Analysis/synthesis coding

Many speech coders are based on a principle called *analysis/synthesis coding*. Instead of coding a waveform, as is normally done in general audio coders and image coders, we have a parametric model of the source. The coder (the analysis part) tries to estimate the model parameters from the signal to be coded. These parameters are sent to the decoder (the synthesis part) which uses them to control the same model and reconstruct the signal.

This will usually work well when we have a narrow class of signals where we have a good model of the source, such as human speech. However, analysis/synthesis coding might not work well for coding of general audio or image signals.

The decoded signal might not be similar to the original signal in a mean square error sense, but can still sound very much like the original signal to a human listener.

Analysis by synthesis

A variant of analysis/synthesis coding is *analysis by synthesis coding*. The coder also contains a decoder, and tries to find the model parameters that gives a decoded signal close (in some sense) to the original signal.

Human speech

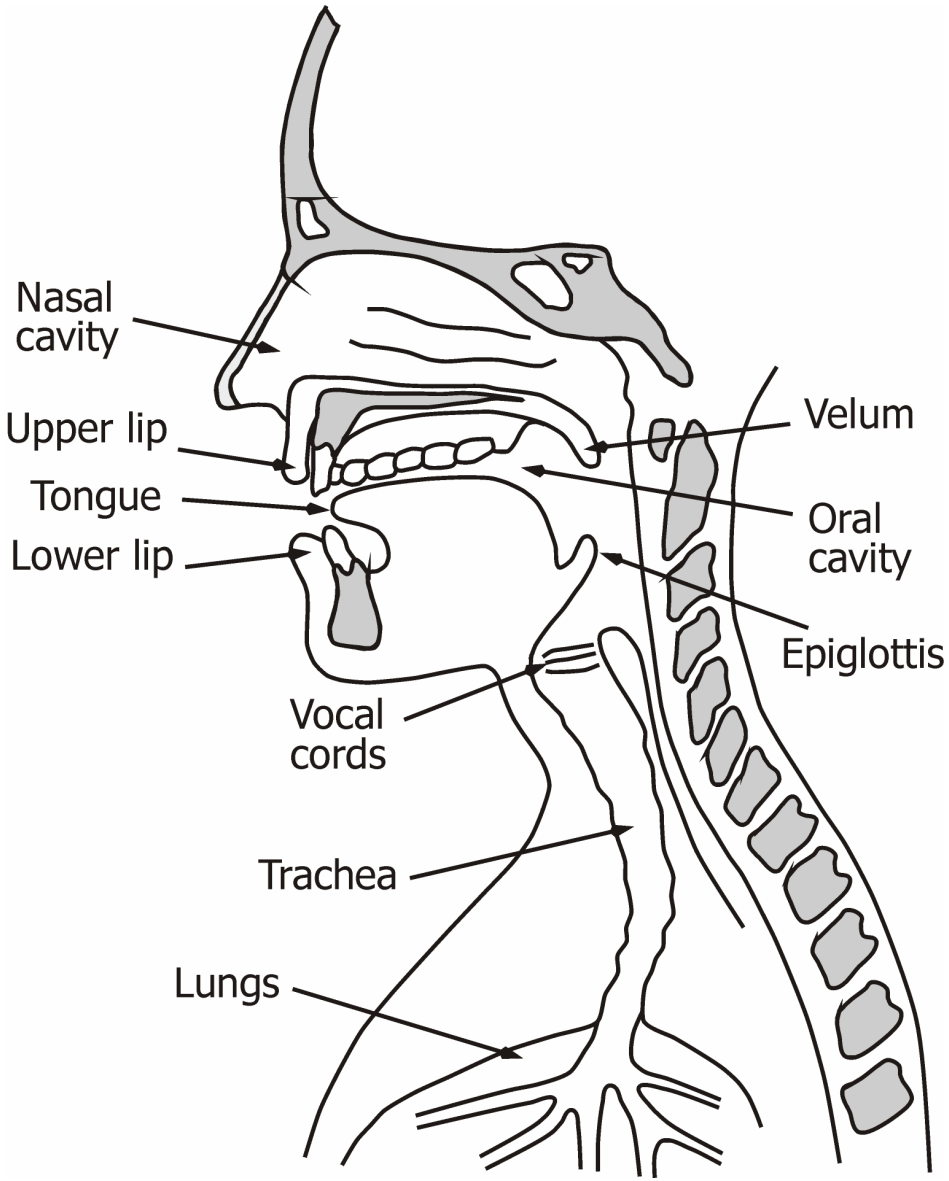
Sound is generated by forcing air through the vocal cords (located in the larynx). If the vocal cords are tense, they vibrate and generate tones and overtones (*voiced sounds*). If the vocal cords are relaxed, a noiselike sound is produced (*unvoiced sounds*).

The sound then passes through the laryngeal cavity, the pharynx and the oral and an nasal cavities.

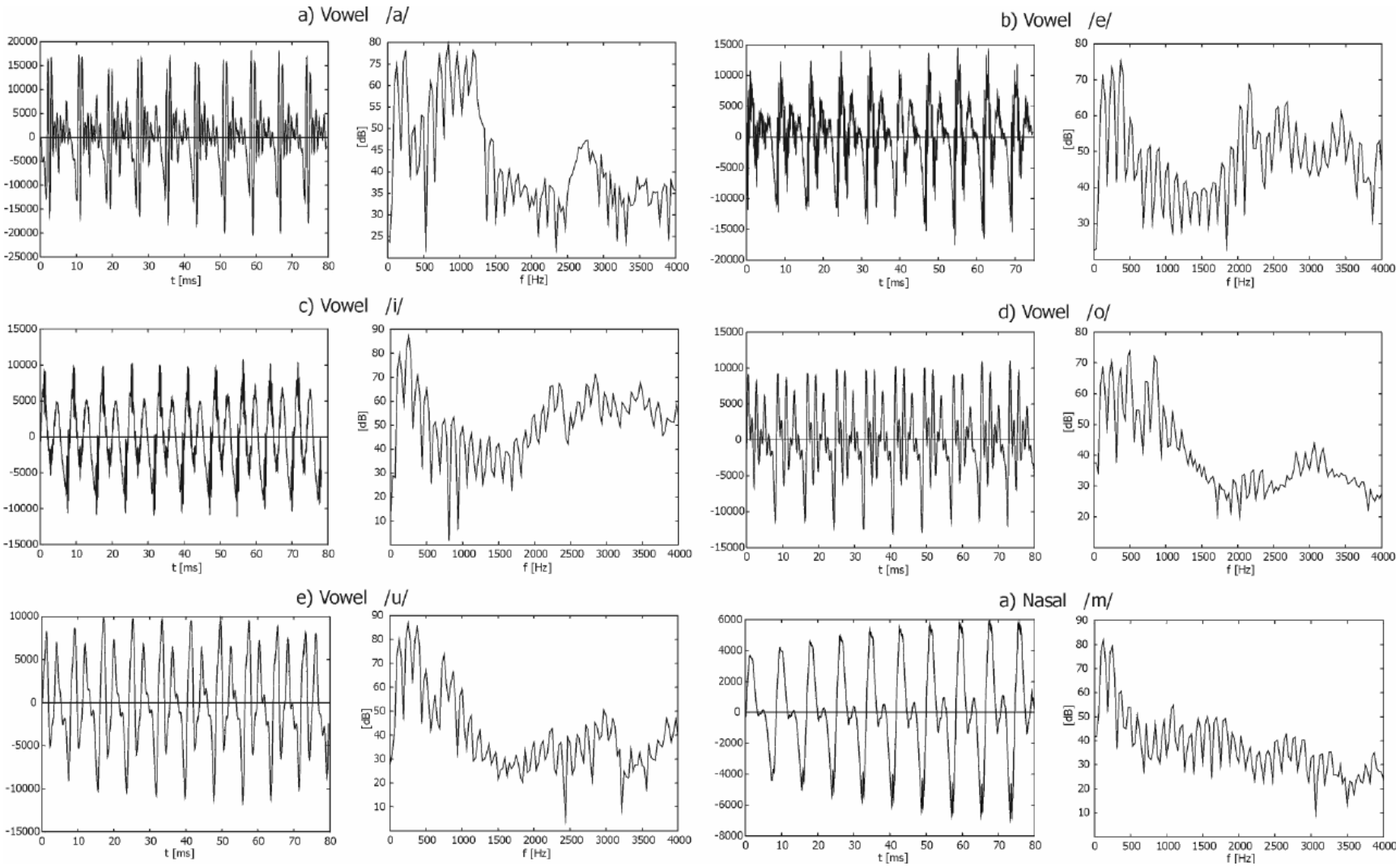
Tongue, lips and teeth are also used to influence the sound.

Everything after the vocal cords (the *vocal tract*) can be well modelled by a linear filter.

