4.5 Discrete Orthogonal Transforms

A. R. Várkonyi-Kóczy and Márk Fék

1997

Department of Measurement and Information Systems
Technical University of Budapest

Discrete Orthogonal Transforms

The main advantages of applying signal transforms:

- Better interpretation of the signal
- Computational efficiency
- Data reduction

The one-dimensional discrete linear signal transforms can be:

- Sequence to vector - signal analysis (S - V)
- Vector to sequence - signal synthesis (V - S)

Properties of Discrete Orthogonal Transforms

- Calculated as a matrix-vector product where
  - The input is a vector of size N
  - The invertible matrix of size $N * N$ contains the weights
- The signal space is a
  - Vector space
  - With the elementary vector algebraic operations

- Improvements can be achieved utilizing
  - The parallelism and
  - Symmetry properties of the transform
- Interpretation of the transform
  - Block oriented - single operation
  - Recursion techniques - continuous flow
  - Combined techniques

- Discrete signals
  - Simple sequences
  - Typically with a limited length
- The transformed signal
  - Typically block oriented operations
  - The representation has the same length/size
  - Emphasize the features
  - Further operations have lower complexity

- Problems may occur in case of
  - Large amount of input data
  - Continuous data flow
- Solution
  - Smaller overlapping blocks can be used
  - But it limits the resolution of the representation
- Signal representation
  - Approximation
  - Performed with the help of known signals

- Signal spaces
  - Generalizations of Euclidean space
  - Every point represents a signal
  - Can be extended to stochastic signals, too
  - Distance remains valid
  - Metric operations
  - Algebraic operations
  - Linearity
  - Norm
  - Scalar product
  - Inner product
    - Orthogonality of two signals can be defined

If the signal is

$\underline{y} = [y_0, y_1, \ \dots \ , y_{N-1}]^T$

and the coordinate system of linearly independent vectors is:

$\underline{c}_m = [c_{0m}, c_{1m}, \dots, c_{N-1,m}]^T$, $m = 0, 1, \ \dots \ , N-1$

then we can write:

$\underline{y} = \sum_{m=0}^{N-1} x_m \underline{c}_m$

Which is equivalent to:

$\underline{y} = \underline{\underline{C}}\, \underline{x}$

where $\underline{x} = [x_0, x_1, \ \dots \ , x_{N-1}]^T$,

and $\underline{\underline{C}} = [\underline{c}_0, \underline{c}_1, \ \dots \ , \underline{c}_{N-1}]$

The representation of $\underline{y}$ in the system of $\underline{\underline{C}}$ is $\underline{x}$:

$$\underline{x} = \underline{\underline{C}}^{-1}\underline{y} = \underline{\underline{T}}\,\underline{y}$$

where $\underline{\underline{T}}$ denotes the transformation matrix.

- columns of $\underline{\underline{T}}$ form the basis sytem for $\underline{x}$
- rows of $\underline{\underline{T}}$ form the reciprocal basis system

The basis systems are related by: $\underline{t}_m^T \underline{c}_n = \delta_{mn}\ , \quad m, n = 0, 1, ..., N-1$

where $\delta_{mn} = 1$, if $m = n$, otherwise 0.

- The matrix $\underline{\underline{T}}$ corresponds to the normal transformation
- The matrix $\underline{\underline{C}}$ corresponds to the inverse transformation

For the well-known DFT:

$$t_{mn} = \frac{1}{N}\, e^{-j\,\frac{2\pi}{N}\,mn}\ , \qquad c_{nm} = e^{j\,\frac{2\pi}{N}\,mn} \quad n, m = 0, 1, ..., N-1\ .$$

Unitary Transforms

- $\underline{\underline{T}} = \underline{\underline{C}}^{-1} = \alpha \underline{\underline{C}}^{*}$, where α is a scalar multiplier
- Their basis (and reciprocal basis) system is orthogonal:

  $\alpha \underline{c}_m^{*} \underline{c}_n = \delta_{mn} \ , \quad m, n = 0, ..., N-1$
- If $\alpha = 1$ then the system has an orthonormal base

The Most Important Discrete Orthogonal Transforms

- DFT: Discrete Fourier Transform

  $t_{mn} = \frac{1}{N} e^{-j\frac{2\pi}{N}mn}$

  $c_{mn} = e^{j\frac{2\pi}{N}mn}$

  $m, n = 0, ..., N-1$

- c-DFT: Complementary Discrete Fourier Transform

  $t_{mn} = \frac{1}{N} e^{-j\frac{(2m+1)\pi}{N}n}$

  $c_{mn} = e^{j\frac{(2m+1)\pi}{N}n}$

  $m, n = 0, ..., N-1$

- DHT: Discrete Hartley Transform

$$t_{mn} = \frac{1}{N}\mathsf{cas}(\frac{2\pi}{N}mn) = \frac{1}{N}\left[\cos(\frac{2\pi}{N}mn) + \sin(\frac{2\pi}{N}mn)\right]$$

$$c_{mn} = \frac{1}{N}\mathsf{cas}(\frac{2\pi}{N}mn) \qquad m,n = 0,...,N-1$$

- DCT-I: Discrete Cosine Transform-I

$$t_{mn} = c_{mn} = \left(\begin{array}{ll} \sqrt{\frac{1}{N}}\ , & m,n=0 \\ \sqrt{\frac{2}{N}}\cos(\frac{\pi}{N}mn)\ , & m,n = 1,...,N-1 \end{array} \right.$$

- DST-I: Discrete Sine Transform-I

$$t_{mn} = c_{mn} = \sqrt{\frac{2}{N}} \sin\left(\frac{\pi}{N} mn\right) \qquad m, n = 1, ..., N-1$$

- DCT-II: Discrete Cosine Transform-II

$$t_{mn} = c_{mn} = \left(\begin{array}{ll} \sqrt{\frac{1}{N}}\,, & m = 0 \\ \sqrt{\frac{2}{N}} \cos\left[\frac{\pi}{N} m(n+0.5)\right]\,, & n = 0; \quad m, n = 1, ..., N-1 \end{array}\right.$$

- DST-II: Discrete Sine Transform-II

$$t_{mn} = c_{mn} = \left(\begin{array}{ll} \sqrt{\frac{2}{N}} \sin\left[\frac{\pi}{N} m(n+0.5)\right] \;, & n = 0;\; m,n = 1,...,N-1 \\ (-1)^n \sqrt{\frac{1}{N}} & m = N \end{array} \right.$$

- DCT-III: Discrete Cosine Transform-III

$$t_{mn} = c_{mn} = \left(\begin{array}{ll} \sqrt{\frac{1}{N}} \;, & n = 0 \\ \sqrt{\frac{2}{N}} \cos\left[\frac{\pi}{N} n(m+0.5)\right] \;, & m = 0; \\ & m,n = 1,...,N-1 \end{array} \right.$$

- DST-III: Discrete Sine Transform-III

$$t_{mn} = c_{mn} = \left(\begin{array}{ll} \sqrt{\frac{2}{N}} \sin\left[\frac{\pi}{N} n(m+0.5)\right] \ , & m = 0;\ m, n = 1, ..., N-1 \\ (-1)^m \sqrt{\frac{1}{N}} \ , & n = N \end{array} \right.$$

- DCT-IV: Discrete Cosine Transform-IV

$$t_{mn} = c_{mn} = \sqrt{\frac{2}{N}} \cos\left[(n+0.5)(m+0.5)\frac{\pi}{N}\right] \ , m, n = 1, ..., N-1$$

- DST-IV: Discrete Sine Transform-IV

  $t_{mn} = c_{mn} = \sqrt{\frac{2}{N}} \sin\left[(m+0.5)(n+0.5)\frac{\pi}{N}\right]$,

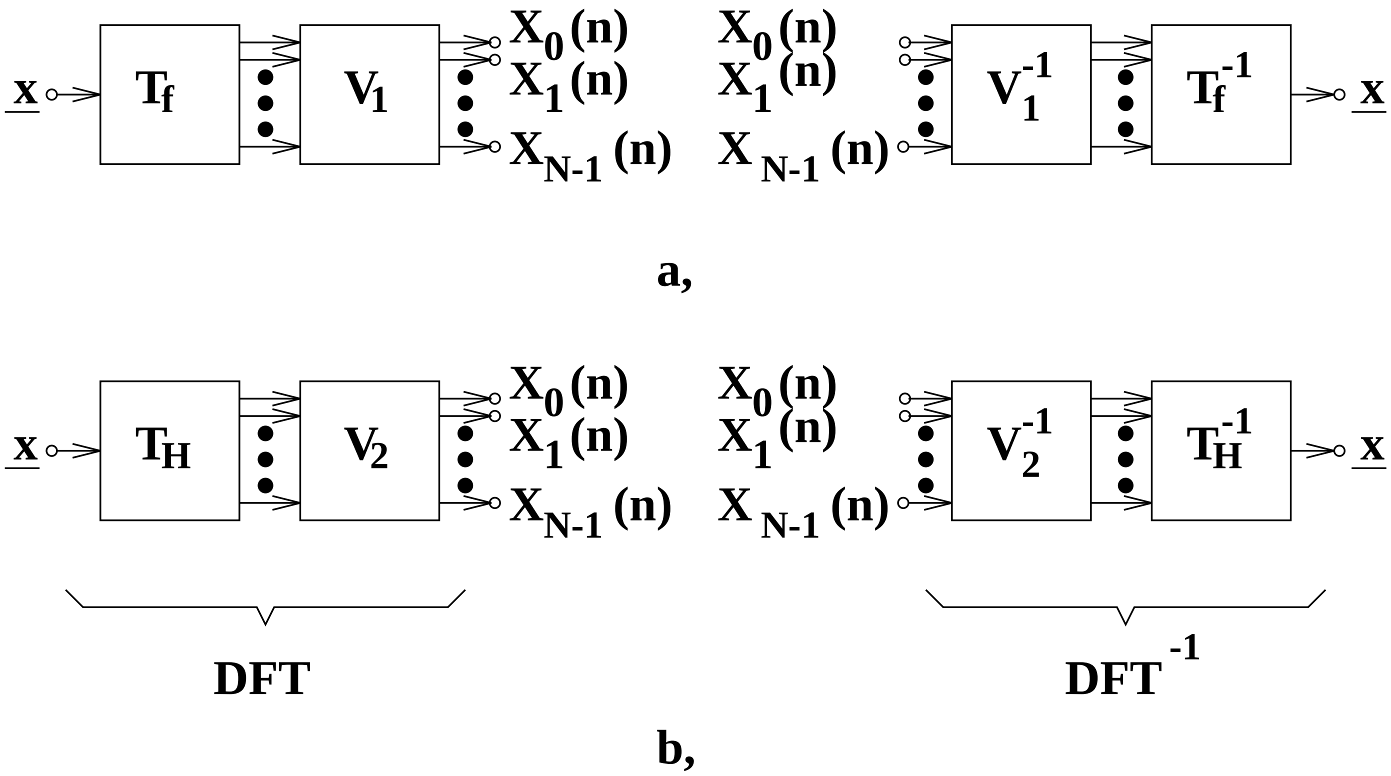
  $m, n = 0, 1, ..., N-1$

- WHT: Walsh-Hadamard Transform

  $t_{mn} = \frac{1}{N}\mathsf{wal}(m,n)$; $c_{mn} = \mathsf{wal}(m,n)$, where

  $\mathsf{wal}(m,n) = (-1)^{<m,n>}$ with $<m,n> = \sum_{k=0}^{M-1} m_k n_k$,

  $M = \log_2 N$, m_k and n_k are the coefficients of the binary representation of m and n


x
T_f
V_1
$X_0(n)$
$X_1(n)$
$X_{N-1}(n)$
$X_0(n)$
$X_1(n)$
$X_{N-1}(n)$
V_1^{-1}
T_f^{-1}
x
a,
x
T_H
V_2
$X_0(n)$
$X_1(n)$
$X_{N-1}(n)$
$X_0(n)$
$X_1(n)$
$X_{N-1}(n)$
V_2^{-1}
T_H^{-1}
x
DFT
DFT^{-1}
b,

Karhunen-Loève Transform

- Its objective is data compression
- Orthogonal transform
- $(N - M)$ components are discarded
- The signal is reconstructed using the retained M components
- Mean-square error criterion is applied

Deducting the Optimal Orthogonal Transform

- $\underline{\underline{T}}$ orthogonal transform is given by

  $$\underline{\underline{T}}^T = [\underline{t}_0, \underline{t}_1, \ \dots\ , \underline{t}_{N-1}]; \quad \underline{t}_m^T \underline{t}_n = \begin{cases} 1, & m = n \\ 0, & m \neq n \end{cases}$$

- For each $\underline{X}$ we obtain

  $\underline{Y} = \underline{\underline{T}}\,\underline{X}$,

  where $\underline{X}^T = [x_0, x_1, \ \dots\ , x_{N-1}]$

  and $\underline{Y}^T = [y_0, y_1, \ \dots\ , y_{N-1}]$

- We know that

  $\underline{\underline{T}}^T \underline{\underline{T}} = \underline{\underline{I}}$, hence

  $\underline{X} = \underline{\underline{T}}^T \, \underline{Y} = [\underline{t}_0, \underline{t}_1, \ \dots\ , \underline{t}_{N-1}]\underline{Y}$

  which yields

  $\underline{X} = y_0\underline{t}_0 + y_1\underline{t}_1 + \ \dots\ + Y_{N-1}\underline{t}_{N-1} = \sum_{i=0}^{N-1} y_i\underline{t}_i$

- we wish
  - to retain a subset $\{y_0, y_1, \ \dots\ , y_M\}$
  - to estimate $\underline{X}$

- By replacing the remaining $N - M$ components of $\underline{Y}$ by constants b_i

  $\underline{\tilde{X}}(M) = \sum_{i=0}^{M-1} y_i \underline{t}_i + \sum_{i=M}^{N-1} b_i \underline{t}_i$

  where $\underline{\tilde{X}}(M)$ denotes the estimate of $\underline{X}$

- The error is calculated by neglecting the $N - M$ terms

  $\Delta\underline{X} = \underline{X} - \underline{\tilde{X}}(M) = \sum_{i=M}^{N-1} (y_i - b_i)\underline{t}_i$

  where $\Delta\underline{X}$ is the error vector

  $\Delta\underline{X} = \underline{X} - \sum_{i=0}^{M-1} y_i \underline{t}_i - \sum_{i=M}^{N-1} b_i \underline{t}_i$

- The mean-square error is

$$\varepsilon(M) = \mathsf{E}\left\{x\left|\left|\Delta \underline{X}^2\right|\right|\right\} = \mathsf{E}\left\{(\Delta \underline{X})^T(\Delta \underline{X})\right\}$$

$$\varepsilon(M) = \mathsf{E}\left\{\sum_{i=M}^{N-1}\sum_{j=M}^{N-1}(y_i - b_i)(y_j - b_j)\underline{t}_i^T\underline{t}_j\right\}$$

$$\varepsilon(M) = \sum_{i=M}^{N-1}\mathsf{E}\left\{(y_i - b_i)^2\right\}$$

- We want to minimize $\varepsilon(M)$ in a two steps process
  - choosing optimum b_i
  - choosing optimum $\underline{t}_i$

- The optimum b_i are obtained

  $\frac{\partial}{\partial b_i}\mathsf{E}\left\{(y_i - b_i)^2\right\} = -2\left[\mathsf{E}\{y_i\} - b_i\right] = 0$

  which yields $\quad b_i = \mathsf{E}\{y_i\}$

- We obtain from previous equations that

  $y_i = \underline{t}_i\underline{X}$,thus

  $b_i = \underline{t}_i^T \mathsf{E}\{\underline{X}\} = \underline{t}_i^T \underline{\bar{X}}$

- We can write $\varepsilon(M)$ as

  $\varepsilon(M) = \sum_{i=M}^{N-1} \mathsf{E}\left\{(y_i - b_i)^2\right\}$

- We substitute

  $y_i = \underline{t}_i^T \underline{X}$ and $b_i = \underline{t}_i \underline{\bar{X}}$

  to get

  $\varepsilon(M) = \sum_{i=M}^{N-1} \underline{t}_i^T \mathsf{E}\left\{(\underline{X} - \underline{\bar{X}})(\underline{X} - \underline{\bar{X}})^T\right\} \underline{t}_i$

  where

  $\underline{\underline{\Sigma}}_x = \mathsf{E}\left\{(\underline{X} - \underline{\bar{X}})(\underline{X} - \underline{\bar{X}})^T\right\}$ is the covariance matrix of $\underline{X}$

  We have

  $\varepsilon(M) = \sum_{i=M}^{N-1} \underline{t}_i^T \underline{\underline{\Sigma}}_x \underline{t}_i$

- We want to minimize $\varepsilon(M)$ with respect to $\underline{t}_i$
- The constraint is $\underline{t}_i \underline{t}_i^T = 1$
- The method of Lagrange multipliers is used:

  $\widehat{\varepsilon}(M) = \varepsilon(M) - \sum_{i=M}^{N-1} \beta_i [\underline{t}_i^T \underline{t}_i - 1] = \sum_{i=M}^{N-1} \left\{ \underline{t}_i^T \underline{\underline{\Sigma}}_x \underline{t}_i - \beta_i [\underline{t}_i^T \underline{t}_i - 1] \right\}$

  where β_i denotes the Lagrange multiplier
- Thus we get

  $\underline{\underline{\Sigma}}_x \underline{t}_i = \beta_i \underline{t}_i$
- $\underline{t}_i$ is the eigenvector of the covariance matrix $\underline{\underline{\Sigma}}_x$

