* * * * *

PROJECT PROPOSAL FOR EE 507

LINEAR SYSTEM IDENTIFICATION BY USING RANDOM PROCESS

Benmei Chen

P.O.BOX 241, GONZAGA U.

SPOKANE, WA 99258

MARCH 31, 1987

LINEAR SYSTEM IDENTIFICATION BY USING RANDOM PROCESS

I. INTRODUCTION TO THE PROBLEM

System identification is a very important procedure in some automatic control or digital control designs. This project is to identify the unknown stable linear systems by applying the results from random process, and using the digital technique.

Consider the following system, h(t), with the input x(t) and the output y(t):

Let the input x(t) be a white noise. Then, using the results from random process, we can determine h(t) from x(t) and y(t).

In order to see how this design works, we can apply it to some given systems, such as h(t) = u(t) or h(t) = exp(-at) u(t), where u(t) is the unit step function. And we can also compare the computed results to the exact systems.

II. SOLUTION TO THE PROBLEM

Recall the results in EE 507 notes, the input-output cross-correlation function of the system above is given as

$$\begin{aligned} Rxy(\tau) &= E[\ x(t)\cdot y(t+\tau)\ ] \\ &= E[\ x(t) \int_0^\infty x(t+\tau-a)\cdot h(a)\ da\ ] \\ &= \int_0^\infty E[\ x(t)\cdot x(t+\tau-a)\ ]\cdot h(a)\ da \\ &= \int_0^\infty Rx(\tau-a)\cdot h(a)\ da \qquad \ldots\ldots \qquad (1) \end{aligned}$$

If the input x(t) is a white noise with the autocorrelation function

$$Rx(\tau) = Q\,\delta(\tau)$$

then the Equation (1) becomes

$$\begin{aligned} Rxy(\tau) &= \int_0^\infty Q\,\delta(\tau-a)\cdot h(a)\ da \\ &= Q\ h(\tau) \end{aligned}$$

For a large T, we have

$$\begin{aligned} h(\tau) &= Rxy(\tau)\ /\ Q \\ &\cong [\ \int_0^T x(t)\cdot y(t+\tau)\ dt\ ]\ /\ (QT) \\ &= [\ \int_0^T x(t-\tau)\cdot y(t)\ dt\ ]\ /\ (QT) \end{aligned}$$

Sampling both the input and output of the system by the time period of Δt, it is easy to see that

$$h(\tau) \cong [\ \sum_{K=1}^{T/\Delta t} x(k\cdot\Delta t-\tau)\cdot y(k\cdot\Delta t)\cdot\Delta t\ ]\ /\ (QT) \qquad \ldots \qquad (2)$$

Now we can write a computer program for the Equation (2) to obtain the discrete values of h(t). Furthermore, we can find out the z-transfer function of the system as well.

One of the key problems for this project is to generate the white noise. Recalling the results in Sections 6.1 to 6.3 in the text, we know that the process Z(t) is an idealized model (for t > 0) for a thermal-noise (we can treat it as a white noise). And Z(t) is the derivative of the Wiener process and is defined as

$$Z(t) \triangleq \lim_{\Delta t \to 0} Z_{\Delta}(t)$$

and

$$Z_{\Delta}(t) \triangleq \sum_{n=0}^{\infty} Zn \cdot \Delta w \cdot \delta(t-n\Delta t), \qquad t>=0,$$

$$\text{Prob}\{ Zn=+1 \} = \text{Prob}\{ Zn=-1 \} = 0.5$$

where Δt and Δw are known as constants and $\delta(.)$ is the inpluse function. Zn is the discrete-time ± 1 Bernoulli process.

Since the RND function (avarible almost in any computer) is uniformly destributed on [0,1], we can use it to generate the discrete-time Bernoulli process and then the white noise.

III. LIMITATION

According to the sampling theorem, the maximum time internal of sampling is 1 / (2B), where B is the bandwidth of the signal. However in this particular problem we did, the bandwidth of the white noise is infinite if it is ideal. And the bandwidth of the system h(t) is unknown. So, the computed results of Equation (2) may be unreasonable for some systems.

IV. ESTIMATION OF EFFORT

I think this project will require more than 40 hours of effort for

(1) writing and debudding the program.

(2) applying the project to some systems.

(3) writing the project report.

V. COMPLETION DATE

April 16, 1987

* * * * *

THANK YOU !

Benmie,
you could check the "whiteness" of your simulated "white" noise by checking its autocorrelation.
Looks like a very practical little project. Good.
JoeF

90%

* * * * *

PROJECT REPORT FOR EE 507

LINEAR SYSTEM IDENTIFICATION BY USING RANDOM PROCESS

Benmei Chen

P.O. Box 241, GONZAGA U.

SPOKANE, WA 99258

APRIL 21, 1987

Good Benmei
B+ for paper
A for course

LINEAR SYSTEM IDENTIFICATION BY USING RANDOM PROCESS

ABSTRACT -- A method to identify the linear system by employing the sampled-hold white noise has been introduced and two particular systems have been checked.

