Predictive Coding II
Combining prediction and quantization

- Requires simultaneous construction of predictor at encoder and decoder
- Use reconstructed samples for forming predictor
DPCM Structure

Red-drawing yields block diagram with typical DPCM structure

- $u'_n$ — Quantized prediction residual (signal to be transmitted)
- $s'_n$ — Reconstructed signal
DPCM Encoder and Decoder

- Split quantizer into encoder mapping $\alpha$ and decoder mapping $\beta$
- Add entropy coding $\gamma$

Note: Encoder contains decoder (except entropy decoding $\gamma^{-1}$)
DPCM: Combine Prediction and Quantization

- Consider zero-mean input signal
- Prediction \( \hat{S}_n \) for a sample \( S_n \) is generated by linear filtering of \( K \) reconstructed samples \( S'_n \) from the past

\[
\hat{S}_n = \sum_{k=1}^{K} h_k S'_{n-k} = \sum_{k=1}^{K} h_k (S_{n-k} + Q_{n-k}) = h^T (S_{n-1} + Q_{n-1}) \tag{1}
\]

with \( Q_n = S'_n - S_n \) being the quantization error signal
- Prediction error variance (for zero-mean input) is given by

\[
\sigma_U^2 = E\{ U_n^2 \} = E\{ (S_n - \hat{S}_n)^2 \} = E\{ (S_n - h^T S_{n-1} - h^T Q_{n-1})^2 \} = E\{ S_n^2 \} + h^T E\{ S_{n-1} S_{n-1}^T \} h + h^T E\{ Q_{n-1} Q_{n-1}^T \} h - 2 h^T E\{ S_n S_{n-1} \} - 2 h^T E\{ S_n Q_{n-1} \} + 2 h^T E\{ S_{n-1} Q_{n-1}^T \} h \tag{2}
\]
DPCM: Combine Prediction and Quantization

Continue: Prediction error variance

\[ \sigma_U^2 = \mathbb{E}\{ S_n^2 \} + h^T \mathbb{E}\{ S_{n-1} S_{n-1}^T \} h + h^T \mathbb{E}\{ Q_{n-1} Q_{n-1}^T \} h \]
\[ -2h^T \mathbb{E}\{ S_n S_{n-1} \} - 2h^T \mathbb{E}\{ S_n Q_{n-1} \} + 2h^T \mathbb{E}\{ S_{n-1} Q_{n-1}^T \} h \]
\[ \sigma_U^2 = \sigma_S^2 - 2h^T c_n + h^T C_K h + \]
\[ h^T \mathbb{E}\{ Q_{n-1} Q_{n-1}^T \} h - 2h^T \mathbb{E}\{ S_n Q_{n-1} \} + 2h^T \mathbb{E}\{ S_{n-1} Q_{n-1}^T \} h \]  

Optimal prediction coefficients \( h \) depend on

- Statistical properties of input signal (\( C_K \) and \( c_n \))
- Statistical properties of quantization error signal

Optimal quantizer design depends on statistical properties of prediction error

Iterative numerical optimization of quantizer and predictor design
DPCM for Gauss-Markov Source

Estimate distortion-rate function for DPCM of Gauss-Markov Source

- Consider zero-mean Gauss-Markov process
  \[ S_n = Z_n + \varrho \cdot S_{n-1} \]  
  \( (4) \)

- Consider a one-tap linear prediction filter \( h \)

- Auto-covariance matrix and cross-covariance vector
  \[ C_1 = \sigma_S^2 \quad \text{and} \quad c_n = \sigma_S^2 \varrho \]  
  \( (5) \)

- Prediction error variance
  \[
  \sigma_U^2 = \sigma_S^2 - 2h\sigma_S^2\varrho + h^2\sigma_S^2 + h^2E\{ Q_{n-1}^2 \} \\
  -2hE\{ S_n Q_{n-1} \} + 2h^2E\{ S_{n-1} Q_{n-1} \} \\
  = \sigma_S^2(1 - 2\varrho + h^2) + h^2E\{ Q_{n-1}^2 \} \\
  -2hE\{ Z_n Q_{n-1} \} - 2h\varrho E\{ S_{n-1} Q_{n-1} \} + 2h^2E\{ S_{n-1} Q_{n-1} \} \\
  = \sigma_S^2(1 - 2\varrho + h^2) + h^2E\{ Q_{n-1}^2 \} + 2h(h - \varrho)E\{ S_{n-1} Q_{n-1} \} 
  \]  
  \( (6) \)
DPCM for Gauss-Markov Source

