DIGITAL SIMULATION FOR CONTROL SYSTEM DESIGN

R. D. Brennan
International Business Machines Corporation
Los Gatos, California

ABSTRACT

Simulation of dynamic systems by use of analog computers is well established as a practical tool of the control engineer. With the advent of fast digital computers, it has become not only feasible but attractive to perform many such investigations using digital simulation. As an example, a study of a simple control system is presented to illustrate the ease and flexibility of this technique in design.

Digital simulation permits the engineer who has no specialized training in computer programming or operation to readily utilize simulation in both the design and analysis aspects of his work. It has proven a fast, effective, and economical tool for general investigations of continuous, dynamic systems and is of particular interest to the practicing control engineer.

DIGITAL SIMULATION FOR CONTROL SYSTEM DESIGN

R. D. Brennan
International Business Machines Corporation
Los Gatos, California

Simulation techniques have been used increasingly in recent years for the design and analysis of control systems. The analog computer has been used for innumerable such studies and has proved a convenient and flexible tool. However, the necessity of scaling problem variables into reasonable voltage levels, and the operational difficulties inherent in analog circuitry combine to present the analog user with a number of irksome problems. These difficulties mount as the size, complexity, and accuracy requirements of the problem increase. It is not coincidental, therefore, that considerable attention has been given in the last several years to the development of special digital languages for simulation of continuous systems.

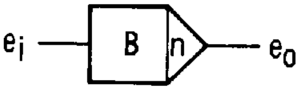
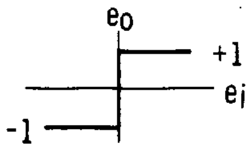
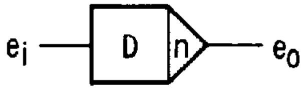
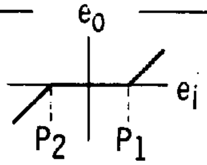
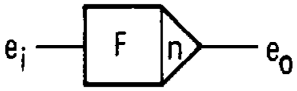
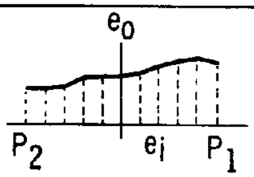
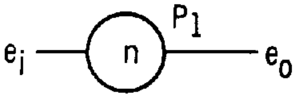
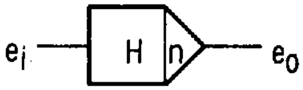
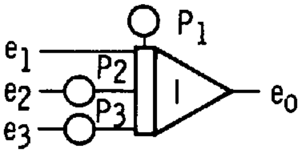
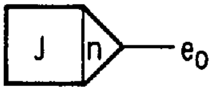
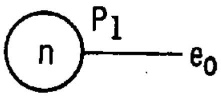
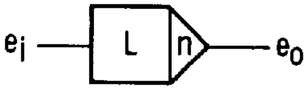
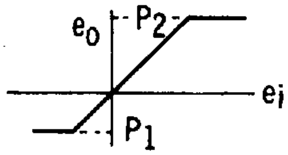
A major portion of this activity has been in development of the "digital analog simulators"; during the past ten years upwards of thirty separate programs of this type have been reported. These programs provide a complement of functional elements similar to those of the analog computer and a block-oriented language for specification of their interconnection. These "digital analog simulators" model the elements and organization of analog computers and provide numerical routines that are equivalent to such standard analog elements as integrators, summers, inverters, multipliers, and function generators.

These programs also provide those special-purpose devices commonly assembled from several analog elements, e.g., division, absolute value, square root, exponential, sine function, limiting, dead space, and time delay units. Just as the computer patchboard electrically links analog computing elements, the simulation language interconnects the numerical routines.

The 1130 Continuous System Modeling Program (1130 CSMP) is a new program of this general type specially developed for the environment of the design engineer. It employs a familiar block-oriented input language and offers an online, interactive mode of operation during development and testing of a simulation model. The simplicity of the language statements and the console procedures enables a user to rapidly gain proficiency with this program. Simplicity and flexibility are, in fact, its foremost characteristics.

1130 CSMP is an adaptation of the PACTOLUS program for the IBM 1620. The computing speed of the IBM 1130 makes it feasible to handle more complex processes, so that, for many types of investigations, 1130 CSMP obviates any requirement for an analog computer facility. Indeed, both the design and implementation of a simulation study are considerably simpler with this system than with an analog computer.

