A Digital Filter Implementation of the Deep Space Transponder Phase Lock Loop Integrator

J. W. Heller
Communications Systems Research Section

The phase lock loop integrating filter in the NASA standard deep space transponder has suffered in reliability due to the malfunctioning of feedback capacitors. In this paper an alternative design, using a digital filter, is presented which eliminates these critical components. This filter can also be used in other flight and ground equipment utilizing phase lock loops.

I. Introduction

In the present version of the NASA deep space transponder the phase lock loop integrating filter has an extremely long time constant of 3000 seconds. The electronic circuit which realizes this consists of an operational amplifier with a large (11.6 μf) feedback capacitance. These capacitors are critical as any leakage current can drastically affect the pole location of the desired transfer function. Unfortunately, the capacitors have not been extremely reliable, causing a degradation of the loop performance. This article discusses a digital implementation of the integrating filter and, although capacitors are not eliminated, the capacitances are much smaller and used in an antialiasing filter where the filter requirements are less critical.

II. Analysis of the Linear Integrating Filter

The large feedback capacitance is a consequence of the specified filter transfer function

$$F(s) = \frac{1 + \tau_2 s}{1 + \tau_1 s} \tag{1}$$

where

$$\tau_2 \approx \frac{4d^2 + 1}{2W}$$

and

$$\tau_1 \approx \frac{\tau_2^2}{4d^2} \cdot A \cdot K$$

where d is the loop damping, W is the loop noise bandwidth, K is the open loop gain and A is the rms signal amplitude. For the deep space transponder $\tau_2 \approx 0.083$ seconds and $\tau_1 \approx 3000$ seconds.

A differential input integrating filter circuit similar to the one in the transponder is shown in Fig. 1. The transfer function for this circuit is

$$e_o = -\frac{\dfrac{R_F}{R_2}\dfrac{\left(\frac{1}{sC}+R_1\right)}{\left(\frac{1}{sC}+R_1+R_F\right)}}{1+\dfrac{1}{A}+\dfrac{R_F}{AR_2}\dfrac{\left(\frac{1}{sC}+R_1\right)}{\left(\frac{1}{sC}+R_1+R_F\right)}} \cdot e_i \qquad (2)$$

Since A is large ($3 \cdot 10^5$) and the ratio R_F/AR_2 is small, then Eq. (2) simplifies to

$$e_o \approx -\frac{R_F}{R_2}\frac{\left(\frac{1}{sC}+R_1\right)}{\left(\frac{1}{sC}+R_1+R_F\right)} e_i \qquad (3a)$$

This equation can be rearranged to resemble Eq. (1)

$$\frac{e_o}{e_i} = -\frac{R_F}{R_2}\frac{1+R_1Cs}{1+(R_1+R_f)Cs} \qquad (3b)$$

and by comparison with Eq. (1)

$$\tau_2 = R_1C \qquad (4)$$

and

$$\tau_1 = (R_1+R_F)C \qquad (5)$$

For $\tau_1 = 3000$ seconds Eq. (5) implies a large value for C. For example, in the present transponder filter design R_F is realized using a T-network (not shown in Fig. 1) to achieve a value of 260 MΩ. However, even this resistance summed with the 7.5 kΩ resistance chosen for R_1 requires a steep capacitance of 11.6 μf which, in practice, requires two 5.8-μf capacitors in parallel. Unfortunately, even small leakage currents in these capacitors can produce large changes in the circuit transfer function. Therefore alternative designs which reduce or eliminate the capacitors are worth considering. One such design using a digital filter performs the integration using binary additions and multiplications. No large capacitances are required, and where capacitors are used their leakage effect on the transfer characteristic is much less than for the present circuit.

