Multimedia Video Coding & Architectures (5LSE0), Module 05

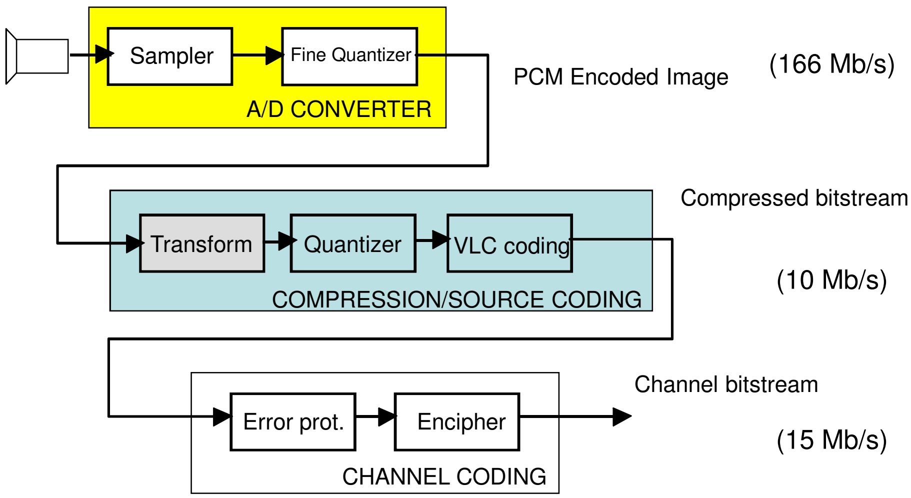
Differential PCM (DPCM) and Linear Predictive Coding (LPC)

Peter H.N. de With
(p.h.n.de.with@tue.nl)

slides version 1.0

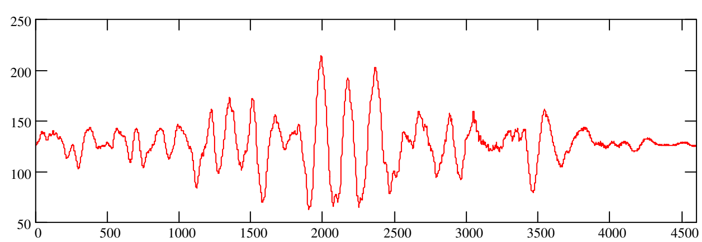
5LSE0 - Mod 05
Part 1
Correlation and Prediction

General Compression System



Correlation in Signals – (1)

* **Meaningful signals are often highly predictable:**
  From the behavior in the past, the future signal values can be approximately estimated



* **(Linear) Predictability has something to do with the autocorrelation function, so ...**

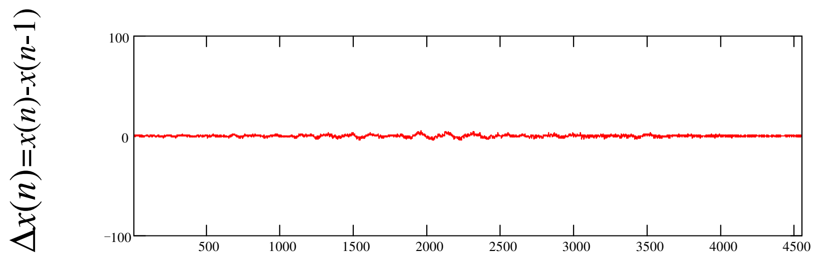
5

Correlation in Signals – (2)

- Differential PCM works on the principle that:

  Anything that can be predicted from the signal's past can be reconstructed by the decoder

- Simple example: Predict next signal value on basis of actual one. Send only differences to the decoder



6

Effect of Prediction on Quantization

- If the same quantizer is used on $x(n)$ and $\Delta x(n)$, then compare:

