Please set a version number using \version

EGT??? (use \yearcode to set this) CRIB

Wednesday 2nd May 2014 14.00 to 15.30

Module 3F3

SIGNAL AND PATTERN PROCESSING

Worked Solutions

STATIONERY REQUIREMENTS

SPECIAL REQUIREMENTS TO BE SUPPLIED FOR THIS EXAM

You may not start to read the questions printed on the subsequent pages of this question paper until instructed to do so.

Please set a version number using \version

1 (a) Show that the multiplication of two complex numbers takes 4 real multiplications and 2 real additions. Hence give the *exact* operation count in real multiplies and additions for the length-N DFT. Assume that all complex exponentials have been pre-computed and stored ready for use. [20%]

Solution:

Take two complex numbers and multiply:

$(a+jb)(c+jd) = (ac-bd)+j(bc+ad)$

Hence 4 real multiplies and 2 real adds (not three since the middle add refers just to storage of the real and imaginary parts of the solution).

(b) Discuss the advantages of the FFT over the DFT for computation of the frequency components of a data vector. Give three examples of engineering or scientific areas where the FFT/ DFT are used in practice. [20%]

Solution:

FFT runs much faster for long data sequences: $5N\log_2(N)$ as compared to around $8N^2$ for full DFT.

FFT is computed in-place (requires no extra memory store.

FFT/DFT are applied in speech or image coding, in astonomy for detecting planets/ exoplanets, in the analysis of building vibration modes, and numerous others...!

(c) Explain how the DFT $\mathbf{X} = [X_0\, X_1 \ldots X_{N-1}]^T$ of a length-N signal vector $\mathbf{x} = [x_0\, x_1 \ldots x_{N-1}]^T$ may be written in matrix-vector form as:

$$\mathbf{X} = \mathbf{W}\mathbf{x},$$

giving the form of the matrix $\mathbf{W}$ in terms of the column vectors

$$\mathbf{w}_p = [1\ \exp(-2jp\pi/N)\ \exp(-4jp\pi/N)\ \ldots\ \exp(-2(N-1)jp\pi/N)]^T,\ p=0,1,\ldots,N-1.$$

[20%]

Solution:

Please set a version number using \version

If we define $w = \exp(-j2\pi/N)$ then we can write:

$$
\begin{aligned}
G_p &= \sum_{n=0}^{N-1} g_n w^{np} \\
&= \begin{bmatrix} 1 & w^p & w^{2p} & \dots & w^{(N-1)p} \end{bmatrix} \begin{bmatrix} g_0 \\ g_1 \\ \vdots \\ g_{N-1} \end{bmatrix} \\
&= (\mathbf{w}_p)^T \mathbf{g}
\end{aligned}
$$

where

$$
\mathbf{w}_p = \begin{bmatrix} 1 \\ w^p \\ w^{2p} \\ \dots \\ w^{(N-1)p} \end{bmatrix}
$$

[Note that $.^H$ denotes the *Hermitian* transpose, i.e. the complex conjugate of the transpose].

(d) Show that

$$
\mathbf{w}_q^H \mathbf{w}_p = \begin{cases} N, & q = p, \\ 0, & \text{otherwise,} \end{cases}
$$

where p and q may take any values in the range $0, 1, \dots, N-1$.
Hence find the matrix-vector form of the inverse DFT:

$$
\mathbf{x} = \mathbf{V}\mathbf{X},
$$

where again matrix $\mathbf{V}$ should be expressed in terms of the vectors $\mathbf{w}_p$. [40%]
Solution:

(e) Hence derive, by direct application of matrix algebra, the inverse DFT in the form:

$$
\mathbf{x} = \mathbf{H}\mathbf{X}
$$

where the elements of $\mathbf{H}$ should be carefully defined. [30%]

Please set a version number using \version

[Hint: consider the products $\mathbf{m}_i^H \mathbf{m}_j$ in the case where $i = j$ and $i \neq j$. Note that $\mathbf{m}_i^H$ is the complex conjugate of the transpose of vector $\mathbf{m}_i$]

Solution:

Now list all the G_p s in a vector $\mathbf{G}$ to obtain:

$$\mathbf{G} = \begin{bmatrix} G_0 \\ G_1 \\ \vdots \\ G_{N-1} \end{bmatrix} = \begin{bmatrix} \mathbf{w}_0^T \\ \mathbf{w}_1^T \\ \vdots \\ \mathbf{w}_{N-1}^T \end{bmatrix} \mathbf{g}$$
$$= \mathbf{W}\mathbf{g}$$

Since the rows of $\mathbf{W}$ are orthogonal, i.e. $\mathbf{w}_p^H \mathbf{w}_q = 0$ for $p \neq q$, and $\mathbf{w}_p^H \mathbf{w}_p = N$ for all p, we have:

$$\mathbf{W}\mathbf{W}^H = N\mathbf{I}$$

i.e.

$$\mathbf{W}^{-1} = \frac{1}{N}\mathbf{W}^H$$

Hence we immediately have an alternative (and simpler) derivation of the inverse DFT:

$$\mathbf{g} = \mathbf{W}^{-1}\mathbf{G} = (1/N)\ \mathbf{W}^H\mathbf{G}$$

Please set a version number using \version

2 (a) The bilinear transform

$$s = \frac{1-z^{-1}}{1+z^{-1}}$$

can be used for the design of a digital filter from an analogue prototype filter.