III. Digital Filter Design

In order to preserve the transient response of the linear integrator an impulse-invariant, infinite impulse response, digital filter structure was derived as follows. The impulse response of the linear filter

$$h(t) = \mathscr{L}^{-1}[H(s)] = \mathscr{L}^{-1}\left[\frac{1+\tau_2 s}{1+\tau_1 s}\right]$$

$$= \frac{\tau_2}{\tau_1}\delta(t) + \left(\frac{1}{\tau_1} - \frac{\tau_2}{\tau_1^2}\right)e^{-t/\tau_1}\,u(t) \qquad (6)$$

was set equal to the desired digital filter output at each sampling interval; i.e.

$$h(n) = h(t)\Big|_{t\to nT} . \qquad (7)$$

Then the z transform of the sampled impulse response was determined

$$H(z) = Z[h(n)] = \frac{\left(\dfrac{\tau_2+1}{\tau_1} - \dfrac{\tau_2}{\tau_1^2}\right) - \dfrac{\tau_2}{\tau_1}e^{-T/\tau_1}z^{-1}}{1 - z^{-1}e^{-T/\tau_1}}$$

and from this system function the digital filter structure can be drawn (Fig. 2). The choice of the sampling interval T depends on the precision with which the coefficients can be represented in the filter hardware (i.e., register width) and the antialiasing filter requirements. The latter is needed to attenuate signal frequencies above the Nyquist rate, $f_s/2$ (=1/2T).

IV. Implementation

Figure 3 shows the schematic diagram for the digital filter and its associated circuits. The left-most operational amplifier is used for scaling and retains the differential input configuration of the original linear filter. The second op-amp serves as an antialiasing filter using a 3-pole Butterworth low-pass filter design. The cutoff frequency of 1 Hz was chosen so that aliasing from frequencies greater than 10 Hz are attenuated by at least 54 dB and therefore are less than the least significant bit of the digital filter's 9 bit digital-to-analog converter.

The analog-to-digital conversions, digital filtering and digital-to-analog conversion are all performed by an Intel 2920 digital signal processing integrated circuit. A functional diagram is shown in Fig. 4. The 2920 is the most compact of the digital signal processors available as it incorporates a digital-to-analog converter (but not the voltage reference) on the chip along with the usual digital components such as an ALU, registers, etc. The filter structure is realized in software stored in an erasable memory capable of holding 192 instructions. Approximately 35 instructions are needed to convert the analog signal into a 9 bit binary representation. The reverse process requires another 15 memory locations. In addition, a "correction curve" was programmed to minimize the errors resulting from the nonlinear output of the 2920's sample-and-hold buffer (for a discussion of this problem see Ref. 1). Also, a software timer was included in order to execute the filter code at the appropriate submultiple frequency of the 2920 hardware clock. The submultiple frequency was set equal to the sampling frequency of 20 Hz. Figure 5 shows an excerpt of the 2920 code listing the digital filter structure. Although the internal register width of the 2920 is 25 bits, this was not of sufficient accuracy for representing the pole coefficient, so much of the filter computation was done using two registers for each variable (extended precision).

The op-amp to the right of the 2920 is used as an adjustable-gain voltage follower. A fourth op-amp provides a precise reference voltage for the 2920's digital-to-analog converter.

V. Performance

The circuit was tested for its step response and frequency response. The output for a 0.75-volt step input was recorded over a 3-hour period at 5-minute intervals and compared with the ideal response to the linear integrator. The maximum error (excluding near-zero values seen during the first minute and attributable to offsets, etc.) was 1.5 percent. Frequency responses of the digital and ideal filters are shown in Fig. 6. It was not possible to test the filter in a loop due to budget restrictions.

VI. Conclusion

It appears possible to replace the present linear integrator in the NASA deep space transponder with a digital filter that is not as susceptible to capacitor reliability problems as are the filters presently used.

Reference

1. Heller, J., "An Evaluation of the Intel 2920 Digital Signal Processing Integrated Circuit," *TDA Progress Report 42-63*, Jet Propulsion Laboratory, Pasadena, Calif., pp. 108-113, June 15, 1981.

