Lecture 7

2-D IIR Filtering

Announcements

- PS #2 has been posted
  - Due: Friday, Sept. 12, 2003 (campus)
  - Due: Friday, Sept. 26, 2003 (video)
- New .pdf files of Lectures 1—8 have been posted.
  - Blurred equation problem solved.
  - Slides adjusted to agree with coverage.
  - Typos remain

2-D Difference Equations

- The concept of a difference equation can be extended to two (or more) dimensions.

$$\sum_{k_1}\sum_{k_2} b[k_1,k_2]y[n_1-k_1,n_2-k_2] = \sum_{r_1}\sum_{r_2} a[r_1,r_2]x[n_1-r_1,n_2-r_2]$$

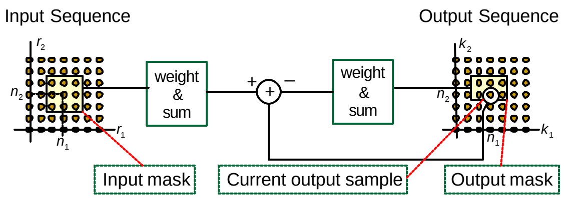
  - The sums have finite limits.
  - The size of *b* is related to the filter order.
  - The system is LTI if it is at initial rest.
  - The state space associated with a 2-D difference equation is of infinite dimensionality.

Recursive Computability

$$y[n_1,n_2] = \sum_{r_1,r_2} a[r_1,r_2]x[n_1-r_1,n_2-r_2] - \sum_{\substack{k_1,k_2 \\ [k_1,k_2]\neq[0,0]}} b[k_1,k_2]y[n_1-k_1,n_2-k_2]$$

- To use this as an algorithm, the samples of *y* on the right need to be computed before the one on the left.
- If such an ordering exists, the system is **recursively computable**.
  - If there is one such ordering, there is an infinite number of them.

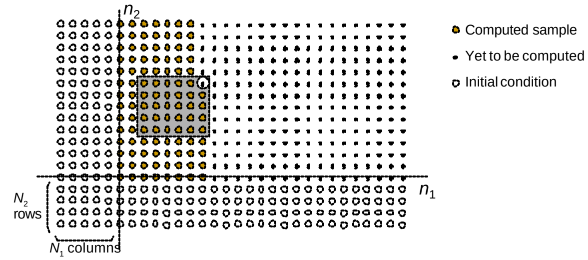
Input and Output Masks



- The **input and output masks** isolate the samples of the input and output sequences that are used to compute the current output sample.
- Their shape is the shape of the support of *a* and *b* (rotated through 180º.

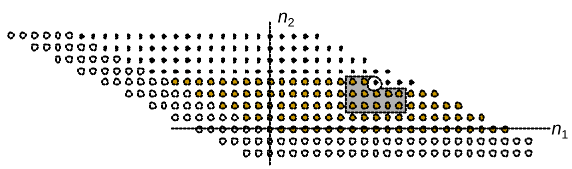
Quarter-Plane Masks

- Quarter-plane filters have masks with the "hole" in a corner (four types).



Nonsymmetric Half-Plane Masks

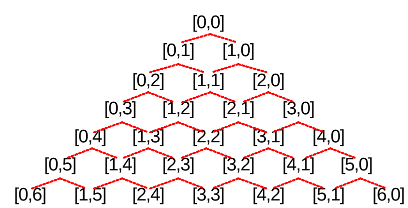
- Nonsymmetric half-plane masks have their "hole" at a jag in the side of the mask (eight types).



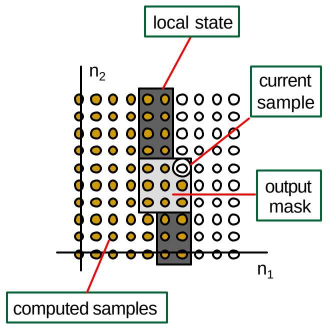
- Most general form of recursive system.
- To get a rectangular shaped output requires computing extraneous output samples.

Ordering the Computations

- There are an infinite number of possible orderings for recursively computable filters.
- Not all of them are equivalent in terms of storage, parallelism.