For $x(n)$ $\quad \sigma_q^2 = \sigma_x^2 \sigma_{q,\text{prototype}}^2$

For $\Delta x(n)$ $\quad \sigma_q^2 = \sigma_{\Delta x}^2 \sigma_{q,\text{prototype}}^2$

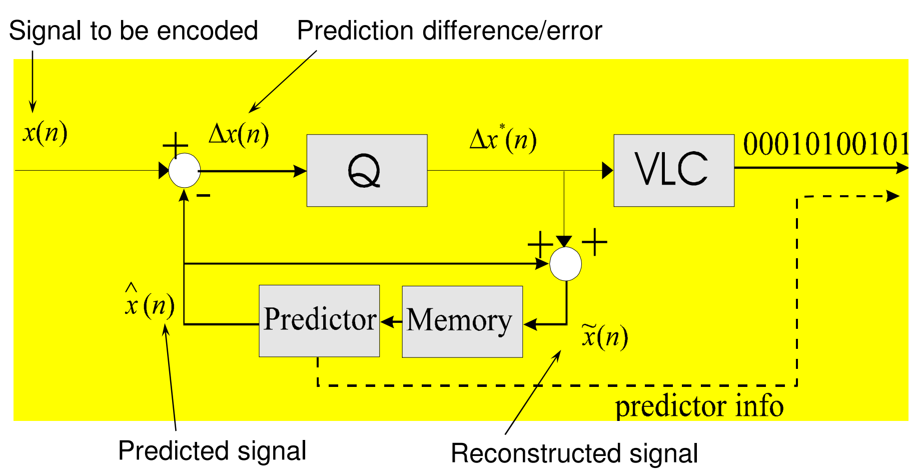
Quality ***scales* with variance** of "signal to be quantized"

Principle of Differential PCM

- **DPCM encoder consists of three steps:**
  - ***Predict*** the current signal value $x(n)$ from past values $x(n\text{-}1)$, $x(n\text{-}2)$, ...
  - ***Quantize*** the difference signal = prediction error
  - ***Encode,*** e.g. VLC the prediction difference (prediction error)
- **DPCM Decoder reproduces same prediction of $x(n)$**
- **Prediction in encoder is based on quantized values**

$$\tilde{x}(n-1), \tilde{x}(n-2), \ldots$$

Diagram of DPCM Encoder & Signals

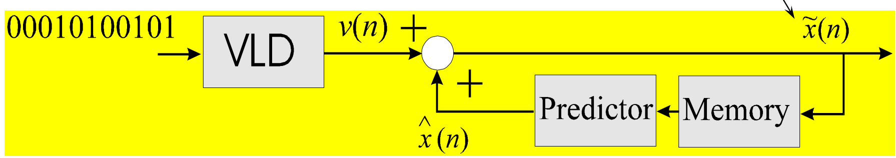


DPCM Decoder Diagram

System aspects 1: Loop

- **Closed loop** in encoder mimics decoder to avoid decoding mismatch (noise inside)

Reconstructed signal



Predicted signal

System aspects 2:

Decoder is typically simpler, decorrelation, inv. Q, no decision making

What Linear Predictor to Use?

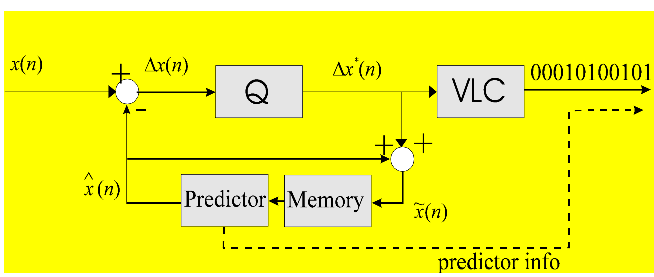
- **What is "best" predictor?**
  - Theoretically: $\hat{x}(n) = E[x(n) | \tilde{x}(n-1), \tilde{x}(n-2), \ldots, \tilde{x}(n-N)]$
  - Practically: Linear predictors only
- **Examples:**
  - PCM $\hat{x}(n) = 0$
  - Simple differences $\hat{x}(n) = \tilde{x}(n-1)$
  - Average last two samples $\hat{x}(n) = h_1 \tilde{x}(n-1) + h_2 \tilde{x}(n-2)$
  - General linear predictor $\hat{x}(n) = \sum_{k=1}^{N} h_k \tilde{x}(n-k)$
- **How to evaluate "success/usefulness" of prediction (compared to PCM): *Prediction Gain***

