LUND INSTITUTE
OF TECHNOLOGY
Lund University
Department of Electrical and Information Technology

Optimal Signal Processing

Laboratory work
Short version

Martin Stridh
Leif Sörnmo
Bengt Mandersson
2012

Department of Electrical and Information Technology, Lund
University, Sweden}

Speech coding

Speech model

The speech signal consists of voiced sound and unvoiced sound. The voiced sound consist of pulse train (periodical signal) generated in the glottis and then filtered in the mouth. The unvoiced sound (non-periodical signal) is generated from noise and then also filtered in the mouth.

An example of waveform is shown in Fig. 1 below.

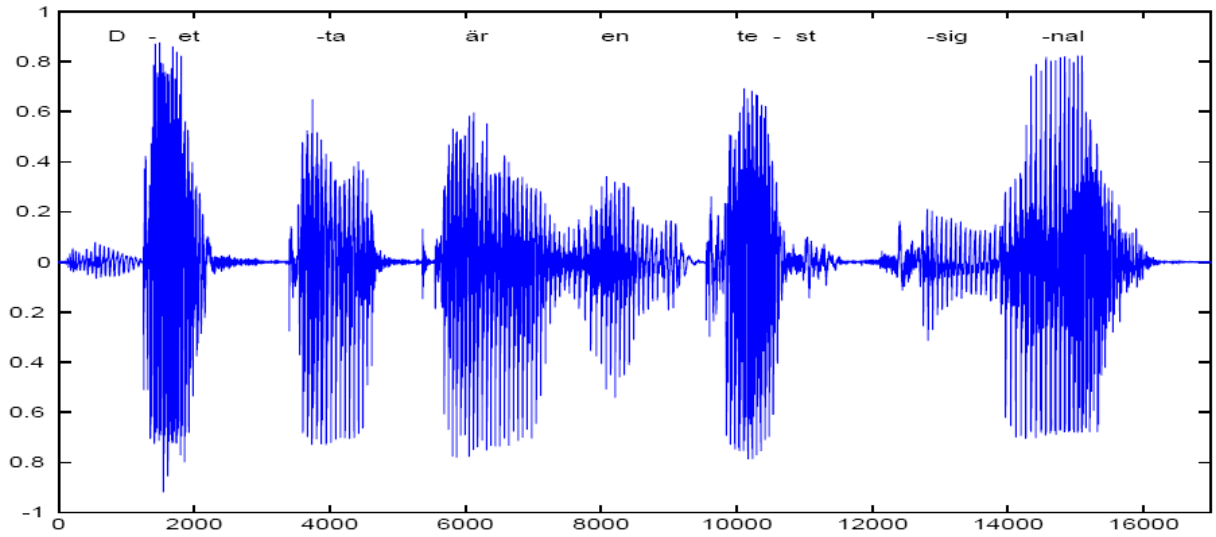


Figure 1: Example of a speech signal s(t) (waveform).

Spectra for periodical sound (left) and for non-periodical sound (right) are shown in Fig. 2.