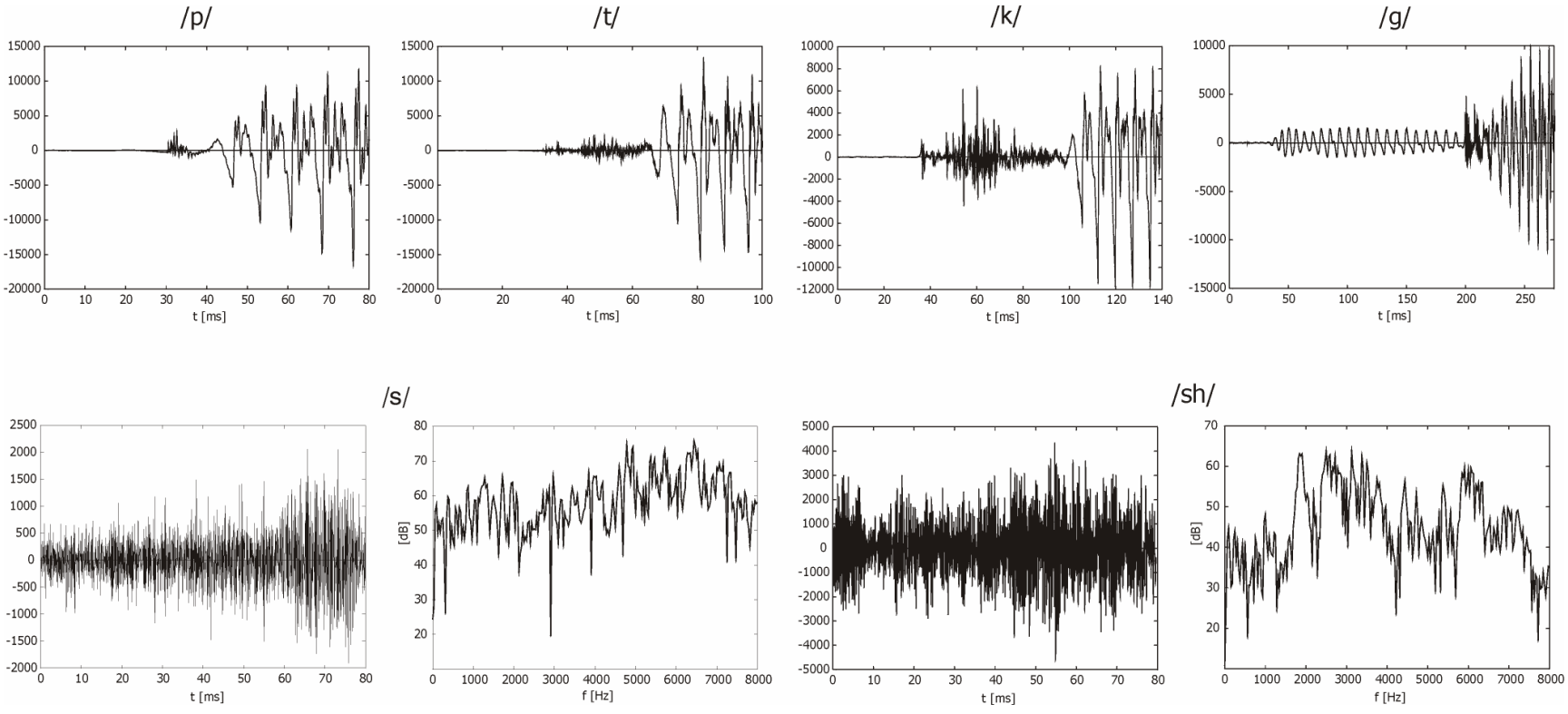
Human speech



Examples of speech sounds

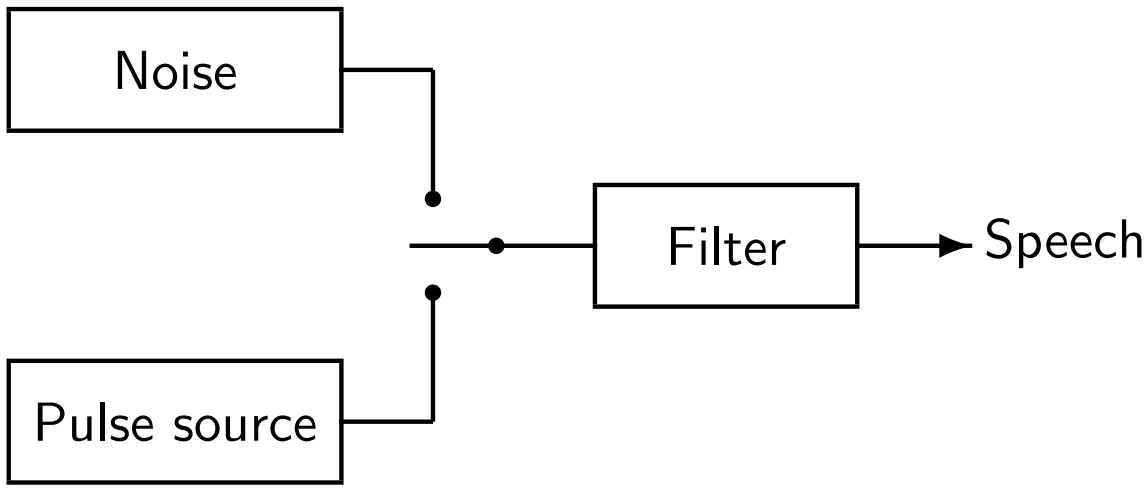


Examples of speech sounds



Model of speech

A simple model of human speech:



The speech is seen as a sequence of either voiced or unvoiced sounds. The voiced sounds are modelled as a filtered pulse train, while the unvoiced sounds are modelled as filtered white noise.

The parameters of the model are filter coefficients, switches between voiced and unvoiced sounds, and the pulse trains.

Model of speech, cont.

The speech signal y_n is modelled as

$$y_n = \sum_{i=1}^{M} a_i y_{n-i} + G\epsilon_n$$

The coder splits the signal into shorts segments of, typically a few hundred samples (at sampling frequency 8 kHz). For each segment the coder estimates if the sound is voiced or unvoiced. For voiced sounds a suitable pulse train is estimated. Filter parameters a_i and G are estimated. All these parameters are sent to the receiver, which can then decode the sound using the model.

The coding us thus a kind of linear predictive coding. One major difference, compared to our earlier description of predictive coding, is that the main part of the bit rate is used to send filter coefficients and not the prediction error.

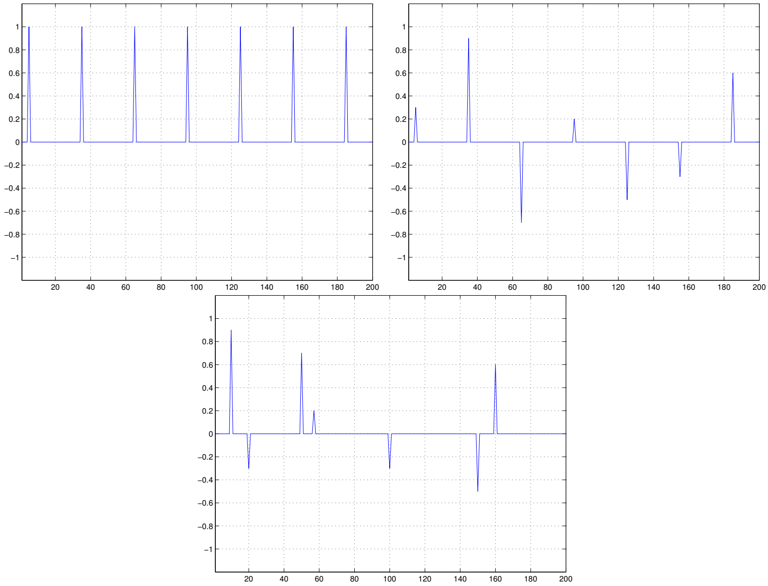
Pulse trains

The pulse trains can vary in complexity between different coders. Often the *pitch period*, corresponding to the fundamental frequency of the sound, is estimated.

The simplest pulse trains use pulses of the same amplitude at constant intervals. The pulse train can then be described just by the pitch period and the start position of the first pulse.

We can also let the amplitudes and positions of the pulses very more freely. It is then possible to get a pulse train that fits the signal better, but at the cost of a higher bit rate.

Pulse trains, examples



Voiced or unvoiced?

Voiced sounds usually have a larger energy (larger amplitude) than unvoiced sounds.

Unvoiced sounds usually contain higher frequencies than voiced sounds.

One way of determining if a segment is voiced or unvoiced can be to comapare the signal energy with the energy of the background noise, and to count the number of zero crossings of the sounds.

Estimating the pitch period

The auto correlation function $R_{yy}(k)$ can be used to estimate the pitch period P. For a periodic signal, the acf has a maximum at $k = P$.