Show that frequencies map from the analogue to digital domain according to the relationship $\omega = \tan(\Omega/2)$ where ω and Ω should be carefully defined, including their units. [20%]

Solution:

Let $\psi(z)$ be the bilinear transform. Then set $z = \exp(j\omega)$. Now:

$$\begin{aligned}\psi(\exp(j\Omega)) &= \frac{1-\exp(-j\Omega)}{1+\exp(-j\Omega)} \\ &= \frac{\exp(-j\Omega/2)(\exp(j\Omega/2)-\exp(-j\Omega/2))}{\exp(-j\Omega/2)(\exp(+j\Omega/2)+\exp(-j\Omega/2))} \\ &= \frac{j\sin(\Omega/2)}{\cos(\Omega/2)} \\ &= j\tan(\Omega/2)\end{aligned}$$

which is purely imaginary, with the required frequency warp.

ω is in $rad.s^{-1}$ and Ω is normalised digital frequency in $rad.sample^{-1}$

(b) Discuss how the bilinear transform can be used to design digital filters from analogue prototypes. Your discussion should include a definition of the type of filters that can be designed, stability of the filter, and any distortions that may be introduced to the frequency and impulse responses. [30%]

Solution:

The steps of the bilinear transform method are as follows:

1. Warp the digital critical (e.g. bandedge or "corner") frequencies ω_i , in other words compute the corresponding analogue critical frequencies $\omega_i = tan(\Omega_i/2)$.

2. Design an analogue filter which satisfies the resulting filter response specification.

3. Apply the bilinear transform to the s-domain transfer function of the analogue filter to generate the required z-domain transfer function.

Can design IIR filters with this method from analogue prototype. Distortions are introduced in the non-linear frequency warping - will distort phases even in sections with constant amplitude response. The impulse response is distorted in an unpredictable way.

Please set a version number using \version

To get one type of filter from another, use transformation functions in the analogue domain (also possible in the digital domain, but not covered in this course):

Lowpass to Lowpass: Set $s' = s/\omega_c$ to give lowpass with cutoff at ω_c

Lowpass to Highpass: $s' = \frac{s^2+\omega_l\omega_u}{s(\omega_u-\omega_l)}$ to give bandpass with lower cutoff at ω_l, upper cutoff at w_u

Lowpass to Bandpass: $s' = \frac{s^2+\omega_l\omega_u}{s(\omega_u-\omega_l)}$ to give bandpass with lower cutoff at ω_l, upper cutoff at w_u

Lowpass to Bandstop: $s' = \frac{s(\omega_u-\omega_l)}{s^2+\omega_l\omega_u}$ to give bandstop with lower cutoff at ω_l, upper cutoff at w_u

[One or two of the above examples would suffice]

(c) Explain why the substitution $s = \frac{s'^2+\omega_l\omega_u}{s'(\omega_u-\omega_l)}$ can be used to take a low-pass analogue prototype and obtain a bandpass filter with lower cutoff at ω_l, upper cutoff at ω_u. [20%]

Solution:

Answer: this is a low-pass to bandpass transformation with lower and upper cut-offs at ω_1 and ω_2, respectively. Taking $s = j\omega$ we get the equivalent frequency substitution:

$$\omega \to \frac{\omega^2 - \omega_1\omega_2}{\omega(\omega_2 - \omega_1)}$$

In particular, taking $\omega = 0$, we find that this corresponds to $\omega' = \infty$ in the original prototype filter, and so does $\omega = \infty$. Hence frequencies 0 and ∞ are heavily attentuated in the transformed filter. ω_1 corresponds to $\omega' = -1$ in the original filter and and ω_2 to $\omega' = +1$, which means that these two values define corner (typically 3dB) frequencies in the new filter. Finally, the frequency $\omega' = 0$ corresponds to $\omega = \sqrt{\omega_1\omega_2}$ (the 'geometric mean') in the bandpass design, a frequency that lies in the passband between ω_1 and ω_2, hence the bandpass region is between ω_1 and ω_2.