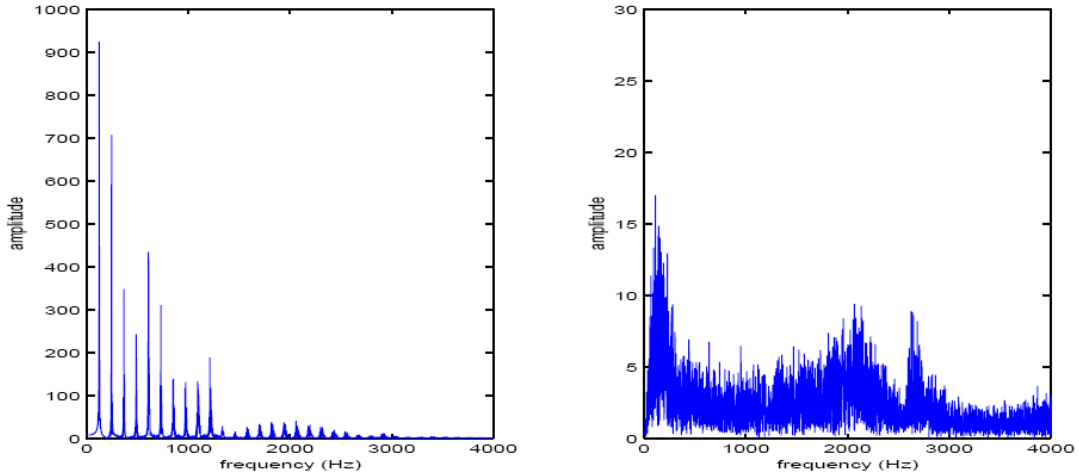


Figure 2. Left: Spectrum for the vowel 'a'. Right: Spectrum for non-voiced sound's'.

Laboratory work 1: GSM Speech coding

Preparation before the laboratory work

Read chapter 4.7.2, 5.2.6 in the textbook by Hayes.

Introduction
Global System for Mobile communications (GSM) is a digital system for mobile communication. It is developed in Europe but spread all over the world.

In 1982, Conference of European Posts and Telegraphs (CEPT) start a special group *Groupe Special Mobile* for development of a new digital mobile phone system for Europe.

In 1989, the project was move to *European Telecommunication Standards Institute* (ETSI). The first recommendation published in 1990 and the first system started around 1991.

In this laboratory work. we will look at the speech coder kbit/s Linear Predictive Coder - Long Term Prediction - Rectangular Pulse Excitation} (LPC-LTP-RPE), *GSM Full-rate* (FR) there FR means 13 kbit/s.

Description of the GSM speech coder

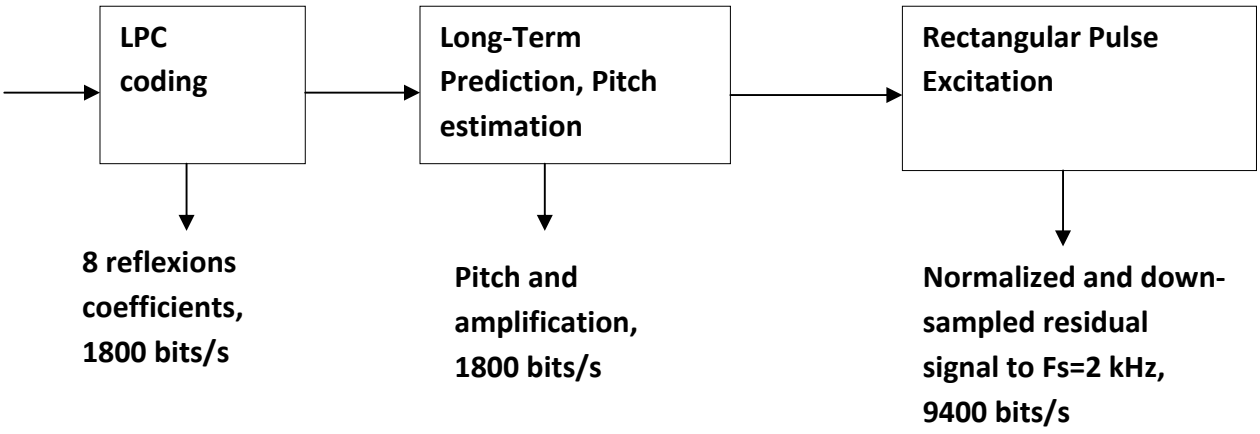


Figure: Block diagram for the GSM FR speech coder.

Input signal
Sample with 8 kHz sampling rate and 13 bits resolution.

Pre-filtering
First high pass filtering in two steps,

$$H_{offset}(z) = \frac{1 - z^{-1}}{1 - \alpha z^{-1}} \qquad \alpha = 32735 * 2^{-15}$$

$$H_{preemph}(z) = 1 - \beta z^{-1} \qquad \beta = 28180 * 2^{-15}$$

Linear predictive coding
In GSM, the filter parameters are updated every 160 samples (20 ms). The signals are first divided into blocks of 160 samples.