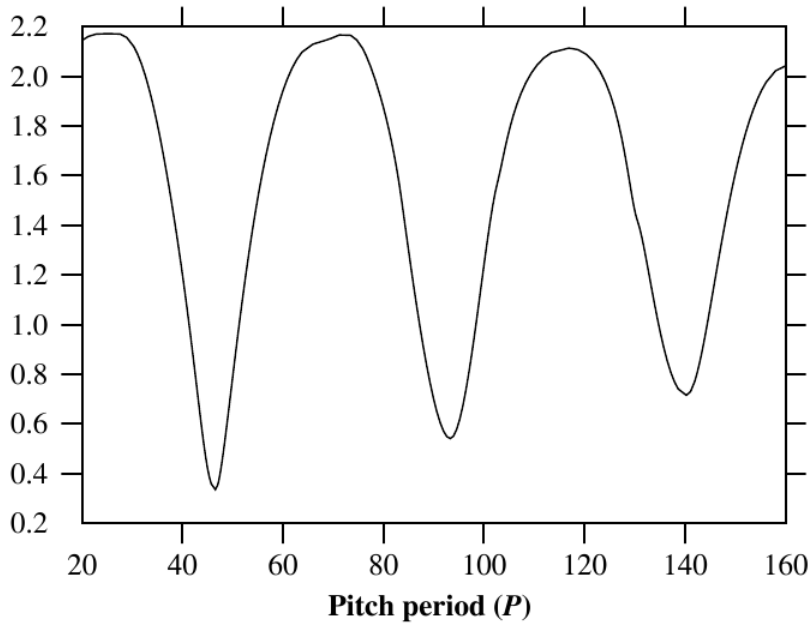
Another, better method is to use the *average magnitude difference function* (AMDF). It is defined by

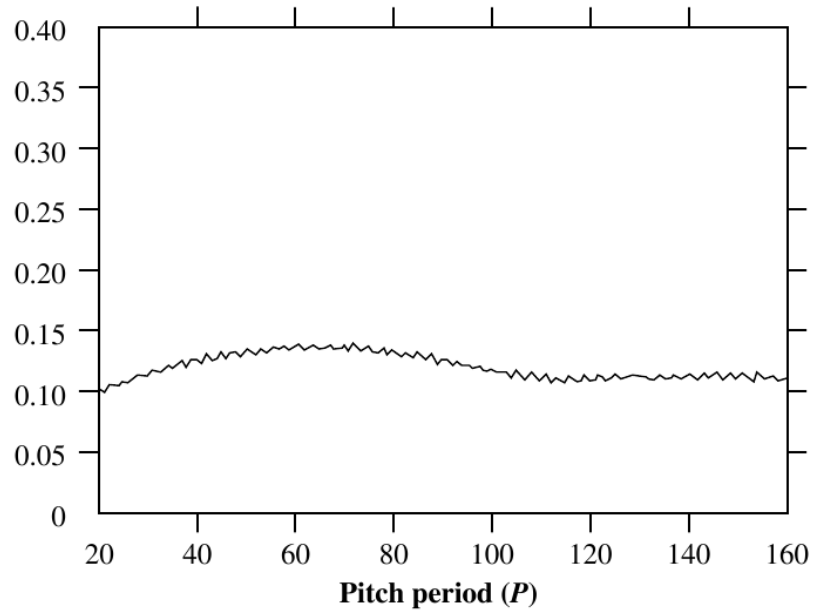
$$AMDF(k) = \frac{1}{N} \sum_{i=k_0+1}^{k_0+N} |y_i - y_{i-k}|$$

where k_0 depends on which segment we're in and N is the size of the segment. The AMDF will have a minimum where k is equal to the pitch period of the segment.

The AMDF can also be used to determine if the segment is voiced or unvoiced. For unvoiced sounds the AMDF will have very shallow minima, not much different from the average value of the AMDF.