1130 CSMP provides a complement of 25 standard simulation elements, plus a group of "Special" elements which the user can tailor to his particular needs. Table 1 illustrates a representative group of 1130 CSMP elements, their diagrammatic and language symbols, and definitions of their functional operation. The user starts by developing a block diagram showing the interconnections of the elements required to implement his model. He then translates the diagram into a corresponding set of 1130 CSMP language statements.

ELEMENT TYPE	LANGUAGE SYMBOL	DIAGRAMMATIC SYMBOL	DESCRIPTION
BANG-BANG	B		
DEAD SPACE	D		
FUNCTION GENERATOR	F		
GAIN	G		$e_0 = P_1 e_i$
HALF POWER	H		$e_0 = \sqrt{e_i}$ SQUARE ROOT
INTEGRATOR	I		$e_0 = P_1 + \int (e_1 + e_2 P_2 + e_3 P_3)\, dt$
JITTER	J		RANDOM NUMBER GENERATOR BETWEEN ±1
CONSTANT	K		$e_0 = P_1$
LIMITER	L		

n REPRESENTS THE BLOCK NUMBER

Table 1 Representative Group of 1130 CSMP Elements

A most important feature is the option of entering these statements either via punched cards or directly from the console keyboard. During the introduction of a problem in the online mode, the user is provided with automatically typed instructions which guide him through the procedures. During a run, the user has the ability to interact with the model as directly and spontaneously as he would with an analog computer.

The simplicity and flexibility of 1130 CSMP is illustrated by its use for a control system study. This investigation was performed several years ago using an analog computer facility.[1] The experiments were repeated using 1130 CSMP to demonstrate the superiority of the digital approach.

The study is concerned with design of a simple adaptive system for control of a process characterized by the transfer function

$$G(s) = \frac{10s + 100}{s^2}$$

Figure 1 illustrates the proposed design for the control system. The process can readily be controlled if the operating environment permits a sampling period, T, which is sufficiently short to ensure satisfactory performance for all anticipated conditions. This results in an inefficient design in the sense that the sample-and-hold operation must be performed continually, even during periods of essentially steady-state behavior. The controller may very well be a digital computer which is being time shared among a number of tasks; if the sampling operation is unnecessary, the computer time can be used more profitably attending to these other tasks.

[1] R. C. Dorf, M. C. Farren, and C. A. Phillips, "Adaptive Sampling Frequency for Sampled-Data Control Systems," IRE Transactions on Automatic Control, Vol. AC-7, No. 1, January 1962, pp. 38-47.

The purpose of the simulation was to determine experimentally whether satisfactory control could be achieved using a variable sampling period determined as some simple function of the error signal. One particularly simple scheme is to provide two sampling periods--a "short" period for fast sampling, and a "long" period for slow sampling. This bi-frequency sampling scheme would be under control of the error signal. "Fast" sampling would be used when the error signal indicates rapid changes in the process; "slow" sampling would be used as the response of the process approaches steady-state behavior.

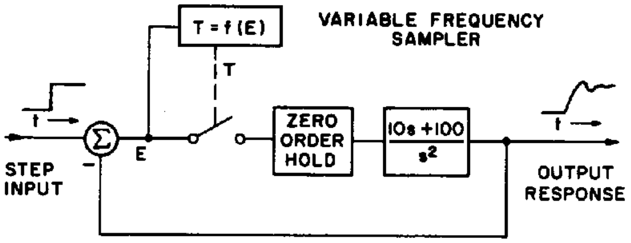


Figure 1 Sampled-data control system example

The 1130 CSMP block diagram for this design problem is shown in Figure 2. Most of the block elements are immediately identifiable to the analog computer user. Block 3 performs the sample-and-hold operation whenever supplied with a positive trigger pulse.