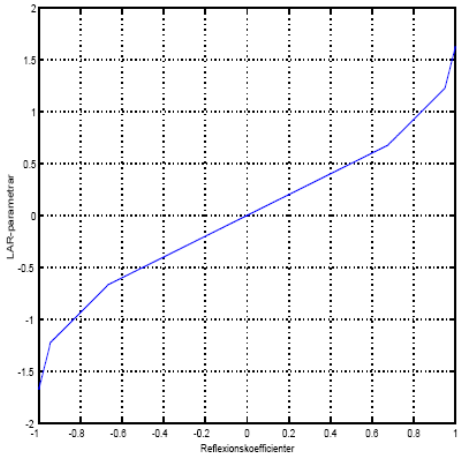
$$V(z) = \frac{1}{A(z)} = \frac{1}{1 + a_1 z^{-1} + a_2 z^{-2} + \ldots + a_8 z^{-8}}$$

First, estimate $r_s(0) - r_s(8)$ with (Hayes [4.153]).

$$r_s(k) = \sum_{i=k}^{159} s(i)s(i-k) \qquad k = 0..8$$

Then, determine *A(z)*-parameters and the reflection coefficients. To increase the resolution for values closed to -1 and 1, determine the Log-Area Ratios (LAR) according to

$$LAR_i = log_{10}(\frac{1 + \gamma_i}{1 - \gamma_i})$$



Expander for the reflection parameters.

Eight LAR-parameters use 36 bits for each segment of 160 samples
which gives 1800 bits/s.

Pitch analysis and residual coding

The pitch is estimation by using long term prediction. The residual after the pitch extraction are down-sampled to 2000 Hz.

GSM FR decoder

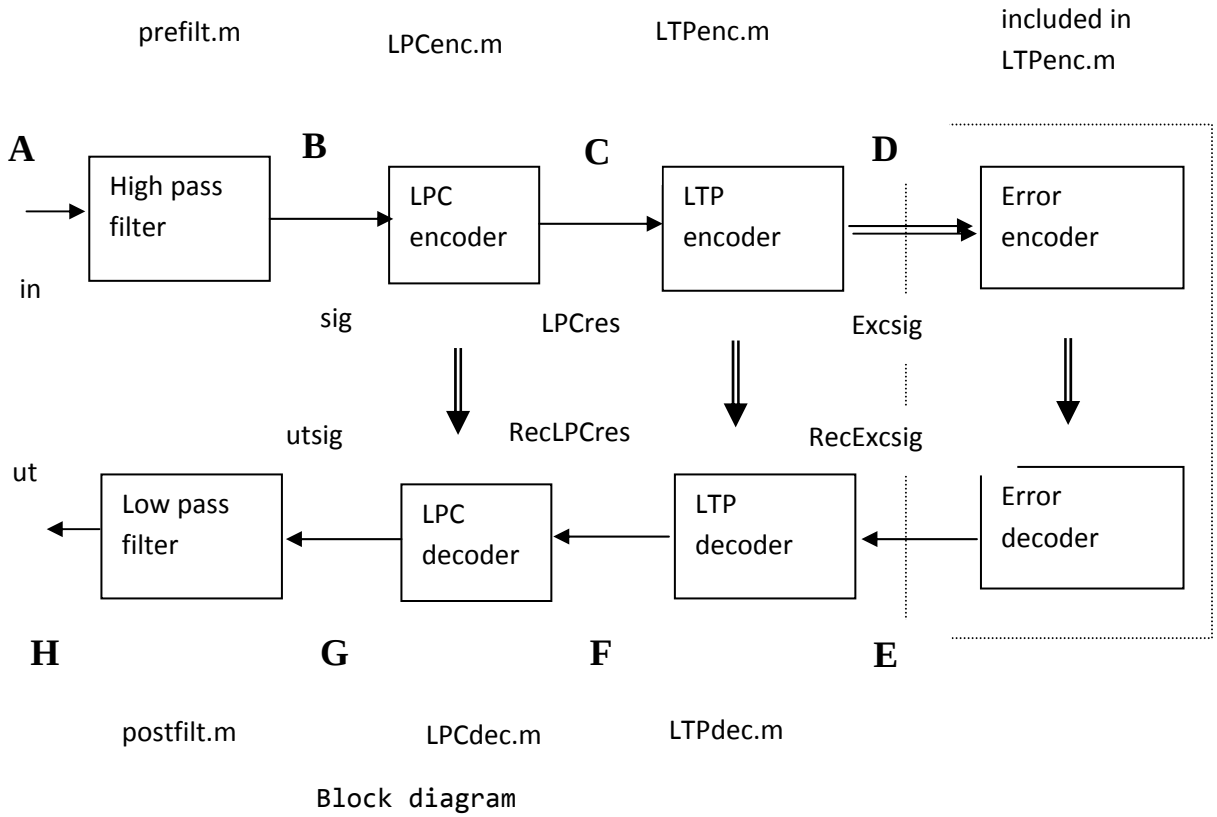
The decoder receive the error signal, add the pitch and, then pass through the LPE encoder and finally the post filter.