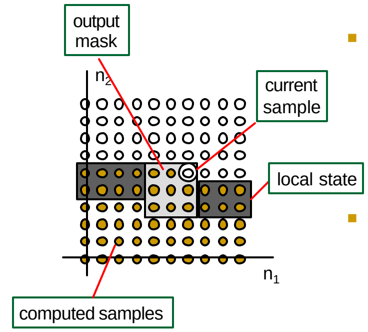


Column-wise Implementation



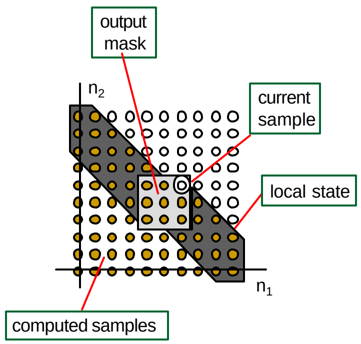
- For an L_1xL_2 point output mask, the column-wise implementation requires approximately L_1-1 **columns** of storage.
- L_1L_2 multiplications per output sample.

Row-wise Implementation



- For an L_1xL_2 point output mask, the row-wise implementation requires approximately L_2-1 **rows** of storage.
- L_1L_2 multiplications per output sample.

Diagonal Implementation



- For an L_1xL_2 point output mask, the row-wise implementation requires approximately L_1+L_2-1 **diagonals** of storage.
- L_1L_2 multiplications per output sample.
- Maximally parallel
  - All elements on the diagonal can be computed simultaneously.

Boundary Conditions

- If the system is to be LSI, the initial conditions must be **zero** and they must lie **outside** the support of the output.
- Why?
  - Linear implies zero input ➔ zero output. zero output ➔ zero boundary values.
  - LTI implies
    - $y[n_1,n_2] = x[n_1,n_2] ** h[n_1,n_2]$ must be the same as the result of solving the difference equation. Therefore, the only place we can place boundary conditions is outside the support of $y[n_1,n_2]$.

Finding the Support of $y[n_1,n_2]$

- Once we define 2-D z-transforms we can say

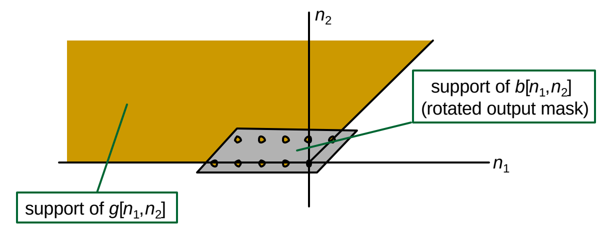
$$\begin{aligned} Y_z(z_1,z_2) &= X_z(z_1,z_2)H_z(z_1,z_2) \\ &= X_z(z_1,z_2)\frac{A_z(z_1,z_2)}{B_z(z_1,z_2)} \\ &= X_z(z_1,z_2)A_z(z_1,z_2)\frac{1}{B_z(z_1,z_2)} \\ y[n_1,n_2] &= x[n_1,n_2] ** a[n_1,n_2] ** g[n_1,n_2] \end{aligned}$$

support known — support known — ?

- If we knew the support of g, we could figure out the support of y.

Getting the Support of $g[n_1,n_2]$

- The support of $g[n_1,n_2]$ will be "wedge-shaped" with its vertex at the origin. It will be the **smallest** wedge that **contains** b.



2-D z-Transforms

$$H_z(z_1,z_2) = \sum_{n_1,n_2} h[n_1,n_2] z_1^{-n_1} z_2^{-n_2}$$

- Those values of (z_1,z_2) where the sum converges absolutely constitute the **region of convergence**.
- In general the region of convergence is a 4-D "annulus" called a **Reinhardt domain**.
- The set of points (z_1,z_2) with $z_1=e^{j\omega_1}$, $z_2=e^{j\omega_2}$ is called the **unit bicircle**.
- If the region of convergence includes the unit bicircle, then the Fourier transform exists and, if it is a system function, the system is stable.