- β_i is the ith eigenvalue

- Denoting β_i by λ_i we obtain

  $\varepsilon_{min}(M) = \sum_{i=M}^{N-1} \lambda_i$

- The KLT is the optimum with respect to the mean-square error

- Proper a priori knowledge is needed

- No a priori information is available
- Their basis vectors closely approximate the eigenvectors of a class of matrices called Toeplitz matrices

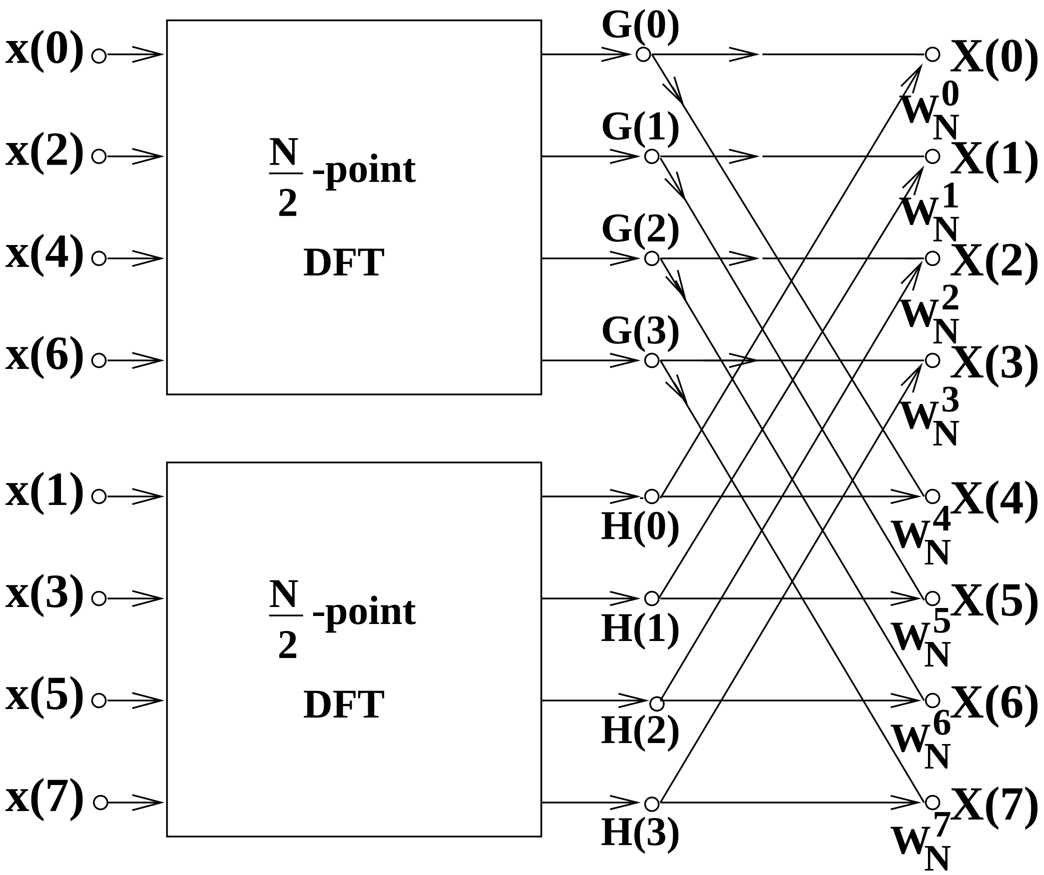
$$\underline{\underline{\Phi}} = \begin{bmatrix} 1 & \rho & \rho^2 & \dots & \rho^{N-1} \\ \rho & 1 & \rho & \dots & \rho^{N-2} \\ \rho^{N-1} & \rho^{N-2} & \rho^{N-3} & \dots & 1 \end{bmatrix} \quad 0 < \rho < 1$$

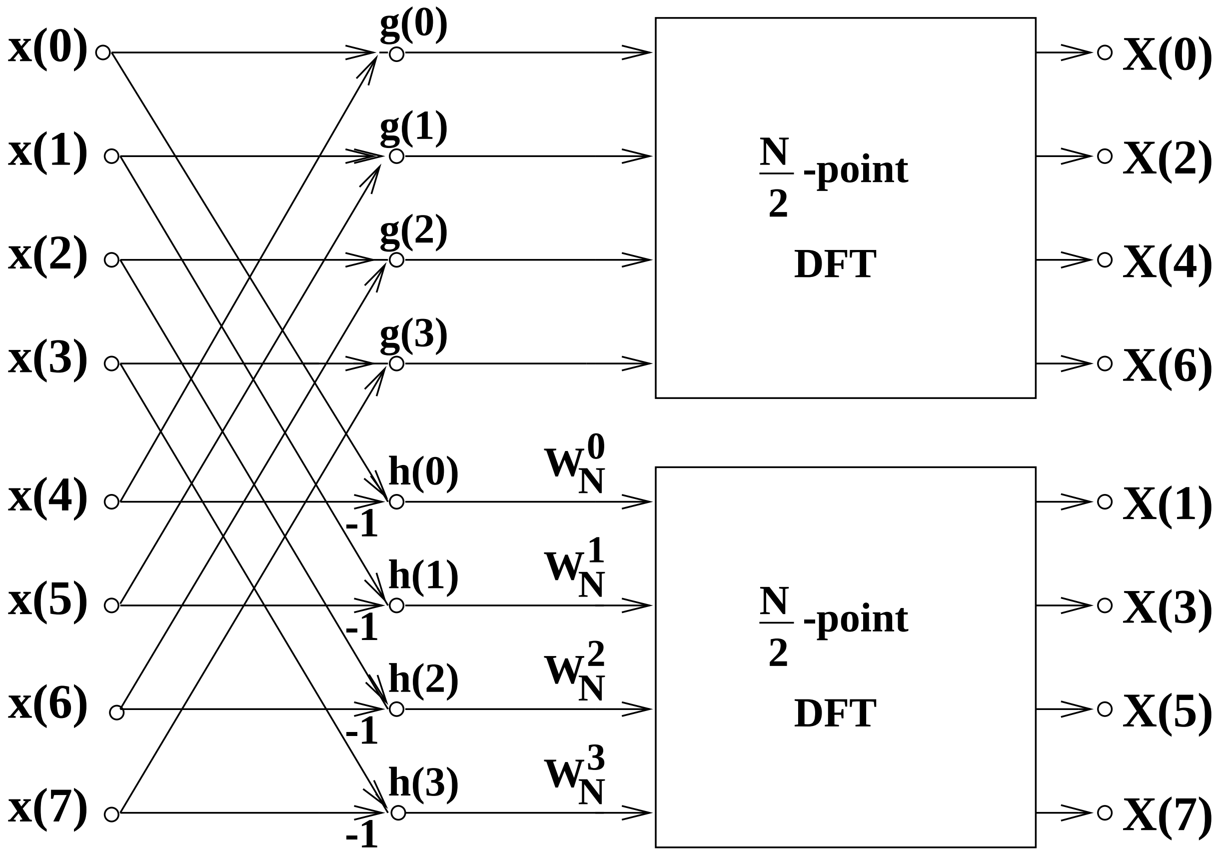
- DCTs are applied for:
  - image processing
  - data reduction

Fast algorithms

- Properties of DFT
  - orthogonality
  - symmetry
  - regularity
  - complex exponential kernel

- implementation
  - input side: decimation in time
  - symmetrically located input data pairs
  - output side: decimation in frequency


x(0)
x(2)
x(4)
x(6)
x(1)
x(3)
x(5)
x(7)
N/2 -point DFT
N/2 -point DFT
G(0)
G(1)
G(2)
G(3)
H(0)
H(1)
H(2)
H(3)
X(0)
X(1)
X(2)
X(3)
X(4)
X(5)
X(6)
X(7)
W_N^0
W_N^1
W_N^2
W_N^3
W_N^4
W_N^5
W_N^6
W_N^7


x(0)
x(1)
x(2)
x(3)
x(4)
x(5)
x(6)
x(7)
g(0)
g(1)
g(2)
g(3)
h(0)
h(1)
h(2)
h(3)
-1
-1
-1
-1
W_N^0
W_N^1
W_N^2
W_N^3
$\frac{N}{2}$ -point DFT
$\frac{N}{2}$ -point DFT
X(0)
X(2)
X(4)
X(6)
X(1)
X(3)
X(5)
X(7)

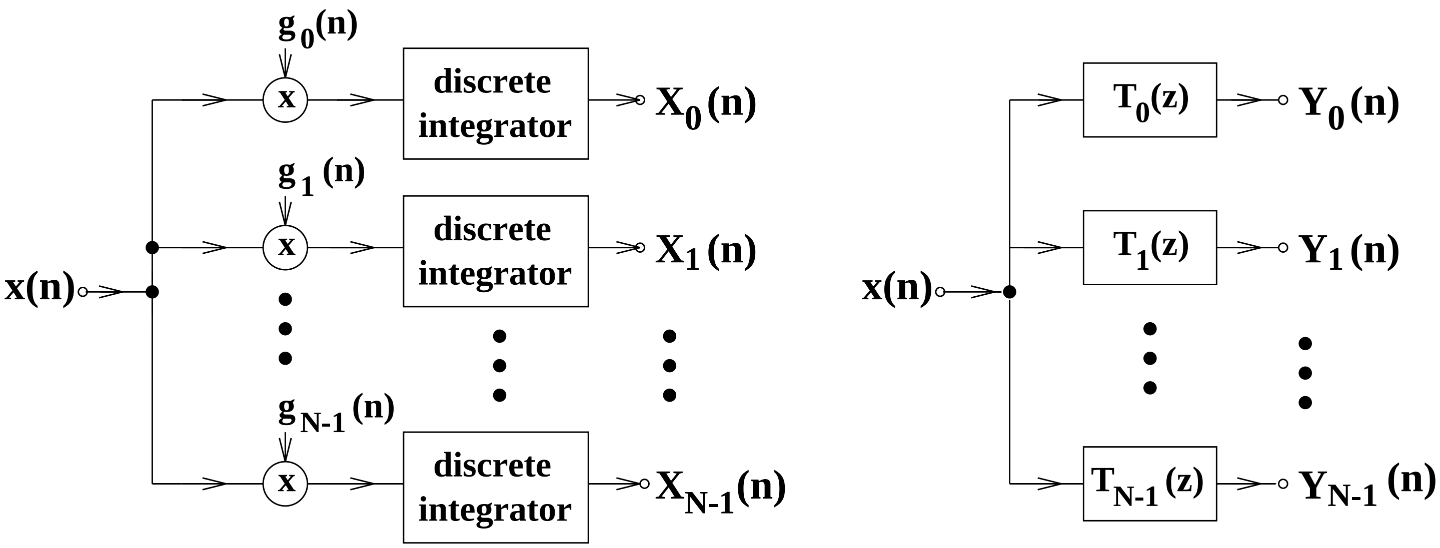
Time-recursive discrete orthogonal transforms

- continous input data flow
- sequence - vector transformer
- input is composed of the last N input samples
- window is shifted one by one
- $\underline{Y}(n) = \underline{\underline{T}}\,\underline{x}(n)$
- $\underline{Y}(n)$ is the output of a FIR filter-bank

- the transfer function of the mth filter is

$$T_m(z) = t_{m,N-1} + t_{m,N-2} z^{-1} + \,...\, + t_{m,0} z^{-(N-1)} \;, \;\; ; \quad m = 0, ..., N-1$$

- weights are related to the mth row of $\underline{\underline{T}}$
- block-oriented
- sliding-window transformation


g_0(n)
x
discrete integrator
X_0(n)
g_1(n)
x
discrete integrator
X_1(n)
x(n)
g_N-1(n)
x
discrete integrator
X_N-1(n)
T_0(z)
Y_0(n)
T_1(z)
Y_1(n)
x(n)
T_N-1(z)
Y_N-1(n)

a, b,

- Let's suppose periodic signals with periods of N
- The multiplier coefficients are the reciprocal basis vectors
- discrete integrator sums up the products
- complete result only for $n = N - 1$
- can be operated for arbitrary $n \geq 0$

- For the case of DFT

  $T_m(z) = \frac{1}{N}\left[z_m^{-(N-1)} + z_m^{-(N-2)}z^{-1} + \ \ldots\ + z^{-(N-1)}\right] =$

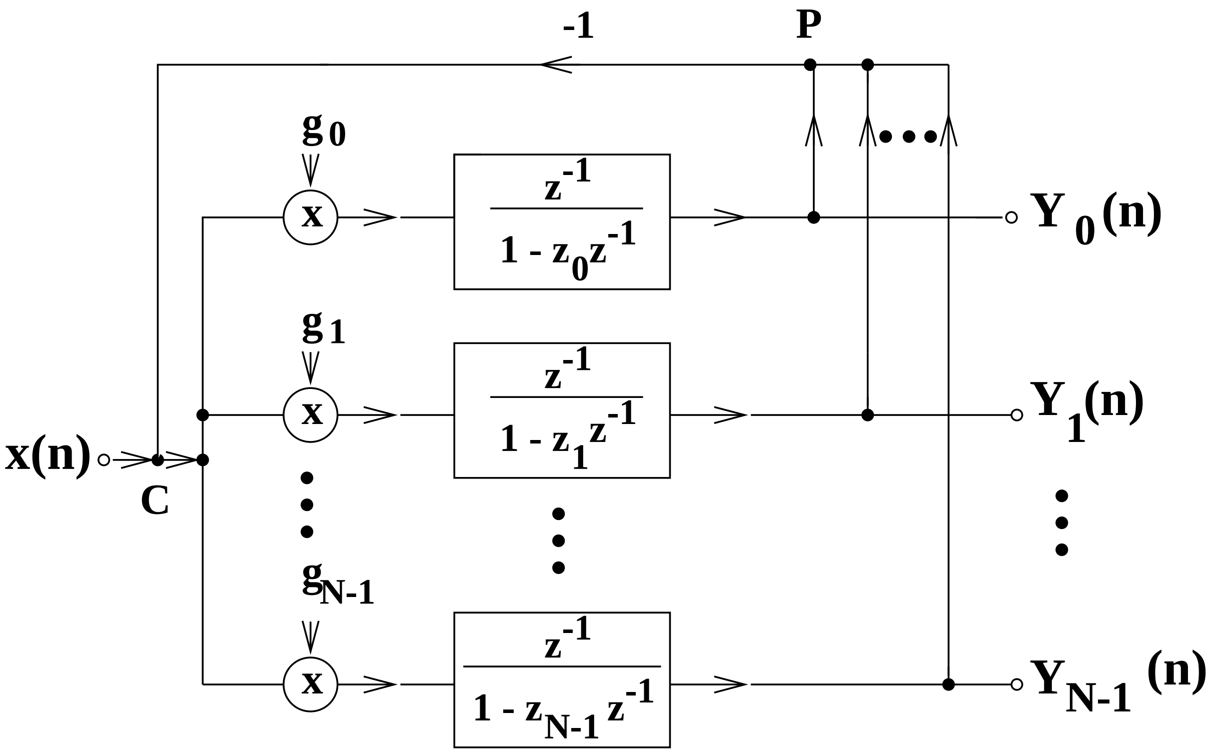
  $= \frac{z_m}{N}\ \left[1 + z_m z^{-1} + \ \ldots\ + z_m^{N-1} z^{-(N-1)}\right] =$

  $= \frac{z_m}{N}\ \frac{1-z^{-N}}{1-z_m z^{-1}}$

  where $z_m = e^{j(2\pi m/N)}$, $m = 0, 1,\ \ldots\ , N-1$

- Lagrange structure filter bank

- first-order IIR transfer functions having poles on the unit circle may lead to numerical problems under finite precision conditions


-1
P
g_0
g_1
g_{N-1}
x
x
x
$\frac{z^{-1}}{1 - z_0 z^{-1}}$
$\frac{z^{-1}}{1 - z_1 z^{-1}}$
$\frac{z^{-1}}{1 - z_{N-1} z^{-1}}$
x(n)
C
$Y_0(n)$
$Y_1(n)$
$Y_{N-1}(n)$

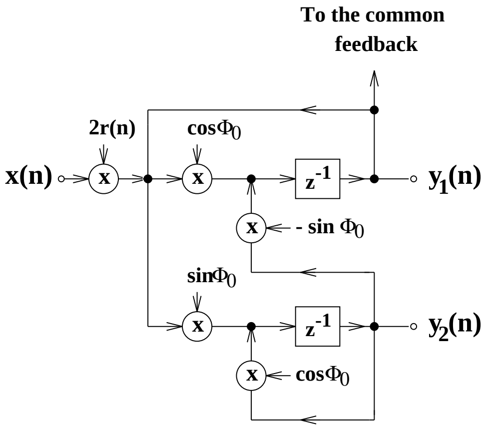
- Alternative structure
  - based on the observer theory
  - parallel
  - complex coefficient
  - first-order resonators within a common feedback loop
  - resonator poles are on the unit circle

- weighting coeficients are

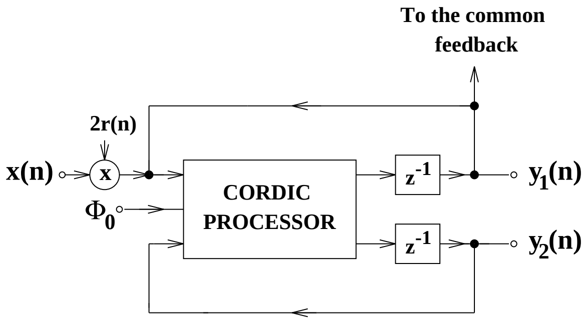
$$g_m = \frac{1}{N} z_m\ , \qquad m = 0, 1,\ \ldots\ , N-1$$

