The Route Logic Program

The route-logic program, unlike the calculation program, has little to do with formulas and arithmetic. In this part of the operation, the emphasis is on logical decision—using the ability of the computer to branch, and to choose one of two courses of action in accordance with conditions it finds.

The basic concept of the route-logic program may be simply stated. Let each train run as computed by the calculation program until a conflict occurs. Next, determine what action should have been taken to avoid the situation, then go back and recompute, this time taking the necessary steps to prevent conflict

Or, as it has been more vividly stated, "Let the trains go tearing down the track at 900 miles an hour until they crash into each other, then go back and put the train on the siding that should have been there in the first place." In some respects, at least, the computer has more freedom than a dispatcher has!

In the calculation program, the railroad is considered as made up of consecutive grade and curve sections. In the route-logic program, the railroad is considered as made up of consecutive signal blocks. Each signal block is represented by a storage location on the memory drum.

Of the ten digits in each signal block memory location, four digits represent the length of the block in tens-of-feet, four digits indicate the presence of a train in the block, and the remaining two digits indicate the type of signal section, such as inter-

*THE CALCULATION PROGRAM

Acceleration is equal to force divided by mass. For a train this may be written as:

$$\text{Acceleration} = \frac{\text{Locomotive Tractive Effort} \pm \text{Gravity Component} - \text{Curve Resistance} - \text{Rolling Resistance}}{\text{Mass}}$$

The rolling resistance is determined from the Davis formula:

$$R = 1.3 + \frac{29}{W} + kV + \frac{CAV^2}{Wn}$$

where: R = tractive resistance, pounds per ton; W = average weight per axle, tons; k = friction constant; V = speed, mph; C = streamlining factor; A = effective cross-sectional area, sq. ft; n = total number of axles.

The acceleration of the train will be:

$$a = \frac{F}{100W} - \frac{R}{100} - 0.2P - 0.08D$$

where: a = acceleration, mph per second; F = locomotive tractive force, pounds; 100W = effective mass, including inertia effect of rotating wheels; R = train resistance from Davis formula; P = percent grade of the track; D = degrees curvature of the track.

The force of the locomotive may be solved from the equation.

$$F = \frac{375 \times HP \times 0.8}{V^*}$$

where: F = available force of the locomotive, pounds; HP = rated horsepower of the locomotive; V* = train speed or minimum speed for maximum horsepower, whichever is larger.

The minimum speed for maximum horsepower is considered to be the speed at which the locomotive tractive effort is equal to 25 percent of the weight of the locomotive on its driving axles.

The computer calculates the acceleration available and if a change in velocity is indicated, it is solved from the equations:

$$V_2 = V_1 + at \qquad X = \frac{V_1 + V_2}{2} \times t$$

where: a = acceleration available at speed V_1; V_1 = the initial speed; t = a six second constant acceleration period; V_2 = the speed after the acceleration; X = the distance traveled.

No change in velocity is indicated either by zero acceleration available or when the speed is greater than a speed limit but not greater than the speed limit plus five miles per hour. The time to travel the next distance is then calculated by dividing the distance by the speed. Thus the time to travel from one portion of the railroad to another may be calculated for any train.

References: "Train Resistance of Freight Trains under Various Conditions of Loading and Speed," AREA Proceedings, vol. 42, 1941, pp. 69-86.

mediate, approach, mainline at siding, or other. The computations required on moving from one block to another vary with the kinds of blocks involved, and these last two digits are used to select the computing routine required.

Trains are identified by 4-digit numbers, called train markers. When a train is in a particular signal block, its marker is part of the 10-digit number stored at the corresponding address. As the train moves along the railroad, the marker advances from address to address accordingly.

Train Marker Locates Data

At first glance, it might appear logical to use the actual train number as the train marker. A more useful marker, however, is the four-digit number which identifies the first of a series of 10 drum storage sections in which the complete data for the train is stored. Thus any time the computer finds a train in a block, it knows immediately from the train marker where to locate any additional information it needs about the train. Conversely, among the train data is stored the four-digit number which identifies the signal section the train is in at any time.