- **Prediction error variance**

\[
\sigma_U^2 = \sigma_S^2 (1 - 2h\varrho + h^2) + h^2 \mathbb{E}\{ Q_{n-1}^2 \} + 2h(h - \varrho)\mathbb{E}\{ S_{n-1}Q_{n-1} \}
\]  

(7)

- Consider optimal filter for high rates: \( h = \varrho \)

\[
\sigma_U^2 = \sigma_S^2 (1 - \varrho^2) + \varrho^2 D
\]  

(8)

- **Distortion-rate function of scalar quantizer**

\[
D(R) = \sigma_U^2 \cdot g(R)
\]  

(9)

- **Resulting prediction error variance**

\[
\sigma_U^2 = \sigma_S^2 (1 - \varrho^2) + \varrho^2 \sigma_U^2 \cdot g(R)
\]

\[ \Rightarrow \sigma_U^2 = \sigma_S^2 \cdot \frac{1 - \varrho^2}{1 - \varrho^2 g(R)} \]  

(10)
DPCM for Gauss-Markov Source

Upper bound for distortion-rate function

- Consider filter $h = \varrho$ (optimal filter for high rates)

  ➤ Prediction error variance

  \[
  \sigma_U^2 = \sigma_S^2 \cdot \frac{1 - \varrho^2}{1 - \varrho^2 g(R)} \tag{11}
  \]

  ➤ Operational distortion-rate function

  \[
  D(R) = \sigma_U^2 \cdot g(R)
  = \sigma_S^2 \cdot \frac{1 - \varrho^2}{1 - \varrho^2 g(R)} \cdot g(R) \tag{12}
  \]

  ➤ Operational distortion-rate function at high rates

  \[
  D(R) = \varepsilon^2 \cdot \sigma_S^2 \cdot (1 - \varrho^2) \cdot 2^{-2R} \tag{13}
  \]
Joint Design of Predictor and Quantizer

Initialization

- Choose sufficiently large training set \( \{ s_n \} \) and a Lagrange multiplier \( \lambda \)
- Design optimal predictor \( h \) for \( s_n \) given original samples \( s_{n-1}, s_{n-2}, \cdots \)

Iterative design of predictor and quantizer

1. Design quantizer for predictor \( h \)
   - a. Generate training set \( \{ u_n \} \) by conducting DPCM encoding of training set \( \{ s_n \} \) with predictor \( h \) and current quantizer \((\alpha, \beta, \gamma)\)
   - b. Design quantizer \((\alpha, \beta, \gamma)\) for training set \( \{ u_n \} \) and given \( \lambda \)
   - c. Repeat previous two steps until convergence

2. Generate reconstructed samples \( \{ s'_n \} \) by conducting DPCM encoding of training set \( \{ s_n \} \) with predictor \( h \) and quantizer \((\alpha, \beta, \gamma)\)

3. Design optimal predictor \( h \) for \( s_n \) given reconstructed samples \( s'_{n-1}, s'_{n-2}, \cdots \)

4. Repeat previous three steps until convergence
Experiment: DPCM of Gauss-Markov Source ($\varrho = 0.9$)

→ Prediction error variance $\sigma^2_U$ depends on bit rate

$$
\frac{\sigma^2_U(R)}{\sigma^2_S} = \sigma^2_S \cdot \frac{1 - \varrho^2}{1 - \varrho^2 g(R)}
$$

$$
\sigma^2_U(\infty) = \sigma^2_S \cdot (1 - \varrho^2)
$$
Experiment: DPCM of Gauss-Markov Source ($\varrho = 0.9$)

- **High rates:** Shape and memory gain are achievable
- **Low rates:** Worse than transform coding

The distortion-rate function $D(R)$ is given by $D(R) = \sigma_u^2(R) g(R)$, where $\sigma_u^2(R)$ is the space-filling gain, equal to $1.53$ dB.

The distortion $G_P = 7.21$ dB is achieved for $R = 4$ bits/sample.

High rates shape and memory gain are achievable.