- transfeer function is

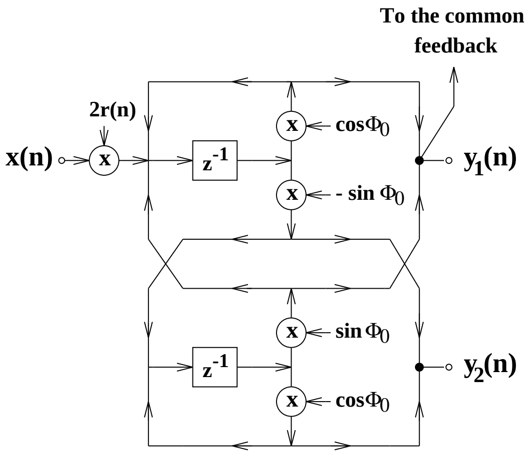
$$T_m(z) = \frac{1}{N}\ \left(z_m z^{-1} + z_m^2 z^{-2} +\ \ldots\ + z_m^N z^{-N}\right)$$


To the common feedback
2r(n)
cosΦ0
x(n)
y1(n)
- sin Φ0
sinΦ0
y2(n)
cosΦ0

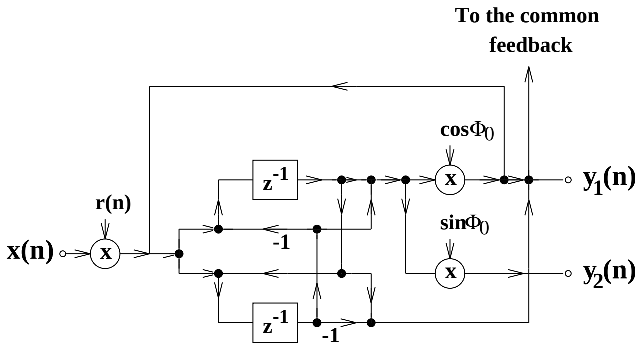
a,


To the common feedback
2r(n)
x(n)
Φ0
CORDIC PROCESSOR
y1(n)
y2(n)

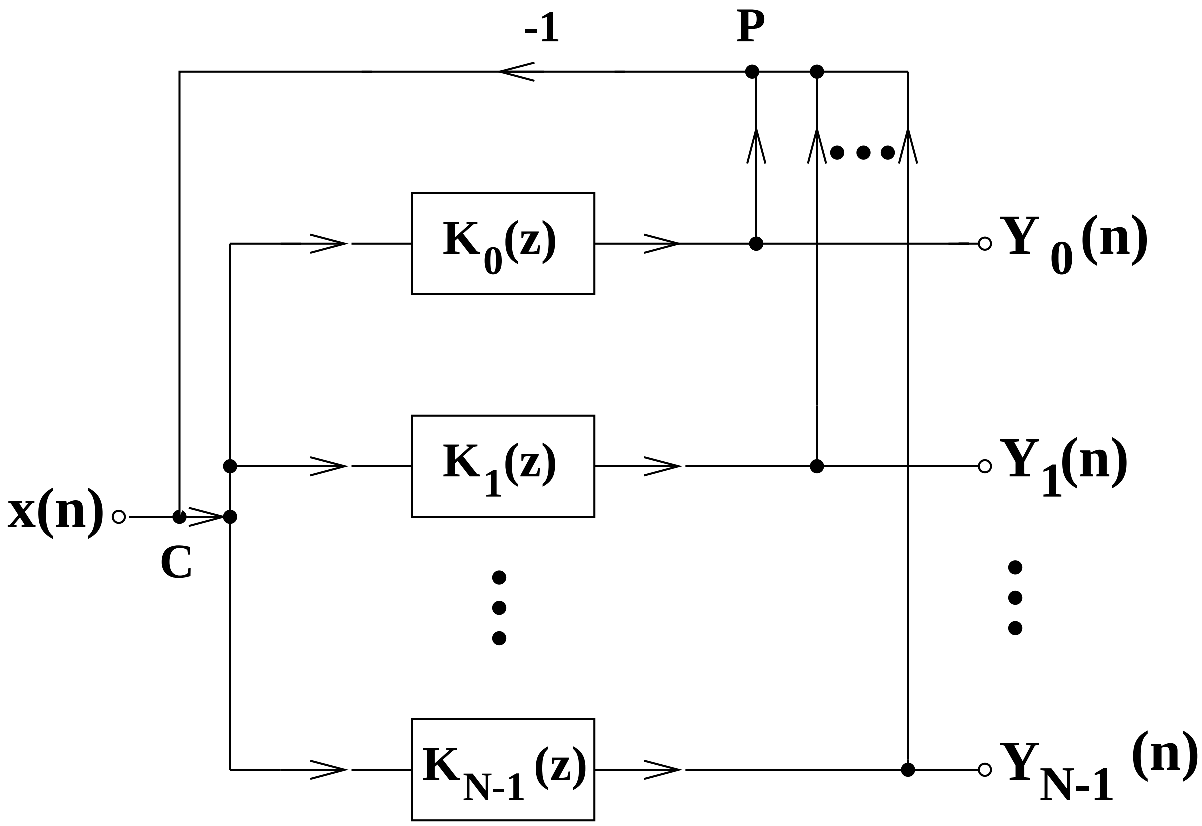
b,

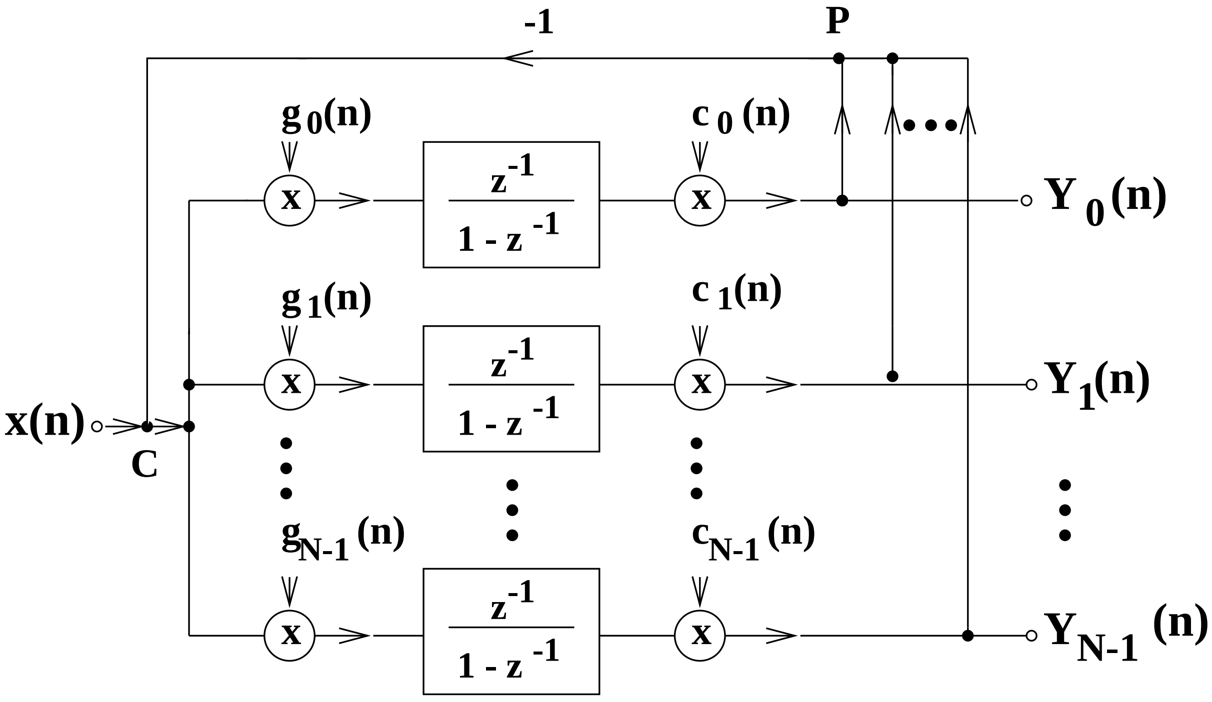

To the common feedback
2r(n)
x(n)
cosΦ0
- sin Φ0
y1(n)
sin Φ0
cosΦ0
y2(n)

c,


To the common feedback
r(n)
x(n)
cosΦ0
sinΦ0
-1
-1
y1(n)
y2(n)

d,


-1
P
$K_0(z)$
$Y_0(n)$
$K_1(z)$
$Y_1(n)$
x(n)
C
$K_{N-1}(z)$
$Y_{N-1}(n)$


-1
P
$g_0(n)$
$c_0(n)$
x
$\frac{z^{-1}}{1 - z^{-1}}$
x
$Y_0(n)$
$g_1(n)$
$c_1(n)$
x
$\frac{z^{-1}}{1 - z^{-1}}$
x
$Y_1(n)$
x(n)
C
$g_{N-1}(n)$
$c_{N-1}(n)$
x
$\frac{z^{-1}}{1 - z^{-1}}$
x
$Y_{N-1}(n)$

- linear combination of resonator transfer functions of

  $g_n z^{-1}/(1 - z_n z^{-1})$, $n = 0, 1, ..., N - 1$

- arbitrary transformations
  - series expansion coefficients and
  - signal components are calculated

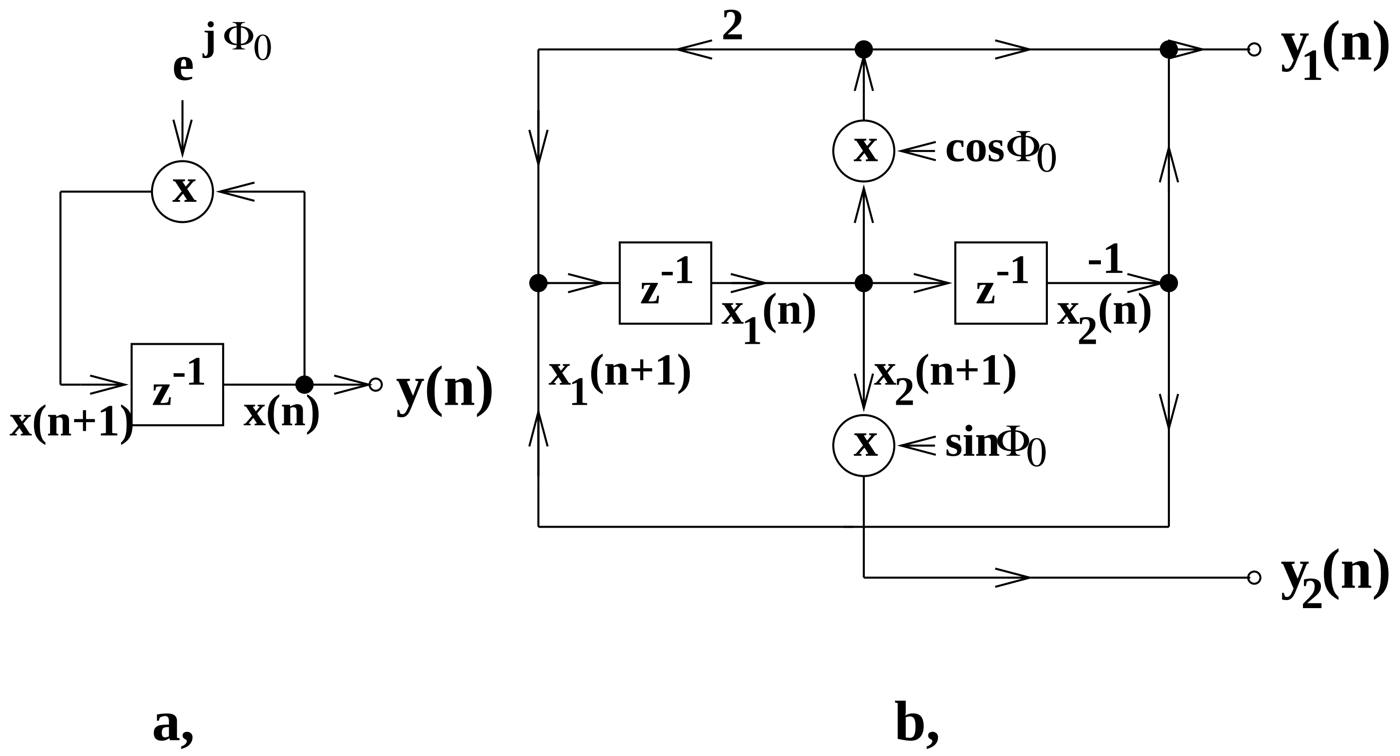
- transfer value from the input to point P

  $H_P(z) = z^{-N}$ (delay of a complete block)

- for periodic inputs with a period of N
  - difference at point C will be zero
  - resonators generate sinusoidal components without "disturbance"
- If there is a difference at point C, the error signal modifies the resonator states to reduce it
- very high loop gain

Inverse transforms

- V-S transformer structures
- start from the transform-domain coefficients
- parallel vectors
- produce a serial sequence
- periodic repetition results in a periodic waveform


e jΦ0
x
z-1
x(n+1)
x(n)
y(n)
2
y1(n)
x
cosΦ0
z-1
x1(n)
z-1
-1
x2(n)
x1(n+1)
x2(n+1)
x
sinΦ0
y2(n)
a,
b,

Sinusoidal waveform generation

- complex coefficient first-order system

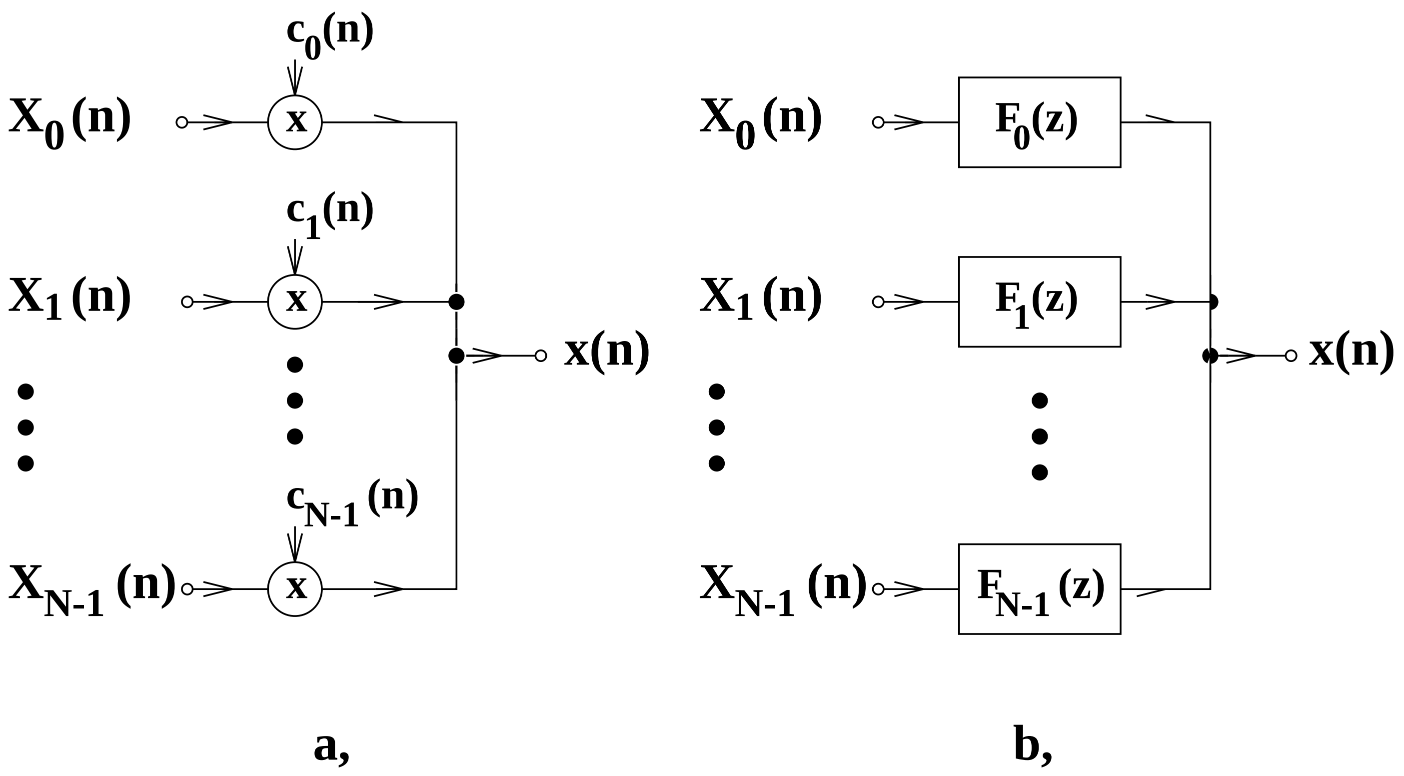
  $x(n+1) = e^{j\Phi_0}\, x(n)$
  $y(n) = x(n)$
  If $x(0) = 1$ then Re $y(n) = \cos(\Phi_0 n)$, Im $y(n) = \sin(\Phi_0 n)$, $n = 0, 1, \ \ldots$