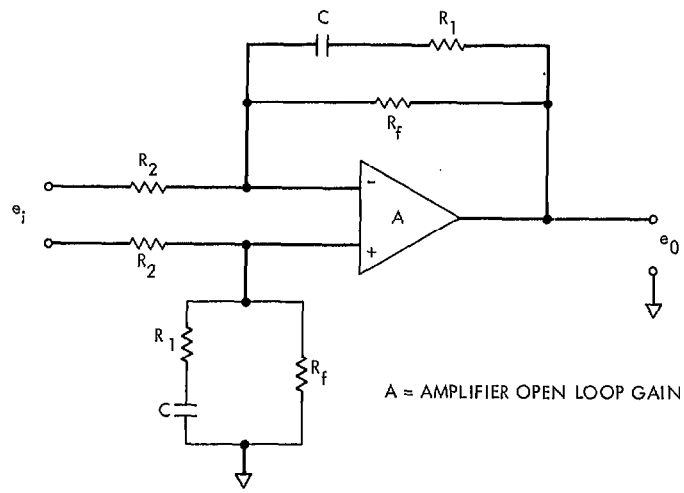

Fig. 1. Simplified schematic of the deep space transponder integrator

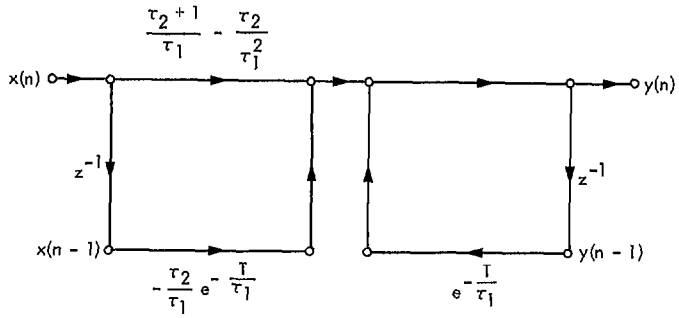

Fig. 2. Digital filter structure for integrator

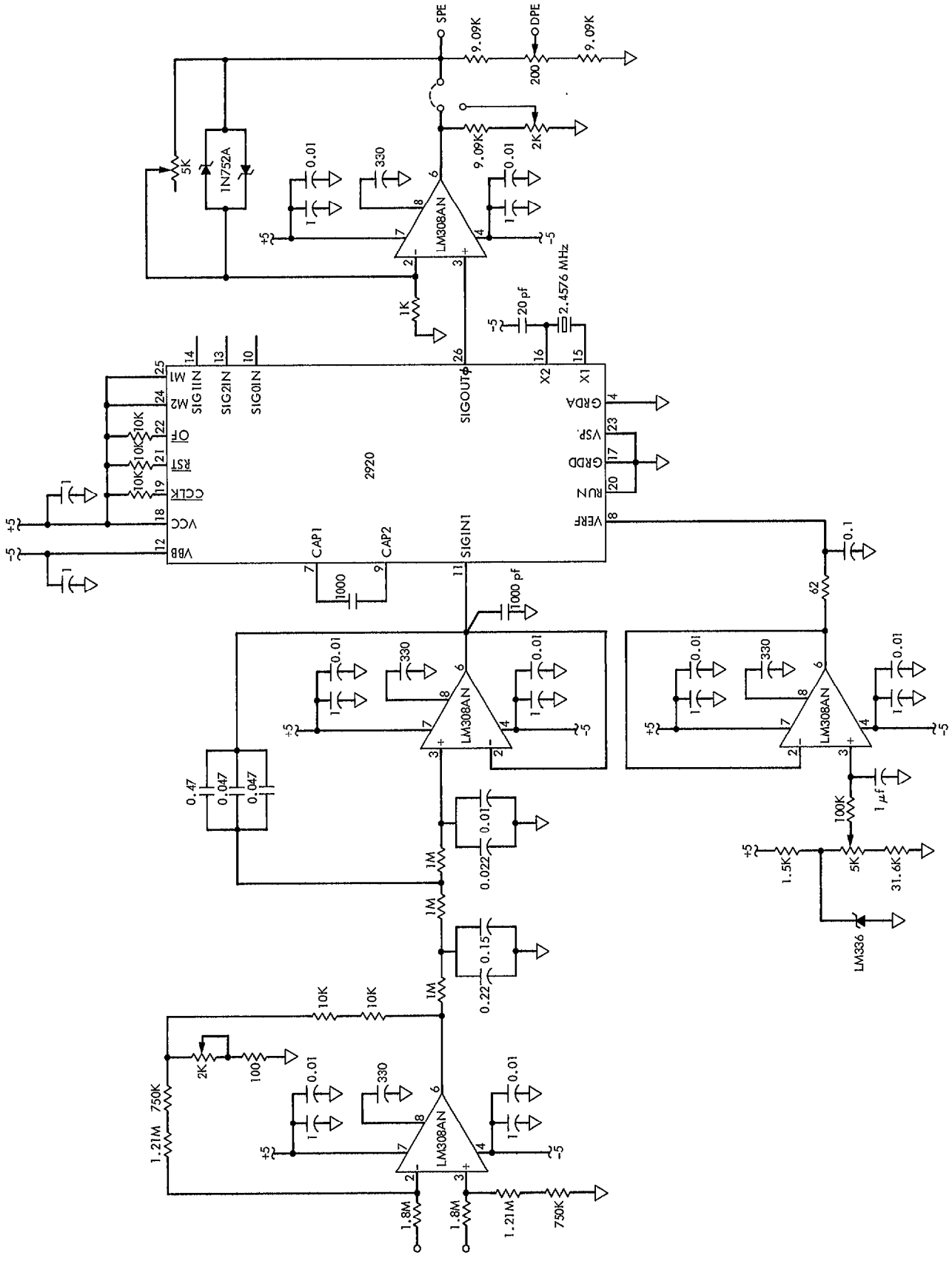

Fig. 3. Digital filter schematic diagram

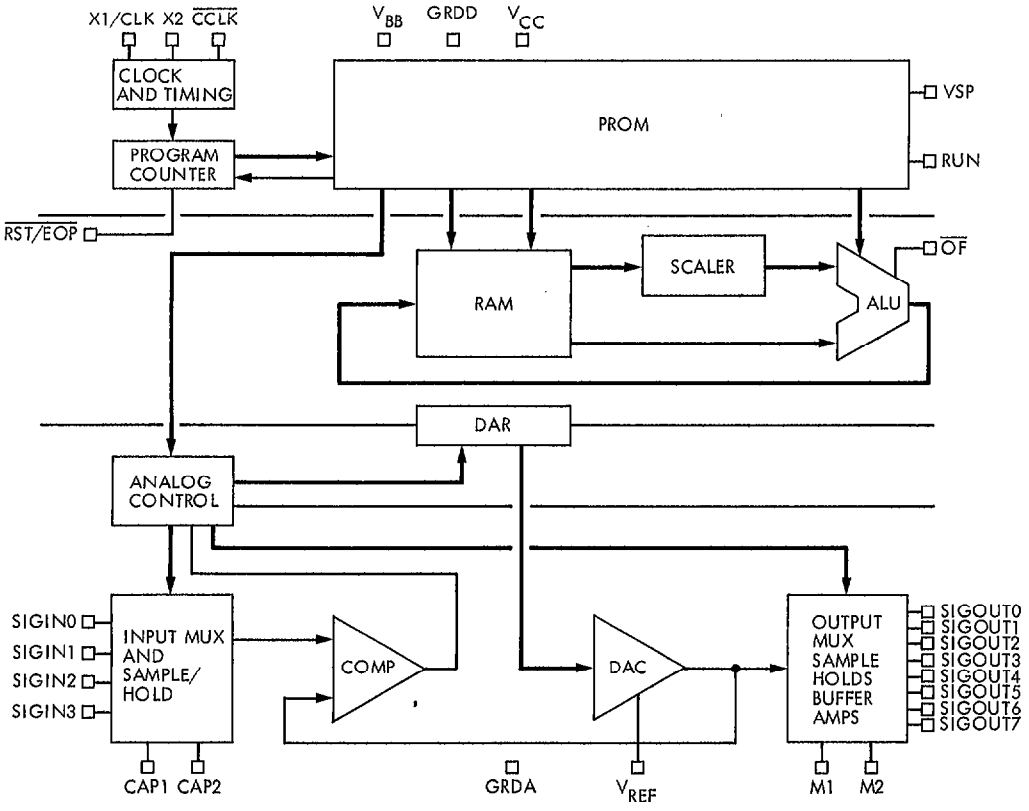

Fig. 4. Block diagram of 2920 signal processor

```
LDA 0,DAR,R03,INPUT SCALED HERE
    ; 0=0.125000000*DAR
ADD 0,0,R13
    ; 0=0.125015258*DAR
SUB 0,DAR,R03
    ; 0=1.52588071/10**5*DAR
ADD 0,0,R02
    ; 0=1.9072508/10**5*DAR
ADD 0,0,R06
    ; 0=1.9371533/10**5*DAR
SUB 0,0,R04
    ; 0=1.8160811/10**5*DAR
;END OF MACRO ADCONV
;
;SUB MULTIPLE SAMPLING TIMER
;
SUB CNT, KP1, R06;SUBTRACT 0.000000001 FROM CNT
LDA DAR, CNT, R00;MOVE COUNT (CNT) TO DAR TO CHECK SIGN
LDA CNT, KP6, R00, CNDS;IF NEGATIVE COUNT RE-INITIALIZE TO 389
