Efficient Convolution with Overlap-Add

Lab 7, DSP

Objective

Students should know the Overlap-Add algorithm and be able to implement it in Matlab.

Theoretical aspects

The Overlap-Add algorithm is a method of computing the convolution of a very long signal with a smaller-length second signal, that works by splitting the first signal into smaller parts and performing separate convolutions of each part with the second signal.

The convolutions themselves can be computed in various ways, either in frequency or in time domain. Here, we will implement it using the convolution equation in time domain.

The algorithm is summarized in the following figure.

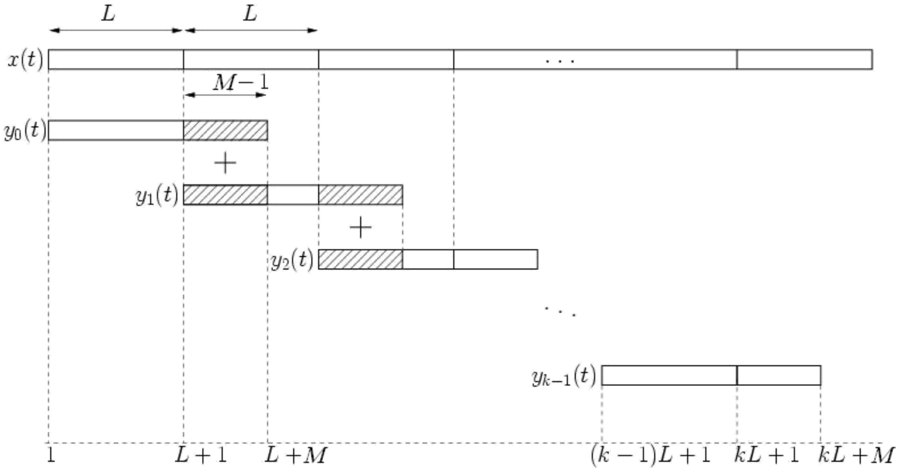

Figure 1: Overlap Add principle [wikipedia.org]

The steps of the algorithm for convolving a long signal x with a vector h of length M are as follows:

1. Split the signal x into smaller parts of size L
2. Convolve the first part of x, $x^{(1)}$ with h. The result $y^{(1)}$ has length $L + M - 1$.
3. The first L samples of the output are written to the output vector.
4. Convolve the second part of x, $x^{(2)}$, with h. The result $y^{(2)}$ has length $L + M - 1$.
5. Add the last $M - 1$ samples of y^{M-1} to the beginning of $y^{(2)}$.
6. Go back to step 3 and repeat for the next part of x, until the end.

You can read more details on the Wikipedia page or in the reference material.

The advantage of the Overlap-Add is that the individual convolutions are performed only with signals of length up to L, which eases memory and computational constraints.

Exercises

We will implement an Overlap-Add convolution to create a reverberation effect.

1. Download the "IM Reverbs Pack" archive from http://www.voxengo.com/impulses/. It contains impulse responses that create a reverberation effect. Unzip and play the file "Scala Milan Opera Hall.wav".

2. Load file "Scala Milan Opera Hall.wav" in Matlab (use `audioread()`). Restrict the data to about 1 second length. Call the resulting vector **`h`**.

3. Load the first 4 seconds of "Kalimba.mp3" (use `audioread()`), convolve with **`h`** and play the result (use `audioplayer()`).

4. Write a script to process the whole file "Kalimba.mp3" using Overlap-Add, after the following schematic:

```
% read "Scala Milan Opera Hall.wav" and save as vector h
...

% define M = the length of h
M = ...

% read first 4 seconds of "Kalimba.mp3" in vector x
...

% define L = the length of x
L = ...

% convolve with h, write the output in y
y = conv(x, h);
```

```
% initialize "overlap" vector with M-1 zeros
overlap = zeros(1, M-1);

while(1)
    % Play the first L samples from y.
    % Move into a temporary vector y2 for safety and play from there.
    y2 = y(1:L);
    player = audioplayer(y2,Fs);
    play(player);

    % read next 4 seconds of file
    ...

    % convolve with h, write the output in y
    y = ...

    % add previous "overlap" to the first M-1 elements of y
    y(1:(M-1)) = ...

    % save the last M-1 samples from y under the name "overlap", to be used in the ne
    overlap = ...

    % wait for playing to end
    while (player.isplaying)
      ; % do nothing
    end
end
```

Final questions

1. TBD