SEL-67-085

DIGITAL FILTER DESIGN TECHNIQUE AND THE REALIZATION OF TRANSFER AND IMMITTANCE FUNCTIONS BY USING DIGITAL ELEMENTS

by

Chia-Peir Yu

November 1967

Scientific Report No. 22

Prepared under
National Aeronautics and Space Administration
Grant NsG-377

Radioscience Laboratory
Stanford Electronics Laboratories
Stanford University Stanford, California

PRECEDING PAGE BLANK NOT FILMED.

DIGITAL FILTER DESIGN TECHNIQUE AND THE REALIZATION OF TRANSFER AND IMMITTANCE FUNCTIONS BY USING DIGITAL ELEMENTS

By Chia-Peir Yu

ABSTRACT

Realization of two kinds of network functions by using digital elements are presented; these are (1) transfer functions, including the design of digital filters and spectrum analyzers where the real time implementation is made possible by using high speed, small size, integrated digital building blocks. Methods of digital integration are discussed whereby a configuration of the digital integrator is proposed as the basic digital element for real time device implementation. (2) Driving point impedance and admittance functions are also realizable by using digital elements provided that analog-to-digital and digital-to-analog converters are available. If both converters can be made available in integrated circuit form, then any analog components, either a single element or a combination of single elements, existing in the state of the art can be realized on a chip with digital elements. In this paper, all the synthesizing works are carried out by using Laplace transformation, except in the section where the time varying coefficients transfer function is discussed.

PRECEDING PAGE BLANK NOT FILMED.

CONTENTS

PRECEDING PAGE BLANK NOT FILMED.

TABLES

PRECEDING PAGE BLANK NOT FILMED.

ILLUSTRATIONS

ILLUSTRATIONS (Cont)

ILLUSTRATIONS (Cont)

PRECEDING PAGE BLANK NOT FILMED.

ACKNOWLEDGMENT

The author wishes to express his appreciation and indebtedness to Professor Allen M. Peterson, who supervised this study, and whose invaluable assistance and guidance made the completion of this report possible.

Chapter I
INTRODUCTION

Many reports in the area of designing the real time digital filters have recently been published [4,5,8]. Their basic principles are that the analog input signal undergoes the process of spectrum shaping, then feeds through digital hardware in which a general purpose computer is utilized to perform operations such as delay, storage, addition, subtraction, and multiplication in a way to satisfy a set of specified difference equations between input and output. In other words, the digital output of the computer, after a set of calculations, and the digital input to the computer satisfy a specified transfer function in z-transform. The transfer function in z-transform can be obtained by direct transformation from Laplace transform. From the transfer function, a set of difference equations can be found and can be executed on the digital computer; the data output of the computer can be collected as the final output, then the transfer function of a digital filter is realized. This technique of filtering by implementation of difference equations is sometimes referred to as "recursive filtering," if the present value of the output depends not only on the present and past values of the input but also on the previous values of the output; or "nonrecursive filtering," if the present value of the output does not depend on the previous values of the output [4].

This paper discusses two topics: (1) transfer function realization and (2) driving point function realization using digital elements. The former concerns digital filter design technique. The latter concerns the individual component realization and the complicated network realization. The philosophy and approach discussed are different from those mentioned above. Essentially what we want is to construct small-size real time digital filters and other real time components by using digital building blocks as substitutes for analog components; these were made possible due to increasing speeds and decreasing cost of microelectronic digital circuitry. We use a digital integrator (a device containing digital building

blocks such as registers, adders, and logic gates which perform digital integration) as the basic building block for the realization of the transfer and driving point functions. Therefore, the mathematical models are differential equations rather than difference equations. Though the Laplace transformation was used to specify the transfer and driving point functions, the outputs were not precisely the same as those outputs specified by the transformation. However, they are essentially a very good approximation to the specified function and they are compatible with the continuous (analog) type of network.

As far as the digital filters are concerned, their input and output are not discrete, but are piecewisely continuous increments. Digital increments are the only data transferred between the digital integrators inside the digital filter. By this means, much shorter time is needed as compared to transferring of the full word. If only an analog signal is available at the input, a device called Analog to Digital Increment Converter (ADIC) can be used. More will be said concerning ADIC in Appendix D. Digital increments and up to date quantized output are both available at the output of the digital filters. The digital filters designed in this fashion definitely have an advantage over the analog filters because of their small size, accuracy, stability, and real time controllability. They also have a definite advantage over the sample-data digital filters because of their small size, compactness, fast speed, and shorter delay time.

For the driving point functions realization, if analog-to-digital increment (ADIC) and digital-to-analog (DAC) converters are available, then all the driving point functions can be realized by using digital elements only. It is also hoped that at some future time, ADIC and DAC can both be integrated, at which time all the driving point functions can be integrated without difficulty, by the method discussed in Chapter V. High Q inductors or high capacity capacitors can then be made available.

The method has additional advantages: (1) the realization of negative elements is as easy as positive elements; (2) the accuracy of the component value can be improved by increasing the length of the registers; and (3) the size of the network realized can be made very small and it is, in fact, independent of the frequency, provided that the frequency is not too high.

Chapter II
DIGITAL INTEGRATOR

1. Principles of Digital Integration

Digital integrators employ an approximate representation of the integral of a function as the sum of the areas of elementary rectangles, each of which corresponds to a definite increment Δx of the independent variable x.

Suppose that we are required to find the integral of a given function $y = f(x)$. This integral represents the area bounded by the curve $y = f(x)$ and the abscissa, but the integral may be approximated as the sum of the areas of the elementary rectangles. The height of each of the rectangles is the current ordinate y and the base is the increment Δx of the independent variable x; each increment is obtained by dividing the entire range of change in variable x into equal increments. As shown in Fig. 1, the integral z is found to be

$$z = \int_{x_o}^{x} y\, dx = \lim_{n \to \infty} \sum_{i=1}^{i=n} y_i \Delta x_i + R_o \approx \sum_{i=1}^{i=n} y_i\, \Delta x_i + R_o \qquad (2.1)$$

where $y = f(x)$

and

$$\Delta x_i = x_{i+1} - x_i \qquad (2.2)$$

is the increment of independent variable x.

If we make $\Delta x_i = 1$, then

$$z \approx \sum_{i=1}^{i=n} y_i + R_o \qquad (2.3)$$

where R_o is the initial value of the integral.

Thus to compute the integral, it is sufficient to add all the ordinates corresponding to each of the elementary rectangles. Consequently, the integration process reduces to a summation of the members representing the ordinates. Each ordinate is equal to the preceding ordinate plus (or minus) the ordinate increment Δy_i for the interval $\Delta x_i = x_{i+1} - x_i$. Hence the value of ordinate y_i may be computed by adding to each preceding ordinate the increment Δy. For example,

$$y_1 = y_o + \Delta y_1$$

$$y_2 = y_1 + \Delta y_2$$

$$\vdots$$

$$y_n = y_{n-1} + y_n = y_o + \Delta y_1 + \Delta y_2 + \ldots + \Delta y_n \qquad (2.4)$$

Hence it is seen that the current value of the ordinate y_i may be obtained by accumulating all increments of the ordinate up to the increment Δy_i. (This integration method is called Triangular Integration.)

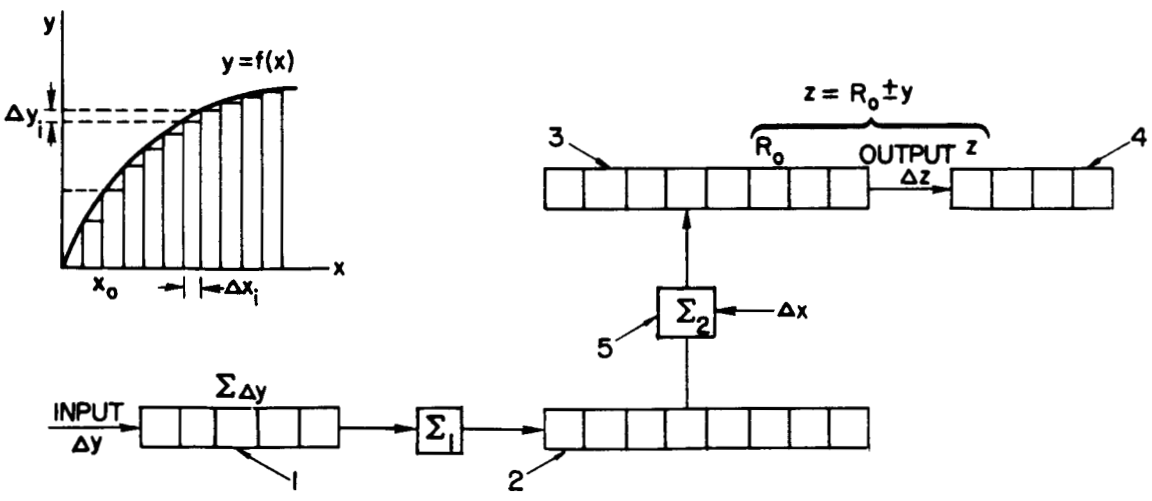


Fig. 1 BLOCK DIAGRAM OF DIGITAL INTEGRATOR

2. Operation of the Digital Integrator

Referring to Fig. 1, assume that increments Δx, Δy, and Δz are given in the form of individual pulses. The reversible counter 1 counts the incoming pulses. For each integration step, the counter accepts pulses from several input channels, and the number of pulses accumulated by the counter is regarded as an increment of the integrand, where the increment is the sum of several elementary increments, i.e., $\Sigma\Delta y$.

For each integration step, the number $\Sigma\Delta y$ stored in counter 1 is summed with the number y_o stored in register 2 (y register) by means of the Σ_1 adder. As a result of addition (or subtraction), a new ordinate $y = y_o \pm \Sigma\Delta y$ is obtained for each integration step.

During each integration step, the number y stored in register 2 is added to register 3 (R register) wherein the number corresponding to the sum of the ordinates (i.e., the value of the integral) is stored. Summation of numbers y and R_o is achieved by the Σ_2 adder,

$$z = R_o \pm y$$

where R_o is the number of the integral initially stored in register 3. Summation is performed each time the input Δx of the integrator receives a pulse representing an increment of the independent variable x.

Let y_i be the contents of the y register when the i-th Δx pulse occurs. If the R register has sufficient capacity, after the n-th Δx pulse it will contain the sum

$$\sum_{i=1}^{n} y_i \, \Delta x_i$$

which is an approximation of

$$\int_o^{x_n} y \, dx$$

where

$$x_n = \sum_{i=1}^{n} \Delta x_i$$

However, if the R register has the same length (or less) as the y register and register 4 (termed the Δz register) is appended to the R register, the Δz register will be used to store the carriers generated in the R register. Hence during summation the R register may overflow and we have

$$\Delta z_i + R_i = R_{i-1} + y_i \, \Delta x_i \tag{2.5}$$

where R_i is the number in the R register after the i-th Δx pulse has added y_i to the R register. Therefore,

$$\sum_{i=1}^{n} \Delta z_i = \sum_{i=1}^{n} y_i \, \Delta x_i + R_o - R_n \tag{2.6}$$

and the sum of the Δz_i binary bits is an approximation to

$$\sum_{i=1}^{n} y_i \, \Delta x_i$$

with a round-off error $R_o - R_n$. Therefore the device is an approximate integrator. Register 3 and register 4 may be regarded as two parts of a single register having 2n bits. Register 3 holds the less significant bits of integral z, and register 4 holds the most significant bits located at $(n+1)$ to 2n bit positions. In such an arrangement, integral z has at the most twice as many bits as the integrand register 2.

The process of accumulating the overflow pulses Δz from the integrator output by register 4 is an integration process, as already shown.

In general, the integral increment Δz is

$$\Delta z = k\ y\ \Delta x \tag{2.7}$$

where k is a constant scale factor. For binary numbers

$$k = \frac{1}{2^n} \tag{2.8}$$

and n is the number of bits of register y or z, where R_o denotes the number in register 3.

The coefficient k is the scale factor of the digital integrator. It signifies that, for $y = 1$ and $\Delta x = 1$, 2^n summations (or 2^n steps) are required to obtain one overflow pulse Δz at the output of register 3. If y is equal to 2^n and $\Delta x = 1$, then there will be an overflow pulse for each of the integration steps.

Converting from increments Δx, Δy, and Δz to derivatives of x, y, and z in time, the formula $\Delta z = k\ y \Delta x$ can be written as follows

$$\frac{dz}{dt} = k\ y\ \frac{dx}{dt} = k\left(\int \frac{dy}{dt}\ dt\right) \frac{dx}{dt} \tag{2.9}$$

or in the form of an integral

$$z = k \int y\ dx \tag{2.10}$$

3. Trapezoidal Integration

The error in rectangular integration (as in section 1) can be reduced by using the trapezoidal rule such that the curve $y = f(x)$ at each interval Δx is approximated by a chord. This rule is equivalent to a summation of the areas of "mean" rectangles; as shown in Fig. 2, each rectangle has a mean ordinate approximated by

$$y_o + \frac{1}{2}\ (\Delta y)$$

The increment of the integral (the area of an elementary rectangle) is

$$\Delta z = \left(y + \frac{1}{2} \Delta y \right) \Delta x \tag{2.11}$$

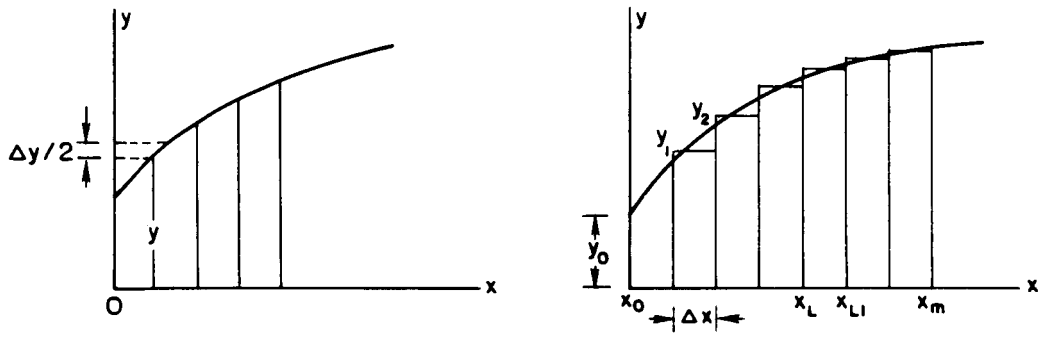


Fig. 2 TRAPEZOIDAL INTEGRATION

If the trapezoidal integration method is employed, the integral is approximated by

$$\int_{x=x_o}^{x_n} F(x)\, dx \approx \frac{x_n - x_o}{n} \left(\frac{y_o + y_1}{2} + \frac{y_1 + y_2}{2} + \ldots + \frac{y_{n-1} + y_n}{2} \right)$$

$$= \frac{x_n - x_o}{n} \left(\frac{y_o + y_n}{2} + y_1 + y_2 + \ldots + y_{n-1} \right) \tag{2.12}$$

Similar accuracy in integration may be obtained by summing the areas of rectangles having at each interval a mean ordinate (refer to Fig. 3) of

$$y_m = y + \frac{1}{2} (\Delta y) \tag{2.13}$$

where y is the value of the current ordinate and Δy is the ordinate increment for the given interval.

The approximate value of the interval will then be

$$\int_{x=x_0}^{x_n} y\,dx \approx \frac{x_n - x_o}{n} (y_{m1} + y_{m2} + \ldots + y_{mn}) \qquad (2.14)$$

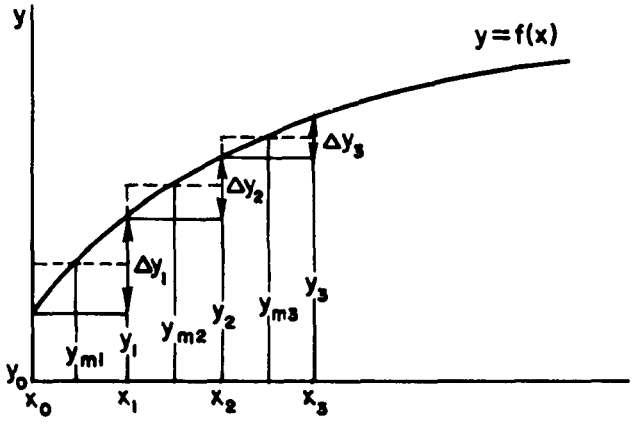


Fig. 3 INTEGRATION BY THE "MEAN" RECTANGLE METHOD

Integration by using Eq. (2.14), which approximates the accuracy of the trapezoidal formula, consists of three summing operations:

a. The next ordinate value y^* is determined by adding the increment Δy at the integrator output, i.e.,

$$y_o \pm \Delta y_1 = y_1^*$$

$$y_1 \pm \Delta y_2 = y_2^*, \quad \text{etc.} \qquad (2.15)$$

b. Half of the ordinate increment is added to the initial value of the ordinate, i.e.,

$$y_o \pm \frac{\Delta y_1}{2} = y^*_{m1}$$

$$y_1 \pm \frac{\Delta y_2}{2} = y^*_{m2}$$

$$y_2 \pm \frac{\Delta y_3}{2} = y^*_{m3}, \qquad \text{etc.} \tag{2.16}$$

c. The resulting mean ordinate values are summed in the accumulator (R register). The sum of the mean ordinate values is taken as the approximate value of the integral

$$R \approx y^*_{m1} + y^*_{m2} + y^*_{m3} + \ldots + y^*_{mn} \tag{2.17}$$

provided that $\Delta x = 1$.

Note: the trapezoidal method is significant only where other errors (such as round-off error) are sufficiently small [6]. Hence the ternary method of representing increments Δx, Δy, and Δz is necessary.

The block diagram for integration by the mean-rectangle method is shown in Fig. 4.

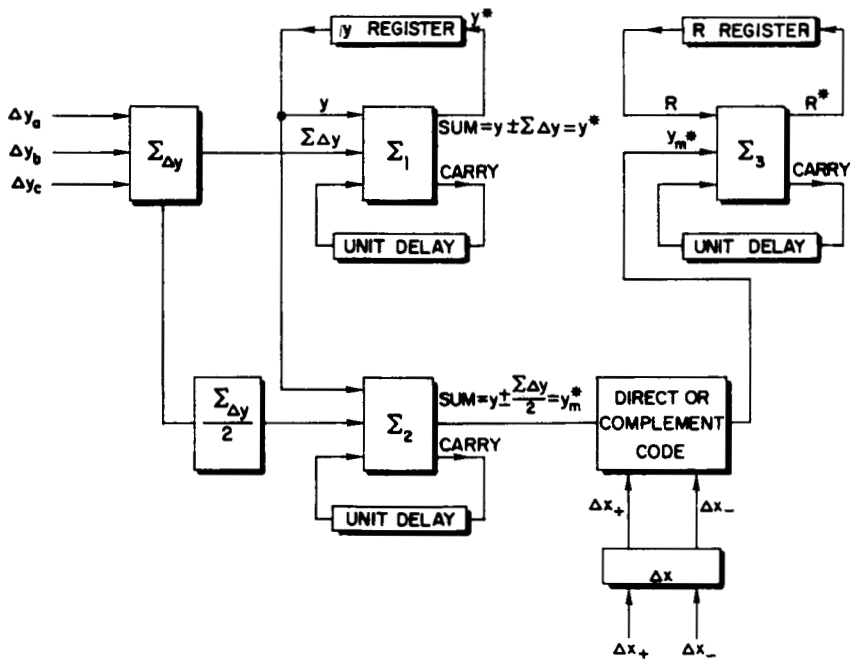


Fig. 4 BLOCK DIAGRAM OF THE INTEGRATION BY THE MEAN-RECTANGLE METHOD

4. Solution of Differential Equations by Using Digital Integrators

Functionally, a digital integrator is represented by a schematic, as shown in Fig. 5, where arrows indicate the direction of data flow. The inputs dx and dy are incremental inputs and dx can be either an incremental time input dt (say clock pulses), or even a function of y. The output dz observes the relations

$$dz = y\,dx$$

or (2.18)

$$z = \int y\,dx$$

Sometimes, more than one digital integrator is used to solve a certain problem, in which case they can be connected in such a way that the overflow of one integrator is connected to the input of the other integrator. From time to time, the scalar multiplication is also required; in that event, an integrator can also be used. The dy input terminal is left open and the content of the y register is set to a desired constant k, then dz = k dx. In other words, the output dz is equal to k times the dx input (see Fig. 6).

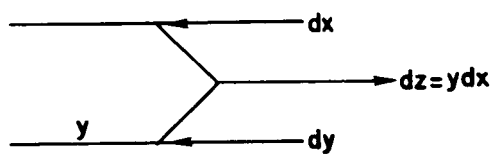


Fig. 5 FUNCTIONAL SCHEMATIC OF A DIGITAL INTEGRATOR

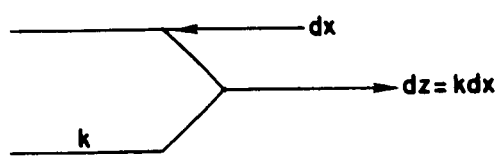


Fig. 6 SCHEMATIC OF A CONSTANT MULTIPLIER

A set of digital integrators can be used to solve an ordinary differential equation of any order or degree, linear or nonlinear, or even a simultaneous set of such equations. Normally, for solving a differential equation, two steps--mapping and scaling--are involved. Mapping consists in specifying how the operational units (integrators and adders) should be interconnected so that the variable or variables of interest are generated within the system. Since a digital integrator has a limited capacity of registers, it is necessary to assure that intermediate results stay within the specified ranges during the running of a problem, so that the estimated maximum values of each of the variables can be scaled to a meaningful range. This is amplitude scaling. Sometimes, frequency scaling is also employed to ensure proper operation. For a real time device, the amplitude and frequency of the input have to be specified in a workable range; therefore no frequency scaling is permissible. But if amplitude scaling is necessary, it can be done either by adjusting the ratio of the analog-to-digital converter, or the digital-to-analog converter, or by using a multiplier to restore the scale factor.

