RESEARCH OF DIRECT DIGITAL CONTROL LOOPS BY HYBRID SIMULATION

By

A. FRIGYES and K. TIHANYI

Department of Process Control, Technical University, Budapest

Received July 25, 1976

1. Introduction

In synthetizing control loops, various simulation procedures play an important role. Both analog and digital simulation are usual. Some part problems of simulation are preferably solved in the analog way, and others by digital methods. Therefore, hybrid computers appear to be suitable tools to solve the over-all simulation of a control loop. Due to their architecture, it is possible to distribute the functions of simulating data acquisition, process identification, and direct digital control in accordance with reality. For instance, the controlled plant can be simulated by the hybrid-analog patchboard, and the digital computer takes over the function of the process control computer by hybrid elements (A/D, D/A converters, control transfer unit).

The simulation of control loops has been studied on a hybrid computation system 1001 TPA-i/AC—04 developed by the Department of Instrumentation and Measurement of the Technical University of Budapest. The analog part of this computer operates in the repetitive mode.

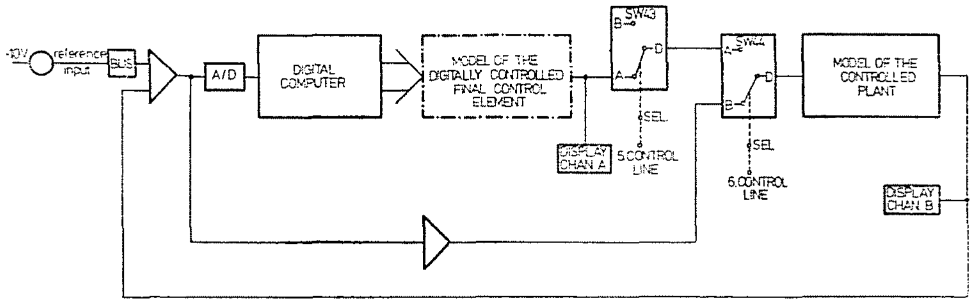
A "stand alone" (operating without executive) program package has been developed. An important part of this program — the simulation of the direct digital control (DDC) algorithm — is presented in this paper (Fig. 1). Other functions, such as process identification, control parameter setting etc., will be discussed in following papers.

2. The configuration of the hybrid computation system used

The following configuration was used:

1001 TPA-i CPU with a 4K words operative memory,
ME-x memory extension unit for storage extension up to 8K words,
console typewriter,
high-speed tape reader,
AC—04 hybrid-analog computer,
AC—04 hybrid-analog interface.

The 1001 TPA-i is a digital minicomputer built up from integrated circuits. Its software is compatible with the PDP—8 system. The word length is 12 bits and the number representation is of binary complement. It has a page configuration of memory, its cycle time is 1.5 μsec. 2×128 words can be addressed directly. The extending fields can be accessed by instruction for field selection.



The emitted logical levels to the control lines		Outputs of the controlled gain amplifiers		The positions (states) of the controlled loop
5	6	SW43	SW44	
0	0	A	A	closed, controlled
X	1	X	B	closed, without controller
1	0	B	A	opened, without controller

Fig. 1. Model of the direct digital control loop

The AC—04 hybrid-analog computer, too, is built up from integrated circuits. It works in the repetitive mode of operation. The time base can be varied from 10 msec to 100 msec by steps of 10 msec. The machine unit is $\pm$ 10 V. The program package uses the following internal hybrid units:

switching amplifiers,

multiplexers, controlled gain amplifiers driven by counters settable by a digital program,

flip-flops, inverters, etc.

The external hybrid units are the following:

A/D and D/A converter system. It converts the analog voltage lying between -10 V and $+10$ V, into real binary numbers of 11 bits. The accuracy of conversion is 1/2 LSB and the speed of A/D and D/A conversion is 60 μsec and 10 μsec, respectively.

Control transfer unit. By its sense lines the most important logical signals of the hybrid-analog part as well as functions, like the beginning or end of the computing cycle can be tested with the help of program steps. The control lines of the control transfer unit serve for handling the hybrid-analog computation. This is done by the control signals of two values (constant signal levels or pulse train).

Programmable digital clock. It measures the analog time or transformed one (i.e. divided by program), and it also makes possible to test the flags swinging in the programmed moments.

Analog-digital display. It displays maximum two addressed outputs.

3. The program realizing digital control

The simulated plant is realized on the hybrid-analog patch-board by plugging. The program package identifies the dynamics of the analog model by sampling its step response in equidistant moments. The identification procedure includes a parameter setting algorithm, too, to calculate the constants of a PID controller.