Low rates are worse than transform coding.
Adaptive DPCM

Audio signals, images, videos
- Instationary signals
- Single predictor not suitable for entire signal
- Adapt predictor (incl. observation set) during encoding/decoding

Backward adaptation
- Simultaneously estimate predictor at encoder and decoder side
- Estimation has to be based on reconstructed samples
- No additional rate, but accuracy, complexity, error resilience issues

Forward adaptation
- Analyze signal at encoder side
- Send predictor as part of the bitstream (requires additional bit rate)
- Simple method: Switched predictors
Backward-Adaptive DPCM

\[ s_n + u_n \xrightarrow{\alpha} q_n \xrightarrow{\gamma} b \xrightarrow{\gamma^{-1}} q_n \]

\[ \hat{s}_n + u'_n \xrightarrow{\beta} s'_n \xrightarrow{\gamma^{-1}} q_n \]

Encoder

Decoder
Forward-Adaptive DPCM

\[ s_n \rightarrow + \rightarrow u_n \rightarrow \alpha \rightarrow q_n \rightarrow \gamma \rightarrow b \rightarrow \gamma^{-1} \rightarrow q_n \]

\[ \hat{s}_n \rightarrow + \rightarrow s'_n \rightarrow \beta \rightarrow u'_n \rightarrow \beta \rightarrow u'_n \rightarrow \hat{s}_n \rightarrow + \rightarrow s'_n \rightarrow s_n \]

encoder

FAP encoder

FAP decoder

decoder
Prediction in Video Coding

Intra-picture prediction (forward adaptive)
- Prediction of a block of samples using neighboring reconstructed samples
- Block of residual samples is typically coded using transform coding
- One of multiple available predictors is used (signaled in bitstream)

Motion-compensated prediction (forward adaptive)
- Prediction of a block of samples using already reconstructed pictures
- Motion vector indicates observation set as well as prediction filter $h$
- Block of residual samples is typically coded using transform coding
- Encoder: Motion search for finding suitable prediction block

Motion vector prediction (fixed or forward adaptive)
- Prediction of motion vectors before lossless coding
- Use motion vectors of neighboring blocks for predicting current motion vector
- Only difference vector is transmitted
Intra-Picture Prediction

Example: 9 intra prediction modes in H.264/AVC
Motion-Compensated Prediction

- Predict current block using a displaced block in an already coded picture
  \[ \hat{s}[x, y] = s'_{\text{ref}}[x + m_x, y + m_y] \]

- Displacement is characterized by displacement vector or motion vector
  \[ m = [m_x, m_y]^T \]

- Estimate suitable motion vector in encoder \( \rightarrow \) **Motion estimation**
- Transmit motion vector to decoder as part of the bitstream
Motion Vector Prediction

Example: Median prediction

- Component-wise median of three neighboring blocks

\[
\hat{m}_x = \text{median}(m_x^A, m_x^B, m_x^C)
\]

\[
\hat{m}_y = \text{median}(m_y^A, m_y^B, m_y^C)
\]

→ Coding of prediction error (lossless)

\[
\Delta m_x = m_x - \hat{m}_x
\]

\[
\Delta m_y = m_y - \hat{m}_y
\]
Hybrid Video Encoder

input video pictures
(in coding order and partitioned into blocks)

encoder control

transform & quantization

scaling & inv. transform

entropy coding

bitstream

coding modes, intra pred. modes, motion data

buffer for current picture

reconstructed video
(for monitoring)

decoded picture buffer

transform coefficient levels and quantization parameters

coding mode decision

in-loop filtering

motion estimation

intra mode decision

intra-picture prediction

motion-comp. prediction

s_r'[x + m_x, y + m_y]

f(s_k[x, y])

s_k'[x, y]
Hybrid Video Decoder

- **bitstream**
  - entropy decoding
  - transform coefficient levels and quantization parameters
  - scaling & inv. transform
  - intra-picture prediction
  - motion-comp. prediction

- **transform coefficient levels and quantization parameters**
  - intra-prediction prediction
  - in-loop filtering
  - decoded picture buffer
  - decoded video

- **coding modes**
  - intra prediction modes
  - motion data

- **decoded video**
  - buffer for current picture
  - decoded video
Part Summary

Prediction
- Estimate random variable from already observed random variables
- Optimal predictor: Conditional mean

Linear and affine prediction
- Simple and efficient structure
- Optimal predictor given by Yule-Walker equations
- AR($m$) processes: Optimal predictor has $m$ coefficients
- Optimal prediction error is orthogonal to input signal
- Non-matched predictor can increase signal variance

Predictive quantization: DPCM
- Combination of affine prediction and ECSQ is simple and efficient
- Can exploit linear dependencies between samples
- Forward and backward adaptation