Example 1: Solve the following differential equation:

$$\frac{d^2y}{dt^2} - \frac{dy}{dt} - \sin y = 0$$

Solution: Differentiating the given differential equation once, we get

$$d\ddot{y} = d\dot{y} + d(\sin y)$$

The solution y can be obtained by interconnecting the digital integrators, as shown in Fig. 7, where the independent variable input is dt. The initial conditions of $y(0)$, $\dot{y}(0)$, and $\ddot{y}(0)$ have not been taken into consideration. They can be treated by adding one extra register, called I register, to store the initial condition for each integrator. The data transfer from I register to y register will be done at the beginning of the operation cycle.

Similarly, nonlinear differential equations can be solved without difficulty.

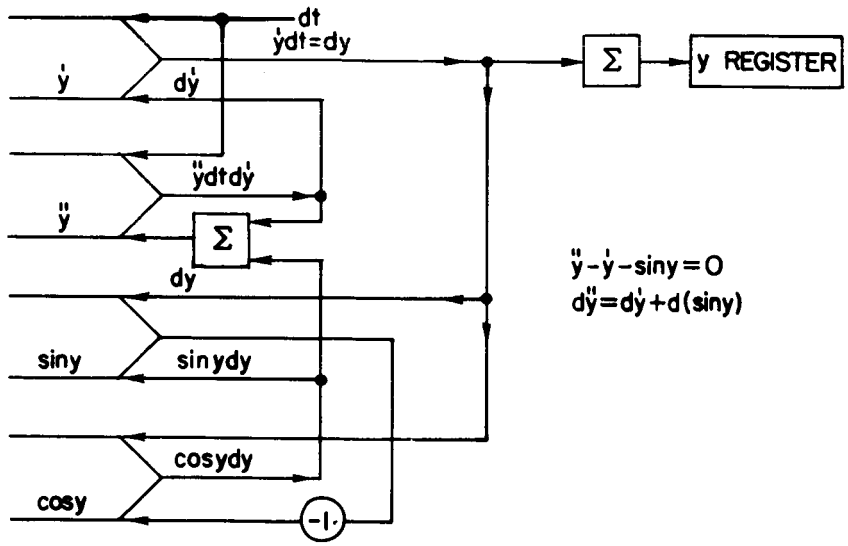


Fig. 7 CONNECTIONS FOR SOLUTION OF $\ddot{y} - \dot{y} - \sin y = 0$

Example 2: Solve the following nonlinear differential equation

$$\ddot{y}(t) - y(t)\,\dot{y}(t) + y(t)^2 = 0$$

Solution: Differentiating the given equation once, we have

$$d\ddot{y} = \dot{y}dy + yd\dot{y} - 2ydy$$

The solution can be obtained by interconnecting digital integrators, as those shown in Fig. 8. For simplicity, the multiplication by a constant has been shown by a circle, with desired constant indicated.

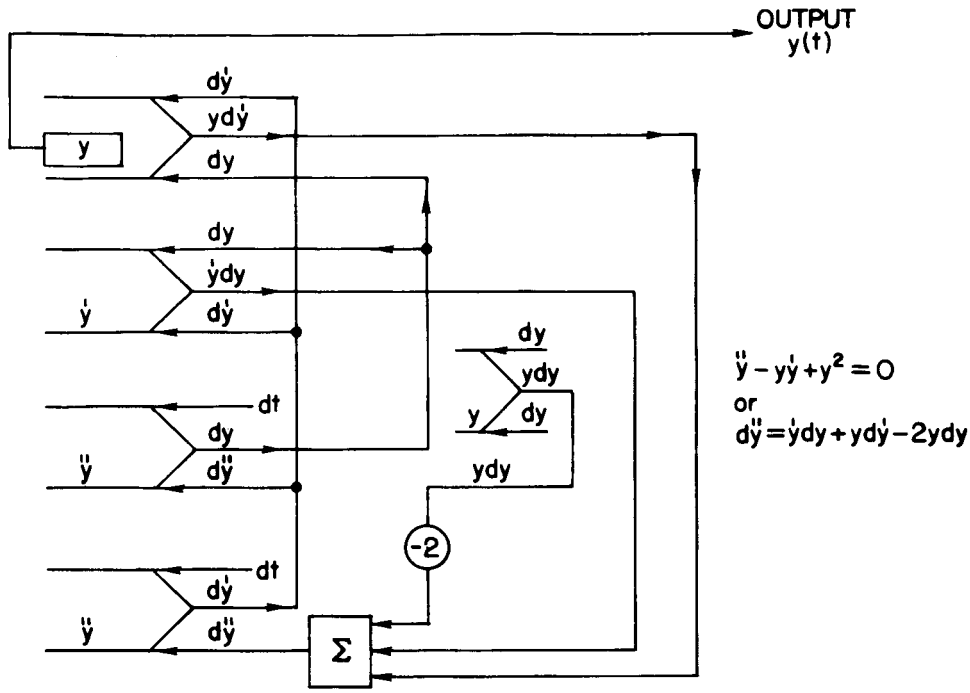


Fig. 8 CONNECTIONS FOR SOLUTION OF $\ddot{y} - y\,\dot{y} + y^2 = 0$

Chapter III

REALIZATION OF TRANSFER FUNCTIONS BY USING DIGITAL INTEGRATORS

The transfer function is defined as the ratio of the Laplace transform of the output quantity to the Laplace transform of the input, with the restriction that the initial conditions appearing in the transformed differential equation (or equations) are all zero [11]. Let

$$\text{Transfer function} = G(s) = \frac{\mathcal{L}\{y(t)\}}{\mathcal{L}\{x(t)\}} = \frac{Y(s)}{X(s)} \tag{3.1}$$

where $y(t)$ and $x(t)$ are output and input, respectively, of a system. Suppose we have a black box with transfer function $G(s)$. It is our purpose to relaize the black box with digital elements.

Example 3: Given a simple transfer function

$$G(s) = \frac{1}{s + 1} = \frac{Y(s)}{X(s)}$$

where s is the complex variable. Realize $G(s)$ with only digital elements, and show the steps of realization in detail.

Solution: First, transforming the given function back to the time domain, we have

$$\dot{y}(t) + y(t) = x(t)$$

or

$$d\dot{y} = dx - dy$$

The network having the above characteristics can be realized by interconnecting the digital building blocks, as shown in Fig. 9.

For convenience, the network realized contains two registers of eight bits each (see Fig. 10). The eight whole number bits have a

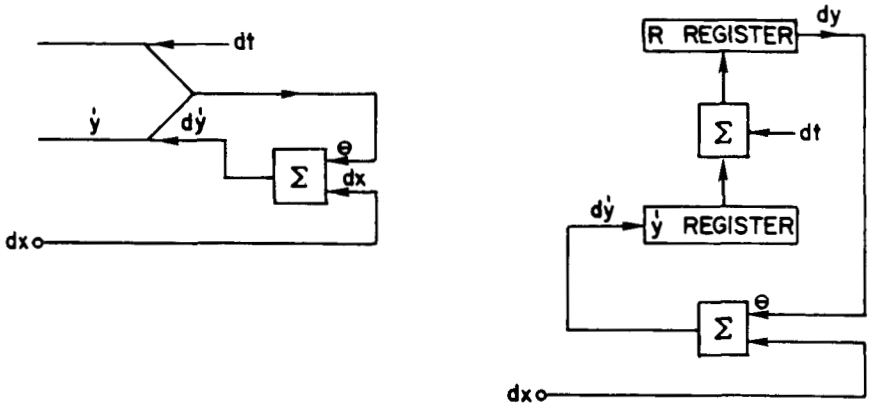


Fig. 9 REALIZATION OF TRANSFER FUNCTION $1/(s + 1)$

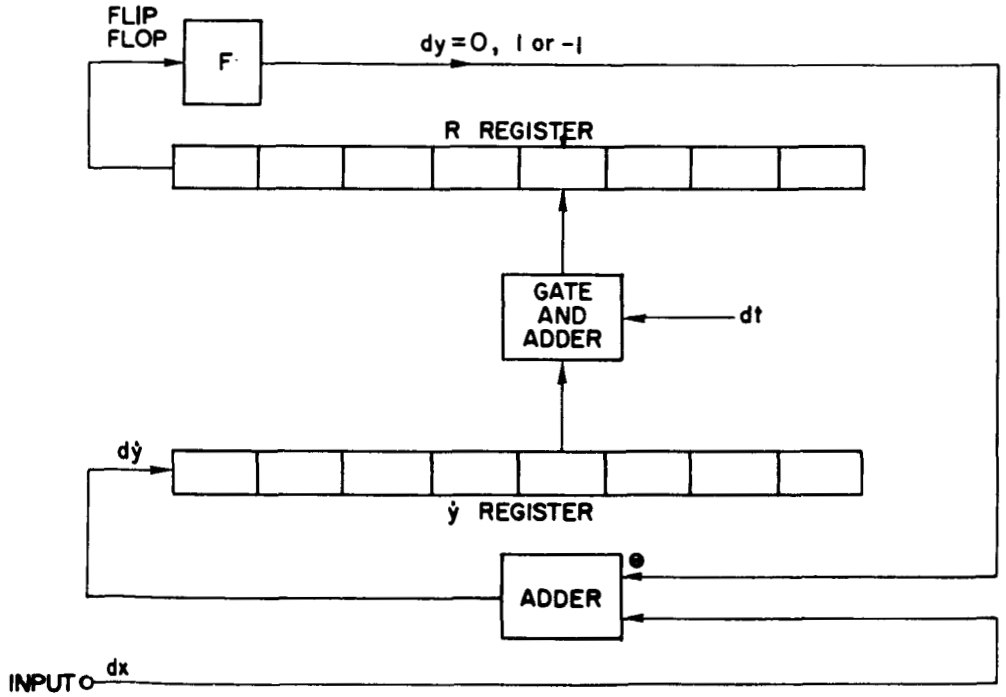


Fig. 10 TRANSFER FUNCTION $1/(s + 1)$ WITH 8-BIT REGISTERS

capacity of $2^8 = 256$ maximum. The input dx can be obtained as the difference of the two consecutive samples of the input. In other words, dx can be obtained by letting

$$dx_i = x(t_i) - x(t_{i-1})$$

During the presence of the dt pulse, the duration of dt can be roughly divided into three sub-time intervals, T_1, T_2, and T_3. In the time interval T_1, the summation of dx and -dy is performed, where -dy comes from the flip flop F, which contains the overflow of the R register. Within T_2, the addition of (dx-dy) to the contents of the $\dot{y}$ register will be placed into the $\dot{y}$ register. Meantime, the flip flop F will be reset. Then in interval T_3, the contents of the $\dot{y}$ register is added to the contents of the R register and the result is placed in the R register. If the R register has no overflow, then the flip flop F output is zero, i.e., $dy = 0$. If there is an overflow, the output of F is one. In other words, $dy = 1$ or $dy = -1$. The choice depends on the sign of the R register.

For the case, of Example 3, $dy = -1$ if $R \leq -256$ and $dy = 1$ if $R \geq 256$.

The structure of the digital integrator can be connected either in serial or in parallel fashion. The structure is much simpler for the serial digital integrator but the operation speed is much slower.

The simulation of the given problem has been done on the digital computer. The computer program of the sinusoidal response and unit step response of the network are shown in Appendices B and C. The step-by-step calculations are tabulated in Tables 1 and 2, and the graphical responses are shown in Figs. 11 and 12, where the method of triangular integration with the ternary code is assumed.

On the basis of the results of the unit step response and sinusoidal response, a comparison of the digital realization and the analog realization can be made. From the last two columns of Tables 1 and 2, it is seen that the contents of YREG (digital) are a good approximation to the real solution (analog). Note that the figures tabulated have been scaled 256 times. At most they differ by one per 256 for registers of 8-bit length.

Table 1

SINUSOIDAL RESPONSE OF
$G(s) = 1/(s + 1)$ RESULTS BY DIGITAL ELEMENT REALIZATION

Input: $x(t) = 255 \sin t$

Let $t = T/256$, $x_i(t) = 255 \sin(T/256\ i)$, $dx = x_i(t) - XREG_{i-1}$

$XREG_i = XREG_{i-1} + dx$ $\qquad$ $\dot{Y}REG_i = \dot{Y}REG_{i-1} + d\dot{Y}$

$d\dot{Y} = dx - dY$ $\qquad$ $RREG_i = RREG_{i-1} + \dot{Y}REG$

$dY = 1$ while RREG has overflow, otherwise $dY = 0$

$YREG_i = YREG_{i-1} + dY$

Assume that all registers have 8-bit length, and that the method of triangular integration with trinary code is used.

T	XREG	$\dot{Y}$REG	RREG	dY	YREG	Real Solution y(t)
0	0	0	0	0	0	0
1	0	0	0	0	0	0.0019
2	1	1	1	0	0	0.0077
3	2	2	3	0	0	0.0175
4	3	3	6	0	0	0.0310
5	4	4	10	0	0	0.0485
6	5	5	15	0	0	0.0697
7	6	6	21	0	0	0.0948
8	7	7	28	0	0	0.1236
9	8	8	36	0	0	0.1563
10	9	9	45	0	0	0.1927
11	10	10	55	0	0	0.2329
12	11	11	66	0	0	0.2768
13	12	12	78	0	0	0.3244
14	13	13	91	0	0	0.3758
15	14	14	105	0	0	0.4308
16	15	15	120	0	0	0.4895
17	16	16	136	0	0	0.5519
18	17	17	153	0	0	0.6179
19	18	18	171	0	0	0.6876
20	19	19	190	0	0	0.7609
21	20	20	210	0	0	0.8377
22	21	21	231	0	0	0.9182
23	22	22	253	0	0	1.0022
24	23	23	20	1	1	1.0898
25	24	23	43	0	1	1.1809
26	25	24	61	0	1	1.2756
27	26	25	92	0	1	1.3737
28	27	26	118	0	1	1.4754

Table 1 (continued)

T	XREG	YREG	RREG	dY	YREG	Real Solution y(t)
32	31	30	232	0	1	1.9166
33	32	31	7	1	2	2.0355
41	40	38	27	1	3	3.1079
64	63	56	95	0	7	7.3334
96	93	78	209	0	15	15.7509
128	122	96	192	0	26	26.6718
160	149	110	175	0	39	39.6026
192	173	120	33	1	54	54.0565
224	195	126	153	0	69	69.5563
256	214	129	159	0	85	85.6381
288	230	129	203	0	101	101.8551
320	241	124	171	0	117	117.7813
352	250	118	216	0	132	133.0156
384	254	108	244	0	146	147.1856
416	254	95	157	0	159	159.9513
448	251	81	178	0	170	171.0087
480	244	65	210	0	179	180.0927
512	232	46	188	0	186	186.9797
576	199	6	71	0	193	193.4906
640	153	-37	-172	0	190	189.6576
704	98	-78	-292	0	176	175.3460
768	36	-116	-115	1	151	151.1551
800	5	-131	-222	0	136	135.7300
832	-27	-146	-59	1	118	118.3627
896	-89	-168	-194	0	79	78.8314
960	-145	-181	-163	1	35	34.8820
992	-170	-183	-104	1	12	12.0636
1000	-176	-183	-33	1	7	6.3290
1024					-11	
1152					-97	
1280					-158	
1408					-180	

Table 2

UNIT STEP RESPONSE OF $G(s) = 1/(s + 1)$ RESULTS
BY DIGITAL ELEMENTS REALIZATION

T	XREG	$\dot{Y}$REG	RREG	dY	YREG	Real Solution y(t)
0	0	0	0	0	0	0
1	255	255	255	0	0	0.9980
2	255	255	253	1	1	1.9922
3	255	254	251	1	2	2.9824
4	255	253	248	1	3	3.9689
5	255	252	244	1	4	4.9514
6	255	251	239	1	5	5.9302
7	255	250	233	1	6	6.9051
8	255	249	226	1	7	7.8762
9	255	248	218	1	8	8.8436
10	255	247	209	1	9	9.8072
11	255	246	199	1	10	10.7670
12	255	245	188	1	11	11.7230
13	255	244	176	1	12	12.6754
14	255	243	163	1	13	13.6240
15	255	242	149	1	14	14.5690
16	255	241	134	1	15	15.5102
17	255	240	118	1	16	16.4478
18	255	239	101	1	17	17.3817
19	255	238	83	1	18	18.3120
20	255	237	64	1	19	19.2387
21	255	236	44	1	20	20.1617
22	255	235	23	1	21	21.0811
23	255	234	1	1	22	21.9970
24	255	233	234	0	22	22.9093
25	255	233	211	1	23	23.8180
26	255	232	187	1	24	24.7232
27	255	231	162	1	25	25.6249
28	255	230	136	1	26	26.5230
29	255	229	109	1	27	27.4177
30	255	228	81	1	28	28.3088
31	255	227	52	1	29	29.1965
32	255	226	22	1	30	30.0807
33	255	225	247	0	30	30.9615
64	255				56	56.6269
96	255				80	80.0539
128	255				100	100.7281
160	255				118	118.9730
192	255				134	135.0741
224	255				149	149.2833
256	255				161	161.8228
288	255				172	172.8889
320	255				182	182.6547
352	255				191	191.2730
384	255				198	198.8786

Table 2 (continued)

T	XREG	YREG	RREG	dY	YREG	Real Solution y(t)
416	255				205	205.5906
448	255				211	211.5138
480	255				216	216.7411
512	255				221	221.3541
576	255				228	229.0177
640	255				234	234.9862
704	255				239	239.6344
768	255				242	243.2545
800	255				244	244.7521
832	255				245	246.0738
896	255				247	248.2694
960	255				249	249.9794
992	255				250	250.6868

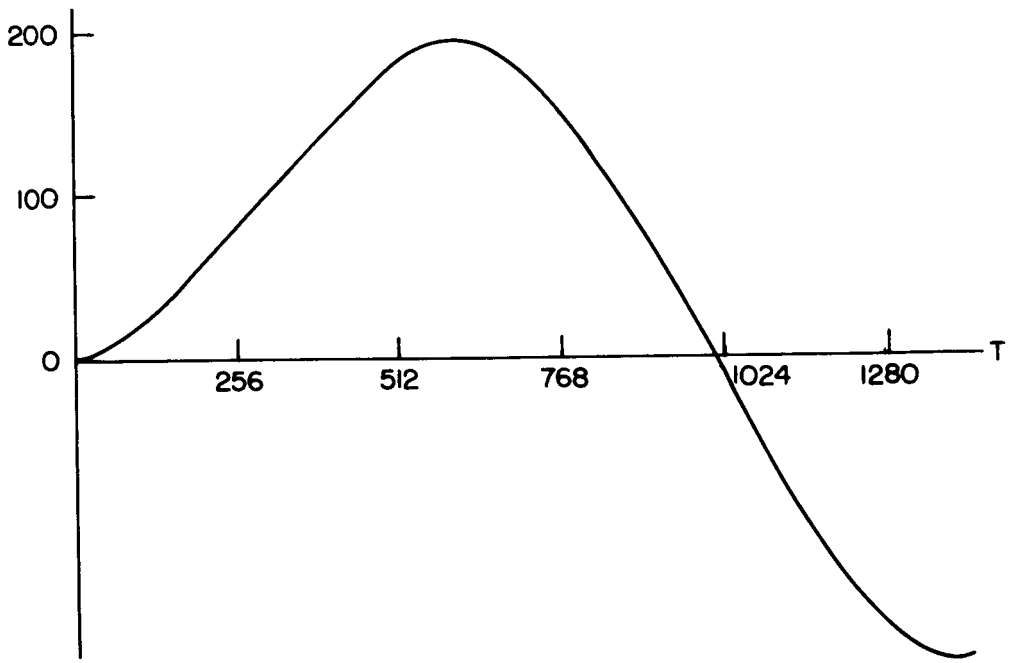


Fig. 11 SINUSOIDAL RESPONSE OF $G(s) = 1/(s + 1)$
WITH INPUT $x(t) = 255 \sin (T/256)$

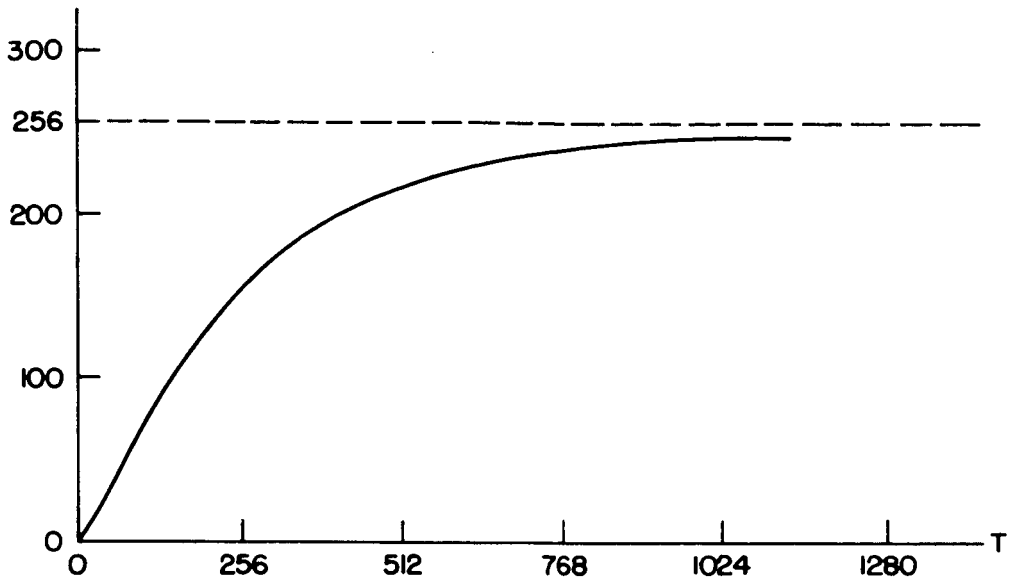


Fig. 12 STEP RESPONSE OF $G(s) = 1/(s + 1)$
WITH INPUT $x(t) = 255$

If registers with more than 8 bits are used, the accuracy will be proportionally increased.