I. INTRODUCTION

CONSIDER the following linear time-invariant system [1] in Fig.1,

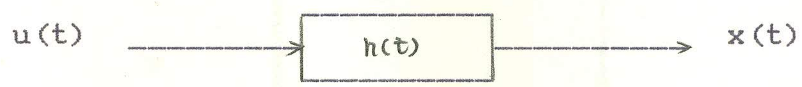

Fig. 1 The linear time-invariant system

where $u(t)$ is the input signal, $x(t)$ is the output and $h(t)$ is the impluse response of the system. Assume both $h(t)$ and $u(t)$ are causal, it can be obtained that [2],

$$\begin{aligned} x(t) &= \int_0^\infty u(\tau) \cdot h(t-\tau)\, d\tau \\ &= \int_0^t u(\tau) \cdot h(t-\tau)\, d\tau \\ &= \int_0^t u(t-\tau) \cdot h(\tau)\, d\tau \qquad \ldots \qquad (1) \end{aligned}$$

The impluse response of system, $h(t)$ may be determined by applying an impluse signal to the input. However, it is difficult to have an ideal impluse signal in real world,and it is difficult to process the procedure in computer too.

This project is introducing a method to identify the system by using the random process and the computing technique.

II. SOLUTION

§1. The theoretical solution:

Recall the result in notes of EE 507, the input-output cross-correlation function of the system [1] is given by

$$
\begin{aligned}
Rux(\tau) &= E[\ u(t)\ x(t+\tau)\ ] \\
&= E[\ u(t)\cdot\int_0^\infty u(t+\tau-a)\cdot h(a)\ da\ ] \\
&= \int_0^\infty E[\ u(t)\cdot u(t+\tau-a)\ ]\cdot h(a)\ da \\
&= \int_0^\infty Ru(\tau-a)\cdot h(a)\ da \qquad \ldots \qquad (2)
\end{aligned}
$$

If the input u(t) is a white noise with an autocorrelation function,

$$Ru(\tau) = Q\cdot\delta(\tau)$$

where Q is a constant and $\delta(.)$ is an impluse function, then Equation (2) becomes,

$$
\begin{aligned}
Rux(\tau) &= \int_0^\infty Q\cdot\delta(\tau-a)\cdot h(a)\ da \\
&= Q\cdot h(\tau) \qquad \ldots \qquad (3)
\end{aligned}
$$

From the equation (3), it is easy to see,

$$h(\tau) = Rux(\tau)/Q \qquad \ldots \qquad (4)$$

In practice, however, we won't be able to have an ideal white noise with an impluse autocorrelation function. And it is impossible to process the idael white noise in computer as well.

Following, we introduce a continous-time-discrete-magnitude white process (see Figure 2 below). We may call it by the sampled-hold white process because it can be obtained by sampling the ideal white noise and holding the value for a time period ts.

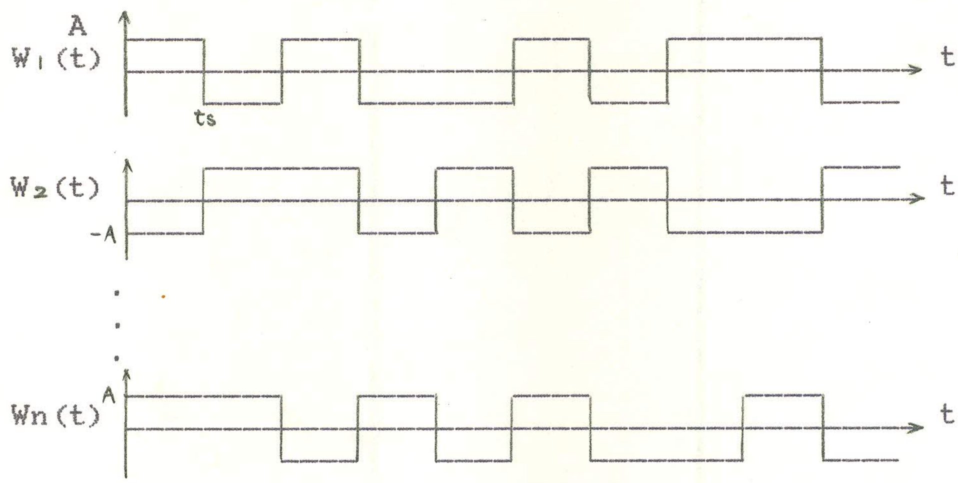

Figure 2: The sampled-hold white process

The mean and the autocorrelation function of the sampled-hold white process are known as [1],

$$E[W(t)] = 0$$

$$Rw(\tau) = \begin{cases} A^2(\tau/ts+1) & -ts < \tau < =0 \\ A^2(-\tau/ts+1) & 0 < \tau < ts \\ 0 & \text{elsewhere} \end{cases}$$

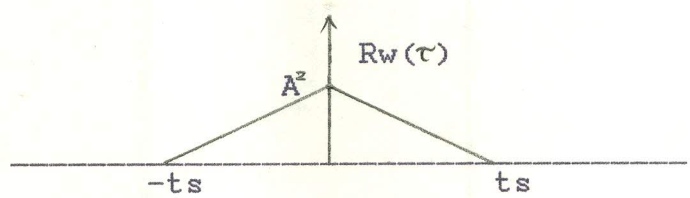

Figure 3: The autocorrelation function of sampled-hold white process

Now, we can apply the sampled-hold white process as the input of the system in Figure 1, from Equation (2), that is

$$Rux(\tau) = \int_0^{\infty} Ru(\tau-a)\cdot h(a)\ da$$

$$= \int_{\tau-ts}^{\tau+ts} Rw(\tau-a)\cdot h(a)\ da \qquad \ldots \qquad (5)$$

If ts is small enough and h(a) is a continous function, then $h(\tau)$ can be regarded as a constant h_τ in the internal $[\tau-ts,\ \tau+ts\ ]$. So that the equation (5) can be written as,

