The better the future of a random process is predicted from the past and the more redundancy the signal contains, the less new information is contributed by each successive observation of the process.

Predictive coding idea:
1. Predict the current sample/vector using an estimate which is a function of past samples/vectors of the input signal.
2. Quantize residual between input signal and its prediction.
3. Add quantizer residual and prediction to obtain decoded sample.

How to obtain the predictor $\hat{S}_n$?

How to combine predictor and quantizer?
Prediction

Linear Prediction

Differential Pulse Code Modulation (DPCM)

Adaptive Differential Pulse Code Modulation (ADPCM)
  - Forward Adaptive DPCM
  - Backward Adaptive DPCM
  - Gradient Descent and LMS Algorithm

Transmission Errors in DPCM
Prediction

- Statistical estimation procedure: value of random variable $S_n$ of random process $\{S_n\}$ is estimated using values of other random variables of the random process.
- Set of observed random variables: $B_n$.
- Typical example: $N$ random variables that precede $S_n$.

\[ B_n = \{S_{n-1}, S_{n-2}, \ldots, S_{n-N}\} \tag{1} \]

\[ S_n \rightarrow \text{Predictor} \rightarrow \hat{S}_n \]

\[ S_n \leftarrow \hat{S}_n \rightarrow U_n \]

- Predictor for $S_n$: deterministic function of observation set $B_n$.

\[ \hat{S}_n = A_n(B_n) \tag{2} \]

- Prediction error.

\[ U_n = S_n - \hat{S}_n = S_n - A_n(B_n) \tag{3} \]
Prediction Performance

- Defining MSE distortion using $u_i = s_i - \hat{s}_i$ and $s'_i = u'_i + \hat{s}_i$

$$d_N(s, s') = \frac{1}{N} \sum_{i=0}^{N-1} (s_i - s'_i)^2 = \frac{1}{N} \sum_{i=0}^{N-1} (u_i + \hat{s}_i - u'_i - \hat{s}_i)^2 = d_N(u, u') \quad (4)$$

- Operational distortion rate function of a predictive coding systems is equal to the operational distortion rate function for scalar quantization of the prediction residuals

- Operational distortion rate function for scalar quant.: $D(R) = \sigma_U^2 \cdot g(R)$
  - $\sigma_U^2$: the variance of the residuals
  - $g(R)$: depends only on the type of the distribution of the residuals

- Neglect the dependency on the distribution type

- Define: predictor $A_n(B_n)$ given an observation set $B_n$ is optimal if it minimizes variance $\sigma_U^2$

- Assume stationary processes: $A_n(\cdot)$ becomes $A(\cdot)$
Optimal Prediction

- Optimization criterion used in literature

\[
\epsilon_U^2 = E \{ U_n^2 \} = E \left\{ (S_n - \hat{S}_n)^2 \right\} = E \left\{ (S_n - A_n(B_n))^2 \right\} \tag{5}
\]

- Minimization of second moment

\[
\begin{align*}
\epsilon_U^2 & = E \left\{ (U_n - \mu_U + \mu_U)^2 \right\} \\
& = E \left\{ (U_n - \mu_U)^2 \right\} + 2E \left\{ (U_n - \mu_U)\mu_U \right\} + E \left\{ \mu_U^2 \right\} \\
& = \sigma_U^2 + \mu_U^2 + 2\mu_U (E \{ U_n \} - \mu_U) \\
& = \sigma_U^2 + \mu_U^2 \tag{6}
\end{align*}
\]

implies minimization of variance \( \sigma_U^2 \) and mean \( \mu_U \)

- Solution: conditional mean

\[
\hat{S}_n^* = A^*(B_n) = E \{ S_n | B_n \} \tag{7}
\]

- Proof: see [Wiegand and Schwarz, 2011, p. 150]
Optimal Prediction for Autoregressive Processes

- Autoregressive process of order $m$ (AR($m$) process)

\[
S_n = Z_n + \mu_S + \sum_{i=1}^{m} a_i \cdot (S_{n-i} - \mu_S)
\]

\[
= Z_n + \mu_S \cdot (1 - a_T^m e_m) + a_T^m S_{n-1}^{(m)}
\]  \hspace{1cm} (8)

where

- $\{Z_n\}$ is a zero-mean iid process
- $\mu_S$ is the mean of the AR($m$) process
- $a_m = (a_1, \cdots, a_m)^T$ is a constant parameter vector
- $e_m = (1, \cdots, 1)^T$ is an $m$-dimensional unit vector

