Summer 1980

BILIN: a Bilinear Transformation Computer Program and its Applications

John Dana Greer
University of Central Florida

BILIN

A BILINEAR TRANSFORMATION COMPUTER PROGRAM AND ITS APPLICATIONS

BY

JOHN DANA GREER

ABSTRACT

Given a transfer function for a differential equation model, an approach for obtaining a solution is by way of the bilinear transformation. The bilinear transform approach is a numerical integration scheme which gives a discrete approximation to the differential equation solution. BILIN applies a series of polynomial transformations to the transfer function H(s). As a result, H(s) is mapped into the complex z plane obtaining the discrete transfer function H(z). From H(z), the difference equation is obtained whose solution y(nT) approximates the actual differential solution y(t). Hence, BILIN provides a means for obtaining discrete transfer functions for the design of digital filters and/or solving linear time-invariant differential equations.

F. O. Simons, Jr.

Director of Research Report

BILIN

A BILINEAR TRANSFORMATION COMPUTER PROGRAM AND ITS APPLICATIONS

BY

JOHN DANA GREER
B.S.E., University of Central Florida, 1979

RESEARCH REPORT

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in the Graduate
Studies Program of the College of Engineering
at the University of Central Florida at Orlando, Florida

Summer Quarter
1980

TABLE OF CONTENTS

I. PROBLEM DEFINITION. 1

Theoretical Basis 3

An Efficient Algorithm for Generating H(z) from H(s). 6

II. SAMPLE APPLICATION OF BILIN 11

III. CONCLUSION. 16

. . . .

APPENDIX. 19

REFERENCES CITED. 24

PROBLEM DEFINITION

Many enginering problems involve real-time simulation of "real world" dynamic systems using computer software and/or specialized hardware models. Typically, these dynamic models are based on a differential equation system model. For purposes of system analysis, it is often desired to obtain the transient solution to the differential model.

Rise time, percent overshoot, settling time and other important system response characteristics can be extracted from the transient response (solution) to evaluate and/or predict system performance. Typically, the system transient response is with respect to a unit impulse, unit step, or unit pulse input.

If the differential equation model is linear time-invariant, the transient solution can always be obtained via Laplace transform provided the input is Laplace transformable, which is usually the case. The solution, i.e., the output, y(t), for a single-input single-output differential equation model can be obtained via inverse Laplace transform:

$$y(t) = \mathcal{L}^{-1}\left[Y(s)\right] \qquad (1)$$

where

$$Y(s) = H(s)\ X(s) \tag{2}$$

$$X(s) = \mathcal{L}\left[x(t)\right]\ \text{, the input} \tag{3}$$

$$H(s) = \mathcal{L}\left[h(t)\right]\ \text{, the transfer function} \tag{4}$$

A Laplace transform solution of (2) using a computer would require programs which factor the denominator polynomial, partial fraction expand Y(s), determine the residues and obtain the inverse Laplace transform of the resulting expression. Considering the different cases (real, imaginary, or complex roots; simple or multiple roots) and lengthy difficult calculations (e.g., iterative algorithms for finding roots of greater than fourth order polynomials), any program which took this approach would be inefficient in terms of speed and memory requirements.[1]

Another possible approach to obtaining a solution of (2) would be to generate a difference equation from (2) in an optimally efficient manner, whose solution, y(nT), can be shown to yield:

$$y(nT) \doteq y(t)\Big|_{t = nT} \tag{5}$$

where y(t) is the actual differential equation solution. Such an optimized approach was taken by Dr. F. O. Simons, Jr., P.E. and Dr. R. C. Harden, P.E., Professors of Engineering of the Department of Electrical Engineering at the University of Central Florida in their published paper "Differential Equation Solutions for Up to 10th Order

System Theory Models with HP-67 Compulators." Their approach is optimal with respect to the number of operations used in obtaining the difference equation coefficients from the differential equation coefficients. Their paper contains an algorithm (originally discovered by Simons) which provides the basis for BILIN and is of second order accuracy.[1]

The authors' basic approach was to take H(s) and, through a series of polynomial transformations, map the transfer function into the complex z plane obtaining H(z). From the numerator and denominator coefficients of H(z), the difference equation, whose solution is y(nT), is directly obtained. Hence, any differential equation solution which can be obtained via Laplace transform can also be solved using Simons' and Harden's approach. More important is that these transformations allow for a direct real-time simulation of dynamic systems. A theoretical basis for their approach follows.

