

REPRESENTATION OF LINEAR DIGITAL NETWORKS

1. Lecture 11 - 52 minutes

Digital Networks

N^{th} order Difference Eq

$$y(n) = \sum_{k=1}^N a_k y(n-k) + \sum_{k=0}^M b_k x(n-k)$$

$$y(n) = \sum_{k=1}^N a_k y(n-k) + \sum_{k=0}^M b_k x(n-k)$$

Basic Operations:

delay $x(n) \xrightarrow{z^{-1}} x(n-1)$
 $X(z) \xrightarrow{z^{-1}} z^{-1} X(z)$

Multiplication $x(n) \xrightarrow{a} ax(n)$

Addition $x_1 \xrightarrow{+} x_2 \rightarrow x_1 + x_2$

Example: $y(n) = ay(n-1) + x(n)$

Signal Flow Graph

branch (jk): input = w_j

output $\triangleq V_{jk} = f_{jk}(w_j)$

Node value = \sum outputs of entering branches

S_{jk} j^{th} source to k^{th} node

a.

$W_k = \underbrace{\sum_{j=1}^N V_{jk}}_{\text{Network}} + \underbrace{\sum_{j=1}^M S_{jk}}_{\text{Source}}$

Matrix Representation of Digital Networks

Linear Signal Flow Graph arbitrary branch (jk)

$$V_{jk} = t_{jk} W_j$$

$$y(n) = a y(n-1) + x(n)$$

$z^{-1} y(z)$

$W_k(z) = \sum_{j=1}^N V_{jk}(z) + \sum_{j=1}^M S_{jk}(z)$

$V_{jk}(z) = F_{jk}(z) W_j(z)$

$S_{jk}(z) = b_{jk} X_j(z)$

$$W_k(z) = \sum_{j=1}^N F_{jk}(z) W_j(z) + \sum_{j=1}^M b_{jk} X_j(z)$$

$$\begin{bmatrix} W_1 \\ W_2 \\ \vdots \\ W_N \end{bmatrix} = \begin{bmatrix} F_{11} & F_{12} & \dots & F_{1N} \\ F_{21} & F_{22} & \dots & F_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ F_{N1} & F_{N2} & \dots & F_{NN} \end{bmatrix} \begin{bmatrix} W_1 \\ W_2 \\ \vdots \\ W_N \end{bmatrix} + \begin{bmatrix} b_{11} & \dots & b_{1M} \\ \vdots & \vdots & \vdots \\ b_{N1} & \dots & b_{NM} \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_M \end{bmatrix}$$

b.

$\underline{W}(z) = \underline{F}^{\dagger}(z) \underline{W}(z) + \underline{B}^{\dagger} \underline{X}(z)$ or: $\underline{W}(n) = \underline{F}_c^{\dagger} \underline{W}(n) + \underline{F}_d^{\dagger} \underline{W}(n-1) + \underline{B}^{\dagger} \underline{X}(n)$

$\underline{F}(z) = \{F_{kj}(z)\}$

$\underline{F}^{\dagger}(z) = \underline{F}_c^{\dagger} + z^{-1} \underline{F}_d^{\dagger}$

$\underline{Y}(z) = \underline{C}^{\dagger} \underline{W}(z)$

Example

$W_1(n) = b W_3(n)$

$W_2(n) = W_1(n-1) + W_3(n-1)$

$W_3(n) = a W_2(n) + X(n)$

c.

$$\begin{bmatrix} w_1(n) \\ w_2(n) \\ w_3(n) \end{bmatrix} = \begin{bmatrix} a & 0 & 0 \\ 0 & a & 0 \\ 0 & 0 & a \end{bmatrix} \begin{bmatrix} w_1(n-1) \\ w_2(n-1) \\ w_3(n-1) \end{bmatrix}$$

$$+ \begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} w_1(n-1) \\ w_2(n-1) \\ w_3(n-1) \end{bmatrix}$$

$$+ \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \chi(n)$$

Network Computable

Nodes can be numbered
so that F_c^T is

$$F_c^T = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \vdots & \vdots & \vdots \\ 0 & 0 & 0 \end{bmatrix}$$

Non computable Network

d.

2. Comments

The class of discrete-time systems characterized by linear constant-coefficient difference equations plays a particularly important role in digital signal processing, in part because many of the issues involved in their implementation are well understood. In this lecture we discuss the representation of this class of systems in terms of signal-flow graphs and matrices. In the two following lectures we will utilize this representation to discuss a number of basic network structures for infinite impulse response and finite impulse response digital filters.

This lecture is directed primarily toward introducing the notation of signal-flow graphs and the corresponding matrix representation. Although the concept of computability is discussed briefly in the lecture we will not be relying on this idea in the following lectures. The notion of computability emphasizes, however, the fact that we are viewing the signal-flow graph as representing a computational algorithm for the system, i.e. it represents an implementation of a digital filter. Some flow-graph representations of a digital filter may be mathematically valid but cannot be implemented. This class corresponds to signal-flow graphs which contain delay-free loops.

3. Reading

Text: Sections 6.0 (page 290) through 6.2.

4. Problems

Problem 11.1

Consider the discrete-time system represented by the linear constant-

coefficient difference equation

$$y(n] - \frac{3}{4} y[n - 1] + \frac{1}{8} y[n - 2] = x[n]$$

(a) Draw a block-diagram representation of the system in terms of adders and delay and coefficient multiplication branches.

(b) Draw a linear signal-flow graph representation of the system.

Problem 11.2

In figure P11.2-1 is shown a digital network.

(a) For this network, determine the matrices F^t , B^t and C^t as specified on chalkboard (c).