The closeness of the two solutions--digital and analog--is shown in Fig. 13; if trapezoidal integration is employed in the design integrators, the accuracy of the digital solution will be further improved.

It is noteworthy that even if the input is not started from zero at $t = 0$ as was assumed in Tables 1 and 2, the digital realization is still valid.

For better approximation, more register bits can be used. In the presence of the fractional numbers, as well as the whole numbers, a few fractional number bits can be attached to the end of the whole numbers and a fixed decimal point assigned. For example, to maintain an accuracy of 10^{-3}, ten bits corresponding to the fractional number will be used in addition to the whole number bits, since $2^{-10} = 1/1024 \approx 10^{-3}$.

The above simple example is used to illustrate how the given transfer function can be realized. As for the realization of a complicated

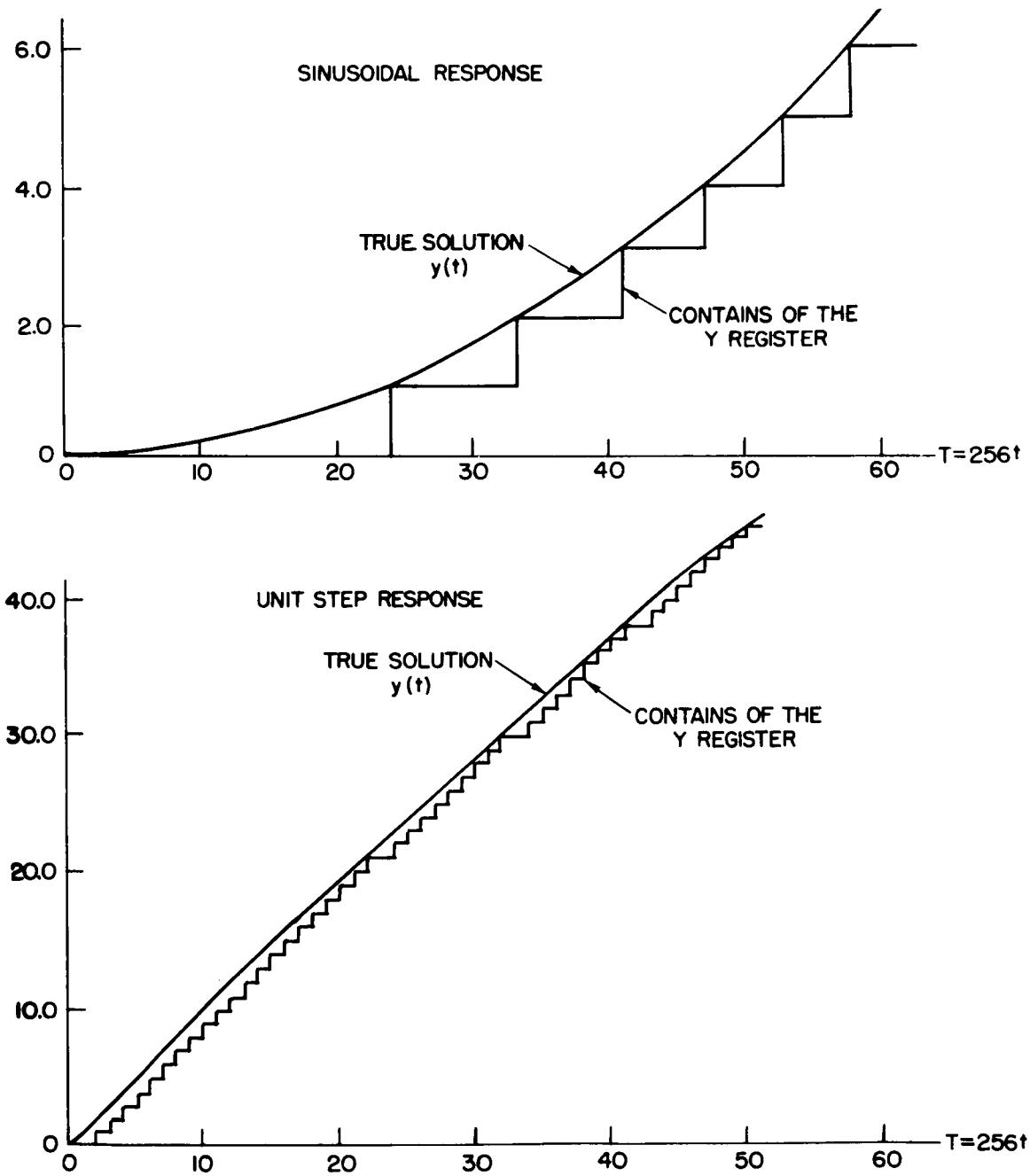


Fig. 13 COMPARISON OF THE TWO RESPONSES
OF ANALOG AND DIGITAL REALIZATION

transfer function, we could use more digital integrators. In general, a transfer function has the form

$$G(s) = \frac{b_n s^n + b_{n-1} s^{n-1} + \ldots + b_1 s + b_o}{s^m + a_{m-1} s^{m-1} + \ldots + a_1 s + a_o} = \frac{Y(s)}{X(s)} \qquad (3.2)$$

where $m \geq n$.

After dividing by s^n, $G(s)$ has the following expression

$$G(s) = \frac{Y(s)}{X(s)} = \frac{b_n + b_{n-1} s^{-1} + \ldots + b_1 s^{1-n} + b_o s^{-n}}{s^{m-n} + a_{m-1} s^{m-1-n} + \ldots + a_1 s^{1-n} + a_o s^{-n}} \qquad (3.3)$$

The realization of this general transfer function is quite tedious but straightforward; the realization of a specific case, $m = 5$, $n = 3$ is shown as follows:

Example 4: Realize the following transfer function:

$$G_1(s) = \frac{Y(s)}{X(s)} = \frac{b_3 + b_2 s^{-1} + b_1 s^{-2} + b_o s^{-3}}{s^2 + a_4 s + a_3 + a_2 s^{-1} + a_1 s^{-2} + a_o s^{-3}}$$

or

$$s^2 Y(s) = -a_4 s Y(s) + [b_3 X(s) - a_3 Y(s)] + [b_2 X(s) - a_2 Y(s)] s^{-1}$$

$$+ [b_1 X(s) - a_1 Y(s)] s^{-2} + [b_o X(s) - a_o Y(s)] s^{-3}$$

Solution: In the time domain, we have

$$\ddot{y} = -a_4 \dot{y} + (b_3 x - a_3 y) + \int (b_2 x - a_2 y)\, dt + \iint (b_1 x - a_1 y)\, dt\, dt$$

$$+ \iiint (b_o x - a_o y)\, dt^3$$

or

$$d\ddot{y} = -a_4\, d\dot{y} + (b_3 dx - a_3 dy) + (b_2 x - a_2 y)\, dt + \int (b_1 x - a_1 y)\, dt^2$$

$$+ \iint (b_o x - a_o y)\, dt^3$$

The realized network is shown in Fig. 14.

Another noteworthy subject is that those transfer functions to be realized need not have positive coefficients, since we are dealing with numbers internally while we are realizing the transfer functions. No extra efforts are required to change the addition operation to subtraction; therefore, transfer functions with negative coefficients, i.e., either negative transfer functions or transfer functions with transmission zeros at the right-half complex frequency plane, can be realized as easily as those illustrated.

The following examples will help in understanding the subject.

Example 5: Realize the following transfer functions:

a. $G_2(s) = \dfrac{Y(s)}{X(s)} = -\dfrac{1}{s + 1}$

b. $G_3(s) = \dfrac{Y(s)}{X(s)} = \dfrac{s - 1}{s^2 + s + 1}$

Solutions:

a. Step 1 Transform $G_2(s)$ back to time domain

$$\dot{y}(t) + y(t) = -x(t)$$

Step 2 Take the derivative of the above expression

$$d\dot{y} = -dx - dy$$

Step 3 After mapping the solution, the realized network of $G_2(s)$ is as shown in Fig. 15.

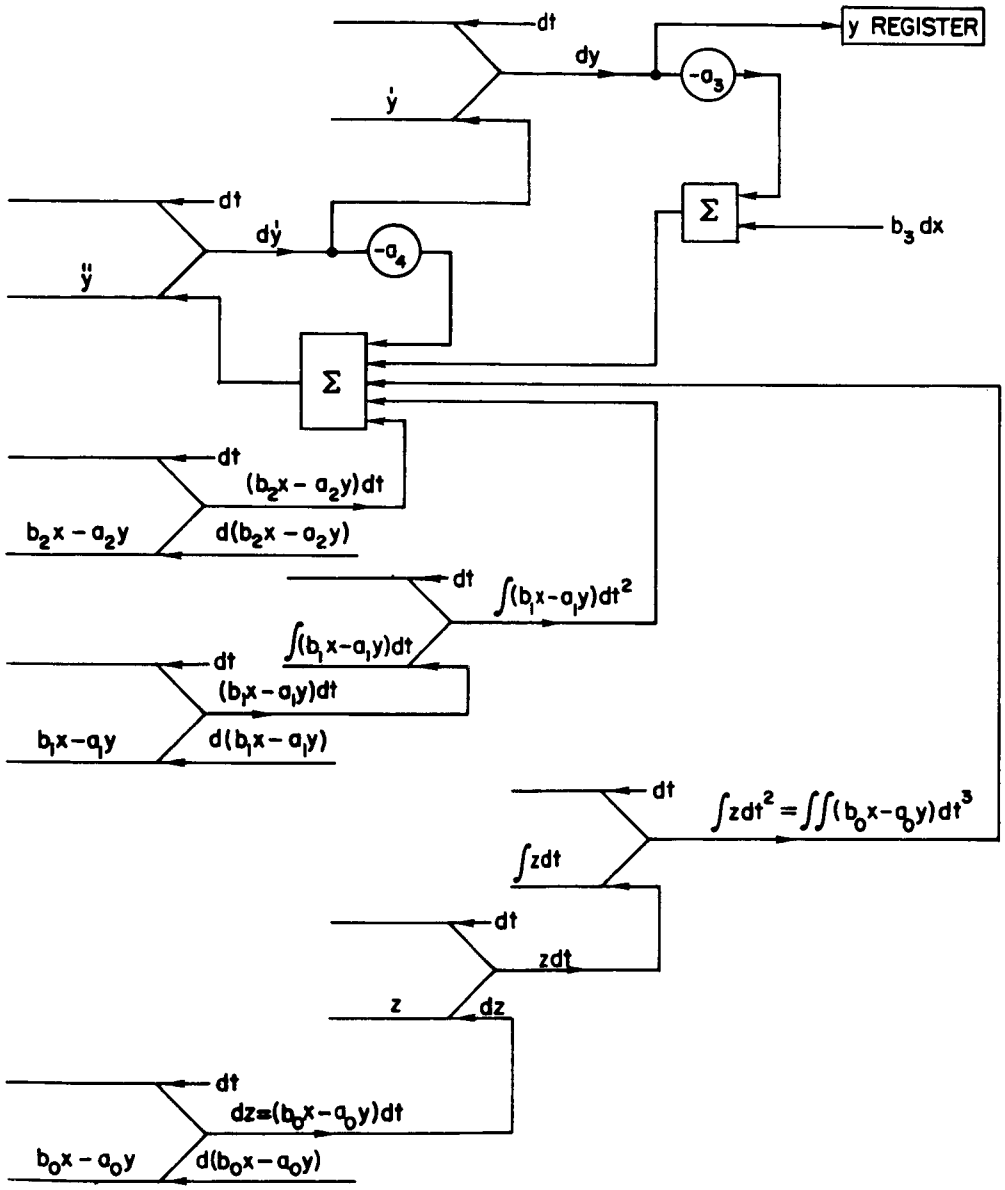


Fig. 14 REALIZATION OF TRANSFER FUNCTION

$$G_1(s) = \left(b_3 + b_2s^{-1} + b_1s^{-2} + b_0s^{-3}\right) \Big/ \left(s^2 + a_4s + a_3 + a_2s^{-1} + a_1s^{-2} + a_0s^{-3}\right)$$

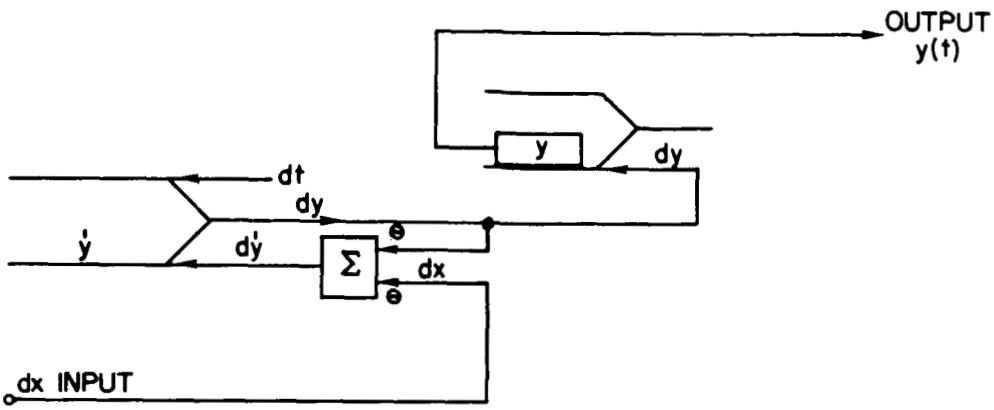


Fig. 15 REALIZATION OF $G_2(s) = -1/(s + 1)$

b. Step 1 Transform $G_3(s)$ back to the time domain with s divided through.

$$\dot{y}(t) + y(t) + \int y(t)\, dt = x(t) - \int x(t)\, dt$$

Step 2 Take the derivative of the above expression

$$d\dot{y} = dx - dy - [x(t) + y(t)]\, dt$$

Step 3 After mapping the solution, the realized network is as shown in Fig. 16.

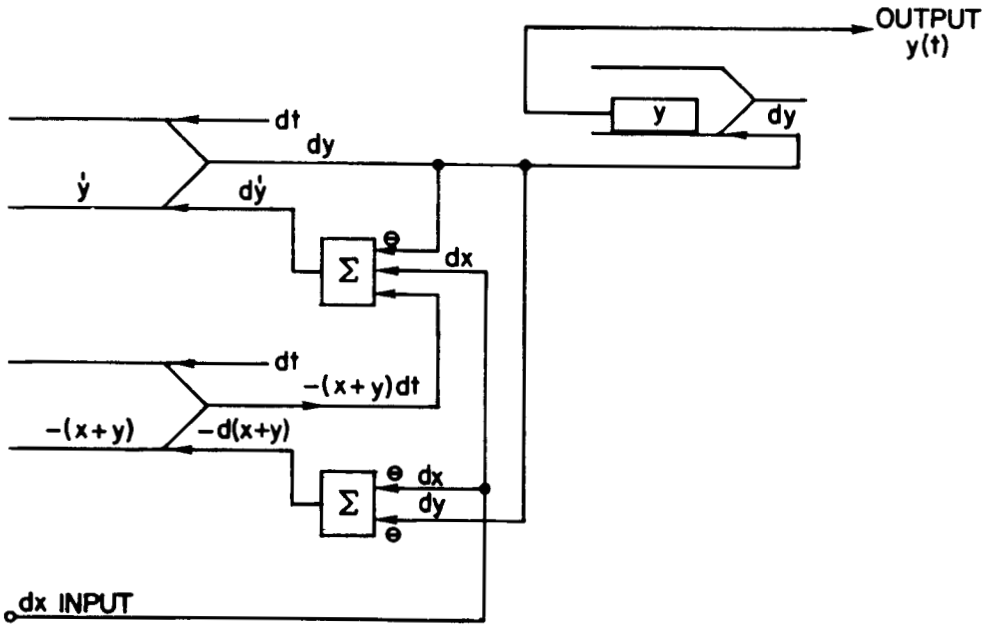


Fig. 16 REALIZATION OF $G_3(s) = (s + 1)/(s^2 + s + 1)$

Chapter IV
DIGITAL FILTER DESIGN TECHNIQUE

1. Digital Filters

Generally speaking, the term digital filter refers to the computational process or algorithm by which a sampled signal or sequence of numbers (acting as an input) is transformed into a second sequence of numbers termed the output signal. So far, the real time digital filters designed are utilizing digital computers to execute the difference equation of which the desired transfer functions of digital filters in z-transform are satisfied [4,5,8].

The digital filters we are proposing here are somewhat different from the prescribed definition and have different structures. Physically, they are composed of small size digital building blocks, such as registers, logic gates, and adders, and they deal with quantized signals rather than with discrete signals. Their functions are analogous to the continuous (analog) filters, and they are actually very good approximations to analog filters. The digital filters designed in the proposed fashion have definite advantages over the analog filters and sample-data digital filters because of their small size, accuracy, stability, and real time controllability.

2. Digital Filter Design Techniques and Implementation

The design procedures for continuous filters, such as Butterworth and Chebyshev filters, are treated in standard texts [3,9,10]. They are discussed briefly in Appendix A.

The technique of designing a digital filter by using digital integrators is fairly straightforward. After the designer decides which type of filter fits his needs, he follows the outlines listed in Appendix A until he gets the desired transfer function. He then transforms the transfer function back to time domain whereby a linear differential equation can be formed. He follows the procedures listed in section III and obtains a diagram of interconnected digital integrators. Thus the design of the digital filter is completed.

Two methods of realization are shown here.

Method 1

Suppose a desired transfer function of a digital filter has been found; for example,

$$G(s) = \frac{k}{s^3 + fs^2 + gs + h} = \frac{Y(s)}{X(s)} \tag{4.1}$$

Then after cross-multiplying, we obtain

$$(s^3 + fs^2 + gs + h)\, Y(s) = k\, X(s) \tag{4.2}$$

Transforming back to time domain, we get

$$\dddot{y}(t) + f\ddot{y}(t) + g\dot{y}(t) + hy(t) = k\, x(t) \tag{4.3}$$

Differentiating once and rearranging, we have

$$d\dddot{y} = k\, dx - f\, d\ddot{y} - g\, d\dot{y} - h\, dy \tag{4.4}$$

The structure can be formed in series; that is, the output of one integrator feeds into the input of another integrator, as shown in Fig. 17.

Method 2

A different method can be used to deal with the design of a digital filter, but the same transfer function as used in Method 1, i.e.,

$$G(s) = \frac{k}{s^3 + fs^2 + gs + h} = \frac{Y(s)}{X(s)} \tag{4.1}$$

can be expanded into partial fraction form

$$G(s) = \frac{k}{(s + a)\,(s^2 + bs + c)} = \frac{\ell}{s + a} + \frac{rs + q}{s^2 + bs + c}$$

$$= G_a(s) + G_b(s) \tag{4.5}$$

where

$$G_a(s) = \frac{\ell}{s + a} = \frac{Y_1(s)}{X(s)}$$

$$G_b(s) = \frac{rs + q}{s^2 + bs + c} = \frac{Y_2(s)}{X(s)}$$

$$Y_1(s) + Y_2(s) = Y(s) \tag{4.6}$$

and a, b, c, k, ℓ, r, and q are real constants.
Since

$$Y_1(s) = G_a(s)\ X(s)\ ;\quad Y_2(s) = G_b(s)\ X(s) \tag{4.7}$$

in the time domain

$$\dot{y}_1(t) + ay_1(t) = \ell x(s)\ ;\quad \ddot{y}_2(t) + b\dot{y}_2(t) + cy_2(t) = r\dot{x}(t) + qx(t)$$

or

$$d\dot{y}_1 = \ell dx - a\ dy_1 \tag{4.8}$$

and

$$\dot{y}_2(t) + by_2(t) + c\int y_2(t)\ dt = rx(t) + q\int x(t)\ dt$$

Then differentiating once, we get

$$d\dot{y}_2 = -b\ dy_2 + r\ dx + (qx\ dt - cy_2\ dt) \tag{4.9}$$

The overall output $y(t) = y_1(t)+y_2(t)$ can be found and its structure can be illustrated, as in Fig. 18.