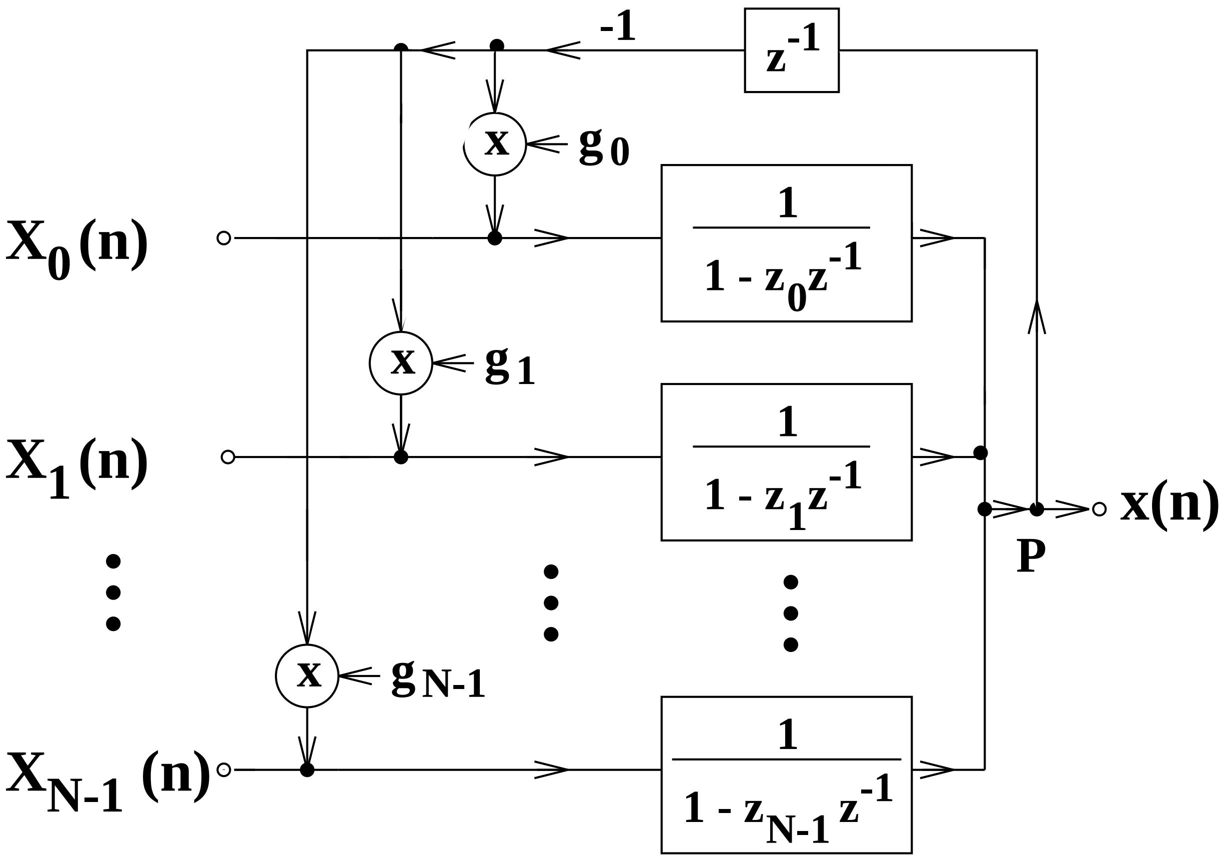
- real coefficient, second-order system

  $x_1(n+1) = 2\cos\Phi_0\, x_1(n) - x_2(n)$
  $x_2(n+1) = x_1(n)$
  $y_1(n) = \cos\Phi_0\, x_1(n) - x_2(n)$
  $y_2(n) = \sin\Phi_0\, x_1(n)$

  If $x_1(0) = 0$, and $x_2(0) = -1$ then
  $y_1(n) = \cos(\Phi_0\, n), \qquad y_2(n) = \sin(\Phi_0\, n)$

- the structures can be considered as
  - digital oscillators
  - two dimensional inverse transformers
- the initial conditions
  - correspond to the transform domain representation
  - specify the sinusoidal waveform


c0(n)
X0(n)
x
c1(n)
X1(n)
x
x(n)
cN-1(n)
XN-1(n)
x
a,
X0(n)
F0(z)
X1(n)
F1(z)
x(n)
XN-1(n)
FN-1(z)
b,


-1
z-1
x
g0
X0(n)
1
1 - z0z-1
x
g1
X1(n)
1
1 - z1z-1
x(n)
P
x
gN-1
XN-1 (n)
1
1 - zN-1 z-1

Alternative structures

- basis functions

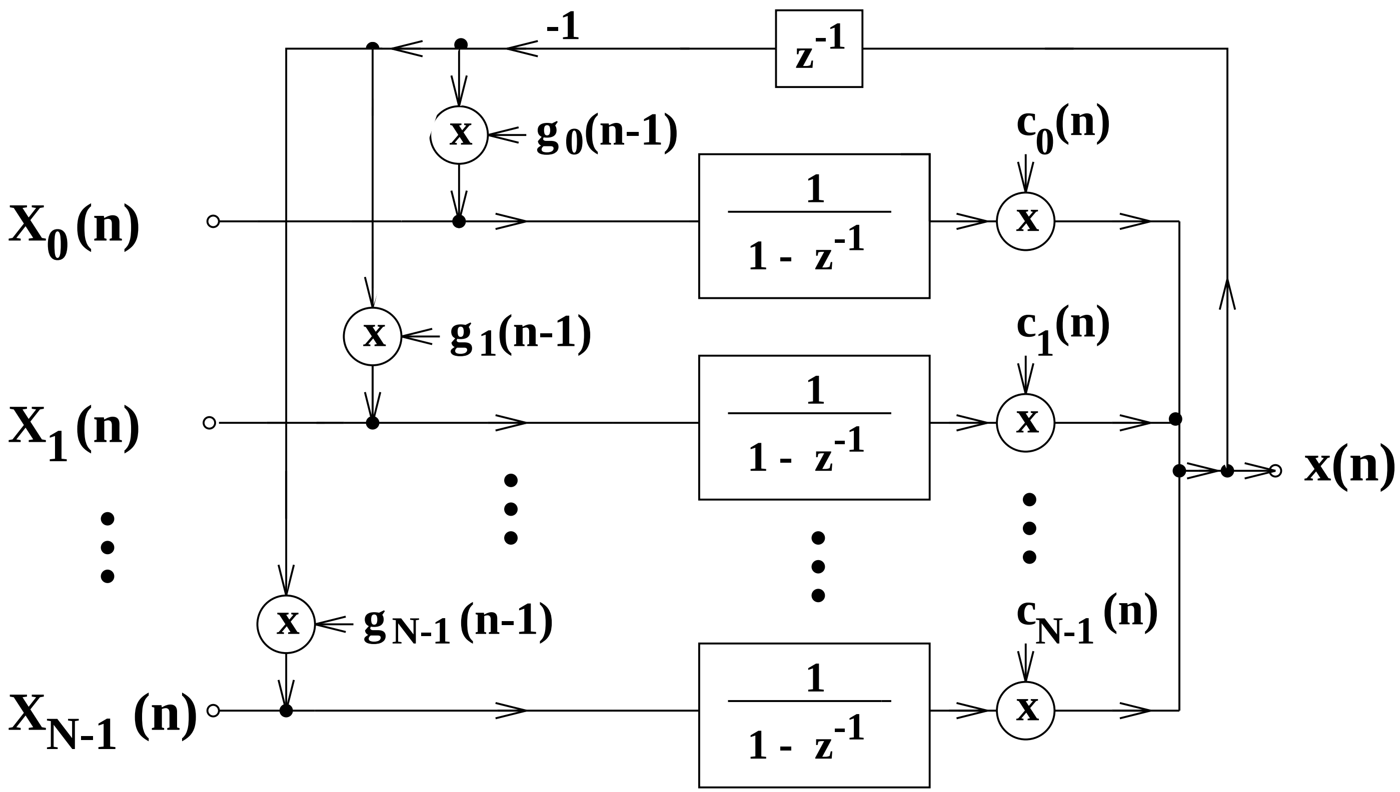
$$\underline{c}(n) = [c_0(n),\ c_1(n),\ \ldots,\ c_{N-1}(n)]^T,$$
$$\underline{X}(n) = [X_0(n),\ X_1(n),\ \ldots,\ X_{N-1}(n)]^T, \qquad n = 0, 1, \ldots$$
$$x(n) = \underline{c}^T(n)\underline{X}(n)$$

- base-generating filter banks

$$X_m(z) = X_m(0)\,\frac{1}{1-z^{-N}}\,, \qquad m = 0, 1, \ldots, N-1$$

$$F_m(z) = \frac{\frac{1}{1-z_m z^{-1}}}{1+\frac{1}{N}\sum_{n=0}^{N-1}\frac{z_n z^{-1}}{1-z_n z^{-1}}} = \frac{1-z^{-N}}{1-z_m z^{-1}} =$$
$$= 1 + z_m z^{-1} + \ \ldots\ + z_m^{N-1} z^{-(N-1)}$$

- basis function synthesizer
  - transposed version of the basic-reciprocal basis synthesizer

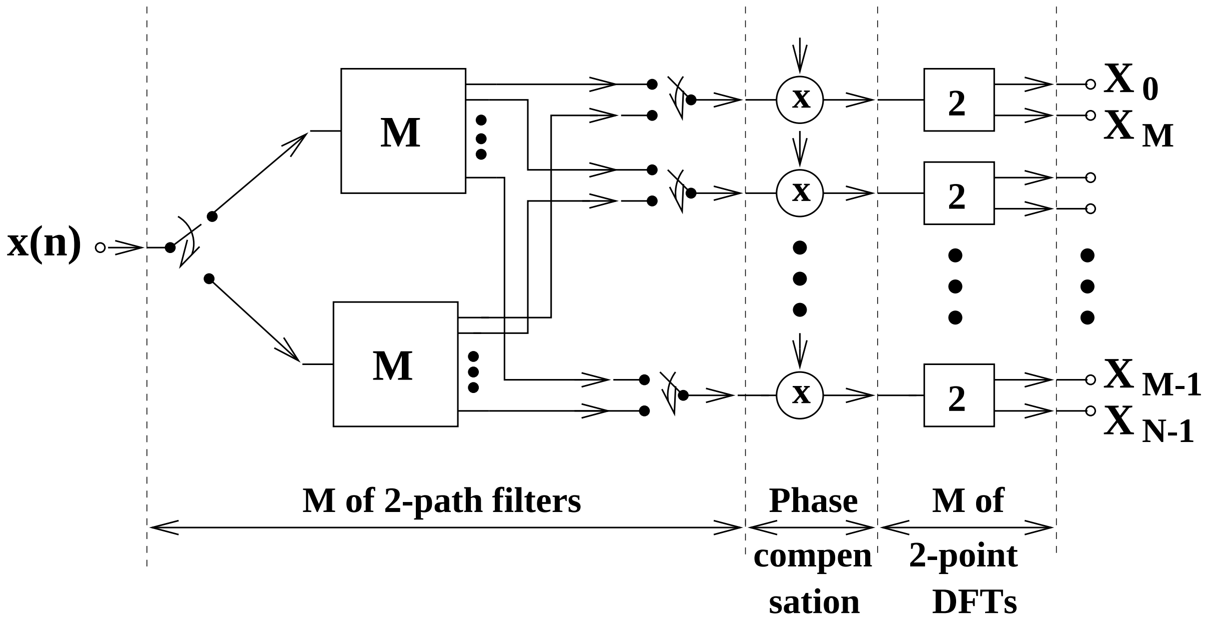

-1
z-1
g0(n-1)
c0(n)
X0(n)
1
1 - z-1
x
g1(n-1)
c1(n)
X1(n)
x(n)
gN-1(n-1)
cN-1(n)
XN-1(n)

Features of the time recursive analyzer-synthesizer pairs

- advantages
  - S-V, V-S transforms
  - high degree of parallelism
  - local interconnections
- limitations
  - additions with multiple inputs (linear combinations)
  - pipelining is limited in feedback systems

Fast algorithms for time-recursive transforms

- Combining
  - Block-oriented fast transforms
  - Polyphase decomposition
  - Composite filter-bank structure
  - Different output rates
- Modes of operation
  - Maximum decimated mode
  - Sliding window


x(n)
M
M
x
x
x
2
2
2
X 0
X M
X M-1
X N-1
M of 2-path filters
Phase
compen
sation
M of
2-point
DFTs

Polyphase decomposition of the DFT

- The definition of the DFT is

  $X_m = \frac{1}{N} \sum_{n=0}^{N-1} x(n) e^{-j\frac{2\pi}{N}mn}$

  where $m = 0, 1, \ldots, N-1$

- Let's suppose that $N = LM$, L and M are integers

- We decompose the sequence $x(n)$, $n = 0, 1, \ldots, N-1$, into L blocks each having M elements

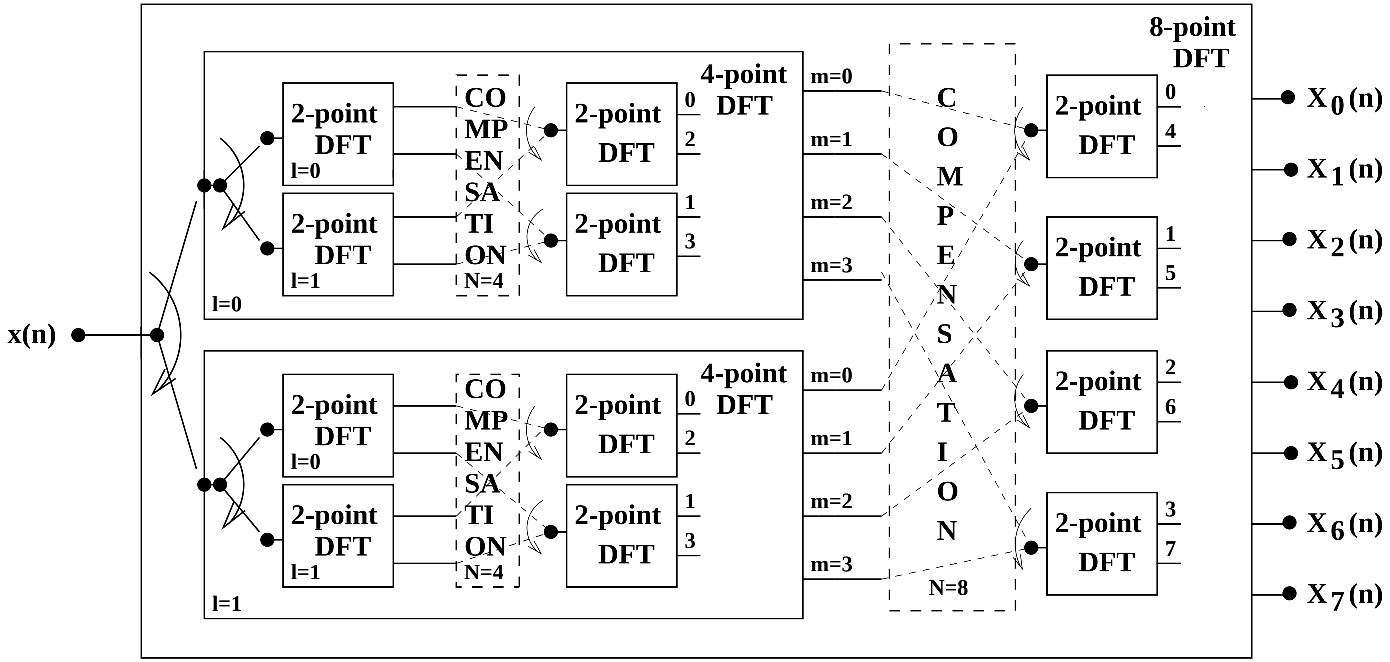
$$
\begin{aligned}
X_m &= \frac{1}{N}\sum_{k=0}^{M-1} x(kL)e^{-j\frac{2\pi}{N}mkL} + \frac{1}{N}\sum_{k=0}^{M-1} x(kL+1)e^{-j\frac{2\pi}{N}m(kL+1)} + \ \ldots \\
&+ \frac{1}{N}\sum_{k=0}^{M-1} x(kL+L-1)e^{-j\frac{2\pi}{N}m(kL+L-1)} = \\
&= \frac{1}{L}\sum_{l=0}^{L-1} \mathsf{DFT}_M(m,l)e^{-j\frac{2\pi}{N}ml} \qquad (1)
\end{aligned}
$$

- DFT(m,l) denotes the lth DFT value from the L M-point DFTs at the mth "frequency"