MATLAB files for the laboratory work

function out=convert(in, Fs_in,out, no_of_bits_in, no_of_bits_out)
function [LPCres,KvantLAR]=LPCenc(sig,AntalPar);
function utsig=LPCdec(KvantLAR,RecLPCres);
function [Excsig,RecExcsig,KvantN,Kvantb,KvantM,KvantMax,KvantExc]=LTPenc(LPCres);
function RecLPCres=LTPdec(KvantN,Kvantb,KvantM,KvantMax,KvantExc,Res);
function ut=prefilt(in);
function ut=postfilt(in);

Laboratory exercises

We will determine the signals in different steps in the speech encoder/decoder. The step are shown below together with the Matlab command in each step.



Block diagram

Matlab commands for the various steps:

Task 1: Convert to Fs=8 kHz
signal=convert(in,Fs,8000,bits,13) % if not Fs=8 kHz
Task 2: (A-B): Do high pass filtering
sig=prefilt(insignal);
Task 3: (B-C) Determine the LPC coeff. and the residual
[LPCres,KvantLAR]=LPCenc(sig,8);
Task 4-5: (C-D-E) Determine pitch, amplitude and downsampled residual
[Excsig,RecExcsig,KvantN,Kvantb,KvantM,KvantMax,KvantExc]=LTPenc(LPCres);
Task 6: Add pitch, and amplitude
RecLPCres=LTPdec(KvantN,Kvantb,KvantM,KvantMax,KvantExc,Res);
Task 7: Filter with the LPC coefficients
utsig=LPCdec(KvantLAR,RecLPCres);
Task 8: Do low pass filtering
ut=postfilt(utsig);

Preparation: Record the test signals

Start WaveStudio (or some other program for recording) and record 1 second of a voiced sound (vowel 'a') and one second of a non-voiced sound ('s').

Press Start.
Speak into the microphone.
Press Stop.
Select one seconds of the signal.
Then Edit and Copy.
Then File and New.
Paste Edit.

Save as 'a_sound.wav' resp. ' s_sound.wav' in your own directory.
Start **Matlab** and type ***init_osb***.

Task 1: Prepare the signals to 8 kHz sampling rate

Read your files with ***wavread*** and convert to 16-bits, 8 kHz sampling rate, ***convert.*** Now select one second.

```
[in,Fs,bits]=wavread(myfile);
signal=convert(in,Fs,8000,bits,13)      % if not Fs=8 kHz
insignal=signal(1:8000);                %select about 1 s if signals to long
```

Hint: Use exactly the name of the signals below in the reminder of the commands.