- Prediction of $S_n$ given the vector $S_{n-1} = (S_{n-1}, \cdots, S_{n-N})$ with $N \geq m$

\[
E \{S_n \mid S_{n-1}\} = E \{Z_n + \mu_S (1 - a_N^T e_N) + a_N^T S_{n-1} \mid S_{n-1}\}
\]

\[
= \mu_S (1 - a_N^T e_N) + a_N^T S_{n-1}
\]  \hspace{1cm} (9)

where $a_N = (a_1, \cdots, a_m, 0, \cdots, 0)^T$
Affine Prediction

- Affine predictor
  \[ \hat{S}_n = A(S_{n-k}) = h_0 + h_N^T S_{n-k} \]  
  \[ (10) \]
  where \( h_N = (h_1, \cdots, h_N)^T \) is a constant vector and \( h_0 \) a constant offset

- Variance \( \sigma^2_U \) of prediction residual only depends on \( h_N \)
  
  \[ \sigma^2_U(h_0, h_N) = E \left\{ (U_n - E\{U_n\})^2 \right\} \]
  
  \[ = E \left\{ (S_n - h_0 - h_N^T S_{n-k} - E\{S_n - h_0 - h_N^T S_{n-k}\})^2 \right\} \]
  
  \[ = E \left\{ (S_n - E\{S_n\} - h_N^T (S_{n-k} - E\{S_{n-k}\}))^2 \right\} \]  
  \[ (11) \]

- Mean squared prediction error
  
  \[ \epsilon^2_U(h_0, h_N) = \sigma^2_U(h_N) + \mu^2_U(h_0, h_N) = \sigma^2_U(h_N) + E \left\{ S_n - h_0 - h_N^T S_{n-k} \right\}^2 \]
  
  \[ = \sigma^2_U(h_N) + (\mu_S (1 - h_N^T e_N) - h_0)^2 \]  
  \[ (12) \]

- Minimize mean squared prediction error by setting
  
  \[ h_0^* = \mu_S (1 - h_N^T e_N) \]  
  \[ (13) \]
The function used for prediction is linear, of the form

\[ \hat{S}_n = h_1 \cdot S_{n-1} + h_2 \cdot S_{n-2} + \cdots + h_N \cdot S_{n-N} = h_N^T S_{n-1} \] (14)

Mean squared prediction error

\[
\sigma^2_U(h_N) = E \left\{ (S_n - \hat{S}_n)^2 \right\} = E \left\{ (S_n - h_N^T S_{n-1})(S_n - S_{n-1}^T h_N) \right\}
\]
\[
= E \left\{ S_n^2 \right\} - 2E \left\{ h_N^T S_{n-1} S_n \right\} + E \left\{ h_N^T S_{n-1} S_{n-1}^T h_N \right\}
\]
\[
= E \left\{ S_n^2 \right\} - 2h_N^T E \left\{ S_n S_{n-1} \right\} + h_N^T E \left\{ S_{n-1} S_{n-1}^T \right\} h_N
\] (15)

since \( h_N \) is not a random variable.
Auto-Covariance Matrix and Auto-Covariance Vector

- Variance $\sigma_S^2 = E \{ S_n^2 \}$.
- Auto-covariance vector (for zero mean: auto-correlation vector)

$$c_k = E \{ S_n S_{n-k} \} = \sigma_S^2 \cdot \begin{pmatrix} \rho_k \\ \vdots \\ \rho_i \\ \vdots \\ \rho_N + k - 1 \end{pmatrix}$$

with $\rho_i = E \{ S_n \cdot S_{n-i} \} / \sigma_S^2$ (16)

- Auto-covariance matrix (for zero-mean: auto-correlation matrix)

$$C_N = E \{ S_{n-1} S_{n-1}^T \} = \sigma_S^2 \cdot \begin{pmatrix} 1 & \rho_1 & \rho_2 & \cdots & \rho_{N-1} \\ \rho_1 & 1 & \rho_1 & \cdots & \rho_{N-2} \\ \rho_2 & \rho_1 & 1 & \cdots & \rho_{N-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \rho_{N-1} & \rho_{N-2} & \rho_{N-3} & \cdots & 1 \end{pmatrix}$$ (17)
Optimal Linear Prediction

- Prediction error variance

\[ \sigma^2_U(h_N) = \sigma^2_S - 2h_N^T c_k + h_N^T C_N h_N \]  
(18)

