

ALGORITHMS AND COMPUTER METHODS IN DIGITAL SIGNAL PROCESSING

S. Hossein Mousavinezhad, Professor & Chairperson, and
Dean R. Johnson, Associate Professor
Department of Electrical and Computer Engineering
Western Michigan University
Kalamazoo, Michigan
h.mousavinezhad@wmich.edu
dean.johnson@wmich.edu
(616) 387-4057
FAX (616) 387-4024

I. INTRODUCTION

Digital Signal Processing (DSP) is a growing and important subject area in the field of electrical/computer engineering. DSP applications can be found in areas such as speech recognition, image enhancement, biomedical instrumentation and automotive electronics. At Western Michigan University, we were one of the first schools to introduce a DSP course at the undergraduate level. Presently we offer two courses, one at the graduate level. A graduate course on image (or two-dimensional signal) processing is under development. The subject of signals and systems (and the DSP) are heavily dependent on mathematical concepts such as convolution, difference/differential equations, design of filters, transform methods and power spectrum. With the availability of such “powerful” tools like: MATHCAD, MATLAB, MATHEMATICA, MAPLE, LABVIEW, engineering educators need to take special care in where and when to introduce these engineering software packages to students. Historically, signal processing has been a subject where students have great difficulty in understanding the underlying principles and visualizing often abstract concepts which are fundamental to this important and growing area.

Authors have had experience in teaching courses and conducting research in the areas of signals & linear systems and digital signal processing and welcome this opportunity to share these ideas with engineering education community. In Section II we show specific examples of filter design methods and solution of difference equations using MATLAB and MATHCAD. Section III introduces algorithms developed using MATHEMATICA to solve n-th order ordinary differential equations (ODEs). In section IV we make some concluding remarks and propose further work. The paper ends with a discussion of the appropriate use of these computer tools and algorithms at the right place and the right time so that students using them will not be “just pushing buttons.” It is noted that many current textbooks use these tools (see, for example, [1]) and one needs to

be careful that students will learn the main and underlying ideas for the subject matter under study before they are exposed to these computer methods.

II. DSP Examples

Among electrical/computer engineering subjects, possibly linear systems theory and digital signal processing (including image processing and pattern recognition) are those usually identified by students to contain a lot of mathematical concepts some, in their opinion, being “pure math”. At our school, the ECE programs start with a study of mathematics and basic sciences, courses on digital logic and electric circuits, followed by courses on microprocessor engineering, computer science and other courses including those specific to the student’s chosen major (electrical or computer engineering). The Signals & Systems track include courses in linear systems, random signals and processes, communication systems, feedback control systems, and digital signal processing. Some advanced undergraduate students, in preparation for their graduate studies, take further 500-level courses in digital controls and advanced DSP. In this section we show some examples of algorithms and use of MATHCAD and MATLAB in solving signal processing problems encountered in signals & systems courses.

EXAMPLE II.1

In this example we show the solution of a difference equation for a causal linear time (shift) invariant discrete-time system represented by the following linear constant coefficient difference equation (LCCDE):

$$y(n) - 3y(n-1) + 2y(n-2) = 6x(n) - 10x(n-1) + 2x(n-2) ,$$

for input $x(n) = \delta(n)$.

Note that the solution is the impulse response (IR) of the system and is important in the study of filters and control systems. We begin the solution by the following recursive algorithm which is derived from the given LCCDE and noting that because the system is causal, $h(-1)=h(-2)=0$ and also the fact that $\delta(n-n_0)=0$ except for $n=n_0$ where it is unity.

Put $n=0$ in LCCDE: $h(0) = 6$, similarly:

$h(1) = 8$; $h(2) = 14$; $h(3) = 26$; $h(4) = 50$. These will be used as a kind of initial conditions (IC) in the problem.