$$Rux(\tau) \cong h_\tau\cdot\int_{\tau-ts}^{\tau+ts} Rw(\tau-a)\cdot da$$

$$\cong h_\tau\cdot ts\cdot A^2 \qquad \ldots \qquad (6)$$

From Eq.(6)

$$h_\tau \cong Rux(\tau)\ /\ (A^2\cdot ts) \qquad \ldots \qquad (7)$$

Assume the u(t) and x(t) are ergodic, then for a large T,

$$h_\tau \cong [\int_0^T u(t)\cdot x(t+\tau)\cdot dt\ ]\ /\ (A^2\cdot ts\cdot T)$$

$$= [\int_0^T u(t-\tau)\cdot x(t)\cdot dt\ ]\ /\ (A^2\cdot ts\cdot T) \qquad \ldots \qquad (8)$$

The intergration in Equation (8) can be approximated by

$$h_\tau \cong [\sum_{K=0}^{T/\Delta t} u(k\cdot\Delta t-\tau)\cdot x(k\cdot\Delta t)\cdot\Delta t\ ]\ /\ (A^2\cdot ts\cdot T)$$

Let $\Delta t=ts$, we have

$$h_\tau \cong [\sum_{K=0}^{T/\Delta t} u(k\cdot ts-\tau)\cdot x(k\cdot ts)\ ]\ /\ (A^2\cdot T) \qquad \ldots \qquad (9)$$

The Equation (9) is easy to be processed in computer.

$2. The practical solution:

Here, two computing algorithms are presented. One is to generate the sampled-hold white process W(t), and the other is to identify the system impluse response h(t).

(1) An algorithm for creating sampled-hold white process W(t) with sampling period ts and time internal [0, T],

```
For  t=0 sec.  to  T sec.  step  ts sec.  Do
   If  RND<.5  then
       W=A ( a constant, the magnitude of W(t) )
   else
       W=-A
   End  if
   W(t)=W
End  of  i  loop
```

In this algorithm, the RND is a random variable uniformly distributed on [0, 1). It is almost avairable in all computers.

(2) The algorithm to obtain the discrete system impluse response h(τ) for a sampling period ta and 0 < t < To .

```
For τ=0 sec.  to  To sec.  step  ta sec.  Do
    h(τ) = 0  ( initial )
    For  t=τ sec. to  T sec.  step  ts sec.  Do
       h(τ) = h(τ) + u(t)* x(t-τ)
```

```
        End  of  t  loop
        h(τ) = h(τ) / (A^2 * T)
    End  of  τ  loop
```

This algorithm is corresponding to the Equation (9). The general computer program in VAX-11 FORTRAN is given in Appendix A.

III. RESULTS

In order to check how this method works, we applied it to the following two particular systems:

$$Ha(s) = (2s + .8) / (s + .8s + .5) \quad \ldots \quad (S1)$$

and

$$Hc(s) = (2s - .8) / (s - .8s + 1.16) \quad \ldots \quad (S2)$$

Take the inverse Laplace transform of both two systems, we have the impluse system responses,

$$ha(t) = \exp(-.3t) + \exp(-.5t) \quad \ldots \quad (H1)$$

and

$$hc(t) = 2 \exp(-.4t) \cos(t) \quad \ldots \quad (H2)$$

Recall the Equation (1) in part I,

$$x(t) = \int_0^t u(\tau) \cdot h(t-\tau) \cdot d\tau$$

Let $t = n \cdot ts$,

$$x(t) = \int_0^{ts} u(\tau)\cdot h(t-\tau)\ d\tau + \int_{ts}^{2ts} u(\tau)\cdot h(t-\tau)\cdot d\tau$$

$$+ \ldots + \int_{(n-1)ts}^{nts} u(\tau)\cdot h(t-\tau)\cdot d\tau$$

$$= u_0\cdot\int_0^{ts} h(t-\tau)\cdot d\tau + u_1\cdot\int_{ts}^{2ts} h(t-\tau)\cdot d\tau$$

$$+ \ldots + u_{n-1}\cdot\int_{(n-1)ts}^{n\cdot ts} h(t-\tau)\cdot d\tau$$

$$= u_0\cdot[H(nts)-H((n-1)\cdot ts)] + u_1\cdot[H((n-1)ts)-H((n-2)\cdot ts)]$$

$$+ \ldots + u_{n-1}\cdot[H(ts)-H(0)] \quad \ldots\ldots \quad (10)$$

where u_K , k=0, 1, ..., n-1, is hold to be a constant within the sampled period and H(.) is the original function of h(t), $H'(t) = h(t)$.

From the Equation (10), we can obtain the exact output of the systems if we know the original function of h(t).

Apply this and the algorithms we had in part II to the systems (S1) and (S2), we got very reasonable results (see Fig. 4-7 and the data in Appendix B).