Theoretical Basis

Consider

$$y(nT) \doteq y(t)\Big|_{t = nT} \qquad (6)$$

where y(t) is the actual differential equation solution and T is the sample time period. The differential equation expressed by:

$$w(t) \triangleq y'(t) = \frac{dy(t)}{dt} \tag{7}$$

can be expressed in terms of the first order backward difference given by:

$$w(nT) \triangleq \frac{y(nT) - y(nT-T)}{T} + e_1 \tag{8}$$

or the first order forward difference given by:

$$w(nT-T) \triangleq \frac{y(nT) - y(nT-T)}{T} + e_2 \tag{9}$$

From the Mean Value Theorem of Calculus, it can be concluded that e_1 and e_2 tend to be opposite in sign. A simple arbitrary sketch of y(t) versus t can be used to interpret this concept. Thus, taking the z transform of (8) and (9), the following equations can be obtained:

$$W(z) = \frac{Y(z) - z^{-1}Y(z)}{T} + E_1(z) \tag{10}$$

$$z^{-1}W(z) = \frac{Y(z) - z^{-1}Y(z)}{T} + E_2(z) \tag{11}$$

Adding (10) and (11) and solving for W(z), the result is:

$$W(z) = \frac{2}{T}\,\frac{1- z^{-1}}{1+ z^{-1}}Y(z) + \frac{E_1(z)+ E_2(z)}{1 + z^{-1}} \tag{12}$$

Therefore,

$$w(nT) = y'(nT) = y_1'(nT) + e_T(nT) \triangleq w_1(nT) + e_T(nT) \tag{13}$$

where

$$e_T(nT) = \Sigma_0^k e_1(nT) + \Sigma_0^k e_2(nT) \tag{14}$$

i.e., the total cumulative error in the approximation of the differential equation solution is the cumulative sum of the forward and backward difference errors. Since these errors tend to be opposite in sign, they tend to cancel. Hence, $w_1(nT)$ represents a more accurate approximation than does (8) or (9). This result is the basis for Simons' and Harden's approach which is the well known bilinear transform approach consisting of:

Step 1. From the differential equation

$$\sum_0^N a_n \frac{d^n y(t)}{dt^n} = \sum_0^M b_m \frac{d^m x(t)}{dt^m} \tag{15}$$

obtain the Laplace transform model

$$Y(s) = \frac{\Sigma_0^M b_m s^m}{\Sigma_0^N a_n s^n} X(s) \tag{16}$$

for which the transfer function is

$$H(s) = \frac{Y(s)}{X(s)} = \frac{\Sigma_0^M b_m s^m}{\Sigma_0^N a_n s^n} \tag{17}$$

Step 2. Generate H(z) defined via

$$H(z) \triangleq \left. \frac{\Sigma_0^M b_m s^m}{\Sigma_0^N a_n s^n} \right|_{s = \frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}} = \frac{2}{T}\frac{z-1}{z+1}} \tag{18}$$

Step 3. Solve the difference equation defined by

$$Y(z) = H(z)X(z) \tag{19}$$

for $y(nT) \triangleq y(t)$ with $t = nT$ as follows:

From step 2, H(z) is obtained in the form

$$H(z) = \frac{\sum_{0}^{M=N} d_m z^{-m}}{\sum_{0}^{N} c_n z^{-n}} \qquad (20)$$

Next, H(z) is put into the form

$$H(z) = \frac{\sum_{0}^{M=N} f_m z^{-m}}{1 + \sum_{1}^{N} e_n z^{-n}} = \frac{Y(z)}{X(z)} \qquad (21)$$

where $f_m = d_m/c_0$ and $e_n = c_n/c_0$. It follows that

$$Y(z) = \sum_{0}^{M} f_m z^{-m} X(z) - \sum_{1}^{N} e_n z^{-n} Y(z) \qquad (22)$$

Taking the inverse z transform, the corresponding system output is

$$y(nT) = \sum_{0}^{M=N} f_m x(nT-kT) - \sum_{1}^{N} e_n y(nT-kT) \qquad (23)$$

The solution obtained has errors no larger than third order.[1]

An Efficient Algorithm for Generating H(z) from H(s)