Note that for $n > 2$, we have a homogeneous LCCDE and with $x(n) = \delta(n)$ we have: $y(n) = h(n)$ (IR), as a solution try λ^n and obtain the characteristic equation (CE):

$$\lambda^n - 3\lambda^{n-1} + 2\lambda^{n-2} = 0 \quad \text{or} \quad \lambda^2 - 3\lambda + 2 = 0$$

Roots are $\lambda_1 = 1$ and $\lambda_2 = 2$. At this point we can point out that MATLAB’s ROOTS command can be used, as an example, to find the roots of more complex CEs. For this simple case the commands are:

```
>>A=[1 -3 2];
>>ROOTS(A)
ANS =
     2
     1
```

With the roots of the characteristic equation we can compute the impulse response as:

$$h(n) = C_1 + C_2(2)^n \quad ; \quad n > 2.$$

Using previously derived initial conditions one gets the following set of equations for the unknown coefficients C_1 and C_2 :

$$C_1 + 8C_2 = 26$$

$$C_1 + 16C_2 = 50$$

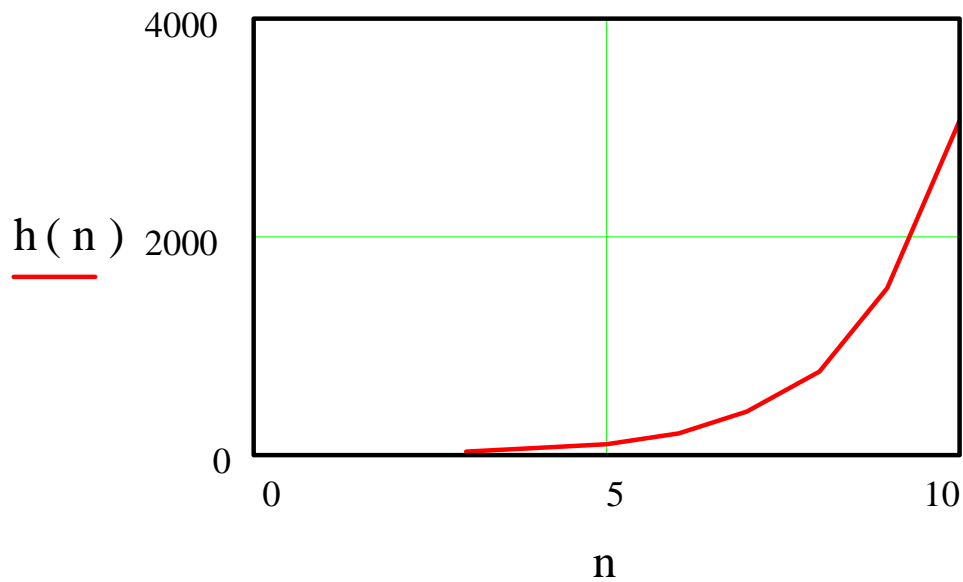
With results $C_1 = 2$ and $C_2 = 3$. Again note that for large number of unknowns we may put the systems of equations into a matrix form $AX = B$ where X contains the unknown coefficients. For our example the MATLAB commands are:

```
>>A=[1 8;1 16];
>>B=[26;50];
>>X=inv(A)*B
X =
     2
     3
```

Finally the complete solution for the impulse response is found as:

$$h(n) = 2 + 3(2)^n \quad ; \quad n > 2 \quad \text{and} \quad h(0) = 6, h(1) = 8, h(2) = 14.$$

We now use MATHCAD to present a plot of $h(n)$ vs. n . Note that it is easy to explain to students that the system is unstable and they can verify this by looking at MATHCAD plots as integer n is increased and $h(n)$ grows exponentially.



EXAMPLE II.2

One of the important applications of signal processing is in the area of digital filtering. With a rich history behind them, filters or frequency selective devices are crucial components in communication systems or in biomedical instrumentation, to just name a few areas of application of filters. In a given frequency range of operation there can be many advantages of using a digital filter as compared to analog filter. It is important to keep in mind that a large amount of filter literature is in the domain of classical filters such as Butterworth, Chebyshev or elliptic filter design approximation theory. As an example, one can design a filter based on Chebyshev I approximation and then use the bilinear transformation to design the corresponding digital filter. MATLAB's signal processing toolbox can be utilized to facilitate the design of analog filter (AF). As a numerical example, consider the following:

Let's design a Chebyshev type I lowpass filter with order 5 and a ripple of 1 dB in the passband. The MATLAB commands are :

```
>>[z,p,k]=cheblap(5,1)
z =
    []
p=
    -0.0895 + 0.9901i
    -0.2342 + 0.6119i
    -0.2895 + 0.0000i
    -0.2342 - 0.6119i
    -0.0895 - 0.9901i
```