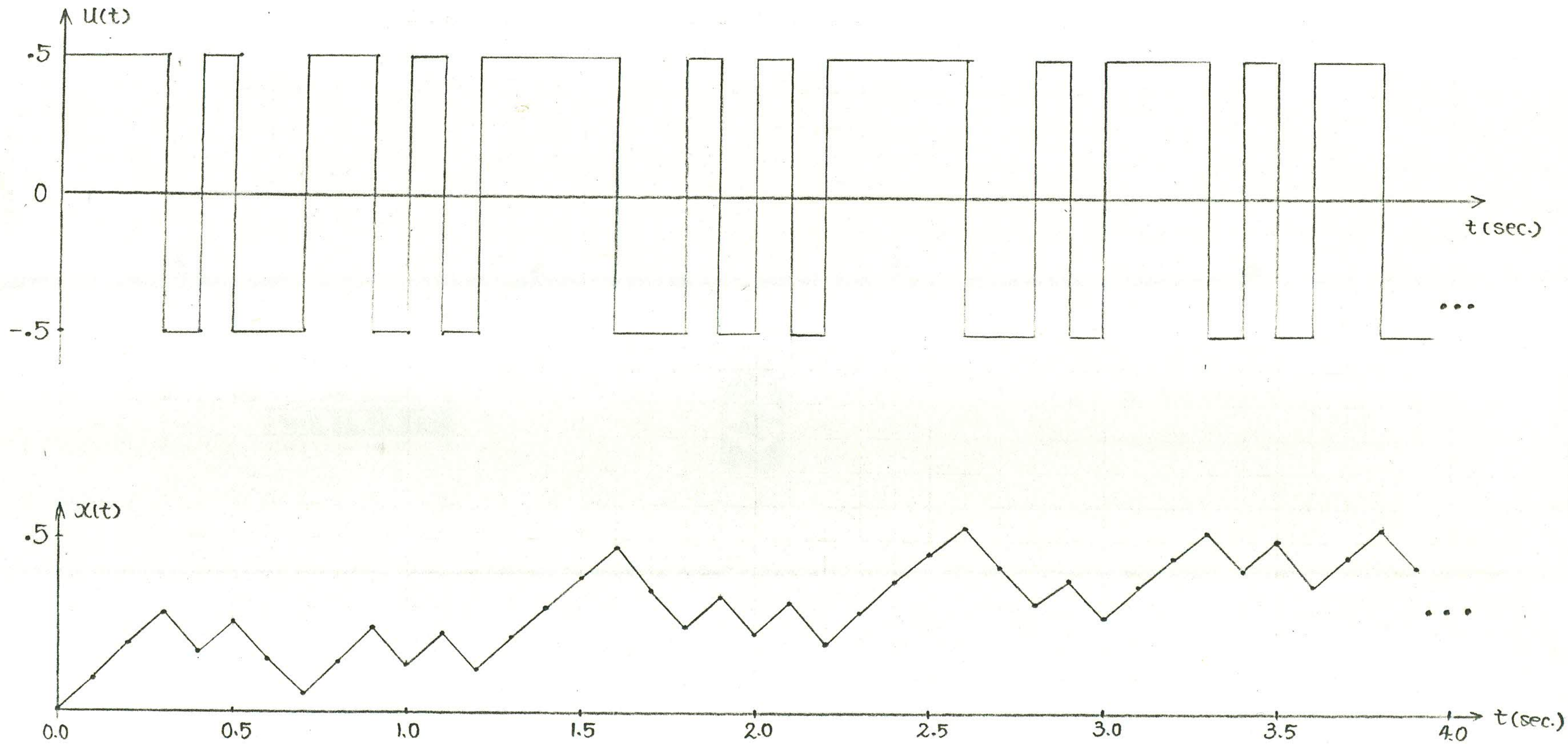

Figure 4: The input and output signals for system (S1)