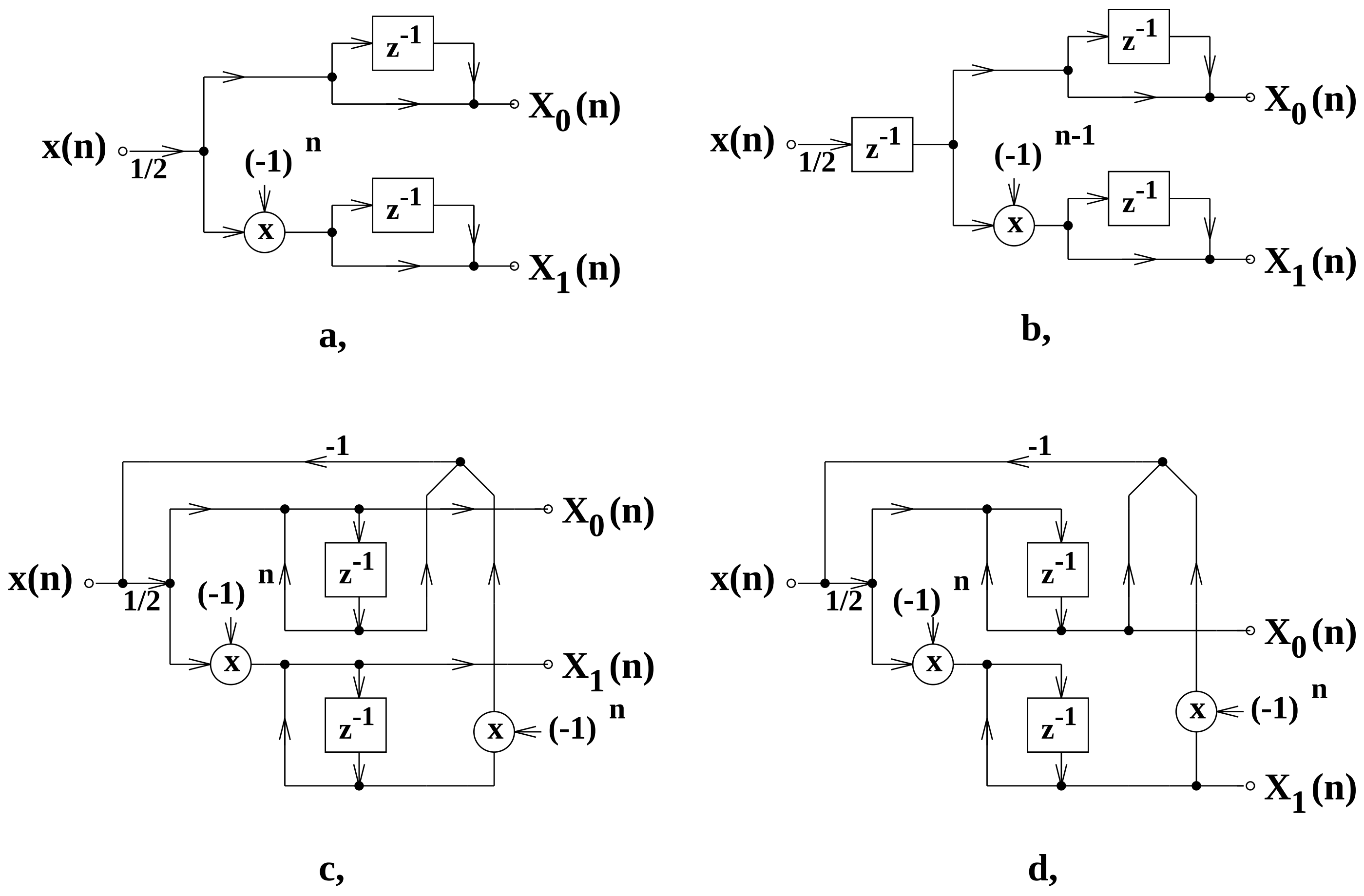
- For $m = 0, M, 2M,$ ... an L-point DFT is to be evaluated
- for other values of m the DFT(m, l) values should be modified by a correction factor

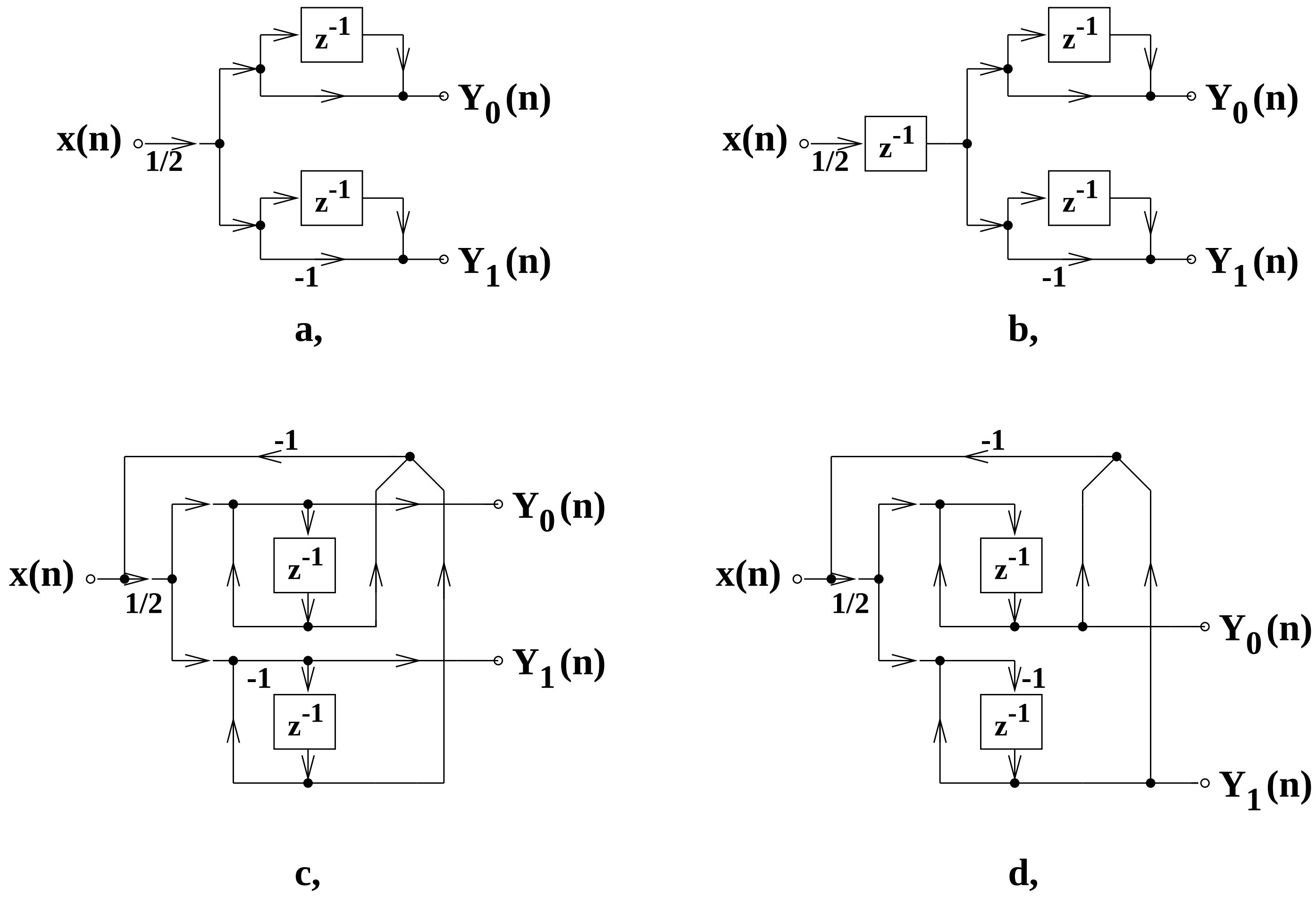
$$
\begin{aligned}
X_m &= \text{DFT}_L\left(\text{DFT}_M(m,l)\right)\ , \\
&\qquad\qquad \text{if}\ \ m = 0, M, ..., (L-1)M \\
&= \text{DFT}_L\left(\text{DFT}_M(m,l)\ e^{-j\frac{2\pi}{N}l}\right)\ , \\
&\qquad\qquad \text{if}\ \ m = 1, M+1, ..., (L-1)M+1 \\
&\ \vdots \qquad\qquad \vdots \\
&= \text{DFT}_L\left(\text{DFT}_M(m,l)\ e^{-j\frac{2\pi}{N}(M-1)l}\right)\ , \\
&\qquad\qquad \text{if}\ \ m = M-1, 2M-1, ..., LM-1\ ,
\end{aligned}
$$

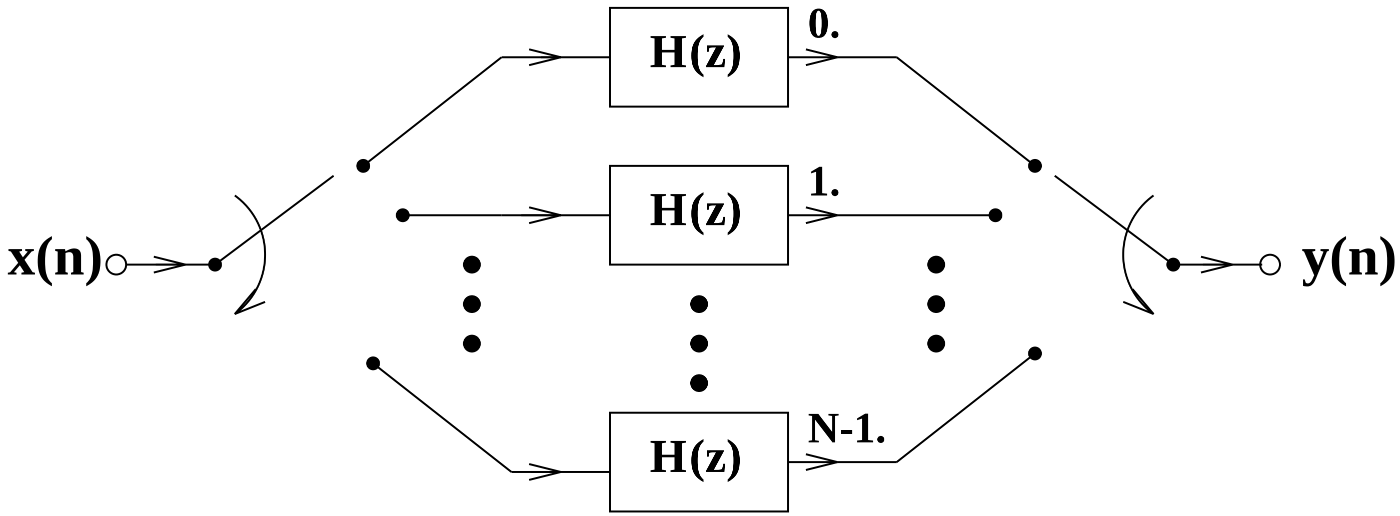
- If M and/or L are not relative primes further decompositions are possible



x(n)
l=0
l=1
2-point DFT
l=0
l=1
COMPENSATION
N=4
4-point DFT
0
2
1
3
m=0
m=1
m=2
m=3
COMPENSATION
N=8
8-point DFT
4
5
6
7
$X_0(n)$
$X_1(n)$
$X_2(n)$
$X_3(n)$
$X_4(n)$
$X_5(n)$
$X_6(n)$
$X_7(n)$


x(n)
1/2
$(-1)^n$
z^{-1}
z^{-1}
x
$X_0(n)$
$X_1(n)$
a,
x(n)
1/2
z^{-1}
$(-1)^{n-1}$
z^{-1}
z^{-1}
x
$X_0(n)$
$X_1(n)$
b,
-1
x(n)
1/2
$(-1)^n$
z^{-1}
z^{-1}
x
x
$(-1)^n$
$X_0(n)$
$X_1(n)$
c,
-1
x(n)
1/2
$(-1)^n$
z^{-1}
z^{-1}
x
x
$(-1)^n$
$X_0(n)$
$X_1(n)$
d,


x(n)
1/2
z^{-1}
-1
$Y_0(n)$
$Y_1(n)$
a,
b,
c,
d,


x(n)
y(n)
H(z)
H(z)
H(z)
0.
1.
N-1.

Polyphase filtering

- The overall transfer function is $H(z^{-N})$
- The transfer function of the mth 1-path, M-point DFT filter is

  $H_m(z) = \frac{1}{M} \left(1 - z^{-M}\right) \frac{1}{1 - z_m \, z^{-1}}$

- The transfer function of the mth 2-path, M-point DFT filter (without pipelining) is

  $H_m(z^2) = \frac{1}{M} \left(1 - z^{-2M}\right) \frac{1}{1 - z_m \, z^{-2}}$

  where $z_m = e^{j(2\pi/M)m}$, $m = 0, 1, ..., M - 1$

- To get the N-point DFT filter the equation above should be multiplied by

$$\frac{1}{2}\left(1 + z_m^{\frac{1}{2}}\, z^{-1}\right)$$

- The counterpart for $m = M, M+1, \ \dots, 2M-1 (= N-1)$ is

$$\frac{1}{2}\left(1 - z_m^{\frac{1}{2}}\, z^{-1}\right)$$

Generalized fast time-recursive evaluation

- Let's suppose to have two M dimensional orthogonal matrices $\underline{\underline{T}}_1$ and $\underline{\underline{T}}_2$
- We compose an orthogonal $2M$ dimensional $\underline{\underline{T}}$ matrix

$$\underline{\underline{T}} = \begin{bmatrix} \begin{bmatrix} a_{11}\,\underline{\underline{I}}_M \end{bmatrix} & \begin{bmatrix} a_{12}\,\underline{\underline{I}}_M \end{bmatrix} \\ \begin{bmatrix} a_{21}\,\underline{\underline{I}}_M \end{bmatrix} & \begin{bmatrix} a_{22}\,\underline{\underline{I}}_M \end{bmatrix} \end{bmatrix} \begin{bmatrix} \underline{\underline{T}}_1 & \underline{\underline{0}} \\ \underline{\underline{0}} & \underline{\underline{T}}_2 \end{bmatrix}$$

- $\underline{\underline{I}}_M$ is the M dimensional unit matrix /par /par /par /par

- $\underline{\underline{\mathrm{A}}}$ is orthogonal

$$\underline{\underline{A}} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

- The composition can be extended toward larger scale $\underline{\underline{A}}$ matrices

- The input sequence is decomposed into $L = 2$ groups each having a separate processing unit

- The decomposition is given by

  $\underline{\underline{T}}\,\underline{x} = [\underline{t}_0 \;\; \underline{0} \;\; \underline{t}_2 \;\; \underline{0} \;\; \ldots \;\; \underline{t}_{N-2} \;\; \underline{0}]\,\underline{x} + [\underline{0} \;\; \underline{t}_1 \;\; \underline{0} \;\; \underline{t}_3 \;\; \ldots \;\; \underline{0} \;\; \underline{t}_{N-1}]\,\underline{x}$

- $\underline{t}_m$ denotes the mth column of $\underline{\underline{T}}$, $m = 0, 1, \ldots, N-1$

- $\underline{x}$ is the vector of the last N input samples

  $\underline{x}^T = [x_{n-N+1},\; x_{n-N+2},\; \ldots,\; x_n]$

- We introduce
  - $\underline{\underline{T}}_1 = [\underline{t}_0, \underline{t}_2, \ \dots \ , \underline{t}_{N-2}]$
  - $\underline{\underline{T}}_2 = [\underline{t}_1, \underline{t}_3, \ \dots \ , \underline{t}_{N-1}]$
  - $\underline{x}_1^T = [x_{n-N+1}, \ x_{n-N+3}, \ \dots \ , \ x_{n-1}]$
  - $\underline{x}_2^T = [x_{n-N+2}, \ x_{n-N+4}, \ \dots \ , \ x_n]$

- We get

  $\underline{\underline{T}} \, \underline{x} = \underline{\underline{T}}_1 \, \underline{x}_1 + \underline{\underline{T}}_2 \, \underline{x}_2$

- $\underline{\underline{T}}_1$ and $\underline{\underline{T}}_2$ of dimension $N * M$ $(N = 2M)$

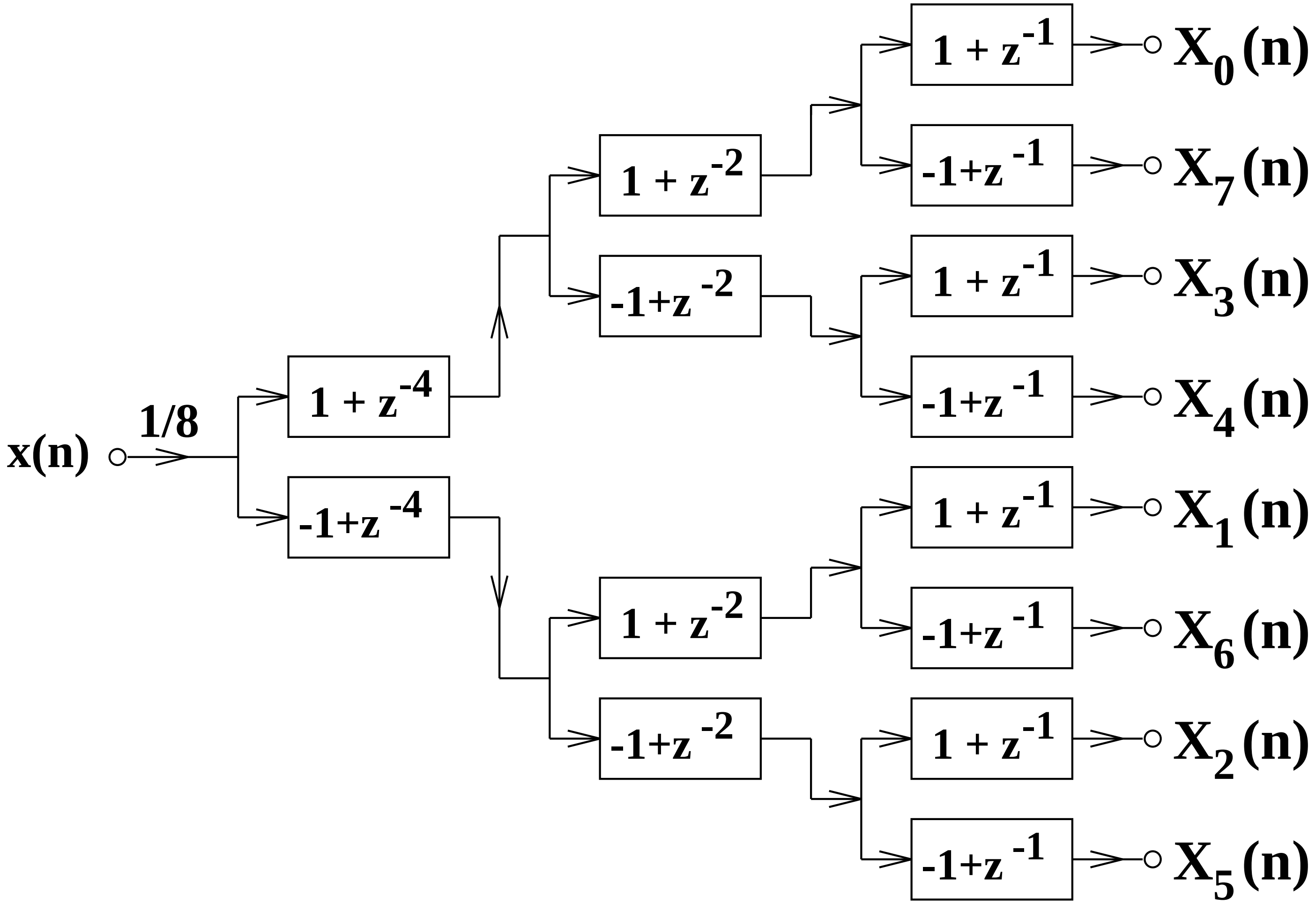
- $\underline{x}_1$ and $\underline{x}_2$ of dimension M