Such that the s-domain transfer function of the filter can be obtained as:

$$H(s) = N(s)/D(s)$$

Where $N(s) = k = 0.1228$ (as obtained from MATLAB) and

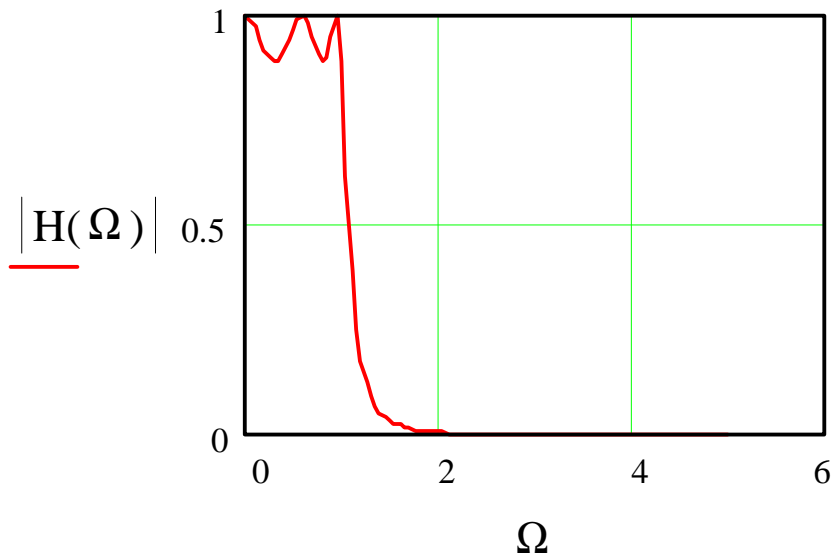
$$D(s) = (s+0.2895)(s^2 + 0.179s+0.9883)(s^2 +0.4648 + 0.42927j).$$

Assuming a sampling period (interval) of 1 second, one can use MATHCAD with the above $H(s)$ and substitute $s = 2(z-1)/(z+1)$ to get the equivalent filter in the z-domain. The frequency response of the digital filter can then be illustrated by using $z = \exp(j\omega)$ with $j=i$ being the imaginary unit and ω representing digital (radian) frequency. We now present results as obtained from MATHCAD:

$$\Omega := 0, .05.. 5$$

$$j := \sqrt{-1}$$

$$H(\Omega) := \frac{.1228}{((j \cdot \Omega + .2895) \cdot (-\Omega^2 + .179j \cdot \Omega + .9883) \cdot (-\Omega^2 + .4684j \cdot \Omega + .42927))}$$

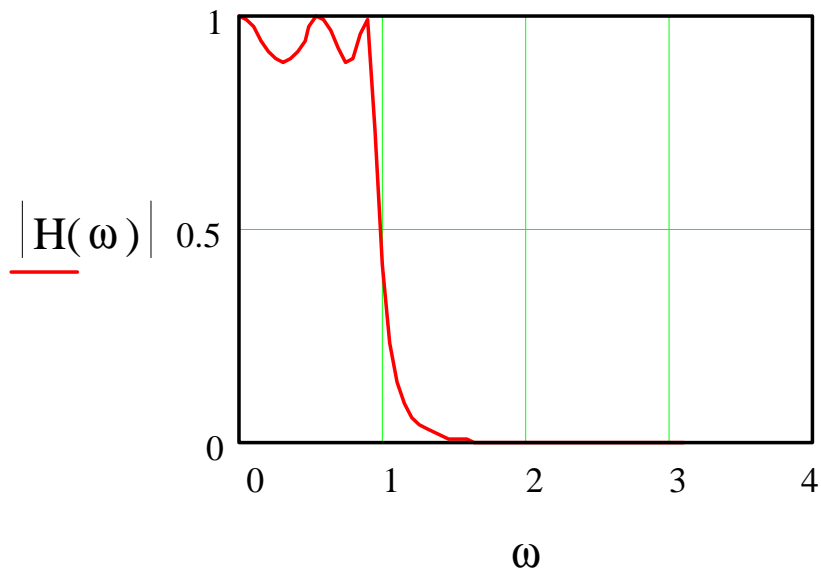


$$\omega := 0, 0.05.. \pi$$

$$z(\omega) := \exp(j \cdot \omega)$$

$$s(\omega) := 2 \cdot \frac{z(\omega) - 1}{(z(\omega) + 1)}$$

$$H(\omega) := \frac{.1228}{((s(\omega) + .2895) \cdot (s(\omega)^2 + .179s(\omega) + .9883) \cdot (s(\omega)^2 + .4684s(\omega) + .42927))}$$