z-Transform Properties

- Linearity

$$ax + by \longleftrightarrow aX_z + bY_z$$

- Shift

$$x[n_1-m_1,n_2-m_2] \longleftrightarrow z_1^{-m_1} z_2^{-m_2} X_z(z_1,z_2)$$

- Convolution

$$x ** y \longleftrightarrow X_z(z_1,z_2)Y_z(z_1,z_2)$$

- Separable signals

$$x[n_1]y[n_2] \longleftrightarrow X_z(z_1)Y_z(z_2)$$

Linear Mapping Property

- If

$$x[n_1,n_2] = \begin{cases} w[m_1,m_2], & n_1 = Im_1 + Jm_2; n_2 = Km_1 + Lm_2 \\ 0, & \text{else} \end{cases}$$

$$IL - JK \neq 0$$

then

$$X_z(z_1,z_2) = W_z(z_1^I z_2^J, z_1^K z_2^L)$$

- Note that *x* is an "expanded" version of *w*. Every sample of *w* appears in *x*. (The property does not work the other way.)

System Functions

$$\sum_{k_1}\sum_{k_2} b[k_1,k_2]y[n_1-k_1,n_2-k_2] = \sum_{r_1}\sum_{r_2} a[r_1,r_2]x[n_1-r_1,n_2-r_2]$$

$$b[n_1,n_2] ** y[n_1,n_2] = a[n_1,n_2] ** x[n_1,n_2]$$

$$B_z(z_1,z_2)Y_z(z_1,z_2) = A_z(z_1,z_2)X_z(z_1,z_2)$$

$$\frac{Y_z(z_1,z_2)}{X_z(z_1,z_2)} = \frac{A_z(z_1,z_2)}{B_z(z_1,z_2)} = H_z(z_1,z_2)$$

- System functions derived from 2-D difference equations are 2-D rational functions.
- $A_z(z_1,z_2)$ derived from input coefficients; $B_z(z_1,z_2)$ derived from output coefficients.

System Functions and Filter Support

- 1st quadrant filter: Negative powers of z_1 and z_2.
- 2nd quadrant filter: Positive powers of z_1; negative powers of z_2.
- 3rd quadrant filter: Positive powers of z_1 and z_2.
- 4th quadrant filter: Negative powers of z_1; positive powers of z_2.

- NSHP filter: Positive **AND** negative powers of one variable. Positive **OR** negative powers of the other.

Inverse 2-D *z*-Transforms

- It is rarely possible to perform inverse 2-D *z*-transforms analytically.
- For signals with NSHP support, they can be evaluated by recursing the difference equation.
- One known pair:

$$\frac{(n_1+n_2)!}{n_1!n_2!}a^{n_1}b^{n_2}u[n_1,n_2] \longleftrightarrow \frac{1}{1-az_1^{-1}-bz_2^{-1}}$$

Root Maps

- Analogous to pole-zero plots in 1-D.
- Roots of higher dimensional polynomials are not isolated points, but instead are manifolds.
- Example:

$$B_z(z_1, z_2) = 1 - b z_1^{-1} z_2^{-1}$$

  - This is zero whenever

$$z_1 = b / z_2$$

  - This traces out a two-dimensional surface in the four-dimensional *z*-space.

Root Maps (continued)

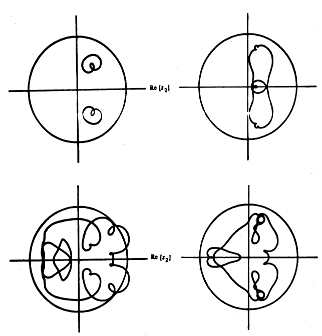
- Definition:

A **root map** consists of two parts:

1. The loci of roots of $B_z(z_1, z_2)$ in the z_2-plane as z_1 moves around its unit circle.
2. The loci of roots of $B_z(z_1, z_2)$ in the z_1-plane as z_2 moves around its unit circle.

Root Map Examples

- The root maps on the right all correspond to first-quadrant lowpass IIR filters.
- One of those filters is unstable.
- Which one?