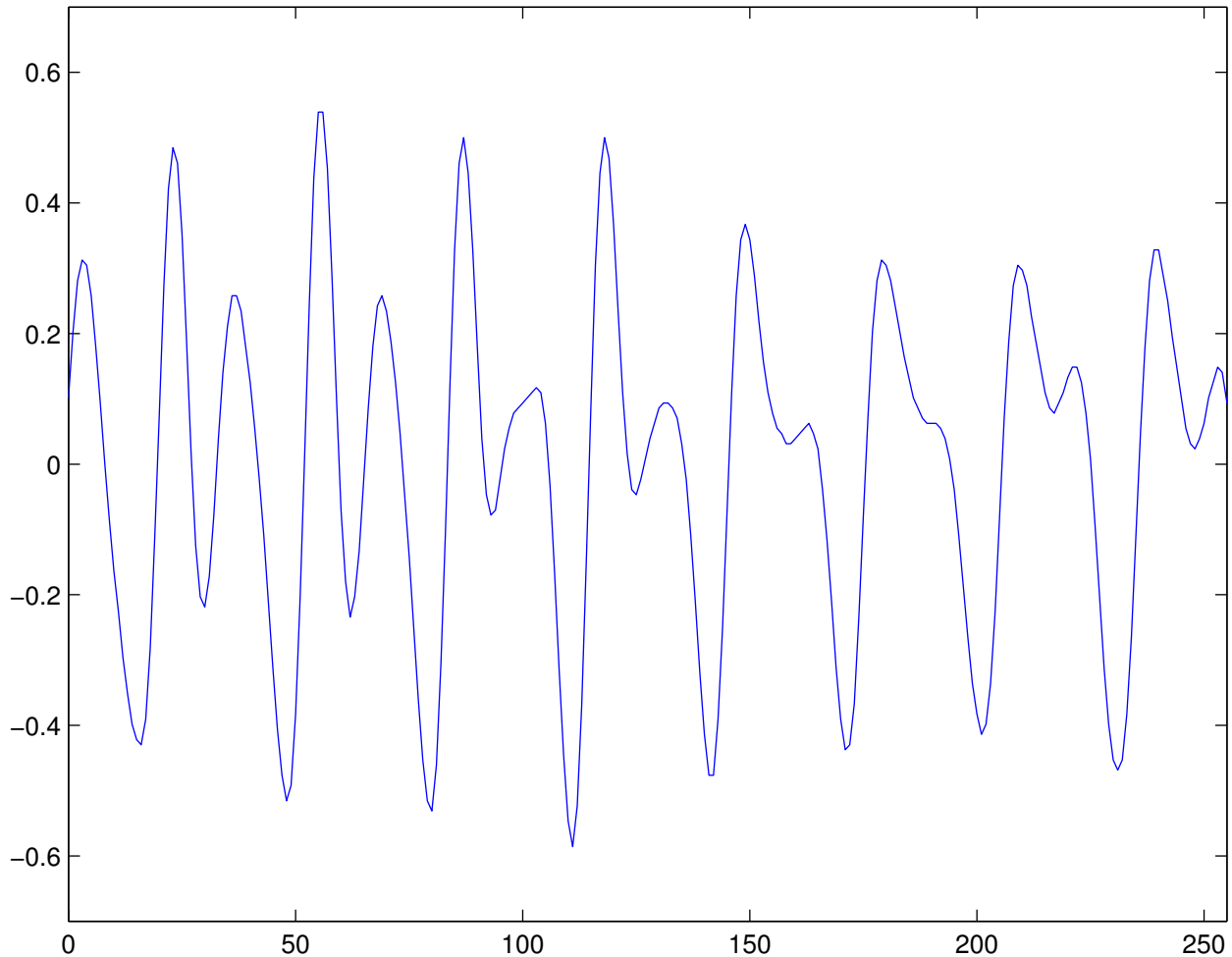
AMDF





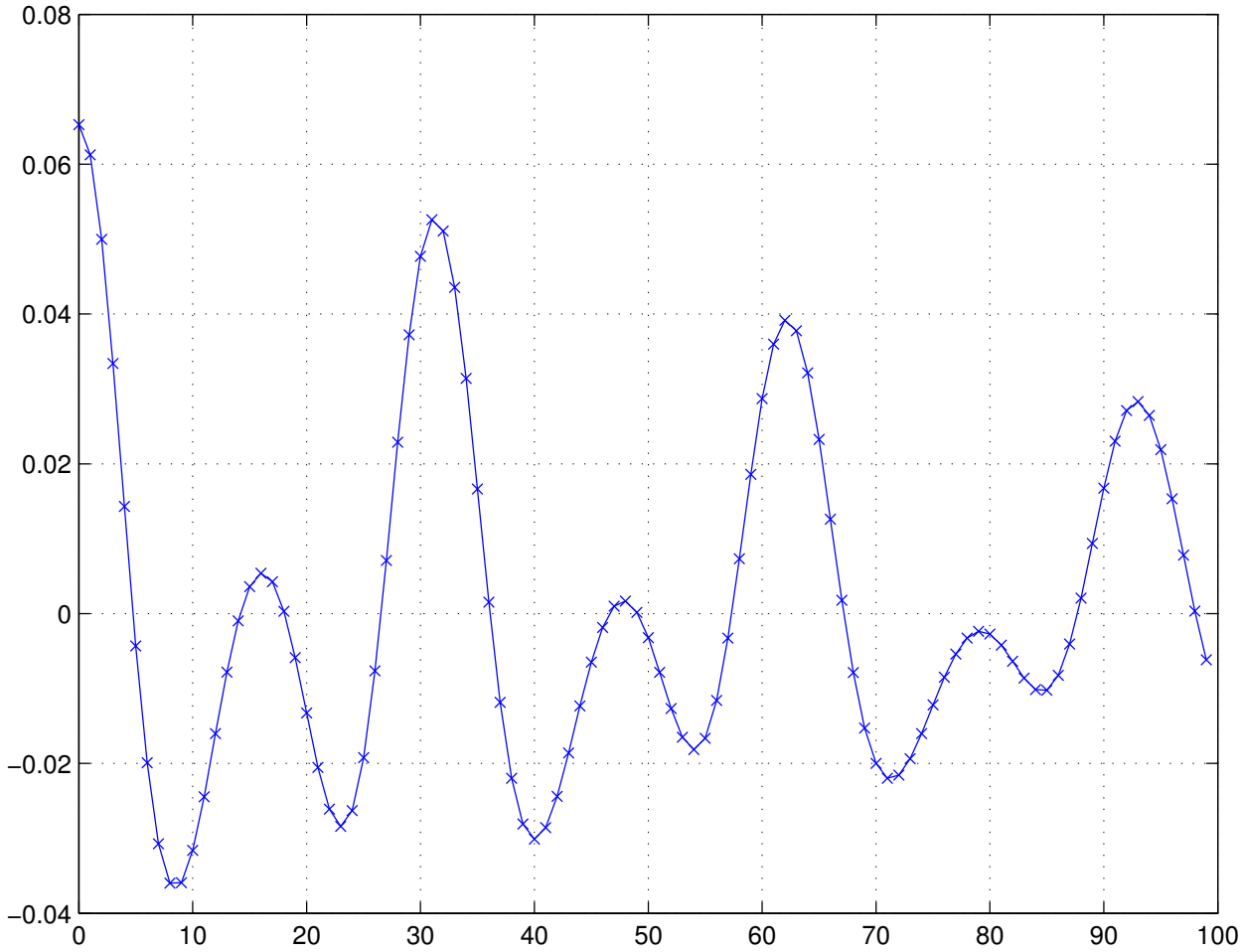
AMDF for a voiced (e) and an unvoiced (s) sound.

Example



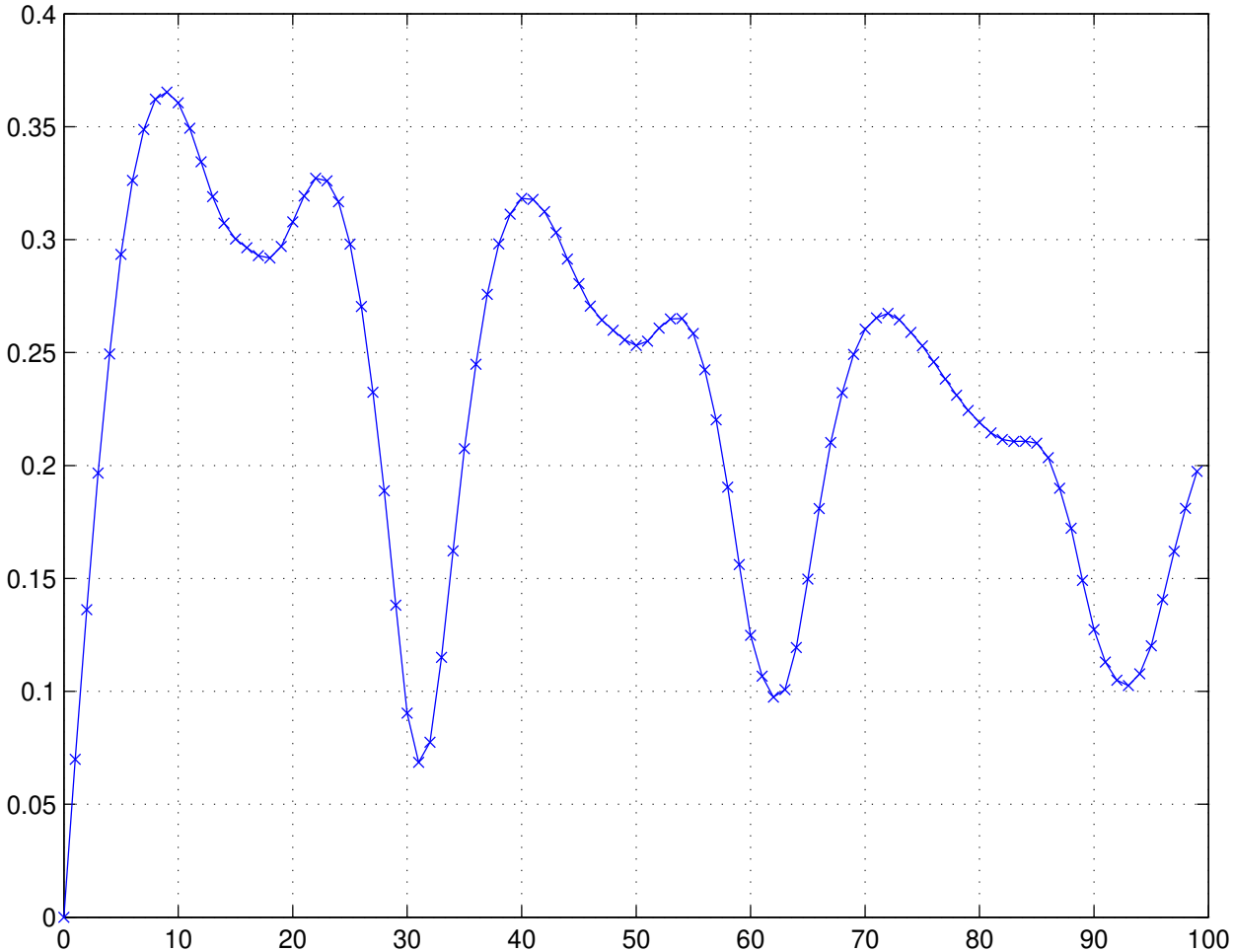
A segment of 256 samples from a speech signal.

Example



Estimated auto correlation function. Gives pitch period 31.

Example



Estimated AMDF. Gives pitch period 31.

Estimating filter coefficients

We want to find a_i such that the average value of the quadratic error e_n^2 is minimized, where

$$e_n^2 = (y_n - \sum_{i=1}^{M} a_i y_{n-i} - G\epsilon_n)^2$$

Minimizing the expected value $E\{e_n^2\}$ gives the following equation system

$$\frac{\partial}{\partial a_j} E\{e_n^2\} = 0 \quad \Longleftrightarrow \quad \sum_{i=1}^{M} a_i E\{y_{n-i} y_{n-j}\} = E\{y_n y_{n-j}\}$$

In order to solve this we need to estimate $E\{y_{n-i} y_{n-j}\}$, which can be done either by the *auto correlation method* or by the *auto covariance method*.

Auto correlation method

We assume that y_n is stationary, which means that

$$E\{y_{n-i}y_{n-j}\} = R_{yy}(|i-j|)$$

In addition, we assume that the signal is 0 outside of the current segment, so that we can estimate the auto correlation function as

$$R_{yy}(k) = \sum_{n=n_0+1+k}^{n_0+N} y_n y_{n-k}$$

where n_0 is the index for the first sample in the sequence and N is the length of the segment.

Auto correlation method, cont.

The equation system can then be written as

$$\mathbf{R}\bar{\mathbf{a}} = \bar{\mathbf{p}}$$

where

$$R = \begin{bmatrix} R_{yy}(0) & R_{yy}(1) & \dots & R_{yy}(M-1) \\ R_{yy}(1) & R_{yy}(0) & \dots & R_{yy}(M-2) \\ \vdots & \vdots & \ddots & \vdots \\ R_{yy}(M-1) & R_{yy}(M-2) & \dots & R_{yy}(0) \end{bmatrix}$$

and

$$\bar{\mathbf{a}} = [a_1 \ a_2 \dots a_M]^T$$

$$\bar{\mathbf{p}} = [R_{yy}(1) \ R_{yy}(2) \dots R_{yy}(M)]^T$$

Solve for $\bar{\mathbf{a}}$.

Auto covariance method

We do not assume that y_n is stationary. We define

$$c_{ij} = E\{y_{n-i}y_{n-j}\}$$

which can be estimated as

$$c_{ij} = \sum_{n=n_0+1}^{n_0+N} y_n y_{n-k}$$

Note that we are using samples outside of the segment in the estimation.

Auto covariance method, cont.

The equation system can then be written as

$$\mathbf{C}\bar{\mathbf{a}} = \bar{\mathbf{s}}$$

where

$$\mathbf{C} = \begin{bmatrix} c_{11} & c_{12} & \dots & c_{1M} \\ c_{21} & c_{22} & \dots & c_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ c_{M1} & c_{M2} & \dots & c_{MM} \end{bmatrix}$$

and

$$\bar{\mathbf{s}} = \begin{bmatrix} c_{10} & c_{20} \dots c_{M0} \end{bmatrix}^T$$

Solve for $\bar{\mathbf{a}}$.

LPC-10

Old american speech coding standard fo the rate 2.4 kbits/s.