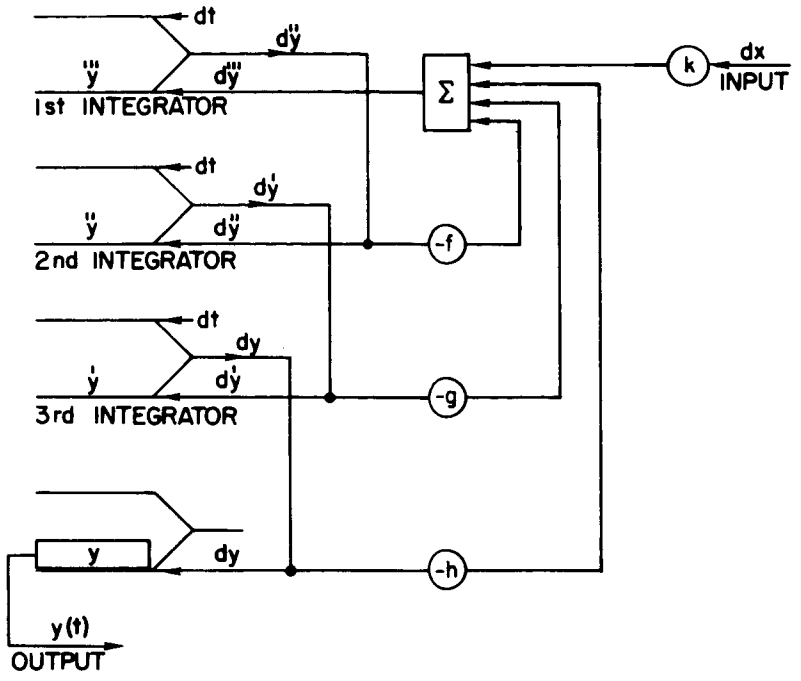


Fig. 17 SERIAL CONNECTION OF THE TRANSFER FUNCTION $G(s) = k/(s^3 + fs^2 + gs + h)$

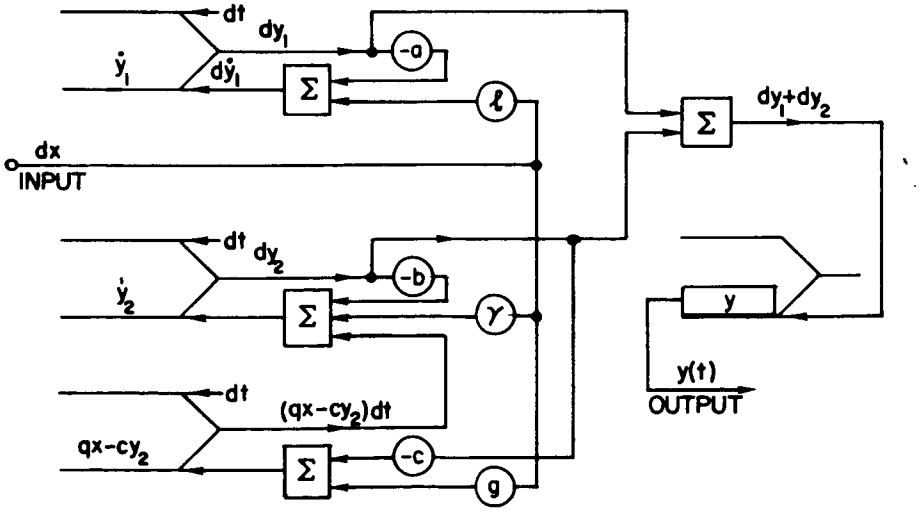


Fig. 18 PARALLEL REALIZATION OF $G(s) = k/(s^3 + fs^2 + gs + h)$

Note: A rational polynomial of n-th order can always be factorized into a product of factors of the first and second order rational polynomials. Therefore, the transfer function of any order can be realized by Method 2 in a parallel fashion.

Comparison of Methods 1 and 2

It has been pointed out in Method 1 that the interconnection of digital integrators is done in serial fashion. During the first few clock pulses of time (dt) only the first integrator starts to update its content. But before any overflow of the first integrator occurs, the second and third integrators remain unchanged. Therefore, the input of the first integrator will not be modified by the outputs of the second and third integrators until some later time. Since the transfer functions of digital filters have different orders of numerators and denominators, the interconnection will be different from case to case. Unless each digital integrator can be integrated on a single chip, there will be too many different units to be handled.

As for Method 2, the advantages are threefold: they are accuracy, simplicity, and economy. Since the interconnection of the integrators is in parallel, the integrators are updating their contents at the same time, it results in somewhat better accuracy. The configuration of digital filters of any order can be broken down to a combination of first and second order configurations. Therefore, the simple standard configurations of first and second order functions can be mass produced in integrated circuit module and can be for all transfer functions.

Proposed Standard Configuration

Since any high order rational polynomials can be factorized into products of lower order polynomials, it is convenient to propose a standard configuration for a transfer function whose order n is less than or equal to three. However, if n should be greater than three the network can be a combination of more packages of $n \leq 3$.

In general, a third order transfer function can be written as

$$G_3(s) = f + \frac{k}{s + h} + \frac{es + c}{s^2 + as + b} \qquad (4.10)$$

The proposed standard configuration of $G_3(s)$ is shown in Fig. 19. It is seen from the figure that the following cases can be obtained.

1. If line 2 and 3 are left open, then Eq. (4.10) becomes

$$G_3(s) = \frac{k}{s + h}$$

2. If line 2 is left open, then

$$G_3(s) = f + \frac{k}{s + h}$$

3. If line 1 and line 3 are left open, then

$$G_3(s) = \frac{es + c}{s^2 + as + b}$$

4. If line 1 is left open, then

$$G_3(s) = f + \frac{es + c}{s^2 + as + b}$$

5. If line 3 is left open, then

$$G_3(s) = \frac{k}{s + h} + \frac{es + c}{s^2 + as + b}$$

Therefore, the configuration shown really works for $n \leq 3$. To increase the flexibility, the real constants a, b, c, d, e, f, h, and k (see constant multiplier, Fig. 6) can be set from outside, so that one may choose whatever constants he wishes. This can be done either by setting the constants serially from the outside before the beginning of the operating cycle, which will remain constant thereafter; on the other hand, if each register of the constants can be accessed from outside they can be set easily to the desired constant. If any one or more of these constants are negative, it is very easy to change the leads around.

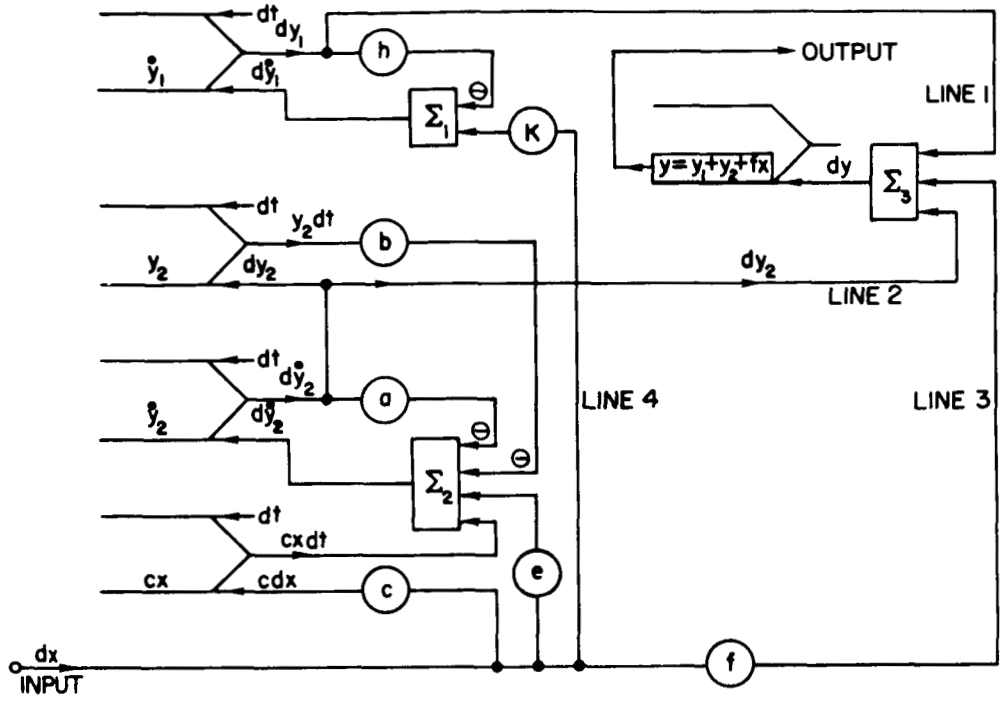


Fig. 19 PROPOSED CONFIGURATION FOR A TRANSFER FUNCTION OF ORDER ≤ 3 $G_3(s) = f + k/(s + h) + (es + c)/(s^2 + as + b)$

In case of necessary, it is also possible to divide the standard configuration into a few sections such that the packaging could be made easier.

It is seen from Fig. 19 that if the configuration shown can be integrated on one chip, there will be some unused parts if the transfer function is of the order of two or one. But not much simplification between configurations of the 3rd and 2nd order transfer functions was observed. Another scheme is also possible; that is, break line 1 and line 4, such that the transfer function of order three is a combination of 1st and 2nd order transfer functions, whereby each one of them can be independently operated.

The following examples are used to illustrate the design and implementation of digital filters using both methods 1 and 2.

Example 6: Design a low pass filter, using the Butterworth transfer function

$$\left|G(jw)\right|^2 = \frac{1}{1 + w^{2n}}$$

to have a magnitude characteristic such that at a frequency three times the cutoff frequency, the magnitude is at least 25 dB down from its value at zero frequency.

Solution: First, let us find the required value of n:

$$\left.\frac{1}{1 + w^{2n}}\right|_{w=3} = 10^{-2.5} \qquad \text{or} \quad \log_{10}(1 + 9^n) = 2.5$$

$$9^n \approx 316 \qquad \text{or} \quad n \approx \frac{2.5}{0.954} = 2.62$$

Thus the required value is the next larger integer n = 3.

For n = 3, the third order Butterworth polynomial is

$$B_3(s) = s^3 + 2s^2 + 2s + 1$$

hence the transfer function G(s) can be obtained as

$$G(s) = \frac{1}{s^3 + 2s^2 + 2s + 1} = \frac{Y(s)}{X(s)}$$

Now we try both methods of realization.

Method 1

Transforming G(s) back to the time domain, we have

$$\dddot{y}(t) + 2\ddot{y}(t) + 2\dot{y}(t) + y(t) = x(t)$$

or

$$d\dddot{y} = dx - 2d\ddot{y} - 2d\dot{y} - 2dy$$

The connection of the solution and the realized network diagram is shown in Figs. 20 and 21, respectively. The sinusoidal response and the step response of the network realized are plotted in Figs. 22 and 23.

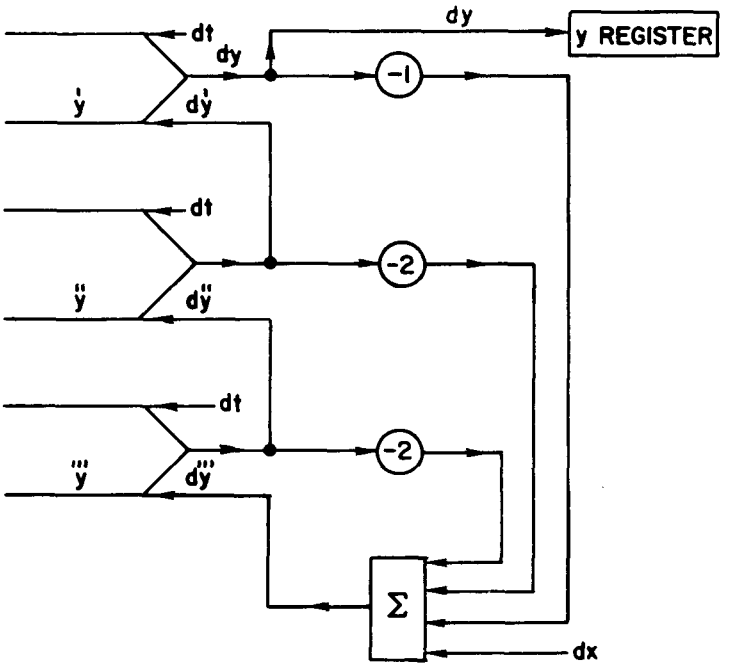


Fig. 20 SOLUTION MAPPING OF METHOD 1

Method 2

$$G(s) = \frac{1}{s^3 + 2s^2 + 2s + 1} = \frac{Y(s)}{X(s)} = \frac{1}{s + 1} - \frac{s}{s^2 + s + 1}$$

$$= G_1(s) - G_2(s)$$

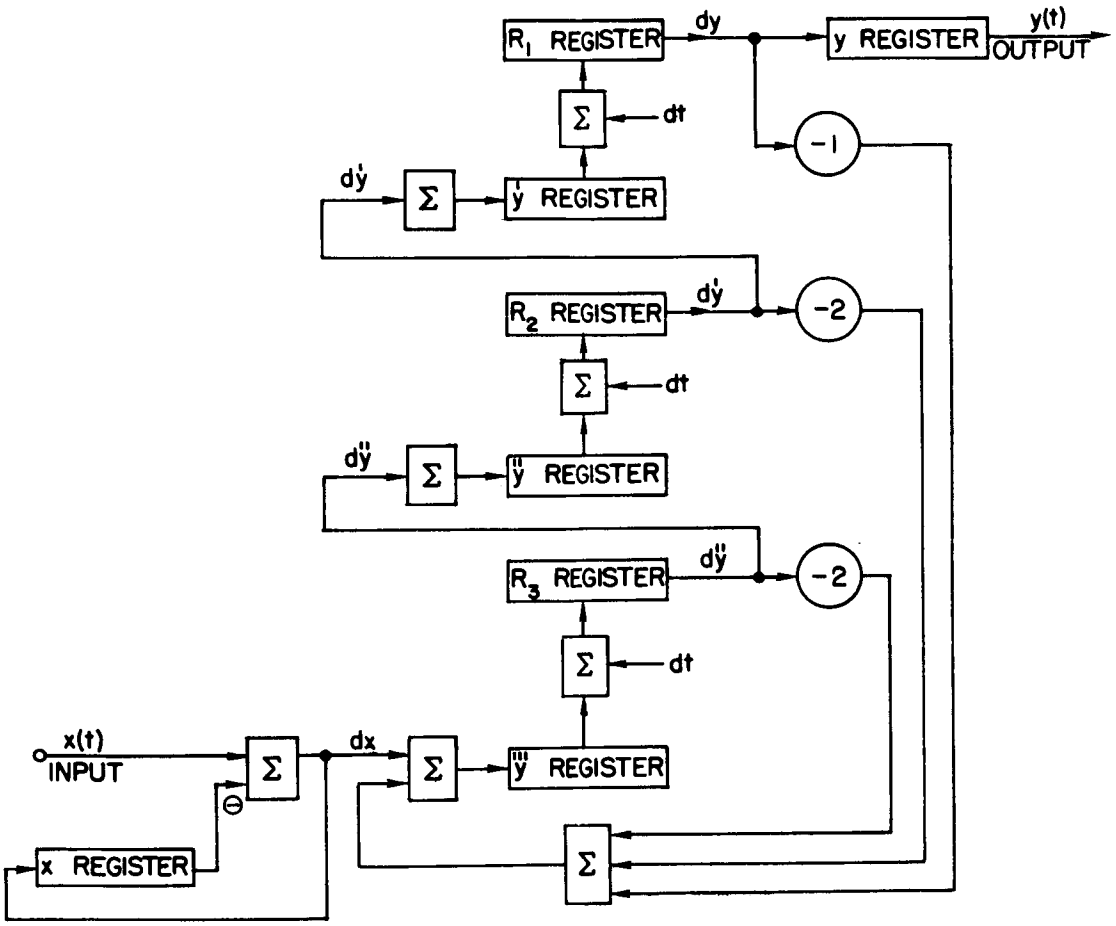


Fig. 21 REALIZED NETWORK OF
$G(s) = 1/(s^3 + 2s^2 + 2s + 1)$ BY METHOD 1

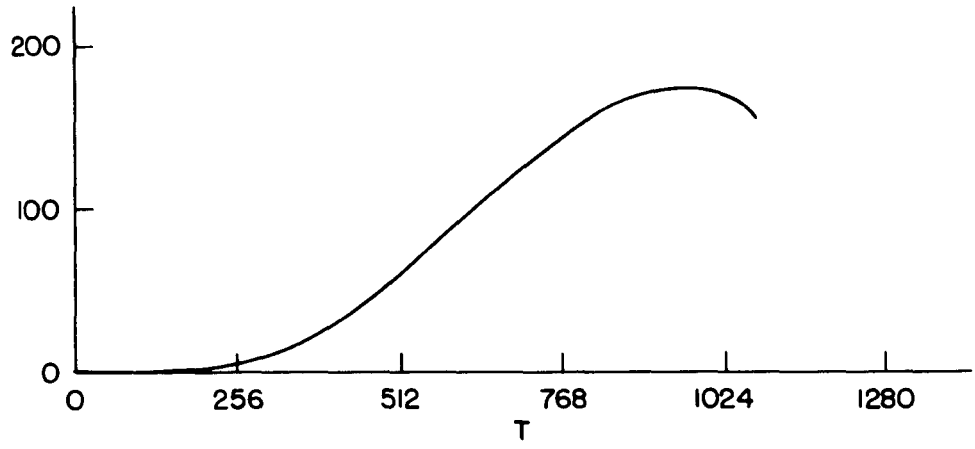


Fig. 22 SINUSOIDAL RESPONSE OF $G(s) = 1/(s^3 + 2s^2 + 2s + 1)$ WITH INPUT $x(t) = \sin(T/256)$

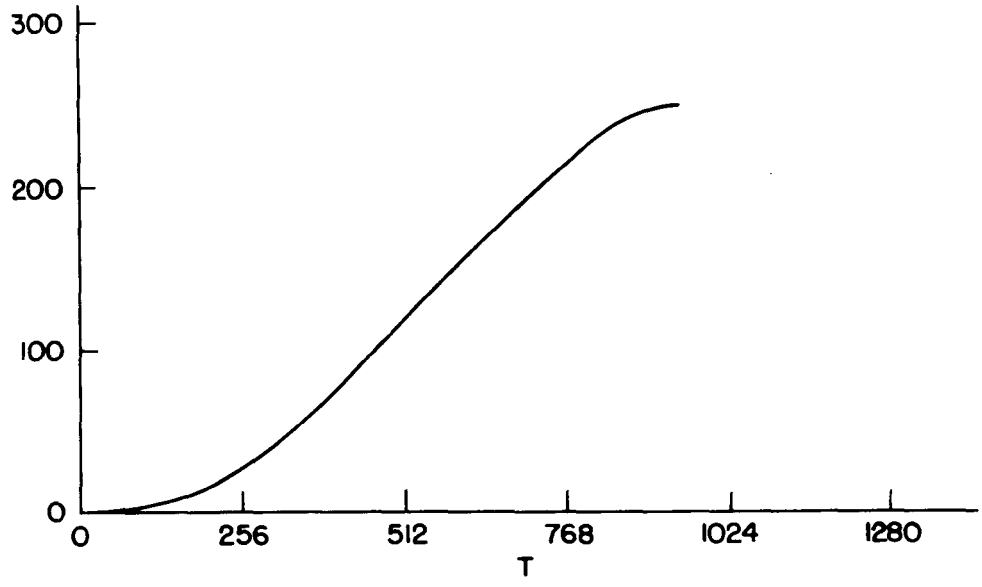


Fig. 23 STEP RESPONSE OF $G(s) = 1/(s^3 + 2s^2 + 2s + 1)$ WITH INPUT $x(t) = 255$

$$G_1(s) = \frac{1}{s + 1} = \frac{Y_1(s)}{X(s)}$$

$$G_2(s) = \frac{1}{s^2 + s + 1} = \frac{Y_2(s)}{X(s)}$$

In the time domain, we have

$$d\dot{y}_1(t) = dx(t) - dy_1(t)$$

and

$$d\dot{y}_2(t) = dx(t) - dy_2(t) - y_2(t)\ dt$$

The output signal $y(t)$ equals $y(t) = y_1(t) - y_2(t)$
The digital filter designed by Method 2 is shown in Fig. 24.

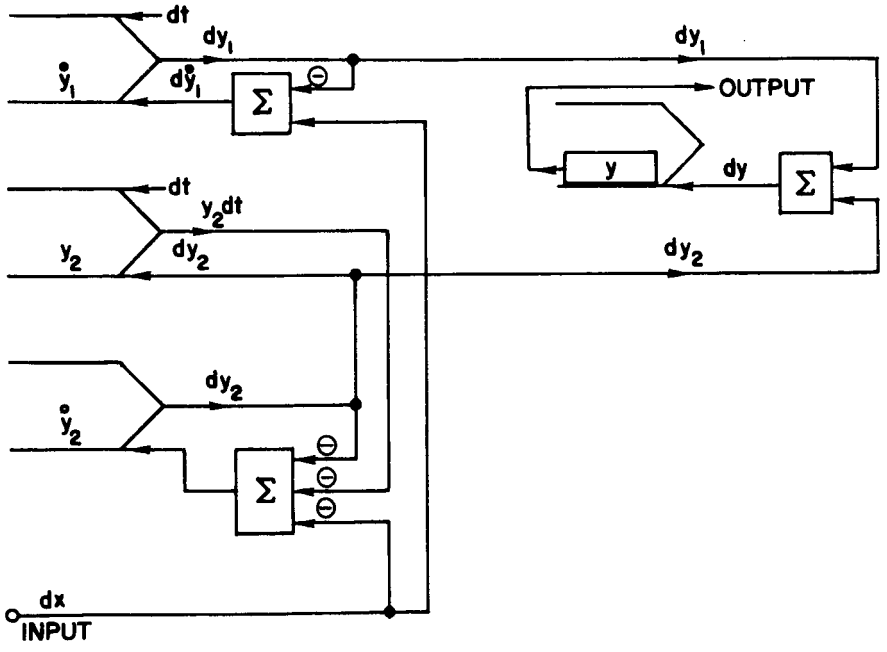


Fig. 24 REALIZATION OF $G(s) = 1/(s + 1) - s/(s^2 + s + 1)$

Example 7: Design a low pass filter utilizing the Chebyshev characteristic

$$|G(jw)|^2 = \frac{1}{1 + \varepsilon^2 C_n(w)}$$

so that

1. The peak-to-peak ripple in the squared magnitude characteristic does not exceed 15 percent of the maximum value.
2. The magnitude response is down at least 50 dB at $w = 4w_c = 4$.