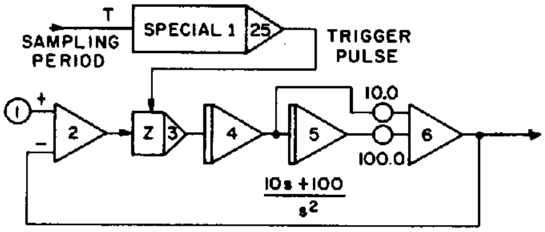
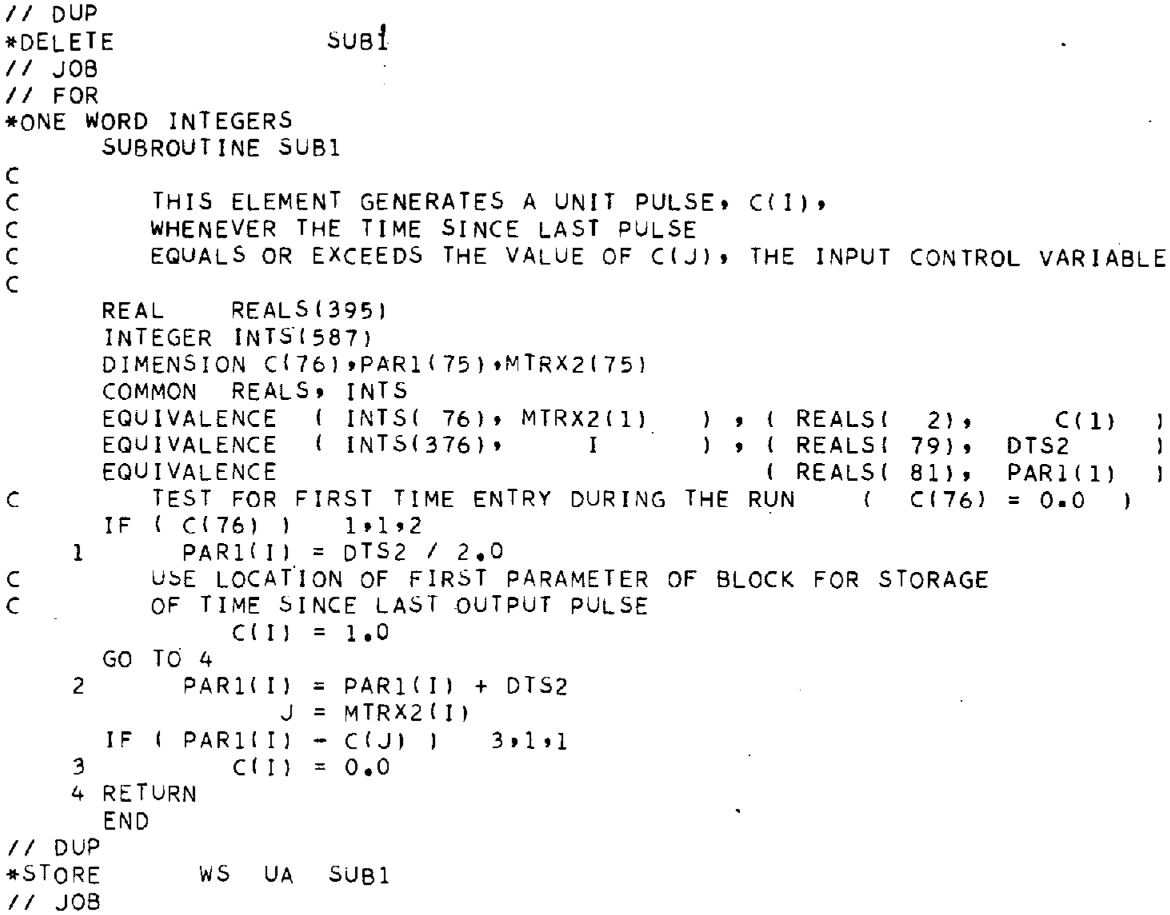


Figure 2 1130 CSMP block diagram for sampled-data control system example

Block 25 supplies the trigger pulses and is a "Special," or "do-it-yourself," type of element. The variable-frequency pulse train used in the study is not available from the standard complement of the 1130 CSMP elements. This special operation is simply defined by the preparation of a simple FORTRAN subroutine as shown in Figure 3.