The parameters of the identified model and of the controller are determined by means of precalculated functions. The control of the model by program can be done in two different ways.

1. An analog PID controller is built up whose parameters are set digitally.

2. No analog controller is used, the PID controller's signals are produced directly by digital computation and they actuate the analog model of the plant. This second method is realized in two ways.

a) The digital actuating signals are switched to the D/A converter, and its analog output actuates the analog part (Fig. 2/a).

b) No D/A converter is used. The computer produces an impulse train which actuates directly the counters of the two controlled gain amplifiers (Fig. 2/b).

Thus it is possible to compare the different actuating modes which, nevertheless, realize the same control algorithm. The first method is not discussed in this paper.

The program is written in the SLANG 1 assembly language, the size of the program is somewhat less than 2K words (without the text and data store and the library routines).

The main parts of the program realizing the DDC are:

Initiating. The hybrid-analog unit and the peripherals are set in the normal position and the input data are read. The latter are the constants of the controller (gain, integration time and derivative action time) and the sampling period of the signals. Depending on the state of the switch register it is possible for the program to execute the identification as well as the algorithm setting the controller. In this case the number of input data are, of course, much greater.

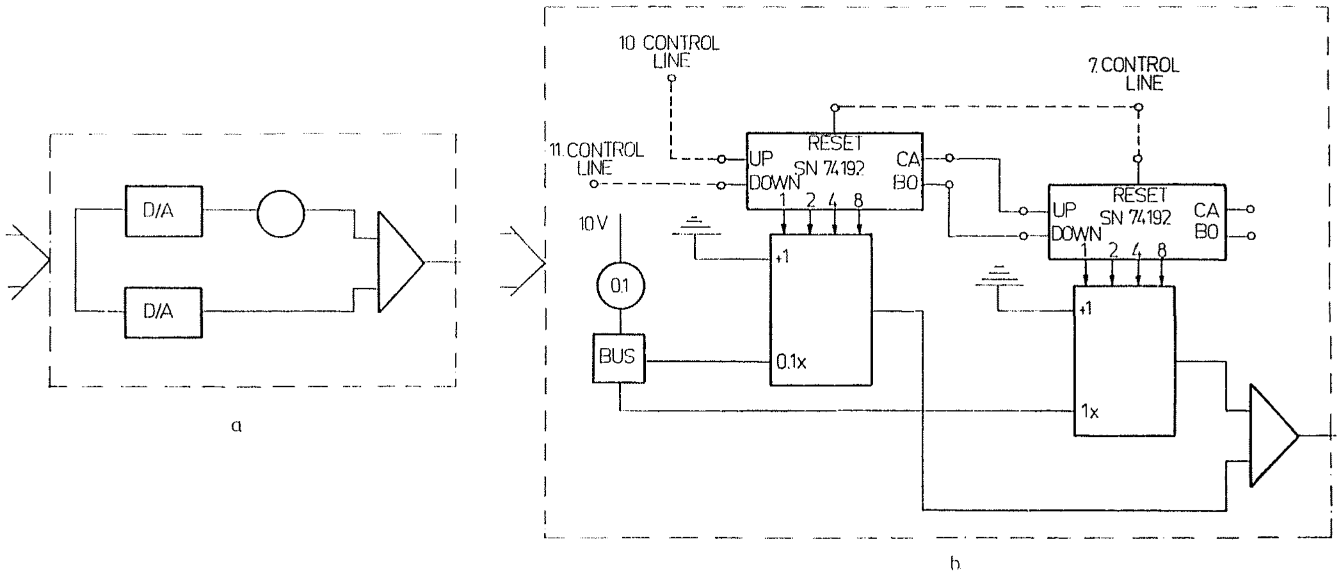


Fig. 2. **Methods of the realization of actuating (the models of the digitally controlled final control elements) a) by D/A converters; b) by controlled gain amplifiers**

Sampling. It samples the error signal, then the sampled data will be stored and converted into appropriate (floating-point) form.

Determining the actuating signal. This program part computes the actuating signal according to the PID algorithm. If the *n-th* sample of the actuating signal is denoted by x_n and that of the error signal by ε_n, then:

$$x_n = A\left(\varepsilon_n + \frac{T}{T_I}\sum_{i=0}^{n}\varepsilon_i + \frac{T_D}{T}(\varepsilon_n - \varepsilon_{n-1})\right) \tag{1}$$

Due to the modular architecture of the program, the above control algorithm can be replaced by any other if it is needed.

Executing the actuating procedure. The floating-point value of the actuating signal is converted into a form necessary to drive the final control element, then the program executes the final control action.