The computer moves the trains through the signal blocks one at a time, with the sequence of moves determined by the clock time at which each train reaches certain key points. These key points are at the end of each block, just before the train passes the next signal, and at the beginning of each block, just before the rear end of the train leaves the block behind.

From the calculation program, the computer determines the clock time at which a train being computed arrives at its next key point. The computer then scans all the other trains and examines the times at which they arrived at their last key points. If one or more of these trains has an earlier time, the computer selects the earliest train and computes it to its next key point, after which it again checks the times of all trains before selecting the next to be advanced.

Included in the data for each train are a plus and minus sign which indicate direction, and "priority" number which serves as an indication of the relative importance of the train when judgments are made as to which must take the siding.

The following tabulation is typical of the logic programmed into the computer, and lists the sequence of decisions made when two trains moving in opposite directions on the main line approach within two blocks of each other. It is assumed that the first train is ready to enter a new signal block. Before calculating the train into the block, the computer asks these questions:

Question	Answer	Decision
1. Are both blocks ahead clear?	No	There is a train ahead.
2. Is the other train taking the siding?	No	A conflict exists.
3. Is the other train going in the same direction?	No	This is a meet, not an overtake.
4. Is the other train of lower priority?	If Yes	Other train must take delay. Set it back into the last available siding passed.
	If No	Other train is of equal or higher priority.
5. Is the other train of higher priority?	If Yes	This train must take delay. Set it back into last available siding passed.
	If No	Trains are of equal priority. Put train that last passed an available siding into that siding.

This chain of logical decision represents the general method of analysis used. Similar sequences, appropriate to each situation, can be followed for the various operating conditions that may occur.

The computer, of course, does not ask questions in the form shown. Instead, each question represents an arithmetical operation of some kind, and the answer "yes" or "no" is actually represented by a plus or minus sign, a zero or non-zero number, or the appearance of the digit 8 or 9 in a specific position in a number.

The basic program is designed to operate all trains so that they get over the railroad in the shortest time permitted by locomotive power and route conflicts. In actual operation, however, trains may be delayed at one or more places for various periods of time. A way freight will stop for local work. A passenger train must make specified station stops according to the passenger timetable, and has scheduled passing times at other points.

To provide for this element in the simulated operation, three sub-programs are included which can insert the following delay elements:

1. A stop for a specified length of time, such as for normal working time of a freight at a given location.
2. A stop for a specified minimum time, but the train may not leave before a scheduled time. This covers passenger train requirements at station stops.
3. An arbitrary delay which does not stop a train, but which holds it to a scheduled time at specified points. This adjusts the calculation for passenger trains with power enough to run faster than schedule, but which are throttled back by the engineman.

Each time a train passes a siding switch, the computer produces an OS report. This appears as a punched IBM card, called an "answer" card, which gives the train number, the time, the location, the speed at which the train was moving, whether it was entering or leaving the siding track, and other information. From these cards it is possible to plot a train chart.

Simulated Operation

Figure 1 shows a train chart prepared from computed data to show the simulated operation of several trains on a single track railroad. To illustrate some points, dashed lines have been included on this chart that would not be plotted normally.

As might be expected, the time-distance lines for the two passenger trains run right through, meeting each other at Frank. The eastbound passenger was put on the siding here, but the meet was apparently non-stop. This could be checked from the running speed shown on the eastbound's answer card for the OS circuit at the east end of Frank.

The time-distance lines for the scheduled freight is more revealing of how the computer went about the job. Note the dashed line for the eastbound scheduled freight at Baker.

This shows that as the computation progressed, this train was OS'd twice at the east end of Baker. The first time it was computed out of Baker, the computer found that it would delay the following passenger, so it was returned to the siding at Baker. The second OS shows the final solution of the problem for this train, with the freight leaving Baker behind the passenger. The meet with the westbound passenger occurred at Dixie, where the freight arrived in time to avoid delaying the passenger, and hence was not set back.