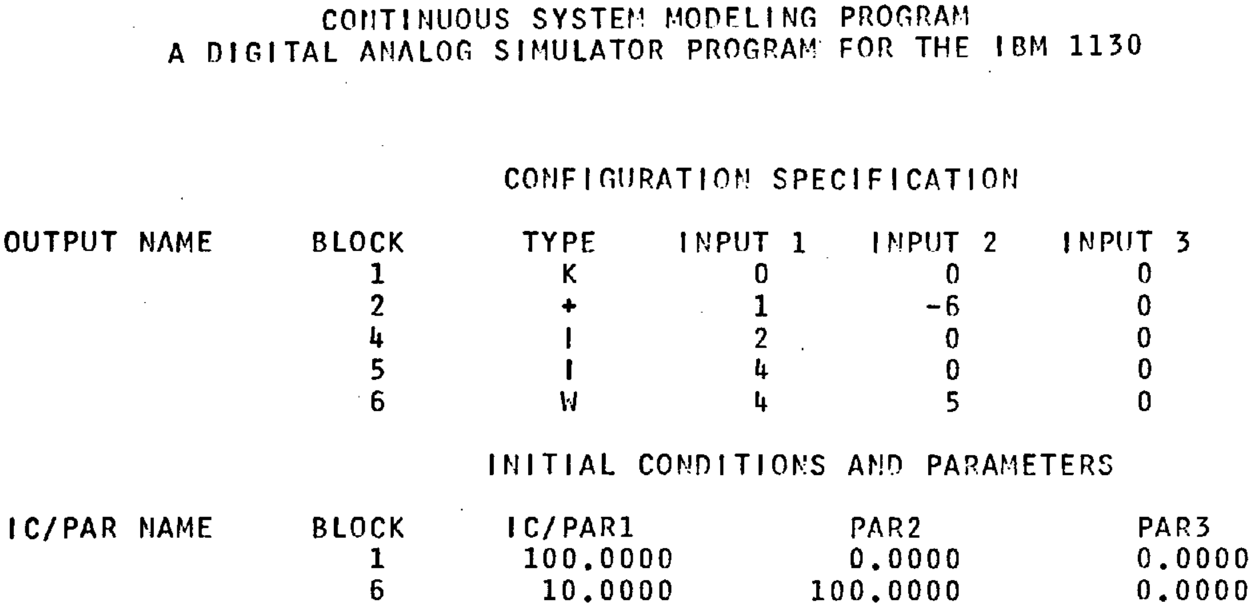


```
// DUP
*DELETE             SUB1
// JOB
// FOR
*ONE WORD INTEGERS
      SUBROUTINE SUB1
C
C        THIS ELEMENT GENERATES A UNIT PULSE, C(I),
C        WHENEVER THE TIME SINCE LAST PULSE
C        EQUALS OR EXCEEDS THE VALUE OF C(J), THE INPUT CONTROL VARIABLE
C
      REAL    REALS(395)
      INTEGER INTS(587)
      DIMENSION C(76),PAR1(75),MTRX2(75)
      COMMON  REALS, INTS
      EQUIVALENCE  ( INTS( 76), MTRX2(1)   ) , ( REALS(  2),     C(1)  )
      EQUIVALENCE  ( INTS(376),      I      ) , ( REALS( 79),  DTS2     )
      EQUIVALENCE                               ( REALS( 81),  PAR1(1)  )
C        TEST FOR FIRST TIME ENTRY DURING THE RUN    (  C(76) = 0.0  )
      IF ( C(76) )    1,1,2
    1      PAR1(I) = DTS2 / 2.0
C        USE LOCATION OF FIRST PARAMETER OF BLOCK FOR STORAGE
C        OF TIME SINCE LAST OUTPUT PULSE
              C(I) = 1.0
      GO TO 4
    2      PAR1(I) = PAR1(I) + DTS2
                 J = MTRX2(I)
      IF ( PAR1(I) - C(J) )   3,1,1
    3         C(I) = 0.0
    4 RETURN
      END
// DUP
*STORE      WS  UA  SUB1
// JOB
```

Figure 3 FORTRAN program defining "Special Element No. 1" as the variable-period time pulse generator