Solution: First, it is necessary to calculate the required value of ε^2. At the trough of the ripple, we have

$$\frac{1}{1 + \varepsilon^2 C_n(1)} = 1 - 0.15 = 0.85 \qquad \text{or} \quad 1 + \varepsilon^2 = \frac{20}{17}$$

hence

$$\varepsilon^2 = 0.175$$

At $w = 4$, we have

$$\frac{1}{1 + \varepsilon^2 C_n(4)} = 10^{-5} \qquad 1 + \varepsilon^2 C_n^2(4) = 10^5$$

or

$$\varepsilon^2 C_n^2(4) \approx 10^5 \qquad \text{implies} \quad C^n(4) = 753$$

In order to find n we have

$$\frac{1}{2}\left[\left(w + \sqrt{w^2 - 1}\right)^n + \left(w + \sqrt{w^2 - 1}\right)^{-n}\right]_{w=4} = 753$$

$$\left(w + \sqrt{w^2 - 1}\right)^n \Big|_{w=4} \approx 1506 \qquad \text{or} \quad n = 3.58$$

Therefore, $n = 4$ will be more than satisfactory.

The location of the poles for the fourth order Chebyshev polynominal can be found as follows:

Pole locations at $S_k = \sigma_k + jw_k$, $k = 1, 2, 3, 4$

$$\sigma_k = \pm \tanh a \sin[(2k - 1)/n](\pi/2)$$

$$w_k = \cos[(2k - 1)/2](\pi/2)$$

where

$$a = \frac{1}{2} \sinh^{-1} \frac{1}{\varepsilon}$$

For the present case, $n = 4$, $\varepsilon^2 = 0.176$

$$a = \frac{1}{4} \sinh^{-1} \frac{1}{0.42} = \frac{1}{4} \sinh^{-1} (2.38) = 0.402$$

$$\tanh a = 0.38$$

Therefore

$$s_1 = -0.144 + j0.924$$

$$s_2 = -0.144 - j0.924$$

$$s_3 = -0.351 + j0.383$$

$$s_4 = -0.351 - j0.383$$

Hence the transfer function $G(s)$ can be obtained

$$G(s) = \frac{Y(s)}{X(s)} = \frac{s_1 s_2 s_3 s_4}{(s - s_1)(s - s_2)(s - s_3)(s - s_4)}$$

$$= \frac{0.111}{(s^2 + 0.288s + 0.416)(s^2 + 0.702s + 0.268)}$$

$$= \frac{0.111}{(s^4 + 0.99s^3 + 0.886s^2 + 0.369s + 0.111)}$$

Method 1 Transforming G(s) back to the time domain

$$\dddot{y}(t) + 0.99\ddot{y}(t) + 0.886\dot{y}(t) + 0.369y(t) + 0.111y(t) = 0.111\ x(t)$$

or

$$d\ddot{y} = 0.111dx - 0.99d\ddot{y} - 0.886d\dot{y} - 0.369dy - 0.111dy$$

The network realized corresponding to the above equation is shown in Fig. 25.

Method 2

$$G(s) = -\frac{0.414s + 0.268}{s^2 + 0.288s + 0.416} + \frac{0.414s + 0.439}{s^2 + 0.702s + 0.268}$$

$$= -G_1(s) + G_2(s)$$

where

$$G_1(s) = \frac{0.414s + 0.268}{s^2 + 0.288s + 0.416} = \frac{Y_1(s)}{X(s)}$$

$$G_2(s) = \frac{0.414s + 0.439}{s^2 + 0.702s + 0.268} = \frac{Y_2(s)}{X(s)}$$

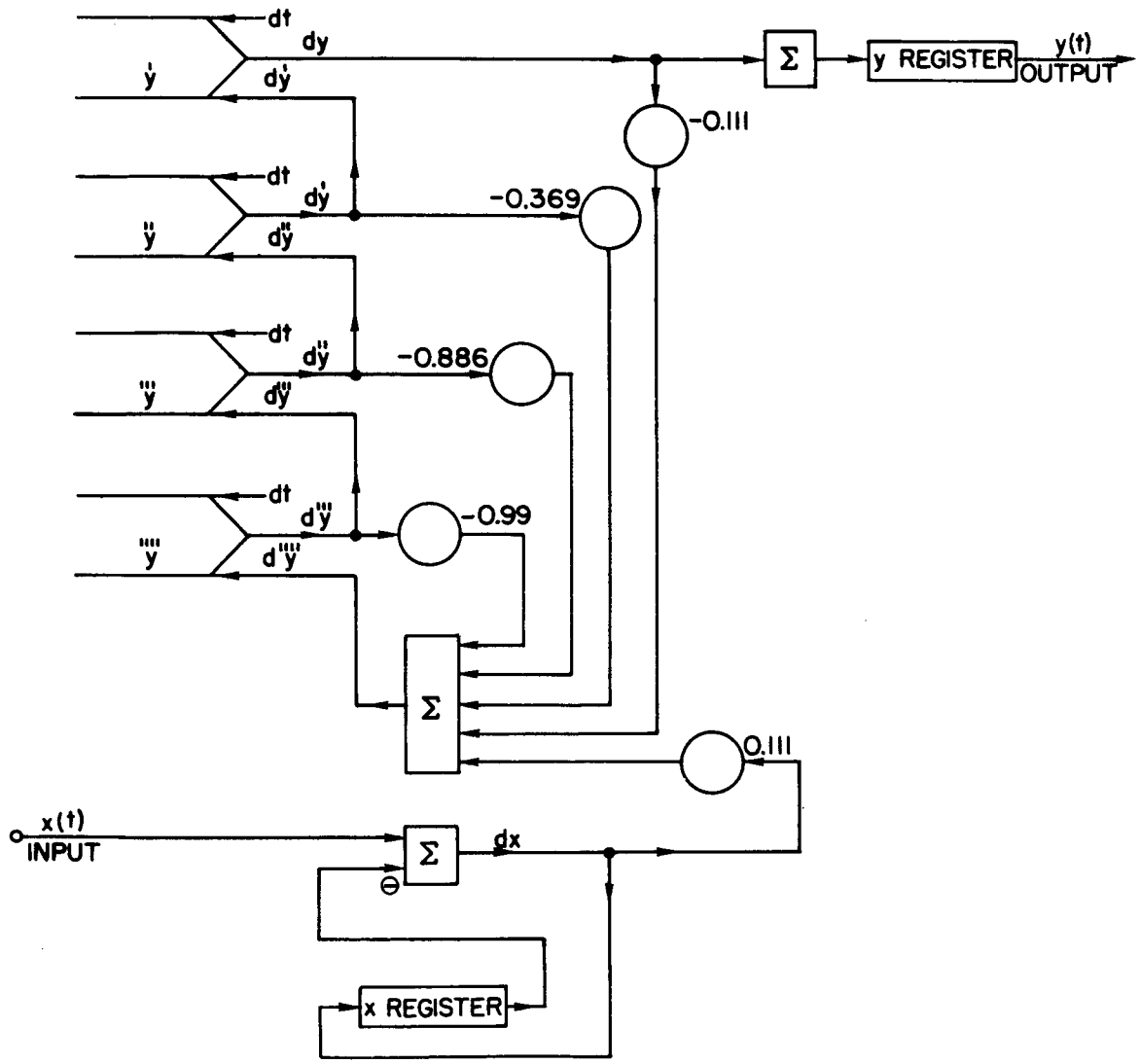


Fig. 25 REALIZATION OF
$G(s) = 0.111/(s^4 + 0.99s^3 + 0.886s^2 + 0.369s + 0.111)$
BY METHOD 1

Transforming $G_1(s)$ and $G_2(s)$ back to the time domain, we have

$$\ddot{y}_1(t) + 0.288\dot{y}_1(t) + 0.416y_1(t) = 0.414\dot{x}(t) + 0.268x(t)$$

and

$$\ddot{y}_2(t) + 0.702\dot{y}_2(t) + 0.268y_2(t) = 0.414\dot{x}(t) + 0.439x(t)$$

or

$$d\dot{y}_1 = 0.414dx - 0.288dy_1 + 0.268xdt - 0.416y_1dt$$

and

$$d\dot{y}_2 = 0.414dx - 0.702dy_2 + 0.439xdt - 0.268y_2dt$$

and the overall output $y(t)$ is

$$y(t) = y_2(t) - y_1(t)$$

The network realized by Method 2 is shown in Fig. 26.

3. Digital Spectrum Analyzer

The spectrum analyzer is a device used to measure the distribution of energy at different frequencies of interest. One way of measuring the frequency spectrum is to measure the energy in the passbands of a bank of narrowband filters. The digital spectrum analyzer we are dealing with is realized by a bank of narrowband digital filters, each with a fixed bandwidth spanning the entire frequency range of interest.

An example of designing a bank of bandpass filters is shown in the following:

Example 8: It is desired to design a bank of bandpass filters with a common input, each 400 cycles per second wide covering the band 300 Hz to 3100 Hz. The contiguous filters are required to cross at -3 dB of the midband gain. Use third order maximally flat approximation.

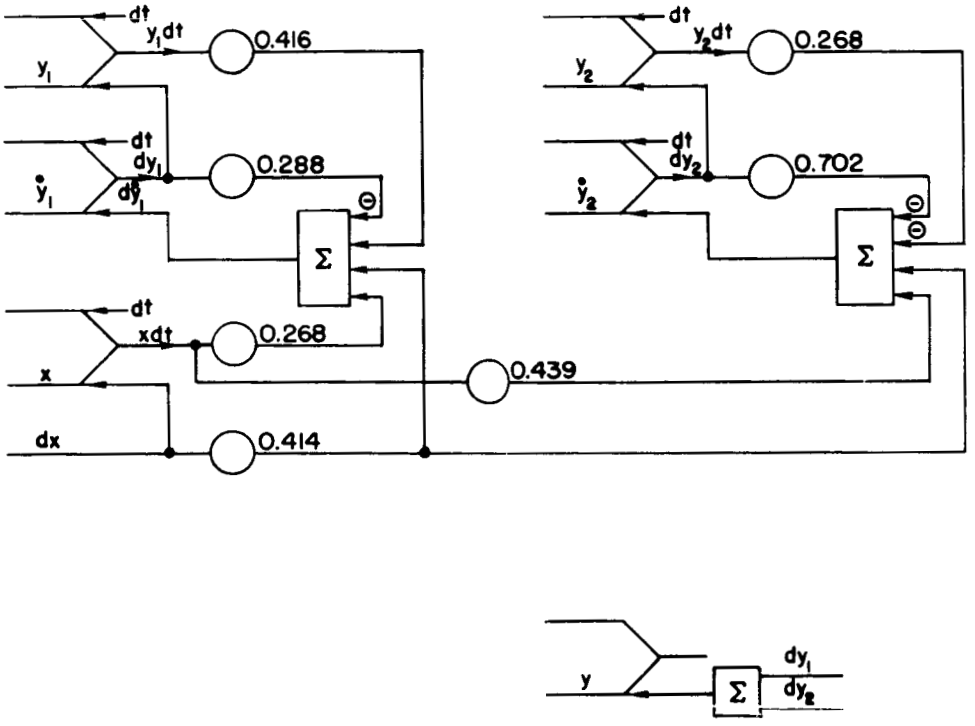


Fig. 26 REALIZATION OF
$G(s) = 0.111/(s^4 + 0.99s^3 + 0.886s^2 + 0.369s + 0.111)$
USING METHOD 2

Solution: The transfer function of each bandpass filter can be found from the frequency transformation of a low pass filter. For a third order maximally flat low pass filter, the transfer function is

$$G_L(s) = \frac{1}{s^3 + 2s^2 + 2s + 1}$$

Let us make the following tranformation:

$$s = \frac{w_o}{B}\left(\frac{p}{w_o} + \frac{w_o}{p}\right) \qquad (4.11)$$

then

$$G_B(s) = \frac{B^3 p^3}{p^6+2Bp^5+(3w_o^2+2B^2)p^4+(B^3+4Bw_o^2)p^3+(3w_o^4+2B^2w_o^2)p^2+2Bw_o^4p+w_o^6} \qquad (4.12)$$

where

B = bandwidth of the bandpass filter

w_o = geometrical mean of the cutoff frequencies of the bandpass filter = $\sqrt{w_1 w_2}$

w_1 = lower cutoff frequency of the bandpass filter

w_2 = higher cutoff frequency of the bandpass filter

p = complex frequency

With the given transformation formula, the first bandpass filter, from 300 Hz to 700 Hz, can be designed first:

$w_o = \sqrt{300 \times 700}$ Hz = 458.258 Hz

B = 400 Hz

Normalizing $B_n = 1$, and $w_{o1} = 458.258/400 = 1.1456$, the transfer function of the first bandpass filter is

$$G_1(p) = \frac{p^3}{p^6 + 2p^5 + 5.937p^4 + 6.25p^3 + 7.79p^2 + 3.446p + 3.26} \qquad (4.13)$$

This transfer function can be realized by Method 1 of section IV.2 without much difficulty. But we will try to realize it in parallel fashion. There is no doubt that theoretically the denominator of $G_1(p)$ can be factorized as $(p^2+ap+b)(p^2+cp+d)(p^2+ep+f)$. But it is very difficult to factor it as the order of the polynomials increases. From another point of view, since we know the pole locations of the transfer function, we might factorize the denominator directly from the transfer function of the lowpass filter:

$$G_L(s) = \frac{1}{s^3 + 2s^2 + 2s + 1} = \frac{1}{s+1} - \frac{s}{s^2 + s + 1}$$

$$= \frac{1}{s+1} - \frac{s}{(s + 0.5 + j0.866)(s + 0.5 - j0.866)}$$

Now, we can make the transformation

$$s = \frac{w_o}{B}\left(\frac{p}{w_o} + \frac{w_o}{p}\right) = \frac{1}{B}\left(p + \frac{w_o^2}{p}\right)$$

then

$$G_B(p) = \frac{Bp}{p^2+p+w_o^2} - \frac{p^3 + w_o^2 p}{\left[p^2+(0.5+j0.866)Bp+w_o^2\right]\left[p^2+(0.5-j0.866)Bp+w_o^2\right]} \tag{4.14}$$

$$G_B(p) = \frac{1}{\frac{1}{B}\left(p + \frac{w_o^2}{p}\right)+ 1} - \frac{\frac{1}{B}\left(p + \frac{w_o^2}{p}\right)}{\left[\frac{1}{B}\left(p + \frac{w_o^2}{p}\right) + 0.5 + j0.866\right]\left[\frac{1}{B}\left(p + \frac{w_o^2}{p}\right) + 0.5 - j0.866\right]}$$

The poles of the second term can be found as follows:

$$p^2 + (0.5 + j0.866)Bp + w_o^2 = 0$$

$$p = \frac{1}{2}\left[-(0.5 + j0.866)B \pm \sqrt{(0.5 + j0.866)^2 B^2 - 4w_o^2}\right]$$

$$= \frac{1}{2}\left[-(0.5 + j0.866)B \pm (\sqrt{u} + j\sqrt{v})\right] = \frac{1}{2}\left[(-0.5B \pm \sqrt{u}) - j(0.866\,B \mp \sqrt{v})\right]$$

where

$$u - v = B^2 - 4w_o^2$$

$$uv = 3B^2/16 \quad \text{and}$$

$$u + v = + \sqrt{(u - v)^2 + 4uv}$$

$$p_1 = (-0.25B + 0.5\sqrt{u}) - j(0.433 - 0.5\sqrt{v})$$

$$p_2 = (-0.25B - 0.5\sqrt{u}) - j(0.433 + 0.5\sqrt{v}) \qquad (4.15)$$

The other two poles can be found by solving

$$p^2 + (0.5 - j0.866)Bp + w_o^2 = 0$$

or

$$p_3 = (-0.25B + 0.5\sqrt{u}) + j(0.433 - 0.5\sqrt{v}) \qquad (4.16)$$

$$p_4 = (-0.25B - 0.5\sqrt{u}) + j(0.433 + 0.5\sqrt{v})$$

Then $G_B(p)$ can be written in partial fraction form as

$$G_B(p) = \frac{Bp}{p^2 + p + w_o^2} - \frac{a_1p + b_1}{(p - p_1)(p - p_3)} - \frac{a_2p + b_2}{(p - p_2)(p - p_4)} \qquad (4.17)$$

where a_1, a_2, b_1 and b_2 are real constants. They can be found in terms of B and w_o from Eq. (4.14). For our present design problem, $B = 1$, $w_{o1} = 1.1456$, and $u = 0.0448$, $v = 4.2948$. Substituting these values into (4.15) and (4.16) we get $p_1, p_3 = -0.144 \pm j0.603$; $p_2, p_4 = -0.356 \mp j1.469$. Substituting them into Eq. (4.17), we obtain

$$G_1(p) = \frac{p}{p^2 + p + 1.312} - \frac{0.545p + 0.105}{p^2 + 0.288p + 0.384} - \frac{0.455p - 0.624}{p^2 + 0.712p + 2.284} \qquad (4.18)$$

There is no doubt that the transfer function $G_1(p)$ of the bandpass filter can be realized by parallel connection of three building blocks, each block containing digital integrators such as those discussed in the previous section.

Similarly, the next bandpass filter of frequency range 700 to 1100 Hz can be designed with $w_{02}^2 = (700 \times 1100)/(400 \times 400) = 4.813$; $B_n = 1$, $u \approx 0$, and $v = 18.25$; therefore,

$$p_1, p_3 = -0.25 \pm j1.704$$

$$p_2, p_4 = -0.25 \mp j\ 2.57$$

and

$$G_2(p) = \frac{p}{p^2 + p + 4.813} - \frac{0.498p + 0.351}{p^2 + 0.5p + 2.96} - \frac{0.502p - 0.851}{p^2 + 0.5p + 7.18} \tag{4.19}$$

The realization of $G_2(p)$ is similar to $G_1(p)$. The realized networks of $G_1(p)$ are shown in Figs. 27 and 28. By using the same methods, $G_2(p)$, $G_3(p)$, $G_4(p)$, ..., $G_7(p)$ can be designed by simply changing the contents of the constant registers, as in Fig. 29. For each $G_i(p)$, three chips of the proposed configurations are used. For a bank of digital filters, as in the present example, twenty-one chips will be used and no doubt they can be mounted on a small card.