The program BILIN implements the bilinear approach for obtaining an approximate solution for a given differential equation on the VAX 11/780. The key to the efficiency of this program, i.e., its small memory requirements and "short" execution time is Simons' algorithm for generating H(z) from H(s) (step 2 of the bilinear transform approach). The scale factor (2/T) in (18) can be handled by

$$b_m \leftarrow (2/T)^m b_m \tag{24}$$

and

$$a_n \leftarrow (2/T)^n a_n \tag{25}$$

i.e., replace each coefficient by itself multiplied by a corresponding power of (2/T). Having taken care of the (2/T) term, a means is needed for generating the d_m and c_n coefficients of

$$H(z) \triangleq \frac{N(z)}{D(z)} = \frac{\sum_0^{M=N} \; d_m z^{-m}}{\sum_0^{N} \; c_n z^{-n}} \tag{26}$$

from H(s); i.e.,

$$H(z) = \left.\frac{\sum_0^M b_m s^m}{\sum_0^N a_n s^n}\right|_{s\,=\,\frac{z-1}{z+1}} = \frac{Y(z)}{X(z)} \tag{27}$$

$$= \frac{\sum_0^M b_m (\frac{z-1}{z+1})^m}{\sum_0^N a_n (\frac{z-1}{z+1})^n} \tag{28}$$

$$= \frac{\sum_0^M b_m (z-1)^m (z+1)^{-m}}{\sum_0^N a_n (z-1)^n (z+1)^{-n}} \tag{29}$$

Multiplying (29) by $(z+1)^N/(z+1)^N$, it follows that

$$H(z) = \frac{\sum_0^M b_m(z-1)^m(z+1)^{N-m}}{\sum_0^N a_n(z-1)^n(z+1)^{N-n}} \overset{\triangle}{=} \frac{N(z)}{D(z)} \tag{30}$$

for which the numerator is also of order N, as originally indicated in (26). Also, H(z) may be put into the equivalent form[1]:

$$= \frac{\sum_0^M b_m\left(\frac{z-1}{z+1}\right)^m}{\sum_0^N a_n\left(\frac{z-1}{z+1}\right)^n} \cdot \frac{(z+1)^N}{(z+1)^N} \tag{31}$$

$$= \frac{Y\left(\frac{z-1}{z+1}\right)(z+1)^N}{X\left(\frac{z-1}{z+1}\right)(z+1)^N} \overset{\triangle}{=} \frac{N(z)}{D(z)} \tag{32}$$

In order to verify the algorithm for obtaining N(z) and D(z), consider just $D(s) = D(z)\big|_{z=s}$ and the following definitions:

$$D(s) \overset{\triangle}{=} (s+1)^N X\left(\frac{s-1}{s+1}\right) \tag{33}$$

$$E(s) \overset{\triangle}{=} X(s-1) \tag{34}$$

$$F(s) \overset{\triangle}{=} s^N E\left(\frac{1}{s}\right) \tag{35}$$

$$G(s) \overset{\triangle}{=} F\left(s+\frac{1}{2}\right) \tag{36}$$

$$J(s) \overset{\triangle}{=} s^N G\left(\frac{1}{s}\right) \tag{37}$$

Then

$$D(s) = J(2s) \tag{38}$$

The algorithm will be verified by showing that the D(s) of (38) is equal to the D(s) defined in (33), obtained from the application of transformations (34) through (38) to X(s). These transformations represent the algorithm discovered by Simons for generating H(z) from H(s). Hence, starting with

$$\begin{aligned}
D(s) &\triangleq J(2s) \\
D(s) &= J(s)\Big|_{s=2s} \\
&= (2s)^N G\left(\frac{1}{2s}\right) \\
&= (2s)^N F\left(s+\frac{1}{2}\right)\Big|_{s=\frac{1}{2s}} \\
&= (2s)^N F\left(\frac{1}{2s}+\frac{1}{2}\right) \\
&= (2s)^N F\left(\frac{s+1}{2s}\right) \\
&= (2s)^N \left[s^N E\left(\frac{1}{s}\right)\right]\Big|_{s=\frac{s+1}{2s}} \\
&= (2s)^N \left(\frac{s+1}{2s}\right)^N E\left(\frac{2s}{s+1}\right) \\
&= (s+1)^N X(s-1)\Big|_{s=\frac{2s}{s+1}} \\
&= (s+1)^N X\left(\frac{2s}{s+1} - 1\right) \\
&= (s+1)^N X\left(\frac{s-1}{s+1}\right) = D(z)\Big|_{z=s}
\end{aligned} \tag{39}$$

it can be concluded that the defined transformations are equivalent to the bilinear transformation of (27). Therefore, by applying the given transformations to Y(s) and X(s), the coefficients of H(z) are obtained.[1]