Note that as mentioned earlier, ω represents digital frequencies while Ω denotes analog frequency. Note that natural mathematical expressions can be directly entered into MATHCAD worksheet and then we can do processing, etc. This is considered to be a positive feature for the MATHCAD software package while the availability of toolboxes make MATLAB more useful in other signal processing applications. We have used both of these tools in our classes (in addition to other packages) and found that while MATHCAD is also popular with students, the lack of availability of the “student edition” of this tool for WINDOWS applications may be a drawback. The Computer-Aided Engineering (CAE) center of our college of engineering has many packages available for student use including the latest professional versions of MATLAB and MATHCAD. While we presented an example in this section of IIR filters, computer methods are also extremely useful for design of FIR (finite impulse response) digital filters. We intend to present more viewgraphs during the Conference.

III. ODE Examples and Algorithms

Ordinary differential equations (ODEs) offer models useful for the treatment of linear systems and signal processing problems. Here we present algorithms for solving ODEs of the n th order. A time series analytical solution of n th order linear differential equations with step input follows.

We are interested in applying machine-generated Laplace transforms to a set of non-homogeneous second through fifth order linear differential equations in order to gain insight on the solution of the n th order case. The target differential equations have constant (but unknown) coefficients and are excited by a step function. In the analysis presented here (performed using the Mathematica mathematical programming language), zero initial conditions shall be assumed.

The target differential equations are expressed in terms of unknown constants **a-f** and excitation function **f[t]**, as shown in input statement 10:

```
In[10]:=
deq2 = a y''[t] + b y'[t] + c y[t] == f[t];
deq3 = a y'''[t] + b y''[t] + c y'[t] + d y[t] == f[t];
deq4 = a y''''[t] + b y'''[t] + c y''[t] + d y'[t] + e
y[t]==f[t];
deq5 = a y'''''[t] + b y''''[t] + c y'''[t] + d y''[t] + e
y'[t]+ f
y[t]==f[t];
```

Our analysis begins by loading Mathematica's Laplace Transform package:

```
In[11]:=
Needs["Calculus`LaplaceTransform`"]
```

```
In[12]:=
ILT = InverseLaplaceTransform;
```

The characteristic equation for a second order system may be derived as:

```
In[13]:=
ce2 = deq2[[1]] /.
  {(a_.)*Derivative[i_][y_][t_] -> a*s^i,
   (a_.)*(y_)[t_] -> a}
```

```
Out[13]=
c + b s + a s2
```

The solution to a second order differential equation excited by a step input follows as:

```
In[14]:=
OfTheForm[ILT[1/(s*ce2), s, t], t]
```

```
Out[14]=
1
- + c[1] Exp[t r[1]] + c[2] Exp[t r[2]]
c
```

The last operation employs a special encapsulation function [4] **OfTheForm** to encapsulate root objects **r[i]** and expansion coefficient objects **c[i]** in this solution. The tabulated roots and expansion coefficients are:

```
In[15]:=
{r[1], r[2]}//TableForm
```

```
Out[15]//TableForm=
      2
-b   Sqrt[b  - 4 a c]
-----
2 a      2 a

      2
-b   Sqrt[b  - 4 a c]
---- + -----
2 a      2 a
```

```
In[16]:=
Together[{c[1], c[2]}]//TableForm
```

```
Out[16]//TableForm=
      2
b - Sqrt[b  - 4 a c]
-----
      2
2 c Sqrt[b  - 4 a c]

      2
-b - Sqrt[b  - 4 a c]
-----
      2
2 c Sqrt[b  - 4 a c]
```


Repeating this procedure for the third, fourth and fifth order cases yields the solution forms $\mathbf{y}[\mathbf{t}]$ and expansion coefficients $\mathbf{c}[\mathbf{i}]$ shown below:

Second Order:

$$\frac{1}{c} + c[1] \text{Exp}[t \ r[1]] + c[2] \text{Exp}[t \ r[2]]$$

$$\frac{b - \text{Sqrt}[b^2 - 4 a c]}{2 c \text{Sqrt}[b^2 - 4 a c]}$$

$$\frac{-b - \text{Sqrt}[b^2 - 4 a c]}{2 c \text{Sqrt}[b^2 - 4 a c]}$$

Third Order:

$$\frac{1}{d} + c[1] \text{Exp}[t \ r[1]] + c[2] \text{Exp}[t \ r[2]] + c[3] \text{Exp}[t \ r[3]]$$

$$\frac{-c - b r[1] - a r[1]^2}{a d (r[1] - r[2]) (r[1] - r[3])}$$

$$\frac{c + b r[2] + a r[2]^2}{a d (r[1] - r[2]) (r[2] - r[3])}$$

$$\frac{c + b r[3] + a r[3]^2}{a d (r[1] - r[3]) (-r[2] + r[3])}$$

Fourth Order:

$$\frac{1}{e} + c[1] \text{Exp}[t \ r[1]] + c[2] \text{Exp}[t \ r[2]] + c[3] \text{Exp}[t \ r[3]] + c[4] \text{Exp}[t \ r[4]]$$

$$\frac{-d - c r[1] - b r[1]^2 - a r[1]^3}{a e (r[1] - r[2]) (r[1] - r[3]) (r[1] - r[4])}$$

$$\frac{d + c r[2] + b r[2]^2 + a r[2]^3}{a e (r[1] - r[2]) (r[2] - r[3]) (r[2] - r[4])}$$

$$\frac{d + c r[3] + b r[3]^2 + a r[3]^3}{a e (r[1] - r[3]) (-r[2] + r[3]) (r[3] - r[4])}$$

$$\frac{d + c r[4] + b r[4]^2 + a r[4]^3}{a e (r[1] - r[4]) (-r[2] + r[4]) (-r[3] + r[4])}$$

Fifth Order:

$$\frac{1}{f} + c[1] \text{Exp}[t r[1]] + c[2] \text{Exp}[t r[2]] + c[3] \text{Exp}[t r[3]] + c[4] \text{Exp}[t r[4]] + c[5] \text{Exp}[t r[5]]$$

$$\frac{(-e - d r[1] - c r[1]^2 - b r[1]^3 - a r[1]^4) (r[1] - r[2]) (r[1] - r[3]) (r[1] - r[4]) (r[1] - r[5])}{(e + d r[2] + c r[2]^2 + b r[2]^3 + a r[2]^4) (r[2] - r[3]) (r[2] - r[4]) (r[2] - r[5])}$$

$$\frac{(e + d r[3] + c r[3]^2 + b r[3]^3 + a r[3]^4) (-r[2] + r[3]) (r[3] - r[4]) (r[3] - r[5])}{(e + d r[4] + c r[4]^2 + b r[4]^3 + a r[4]^4) (-r[2] + r[4]) (-r[3] + r[4]) (r[4] - r[5])}$$

$$\frac{(e + d r[5] + c r[5]^2 + b r[5]^3 + a r[5]^4) (-r[2] + r[5]) (-r[3] + r[5]) (-r[4] + r[5])}{(e + d r[3] + c r[3]^2 + b r[3]^3 + a r[3]^4) (-r[2] + r[3]) (r[3] - r[4]) (r[3] - r[5])}$$

Close study of these results reveals a general time series solution to the following nth order differential equation

$$a[n] y^{(n)}[t] + a[n-1] y^{(n-1)}[t] + \dots + a[2] y''[t] + a[1] y'[t] + a[0] y[t] = m f[t]$$

to be:

```
In[35]:=
yf[t_, n_] := Sum[c[i]*E^(r[i]*t), {i, 1, n}] + yf[t]
```

where $yf[t]$ is the forced response and the expansion coefficients $c[i, \dots]$ may be expressed by the following algorithm:

```
In[36]:=
c[i_, coeff_List, root_List, (m_.)*(f_)[(\[Lambda]_.)*t]] :=
Module[{a, r, j, n},
  n = Length[root];
  Do[a[j] = coeff[[n - j + 1]], {j, 0, n}];
  Do[r[j] = root[[j]], {j, 1, n}];
  g[i, n, a, r, m*f[t]]/
  (Product[r[i] - r[j], {j, 1, i - 1}]*Product[r[i]-r[j],
  {j, i + 1, n}])]
```