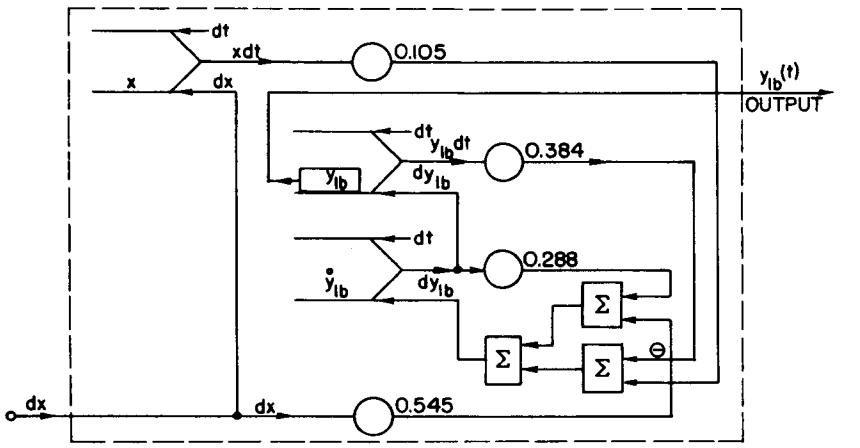
It is seen from Eq. (4.17) that if we factorize the sixth order polynominal with a small error produced at one pole location, the remaining five pole locations will be affected. In other words, a small coefficient perturbation (or truncation) may result in a large shift in root location. On the other hand, the parallel two-pole filter combination will result in a better performance, since the coefficient perturbation of the first term will not change the pole position in the second and/or the third terms. Therefore, it is advisable to use the parallel combination in the design.

$$G_{1b}(p) = \frac{0.545p + 0.105}{p^2 + 0.288p + 0.384}$$

or

$$d\dot{y}_{1b} = 0.105xdt - 0.384y_{1b}dt + 0.545dx - 0.288dy_{1b}$$

$G_{1b}(p)$ can be realized as shown



$$G_{1c}(p) = \frac{0.455p - 0.624}{p^2 + 0.712p + 2.284}$$

or

$$d\dot{y}_{1c} = 0.455dx - 0.712dy_{1c} - 0.624xdt - 2.284y_{1c}dt$$

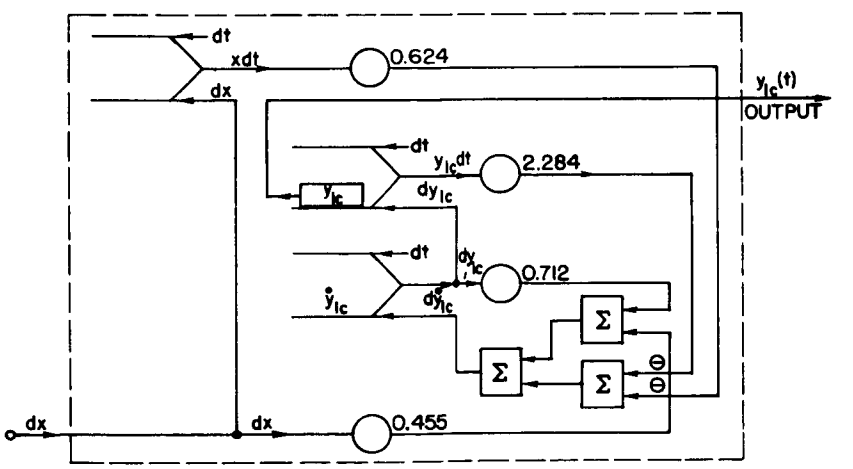
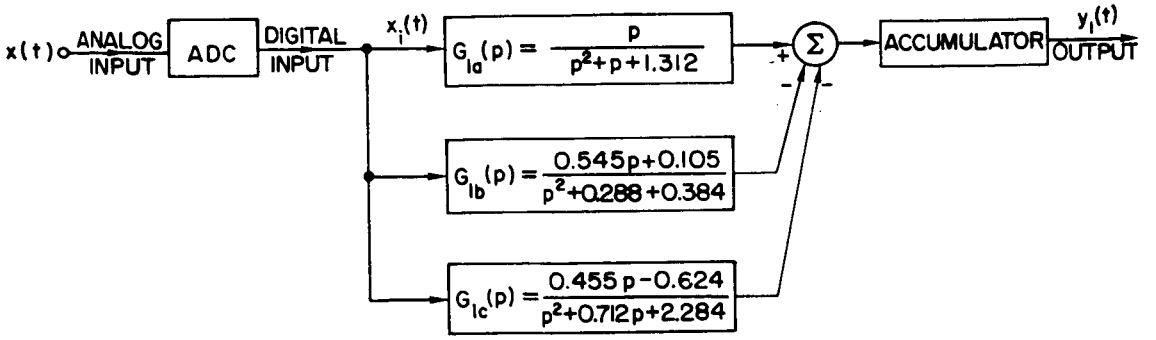


Fig. 28 REALIZATION OF $G_{1b}(p)$ and $G_{1c}(p)$



where

$$G_{1a}(p) = \frac{p}{p^2 + p + 1.312}$$

or

$$d\dot{y}_{1a} = dx - dy_{1a} - 1.312y_{1a}dt$$

can be realized as

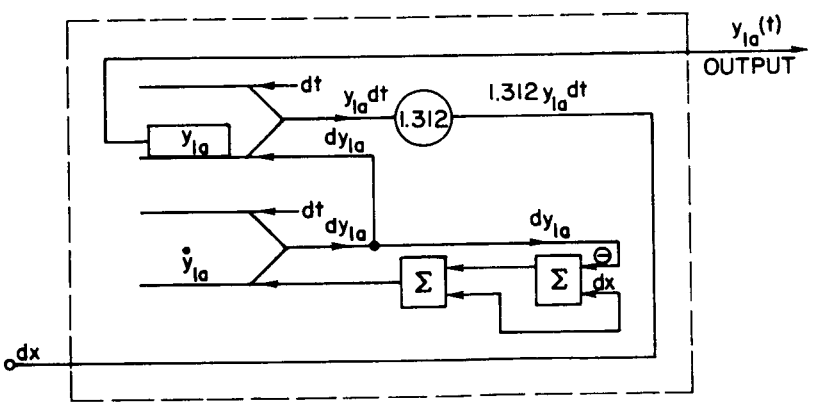


Fig. 27 REALIZATION OF $G_{1a}(p)$

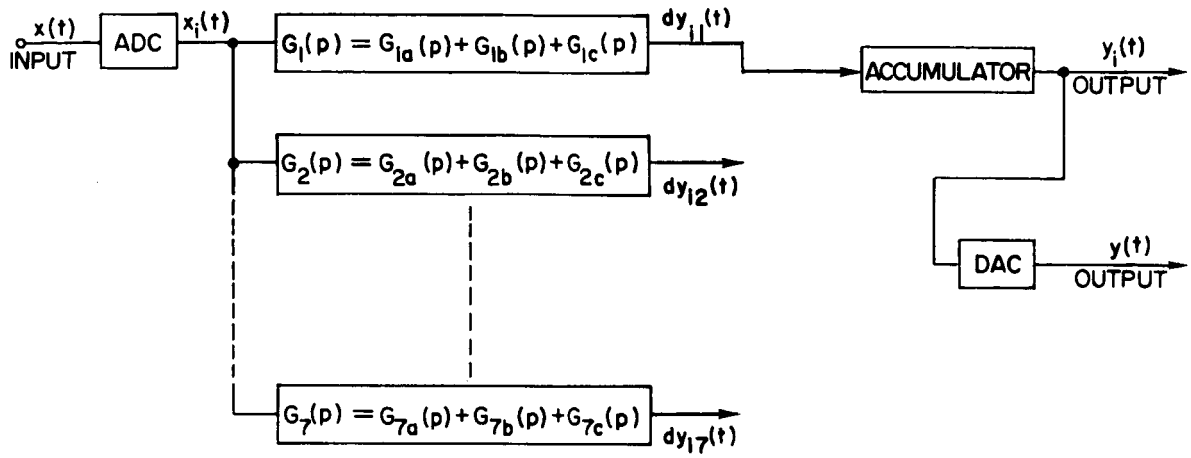


Fig. 29 REALIZATION OF A BANK OF DIGITAL FILTERS

4. Digital Filters with Time Varying Coefficients

It is possible to design a digital filter with time varying coefficients. Since the transfer function of this kind of filter is not defined as are those in section IV.I, a time domain synthesis will be discussed. The following example will illustrate the realizability of this kind of filter.

Example 9: Suppose x(t) and y(t) are the input and output of a digital filter in such a way that the following relation is satisfied:

$$\dddot{y}(t) + a(t)\ddot{y}(t) + b(t)\dot{y}(t) + c(t)y(t) = c(t)x(t)$$

Design such a network.

Solution: It is seen that if a(t), b(t), and c(t) are constants, the problem can be reduced to one of the typical digital filter design problems. Now, with time varying coefficients, it is necessary first to differentiate the equation,

$$d\ddot{y} = d(cx - cy) - d(a\ddot{y}) - d(b\dot{y})$$

then use digital integrators to generate the terms $d(a\ddot{y})$, $d(b\dot{y})$, $d(cy)$, and $d(cx)$. Note that

$$d(a\ddot{y}) = ad\ddot{y} + \ddot{y}da$$

which can be generated by integrators, as shown in Fig. 30, where $d\ddot{y}$ is an output of an integrator and da is the differential of the input $a(t)$, hence it is controllable from the outside. By using more integrators, the given problem can be designed as in Fig. 31.

The author has not yet investigated the advantages or any application of digital filters with non-constant coefficients, but an interesting case of this particular type of filter will be discussed.

As we know, the location of the poles of the Chebyshev filters differ only slightly from those of the Butterworth filters [9]. Therefore, if the time varying coefficients are changing in such a way that the poles are shifting horizontally (in the s-plane) from the Butterworth pole locus to the Chebyshev pole locus, the digital filter will have a changing magnitude square characteristic from the maximum flat response to different equal ripple responses.

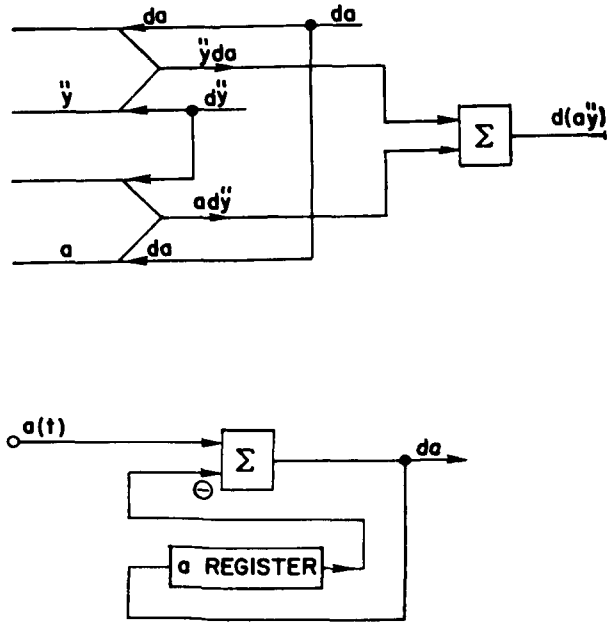


Fig. 30 GENERATION OF $d(a\ddot{y})$

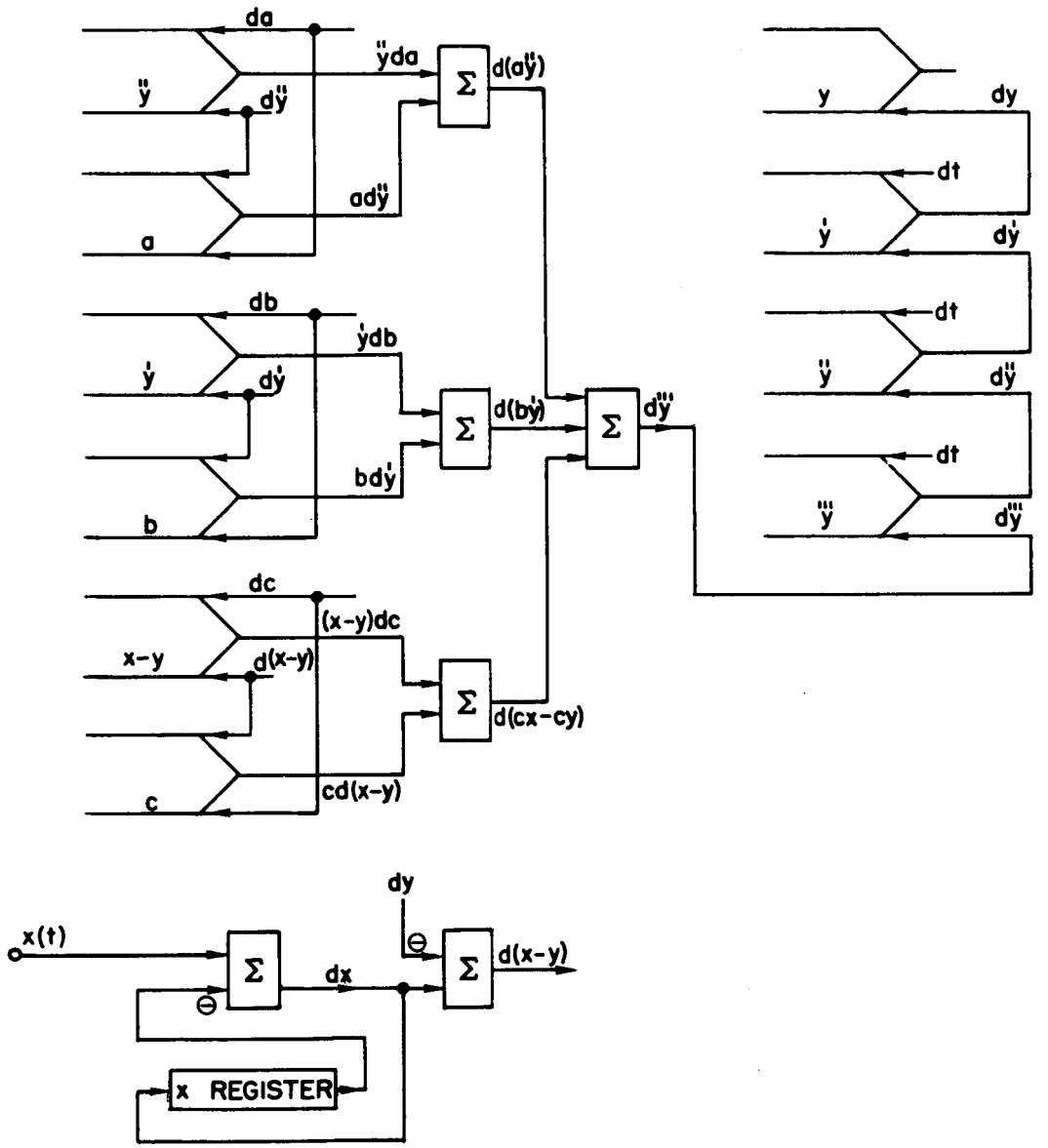


Fig. 31 REALIZATION OF DIGITAL FILTER WITH TIME VARYING COEFFICIENTS

PRECEDING PAGE BLANK NOT FILMED.

Chapter V

DRIVING POINT IMMITTANCE FUNCTION REALIZATION BY USING DIGITAL INTEGRATORS

1. Driving Point Impedance Function Realization

The driving point impedance function $Z_1(s)$ of a network is defined as the ratio of $V_{in}(s)$ to $I_{in}(s)$, where V_{in} and I_{in} are, respectively, the input voltage and input current in Laplance transform form; namely, $Z_1(s) = V_{in}(s)/I_{in}(s)$.

Consider a one-port network as shown in Fig. 32 with an internal current transfer function $G_I(s)$ defined as

$$G_I(s) = \frac{I_2(s)}{I_1(s)} \tag{5.1}$$

where $I_1(s)$ and $I_2(s)$ are the input and output currents, respectively, of the network function $G_I(s)$.
Thus

$$Z_1(s) = \frac{V_{in}(s)}{I_{in}(s)} = \frac{V_{in}(s)}{I_1(s) - I_2(s)}$$

$$= \frac{V_{in}(s)}{I_1(s)\left[1 - G_I(s)\right]} \tag{5.2}$$

Solving for $G_I(s)$, we get

$$G_I(s) = 1 - \frac{V_{in}(s)}{I_1(s)Z_1(s)} \tag{5.3}$$

if we make

$$R_1 I_1(s) = V_{in}(s) \tag{5.4}$$

then

$$G_I(s) = 1 - \frac{R_1}{Z_1(s)} \tag{5.5}$$

where R_1 is a constant (resistance).

From the last two sections, we know that once the transfer function of the network is specified, it can be realized by interconnecting the digital integrators. Therefore, once we specified the desired driving point function, $Z_1(s)$, then $G_I(s)$ can be found. In this way the 1-port network with the desired $Z_1(s)$ can be constructed, as in Fig. 32.

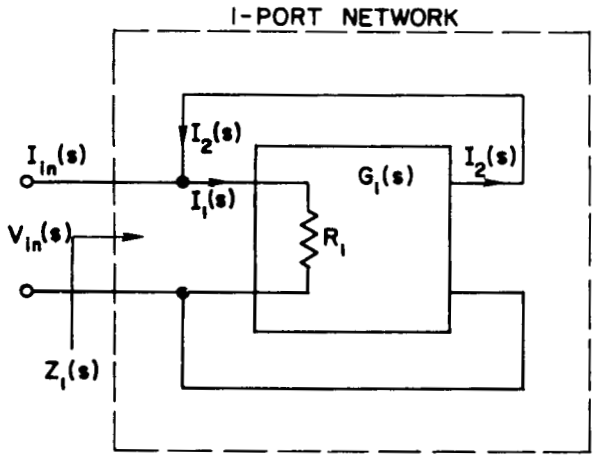


Fig. 32 FEEDBACK CONNECTION USED TO REALIZE THE DRIVING POINT IMPEDANCE FUNCTION

Example 10: Realize the following driving point impedance function by using the prescribed technique

$$Z_1(s) = \frac{s^2 + s + 2}{2s^2 + s + 1} = \frac{V_{in}(s)}{I_{in}(s)}$$

Solution: First, the transfer function $G_I(s)$ has to be found. Using Eq. (5.5), we have

$$G_I(s) = 1 - \frac{R_1}{Z_1(s)} = \frac{s^2(1 - 2R_1) + s(1 - R_1) + (2 - R_1)}{s^2 + s + 2}$$

$$= \frac{I_2(s)}{I_1(s)} = \frac{R_1 I_2(s)}{V_{in}(s)}$$

Let $Y(s) = R_1 I_2(s)$ [note, here $Y(s)$ is not the admittance function] and $X(s) = V_{in}(s)$ then we have

$$(s^2 + s + 2)\, Y(s) = [s^2(1 - 2R_1) + s(1 - R_1) + (2 - R_1)]\, X(s)$$

or

$$\left(1 + \frac{1}{s} + \frac{2}{s^2}\right) Y(s) = \left[(1 - 2R_1) + \frac{(1 - R_1)}{s} + \frac{(2 - R_1)}{s^2}\right] Y(s)$$

Transforming the above expression back to time domain, we have

$$y(t) = (1 - 2R_1)\, x(t) + \int [(1 - R_1)\, x(t) - y(t)]\, dt$$

$$+ \iint [(2 - R_1)\, x(t) - 2y(t)]\, dt^2$$

or

$$dy = (1 - 2R_1)\, dx + [(1 - R_1)\, x - y]\, dt + \int [(2 - R_1)\, x - 2y]\, dt^2$$

The network realized for $G_I(s)$ is shown in Fig. 33. By a connection similar to Fig. 32 the network realization for $Z_1(s)$ is as shown in Fig. 34. The input current to ADC (analog-to-digital converter) has been assumed to be zero.

The equivalent analog networks realized by both Brune's method and Bott and Duffin's method are shown in Figs. 35 and 36 [9].

Note that the method just described needs a controllable current source at the output; this is not easy to get. An alternative way of realizing the driving point impedance function will be shown in the following:

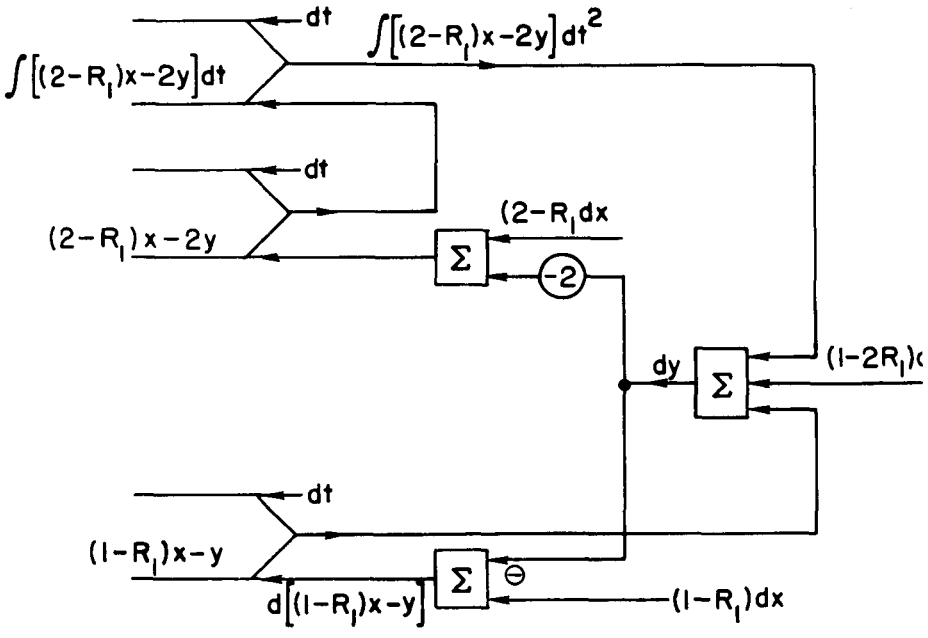


Fig. 33 REALIZATION OF $G(s) = \left[s^2(1-2R_1) + s(1-R_1) + (2-R_1)\right]/(s^2+s+2)$

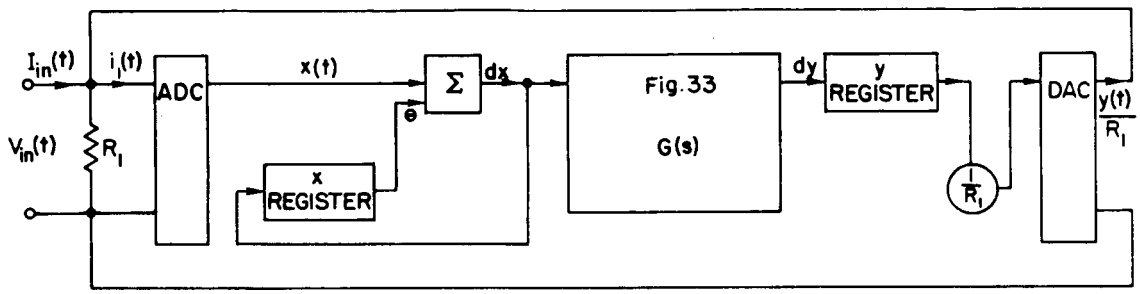


Fig. 34 REALIZATION OF $Z_1(s) = (s^2+s+2)/(2s^2+s+1)$

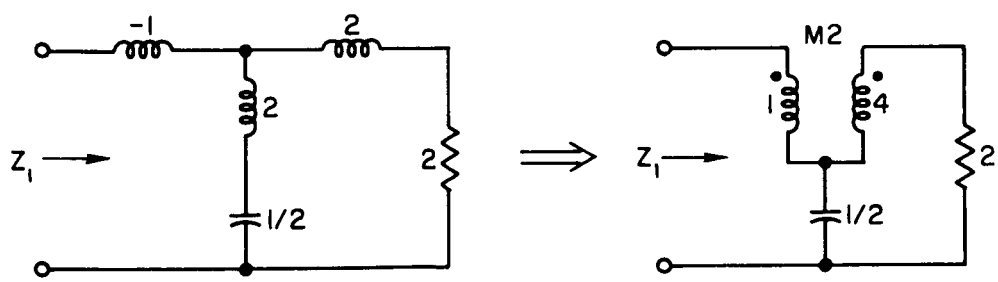


Fig. 35 BRUNE NETWORK REALIZATION OF $Z_1(s) = (s^2+s+2)/(2s^2+s+1)$

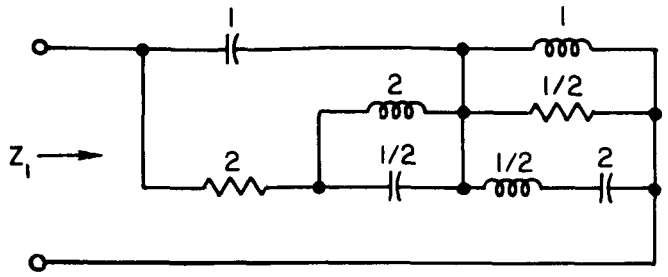


Fig. 36 BOTT AND DUFFIN'S REALIZATION OF $Z_1(s) = (s^2+s+2)/(2s^2+s+1)$

Changing the configuration of Fig. 32 to Fig. 37, let us define the voltage transfer function as

$$G_v(s) = \frac{V_2(s)}{V_{in}(s)} \tag{5.6}$$

and assume that the output voltage has a very small output impedance such that

$$I_2(s) = \frac{V_2(s) - V_{in}(s)}{R_2} \tag{5.7}$$

From Eq. (5.5)

$$G_I(s) = 1 - \frac{R_1}{Z_1(s)} \tag{5.5}$$

then

$$G_I(s) = \frac{I_2(s)}{I_1(s)} = \frac{[V_2(s) - V_{in}(s)]}{V_{in}(s)/R_1} = \frac{R_1}{R_2}\left[\frac{V_2(s)}{V_{in}(s)} - 1\right]$$

$$= \frac{R_1}{R_2}[G_v(s) - 1]$$

$$= 1 - \frac{R_1}{Z_1(s)} \tag{5.8}$$

Solving for $G_v(s)$, we obtain

$$G_v(s) = \frac{R_2}{R_1} + 1 - \frac{R_2}{Z_1(s)} \tag{5.9}$$

With $Z_1(s)$ specified, $G_v(s)$ can be found, and $Z_1(s)$ can be realized without difficulty by interconnecting the digital integrators.