ADD CNT, KP5, R06, CNDS;CONTINUE BUILDING INITIALIZATION SUM
NOP
;
; LOW-PASS FILTER
;
; POLE SECTION
;
;UPDATE INPUT AND DELAY EVERY 1/20 TH OF A SECOND
;
LDA OUT1_P1,OUT0_P1,R00,CNDS;UPDATE DELAY (UPPER WORD)
LDA OUT1_P1X,OUT0_P1X,R00,CNDS;UPDATE DELAY (LOWER WORD)
LDA IN0_P1,0,R00,CNDS;UPDATE INPUT
;
; LOWER MULTIPLICATION
;
LDA OUT0_P1X,OUT1_P1X,R06
    ; OUT0_P1X=0.015625000*OUT1_P1X
SUB OUT0_P1X,OUT1_P1X,R01
    ; OUT0_P1X=-0.48437500*OUT1_P1X
SUB OUT0_P1X,OUT0_P1X,R07
    ; OUT0_P1X=-0.48059082*OUT1_P1X
SUB OUT0_P1X,OUT1_P1X,R04
    ; OUT0_P1X=-0.54309082*OUT1_P1X
ADD OUT0_P1X,OUT0_P1X,R12
    ; OUT0_P1X=-0.54322338*OUT1_P1X
;
;UPPER MULTIPLICATION
;
```

Fig. 5. Excerpt from digital filter instruction sequence

```
LDA OUT0_P1,OUT1_P1,R06
    ; OUT0_P1=0.015625000*OUT1_P1
SUB OUT0_P1,OUT1_P1,R01
    ; OUT0_P1=-0.48437500*OUT1_P1