11

5LSE0 - Mod 05
Part 2
Reconstruction Error, Prediction Gain and Optimal Prediction

12

Reconstruction Error in DPCM – (1)



* **Reconstruction / Coding error:**

$$r(n) = x(n) - \tilde{x}(n)$$
$$= x(n) - \left[\hat{x}(n) + \Delta x^*(n)\right]$$
$$= \Delta x(n) - \Delta x^*(n)$$
$$= q(n)$$

Reconstruction Error in DPCM – (2)

- Overall reconstruction error in DPCM is equal to the quantization error of $Q[\Delta x(n)]$
- DPCM reconstruction error variance: $\sigma_r^2 = \sigma_q^2$
- SNR will be determined by quantizer performance, *relative to the variance of the input signal* $\Delta x(n)$

Rate-Distortion Model

- **Model for relation bit-rate vs. distortion:**
  - Gaussian independent data samples
  - Variance σ_x^2

$$R(D) = \frac{1}{2}\log_2 \frac{\sigma_x^2}{D} \quad \text{(bits/sample)}$$

- **For PCM:** $$R_{PCM} = \frac{1}{2}\log_2 \frac{\sigma_x^2}{\sigma_q^2}$$
- **For DPCM:** $$R_{DPCM} = \frac{1}{2}\log_2 \frac{\sigma_{\Delta x}^2}{\sigma_q^2}$$

(Coding) Gain Obtained by Prediction

* **Theoretical difference in bit rate if *same quantizer* (with same σ_q^2) is used (*same quality of encoded signal*):**

$$R_{PCM} - R_{DPCM} = \frac{1}{2}\log_2 \frac{\sigma_x^2}{\sigma_{\Delta x}^2}$$

Ratio of variances!

* **Advantage of DPCM over PCM depends on the *ratio of signal variance: prediction error variance:***
  - Prediction gain: $G_P = \frac{\sigma_x^2}{\sigma_{\Delta x}^2}$

Prediction Gain Example

Case	G_P	Comments
$\sigma_{\Delta x}^2 < \sigma_x^2$	$G_P > 1$	DPCM outperforms PCM
$\sigma_{\Delta x}^2 = \sigma_x^2$	$G_P = 1$	DPCM identical to PCM
$\sigma_{\Delta x}^2 > \sigma_x^2$	$G_P < 1$	DPCM worse than PCM
$\sigma_{\Delta x}^2 = 0$	$G_P \to \infty$	Perfect prediction

Sometimes the prediction gain is given as: $\frac{1}{2}\log_2 G_P$

- Practical example: G_p=20 => Gain bit rate: 2.1 bits/sample

The Optimal Predictor – (1)

* **Maximal benefit of DPCM is obtained if G_p is large**
  – Make variance of prediction difference $\sigma^2_{\Delta x}$ as **small** as possible

* **Rationale**
  – Make the prediction $\tilde{x}(n)$ as close as possible to $x(n)$ in quadratic sense:
  - Theoretical $\sigma^2_{\Delta x} = E[(x(n) - \tilde{x}(n))^2]$
  - Practical (est.) $\sigma^2_{\Delta x} = \frac{1}{N}\sum_{\text{all } n}(x(n) - \tilde{x}(n))^2$

The Optimal Predictor – (2)