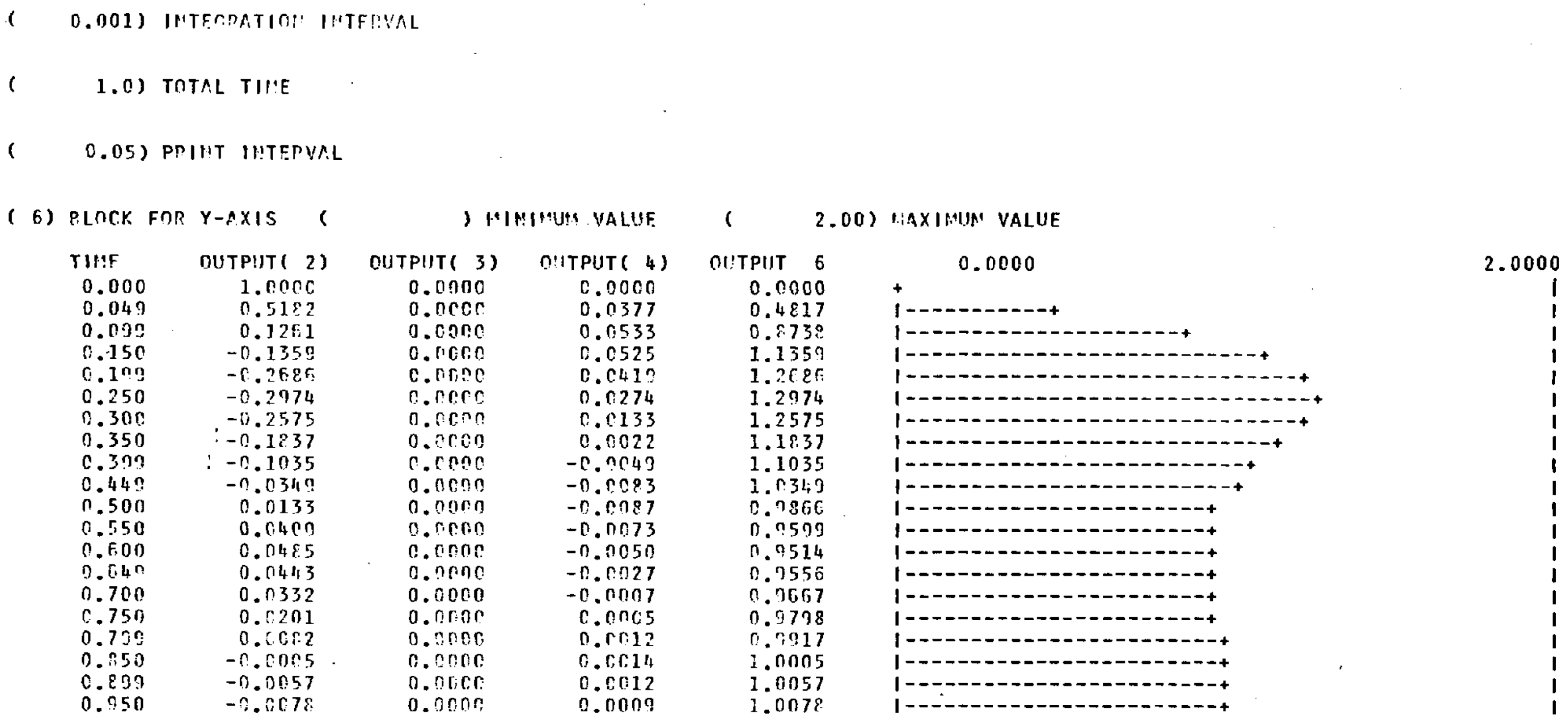
When the elapsed time from the last output pulse reaches a value equal to the current value of the input variable, T, the "Special Element #1" produces another trigger pulse for the sample-and-hold operation. This ability to simply define additional complex logical or nonlinear operations is a significant advantage of digital simulation.

The first step in the simulation was translation of the block diagram into a corresponding set of 1130 CSMP language statements. For the initial run, Block 4 was directly connected to Block 2 (which simulates the summing junction shown in Figure 1); this run thereby simulates the performance of the process assuming continuous rather than sampled-data control. Figure 4 shows the first portion of the record from the console printer and illustrates the Configuration and Parameter Statements needed to translate the block diagram for use with 1130 CSMP. Figure 5 shows the "print-plot" record of the response of the simulated process to a step input. This type of output is provided for those users who don't have the 1627 Plotter.



```
                  CONTINUOUS SYSTEM MODELING PROGRAM
              A DIGITAL ANALOG SIMULATOR PROGRAM FOR THE IBM 1130


                              CONFIGURATION SPECIFICATION

OUTPUT NAME       BLOCK       TYPE      INPUT 1     INPUT 2     INPUT 3
                    1          K           0           0           0
                    2          +           1          -6           0
                    4          I           2           0           0
                    5          I           4           0           0
                    6          W           4           5           0

                             INITIAL CONDITIONS AND PARAMETERS

IC/PAR NAME       BLOCK      IC/PAR1              PAR2               PAR3
                    1        100.0000            0.0000             0.0000
                    6         10.0000          100.0000             0.0000
```