Three Attempts Before Success

Even more revealing of the computer's method of operation is the case of the extra east, which had to try three times to get out of Card before it succeeded.

The extra waited at Card for the eastbound passenger to pass, and then started for Dixie. Somewhere between Card and Dixie, the computer recognized that the lower priority extra would delay the eastbound scheduled freight, so the extra was set back on the siding at Card.

After the scheduled freight passed, the extra started again, and proceeded as far as the west OS section at Dixie. However, the westbound passenger was rapidly approaching the east end of Dixie at this time and would have had to take an unnecessary delay—so once again the extra was put back to the siding at Card, to leave finally after the passenger had passed.

This train chart was plotted manually from the answer cards. It will be recollected, however, that the labor involved in plotting train charts was one of the problems involved in their use.

Here again the computer has been put to work. A supplementary computer program has now been developed which analyzes the answer cards produced by the main program. From these cards the computer prepares another set of cards which can be run on a standard tabulating machine to obtain a machine plot of all the OS points produced by simulated operation. All duplicate OS's resulting from reruns are eliminated by this program. Each OS is represented by a figure 8 printed on the tabulation. The middle of the 8 is taken as the plot point.

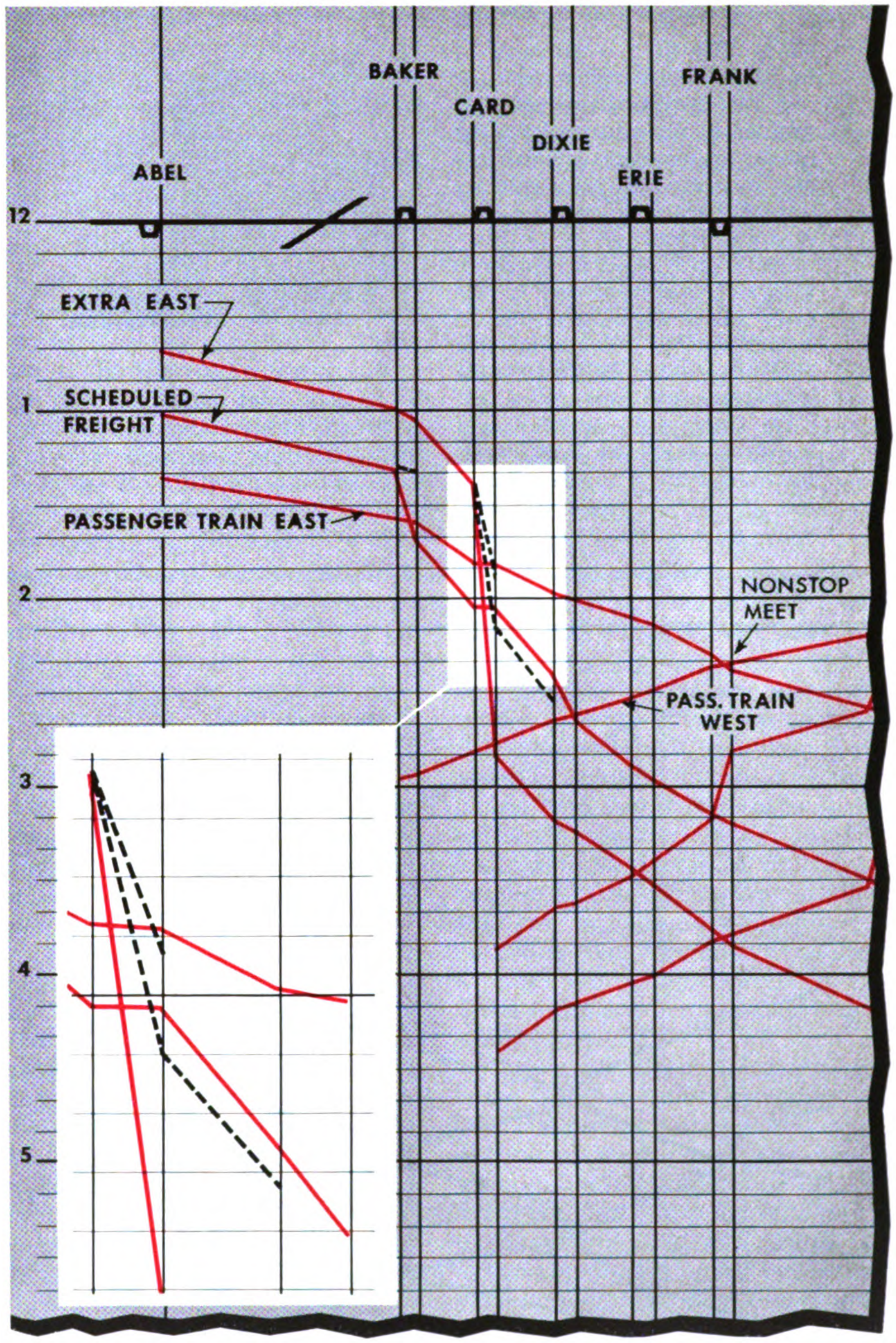


Figure 1