Linear Mapping Theorem

- Let $h[n_1, n_2]$ be the impulse response of a system with NSHP support. Then there exists a linear mapping *T* such that

$$h[\vec{m}] = \begin{cases} g[\vec{T}\vec{m}], & \vec{T}\vec{m} = \text{integer vector} \\ 0, & \text{otherwise} \end{cases}$$

  is a first-quadrant system. Furthermore, both *g* and *h* are stable or both are unstable.

- This implies that we only have to develop stability tests for first-quadrant filters.

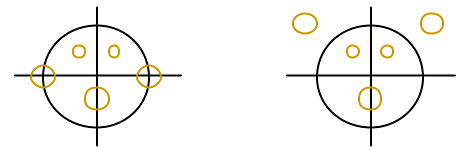
Stability Theorems

- Let $H_z(z_1, z_2) = \dfrac{1}{B_z(z_1, z_2)}$ be a 1st-quadrant recursive filter
- Shanks' Theorem: The filter is stable iff

  $$B_z(z_1, z_2) \neq 0, \quad |z_1| \geq 1, \ |z_2| \geq 1$$

- DeCarlo-StrintzisTheorem: The filter is stable if
  - (a) $B_z(z_1, z_2) \neq 0, \quad |z_1| = 1, \ |z_2| = 1$
  - (b) $B_z(1, z_2) \neq 0, \quad |z_2| \geq 1$
  - (c) $B_z(z_1, 1) \neq 0, \quad |z_1| \geq 1$

Heuristic Discussion



Class A Class B

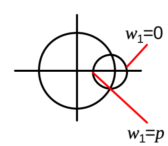
- All "rootlets" are closed connected curves.
- A Class A root map intersects the unit circle.
- A Class B root map does not intersect the unit circle.
- All Class A root maps correspond to **unstable** systems.
  - Follows from Shanks' Theorem.
- Class B root maps are stable if all "rootlets" lie inside the unit circle.

Decarlo-Strintzis Theorem Reconsidered

- DeCarlo-StrintzisTheorem: The filter is stable if
  - (a) $B_z(z_1, z_2) \neq 0, \quad |z_1| = 1, \ |z_2| = 1$
  - (b) $B_z(1, z_2) \neq 0, \quad |z_2| \geq 1$
  - (c) $B_z(z_1, 1) \neq 0, \quad |z_1| \geq 1$
- DeCarlo-StrintzisTheorem (restated): The filter is stable if
  - (a) It is of Class B.
  - (b) The rootlets in the z_2-plane are inside the unit circle.
  - (c) The rootlets in the z_1-plane are inside the unit circle.
- Note: Test (b) and (c) before testing (a) to save time.

Example

- $$B_z(e^{j\omega_1, z_2}) = 1 - ae^{-j\omega_1} - bz_2^{-1} - ce^{-j\omega_1}z_2^{-1}$$
- Setting B_z=0 and solving $z_2 = \dfrac{b + ce^{-j\omega_1}}{1 - ae^{-j\omega_1}}$



$$\left|\frac{b-c}{1+a}\right| < 1 \qquad \left|\frac{a-c}{1+b}\right| < 1$$

$$\left|\frac{b+c}{1-a}\right| < 1 \qquad \left|\frac{a+c}{1-b}\right| < 1$$

Separable IIR Filters

- **Form:**

  $h[n_1,n_2] = f[n_1]g[n_2]$
- **Stability:**

  h is stable if and only if *f* and *g* are stable 1-D filters.
- **Implementation:**
  - 1. Apply the 1-D recursion corresponding to *f* to each row of the input.
  - 2. Apply the 1-D recursion corresponding to *g* to each column of the result.

Zero-Phase Implementations

- There is a two-pass IIR filter implementation that will achieve zero-phase filtering.

$$h[n_1, n_2] ** h^*[-n_1, -n_2] \longleftrightarrow H(\omega_1, \omega_2)H^*(\omega_1, \omega_2) = |H(\omega_1, \omega_2)|^2$$

- The overall frequency response must be nonnegative.
- The second filter must recurse in the opposite direction.
- Getting the initial conditions right for the backwards recursion is tricky if done right.