* **Predictor:** $\hat{x}(n) = \sum_{k=1}^{N} h_k x(n-k)$

  Find weights h_k

* **Minimize:** $\sigma^2_{\Delta x} = E[(x(n) - \hat{x}(n))^2]$

  – note that the prediction takes place on original data

* **Solution for N *linear prediction coefficients* h_k**

$$\frac{\partial \sigma^2_{\Delta x}}{\partial h_i} = 0 \quad \Rightarrow \quad R_X(i) = \sum_{k=1}^{N} h_k R_X(i-k) \qquad i = 1, 2, \ldots, N$$

(Normal or Yule-Walker Equations)

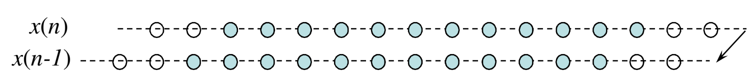
The Optimal Predictor – (3)

$$\begin{bmatrix} R_x(1) \\ R_x(2) \\ \vdots \\ R_x(N) \end{bmatrix} = \begin{bmatrix} R_x(0) & R_x(1) & \cdots & R_x(N-1) \\ R_x(1) & R_x(0) & & R_x(N-2) \\ \vdots & & \ddots & \vdots \\ R_x(N-1) & R_x(N-2) & \cdots & R_x(0) \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_N \end{bmatrix}$$

- **From this set of equations the N linear prediction coefficients can be computed.**
- **Need to know $R_X(k)$: Autocorrelation function of $x(n)$**

Estimation Autocorrelation Function – (1)

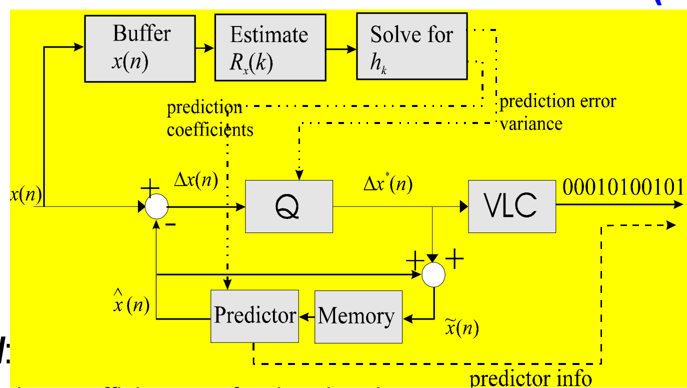
- **Theoretical:** $R_X(k) = E[X(n)X(n-k)]$
- **Practical choice:** $R_X(k) = \frac{1}{N}\sum_{n=0}^{N} X(n)X(n-k)$
- **Practical problems:**
  - Only limited amount of data (especially in ***adaptive*** versions)
  - Boundary effects:



  - Different "estimators" for autocorrelation function exist, with theoretical and practical implications

TU/e PdW / 2017
Fac. EE SPS-VCA
Multimedia Video Coding & A / 5LSE0 /
Module 05 DPCM & Linear Pred. Coding
VCA

Estimation Autocorrelation Function – (2)

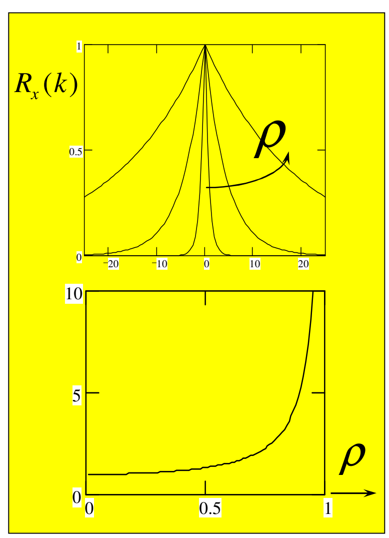


Overhead:

- Prediction coefficients h_k for the signal
- Variance of prediction difference: used to scale the quantizer
- Mean of signal (usually first subtracted)
- Start values for decoding process (boundary values)

Example of Prediction Gain (1-D)