In this arrangement, the horizontal locations representing geographical positions on the railroad are established by selecting certain of the 60 tabulation columns available across the sheet. The vertical scale is a time scale, but not in the conventional sense. Each OS is recorded on a separate line, in time sequence, with time and number of the OS'ing train indicated on the tabulation in the margin.

With the OS points plotted mechanically, the drawing of the train chart is greatly expedited, since it is merely necessary to connect the "8's" printed by the tabulation. Because of the scale distortions involved, the slopes of the time-distance curves cannot be interpreted directly in terms of speed, as on a conventional chart. But the graphic representation of meets and overtakes is still clearly shown. It is anticipated that with refinement of the plotting program, and with more experience in the use of the mechanically plotted chart, such charts will serve for many purposes

Mr. O'Neill is a graduate of Columbia University, holding an M.S. in Industrial Engineering. In addition he has done work in advanced engineering mathematics and in radio and electronics at Cornell University and the University of Rochester.

Mr. O'Neill began his career in 1934 as a junior engineer for the Long Island Lighting Company, leaving there in 1939 to join the N. Y. Public Service Commission as a valuation engineer.

He joined GRS in 1943 as a service engineer on B-29 work, and was made assistant production engineer in 1945. Transferring to the advertising department, he became copy supervisor in 1953. In 1957 he was made supervisor, advertising and traffic analysis, and in September 1959 was appointed manager of the Electronic Control Division.

