ELEC 484 Assignment 5

1. Convolve two sequences (vectors) of 32 numbers in several ways as described below, both in time and frequency domain, and check to see if both give the same result. Repeat by convolving two sound files 3-5 seconds long each (frequency domain only, no need to compare with time domain). For the first sound file, use the flute signal, for the second sound file, use a drum signal.
    a. Circular (cyclic, periodic) convolution in frequency domain (entire file length). Result will be same length as input.
    b. Acyclic (aperiodic) convolution in frequency domain using zero padding (entire file length) (result will be twice the original file length)
    c. Circular (cyclic, periodic) convolution on windowed overlapping segments, raised cosine windows, cyclic shift, and overlap-add (DAFX figure 8.5 where the time-frequency processing is a multiplication with the time/frequency coefficients of a second signal, keep the phases of only one of the signals). See also figure 8.28. Use windows 8 samples long for numbers, 2048 samples long for audio file
    d. Acyclic (aperiodic) convolution in frequency domain using zero padding using rectangular windows (DAFX figure 8.20) and overlap-add. 8 samples long for numbers, 2048 samples long for audio file.

Explain which method(s) is/are correct, and what is wrong with the other methods. Provide the Matlab code, numerical results for short vectors, and copies of the input and output wav files in each case.

2. Take the FFT of a cosine wave which has an integer number of samples per cycle (e.g 8) and plot the result (both amplitude and phase). For example with 8 KHz sampling rate, use 1 KHz cosine wave.

    - Repeat for a different FFT block size, e.g. 256.
    - Repeat for a cosine wave with a non-integer number of samples per cycle.
    - Repeat again after windowing the cosine wave with a raised cosine window (256 sample window).
    - Repeat again after shifting the cosine wave by 90 degrees (e.g. use sine instead).
    - Explain all results.