$$Ha(s) = (2s+.8)/(s^2+.8s+.5)$$

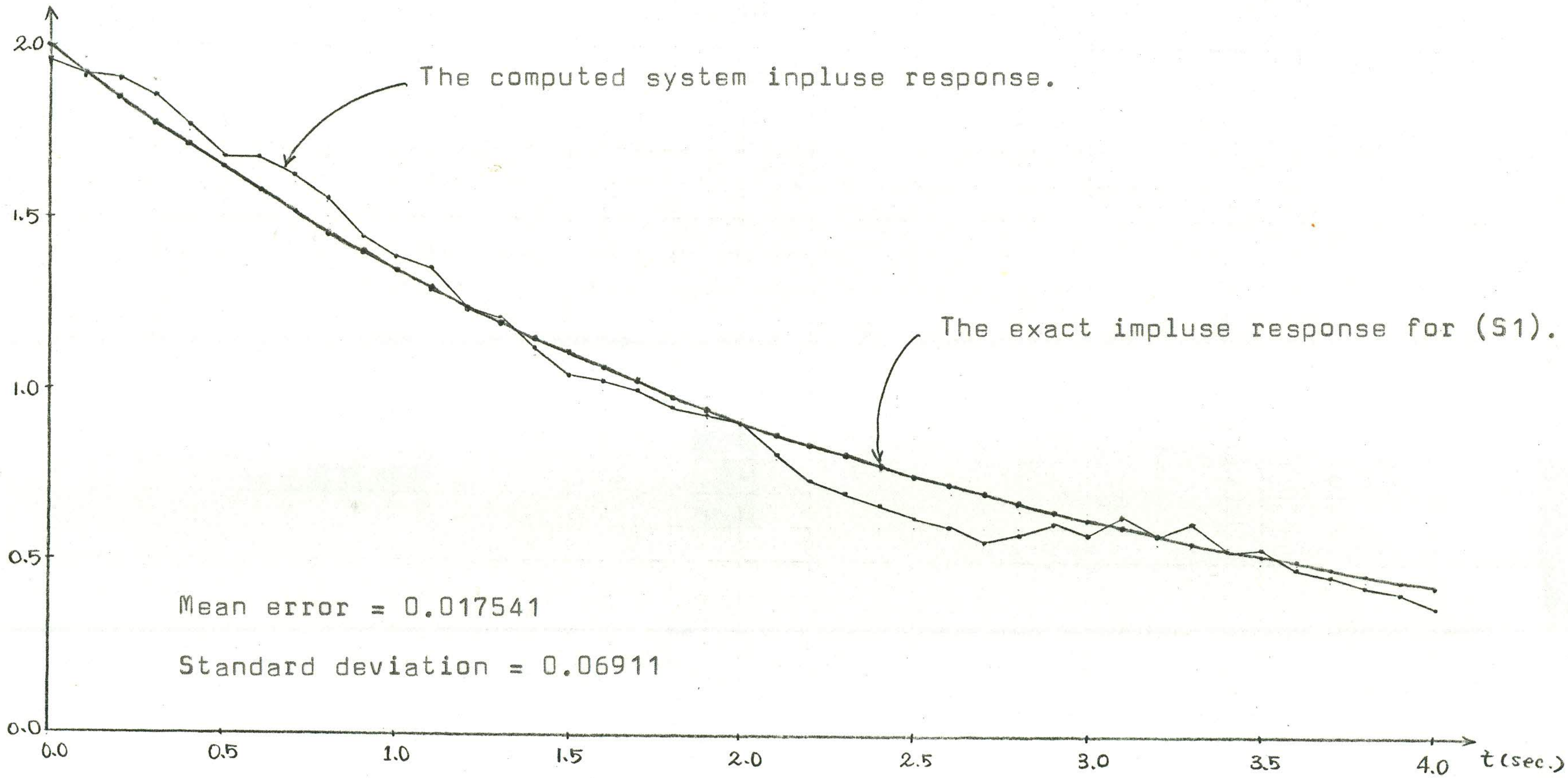

Figure 5: The exact and computed impluse responses for system (S1).

$$Ha(s) = (s+.8)/(s^2+.8s+.5)$$

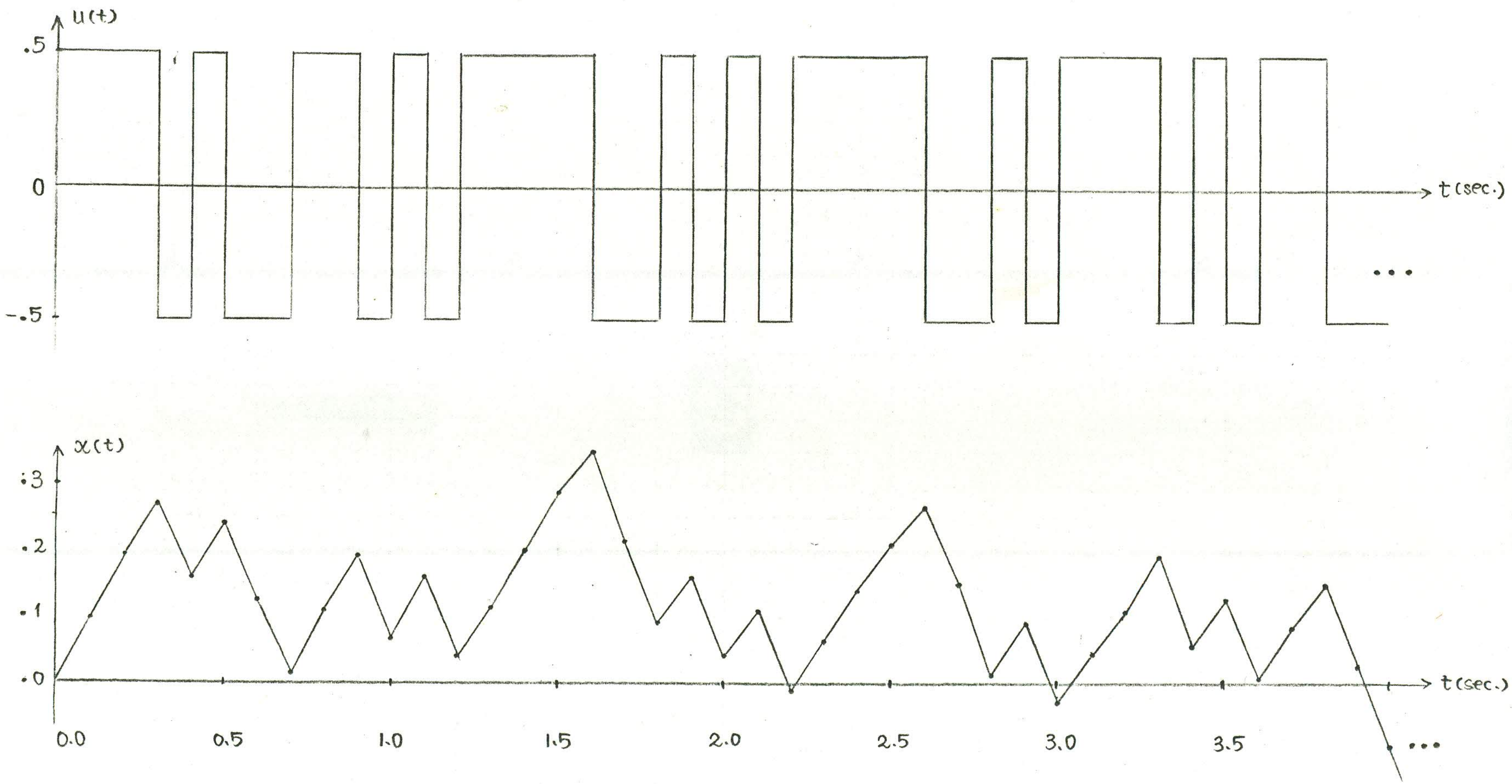

Figure 6: The input and output signals for (S2)