Task 2: High pass filtering

Prefilt your signals with ***prefilt***. Also plot the first 2000 samples of the signal before and after the filter.

```
sig=prefilt(insignal);
soundsc([insignal;sig]);  %listen
```

Task 3: Do LPC analysis

Do the LPC analysis with the function **LPCenc.** Use eight filter coefficients. Plot the LPC residual and the quantified LAR parameters. Is eight coefficients OK? Any difference between voiced and non-voiced sound?
Listen. Explain the results.

```
[LPCres,KvantLAR]=LPCenc(sig,8);
soundsc(LPCres,Fs);
```

Task 4 Compute LPT

Compute the LTP analyzes of the LPC residual using **LTPenc**. Plot the 'Excsig' and the corresponding quantified signal 'RecExcsig'. What is the fundamental frequency
in your vowel 'a'. Listen to the 'Exsig' (error signal after LPT filtering). Compare with 'RecExsig' (reconstructed excitation signal).

[Excsig,RecExcsig,KvantN,Kvantb,KvantM,KvantMax,KvantExc]=LTPenc(LPCres);

soundsc(Excsig,Fs);
soundsc(RecExcsig,Fs);

Task 5: Plot signals

Plot 2000 samples of each signal in each step above.
Listen and compare the vowel and the 's' sound.

Task 6: Reconstruction of LTP

Reconstruction: Decode the quantified parameters and reconstruct
the LPC residual. Use **LTPdec**. Listen and compare with
signals in the step before.
RecLPCres=LTPdec(KvantN,Kvantb,KvantM,KvantMax,KvantExc,Res);

Task 7: Reconstruction of LPC

Synthesize now using **LPCdec**. Listen to the results. Comment.
utsig=LPCdec(KvantLAR,RecLPCres);

Task 8: Low pass filtering

Postfilt with **postfilt**.
ut=postfilt(utsig);
soundsc(ut,Fs);

Task 9: Plot signals in each step

Compare the signals before and after the speech coder. Listen to the signals after each step in the coder.

Task 10: Use white noise as error signal

Now use white noise as error signal (use 'vit' as input to **LTPdec**. Corresponds to 3600 bits/s). Then, rerun step 7,8 and 9.

Laboratory work 2: Power spectrum estimation

Preparation before the laboratory work}

Prepare the laboratory work by reading chapter 8 in Hays' book.

Introduction
In this laboratory work we will test some of the methods for
power spectrum estimation given in Hay's book.

Matlab files and signal files

function Px=pergram(x,n1,n2) % Periodogram in Hayes' book.
function Px=mper(x,win,n1,n2); % Modified Periodogram (see Hayes' book).

function Px=welch(x,L,over,win); % Welch method (Hayes' book).

afib.mat % ECG-signal

And all files from laboratory 1.

Laboratory works

Task 1
Record an 'A-sound' (vowel 'a') and a short sentence (4-5 words)
using Wavestudio and save it into files.

Task 2
Resample the signals to 8 kHz sampling rate and 13 bits
resolution using **convert**. Select about 1 second of the
signals.
(for example signal=signal(1:8000);).

Task 3

Use the functions { **pergram, mper** and **welch**} to analyze the
signals. Downsample by a factor 4 (**decimate**).

Task 4
Can you characterize the sound using your spectra.

Task 5
Load the signal **afib.mat**. Execute your Welch program file. Comment the contents of the power spectrum of the signal.

The signal is sampled with 1000 Hz sampling rate. Down-sample the signal by a factor 50 **decimate**. Then, only frequency components below 10 Hz are remaining. Comments?

Task 6
Modify your Welch-file so that each spectra form the frames are stored in different rows in a matrix. Then, you got a time-frequency analyzer.

Task 7
Analyze the 'A-signal' and the sentence with your time-frequency analyzer. Look for some different vowels in the sentence and describe the power spectra for these vowels.

Task 8
Run the speech coder program using the 'A-signal' and the sentence as inputs (sample rate 8 kHz). Down-sample the signals (*insignal,...,RecExcsig*) in the coder to 2 kHz and analyze the signal with your time-frequency analyzer. Explain the results.