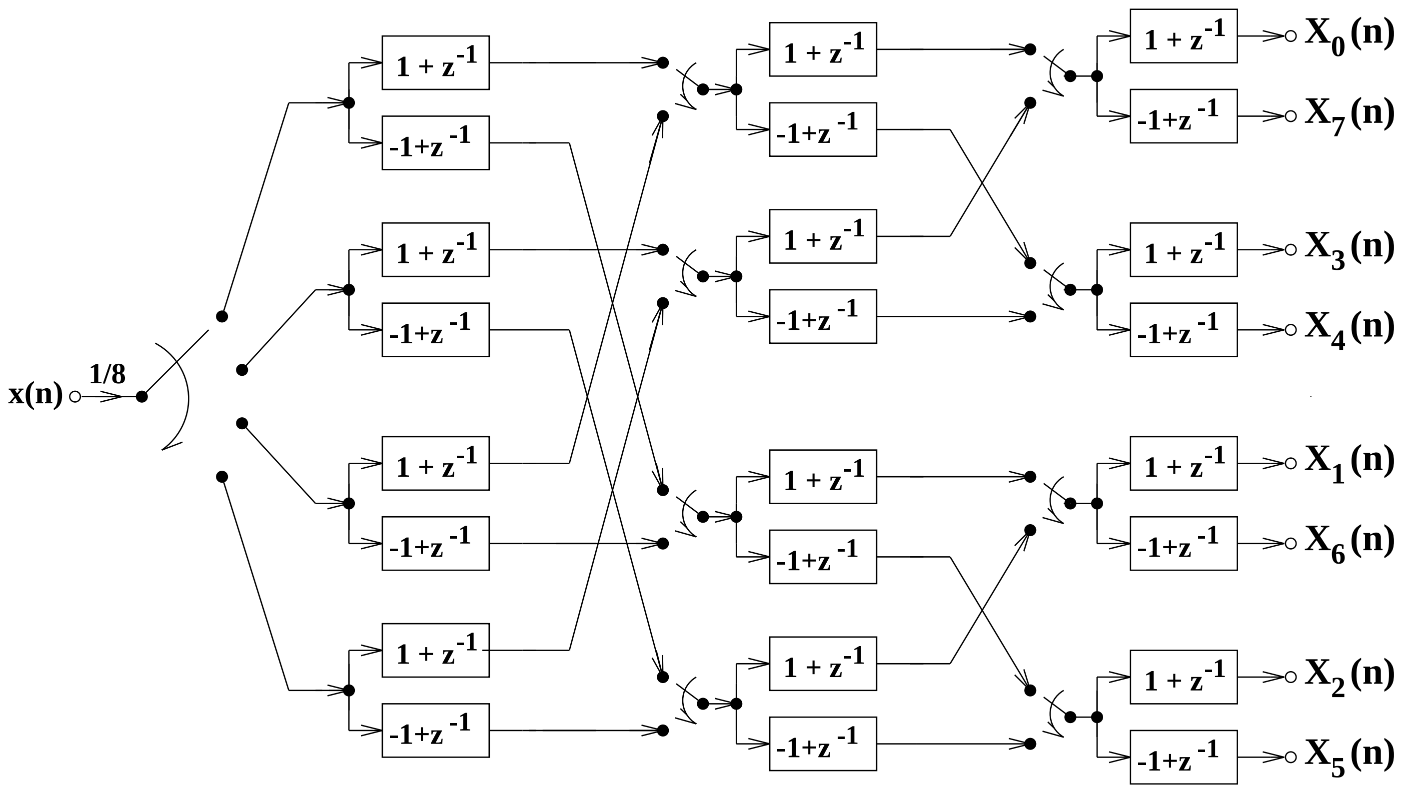
- The decomposition offers parallelization

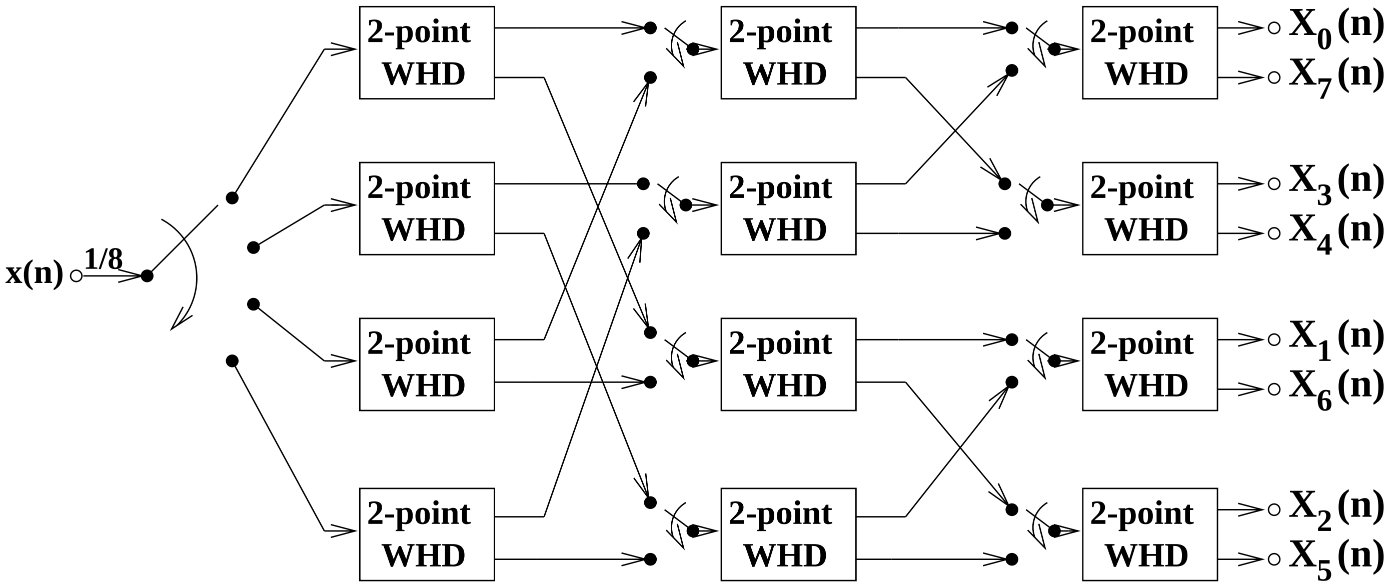
- Symmetry properties are needed to reduce the computational complexity
  - The rows of the $\underline{\underline{T}}_1$ and $\underline{\underline{T}}_2$ are to be considered only up to the index $M-1$
  - Multiplication with the rows of indices $M+m$, $m=0,1,\,\dots,M-1$ can be replaced with multiplications of rows of indices m ($m=0,1,\,\dots,M-1$)
  - The partial products are to be combined properly

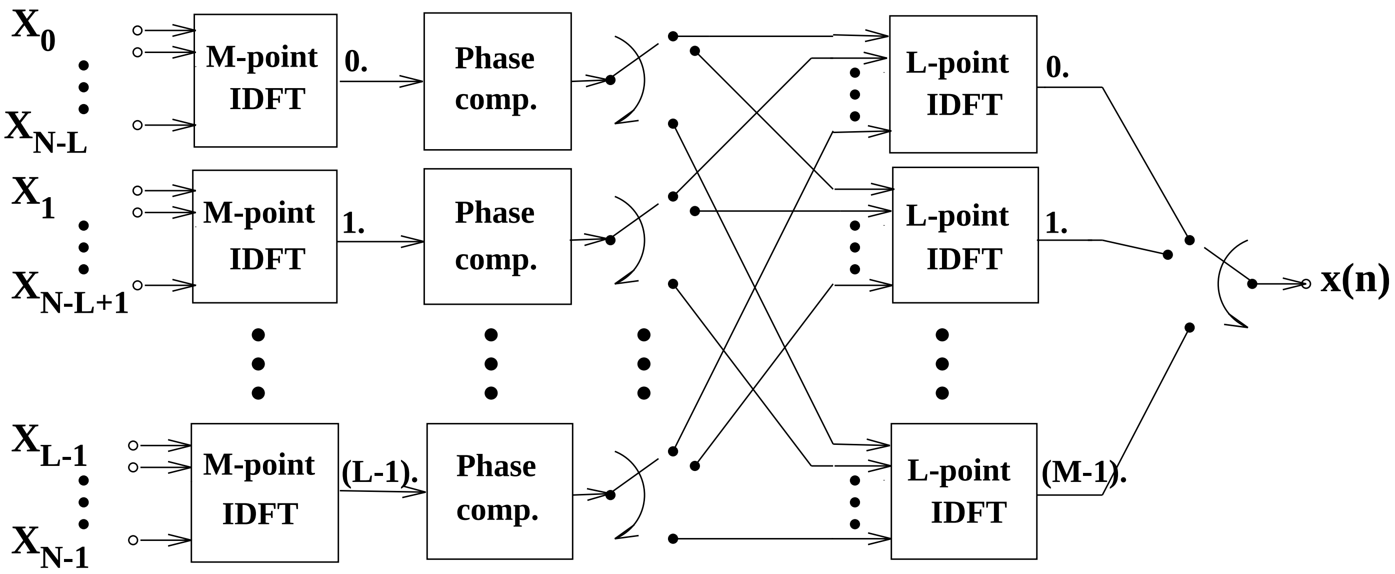
- In our case the result of the transformation can be expressed as the difference of the two terms for indices $m=M,M+1,\,\dots,N-1$

$$\underline{\underline{T}}\,\underline{x}=\underline{\underline{T}}_1\,\underline{x}_1-\underline{\underline{T}}_2\,\underline{x}_2$$

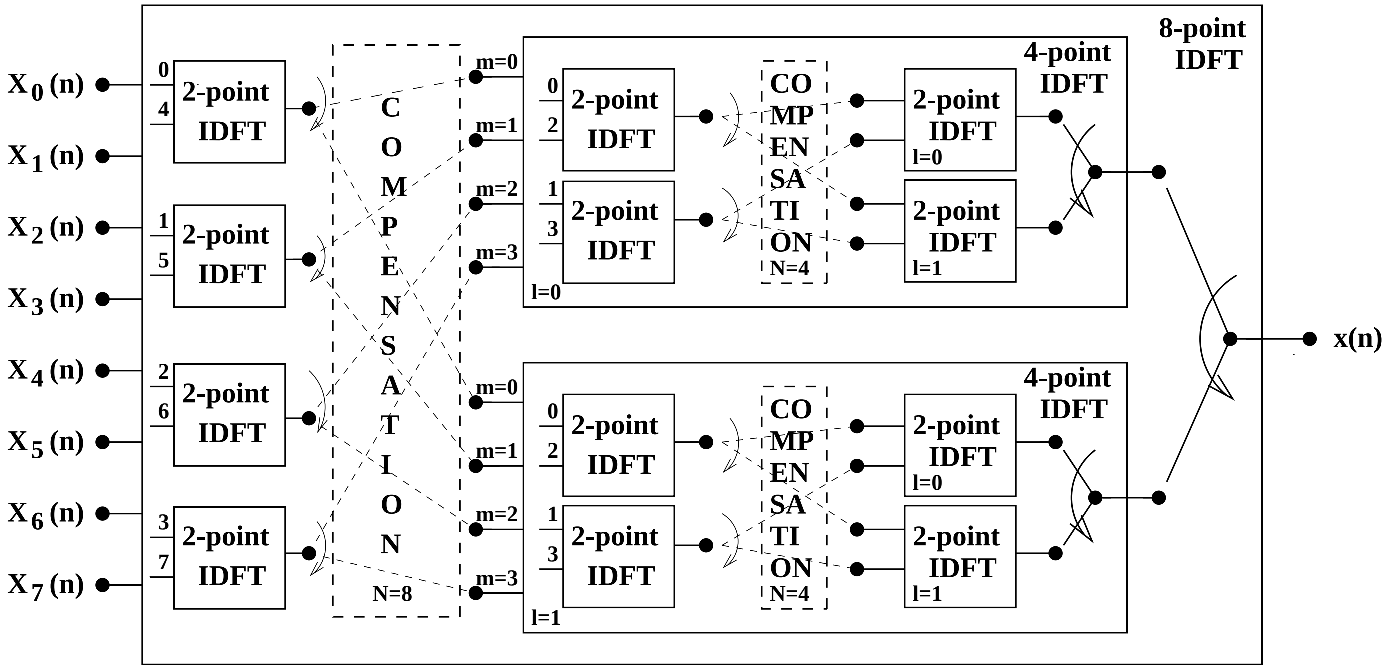

x(n)
1/8
1 + z^-4
-1+z^-4
1 + z^-2
-1+z^-2
1 + z^-2
-1+z^-2
1 + z^-1
-1+z^-1
1 + z^-1
-1+z^-1
1 + z^-1
-1+z^-1
1 + z^-1
-1+z^-1
X0(n)
X7(n)
X3(n)
X4(n)
X1(n)
X6(n)
X2(n)
X5(n)

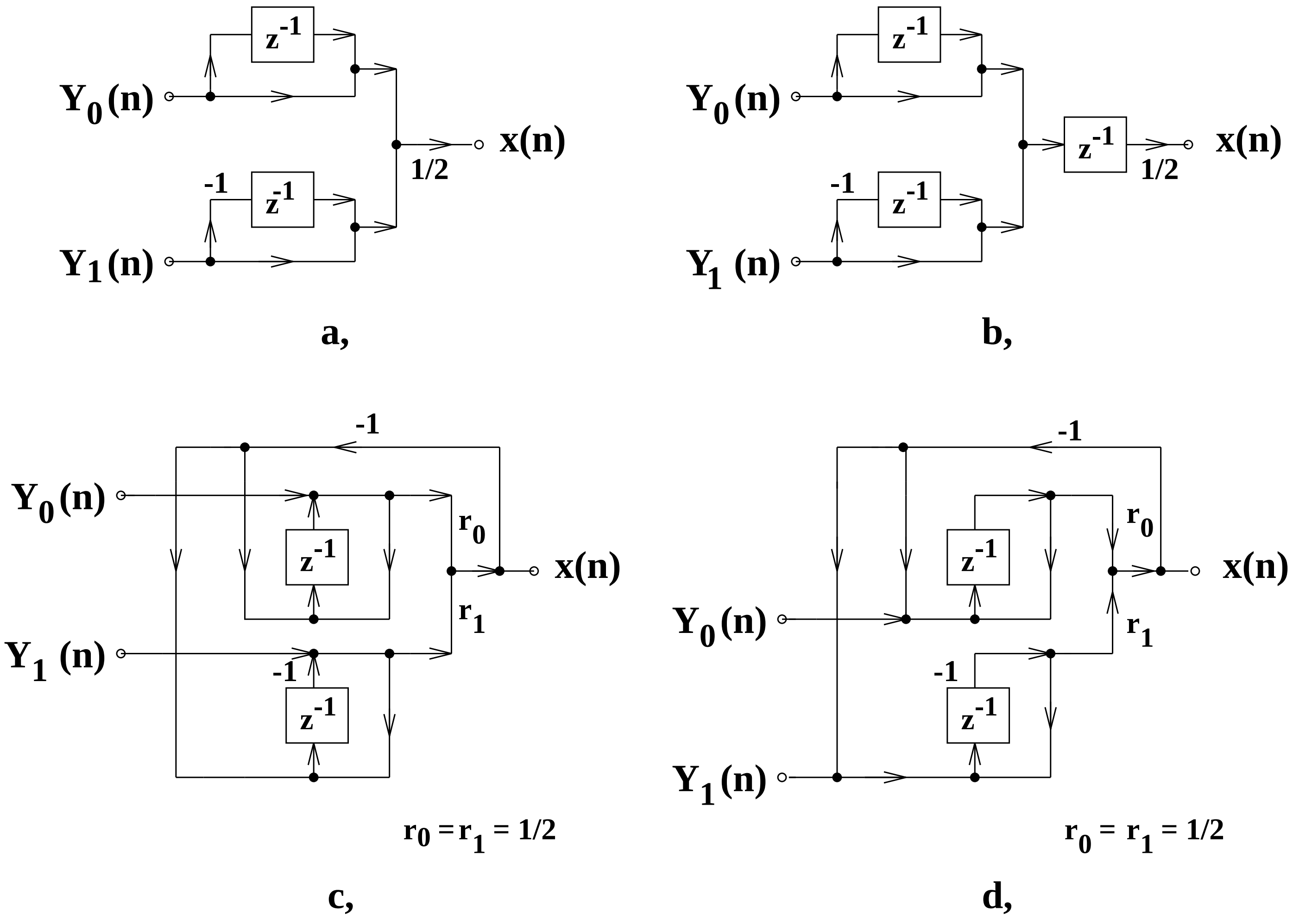

x(n)
1/8
1 + z-1
-1+z -1
1 + z-1
-1+z -1
1 + z-1
-1+z -1
1 + z-1
-1+z -1
1 + z-1
-1+z -1
1 + z-1
-1+z -1
1 + z-1
-1+z -1
1 + z-1
-1+z -1
1 + z-1
-1+z -1
1 + z-1
-1+z -1
1 + z-1
-1+z -1
1 + z-1
-1+z -1
X0(n)
X7(n)
X3(n)
X4(n)
X1(n)
X6(n)
X2(n)
X5(n)


x(n)
1/8
2-point WHD
2-point WHD
2-point WHD
2-point WHD
2-point WHD
2-point WHD
2-point WHD
2-point WHD
2-point WHD
2-point WHD
2-point WHD
2-point WHD
X0(n)
X7(n)
X3(n)
X4(n)
X1(n)
X6(n)
X2(n)
X5(n)

