Homework #1

Due: Thursday February 9, 2006 in class.

This homework covers the following materials: digital signal processing computing algorithms: digital filters, linear transformations, communication algorithms. You must do the homework by yourself. No collaborations are allowed. Late homework will receive a 5% penalty per day. There are total 100 points. This homework is worth 10% of your overall grades. Homework solution will be posted after the class when the homework is due. Hence, NO LATE HOMEWORK SUBMISSION WILL BE ACCEPTED.

You must either type the solution or write it neatly. Each homework submitted must be stapled and with a cover page. On the cover page, you must specify the homework #, your last and first name. Failure to follow instruction will result in 5 point penalty of homework grade.

CC problems: A problem with CC marked behind the point assignment is a completion credit only problem. As long as you show reasonable effort, you will receive full credit. Many of these questions are open-ended and hence may not have absolutely correct answers exist. The grader will be instructed to grade each part independently.

1. (10 points, Convolution) A Töplitz matrix like the one shown below is defined by a sequence \( \{t_k, -3 \leq k \leq 3\} \) that form the first column and the first row of this matrix. It has been said that the product of a Töplitz matrix and a vector can be realized using a convolution operation.

\[
\begin{bmatrix}
    t_0 & t_1 & t_2 & t_3 \\
    t_{-1} & t_0 & t_1 & t_2 \\
    t_{-2} & t_{-1} & t_0 & t_1 \\
    t_{-3} & t_{-2} & t_{-1} & t_0
\end{bmatrix}
\begin{bmatrix}
    x_1 \\
    x_2 \\
    x_3 \\
    x_4
\end{bmatrix}
=
\begin{bmatrix}
    y_1 \\
    y_2 \\
    y_3 \\
    y_4
\end{bmatrix}
\]

(a) (5 points) (i) Show that \( \{y_i\} \) can be obtained by convolving two sequences. (ii) Give explicitly these two sequences in terms of \( \{t_i\} \) and \( \{x_i\} \). (iii) Denote the resulting sequence of convolution as \( \{c_i\} \). What is the length of \( \{c_i\}\)? (iv) Represent \( y_k; 1 \leq k \leq 4 \) as a function of \( c_i \).

(b) (5 points) Assuming general \( N \times N \) Töplitz matrix-vector product. What are the total number of multiplications and additions that need to be performed to compute \( \{y_i\} \) using (i) direct matrix-vector multiplications, (ii) convolution?

2. (10 points) Graph transpose theorem

An important graph-theoretic theorem known as the graph transpose theorem states the following: The transfer function of a single-input, single output signal flow graph remains unchanged if (i) the direction of all arcs are reversed, and (ii) the position of inputs and outputs are inter-changed.

Recall the direct form-I, 2\textsuperscript{nd} order IIR digital filter shown in the digitalfilter.ppt note.

(a) (5 points) Apply the graph transpose theorem to the SFG of the direct form I, and draw the resulting SFG.
(b) (5 points) Derive the transfer function of the SFG you obtained in part (a). Show that it is identical to the transfer function of the original SFG.

3. (15 points) DFT
Refer to the DFT algorithm in the intro.ppt notes. A pseudo code is given below:

```c
/* cos(\theta(n,k))=\cos(-2\pi*n*k/N), \sin(\theta(n,k))=\sin(-2\pi*n*k/N),
/* 0 \leq n, k \leq N-1, are assumed to have been computed in advance,
/* and can be loaded into the memory. So the computation cost is not
/* included in this problem
For k=0:N-1,
    For n=0:N-1,
        C(n,k)=\cos(-2*\pi*n*k/N);
        S(n,k)=\sin(-2*\pi*n*k/N);
    End
End
/* Here is DFT computation
/* Note that X(k) is a complex number where i = square root of -1
/* mcount=# of real multiplications, acount = # of real additions
mcount=0, acount=0,
For k=0:N-1,
    X(k)=0,
    For n=0:N-1,
        X(k)=X(k)+C(n,k)*x(n)+i*S(n,k)*x(n),
        mcount=mcount+2, acount=acount+2,
    End
End
```

(a) (3 points, CC) Turn in the source code of a sequential program in a high level programming language of your choice to implement the DFT algorithm.
(b) (2 points) Give an eight-point DFT example to illustrate the correctness of the computation results. For convenience, report the results by filling the table below:

<table>
<thead>
<tr>
<th>n, k</th>
<th>x(n)</th>
<th>X(k)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.8381</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.0196</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0.6813</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.3795</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.8318</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0.5028</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0.7095</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>0.4289</td>
<td></td>
</tr>
</tbody>
</table>

(c) (2 points) Give the mcount and acount result of this 8-point example, as well as a function of N.
(d) (3 points) Observe a symmetry property exhibited in the result from part (ii) that if \(x(n)\) is a real-valued sequence, then some \(X(k)\)’s are complex conjugate each other. State this property formally (together with the range of \(k\)) and prove it for the general case of \(N\).