SUB OUT0_P1,OUT0_P1,R07
    ; OUT0_P1=-0.49059082*OUT1_P1
SUB OUT0_P1,OUT1_P1,R04
    ; OUT0_P1=-0.54309082*OUT1_P1
ADD OUT0_P1,OUT0_P1,R12
    ; OUT0_P1=-0.54322338*OUT1_P1
;
;ADD CARRY FROM LOWER WORD TO UPPER WORD
;
LDA 0,OUT0_P1X,R01
AND 0,KP4,R00
SUB OUT0_P1X,0,R00
LDA 0,0,R13
ADD OUT0_P1,0,R10
;
;SCALE BACK PRODUCT
;
;LOWER PRODUCT FIRST
;
LDA OUT0_P1X,OUT0_P1X,R07
LDA OUT0_P1X,OUT0_P1X,R08
;
;NEXT COPY UPPER PRODUCT
;
LDA COPY,OUT0_P1,R00
;
;THEN SCALE BACK UPPER PRODUCT
;
LDA OUT0_P1,OUT0_P1,R07
LDA OUT0_P1,OUT0_P1,R08
;
;CREATE MASK TO COPY BITS FROM UPPER PRODUCT
;
LDA MASK,KP7,R13
ADD MASK,KP7,R10
ADD MASK,KP7,R07
ADD MASK,KP7,R04
ADD MASK,KP7,R01                                    @
LDA MASK,MASK,R00
;
;COPY BITS FROM UPPER PRODUCT AND SHIFT TO LOWER PRODUCT POSITION
;
AND COPY,MASK,R00
LDA COPY,COPY,L02
```