Displaying. It controls the printing and the hybrid displaying.

4. The cycle for sampling, computing and actuating

The organization of the cycle is determined by the repetitive operation of the hybrid-analog computer as well as by the realizable range of the time constants (a few milliseconds). Consequently, the values of the sampling interval worth using lie in the range of 100 μsec. But this interval is not sufficient to carry out the computation needed between two subsequent sampling points.

The actuating process synchronizes itself to the computing cycles of the hybrid-analog machine. One cycle of the repetitive operation contains 100 to 1000 sampling points. The transient of the process is built up and visualized in a way that the length of the portion of the transient being modelled, increases by one sampling interval in each subsequent cycle of the repetitive operation. So the time available to compute a single value of the actuating signal is approximately the cycle period of the repetitive operation. If this period is not long enough to get the result, then the computation will go on in the next, and if necessary, in further cycles (Fig. 3). In these cycles the transient does not continue to build up.

The sampled values of the actuating signal calculated step by step in the above-mentioned manner are stored. If at the beginning of a single cycle no computation is going on, then the single sampled values of the actuating signal calculated previously are given out at the proper sampling points. When all precalculated and stored sampled values have been given out, the control loop will be opened to avoid saturation of the analog integrators and the contact between the hybrid-analog and digital parts will be interrupted. Before doing so, the last sample of the error signal has to be measured, which is necessary to compute the next sampled value of the actuating signal. The calculation of the transient is finished (Fig. 4) when all the sampled values of the repetitive cycle are computed. The display of the transient can be kept for any duration. On the display the digitally controlled variable and the stepwise actuating signal appear. Figure 5/a shows an intermediate

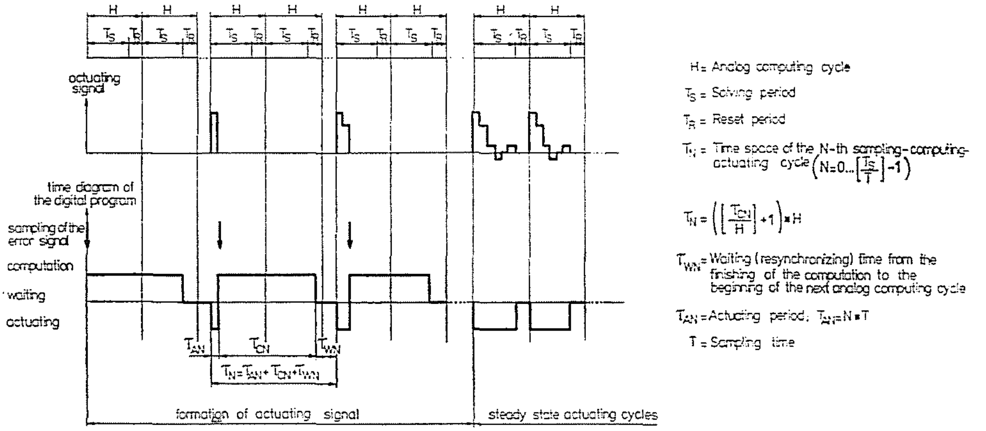
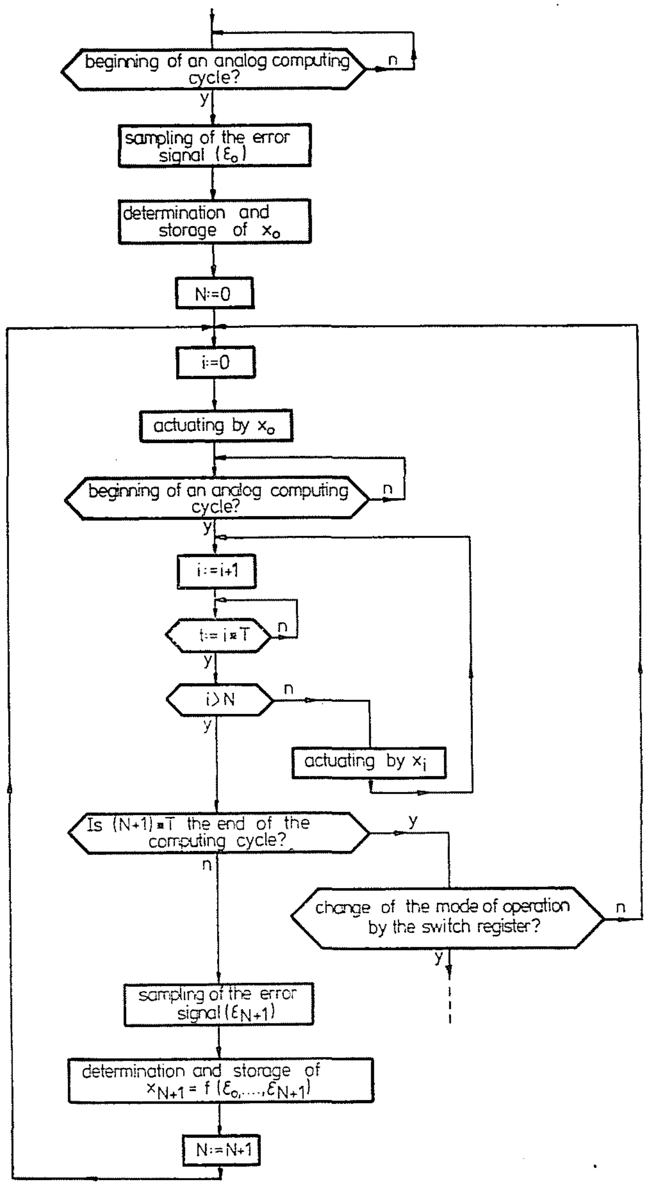