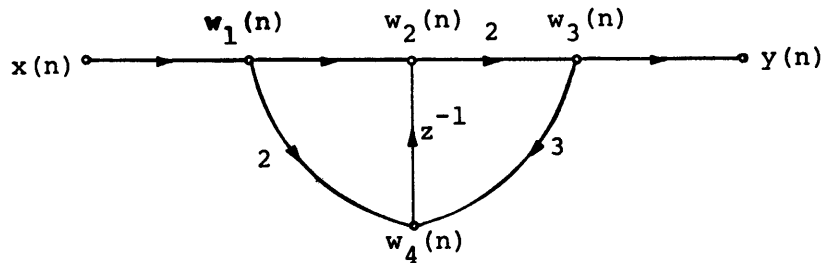


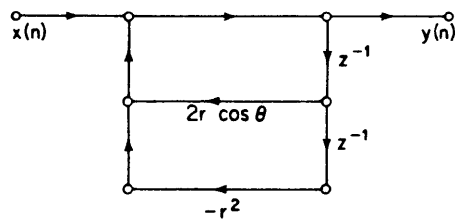
Figure P11.2-1

(b) Determine the matrices F_c^t and F_d^t in the alternative matrix representation on chalkboard (c).

Problem 11.3

Determine the system functions of the two networks in figure P11.3-1 and show that they have the same poles.

Network 1



Network 2

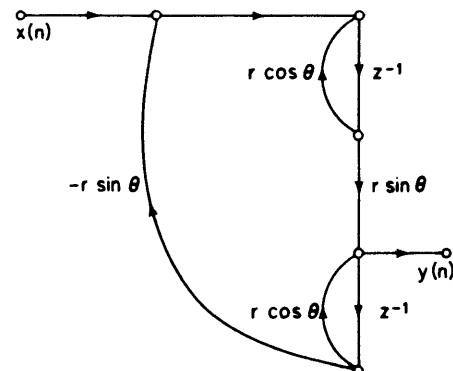


Figure P11.3-1

Problem 11.4

For the network shown in figure P11.4-1, determine all possible ordering of the nodes such that the node variables can be computed in sequence.

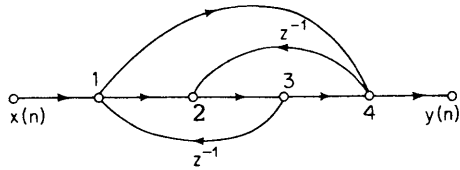


Figure P11.4-1

Problem 11.5*

In figure 11.5-1 is indicated a digital network A with two inputs, $x_1(n)$ and $x_2(n)$, and two outputs, $y_1(n)$ and $y_2(n)$.

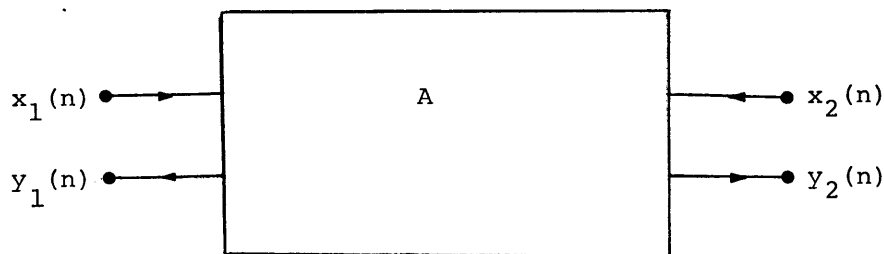


Figure P11.5-1

The network A can be described in terms of the two-part set of equations

$$Y_1(z) = H_1(z) + H_2(z) X_2(z)$$

$$Y_2(z) = H_3(z) + H_4(z) X_2(z)$$

where

$$H_1(z) = \frac{1}{1 - \frac{1}{2}z^{-1}}$$

$$H_2(z) = 1$$

$$H_3(z) = \frac{1 + 2z^{-1}}{1 + \frac{1}{2}z^{-1}}$$

$$H_4(z) = \frac{1}{1 + \frac{1}{2}z^{-1}}$$

(a) Draw a flow-graph implementation of the network. The transmittance of each branch must be a constant or a constant times z^{-1} . Higher-order functions of z^{-1} cannot be used as branch transmittances.

(b) We wish now to connect a network B to the network A as indicated in figure P11.5-2.

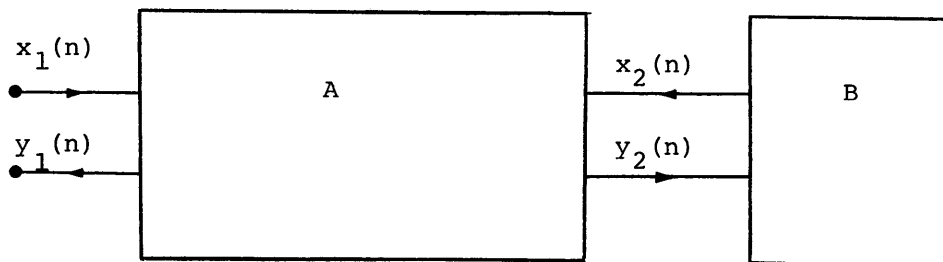


Figure P11.5-2

Network B has no internal source nodes and is characterized by the system function $X_2(z) = Y_2(z) H_B(z)$. Note that because of the notation used for network A, $y_2(n)$ denotes the input to network B and $x_2(n)$ is the output. By examining the flow graph in part (a) determine a necessary and sufficient condition on $H_B(z)$ so that the overall system is computable. If you have difficulty, first try the two possibilities $H_B(z) = 1$ and $H_B(z) = z^{-1}$.

MIT OpenCourseWare
<http://ocw.mit.edu>

Resource: Digital Signal Processing
Prof. Alan V. Oppenheim

The following may not correspond to a particular course on MIT OpenCourseWare, but has been provided by the author as an individual learning resource.

For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.