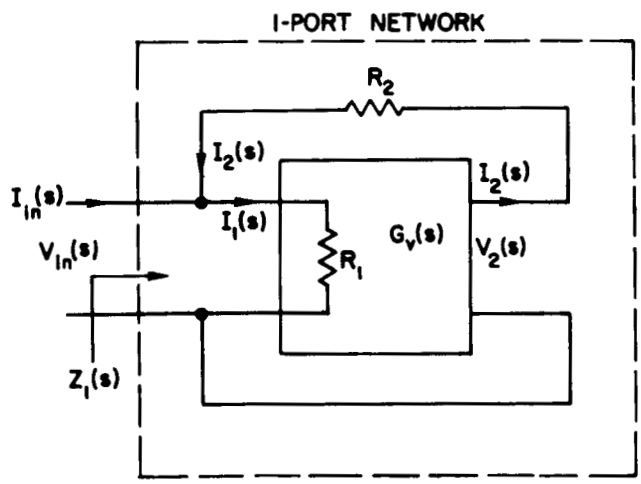


Fig. 37 EQUIVALENT OF FIG. 32

With the method described, the network of Example 10 can also be realized as follows:

$$G_v(s) = \frac{R_2}{R_1} + 1 - \frac{R_2(2s^2 + s + 1)}{s^2 + s + 2}$$

$$= 1 + \frac{R_2}{R_1} - 2R_2 + \frac{R_2 s + R_2}{s^2 + s + 2} = \frac{V_2(s)}{V_{in}(s)} = \frac{Y(s)}{X(s)}$$

$$y(t) = y_1(t) + y_2(t)$$

where

$$y_1(t) = \left(1 + \frac{R_2}{R_1} - 2R_2\right) x(t)$$

$y_2(t)$ = the solution of

$$\dot{dy_2}(t) = R_2\, dx(t) + R_2\, x(t)\, dt - dy_2(t) - 2y_2(t)\, dt\ .$$

The realized network is shown in Fig. 38.

2. Driving Point Admittance Function Realization

The driving point admittance function $Y_1(s)$ of a network is defined as the ratio of $I_{in}(s)$ to $V_{in}(s)$, where $I_{in}(s)$ and $V_{in}(s)$ are, respectively, the input current and input voltage in Laplace transform form; namely,

$$Y_1(s) = \frac{I_{in}(s)}{V_{in}(s)} \tag{5.10}$$

In contrast to the last section, consider a 1-port network, now using the voltage feedback rather than the current feedback, since we are assuming the input as a current controlled source.

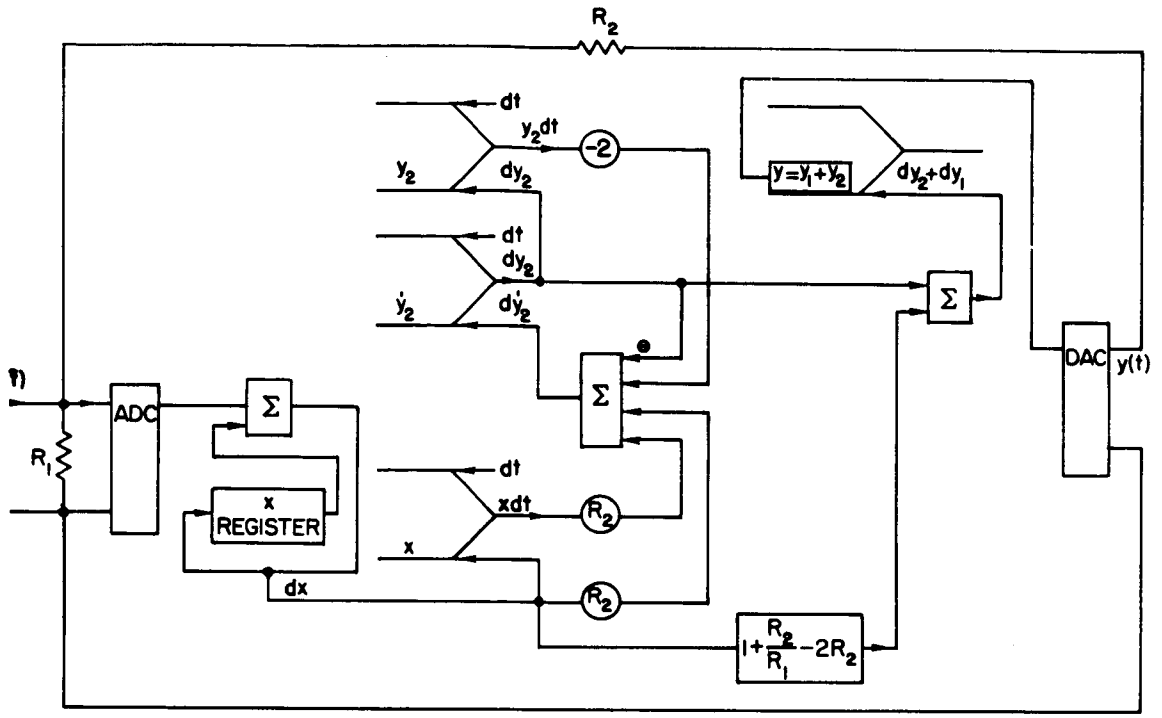


Fig. 38 ALTERNATE REALIZATION OF Z(s)

Referring to Fig. 38, let us define an internal voltage transfer function H(s), such that

$$H(s) = \frac{V_2(s)}{V_1(s)} \tag{5.11}$$

Since $V_1(s) = V_{in}(s) + V_2(s)$, hence

$$Y_1(s) = \frac{I_{in}(s)}{V_{in}(s)} = \frac{I_{in}(s)}{V_1(s) - V_2(s)}$$

$$= \frac{I_{in}(s)}{V_1(s)} [1 - H(s)]$$

or

$$H(s) = 1 - \frac{I_{in}(s)}{V_1(s)\ Y_1(s)} \tag{5.12}$$

If a fixed resistor, R_1' has been connected across terminals 1-2', then

$$\frac{I_{in}(s)}{V_1(s)} = \frac{1}{R_1'} \tag{5.13}$$

hence,

$$H(s) = 1 - \frac{1}{R_1'\ Y_1(s)} \tag{5.14}$$

This is similar to Eq. (5.5). From the method used in the last section, we can easily realize the transfer function $H(s)$ by using digital integrators. With $H(s)$ realized, the desired driving point admittance $Y_1(s)$ can be obtained by connecting $H(s)$, as shown in Fig. 39.

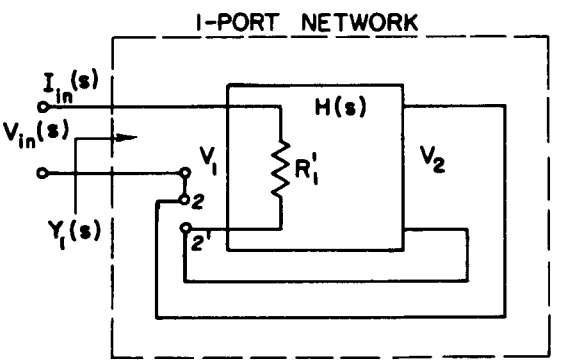


Fig. 39 FEEDBACK CONNECTION USED TO REALIZE DRIVING POINT ADMITTANCE FUNCTION

Example 11: Realize the following driving point admittance function

$$Y_1(s) = \frac{s^2 + s + 2}{2s^2 + s + 1} = \frac{I_{in}(s)}{V_{in}(s)}$$

Solution: First, find the transfer function, H(s), corresponding to the given $Y_1(s)$:

$$H(s) = 1 - \frac{1}{R_1' Y_1(s)} = \frac{(R_1' - 2)s^2 + (R_1' - 1)s + (2R_1' - 1)}{R_1'(s^2 + s + 2)}$$

$$= \frac{V_2(s)}{V_1(s)}$$

Let $V_2(s) = Y(s)$ and $V_1(s) = X(s)$, then

$$R_1'(s^2 + s + 2)\, Y(s) = \left[(R_1' - 2)s^2 + (R_1' - 1)s + (2R_1' - 1)\right] X(s)$$

or

$$\left(1 + \frac{1}{s} + \frac{2}{s^2}\right) Y(s) = \left[\left(1 - \frac{2}{R_1'}\right) + \left(1 - \frac{1}{R_1'}\right)\frac{1}{s} + \left(2 - \frac{1}{R_1'}\right)\frac{1}{s^2}\right] X(s)$$

Transforming back to the time domain, we have

$$y(t) = x(t) + \int \left[\left(1 - \frac{1}{R_1'}\right) x(t) - y(t)\right] dt + \int \left[\left(2 - \frac{1}{R_1'}\right) x(t) - 2y(t)\right] dt^2$$

or

$$dy = dx + \left[\left(1 - \frac{1}{R_1'}\right) x - y\right] dt + \int \left[\left(2 - \frac{1}{R_1'}\right) x - 2y\right] dt^2$$

The realized network of $H(s)$ and $Y_1(s)$ are given in Fig. 40 and Fig. 41, respectively.

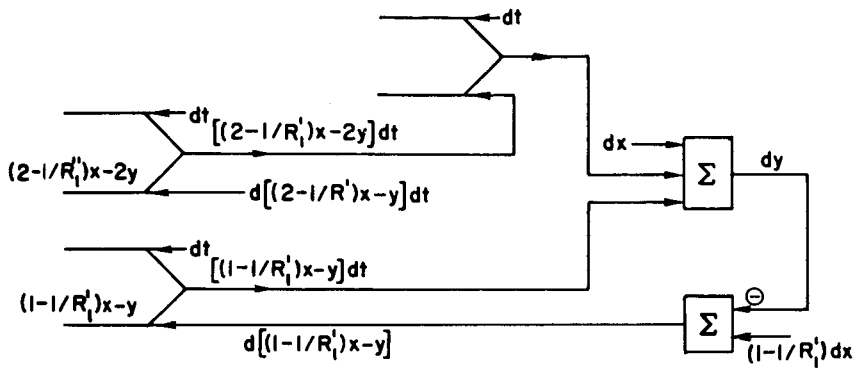


Fig. 40 REALIZATION OF $H(s) = \left[(R_1'-2)s^2 + (R_1'-1)s + (2R_1'-1)\right] / \left[R_1'(s^2+s+2)\right]$

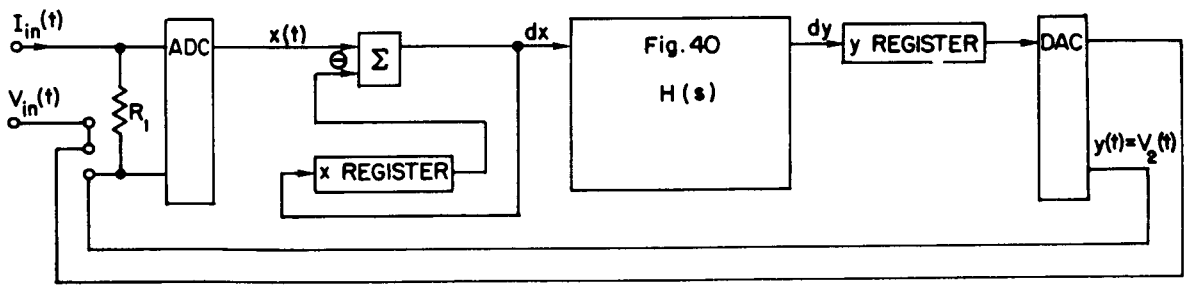


Fig. 41 REALIZATION OF $Y_1(s) = (s^2+s+2)/(2s^2+s+1)$

An interesting result can be drawn from the previous two examples (Ex. 10 and Ex. 11). It is seen that if we normalized the values of R_1 and R_1' or make $R_1 = R_1' = 1$ ohm, then $G(s)$ is exactly the same as $H(s)$. Therefore, the same function can be realized as either an admittance or an impedance function, depending on how the integrators are hooked up.

The following two examples show the realization of a single inductor and single capacitor by using digital building blocks.

Example 12: Realize $Z_1(s) = sL$ by using the above techniques.

Solution: By using Eq. (5.5)

$$G(s) = 1 - \frac{R_1}{Z_1(s)} = 1 - \frac{R_1}{sL} = \frac{sL - R_1}{sL} = \frac{I_2(s)}{I_1(s)} = \frac{R_1 I_2(s)}{V_{in}(s)}$$

Let

$$Y(s) = R_1 I_2(s) , \quad X(s) = V_{in}(s)$$

then

$$Y(s) = \left(1 - \frac{R_1}{sL}\right) X(s)$$

In the time domain, we have

$$y(t) = x(t) - \frac{R_1}{L} \int x(t)\, dt$$

or

$$dy = dx - \frac{R_1}{L} x\, dt$$

The network of $G(s)$ can be realized as in Fig. 42, and the single inductor of value L can be realized as in Fig. 43.

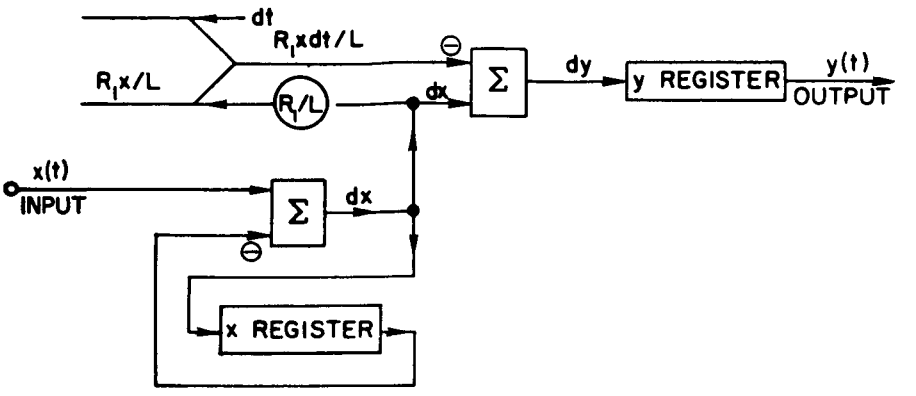


Fig. 42 REALIZATION OF $G(s) = (sL - R_1)/sL$

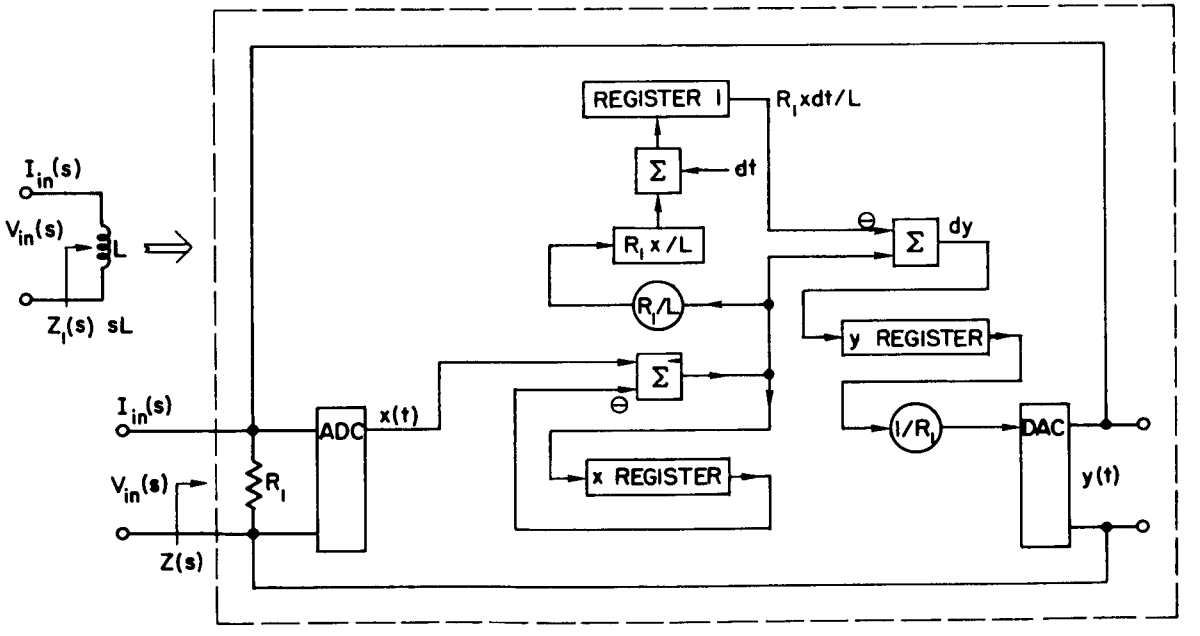


Fig. 43 REALIZATION OF A SINGLE INDUCTOR, $Z_1(s) = sL$

Note that if $x(t)$ is a sinusoidal function whose time integral stays finite all the time, then the output $y(t)$

$$y(t) = x(t) - \frac{1}{R_1'C}\int x(t)\,dt$$

will remain finite. Otherwise, the integral grows increasingly larger and eventually it will cause overflow in the $y(t)$ register, which in turn will not perform the correct operation. From another point of view, the current $y(t)/R_1$, flowing through the inductor will increase indefinitely after a step voltage $x(t)$ = constant is applied.

$$i_L(t) = \frac{1}{L}\int v(t)\,dt$$

Example 13: Realize a single capacitor $Y_1(s) = sC$.

Solution: By using Eq. (5.14)

$$H(s) = 1 - \frac{1}{R_1' Y_1(s)} = \frac{R_1' sC - 1}{R_1' sC} = \frac{V_2(s)}{V_1(s)} = \frac{Y(s)}{X(s)}$$

$$Y(s) = \left(1 - \frac{1}{R_1' sC}\right) X(s)$$

in the time domain we have

$$y(t) = x(t) - \frac{1}{R_1' C}\int x(t)\,dt \quad \text{or} \quad dy = dx - \left(\frac{1}{R_1' C}\right) x\,dt$$

The network of $H(s)$ can be realized as shown in Fig. 44. With $H(s)$ realized, the single capacitor C can be easily obtained as shown in Fig. 45.