Fig. 3. Synchronization of the digital program to the computing cycles of hybrid-analog computer

phase and 5/b the whole transient already calculated. It is convenient to have the possibility of changing the structure of the analog model. Such changes as opening or closing the control loop and connecting or disconnecting the controller can be done by the manually setting the state of the switch register. At some instant the program senses the state of the switch register and, according to the state, controls the switching amplifiers of the hybrid-analog computer.

In this way the sampling interval can be varied in a wide range and the time needed for computing can be measured exactly. (Figs 6/a and b demonstrate the effect of changing the sampling period.) In both Figures the same controller parameters are used. It can be seen that the unstable system can be stabilized by decreasing the sampling period. The values of the measured error signals and computed actuating signals can be printed



T=Sampling time

t=Real time

ε_i= i-th value of the error signal at the moment $t=i*T$

x_i= i-th value of the actuating signal at the moment $t=i*T$

N= Number of the actuating values determined up to any moment

Fig. 4. **The cycle for sampling, computing and actuating**

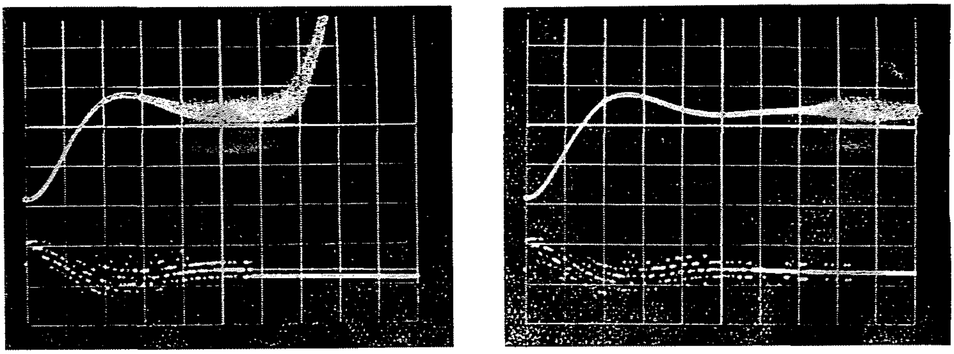


Fig. 5. The actuating cycle a) an intermediate situation; b) all transients already calculated

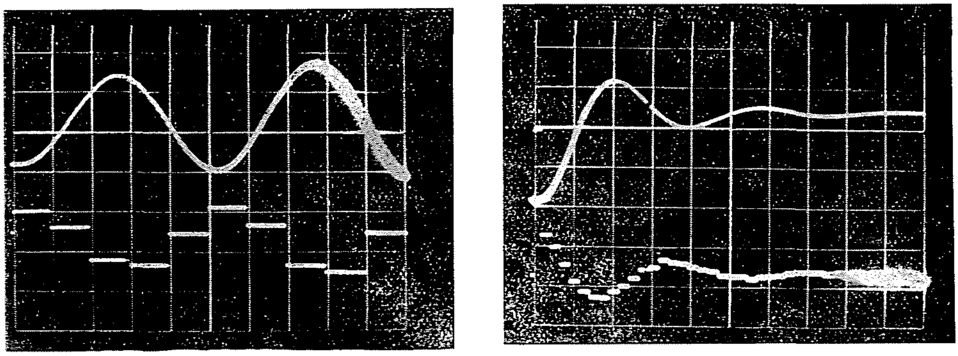


Fig. 6. The effect of changing the sampled period. The transient in the case of a) a longer; b) a shorter sampling period

out. This is a good help in debugging the program and makes the numerical evaluation of the results possible. Printed results are presented in Figure 7.