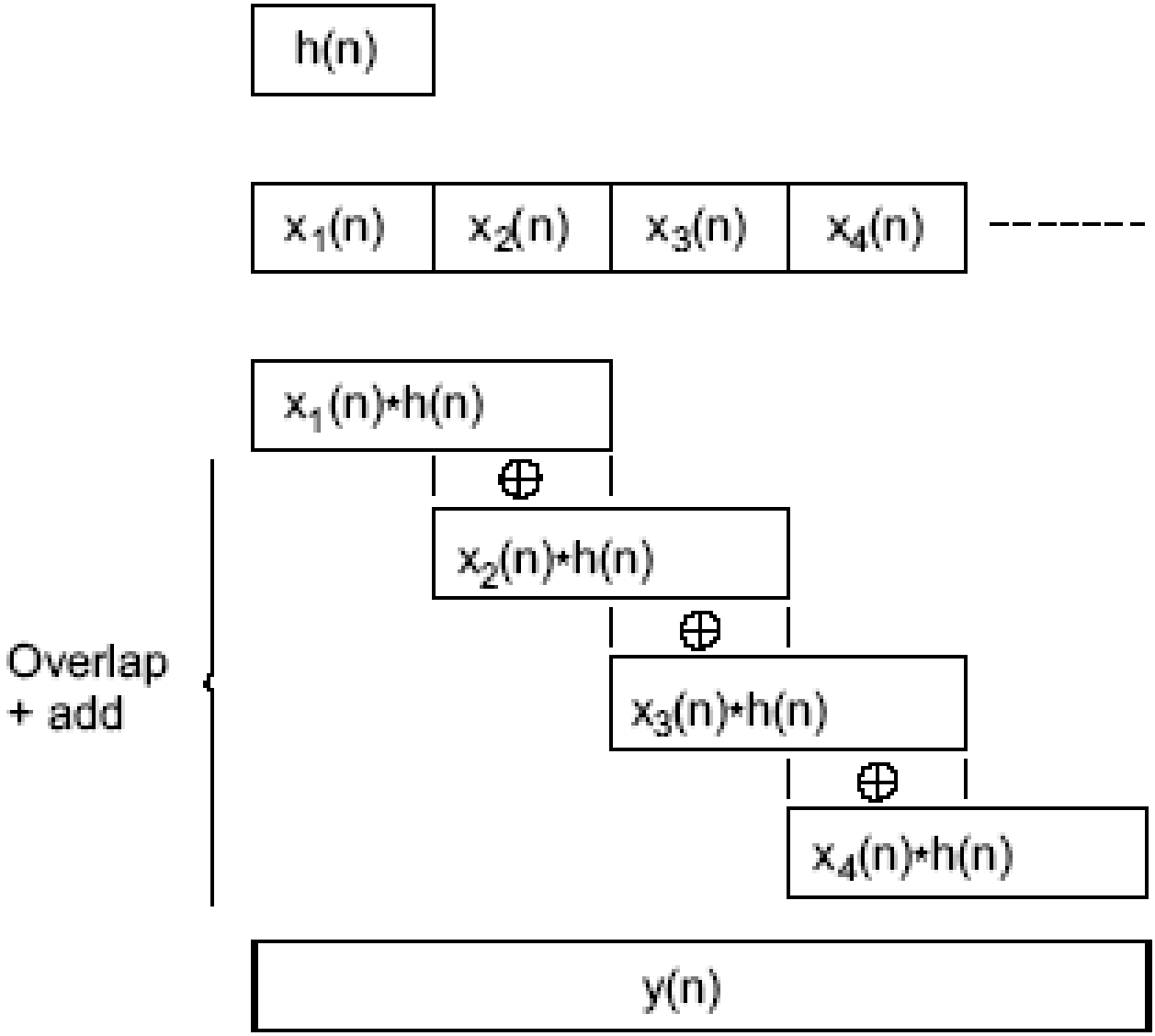

Figure 8.20 FFT filtering.

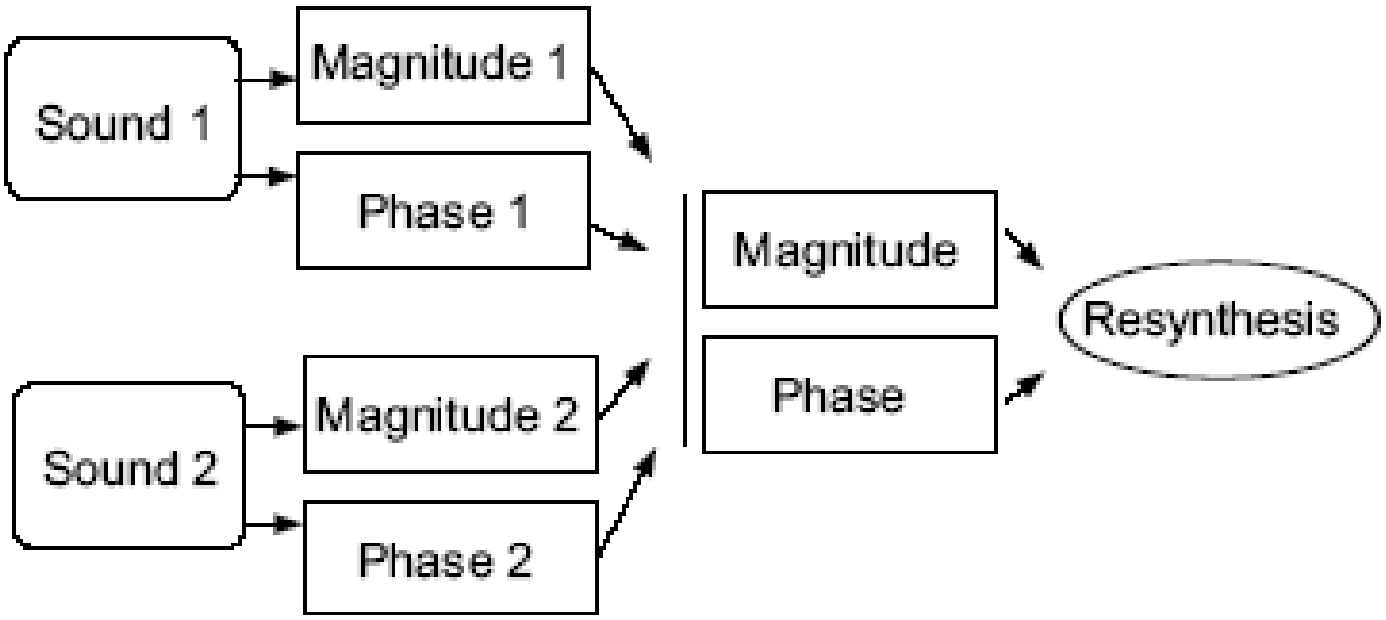

Figure 8.28 Basic principle of spectral mutations.