- Segments of 180 sampel
- Pitch period 60 possible values
- 10 filter coefficients for voiced sounds, 4 coefficients for unvoiced sounds.
- Gives a rather synthetic decoded sound.
- Not so good for high background noise.

Long Term Prediction (LTP)

Often a predictor that utilizes both the most recent samples and samples one pitch period P back in time is used.

$$y_n = \sum_{i=1}^{M} a_i y_{n-i} + \sum_{j=1}^{K} \alpha_j y_{n-P-j+1} + G\epsilon_n$$

The part using α_j is called *long term prediction* and the part using a_i is called *short term prediction*.

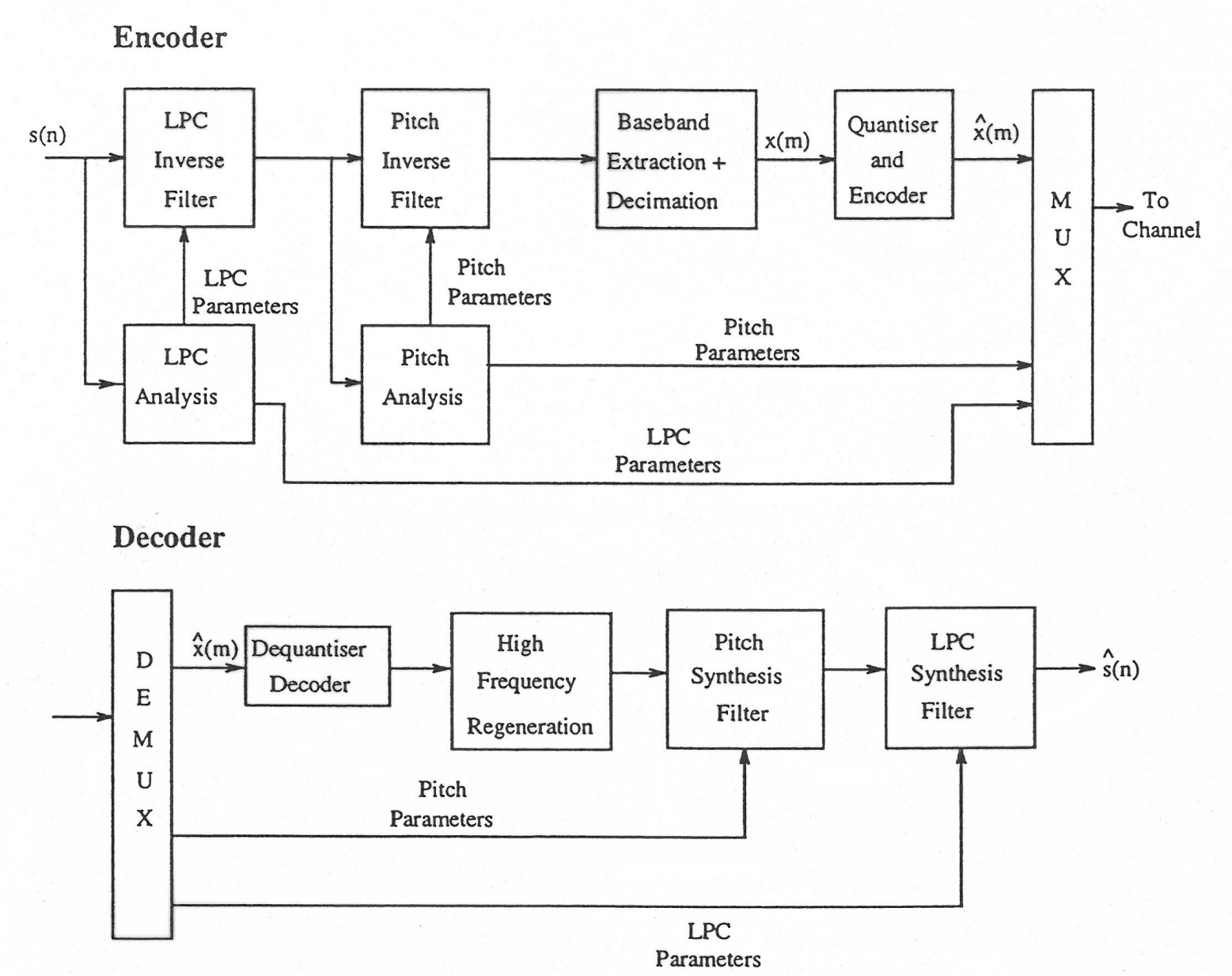
RELP

Regular Excitation Linear Prediction

In a RELP coder no choice between voiced and unvoiced sounds is made. The pitch period P and filter coefficients a_i and α_j are estimated. After inverse filtering we get a residual signal that is lowpass filtered, downsampled (typically a factor 3 or 4) and quantized and sent sampel by sampel.

A RELP coder is thus rather similar to a traditional predictive coder, where the prediction error (the residual signal) is sent. Note that the quantization is outside the predictor loop. This will work for the short segments that are used.

RELP


Encoder
s(n)
LPC Inverse Filter
Pitch Inverse Filter
Baseband Extraction + Decimation
x(m)
Quantiser and Encoder
x̂(m)
M U X
To Channel
LPC Parameters
Pitch Parameters
LPC Analysis
Pitch Analysis
Pitch Parameters
LPC Parameters
Decoder
D E M U X
x̂(m)
Dequantiser Decoder
High Frequency Regeneration
Pitch Synthesis Filter
LPC Synthesis Filter
ŝ(n)
Pitch Parameters
LPC Parameters

Multi-pulse LPC (MP-LPC)

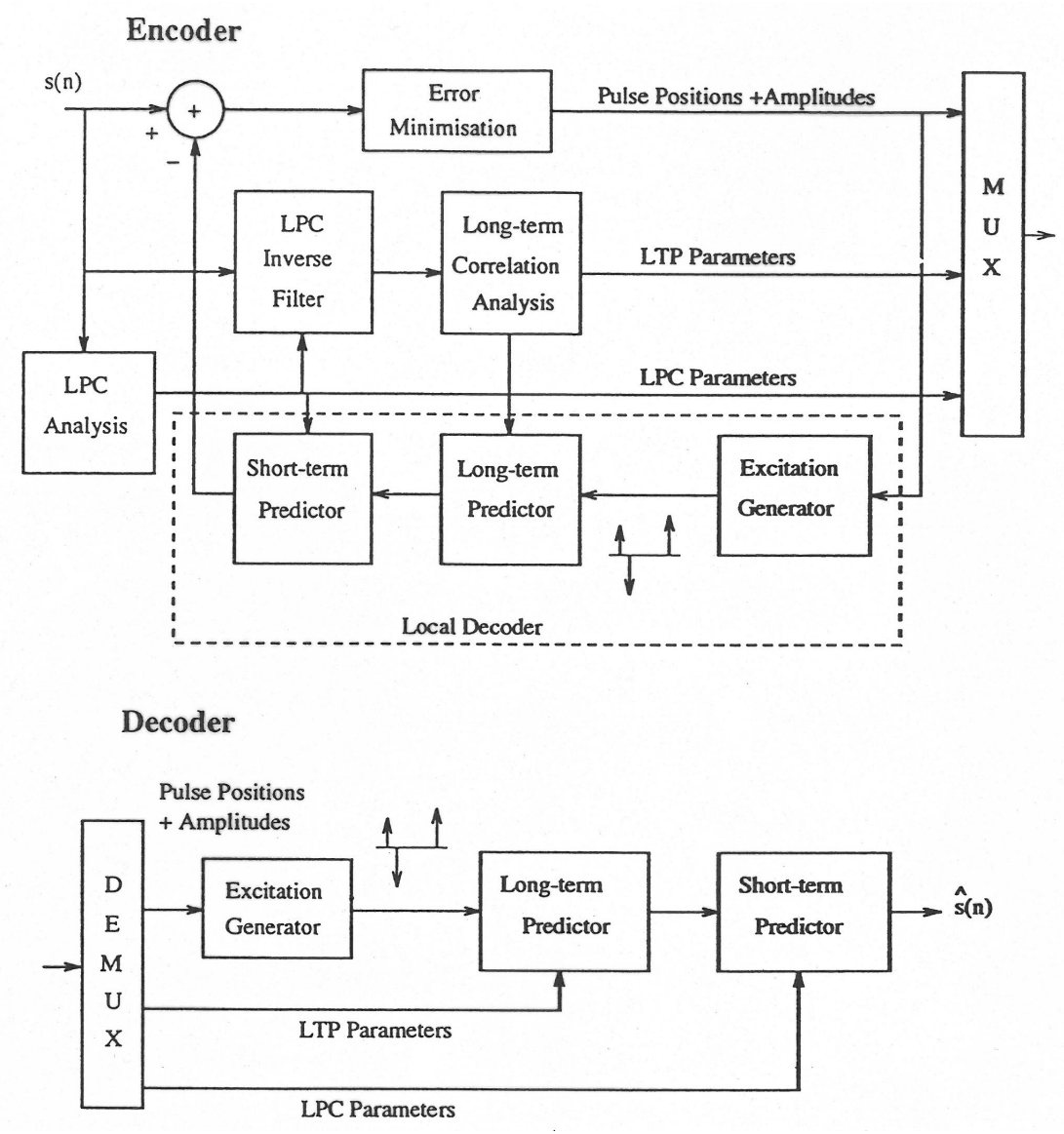
MP-LPC is an analysis by synthesis coder.