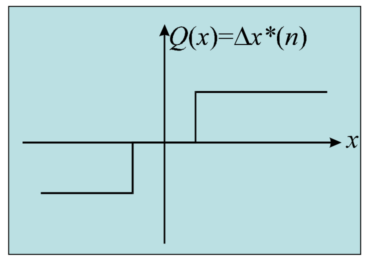
- ∗ **Model:** $R_x(k) = \sigma_x^2 \rho^{|k|}$
- ∗ **Predictor:** $\hat{x}(n) = h_1\, x(n-1)$
- ∗ **"Set" of linear equations:** $R_x(1) = R_x(0) h_1$
- ∗ ***Solution*:** $h_1 = \dfrac{R_X(1)}{R_X(0)} = \rho$
- ∗ **Resulting prediction error variance:** $\sigma_{\Delta x}^2 = \sigma_x^2 (1-\rho^2)$
- ∗ **Prediction gain:** $G_P = \dfrac{1}{1-\rho^2}$



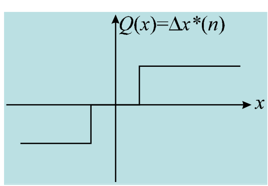
DPCM Artifacts – Coding noise

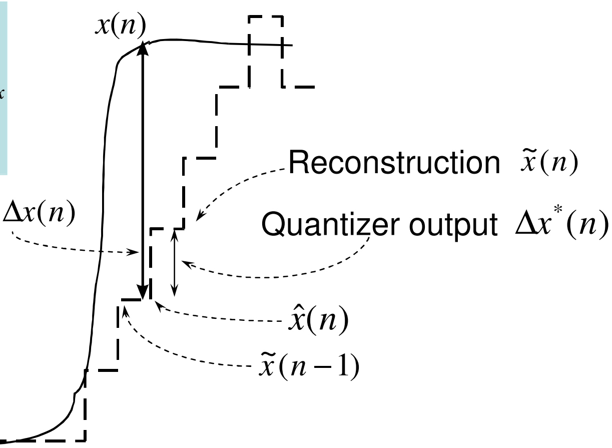
* **Granular noise (like PCM)**
* **Slope overload:**
  - Prediction differences $\Delta x(n)$ too large for quantizer to handle
  - Encoder cannot track rapidly changing signal values (i.e. at signal slopes)
* **Example:**
  - Encoder $\hat{x}(n) = \tilde{x}(n-1)$
  - Decoder $\tilde{x}(n) = \hat{x}(n) + \Delta x^{*}(n)$

$$= \tilde{x}(n-1) + \Delta x^{*}(n)$$



DPCM Slope Overload

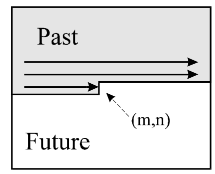




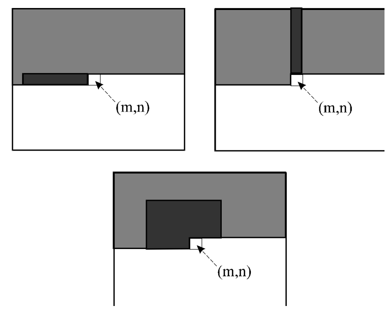
$\hat{x}(n) = \tilde{x}(n-1)$ $\quad$ $\tilde{x}(n) = \tilde{x}(n-1) + \Delta x^{*}(n)$

DPCM on Images / Video

* Same principle as 1-D
* Definition of "Past" and "Future" in images:



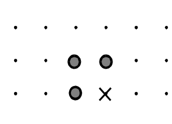
* Predictions:
  – horizontal (scan line)
  – vertical (column)
  – two-dimensional