5. Computation of the actuating signal and its realization by the hybrid-analog computer

The actuating signal is produced in different ways depending on whether a D/A converter or a controlled gain amplifier is used. Therefore, their hybrid peripheral handlers are quite different and the subroutines which compute the actuating signal are only partly similar.

If a D/A converter is used, the digital program computes the actuating signal to Eq. 1, converts it and starts the process of digital-to-analog conversion. To ensure that the converted signals may be exact enough even if the signal levels vary within a wide range, a special network is applied (Fig. 2/a).

MANUAL SETTING OF CONTROLLER
PROPORTIONAL FACTOR =.6
INTEGRATION TIME =18
DERIVATIVE TIME-CONSTANT = 2
SAMPLING PERIOD OF DDC =.8

TIME	ERROR SIGNAL	ACTUATING SIGNAL
+ .00	+ .099	+ .211
+ .80	+.099	+.065
+ 1.60	+.093	+.055
+ 2.40	+0.81	+0.39
+ 3.19	+.060	+.017
+ 3.99	+.037	—.000
+ 4.79	+.013	—.014
+ 5.59	—.007	—.024
+ 6.39	—.024	—.027
+ 7.19	—.035	—.026
+ 7.99	—.039	—.019
+ 8.79	—.038	—.012
+ 9.59	—.031	—.000
+10.39	—.024	+.003
+11.19	—.012	+.017
+11.99	+.000	+.026
+12.79	+.009	+.028
+13.59	+.016	+.028
+14.39	+.023	+.032
+15.19	+.024	+.025
+15.99	+.024	+.024
+16.79	+.022	+.021
+17.59	+.017	+.014
+18.39	+.011	+.009

Fig. 7. Printed results of the program realizing direct digital control

If a controlled gain amplifier is used (Fig. 2/b), its counter summarizes the number of the pulses and thus the actuating signal is computed by the PID velocity algorithm instead of the position algorithm. The resolving power of this method is smaller (the number of the emittable pulses lies between 0 and 99), and the actuating signal is limited to one side. Similar signals, for instance, input signals of control valves often occur in process control practice. If the organization of the actuating routine were cyclical, the impulses could not be generated quickly enough. In this case the signal would not change step-wise at the sampling instants, but it would be extended over a relatively long period. Therefore, the routine is built up in a sequential way (without cycles). There is a part of the routine called "pulse transmitter table", which consists of instructions emitting single impulses. A given number of impulses is produced by the program jumping on a distinct element of the table. The change of the actuating signal between two sampling points is relatively small, therefore, the signal will be approximately step-wise. It is important to note that at the first sampling the change of the signal is relatively large (10—30 impulses), therefore the first value of the actuating signal will be emitted already during the reset period.

An important requirement to the program was to keep its space requirement minimal, considering the small storage space of the computer being used. A contradictory point of view is the need of saving time in the pseudo real-time operations and of the modular architecture of the program.

Summary

In connection with digital controls there are numerous problems which can be examined exactly with great difficulties only or not all (optimal selection of the sampling time, wind-up features, effects of complicated algorithms, etc.). The study discusses a possibility of the simulation of digital controls on a hybrid computation system with repetitive operation, in which various transient processes can be displayed and which offer a possibility of various solutions of correction approaching the reality. The article discusses two methods. In the first case the correcting variable is produced by the position algorithm, in the second case by the velocity algorithm. In the latter case the model simulates a final control element of integrating type.

References

1. KOCHENBURGER, R. J.: Computer Simulation of Dynamic Systems. Prentice-Hall, 1972.
2. KORN, G. A.—KORN, T. M.: Electronic Analog and Hybrid Computers. McGraw-Hill, 1972.
3. 1001 TPA-i Computer Manual I—II. KFKI.
4. The AC—04 Type Repetitive Analog Computer (Manual in Hungarian). Department of Instrumentation and Measurement, Technical University of Budapest, 1974.
5. The Hybrid Computing System Developed by the Department of Instrumentation and Measurement (Information Bulletin).
6. BÁNSÁGI, L.—DR. TÓTH, E.: A Hybrid Computing System (in Hungarian) Mérés és Automatika. 1974. No. 8.
7. TIHANYI, K.—BENYÓ, Z.—SKVOR, V.: A Short Description of Some Hybrid Systems (MEDA 41 TC, TPA-i/AC—04, ADT 7000, PACER). (In Hungarian) Mérés és Automatika, 1975. No. 10.

Prof. Dr. Andor FRIGYES } H-1521 Budapest
Katalin TIHANYI