Fig. 5 (contd)

```
LDA COPY,COPY,L02
LDA COPY,COPY,L02
LDA COPY,COPY,L02
;
;ADD COPY TO SCALED DOWN LOWER WORD; ADD ANY CARRY TO UPPER WORD
;
ADD OUT0_P1X,COPY,R00
LDA Q,OUT0_P1X,R00
AND Q,KP4,R00
SUB OUT0_P1X,Q,R00
LDA Q,Q,R13
ADD OUT0_P1,Q,R10
;
;NOW ADD DELAYED VALUE TO SCALED-BACK DELAYED VALUE
;
ADD OUT0_P1X,OUT1_P1X,R00
LDA Q,OUT0_P1X,R00
AND Q,KP4,R00
SUB OUT0_P1X,Q,R00
ADD OUT0_P1,OUT1_P1,R00
LDA Q,Q,R13
ADD OUT0_P1,Q,R10
;
;AND FINALLY ADD INPUT SAMPLE TO SUM
;
ADD OUT0_P1,IN0_P1,R00
LDA Q,OUT0_P1X,R01
AND Q,KP4,R00
SUB OUT0_P1X,Q,R00
LDA Q,Q,R13
ADD OUT0_P1,Q,R10
;
;
; ZERO SECTION
;
IN0_Z1 EQU OUT0_P1
IN1_Z1 EQU OUT1_P1
LDA OUT0_Z1,IN1_Z1,R06
    ; OUT0_Z1=0.0156250000*IN1_Z1
ADD OUT0_Z1,OUT0_Z1,L02
    ; OUT0_Z1=0.078125000+IN1_Z1
ADD OUT0_Z1,IN1_Z1,R09
    ; OUT0_Z1=0.080078125*IN1_Z1
ADD OUT0_Z1,OUT0_Z1,R11
    ; OUT0_Z1=0.080117224*IN1_Z1
SUB OUT0_Z1,OUT0_Z1,L01
    ; OUT0_Z1=-0.080117224*IN1_Z1
```

Fig. 5 (contd)

```
SUB OUT0_Z1, IN1_Z1, R10
    ; OUT0_Z1=-0.08109378*IN1_Z1
SUB OUT0_Z1, OUT0_Z1, R10
    ; OUT0_Z1=-0.08101459*IN1_Z1
ADD OUT0_Z1, IN1_Z1, R10
    ; OUT0_Z1=-0.00003031*IN1_Z1
ADD OUT0_Z1, IN0_Z1, R00
    ; OUT0_Z1=-0.00003031*IN1_Z1+1.00000000*IN0_Z1
```

Fig. 5 (contd)

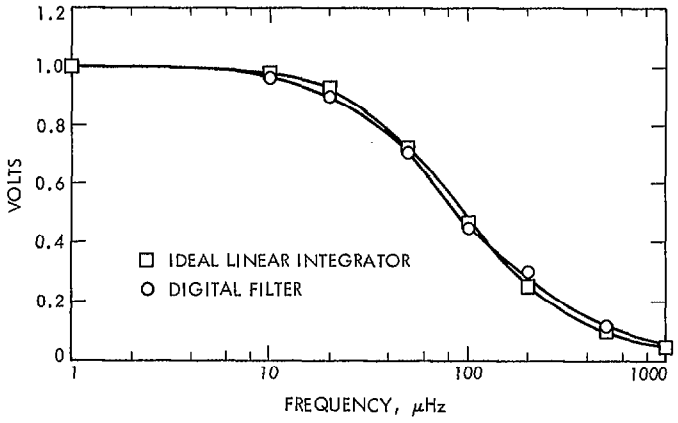

Fig. 6. Frequency response of the linear integrator and the ideal filter