Linear Predictor for Images

* Typical linear predictor for images:

$$x(m,n) = h_{01}x(m,n-1) + h_{10}x(m-1,n) + h_{11}x(m-1,n-1)$$

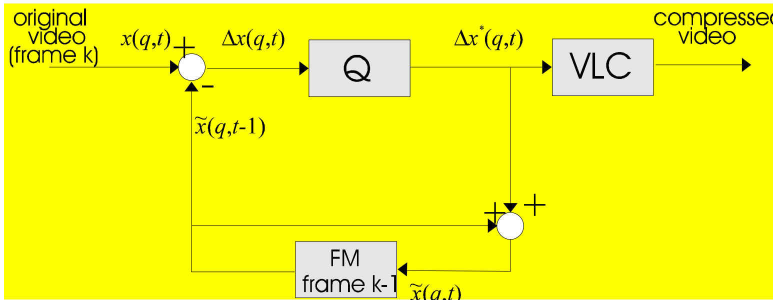


* Yule-Walker for 2-D predictor: Solve for optimal prediction coefficients

Need 2-D correlation function!

$$\begin{bmatrix} R(0,1) \\ R(1,0) \\ R(1,1) \end{bmatrix} = \begin{bmatrix} R(0,0) & R(1,-1) & R(1,0) \\ R(1,-1) & R(0,0) & R(0,1) \\ R(1,0) & R(0,1) & R(0,0) \end{bmatrix} \begin{bmatrix} h_{01} \\ h_{10} \\ h_{11} \end{bmatrix}$$

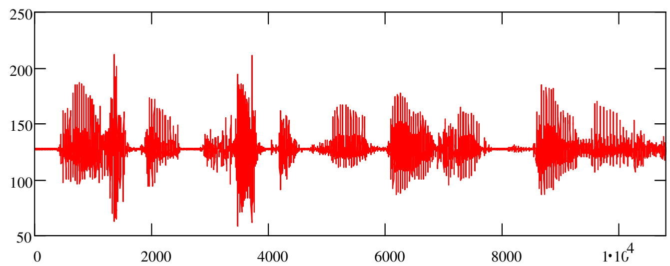
DPCM on Video pictures



* **Usually difference between subblocks in the frames (e.g. 16x16 pixels)**
* **Often fixed simple predictor:** $\hat{x}(q,t) = \tilde{x}(q,t-1)$
  – No need to estimate prediction coefficients

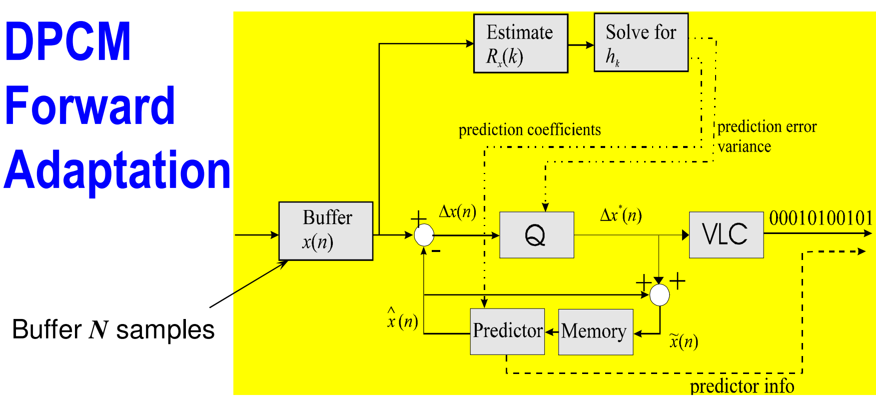
5LSE0 - Mod 05
Part 3
Adaptive Prediction and Systems

Adaptive DPCM systems



- **Apply the same predictor everywhere is not a good idea, because signal correlation changes over time**
  - Adapt predictor (and quantizer) locally
  - Need to estimate autocorrelation function locally
  - Incurs overhead