(d) A digital band-pass filter is required with sample rate of 16kHz. The lower cut-off frequency should be 2kHz and the upper cut-off frequency should be 6kHz. A low-pass analogue prototype filter can be used with transfer function

$$H(s) = \frac{1}{(1+s)}.$$

Use this analogue prototype and the bilinear transform to obtain the transfer function of the corresponding band-pass fiilter and comment on the form of the result. Sketch its frequency magnitude response, paying particular attention to the shape of the passband and any 3dB points. What is the filter's gain at 0kHz, 2kHz, 4kHz, 6kHz and 8kHz? [30%]

Please set a version number using \version

Answer:

First warp the cut-off frequencies from normalised digital 'Ω' to analogue 'ω':

$$\Omega_1 = 2000 \times 2\pi/16000 \rightarrow \omega_1 = \tan(\Omega_1/2) = 0.4142 rad.s^{-1}$$

$$\Omega_2 = 6000 \times 2\pi/16000 \rightarrow \omega_2 = \tan(\Omega_2/2) = 2.4142 rad.s^{-1}$$

Then, using these warped values, perform the low-pass to bandpass transformation from part a):

$$\begin{aligned} H(s) \rightarrow & \frac{1}{\frac{s^2+\omega_1\omega_2}{s(\omega_2-\omega_1)}+1} \\ &= \frac{1}{\frac{s^2+\omega_1\omega_2}{s(\omega_2-\omega_1)}+1} \\ &= \frac{s(\omega_2-\omega_1)}{s^2+\omega_1\omega_2+s(\omega_2-\omega_1)} \end{aligned}$$

Now, apply biliinear transform,

$$s = \psi(z) = \frac{1-z^{-1}}{1+z^{-1}};$$

$$\begin{aligned} H(z) &= H(s)\big|_{s=\frac{1-z^{-1}}{1+z^{-1}}} \\ &= \frac{(\omega_2-\omega_1)(1-z^{-2})}{1+\omega_2-\omega_1+\omega_2\omega_1+(2\omega_1\omega_2-2)z^{-1}+(1+\omega_1\omega_2-(\omega_2-\omega_1))z^{-2}} \\ &= (1-z^{-2})/2 \end{aligned}$$

i.e. in this special case we have ended up with an FIR filter! This was because of the symmetry of the design across frequency.

By design, 2kHz and 6kHz must be at -3dB (corner frequencies from the analogue filter prototype). 4kHz is unity gain, since frequency zero (gain 1 in the analogue prototype) maps to $\sqrt{(\omega_2\omega_1)} = 1 rad.s^{-1}$ under the low-pass to band-pass transformation. The corresponding digital frequency is then $2\tan^{-1}(1) = \pi/2$, which is a quarter of the sample frequency, i.e. 4kHz. 0 and 8kHz have zero gain since the two zeros sit on the unit circle at these frequencies.

Sketch would look like:

Please set a version number using \version

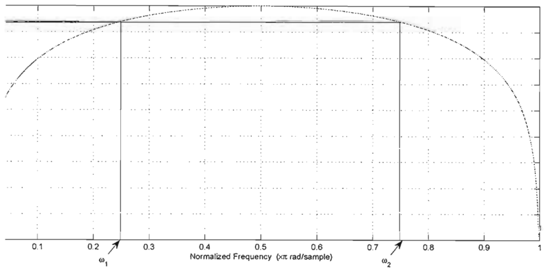

Fig. 1

3 (a) In some applications it is required to *estimate* a signal observed in noise, and in other cases it is required to *detect* the presence or otherwise of a signal. Explain the circumstances in which a *Wiener* filter could be applied, and those in which a *matched* could be the appropriate method, stating the technical assumptions underlying each. Give a practical application of each type of procedure. [30%]

Solution:

The Wiener filter applies to estimation of a stationary random signal in stationary noise. the cross- and auto-correlation functions of the processes must be specified. Example: noise reduction in mobile telephony (using short-term stationary assumptions).

The matched filter detects the presence of a known deterministic signal in random stationary noise. Example: detection of return from a sonar pulse.

(b) In a neuroscience experiment an auditory stimulus is presented to a student, and the neural response is measured through an electrode taped to the head. The measured electrical response signal may be expressed as a known, *deterministic* signal p_t observed in noise, as follows:

$$x_t = ap_t + e_t,$$

where a is an unknown random variable with zero mean and variance σ_a^2, and e_t is a zero mean white noise with variance σ_e^2. $\{e_t\}$ is assumed uncorrelated with a for all times t.