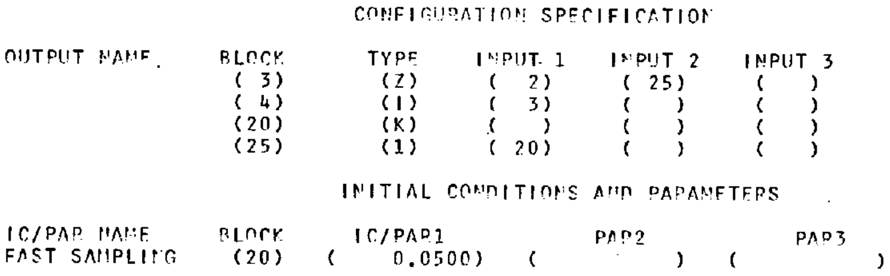
Figure 4 A portion of the console printer output



```
(    0.001) INTEGRATION INTERVAL

(      1.0) TOTAL TIME

(     0.05) PRINT INTERVAL

( 6) BLOCK FOR Y-AXIS   (          ) MINIMUM VALUE      (     2.00) MAXIMUM VALUE
     TIME      OUTPUT( 2)   OUTPUT( 3)   OUTPUT( 4)   OUTPUT  6          0.0000                                   2.0000
      0.000       1.0000       0.0000       0.0000       0.0000      +                                                 |
      0.049       0.5182       0.0000       0.0377       0.4817      |-----------+                                     |
      0.099       0.1261       0.0000       0.0533       0.8738      |---------------------+                           |
      0.150      -0.1359       0.0000       0.0525       1.1359      |---------------------------+                     |
      0.199      -0.2686       0.0000       0.0419       1.2686      |------------------------------+                  |
      0.250      -0.2974       0.0000       0.0274       1.2974      |-------------------------------+                 |
      0.300      -0.2575       0.0000       0.0133       1.2575      |------------------------------+                  |
      0.350      -0.1837       0.0000       0.0022       1.1837      |----------------------------+                    |
      0.399      -0.1035       0.0000      -0.0049       1.1035      |--------------------------+                      |
      0.449      -0.0349       0.0000      -0.0083       1.0349      |-------------------------+                       |
      0.500       0.0133       0.0000      -0.0087       0.9866      |-----------------------+                         |
      0.550       0.0400       0.0000      -0.0073       0.9599      |-----------------------+                         |
      0.600       0.0485       0.0000      -0.0050       0.9514      |-----------------------+                         |
      0.649       0.0443       0.0000      -0.0027       0.9556      |-----------------------+                         |
      0.700       0.0332       0.0000      -0.0007       0.9667      |-----------------------+                         |
      0.750       0.0201       0.0000       0.0005       0.9798      |-----------------------+                         |
      0.799       0.0082       0.0000       0.0012       0.9917      |------------------------+                        |
      0.850      -0.0005       0.0000       0.0014       1.0005      |------------------------+                        |
      0.899      -0.0057       0.0000       0.0012       1.0057      |------------------------+                        |
      0.950      -0.0078       0.0000       0.0009       1.0078      |------------------------+                        |
```

Figure 5 Console printer output showing "print plotter" format

For the second run, the simulation was modified to represent the sampled-data control. These modifications were entered online via the console keyboard. New specifications were simply typed within the parentheses provided by the program; this operation is equivalent to, but simpler than console patching at the analog computer. Figure 6 shows how the variable sampling elements were added. Blocks 20, 25, and 3 were added to the initial configuration and the input to Block 4 was changed from the summing point to the output of Block 3--the sample-and-hold device. The second and third runs were performed using the "fast" and the "slow" sampling respectively. Note that the sampling period, T, was obtained from a Constant element (Block 20). Its value was set at 0.05 for the second run, then changed to 0.1 for the third run. These values correspond to the fastest and slowest sampling found advisable for this process. The 1627 Plotter record of these first three runs is shown in Figure 7.



```
                             CONFIGURATION SPECIFICATION

OUTPUT NAME.      BLOCK        TYPE      INPUT 1     INPUT 2     INPUT 3
                  ( 3)         (Z)       (   2)      ( 25)       (    )
                  ( 4)         (I)       (   3)      (    )      (    )
                  (20)         (K)       (    )      (    )      (    )
                  (25)         (1)       (  20)      (    )      (    )

                            INITIAL CONDITIONS AND PARAMETERS

IC/PAR NAME       BLOCK        IC/PAR1               PAR2                  PAR3
FAST SAMPLING     (20)    (      0.0500)    (              )    (              )
```