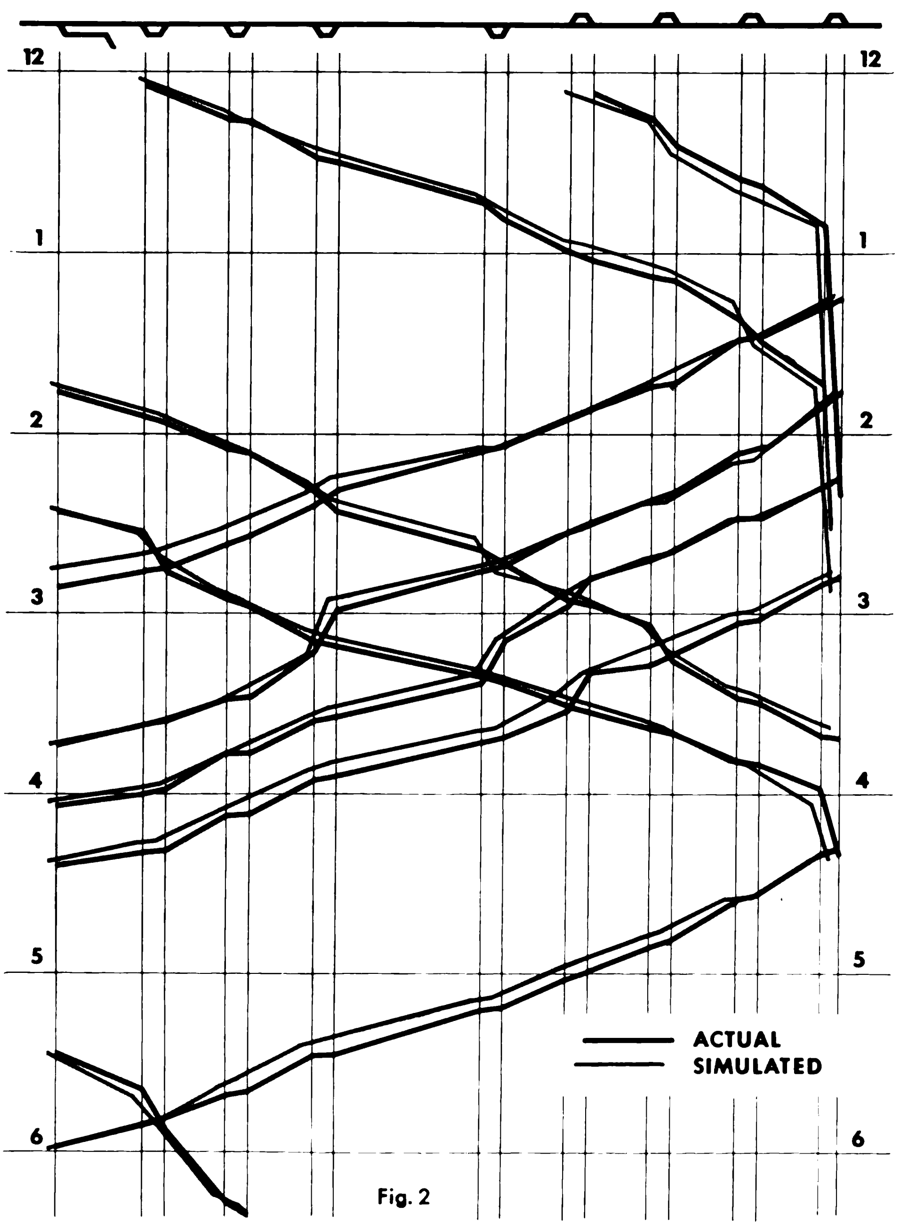


Fig. 2

as well as hand plotted ones, with the resultant elimination of much tedious work.

It appears, therefore, that a workable program has been developed for simulating single-track railroad operation by electronic digital computer. With the IBM 650 computer, this program can handle up to 10 trains at a time over a 100 to 150 mile section of a typical railroad. As each train's run is completed, another train may be added. Ten trains can be computed over the road in about one hour and forty minutes.

The computed information can be of considerable assistance in planning the track and signal layouts for CTC, and for evaluating the results that may be anticipated with a given layout if subsequent changes are made in grades, curves, schedules, or equipment.

New data for a complete change in track and signal layout may be prepared from blueprints in one-half day. Train data, such as power, departing time and so forth, may be changed in a few minutes.

How Accurate Is Simulation?

The question inevitably arises—how closely does simulated train operation approach observed reality?

It is not possible to give final answers, but Figure 2 is representative of results to date. On this train chart, the black lines show the actual train operation observed on a specific day on a specific railroad equipped with CTC. The red lines show the computer simulation of operation of the same trains, on the same day, on the same railroad. The agreement between observed fact and computed results, as demonstrated in the diagram, is encouragingly close.

This is only a beginning. Newer, much faster computers with much larger memories are becoming available. These will make it possible to refine and improve programs, and to compute more and more trains over longer and more complicated track layouts. As more experience is acquired in relating computers to the problems of traffic control, radically new concepts in automation may well appear. No one can accurately predict the future; but, if the past is any guide, these developments will be with us sooner than we think.