The algorithm based on these transformations consists of the following steps:

1. Replace the coefficients of X(s) and Y(s) of H(s) by the same coefficients multiplied by corresponding powers of (2/T).
2. Translate X(s) and Y(s) one unit to the right (which can be done by using an algorithm consisting of a series of synthetic divisions).[2]
3. Reverse the order of the coefficents of X(s) and Y(s).
4. Translate X(s) and Y(s) one half unit to the left (as in 2).
5. Reverse the order of the coefficients of X(s) and Y(s) again.
6. Finally, multiply the i^{th} coefficient of X(s) and Y(s) by 2^i.

SAMPLE APPLICATION OF BILIN

Consider the evaluation of the step response of the dynamic system model specified by:

$$H(s) = \frac{100}{s^2 + 10s + 100} \tag{40}$$

$$= \frac{(10)^2}{s^2 + 2(1/2)10s + (10)^2} \tag{41}$$

$$\triangleq \frac{\omega_n^2}{s^2 + 2q\omega_n + \omega_n^2} \tag{42}$$

the step response, i.e., the analytic solution is:

$$r(t) = 1 - (1-q^2)^{-\frac{1}{2}} \exp(-q\omega_n t) \cdot \sin(\omega_n(1-q^2)^{\frac{1}{2}} t + \emptyset) \quad t \geqq 0 \tag{43}$$

where $\emptyset = \arctan (1-q^2)^{\frac{1}{2}}/q$. Hence,

$$r(t) = 1 - (.75)^{-\frac{1}{2}} \exp(-5t) \cdot \sin(10(.75)^{\frac{1}{2}} t + \emptyset) \quad t \geqq 0 \tag{44}$$

where $\emptyset = \arctan 2(.75)^{\frac{1}{2}}$.

To obtain an H(z) for a given sample time which corresponds to the above H(s), the following data is read in:

$N \leftarrow 2,\ M \leftarrow 0,\ T \leftarrow 0.1T_n = 0.1(2\pi/10)$

$B(1) \leftarrow 100.,\ B(2) \leftarrow 0.,\ B(3) \leftarrow 0.$

$A(1) \leftarrow 100.,\ A(2) \leftarrow 10.,\ A(3) \leftarrow 1.$

The following transfer function is obtained:

$$H_1(z) = \frac{0.06985572794(1 + 2z^{-1} + z^{-2})}{1 - 1.275861690z^{-1} + 0.5552846021z^{-2}}$$

To obtain the difference equation (corresponding to $H_1(z)$) step response which gives a discrete approximation of the analytic solution, the following data is read in:

NPT $\leftarrow$ 100, YnT(1) $\leftarrow$ 0, YnT(2) $\leftarrow$ 0.

XnT(1) $\leftarrow$ 0., XnT(2) $\leftarrow$ 0., XnT(3) $\leftarrow$ 1.

XnT(4) $\leftarrow$ 1., . . . , XnT(100) $\leftarrow$ 1.

To compare the difference equation step response with the system step response, a FORTRAN plotting subroutine (this part of the program is not given in the program listing) was used. By passing to the subroutine the difference equation response and the exact response "sampled" at nT, $0 \leqq n \leqq 99$, the graph in Figure 1, page 14 was obtained. For this graph, y(nT) is plotted versus t/T_n. Note that the "exact solution" starts at y(nT) = 0, but the "approximate solution" starts at some value of y(nT) greater than 0 (this value is the value of f_0 in (21) through (23) for this case).

For a more accurate approximation of the system step response, a sample time of $T = .02T_n$ was used to obtain:

$$H_2(z) = \frac{0.003700709159(1 + 2z^{-1} + z^{-2})}{1 - 1.867399926z^{-1} + 0.8822027631z^{-2}}$$

The corresponding difference equation step response and "sampled" exact response are shown in the plot in Figure 2, page 15. The amount of error introduced by using the shorter sample period is almost indiscernible on the graph.