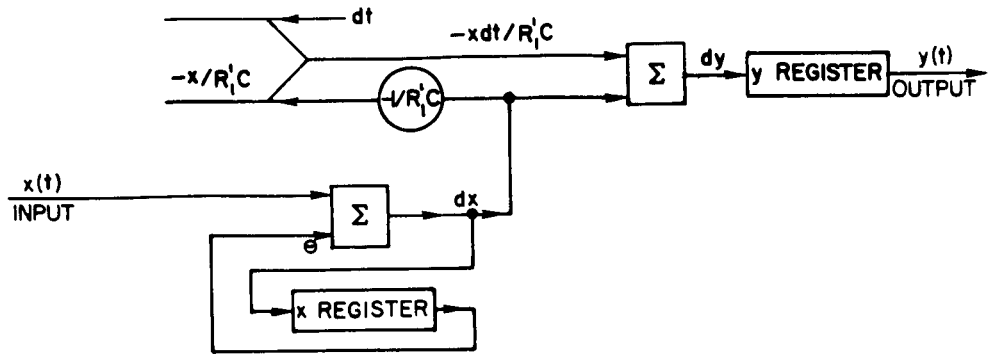


Fig. 44 REALIZATION OF $H(s) = (sR'_1-1)/sR'_1C = Y(s)/X(s)$

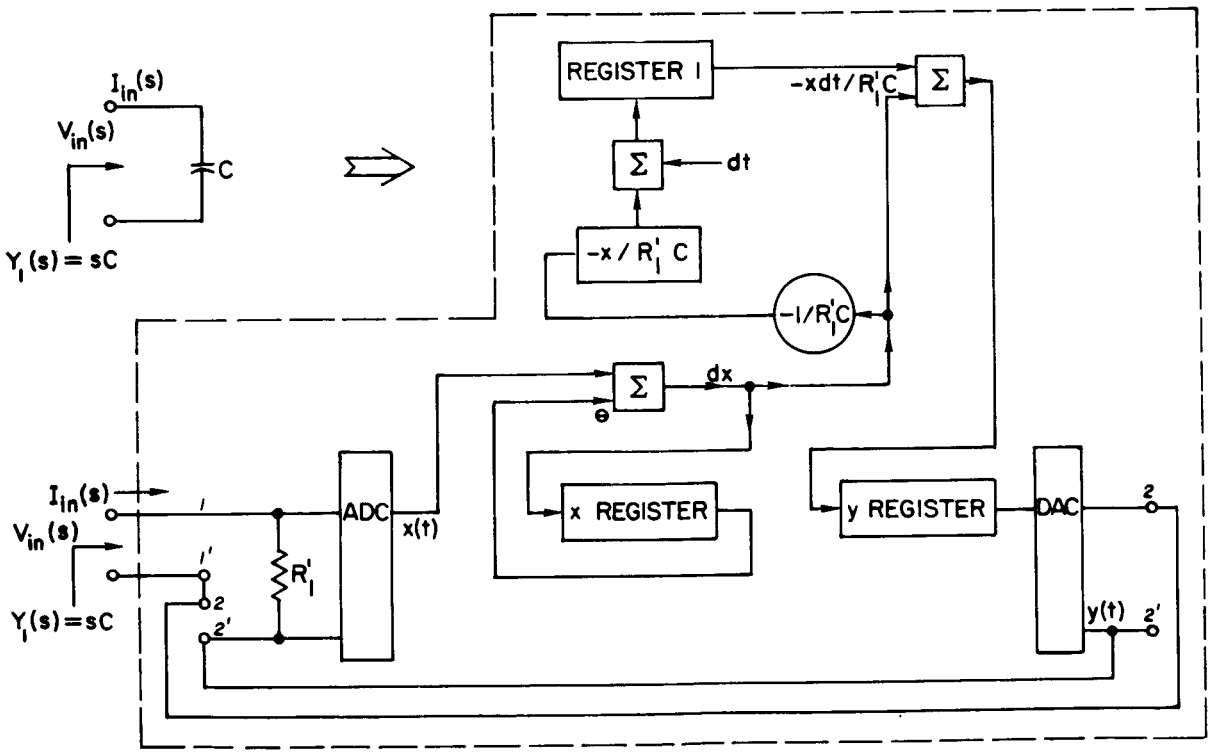


Fig. 45 REALIZATION OF A SINGLE CAPACITOR

From Example 12 we know that with the length of the registers sufficiently long, we could theoretically construct an almost ideal inductor. In other words, a circuit with a very high-Q can be obtained. From the realization techniques presented in this paper it is obvious that any driving point immittance function with negative values or with poles in the right half of the complex frequency plane can be realized without extra effort.

Example 14: Realize the following immittance functions

a. $Z_1(s) = -s$ (negative inductor, $L = -1$)

b. $Y_1(s) = s-1$ (tunnel diode, $C = 1$, $R = -1$)

Solution:

a. Using Eq. (5.9)

$$G_v(s) = \frac{R_2}{R_1} + 1 - \frac{R_2}{-s} = \left(1 + \frac{R_2}{R_1}\right) + \frac{R_2}{s} = \frac{V_2(s)}{V_{in}(s)} = \frac{Y(s)}{X(s)}$$

$$Y(s) = \left(1 + \frac{R_2}{R_1}\right) X(s) + \frac{R_2 X(s)}{s}$$

Transforming back to the time domain, we have

$$y(t) = \left(1 + \frac{R_2}{R_1}\right) x(t) + R_2 \int x(t)\, dt$$

or

$$dy = \left(1 + \frac{R_2}{R_1}\right) dx + R_2\, x\, dt$$

The network with the transfer function $G_v(s)$ can be realized easily, as shown in Fig. 46. With $G_v(s)$ realized, the negative inductor can be realized, as shown in Fig. 47.

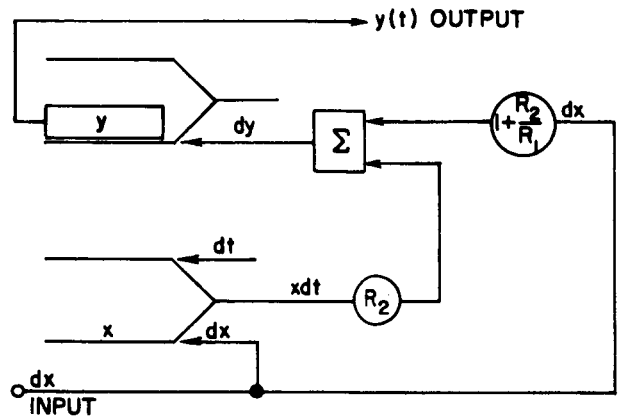


Fig. 46 REALIZATION OF $G_v(s) = (1 + R_2/R_1) + R_2/s$

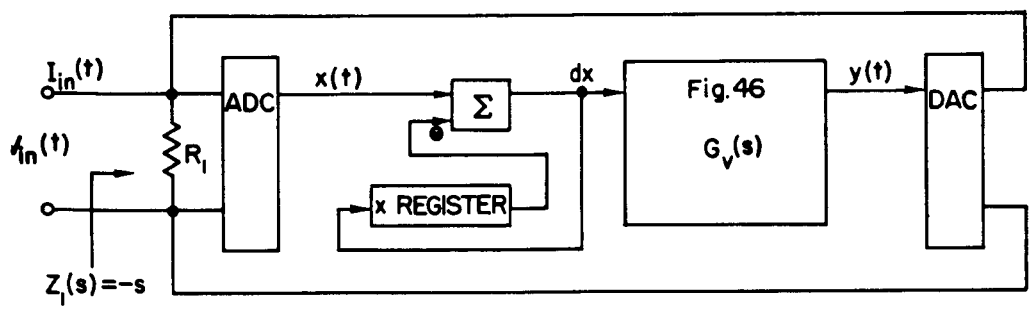


Fig. 47 REALIZATION OF $Z_1(s) = -s$

b. Using Eq. (5.14)

$$H(s) = 1 - \frac{1}{R_1'(s - 1)} = \frac{V_2(s)}{V_1(s)} = \frac{Y(s)}{X(s)}$$

then

$$Y(s) = \left(\frac{R_1's - R_1' - 1}{R_1's - R_1'}\right) X(s) = \frac{1 - \frac{1}{s}\left(1 + \frac{1}{R_1'}\right)}{1 - \frac{1}{s}} X(s)$$

In the time domain, we have

$$y(t) = x(t) + \int \left[y(t) - \left(1 + \frac{1}{R_1'}\right) x(t)\right] dt$$

or

$$dy = dx + \left[y - \left(1 + \frac{1}{R_1'}\right) X\right] dt$$

The network with $H(s) = 1-1/[R_1'(s-1)]$ can be realized, as can the tunnel diode (see Figs. 48 and 49).

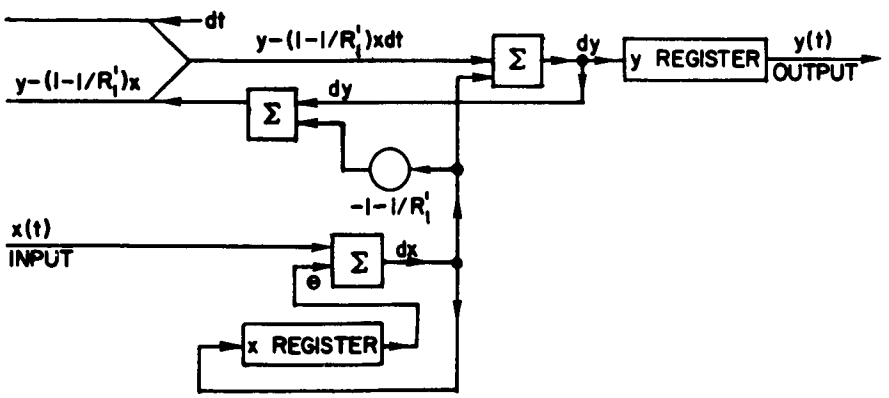


Fig. 48 NETWORK OF $H(s) = 1-1/[R_1'(s-1)]$

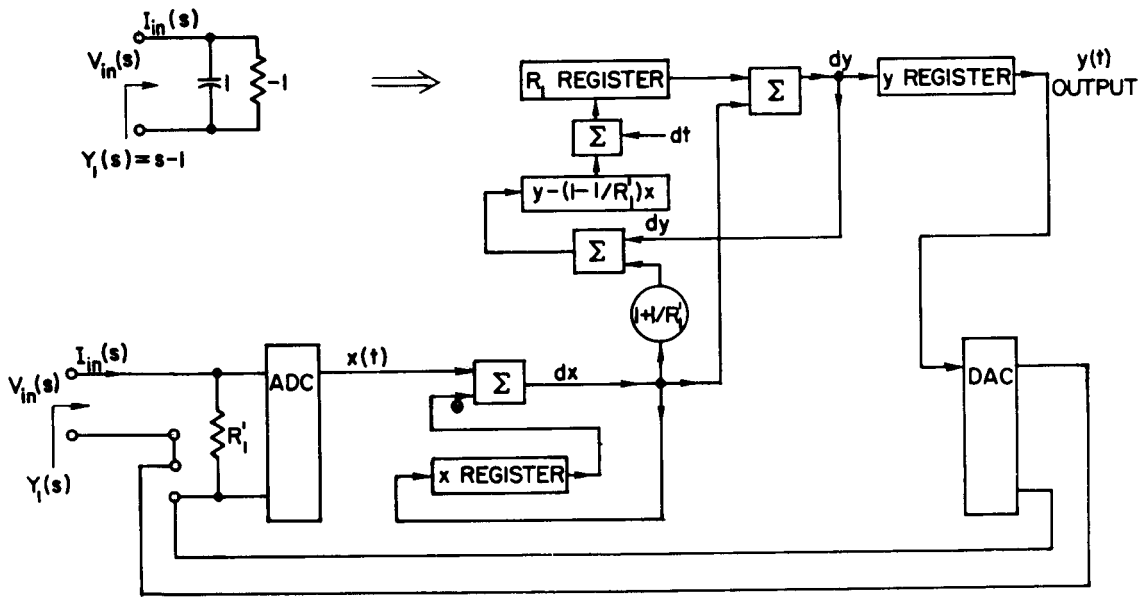


Fig. 49 NETWORK OF $Y_1(s) = s-1$

Appendix A

In the frequency domain approximation, the principal problem is to find a rational function G(s) whose magnitude $|G(jw)|$ approximates the ideal low-pass characteristic, as in Fig. A-1, according to a predetermined error criterion. Two approximations are discussed.

1. The Maximally Flat Low-Pass Filter Approximation

$$|G(jw)| = \frac{1}{\sqrt{1 + w^{2n}}}$$

is known as the n^{th}-order Butterworth or maximally flat low-pass response, and is an approximation of the ideal response of Fig. A-1.

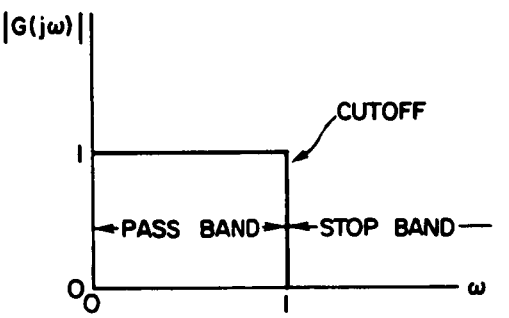


Fig. A-1 IDEAL LOW-PASS FILTER CHARACTERISTIC

The poles of this function are defined by the equation

$$1 + (-s^2)^n = 0$$

Then the pole locations are

$$S_k = \exp\left(j \frac{2k - 1}{n} \frac{\pi}{2}\right) \qquad n \text{ even}$$

$$S_k = \exp\left(j \frac{2k}{n} \frac{\pi}{2}\right) \qquad n \text{ odd}$$

or

$$S_k = \exp\left(j \frac{2k + n - 1}{n} \frac{\pi}{2}\right) \qquad k = 1, 2, 3, \ldots, 2n \ .$$

The poles so defined are located on a unit circle in the s-plane and are symmetrical with respect to both the real and imaginary axes. To form the function $G(s)$ from the given $|G(jw)|^2$, we reject the right-half plane poles, and from the left-half plane poles form the all-pole function,

$$G(s) = \frac{1}{1 + a_1 s + a_2 s^2 + \ldots + a_n s^n}$$

The coefficients of the denominator polynomials of $G(s)$, sometimes called Butterworth polynomials are tabulated in Table A-1.

Table A-1

COEFFICIENTS OF THE DENOMINATOR POLYNOMIALS OF $G(s)$

n	a_1	a_2	a_3	a_4	a_5	a_6
1	1.0000					
2	1.4142	1.0000				
3	2.0000	2.0000	1.0000			
4	2.6131	3.4142	2.6131	1.0000		
5	3.2361	5.2361	5.2361	3.2361	1.0000	
6	3.8637	7.4641	9.1416	7.4641	3.8637	1.0000

2. Chebyshev or Equal-Ripple Approximation

The squared magnitude form

$$|G(w)|^2 = \frac{1}{1 + \varepsilon^2 \, Cn^2(w)}$$

is an equal-ripple approximation of Fig. A-1, where $Cn(w)$ is the n^{th}-order Chebyshev polynomial and $\varepsilon < 1$ is a real constant. Chebyshev polynomials are defined in terms of the real variable z by the equation

$$C_n(z) = \cos(n \cos^{-1} z)$$

Further define $z = \cos w$, then

$$C_n(w) = \cos nw$$

and a recursion formula can be found as

$$C_{n+1}(z) = 2zC_n(z) - C_{n-1}(z)$$

with

$$C_o(z) = 1, \quad C_1(z) = z.$$

The poles of this equal-ripple form of response can be found as $S_k = \sigma_k + jw_k$

where

$$\sigma_k = \pm \sinh a \sin \frac{2k - 1}{n} \frac{\pi}{2}$$

$$w_k = \cosh a \cos \frac{2k - 1}{n} \frac{\pi}{2} \qquad k = 1,2,3,\ldots,2n$$

$$a = \frac{1}{n} \sinh^{-1} \frac{1}{\varepsilon}$$

Again the right-half plane poles are rejected in synthesis procedures.

PRECEDING PAGE BLANK NOT FILMED.

Appendix B

SIMULATION OF SINUSOIDAL RESPONSE OF THE TRANSFER FUNCTION G(s) = 1/(s+1)

The program has been written in extended ALGOL. It has been run on B5500 machine at Stanford University.

```
BEGIN
     REAL  Z,YTRUE,X,XX,XXX,YAPPROX;
     INTEGER  M,N,P,XREG, Y1REG,DX,DY,DYY,YREG,RREG,T,DT,TFINAL;
     READ  (M,N,TFINAL);
           P←2*N;
           YREG←Y1REG←XREG←RREG←0;
     FOR  T←1 STEP 1 UNTIL TFINAL DO
     BEGIN
       Z←T/P;
       X←(P-1)×SIN(Z);
       XX←X-XREG; XXX←ENTIER(ABS(XX));
     IF  XX<0 THEN DX←-XXX ELSE DX←XXX;
         XREG←XREG+DX;
     IF  ABS (RREG)≥M AND RREG>0 THEN
       BEGIN  DY←1; RREG←RREG-M
       END ELSE IF ABS (RREG) ≥M AND RREG<0 THEN
     BEGIN DY←-1; RREG←RREG+M
     END ELSE DY←0;
       DYY←DX-DY;
       Y1REG←Y1REG+DYY;
       RREG←RREG+Y1REG;
     YREG←YREG+DY;
     YTRUE←P×0.5×(EXP(-Z)+SIN(Z)-COS(Z));
       WRITE (T,XREG,Y1REG,RREG,YREG,YTRUE);
       END
       END
     DATA CARD
     256.0      8.0      1500
```

PRECEDING PAGE BLANK NOT FILMED.

Appendix C

UNIT STEP RESPONSE OF G(s) = 1/(s+1) SIMULATION ON B5500 MACHINE

```
BEGIN

     REAL  M,N,P,Z1,Z2,YYREG,YREG,DY,DYY,X1,X2,DX,DELX,RREG,T,TFINAL;
     READ  (M,N,TFINAL);
           P←2*N; YREG←YYREG←RREG←0;
     FOR   T←1 STEP 1 UNTIL TFINAL DO
     BEGIN
       Z1←T/P; X1←(P-1);
     IF T←1 THEN
     BEGIN DX←(P-1)
     END ELSE DX←0;
     IF RREG≥M  THEN
     BEGIN DY←1; RREG←RREG-M
     END ELSE DY←0;
     DYY←DX - DY; YYREG←YYREG+DYY;
     RREG←RREG+YYREG; YREG←YREG+DY;
     WRITE (T,YREG)
     END
END.
```

PRECEDING PAGE BLANK NOT FILMED.

Appendix D

The Analog to Digital Increment Converter (ADIC) is a device which converts the difference of two analog quantities to digital form. In particular, if an analog signal is applied to the input of the ADIC, the difference or the increment of the analog quantities measured at two consecutive bit times are converted to digital form. One implementation is shown in Fig. D-1 where the operational amplifiers are employed to get

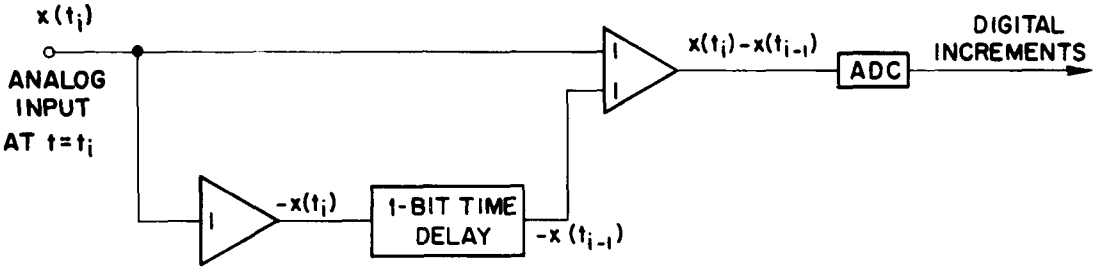


Fig. D-1. ONE IMPLEMENTATION OF ADIC

the difference of the signals at two consecutive bit times. A difference amplifier can be used to replace the two operational amplifiers if it is available. The gain of the amplifiers does not have to be unity. When necessary, gain adjustments can be made to fit the ADC input levels.

Similarly, another possible implementation is shown in Fig. D-2 where the analog signal is converted to digital form first, then subtracted from the previous digital quantity by a digital adder-subtractor. The difference output is the digital increment.

Among the two implementations, the first one, (Fig. D-1), is preferred and a few advantages of it can be mentioned. The input level to the ADC is limited therefore fewer comparators are needed in the ADC and

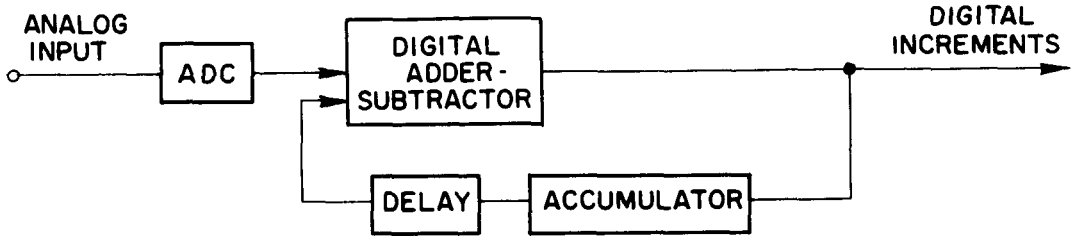


Fig. D-2. ANOTHER IMPLEMENTATION OF ADIC

more accuracy can be achieved. The conversion time of this ADIC is much less than that of the second implementation because only increments are transferred rather than the full words. Besides, the difference of the two analog quantities can be obtained immediately from the difference amplifier whereas the digital subtraction takes time.

REFERENCES

1. Braun, E. L., Digital Computer Design. Academic Press, New York, 1963, pp. 448-520.

2. Jackson, A. S. Analog Computation. McGraw-Hill Book Co., New York, 1960, pp. 578-586.

3. Kuo, F. F. Network Analysis and Synthesis. 2nd ed., John Wiley and Sons, Inc., New York, 1966.

4. Kuo, F. F. and J. F. Kaiser. System Analysis by Digital Computer. John Wiley and Sons, Inc., New York, 1966, pp. 218-285.

5. Mantey, P. E. Digital-computer implementation of linear systems. Report No. SU-SEL-66-063, Stanford Electronics Laboratories, Stanford University, Stanford, Calif., Oct. 1966.

6. Mayorov, F. V. Electronic Digital Integrating Computers--Digital Differential Analyzers. ed. Dr. Y. Chu. American Elsevier Publishing Co., New York, 1964.

7. Nelson, D. J. A foundation for the analysis of analog-oriented combined computer system. Technical Report, Stanford Electronics Laboratories, Stanford University, Stanford, Calif., 1962.

8. Rader, C. M. and B. Gold. Digital filter design techniques. Technical Note 1965-63, M. I. T. Lincoln Laboratory, Cambridge, Mass.

9. Van Valkenburg, M. E. Introduction to Modern Network Synthesis. John Wiley and Sons, Inc., New York, 1960, pp. 373-392.

10. Weinburg, L. Network Analysis and Synthesis. McGraw-Hill Book Co., New York, 1962, pp. 485-553.

11. Clark, R. N. Introduction to Automatic Control System. John Wiley and Sons, Inc., New York, 1962.