The coder estimates filter coefficients. The coder then tries to find an optimal pulse train (position and amplitude for a number of pulses) that will be decoded to a signal as close to the original signal as possible.

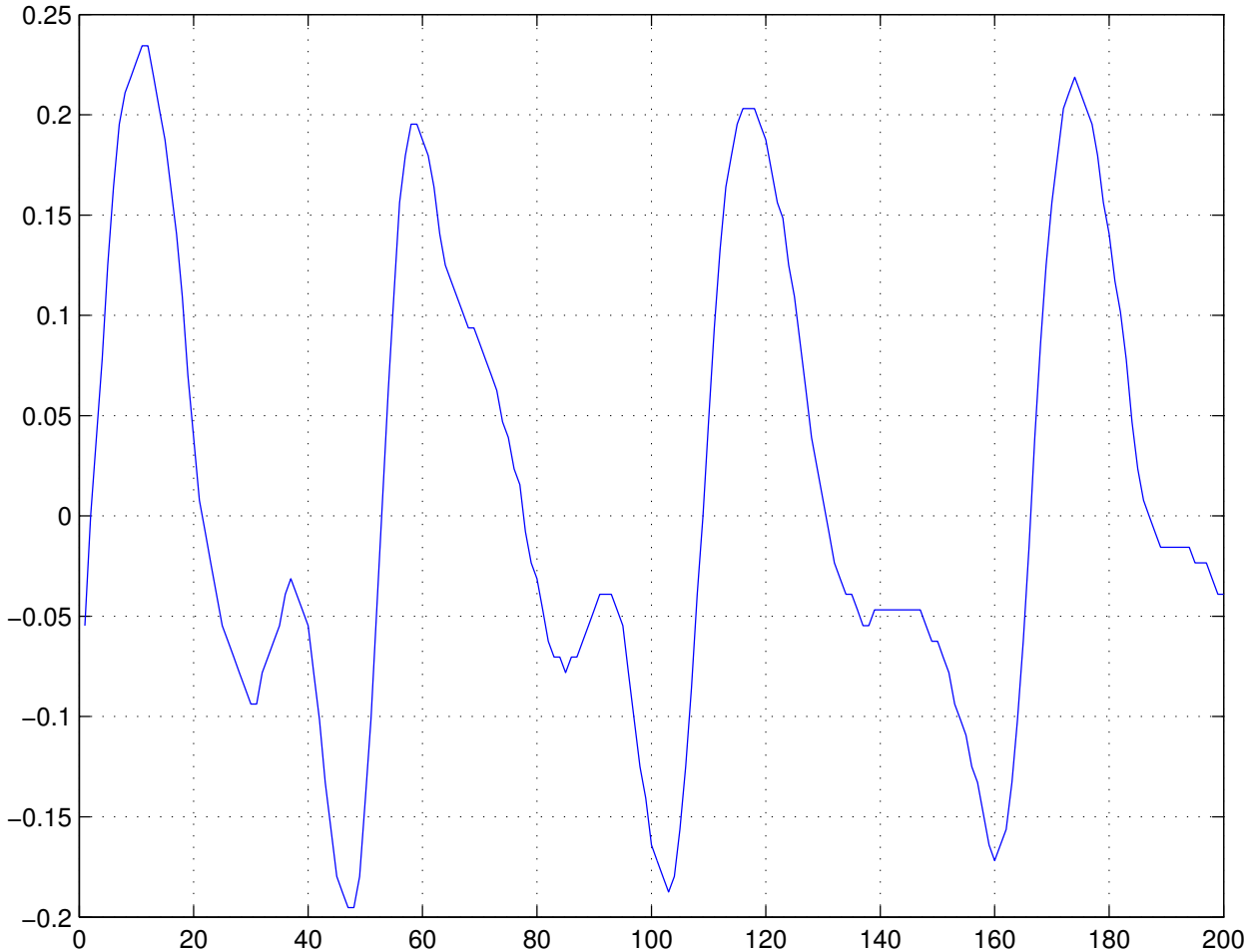
One disadvantage of MP-LPC is that the coding is rather computation intensive.

Used in Skyphone, a system for telephony from airplanes, with the rate 9.6 kbit/s

MP-LPC


Encoder
s(n)
+
+
−
Error
Minimisation
Pulse Positions +Amplitudes
M
U
X
LPC
Inverse
Filter
Long-term
Correlation
Analysis
LTP Parameters
LPC
Analysis
LPC Parameters
Short-term
Predictor
Long-term
Predictor
Excitation
Generator
Local Decoder
Decoder
Pulse Positions
+ Amplitudes
D
E
M
U
X
Excitation
Generator
Long-term
Predictor
Short-term
Predictor
ŝ(n)
LTP Parameters
LPC Parameters

Example, MP-LPC



A segment of 200 samples from a speech signal.

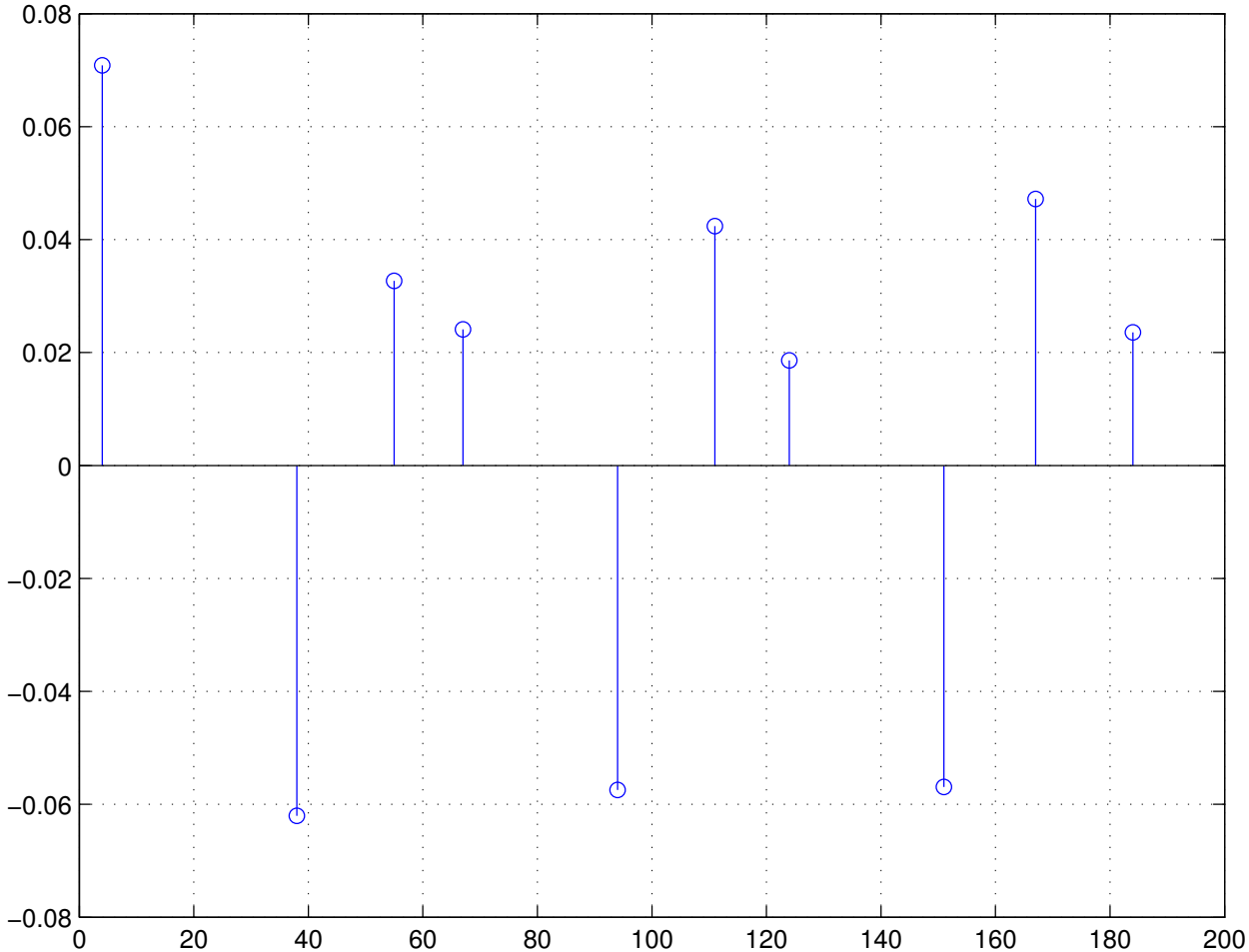
Example, MP-LPC

We adapt a 5 coefficent filter to the signal using the auto correlation method. The filter coefficients (before quantization) are:

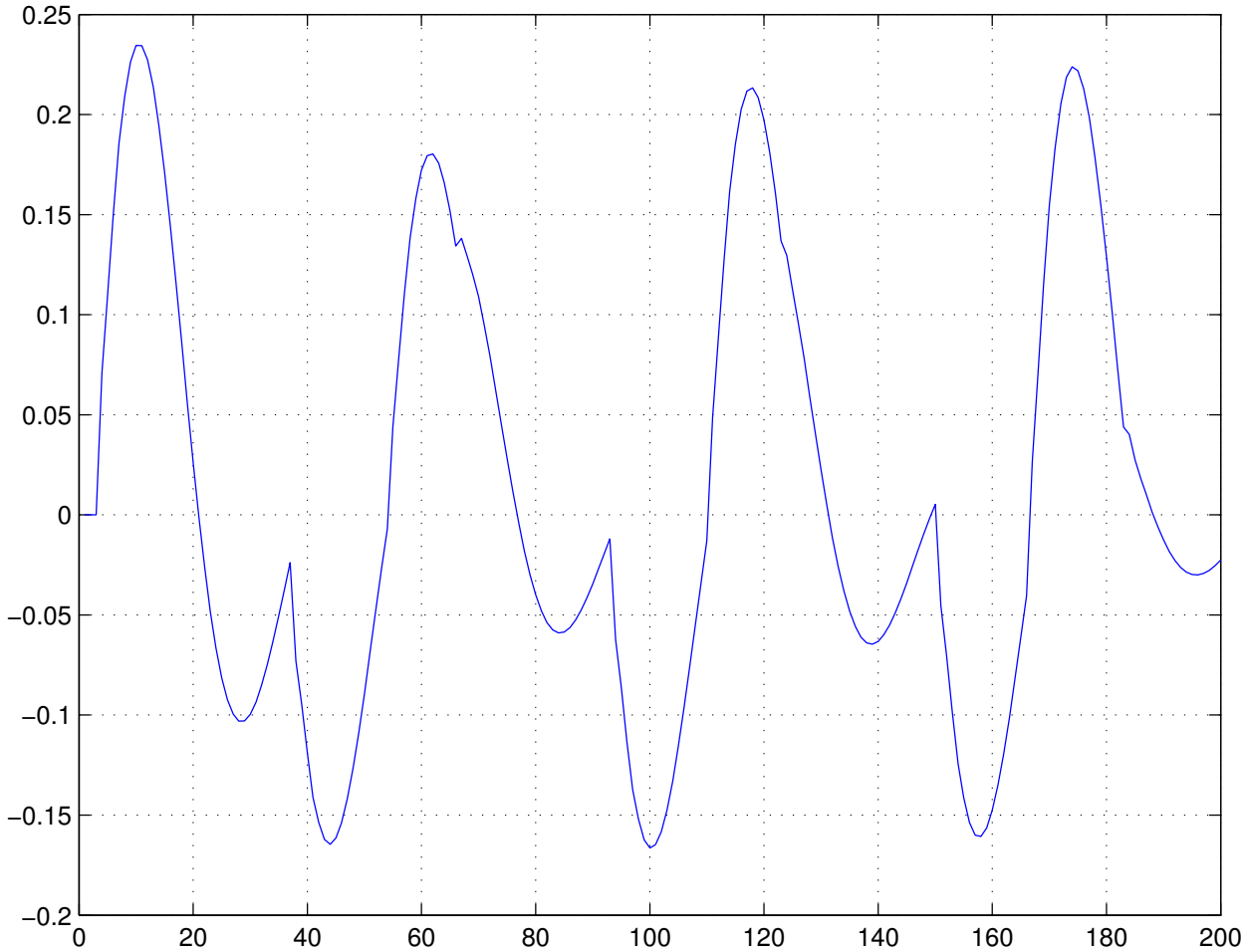
$$\bar{\mathbf{a}} \approx \begin{bmatrix} -1.5373 \\ 0.2515 \\ 0.2400 \\ 0.1754 \\ -0.0912 \end{bmatrix}$$

Example, MP-LPC

A pulse train with ten pulses is optimized so that the decoded signal is as close to the original signal as possible.

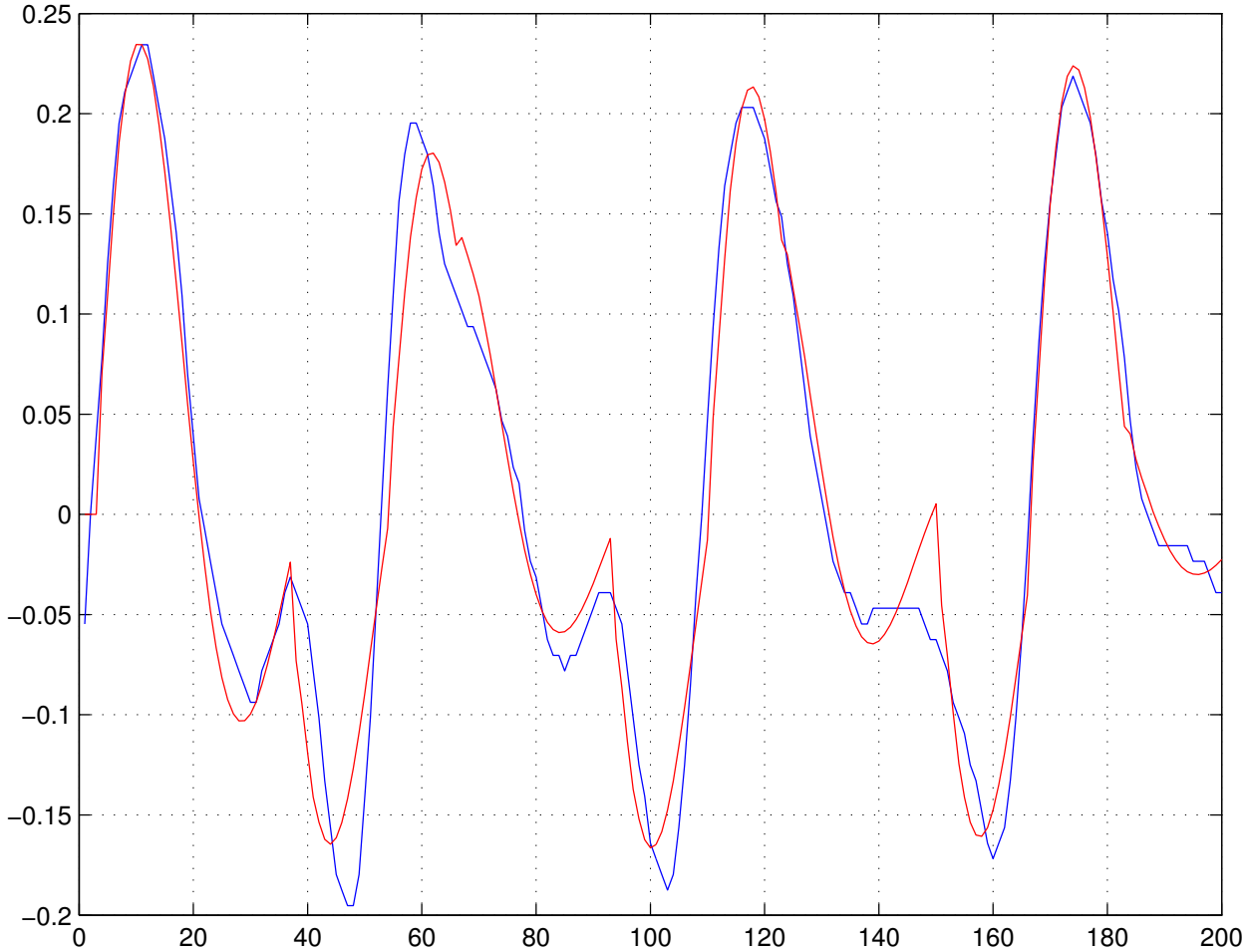


Example, MP-LPC



Decoded signal.

Example, MP-LPC



Original signal and decoded signal.

