332:346 – Exam #2 Review Topics – Spring 2010

- Linear and circular buffer implementations of delays and FIR filters.
- Hardware aspects and circular buffer implementations of FIR filters.
- Infinite and finite geometric series.
- Computing z-transforms using the power series definition.
- Determining the ROC of a z-transform.
- Using the ROC information to invert a z-transform.
- Computing inverse z-transforms by partial fraction expansions.
- What to do when the partial fraction expansion is not valid: long division and remove-restore methods.
- Quickly inverting z-transforms of the form $H(z) = G(z)/(1-z^{-D})$, by periodically replicating and overlapping $g(n)$ with period D, that is, $h(n) = g(n) + g(n-D) + g(n-2D) + \cdots$.
- Computing output $y(n)$ for a given input $x(n)$ using z-transforms. Method: compute $X(z)$, form $Y(z) = H(z)X(z)$, and compute its inverse z-transform.
- Determining $h(n)$ from the knowledge of the input $x(n)$ and output $y(n)$. Method: Compute $Y(z)$ and $X(z)$, form the ratio $H(z) = Y(z)/X(z)$, and compute its inverse z-transform.
- Determining $H(z)$ by analyzing a block diagram. Method: assign labels to all signal lines that do not already have labels, and solve for the ratio $Y(z)/X(z)$.
- Equivalent descriptions of linear filters: Transfer function $H(z)$; I/O difference equations; Impulse response $h(n)$; Frequency response $H(\omega)$; Magnitude and phase responses; Block diagram realization; Sample-by-sample processing algorithm; Pole/zero pattern.
- Passing from one description to another with the help of $H(z)$.
- Sketching the magnitude response by inspecting the pole/zero pattern: draw peaks near poles, draw dips near zeros.
- Sinusoidal steady-state and transient response of a filter. Time constant of a filter, 40-dB and 60-dB time constants. Identifying the transient and steady-state parts of an output $y(n)$.
- Filtering a periodic signal $x(n)$ of period N through a stable filter and determining the periodic steady-state output $y_{\text{steady}}(n)$. Method: expand $x(n)$ into a sum of N DFT-frequency harmonics $\omega_k = 2\pi k/N$, $k = 0, 1, \ldots, N-1$, then determine the steady-state sinusoidal responses of the individual harmonics and sum them.
- Calculating the periodic steady-state output of a stable and causal filter when the input is a causal periodic signal.
- Simple pole/zero designs. First and second order filters. $\Delta\omega = 2(1-R)$.
- Design of peaking or notching 2nd order filters by putting zeros behind the poles or vice versa. Using $\Delta\omega = 2(1-R)$ in 2nd-order notch filter design.
- Tradeoff between time-constant and 3-dB notch width.
- Design peaking or notch filters that have a desired time constant. Method: solve $n_{\text{eff}} = \ln(\epsilon)/\ln(R)$ for R and proceed with usual pole/zero design.
- Roots of unity and their use in the pole/zero design of comb filters for removing periodic noise or enhancing periodic signals. Roots of $z^D = 1$ are $z_k = e^{2\pi jk/D}$, $k = 0, 1, \ldots, D-1$. The roots of $z^D = -1$ are $z_k = e^{\pi j(2k+1)/D}$, $k = 0, 1, \ldots, D-1$.
- Direct, canonical, transposed, and cascade realization forms. Their difference equation descriptions and sample processing algorithms.
- Analyzing arbitrary (i.e., not necessarily direct, canonical, transposed) block diagrams in the z-domain to obtain their transfer function.
- Converting an arbitrary block diagram into a sample processing algorithm. Methodology: (a) give names to the contents of delays, i.e., to the signals coming out of the delays. These are the states and are available at the current time instant. (b) Give names to additional signal lines, if necessary, to simplify the writing of the algorithm. (c) Make sure the computational steps are in the right order (e.g., you can't use the output before it is computed.)
- Circular buffer implementation of the canonical realization. Understanding the difference between buffer contents w_i and delay contents s_i (i.e. filter states). See Example 7.5.4.
- Digital audio effects. Delays, echoes, comb filters, reverberators (plain, allpass, lowpass), and multi-tap delay algorithms, and their circular buffer implementations.

Reading Materials:

All class material and all assigned material.
Textbook sections: section 4.2, chapters 5-7, sections 8.2.1–8.2.4.
programming of filtering algorithms. DSP Labs 2 and 3. All hardware labs.

Practice Problems:

Textbook problems (assigned or not assigned) from above sections.
Homework sets and Examples in text.
Old exam problems (solutions are not available).