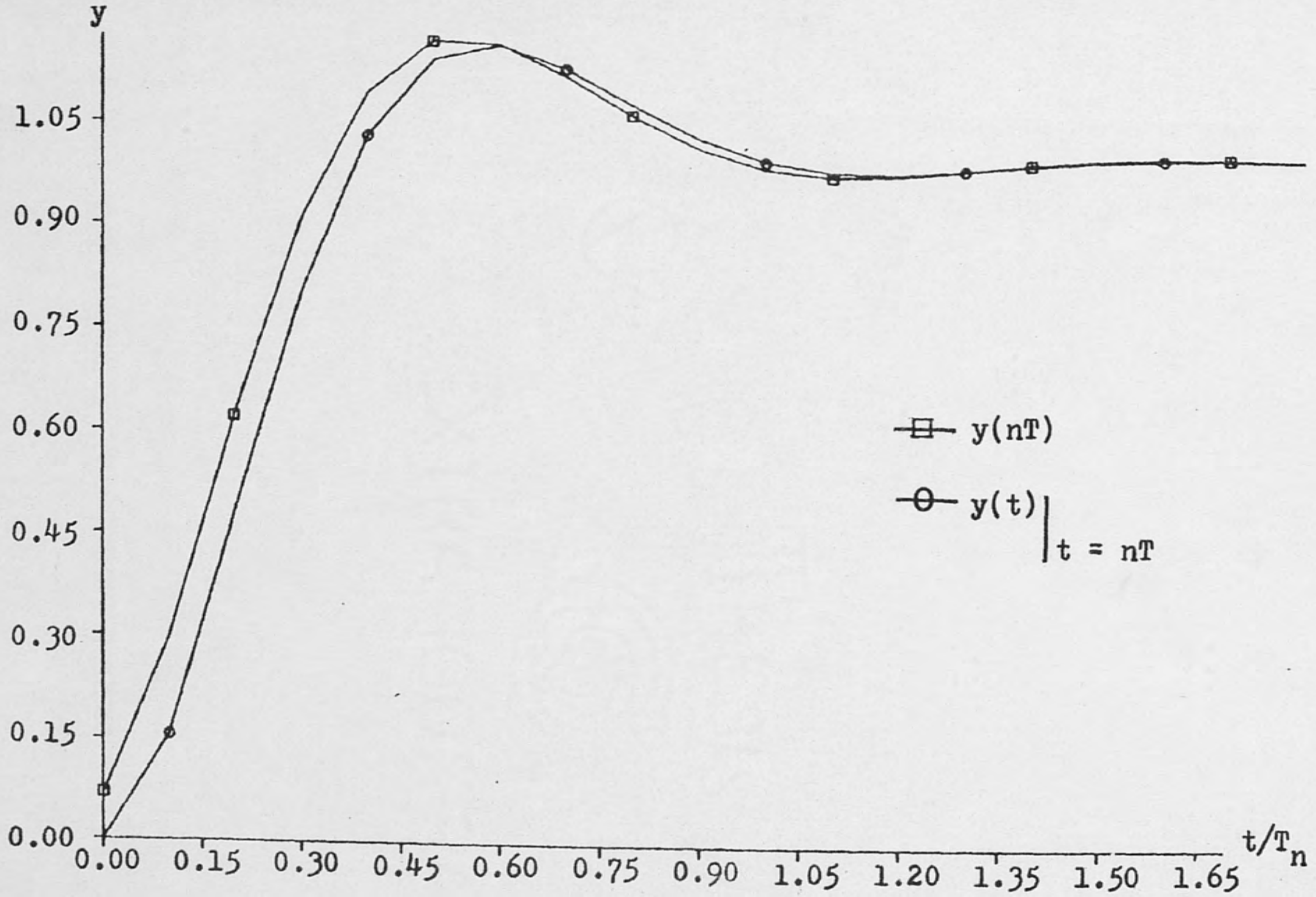

Fig. 1. Step response for sample application model with $T = 0.1T_n$

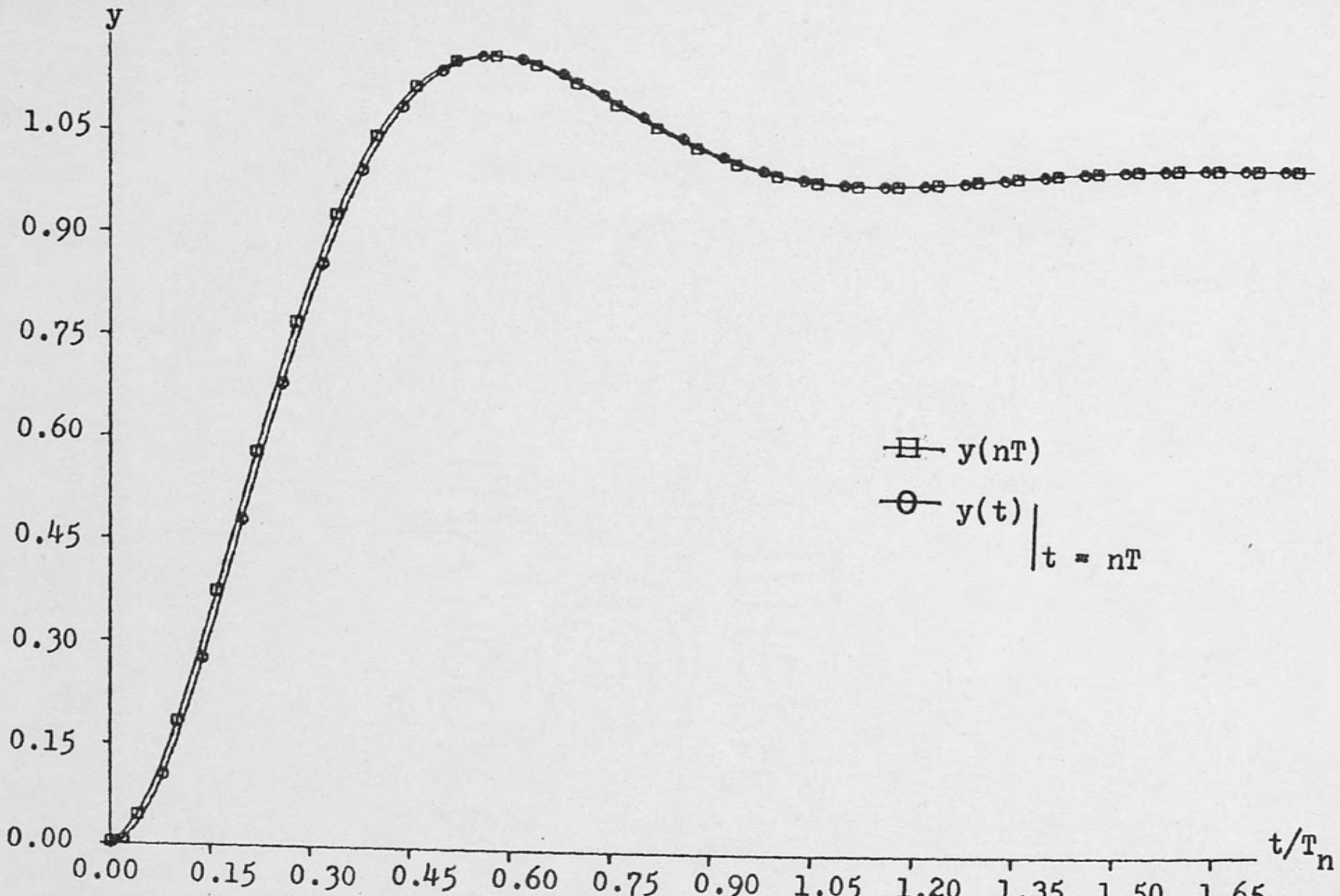

Fig. 2. Step response for sample application model with $T = 0.02T_n$

CONCLUSION

With this algorithm, a means of generating a difference equation solution which approximates the continuous system model response, i.e., the differential equation solution, has been established. The errors introduced (no larger than third order) should be negligible in most cases if the sample time period is chosen small enough with respect to the systems natural frequency period or the magnitude of the systems smallest eigenvalue.

As implemented on the VAX 11/780, BILIN is quite suitable to be used in the design and analysis of digital filters. Since Simons' algorithm is equivalent to the widely used bilinear transformation, BILIN may be used to transform an analog filter design, i.e., an H(s), into a digital filter design, i.e., an H(z), and to observe the digital filter response for a given input and set of initial conditions. This is quite useful since digital filters are used in telecommunications; radar and sonar signal processing; speech, image, and audio signal processing and other related areas.