RPE-LTP

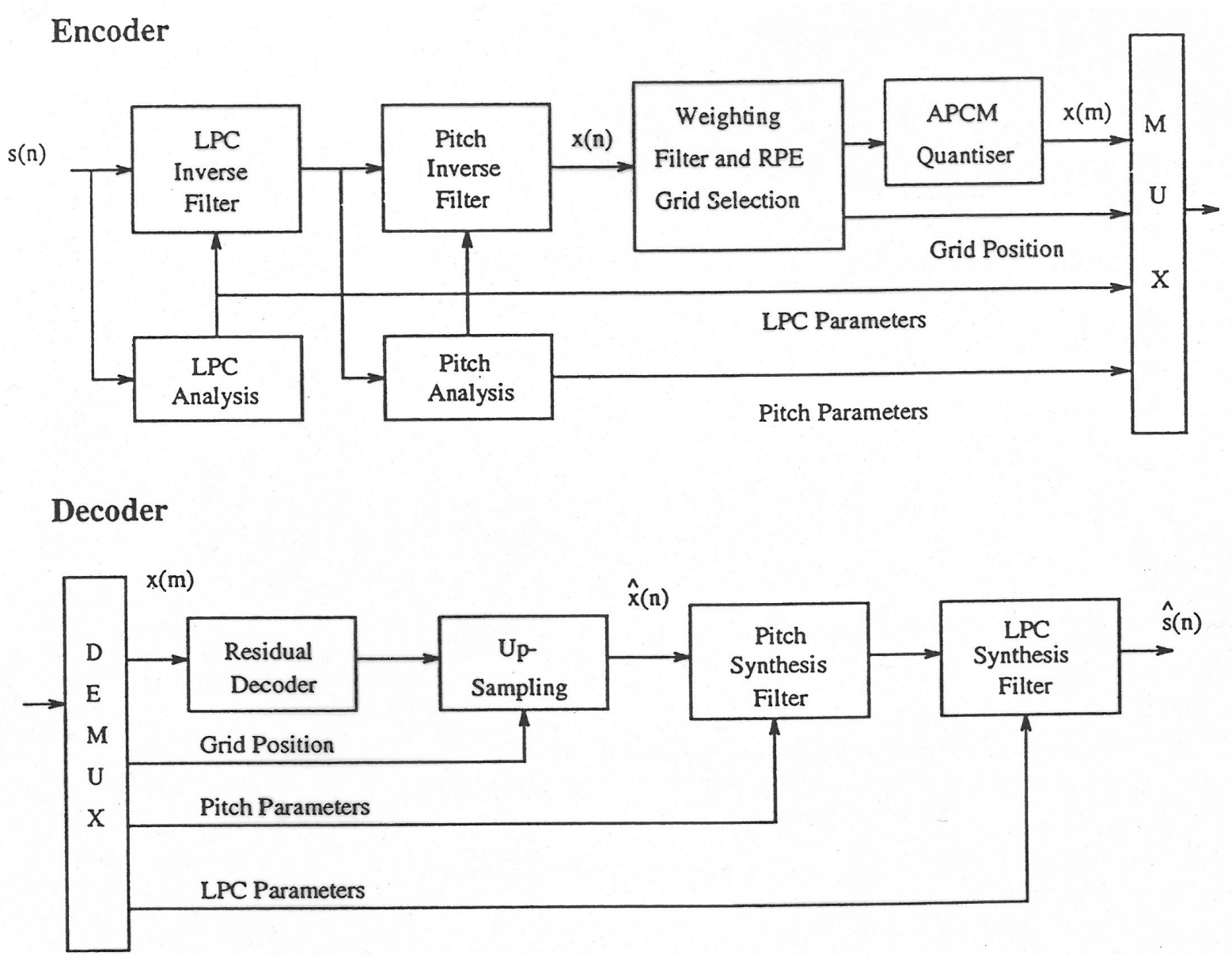
Regular Pulse Excitation with Long Term Prediction

The first coding method used in the GSM system. It has later been replaced by other methods.

Can be seen as a hybrid between RELP and MP-LPC. The coder tries to find a pulse train that is decoded to a signal as close to the original signal as possible. The pulses are limitied to be located in a regular pattern.

The coder uses the rate 13 kbit/s. Including error correction we get the total rate 22.8 kbit/s.

RPE-LTP


Encoder
s(n)
LPC Inverse Filter
Pitch Inverse Filter
x(n)
Weighting Filter and RPE Grid Selection
APCM Quantiser
x(m)
M U X
Grid Position
LPC Parameters
LPC Analysis
Pitch Analysis
Pitch Parameters
Decoder
x(m)
D E M U X
Residual Decoder
Up-Sampling
x̂(n)
Pitch Synthesis Filter
LPC Synthesis Filter
ŝ(n)
Grid Position
Pitch Parameters
LPC Parameters

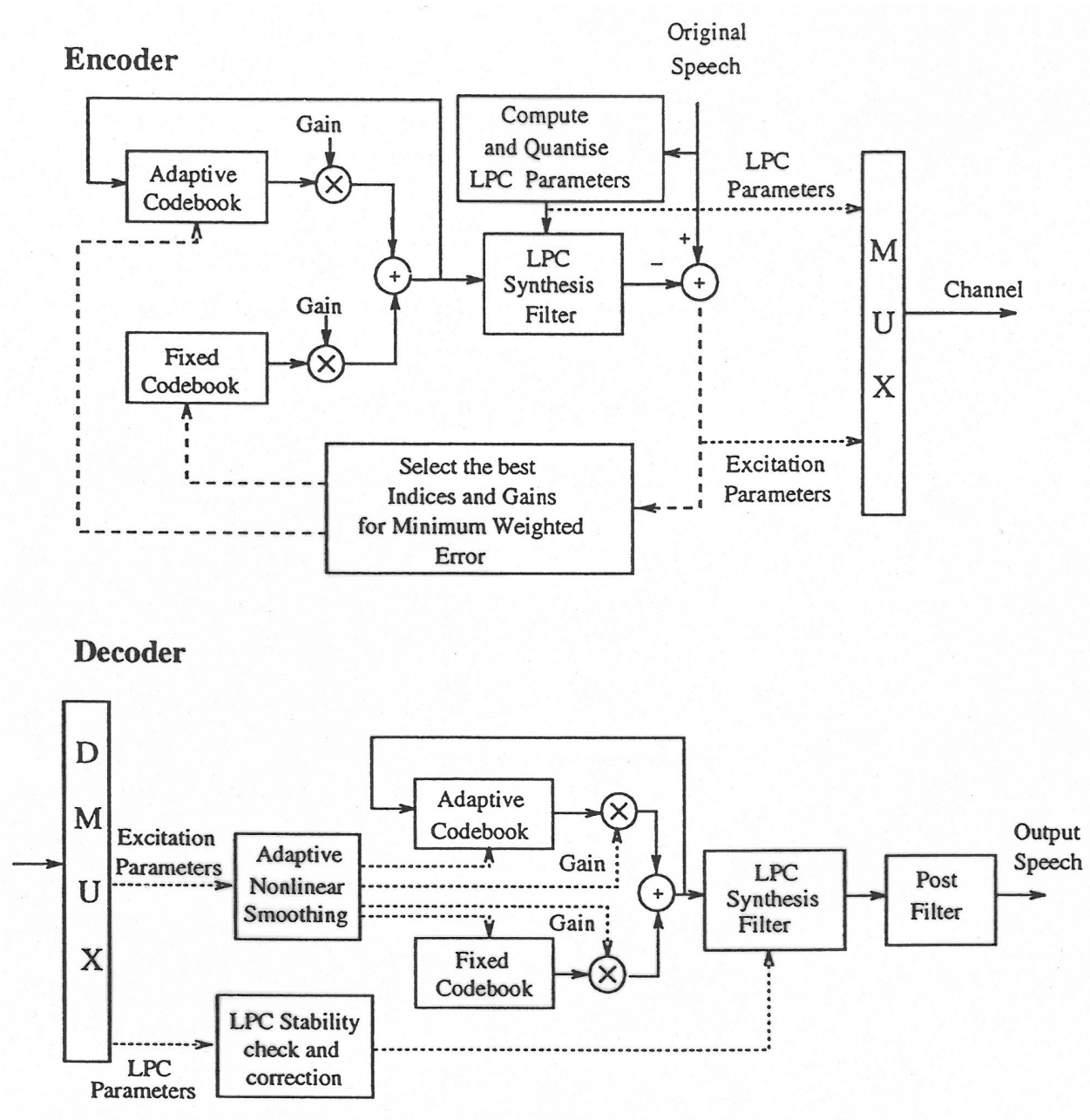
Code Excited Linear Prediction (CELP)

Analysis by synthesis. The coder estimates filter coefficients, and then tries to find an *excitation signal* from a codebook that is decoded to a signal close to the original signal. It is thus a form of vector quantization, often of the type gain-shape. What is sent to the receiver is filter parameters, index in the codebook and gain parameters.

Often a combination of a fixed and an adaptive codebook is used.

There are variants where the filter parameters are estimated using the previous segment. Since the decoder also has access to those old samples, only index data needs to be transmitted.

CELP


Encoder
Original Speech
Gain
Adaptive Codebook
Compute and Quantise LPC Parameters
LPC Parameters
LPC Synthesis Filter
Gain
Fixed Codebook
Channel
M U X
Select the best Indices and Gains for Minimum Weighted Error
Excitation Parameters
Decoder
D M U X
Excitation Parameters
Adaptive Nonlinear Smoothing
Adaptive Codebook
Gain
Gain
Fixed Codebook
LPC Synthesis Filter
Post Filter
Output Speech
LPC Stability check and correction
LPC Parameters

CELP in GSM

Enhanced Full Rate Algebraic CELP
Data rate 12.2 kbit/s

Adaptive Multi-Rate Algebraic CELP
Data rate between 4.75 kbit/s and 12.2 kbit/s (in 8 steps). The coder tries to adapt to the channel quality. If the channel is bad the speech coder will use a low rate and then many bits are used for error correction. For better channels not as many bits are needed for error correction and the speech coder can then use a higher rate.

The channel rate is either 22.8 kbit/s or 11.4 kbit/s (half rate channel).