Figure 6 Console printer output showing online modification

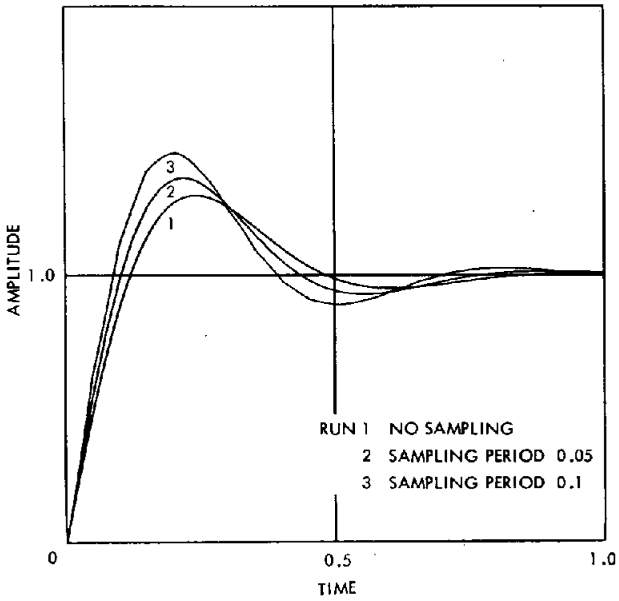


Figure 7

Plotter record for runs 1 - 3

Finally, Block 20--the Constant element supplying parameter T--is replaced by the configuration shown in Figure 8. This provides the bi-frequency sampling of the original control study. The fastest sampling rate is used whenever the absolute value of the error derivative exceeds the threshold determined by Block 15. When the system approaches steady state, the slower sampling rate is used. A plotter record for three values of the threshold is shown in Figure 9. This experimentation is performed simply and the control engineer, on the basis of his evaluation of the system performance, quickly determines an acceptable threshold.

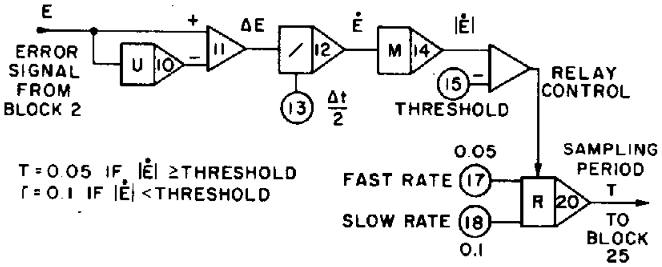


Figure 8 Bi-frequency sampling period simulation

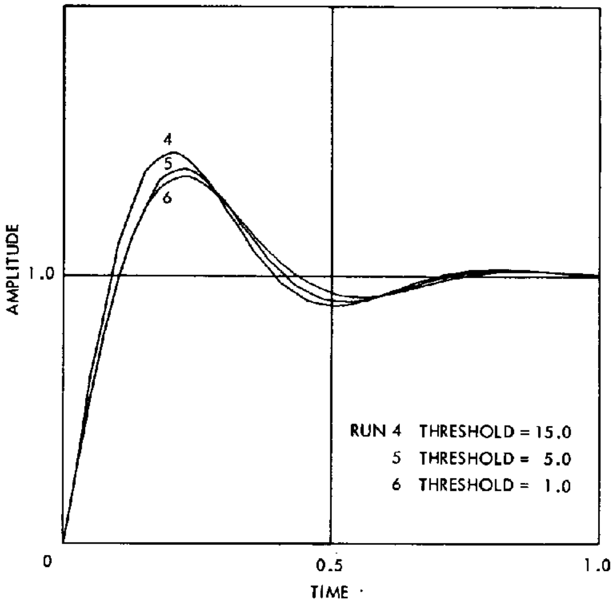


Figure 9 Plotter record for bi-frequency sampling runs

Although this example is rather simple, nevertheless it illustrates how the control system designer can effectively use digital simulation techniques. The task of preparing a problem for simulation and study is comparatively simple. So too, is the task of modifying the simulation for analysis of different design approaches. The modification of both configuration and parameters is readily accomplished online by means of the console features and the options programmed into the 1130 CSMP.