DPCM Forward Adaptation



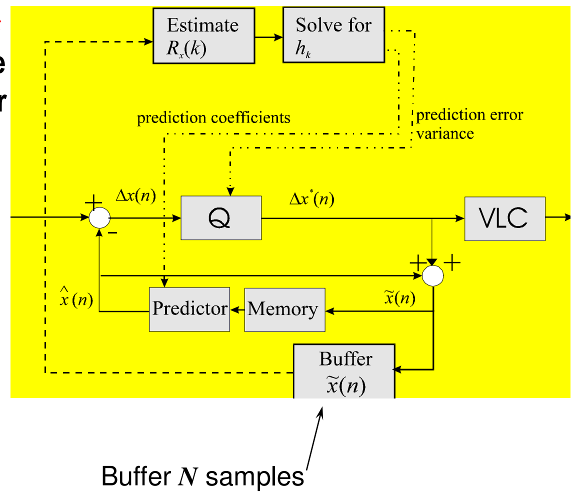
- Effective adaptation needs small segments (e.g. speech: 32 msec, video?)
- Small segments:
  - generate a lot of overhead (prediction coefficients)
  - difficult to estimate autocorrelation reliably
- Make trade-off on basis of prediction gain

31

DPCM with Backward Adaptation

Use information *previous* data segment to calculate prediction coefficients for *current* segment

- Decoder can reproduce the calculation of the prediction coefficients
- *Drawback*: Adaptation may come too late



32

Adaptation by Mode Selection

- **In images, simplify the adaptation by**
  - only allowing to choose among few predictors
  - limiting the incurred overhead (e.g. per 8x8 block)
- **Per block, the "best" predictor is selected**
  - "Best" means: largest prediction gain
- **Example: Three predictors among which to choose**

$$\tilde{x}(m,n) = \hat{x}(m-1,n)$$ "horizontal edge"

$$\tilde{x}(m,n) = \hat{x}(m,n-1)$$ "vertical edge"

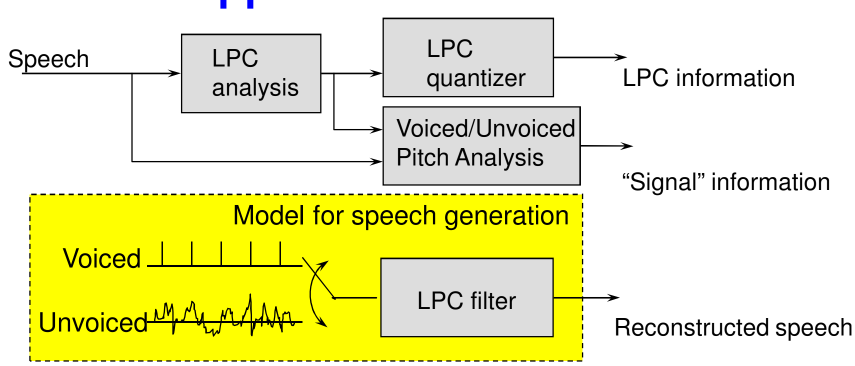
$$\tilde{x}(m,n) = 0.4\hat{x}(m-1,n) + 0.4\hat{x}(m,n-1) + 0.2\hat{x}(m-1,n-1)$$