Determine the mean value of x_t, $\mu_t = E[x_t]$, and the autocorrelation function $r_{XX}[t_1,t_2] = E[x_{t_1}x_{t_2}]$. Explain with justification whether the process x_t is wide-sense stationary. [30%]

Solution:

$$\mu_t = E[a]p_t + E[e_t] = 0$$

Please set a version number using \version

$$r_{XX}[t_1,t_2] = E[(ap_{t_1}+e_{t_1})(ap_{t_2}+e_{t_2})$$
$$= E[a^2]p_{t_1}p_{t_2} + E[e_{t_1}e_{t_2}] + E[ae_{t_2}]p_{t_1} + E[ae_{t_1}]p_{t_2}] = \sigma_a^2 p_{t_1}p_{t_2} + \sigma_e^2\delta[t_2-t_1]+0$$

[last zero since a and e are zero-mean and uncorrelated].

This is non-stationary in general since r_{XX} depends on both t_1 and t_2 (and not on just their difference).

(c) It is required to estimate the unknown amplitude a when x_t is observed over a fixed time interval $t = 0, 1, ..., T-1$ by filtering the data as follows:

$$\hat{a} = \sum_{t=0}^{T-1} b_t x_t$$

where b_t are the filter coefficients.

The accuracy of estimation of a is to be assessed using the mean-squared error criterion:

$$J = E[(a-\hat{a})^2].$$

Show that J can be expressed as:

$$J = \sigma_a^2 - 2\sigma_a^2 \sum_{t=0}^{T-1} b_t + \sigma_e^2 \sum_{t=0}^{T-1} b_t^2 + \sigma_a^2 \sum_{t_1=0}^{T-1}\sum_{t_2=0}^{T-1} b_{t_1}b_{t_2}p_{t_1}p_{t_2}$$

Solution:

$$J = E[(a-\hat{a})^2] = E[(a-\mathbf{x}^T\mathbf{h})^2] = E[a^2] + \mathbf{h}^T E[\mathbf{xx}]^T\mathbf{h} - 2E[a\mathbf{x}^T]\mathbf{h}$$
$$= \sigma_a^2 + \mathbf{h}^T\mathbf{R_x}\mathbf{h} - 2E[a]^2\mathbf{p}^T\mathbf{h}$$

where the (i, j)th element of $\mathbf{R_x}$ is $r_{XX}[i, j]$ and in the last term we again used that a and e were zero mean and uncorrelated.

Expanding this out using r_{XX} from above, we have:

$$J = \sigma_a^2 + \mathbf{h}^T\mathbf{R_x}\mathbf{h}$$
$$= \sigma_a^2 + \sum_{t_1}\sum ....$$
$$= \sigma_a^2 - 2\sigma_a^2 \sum_{t=0}^{T-1} b_t + \sigma_e^2 \sum_{t=0}^{T-1} b_t^2 + \sigma_a^2 \sum_{t_1=0}^{T-1}\sum_{t_2=0}^{T-1} b_{t_1}b_{t_2}p_{t_1}p_{t_2}$$

[40%]

Please set a version number using \version

4 SOLUTION

(a) The likelihood is

$$p(\mathbf{y}|\mathbf{x}, a, \sigma^2) = \prod_n p(y_n|x_n, a, \sigma^2) = \prod_n \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\{-\frac{1}{2\sigma^2}(y_n - ax_n)^2\}$$

Equivalently we maximise

$$\log p(\mathbf{y}|\mathbf{x}, a, \sigma^2) = \sum_n -\frac{1}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}(y_n - ax_n)^2$$

Taking derivatives w.r.t. a and setting to zero gives

$$\sum_n \frac{1}{2\sigma^2}(y_n - ax_n)x_n = 0$$

Solving for a gives

$$a_{\mathrm{ML}} = \left(\sum_{n=1}^{N} y_n x_n\right) / \left(\sum_{n=1}^{N} x_n^2\right)$$

which is the scalar version of the "normal equations". Taking derivatives w.r.t. σ^2 and setting to zero we get:

$$\begin{aligned} -\frac{N}{2\sigma^2} + \frac{1}{2\sigma^4}\sum_n (y_n - ax_n)^2 &= 0 \\ N\sigma^2 &= \sum_n (y_n - ax_n)^2 \\ \sigma^2_{\mathrm{ML}} &= \frac{1}{N}\sum_{n=1}^{N}(y_n - ax_n)^2 \end{aligned}$$

which is the average of the residual errors.

(b) The maximum of the likelihood for this model will be higher than for the model in part (a) since this more complex model has more parameters to fit the data. The value of σ^2 will be lower than for part (a) since again the model can fit the data better and so the average residuals will be smaller (see solution for σ^2_{ML} in part (a)).

(c) Laplacian noise has heavier tails than Gaussian noise, so in general using such a model might be a good idea if we expect to have *outliers*—points where the actual y values are far from the linear prediction. Because this model can better handle outliers it is considered more *robust* than the model in part (a).

END OF PAPER