$$Hc(s) = (2s-.8)/(s^2-.8s+1.16)$$

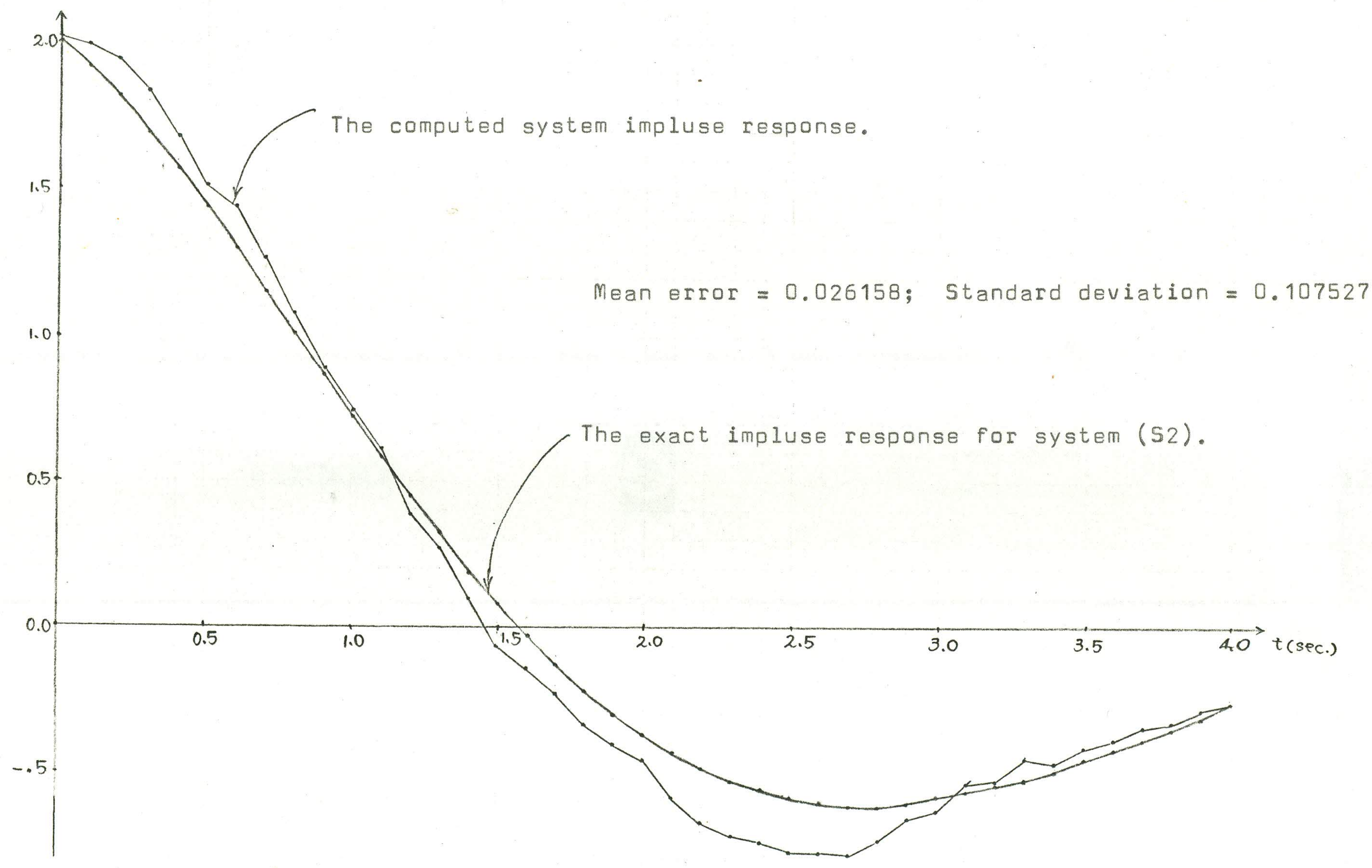

Figure 7: The exact and computed impluse responses for system (S2)

VI. APPLICATIONS

The main application of this project is to identify the systems. However, we can use this project to construct a transversal filter corresponding to the interested systems too. Recall the Equation (1) in part I again,

$$x(t) = \int_0^t u(t-\tau)\cdot h(\tau)\, d\tau$$

The equation above can be also appoximated by

$$x(t) = \sum_{K=0}^{t/ts} u(t-k\cdot ts)\cdot h(k\cdot ts)\cdot ts$$

This result can be realized by using a delay line with taps at the delays $k\cdot ts$; the output of each tap is multiplied by the preset weight $h(k\cdot ts)\cdot ts$, where $h(k\cdot ts)$ is obtained from the system identification and ts is given. Such a system is shown in Figure 8.