Note that the algorithm for $c[i]$ associates the coefficient and root parameters in the Mathematica language as follows.

```
coeff = Table[a[i], {i, n, 0}]
root = Table[r[i], {i, 1, n}]
```

The time series algorithm also makes use of a special generating function $g[i, \dots]$ which characterizes the solution for the step input case:

```
In[37]:=
g[i_, n_, a_, r_, (m_.)*UnitStep[t]] :=
(-m*Sum[a[k]*r[i]^(k - 1), {k, 1, n}])/(a[n]*a[0])
```

The particular solution for the step input can also be shown to be:

```
In[38]:=
yf[t] = m/a[0];
```

Example applications of the method (in Mathematica and MathCAD) as well as solutions of ODEs excited by impulse and exponential forcing functions are detailed at <http://www2.wmich.edu/~johnson>.

IV. Conclusion and Further Work

While we concentrated on use of a few software packages such as MATHCAD, MATLAB and MATHEMATICA, it is apparent that others can also be used and that every tool has its limitations. It is important to regard these as educational tools and present them to students along with careful derivation of necessary algorithms. For example, one good use of MATHCAD in the study of Fourier series will be to let students derive symbolic expressions for its coefficients, a_0 , a_n , b_n given a periodic signal and then enter the following expression into MATHCAD:

$$f(t) = a_0 + \sum_{n=1}^{\infty} [a_n \cos(n\Omega_0 t) + b_n \sin(n\Omega_0 t)],$$

where Ω_0 is the fundamental frequency of the finite-power signal. Students can then experiment with obtaining different waveforms in using finite numbers of terms in the Σ instead of the theoretical upper limit of ∞ . We want to emphasize that as these tools become more powerful, it is important for instructors of these subjects to try to “mix in” theory and practice and let students use computer visualization (CV) to understand often abstract concepts such as convolution, frequency response, various transform methods and time/frequency domain analysis techniques. See references [2] and [3] for more MATLAB examples in the study of linear systems, and a discussion of DSP tools, respectively.

ACKNOWLEDGEMENTS

The authors would like to thank Dr. Leonard Lamberson, WMU Dean of Engineering, and Mr. Sridhar Erra, director of CAE center for their help and encouragement. One of the authors (SHM) thanks Mr. Edward Adams of MathSoft and Prentice Hall for use of MATHCAD+6.0 and classroom edition of MATLAB, respectively.

REFERENCES

- [1]. “Digital Signal Processing, Principles, Algorithms, and Applications”, 3rd edition, by John Proakis and Dimitris Manolakis, Prentice Hall, 1996.
- [2]. “Contemporary Linear Systems Using MATLAB”, by Robert D. Strum and Donald E. Kirk, PWS Publishing, 1994.
- [3]. “Computerized Tools in Digital Signal Processing”, S. Hossein Mousavinezhad, 1996 ASEE Annual Conference, June 24, Washington, D.C.
- [4]. Johnson, D. & Buege, J. Rethinking the Way We Teach Undergraduate Physics and Engineering with Mathematica, Proceedings of the First International Mathematica Symposium (IMS'95), July 16-20, 1995.

Biographical Information

S. HOSSEIN MOUSAVINEZHAD

Dr. Mousavinezhad is Professor and Chair, Department of Electrical and Computer Engineering, Western Michigan University, Kalamazoo, Michigan. His expertise focuses on the applications of digital signal processing in communication systems and bioelectromagnetics. He currently is conducting research into the development of signal processing algorithms and in the area of global aspects of engineering education.

DEAN R. JOHNSON

Dr. Johnson is Associate Professor of Electrical and Computer Engineering, Western Michigan University. His expertise focuses on the computer systems and fiber optics communication systems. He is the author of a book on fiber optics and was a faculty fellow, Jet Propulsion Laboratory, Pasadena, California, 1982, 1991, and 1992.