If BILIN is used in the design of digital filters, the user should be aware that transforming an analog filter into a digital filter using the bilinear transformation is accompanied by a distortion called frequency warping. Except for one matched frequency which is chosen by the designer, critical frequencies such as cutoff, passband, and stopband frequencies will occur at different frequencies for the analog and digital filters. Also, the envelope or group delay of the digital filter will be different than the analog filter as a result of the bilinear transformation. However, if these distortions cannot be tolerated, the analog filter may be "predistorted" in the design of the filter to compensate for either of these phenomena.[3,4]

In general, BILIN may be used as a discrete model of any linear time-invariant system which may be represented by a H(s). If the comments and data input checks were deleted, the system models input and output were read in and output via a FORTRAN loop, and possible other minor modifications were made, BILIN could be used in real-time simulation of linear time-invariant dynamic systems. Also, if BILIN was modified slightly to allow periodic updating of the coefficients of H(s), it could be used in the simulation of linear time-variant systems. In each case, BILIN can be used in place of hardware.

Overall BILIN is easy to use, reasonably accurate, requires little memory, executes quickly, and can easily be adapted to time-variant systems. These characteristics, especially its high efficiency, could allow BILIN to be used for implementing real-time software models. Also, the

H(z) coefficients generated by BILIN provide a numerical specification in the design of digital filters or real-time simulation hardware models. Hence, it can be seen that BILIN is a powerful tool in such areas as systems analysis, digital signal processing, control systems and dynamic system simulation.