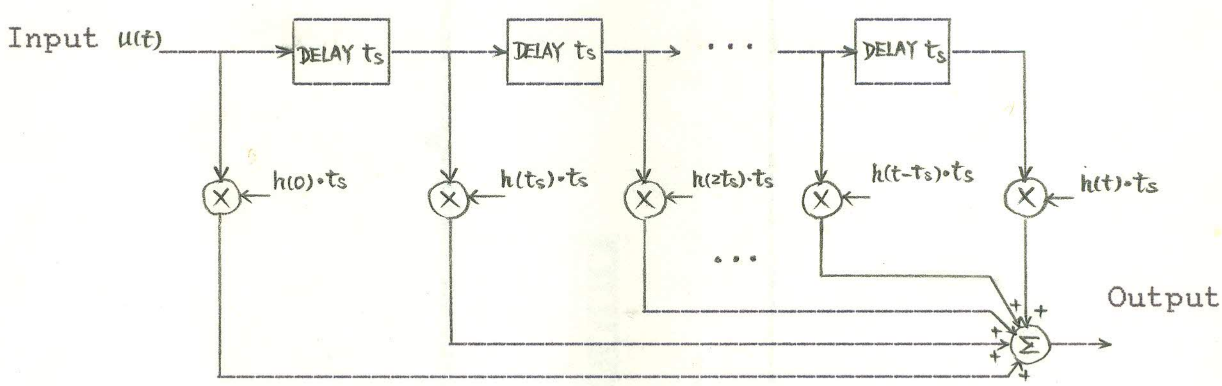

Figure 8: A tapped-delay-line (transversal) filter

A filter constructed using a tapped delay line, tap weights, and adder in the configuration shown in Figure 8 is called a transversal filter.

V. CONCLUSIONS

In this project the sampled-hold white process had been defined and applied to identify the linear systems. Two systems had been checked turning out with nice results. And a application to construct the transversal filter had been introduced as well.

REFERENCES

[1] Dr. Feeley, " Notes of EE 507 "; University of Idaho, 1987

[2] Ferrel G. Stremler, " Introduction to Communication Systems"; New York, 1981

[3] William A. Gardner, " Introduction to Random Processes "; 1986

[4] Arthur Gelb, " Applied Optimal Estimation "; 1974

APPENDIX A: THE FORTRAN PROGRAM IN VAX-11 TO IDENTIFY SYSTEMS.

```
C       *********************************************
C       *      THE LINEAR SYSTEM IDENTIFICATION      *
C       *********************************************
C
        DIMENSION H(40), U(2500), X(2500)
        DO  10  I=55555555, 80000000, 10000
        Y = RAN(K)
        IF  ( Y .LT. .5 ) THEN
             Z=.5
        ELSE
             Z=-.5
        END IF
        M = ( I-55555555)/10000
        U(M) = Z
10      CONTINUE
        X(0) = 0
        DO  30  I=1, 2500
          X(I) = 0
          M = I
          DO  20  J=1, M
            F1 = -EXP(-.03*J)/.3 - EXP(-.05*J)/.5
            F2 = -EXP(-.03(J-1))/.3 - EXP(-.05*(J-1))/.5
            X(I) = X(I) + U(I-J)*(F1-F2)
 20       END DO
 30     CONTINUE
 C      PRINT OUT THE INPUT AND OUTPUT SIGNALS.
 40     FORMAT ( 3F16.8 )
        DO  50  I=0, 40
           PRINT 100, .1*I, U(I), X(I)
 50     CONTINUE
        DO  70  I=1, 40
          H(I) = 0
          M = I+1
          DO  60  K=M, 2500
            H(I)=H(I)+U(K-I-1)*X(K)
 60     END DO
        H(I) = H(I)/(.25*250)
 70     CONTINUE
 80     FORMAT ( 4F16.8 )
        DO  90  I=1,40
          A=EXP(-.03*I)+EXP(-.05*I)
          B=A-H(I)
          P=P+B
          Q=Q+B*B
          T=.1*I
          PRINT 80, T, A, H(I), B
 90     CONTINUE
        PRINT 100, P/40, Q/40
 100    FORMAT ( 2F16.8 )
        END
```

APPENDIX B: THE DATA FOR SYSTEMS (S1) & (S2)

(1) The input and output for system (S1) & (S2)

TIME (sec.)	INPUT u(t)	(S1) OUTPUT	(S2) OUTPUT
0.10	0.500000	0.000000	0.000000
0.20	0.500000	0.098028	0.095008
0.30	0.500000	0.192222	0.180123
0.40	-.500000	0.282740	0.255598
0.50	0.500000	0.173679	0.131799
0.60	-.500000	0.264965	0.209023
0.70	-.500000	0.156639	0.087498
0.80	0.500000	0.052571	-.022629
0.90	0.500000	0.148641	0.068583
1.00	-.500000	0.240955	0.151068
1.10	0.500000	0.133602	0.034900
1.20	-.500000	0.226540	0.120126
1.30	0.500000	0.119786	0.006761
1.40	0.500000	0.213279	0.094716
1.50	0.500000	0.303119	0.174197
1.60	0.500000	0.389460	0.245106
1.70	-.500000	0.472444	0.307773
1.80	-.500000	0.356156	0.172543
1.90	0.500000	0.244450	0.049656
2.00	-.500000	0.333194	0.129113
2.10	0.500000	0.222421	0.010863
2.20	-.500000	0.312058	0.094819
2.30	0.500000	0.202139	-.019127
2.40	0.500000	0.292595	0.068896
2.50	0.500000	0.379516	0.148764
2.60	0.500000	0.463052	0.220519
2.70	-.500000	0.543340	0.284310
2.80	-.500000	0.424461	0.150432
2.90	0.500000	0.310265	0.029051
3.00	-.500000	0.396617	0.110114
3.10	0.500000	0.283543	-.006477
3.20	0.500000	0.370968	0.079151
3.30	0.500000	0.454979	0.156874
3.40	-.500000	0.396617	0.226717
3.50	0.500000	0.417267	0.098829
3.60	-.500000	0.499535	0.173303
3.70	0.500000	0.382543	0.050173
3.80	0.500000	0.466208	0.129422
3.90	-.500000	0.546611	0.201012
4.00	-.500000	0.427833	0.075020