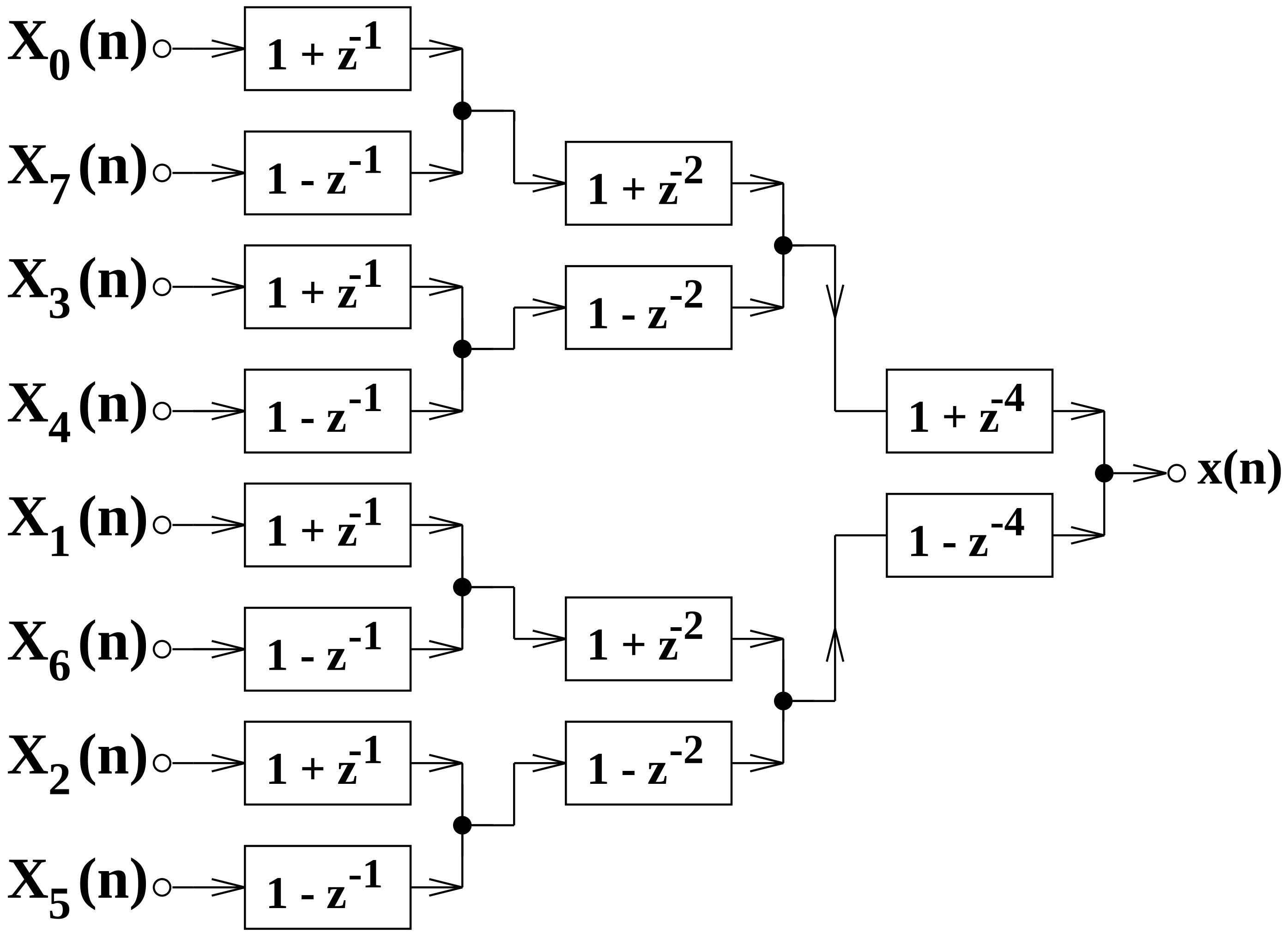


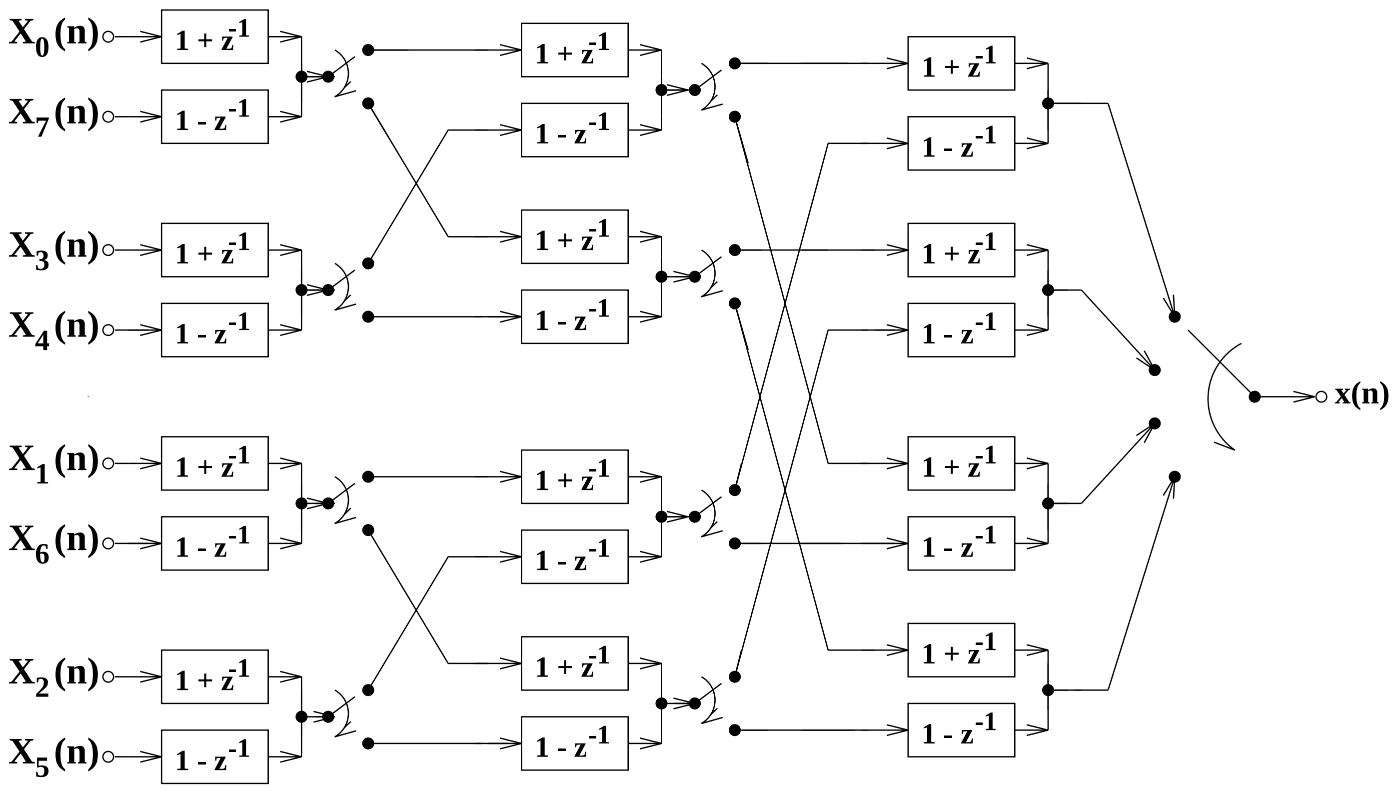
X_0
X_{N-L}
X_1
X_{N-L+1}
X_{L-1}
X_{N-1}
M-point IDFT
M-point IDFT
M-point IDFT
0.
1.
(L-1).
Phase comp.
Phase comp.
Phase comp.
L-point IDFT
L-point IDFT
L-point IDFT
0.
1.
(M-1).
x(n)


8-point
IDFT
4-point
IDFT
$X_0(n)$
$X_1(n)$
$X_2(n)$
$X_3(n)$
$X_4(n)$
$X_5(n)$
$X_6(n)$
$X_7(n)$
2-point
IDFT
C
O
M
P
E
N
S
A
T
I
O
N
N=8
m=0
m=1
m=2
m=3
l=0
l=1
CO
MP
EN
SA
TI
ON
N=4
x(n)

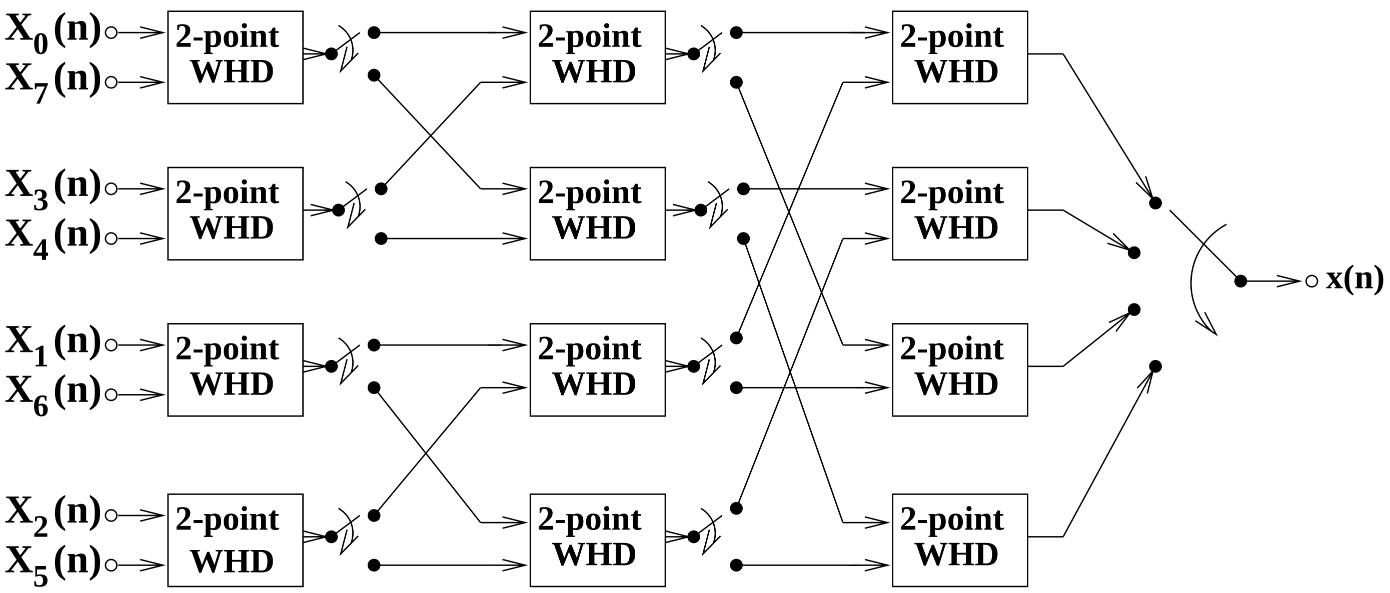


$Y_0(n)$
$Y_1(n)$
z^{-1}
-1
1/2
x(n)
a,
b,
c,
d,
r_0
r_1
$r_0 = r_1 = 1/2$


X0(n)
X7(n)
X3(n)
X4(n)
X1(n)
X6(n)
X2(n)
X5(n)
1 + z-1
1 - z-1
1 + z-1
1 - z-1
1 + z-1
1 - z-1
1 + z-1
1 - z-1
1 + z-2
1 - z-2
1 + z-2
1 - z-2
1 + z-4
1 - z-4
x(n)

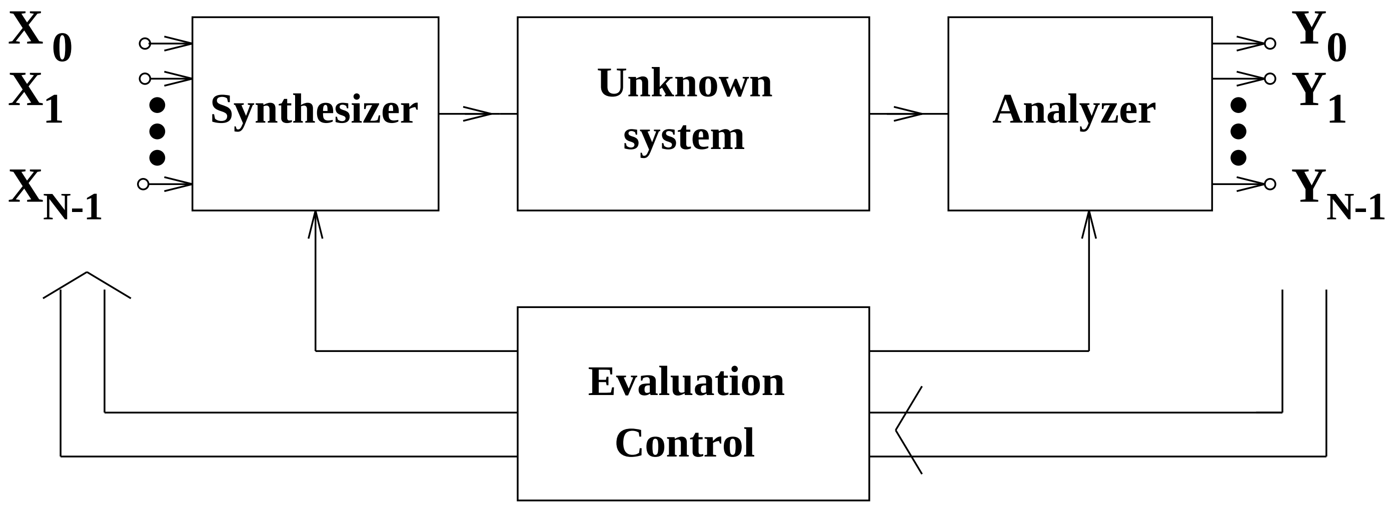


$X_0(n)$
$X_7(n)$
$X_3(n)$
$X_4(n)$
$X_1(n)$
$X_6(n)$
$X_2(n)$
$X_5(n)$
$1 + z^{-1}$
$1 - z^{-1}$
$x(n)$


$X_0(n)$
$X_7(n)$
$X_3(n)$
$X_4(n)$
$X_1(n)$
$X_6(n)$
$X_2(n)$
$X_5(n)$
2-point WHD
x(n)

Applications of the time-recursive transformers

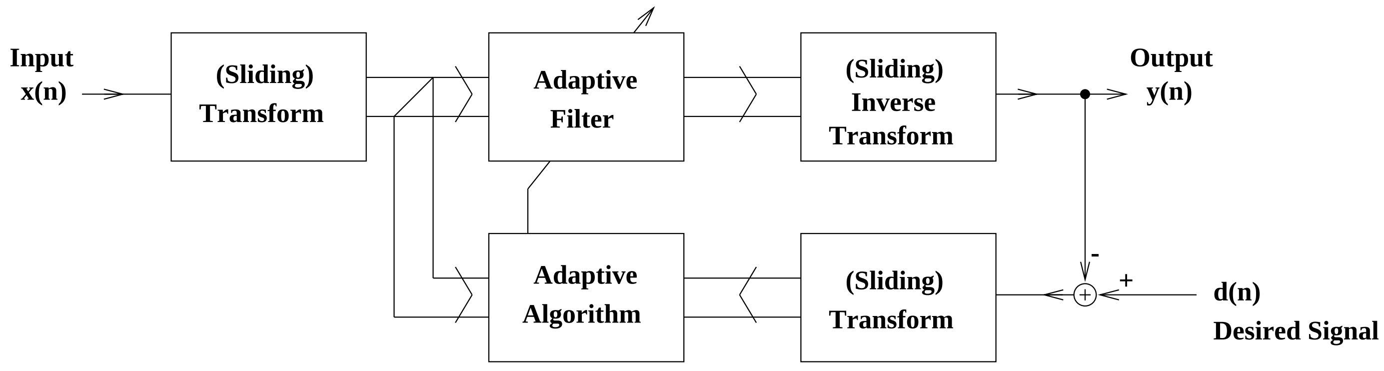
- Transform domain signal processing
  - Paralleization of the adaptation procedure
  - Better distribution of the computational load along time
- Multi-sine measuring
  - Synchronized signal generation and analyzis


X_0
X_1
X_{N-1}
Synthesizer
Unknown system
Analyzer
Y_0
Y_1
Y_{N-1}
Evaluation
Control

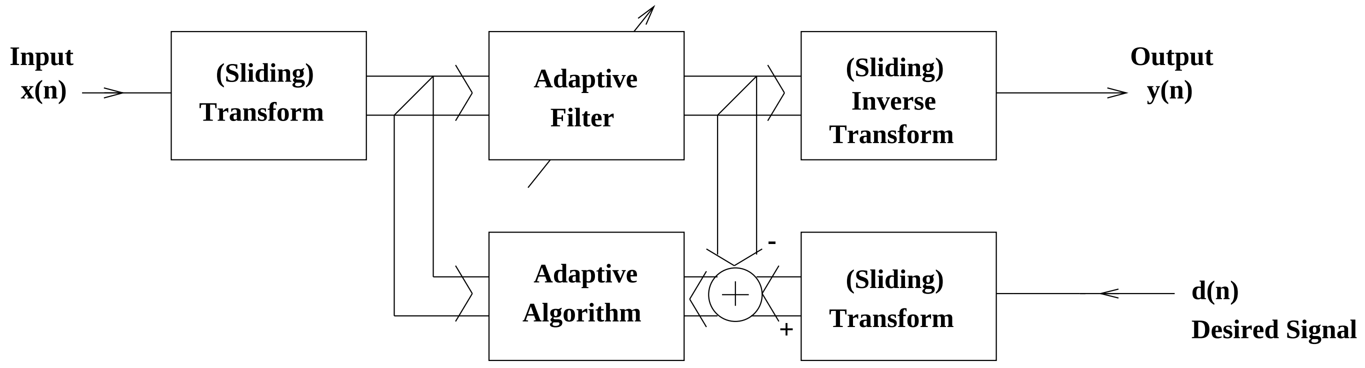
Fast sliding transforms in transform-domain adaptive filtering