no directional preference

Application: Speech Production and LPC

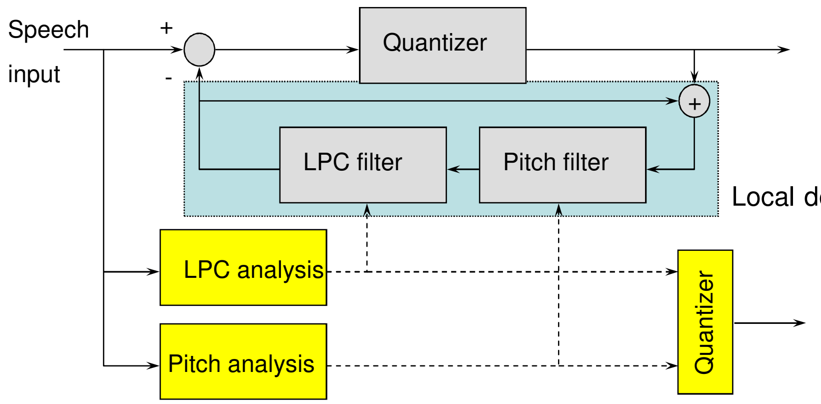
* **Human speech is produced in the vocal tract as combination of vocal cords (in glottis) interacting with articulators in vocal tract**
  – Model this as a signal processing process
* **Human speech perception exhibit masking**
  – Model this as a signal processing (masking) filter
* **Results in DPCM, like speech coding:**
  – LPC Vocoder
  – Linear Predictive Coding (LPC)
  – Analysis-by-synthesis coding (CELP)

Application / LPC Vocoder



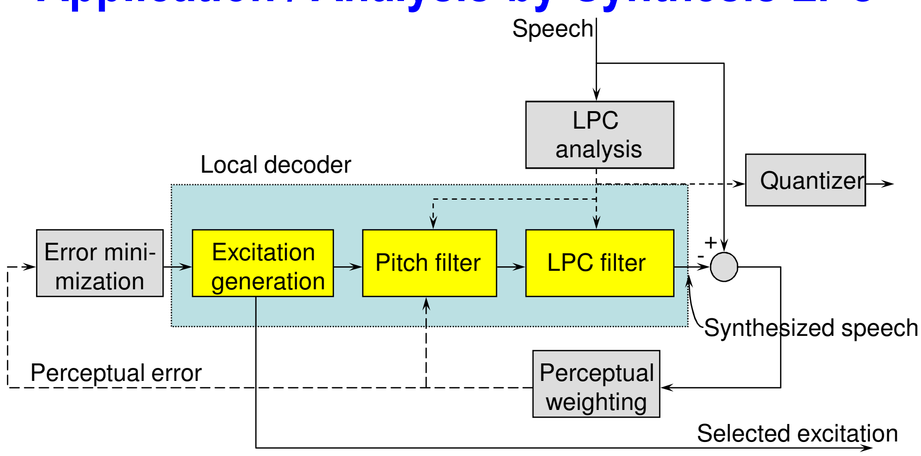
* **Voice Coder (VoCoder) analyzes:**
  – Spectral properties of speech segment
  – Voice/Unvoiced segment information
  – Pitch of signal
* **Objective: Understandable speech**

Application / Adaptive LPC (APC Coder)



- Very similar to (adaptive) DPCM
- Objective: Waveform precise reproduction

Application / Analysis-by-Synthesis LPC



- Marriage between vocoder and APC/DPCM
- *CELP*: Codebook-excited linear prediction

37

Standard Speech Coders using DPCM/LPC

Standard number	ITU G.711	ITU G.726	ITU G.728	ETSI	ITU G.723.1
Year	1972	1990	1992/1994	1994	1995
Type of Coder	**Companded PCM**	**ADPCM**	**LD-CELP**	**VSELP**	**ACELP ++**
Bit rate	**64 kbs**	**16-40 kbs**	**16 kbs**	**5.6 kbs**	**5.3 & 6.3 kbs**
Quality	Toll	< Toll	Toll	GSM	< Toll
Delay	0.125 msec	0.125 msec	0.625 msec	20 msec	30 msec

38

Concluding statements

- **Original DPCM system encoder is a feedback loop**
  - Decoder reconstructs the same signal without side information
- **Adaptive prediction and/or quantization improves SNR**
  - Adaptive prediction to image structure
  - Adaptive quantization for reducing noise in specific areas
- **Model-based prediction such as synthesis enforce decorrelation, but increase the complexity of that stage**
- **Buffering approach: trade-off complexity & performance**
  - Forward: accurate but expensive overhead and backward v.v.
- **Predictive systems often embedded into larger systems**