(e) (3 points, CC) Modify the program you implemented in part (i) so that you can compute the same results. Turn in the source code of the program.

(f) (2 points) Repeat part iii) for program in part iv).

4. (10 points, CC) Read the following paper from the course reference list:

(a) (5 points) Program this 8-point, 1D DCT algorithm in a high-level programming language. Submit the source code. Test run this program with the same 8 data points as shown in problem 3(b). Submit the output.

(b) (5 points) Discuss the implementation issues of this algorithm: How many multiplications, additions, temporal storages (# registers) are needed; What should be the word length? The width of the data path, numerical properties, etc.

5. (25 points)

(a) (5 points) Refer to the direct form-II realization in the class notes digitalfilter.ppt. The time domain equations that describes the 2nd order IIR filter are:
\[
\begin{align*}
    w(n) &= -a(1)w(n-1) - a(2)w(n-2) + x(n) \\
    y(n) &= b(0)w(n) + b(1)w(n-1) + b(2)w(n-2)
\end{align*}
\]

Denote two state variables \(v_1(n) = w(n-1), v_2(n) = w(n-2)\). Derive the state space representation of this digital filter
\[
\begin{align*}
    \mathbf{v}(n+1) &= \mathbf{A} \cdot \mathbf{v}(n) + \mathbf{b} \cdot x(n) \\
    y(n) &= \mathbf{c}^T \mathbf{v}(n) + d \cdot x(n)
\end{align*}
\]

by specifying the \(\mathbf{A}, \mathbf{b}, \mathbf{c}\) and \(d\) matrices (or vectors). You must give derivations. Answer alone will not receive credit.

(b) (3 points) Suppose that the two complex conjugate roots of the characteristic polynomial
\[
A(z) = z^2 + a(1)z + a(2) = 0
\]

are \(r \cos \theta \pm j r \sin \theta\). Represent \(a(1), a(2)\) in terms of \(r\) and \(\theta\).

(c) (7 points) Consider the orthogonal state space realization \((r > 0)\)
\[
\begin{align*}
    \begin{bmatrix} u_1(n+1) \\ u_2(n+1) \end{bmatrix} &= r \cdot \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} u_1(n) \\ u_2(n) \end{bmatrix} + \begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix} x(n) = \mathbf{R} \cdot \mathbf{u}(n) + \mathbf{b} x(n) \\
    y(n) &= \begin{bmatrix} \gamma_1 & \gamma_2 \end{bmatrix} \begin{bmatrix} u_1(n) \\ u_2(n) \end{bmatrix} + \delta \cdot x(n) = \mathbf{Y}^T \mathbf{u}(n) + \mathbf{c} \cdot x(n)
\end{align*}
\]

Find its transfer function \(H(z) = Y(z)/X(z)\). Express the result as a function of \(r, \theta, \beta_1, \beta_2, \gamma_1, \gamma_2, \text{ and } \delta\), and find \(b(0), b(1), \text{ and } b(2)\) as a function of these parameters too.

(d) (5 points) Suppose that the IIR filter can be described by a difference equation
\[
y(n) = 0.96 \cdot y(n-1) - 0.64 \cdot y(n-2) + x(n)
\]
Represent this IIR filter in a orthogonal state space representation (**). Assume that \( \beta_1 = 1 \) and \( \beta_2 = 0 \).

(c) (5 points) Use induction method, prove that for the \( \mathbf{R} \) matrix defined in eq. (**),
\[
\mathbf{R}^k = r^k \begin{bmatrix} \cos(k\theta) & -\sin(k\theta) \\ \sin(k\theta) & \cos(k\theta) \end{bmatrix}
\]

6. (20 points) Quantization Error and Dynamic range

Consider a first order IIR filter
\[
y(n) = a \cdot y(n-1) + x(n) \quad 0 < |a| < 1, \quad |x(n)| < B, \quad y(0) = 0, \quad n = 1, 2, \ldots
\]

(a) (8 points)

i) (4 points) Show that
\[
|y(n)| \leq B \cdot \frac{1-|a|^n}{1-|a|}
\]

ii) (4 points) Suppose that the input data sequence \( \{x(n)\} \) is sampled and quantized to 8 bits integers per sample in 2’s complement representation, and that the parameter \( a \) is represented as a 8-bit fixed point fractional number in 2’s complement representation. How many bits are needed to represent each \( y(n) \) as 2’s complement binary integers? Use the worst case scenario (largest possible value of \( B \), and largest possible value of \( a \), and assume \( n \to \infty \)) to derive your answer.