- The basic idea is
  - Applying FFT for signal segments
  - Performing the adaptation in the frequency domain
  - The adaptation is controlled by the FFT of an error sequence

- Methods
  - Block-oriented
  - Multi-rate filter banks


Input
x(n)
(Sliding)
Transform
Adaptive
Filter
(Sliding)
Inverse
Transform
Output
y(n)
Adaptive
Algorithm
(Sliding)
Transform
-
+
d(n)
Desired Signal

(a)


Input
x(n)
(Sliding)
Transform
Adaptive
Filter
(Sliding)
Inverse
Transform
Output
y(n)
Adaptive
Algorithm
-
+
(Sliding)
Transform
d(n)
Desired Signal

(b)

Standard (not sliding) transformers

- Serial to parallel, parallel to serial conversions
- Transformed input and error
- The adaptation algortihm runs only once for each block of input
- The sampling rate is compromised by the block length

Sliding transformers

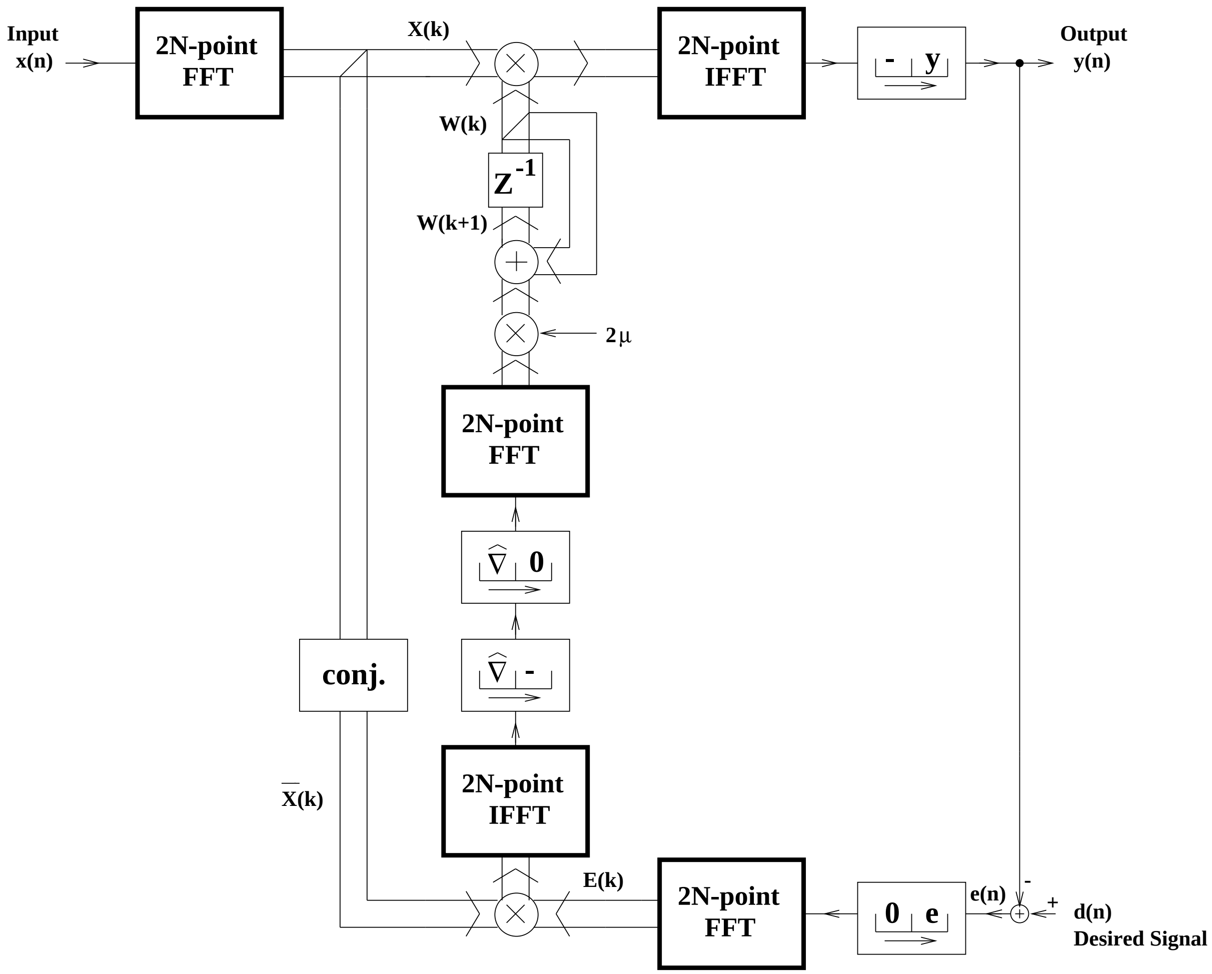
- No extra savings in computations
- Distribution of computational load in time
- Real time applications

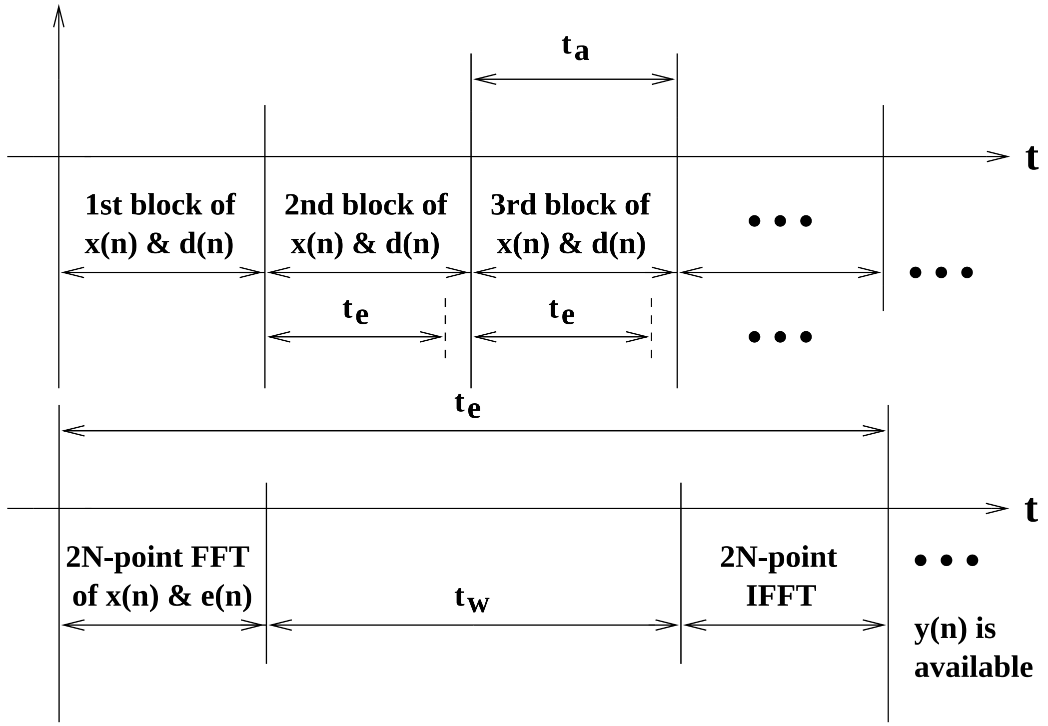
Using band-path filters

- Producing signal componeents instead of Fourier coefficients
- The center frequencies correspond to the Nth root of unity values
- Interpretation
  - The input is deccomposed to components by a filter-bank
  - The components are processed further
  - The output is produced by an N input - single output synthesizer bank

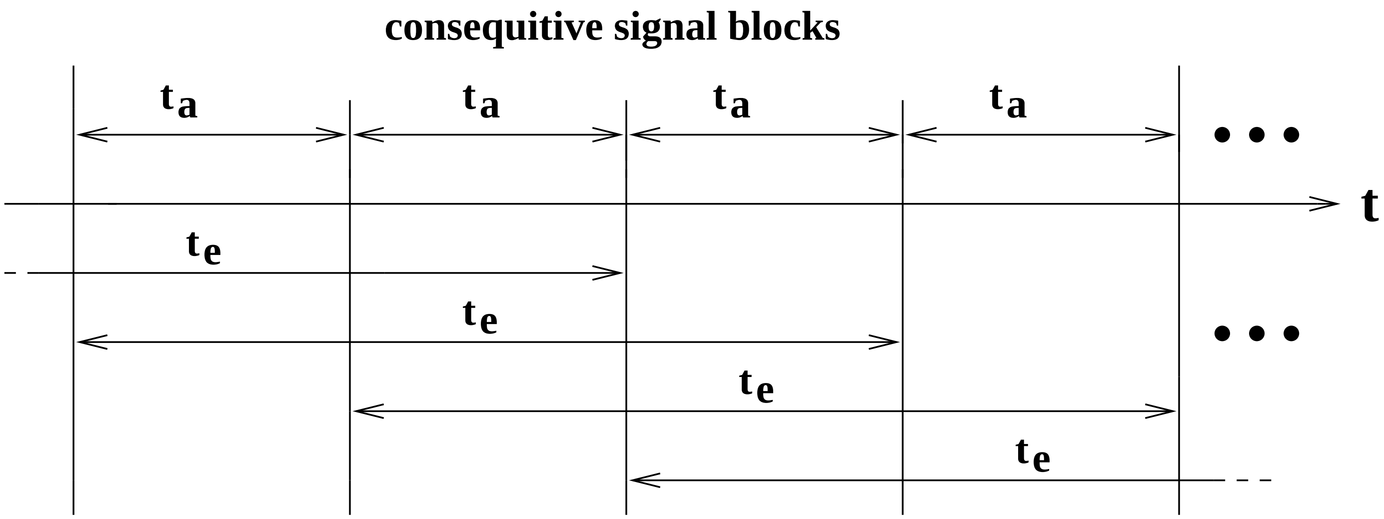
Block adaptive filter approaches

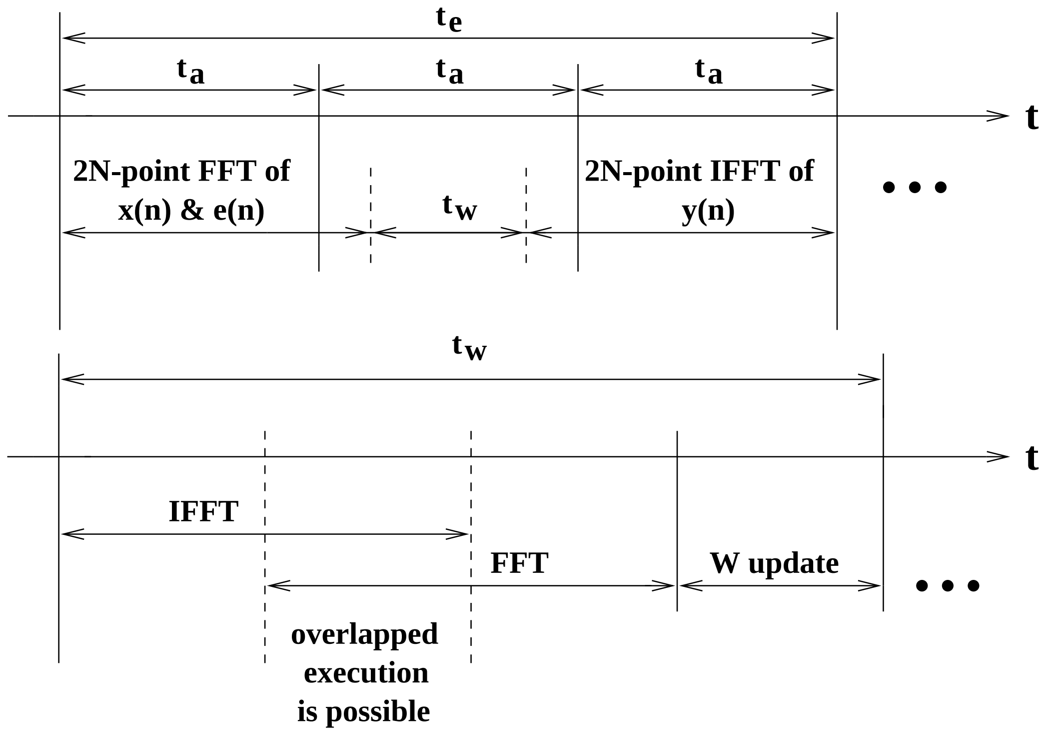
- Overlap-Save Method
- Overlapp-Add Method
- Two subsequent blocks are concatenated
- Double sized transformations are performed

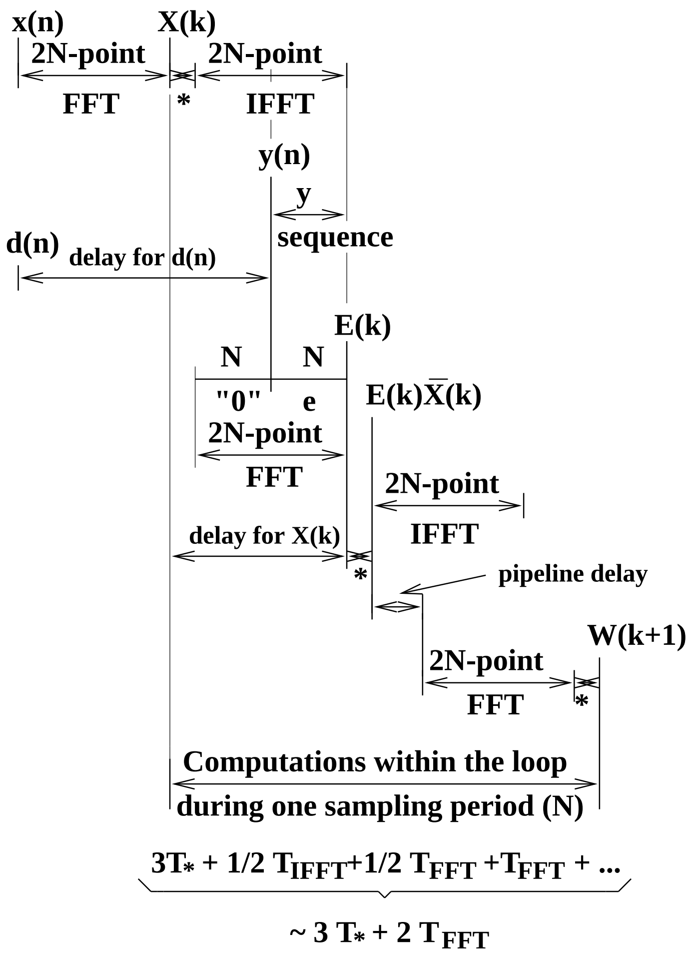

Input
x(n)
2N-point
FFT
X(k)
2N-point
IFFT
- y
Output
y(n)
W(k)
Z-1
W(k+1)
2µ
2N-point
FFT
∇̂ 0
conj.
∇̂ -
X̄(k)
2N-point
IFFT
E(k)
2N-point
FFT
0 e
e(n)
-
+
d(n)
Desired Signal



t_a
t
1st block of x(n) & d(n)
2nd block of x(n) & d(n)
3rd block of x(n) & d(n)
• • •
• • •
t_e
t_e
• • •
t_e
t
2N-point FFT of x(n) & e(n)
t_w
2N-point IFFT
• • •
y(n) is available


consequitive signal blocks
t_a
t_a
t_a
t_a
t
t_e
t_e
t_e
t_e


t_e
t_a
t_a
t_a
t
2N-point FFT of x(n) & e(n)
t_w
2N-point IFFT of y(n)
• • •
t_w
t
IFFT
FFT
W update
• • •
overlapped execution is possible


x(n)
X(k)
2N-point
FFT
*
2N-point
IFFT
y(n)
y
sequence
d(n)
delay for d(n)
E(k)
N
N
"0"
e
2N-point
FFT
E(k)X̄(k)
2N-point
IFFT
delay for X(k)
*
pipeline delay
W(k+1)
2N-point
FFT
*
Computations within the loop
during one sampling period (N)
3T* + 1/2 TIFFT+1/2 TFFT +TFFT + ...
~ 3 T* + 2 TFFT

Savings by sliding transforms

- The required complex multiplications for the maximally decimated sliding N-point FFT

  $C1 = \frac{N}{4} \log_2 \left(\frac{N}{8}\right) + 1$

- The number of complex multiplications which can not be executed before the arrival of the last sample of the block

  $C2 = \frac{N}{2} - \log_2 N$

- For $N = 1024$
  - $C1 = 1793$
  - $C2 = 502$

Implementation of recursive discrete transforms

- Similar HW and SW platform to the realization of FIR/IIR filters
- The simplest solution s to have a separate FIR filter for every channel
- Embedded systems: ASICs, DSPs, FPGAs
- FPGA
  - Bit-serial aritmetics for FPGA based multiplier
  - Complete building blocks like resonators
  - XILINX XC4000 FPGA elements