APPENDIX

```
!        BILIN ACCEPTS THE COEFFICIENTS OF H(s) AND
!        GENERATES THE APPROXIMATE TIME RESPONSE Y(nT)
!        FOR THE DYNAMIC SYSTEM MODEL.
!        Y(nT) IS OBTAINED AS FOLLOWS:
!         (1) GENERATE H(z) FROM H(s) BY A SERIES OF
!             TRANSFORMATIONS EQUIVALENT TO THE BILINEAR
!             TRANSFORMATION
!         (2) OBTAIN COEFFICIENTS OF DIFFERENCE EQUATION
!             CORRESPONDING TO H(z).
!         (3) GENERATE Y(nT), THE SOLUTION TO THE
!             DIFFERENCE EQUATION, FOR THE CORRESPONDING
!             SYSTEM INITIAL CONDITIONS AND DISCRETE-
!                  TIME INPUT X(nT)
         DIMENSION A(21),AR(21),BR(21),B(21),YnT(210)
      &  ,XnT(210)
!        A(N+1),AR(N+1),BR(N+1),B(N+1) ALLOWS BILIN TO
!        HANDLE UP TO A Nth(20th) ORDER DYNAMIC MODEL
         DOUBLE PRECISION A,AR,B,BR,T,SF,YnT,XnT
         READ (4,*) N,M,T
!        N = ORDER OF DENOMINATOR OF H(s), M = ORDER OF
!        NUMERATOR AND T IS THE SAMPLE TIME PERIOD
         WRITE (6,100) N,M,T
 100     FORMAT(I2,I2,G17.10)
         N1=N+1
         N2=N+2
!        READ IN COEFFICIENTS OF TRANSFER FUNCTION
!        [A(1),...,A(N1)]  AND [B(1),...,B(N1)]  ARE
!        [A0,...,AN] AND  [B0,...BM,BM+1,...,BN]
!         THE DENOMINATOR AND NUMERATOR COEFFICIENTS,
!         RESPECTIVELY, OF H(s)
!         IF M (NUMERATOR ORDER) LESS THAN N , READ IN 0
!         FOR B(M+2) THRU B(N+1 ), i.e., BM+1 &...& BN=0
          READ (4,*) (B(I),I=1,N1),(A(J),J=1,N1)
          WRITE (6,200) (B(I),I=1,N1),(A(J),J=1,N1)
 200        FORMAT(' ',5(5G17.10/))
!         (1)
            SF=2./T
!           SF = SCALE FACTOR
            DO 1 L=1,N
 1          A(L+1)=A(L+1)*SF**L
            DO 2 K=1,M
 2          B(K+1)=B(K+1)*SF**K
!           TRANSLATE NUMERATOR & DENOMINATOR OF H(s) ONE
!           UNIT TO THE RIGHT BY A SERIES OF SYNTHETIC
!           DIVISIONS
            K=N
            DO 33 J=1,N
            DO 3 I=1,K
            A(N1-I)=A(N1-I)-A(N2-I)
 3          B(N1-I)=B(N1-I)-B(N2-I)
 33         K=K-1
!           REVERSE ORDER OF COEFFICIENTS OF NUMERATOR &
            DENOMINATOR OF H(s)
            DO 4 L=1,N1
            AR(L)=A(N2-L)
 4          BR(L)=B(N2-L)
!           TRANSLATE NUMERATOR & DENOMINATOR OF H(s) 1/2
!           UNIT TO LEFT
            K=N
            DO 55 J=1,N
            DO 5 I=1,K
            AR(N1-I)=AR(N1-I)+.5*AR(N2-I)
 5          BR(N1-I)=BR(N1-I)+.5*BR(N2-I)
 55         K=K-1
!           REVERSE ORDER OF COEFFICIENTS AGAIN
            DO 6 L=1,N1
            A(L)=AR(N2-L)
 6          B(L)=BR(N2-L)
```

```
!         MULTIPLY ith  COEFFICIENT BY 2 TO THE ith POWER
          DO 7 I=1,N
          A(I+1)=A(I+1)*2**I
 7        B(I+1)=B(I+1)*2**I
!       (2)
!         PUT H(z) IN "NORMALIZED" FORM BY DIVIDING BOTH
!         THE NUMERATOR & DENOMINATOR OF
!         H(z)=[D0+...+DN*z**(-N)]/[C0+...+CN*z**(-N)]
!         BY THE C0 COEFFICIENT
          DO 8 J=1,N1
          B(J)=B(J)/A(N1)
 8        A(J)=A(J)/A(N1)
          WRITE (6,300) (B(N2-I),I=1,N1)
 300      FORMAT ('0',11G17.10)
          WRITE (6,300) (A(N2-I),I=1,N1)
          READ (4,*) NPT,(YnT(I),I=1,N)
!         NPT = NUMBER OF DISCRETE-TIME OUTPUT VALUES.
!         [YnT(1),...,YnT(N)] ARE RESPECTIVELY,
!         [Y(-NT),...,Y(-T)], THE N INITIAL
!         OUTPUT CONDITIONS
          WRITE (6,400) NPT,(YnT(I),I=1,N)
 400      FORMAT ('0',I3,10G17.10)
          NPTN=NPT+N
          READ (4,*) (XnT(J),J=1,NPTN)
!         [XnT(1),...,XnT(N)] ARE RESPECTIVELY
!         [X(-NT),...,X(-T)],THE N INITIAL INPUT
!         CONDITIONS AND [XnT(N1),...,XnT(NPTN)] ARE
!         RESPECTIVELY, [ X(0),...,X[(NPT-1)T]]
          WRITE (6,500) (XnT(I),I=1,NPTN
 500      FORMAT ('0',25(5G17.10/))
          DO 9 L=N1,NPTN
  9       YnT(L)=0.
!       (3)
          SOLVE DIFFERENCE EQUATION FOR Y(nT), n=0,NPTN-1
```

```
        DO 16 K=N1,NPTN
!       INITIALIZE XnT(N1) TO "PRESENT" INPUT, X(nT)
        XnT(N1)=XnT(K)
!       SOLVE FOR OUTPUT,Y(nT),CORRESPONDING TO
!       "PRESENT" INPUT
        DO 12 I=1,N1
 12     YnT(K)=YnT(K)+B(I)*XnT(I)
        DO 13 J=1,N
 13     YnT(K)=YnT(K)-A(J)*YnT(J)
!       "MOVE FORWARD" IN TIME ONE SAMPLE PERIOD
!       LET X((n-1)T)=X(nT)
        DO 14 L=1,N
 14     XnT(L)=XnT(L+1)
!       LET Y((n-1)T)=Y(nT)
        DO 15 M=1,N-1
 15     YnT(M)=YnT(M+1)
!       LET Y((n-1)T)=Y(nT), THE LAST CALCULATED OUTPUT
 16     YnT(N)=YnT(K)
        WRITE (6,500) (YnT(I),I=N1,NPTN
        STOP
        END
```

REFERENCES CITED

1. Simons, F. O., and Harden, R. C. "Differential Equation Solutions for Up to 10th Order System Theory Models with HP-67 Compulators." In Proceedings of the Eleventh Annual Southeastern Symposium on System Theory, pp. 49-53. Clemson, SC: Clemson University, March 1979.

2. D'Azzo, J. J., and Houpis, L. H. Feedback Control System Analysis and Synthesis. New York: McGraw-Hill, 1966.

3. Oppemheim, Alan V., and Schafer, Ronald W. Digital Signal Processing. Englewood Cliffs, NJ: Prentice Hall, 1975.

4. Tretter, Steven A. Introduction to Discrete Time Signal Processing. New York: John Wiley, 1976.