1 Framing and deframing

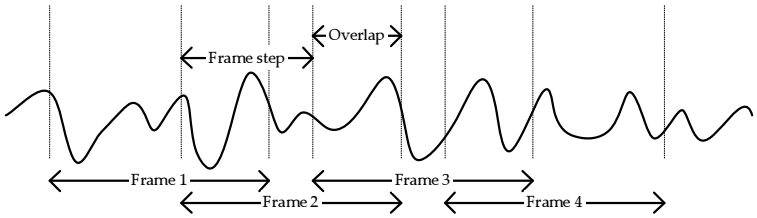
In speech processing it is often advantageous to divide the signal into frames to achieve stationarity. This worksheet describes how to split speech into frames and how to combine the frames into a speech signal.

1.1 Partitioning of a speech signal into frames

Normally a speech signal is not stationary, but seen from a short-time point of view it is. This result from the fact that the glottal system can not change immediately. XXX states that a speech signal typically is stationary in windows of 20 ms. Therefore the signal is divided into frames of 20 ms which corresponds to n samples:

$$n = t_{st} f_s \tag{1.1}$$

When the signal is framed it is necessary to consider how to treat the edges of the frame. This result from the harmonics the edges add. Therefore it is expedient to use a window to tone down the edges. As a consequence the samples will not be assigned the same weight in the following computations and for this reason it is prudent to use an overlap.



Figur 1.1: Illustration of framing. The speech is divided into four frames.

Figure **??** shows how a speech signal is divided into frames. Each frame shares the first part with the previous frame and the last part with the next frame. The time frame step t_{fs} indicates how long time there is between the start time of each frame. The overlap t_o is defined as the time from a new frame starts until the current stops. From this follows that the frame length t_{fl} is:

$$t_{fl} = t_{fs} + t_o \tag{1.2}$$

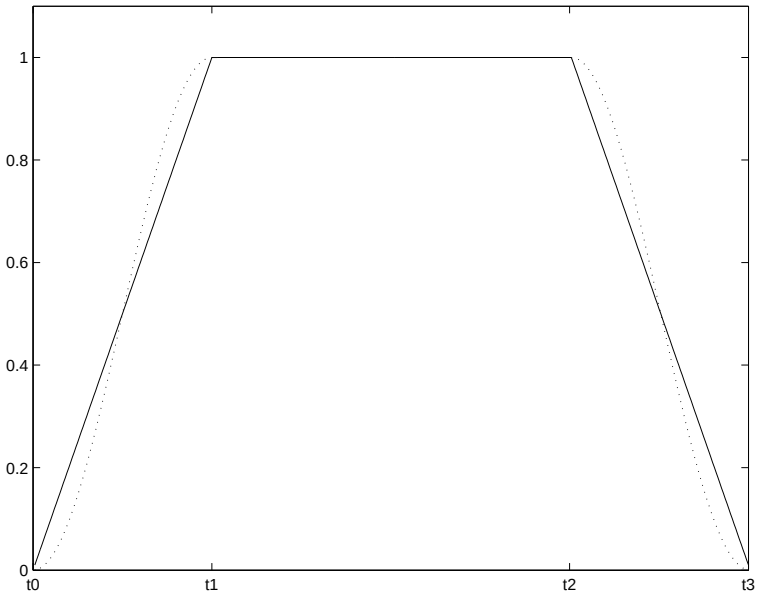
Hence the window has got to be of length t_{fl} which corresponds to $t_{fl} f_s$ samples.

1.2 Deframing

After performing computations on each individual frame it is desirable to combine the frames into a coherent speech signal. This task contains two problems: To take account of the formerly used window and to combine the samples shared by different frames.

A possibillity of the removal of the window is to multiply the frame by the reciprocal window. Before doing this it is essential to take precaution that the window is different from zero. This claim restrain the number of possible windows for example the triangular, the hanning and the blackman window.

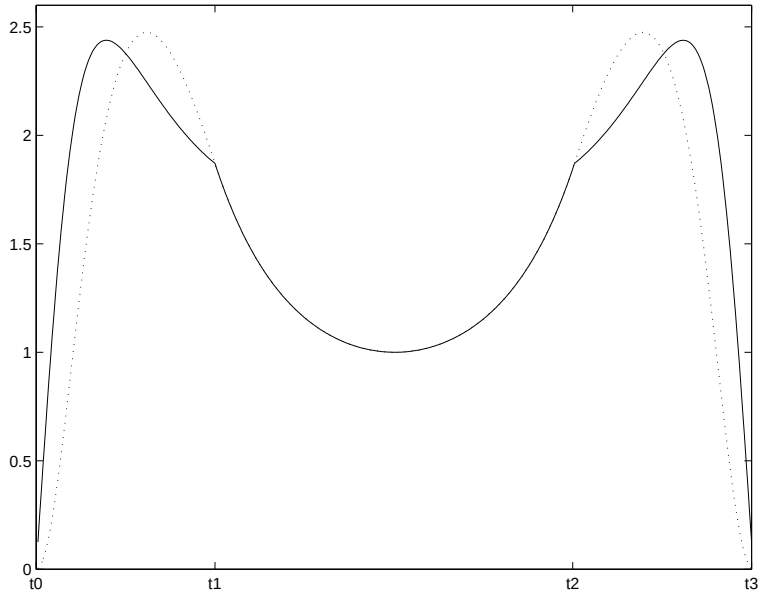
The common samples of two frames can be combined by averaging in such a way that the closer the samples get to the edge of the frame the less weight they are given. Figure **??** shows two functions that average the beginning and the end of a frame. The functions are based on a frame that has got an overlap from t_1 to t_2 and from t_3 to t_4. The solid line is a half triangular window in each end and the dashed line is a half hanning window.



Figur 1.2: Two different functions to average the overlap of frames.

The triangular window provides a constant amplitude but gives a point of discontinuity while the hanning window have the opposite effect.

Figure **??** shows the functions shown in **??** divided by a hamming window to compensate for the windowing performed by the framing of the signal. Multiplication by one of these functions will dewindow the signal and average the overlapping sections. This product can be converged to the speech signal by joining the frames by addition in the overlapping



Figur 1.3: Two functions that removes a hamming window and averaging the overlaps of two frames.

zones.

1.3 Matlab Code

```
close all
clear all
clc

[speech,fs,nbits]=wavread('ANN10045.wav');
% t=0:1/20000:1;
speech=speech(40000:75000);%sin(2*pi*542*t)';%
speech8=resample(speech,16000,fs);
fs8=16000;

SegmentStep8=fs8*.025;
Overlap8=fs8*.015;
SegmentLength8=SegmentStep8+Overlap8;
SpeechLength8=length(speech8);
nSegments8=floor(SpeechLength8/(SegmentStep8))-1;
Window8=hamming(SegmentLength8);

de=hanning(2*Overlap8-1)';
dewindow=[de(1:Overlap8), ones(1,SegmentLength8-2*Overlap8) de(
    Overlap8:end)]'./Window8;%

recons=zeros(SpeechLength8,1);

for i=1:nSegments8
    speech8Segment(:,i)=speech8((i-1)*SegmentStep8+1:i*SegmentStep8
        +Overlap8);
    speechW8(:,i)=Window8.*speech8Segment(:,i);
    speechde(:,i)=speechW8(:,i).*dewindow;
    recons((i-1)*SegmentStep8+1:i*SegmentStep8+Overlap8)=...
        speechde(:,i)+recons((i-1)*SegmentStep8+1:i*SegmentStep8+
            Overlap8);
end
```