APPENDIX B: THE DATA FOR SYSTEMS (S1) & (S2)

(2) The Exact and computed impluse response for system (S1)

TIME (sec.)	EXACT h(t)	COMPUTED h(t)	ERROR
0.00	2.000000	1.959443	0.040557
0.10	1.921675	1.907846	0.013829
0.20	1.846602	1.899722	-.053120
0.30	1.774639	1.847401	-.072762
0.40	1.705651	1.770145	-.064494
0.50	1.639509	1.680864	-.041355
0.60	1.576088	1.686978	-.110890
0.70	1.515272	1.621921	-.106649
0.80	1.456948	1.544329	-.087381
0.90	1.401008	1.441172	-.040164
1.00	1.347349	1.394400	-.047051
1.10	1.295874	1.361603	-.065730
1.20	1.246488	1.241553	0.004935
1.30	1.199103	1.214590	-.015487
1.40	1.153632	1.119249	0.034383
1.50	1.109995	1.037584	0.072411
1.60	1.068112	1.033676	0.034436
1.70	1.027911	1.005929	0.021981
1.80	0.989318	0.959700	0.029618
1.90	0.952267	0.934910	0.017357
2.00	0.916691	0.904056	0.012635
2.10	0.882530	0.806194	0.076336
2.20	0.849722	0.736199	0.113523
2.30	0.818213	0.699530	0.118683
2.40	0.787946	0.670090	0.117856
2.50	0.758871	0.620324	0.138547
2.60	0.730938	0.597112	0.133826
2.70	0.704098	0.558327	0.145771
2.80	0.678308	0.576432	0.101876
2.90	0.653522	0.611852	0.041670
3.00	0.629700	0.581083	0.048617
3.10	0.606802	0.633850	-.027048
3.20	0.584789	0.590923	-.006134
3.30	0.563627	0.608492	-.044865
3.40	0.543278	0.528905	0.014373
3.50	0.523712	0.533499	-.009787
3.60	0.504894	0.492902	0.011992
3.70	0.486796	0.468780	0.018016
3.80	0.469388	0.429416	0.039972
3.90	0.452641	0.409610	0.043031
4.00	0.436530	0.370649	0.065881

mean error = 0.017541 standard deviation = 6.911241E-02

APPENDIX B: THE DATA FOR SYSTEMS (S1) & (S2)

(3) The Exact and computed impluse response for system (S2)

TIME (sec.)	EXACT h(t)	COMPUTED h(t)	ERROR
0.00	2.000000	2.067218	-.067218
0.10	1.911979	1.994654	-.082675
0.20	1.809431	1.944903	-.135472
0.30	1.694615	1.835174	-.140559
0.40	1.569753	1.689932	-.120179
0.50	1.437008	1.518912	-.081904
0.60	1.298464	1.434804	-.136340
0.70	1.156111	1.275768	-.119657
0.80	1.011826	1.096838	-.085012
0.90	0.867365	0.892931	-.025566
1.00	0.724351	0.742459	-.018108
1.10	0.584265	0.610401	-.026136
1.20	0.448442	0.394151	0.054291
1.30	0.318067	0.272902	0.045165
1.40	0.194174	0.092088	0.102086
1.50	0.077643	-.070643	0.148286
1.60	-.030793	-.147355	0.116561
1.70	-.130550	-.236387	0.105837
1.80	-.221182	-.334957	0.113775
1.90	-.302383	-.402862	0.100478
2.00	-.373974	-.467548	0.093574
2.10	-.435895	-.589816	0.153921
2.20	-.488201	-.677538	0.189338
2.30	-.531047	-.722059	0.191011
2.40	-.564686	-.746970	0.182284
2.50	-.589449	-.785974	0.196526
2.60	-.605743	-.786904	0.181162
2.70	-.614038	-.795346	0.181308
2.80	-.614856	-.742403	0.127546
2.90	-.608764	-.665871	0.057107
3.00	-.596360	-.647548	0.051188
3.10	-.578268	-.541427	-.036841
3.20	-.555126	-.530746	-.024380
3.30	-.527581	-.457152	-.070429
3.40	-.496278	-.479008	-.017271
3.50	-.461855	-.412421	-.049434
3.60	-.424934	-.393095	-.031839
3.70	-.386119	-.356956	-.029163
3.80	-.345988	-.333277	-.012711
3.90	-.305089	-.290464	-.014625
4.00	-.263937	-.270497	0.006560

mean error = 0.026158 standard deviation = .1075275