(b) (12 points) Due to the multiplication with a fractional number \( a \), the computation will subject to quantization error (due to rounding, saturation, or truncation). During each iteration, \( y(n) \) will be a binary number with both integral part as well as \( l-1 \) (\( l > 1 \)) fractional binary bits. To analyze the effect of quantization, denote \( Q[y] \) to be a quantized quantity of a précised value of variable \( y \) such that
\[
\hat{y} = Q[y] = y + \varepsilon
\]

where \( \varepsilon \in [-2^{-l} + 2^{-l}] \) is a uniformly distributed random variable. The quantization errors \( \varepsilon \) for different values of \( y \) are assumed to be i.i.d.

i) (2 points) Show that \( E\{\varepsilon\} = 0 \), \( \text{Var.}\{\varepsilon\} = 2^{-2l}/3 \).

ii) (2 points) For the \( n^{th} \) iteration of the IIR filter, computed value of \( y(n) \) subject to two types of quantization error: the quantization error inherited from the previous iterations that accompanying \( y(n-1) \), as well as additional quantization error imposed after the multiplication and addition. Thus,
\[
\hat{y}(n) = Q[a \cdot \hat{y}(n-1) + x(n)] = a \cdot \hat{y}(n-1) + x(n) + \varepsilon(n) = y(n) + \delta(n)
\]

where \( \delta(n) \) is the accumulated quantization error on \( y(n) \), \( \varepsilon(n) \) is the uniform quantization error described in part (i), and \( \hat{y}(n) \) is the computed value of \( y(n) \), taking into account all the accumulated quantization errors.

Show that \( \delta(n) = a \cdot \delta(n-1) + \varepsilon(n) \).

iii) (2 points) Moreover, use an analysis similar to part (a), deduce a worst case upper bound for \( \delta(n) \):
iv) (3 points) Assume that $2^{-m-1} < 1 - |a| \leq 2^{-m}$. Solve for a lower bound of $l$ such that the maximum accumulated quantization error $\max |\delta(n)| \leq 2^{-1}$:

v) (3 points) Suppose that we want the integral portion of each $y(n)$ to be free of quantization error, as well as free of overflow situations, given that both $\{x(n)\}$ and $a$ are represented in 8 bits (cf. part (a) ii)). Furthermore, assume that $a = 0.875$. What is the word length of the accumulator that should be used to represent $y(n)$? What many bits in the integral portion and how many bits in the fractional portion? Repeat this question consider the worst case scenario of $a$.

7. (10 points) In discrete wavelet transformation, a common operation is down-sampling the output obtained from a digital filter by 2:

$$x[n] \xrightarrow{H(z)} w[n] \xrightarrow{\downarrow 2} y[n]$$

where $w[n] = x[n] ** h[n]$ is the convolution of the input with the FIR filter. $y[n] = w[2n]$ or $y[n] = w[2n+1]$ are $w[n]$ down-sampled by a factor of 2. Note that throwing away half of the output is quite a waste in terms of computation resources! For convenience, let us denote $x_{\text{even}}[n] = \{x[0], x[2], \ldots\}$ and $h_{\text{even}}[n] = \{h[0], h[2], \ldots\}$ to be the even indexed terms of the sequences $\{x[n]\}$ and $\{h[n]\}$ respective. Similarly, let us denote $x_{\text{odd}}[n] = \{x[1], x[3], \ldots\}$ and $h_{\text{odd}}[n] = \{h[1], h[3], \ldots\}$ to be the odd indexed terms of these sequences.

(a) (5 points) Show that

$$w_{\text{even}}[n] = x_{\text{even}}[n] ** h_{\text{even}}[n] + x_{\text{odd}}[n] ** h_{\text{odd}}[n]$$

$$w_{\text{odd}}[n] = x_{\text{even}}[n] ** h_{\text{odd}}[n] + x_{\text{odd}}[n] ** h_{\text{even}}[n]$$

As such, one my split $x[n]$ into even and odd parts as well as $h[n]$ into even and odd parts and convolve these half-length sequence individually and add the results.

(b) (5 points) In general, this method can be applied to other down-sampling factors, and leads to a poly-phase formulation: Denote $x_k[n] = x[Mn+k]$, $h_k[n] = h[Mn+k]$, $w_k[n] = w[Mn+k]$, $0 \leq k \leq M-1$, show that

$$w_k[n] = \sum_{m=0}^{M-1} h_m[n] ** x_{m+k}[n]$$