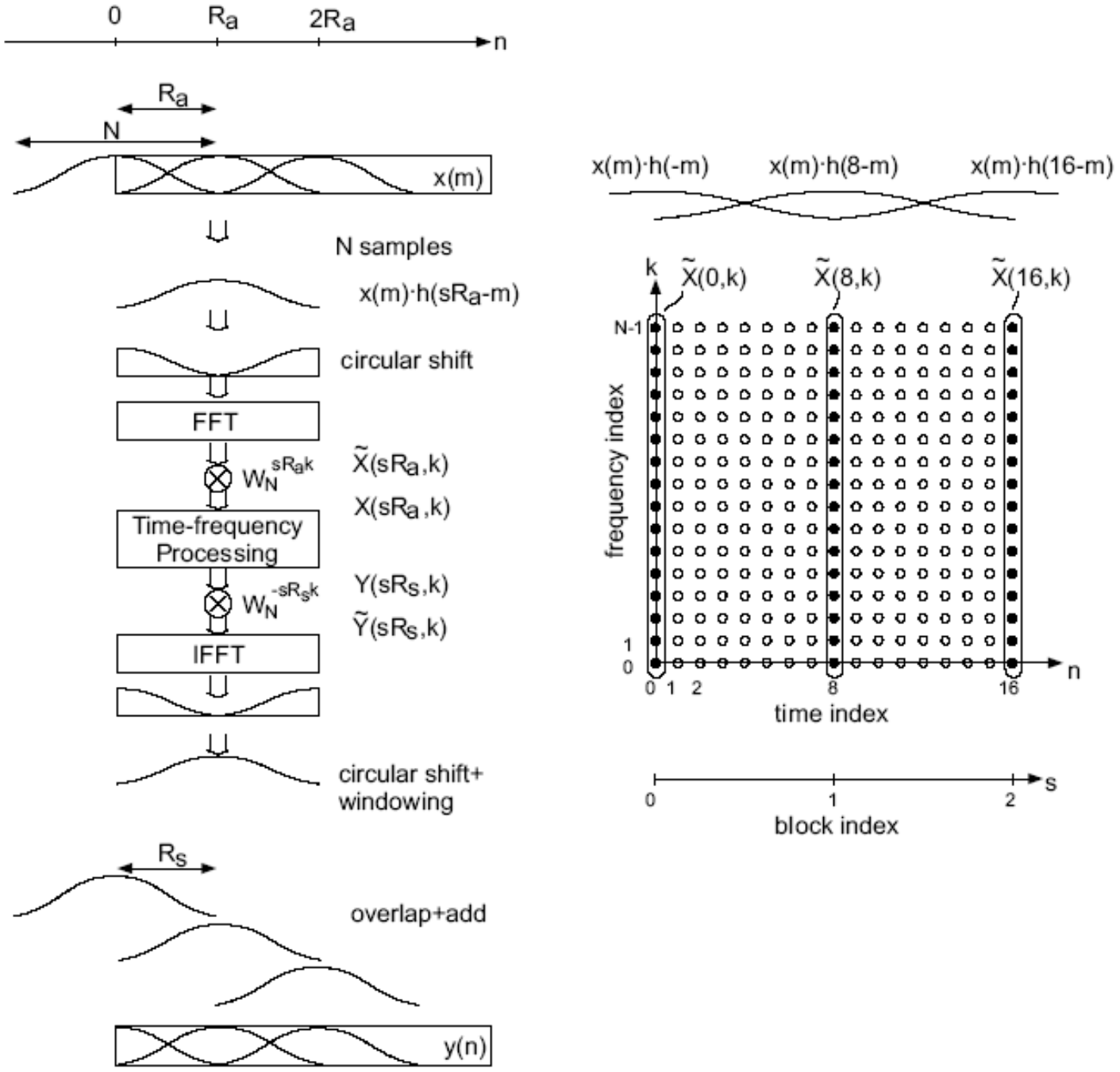
0
Ra
2Ra
n
Ra
N
x(m)
N samples
x(m)·h(sRa-m)
circular shift
FFT
WN sRak
X̃(sRa,k)
X(sRa,k)
Time-frequency Processing
WN -sRsk
Y(sRs,k)
Ỹ(sRs,k)
IFFT
circular shift+ windowing
Rs
overlap+add
y(n)
x(m)·h(-m)
x(m)·h(8-m)
x(m)·h(16-m)
k
X̃(0,k)
X̃(8,k)
X̃(16,k)
N-1
frequency index
1
0
0 1 2
8
16
n
time index
0
1
2
s
block index

Figure 8.5