- Minimization of \( \sigma^2_U(h_N) \) yields a system of linear equations

\[ C_N \cdot h_N = c_k \]  
(19)

- When \( C_N \) is non-singular

\[ h^*_N = C_N^{-1} \cdot c_k \]  
(20)

- Minimum of \( \sigma^2_U(h_N) \) is given as (with \( (C_N^{-1} c_k)^T = c_k^T C_N^{-1} \))

\[
\begin{align*}
\sigma^2_U(h^*_N) &= \sigma^2_S - 2 (h^*_N)^T c_k + (h^*_N)^T C_N h^*_N \\
&= \sigma^2_S - 2 (c_k^T C_N^{-1}) c_k + (c_k^T C_N^{-1}) C_N (C_N^{-1} c_k) \\
&= \sigma^2_S - 2 c_k^T C_N^{-1} c_k + c_k^T C_N^{-1} c_k \\
&= \sigma^2_S - c_k^T C_N^{-1} c_k.
\end{align*}
\]  
(21)

- In optimal prediction, signal variance \( \sigma^2_S \) is reduced by \( c_k^T C_N^{-1} c_k \)
Verification of Optimality

The optimality of the solution can be verified by inserting $h_N = h^*_N + \delta_N$ into

$$\sigma^2_U(h_N) = \sigma^2_S - 2 h^T_N c_k + h^T_N C_N h_N$$

yielding

$$\sigma^2_U(h_N) = \sigma^2_S - 2(h^*_N + \delta_N)^T c_k + (h^*_N + \delta_N)^T C_N (h^*_N + \delta_N)$$

$$= \sigma^2_S - 2(h^*_N)^T c_k - 2 \delta^T_N c_k +$$

$$+ (h^*_N)^T C_N h^*_N + (h^*_N)^T C_N \delta_N + \delta^T_N C_N h^*_N + \delta^T_N C_N \delta_N$$

$$= \sigma^2_U(h^*_N) - 2 \delta^T_N c_k + 2 \delta^T_N C_N h^*_N + \delta^T_N C_N \delta_N$$

$$= \sigma^2_U(h^*_N) + \delta^T_N C_N \delta_N$$

The additional term is always non-negative being equal to 0 only if $h_N = h^*_N$

$$\delta^T_N C_N \delta_N \geq 0$$
The Orthogonality Principle

- Important property for optimal affine predictors

\[
E \{ U_n S_{n-k} \} = E \left\{ (S_n - h_0 - h_N^T S_{n-k}) S_{n-k} \right\} \\
= E \{ S_n S_{n-k} \} - h_0 E \{ S_{n-k} \} - E \left\{ S_{n-k} S_{n-k}^T \right\} h_N \\
= c_k + \mu_S^2 e_N - h_0 \mu_S e_N - (C_N + \mu_S^2 e_N e_N^T) h_N \\
= c_k - C_N h_N + \mu_S e_N (\mu_S (1 - h_N^T e_N) - h_0).
\]  

(25)

- Inserting

\[
h_N^* = C_N^{-1} \cdot c_k \quad \text{and} \quad h_0^* = \mu_S (1 - h_N^T e_N)
\]

(26)

yields

\[
E \{ U_n S_{n-k} \} = 0
\]

(27)
Orthogonality Principle and Geometric Interpretation

- For optimal affine prediction, the prediction residual $U_n$ is uncorrelated with the observation vector $S_{n-k}$

\[ E \{U_n S_{n-k}\} = 0 \]  \hspace{1cm} (28)

- Therefore for optimum affine filter design, prediction error should be orthogonal to input signal

- Approximate a vector $S_0$ by a linear combination of $S_1$ and $S_2$
- Best approximation $\hat{S}_0^*$ is given by projection of $S_0$ onto the plane spanned by $S_1$ and $S_2$
- Error vector $U_0^*$ has minimum length and is orthogonal to the projection
One-Step Prediction I

- Random variable $S_n$ is predicted using the $N$ directly preceding random variables $S_{n-1} = (S_{n-1}, \cdots, S_{n-N})^T$
- Using $\phi_k = E \{ (S_n - E \{S_n\}) (S_{n+k} - E \{S_{n+k}\}) \}$, the normal equations are given as

\[
\begin{bmatrix}
\phi_0 & \phi_1 & \cdots & \phi_{N-1} \\
\phi_1 & \phi_0 & \cdots & \phi_{N-2} \\
\vdots & \vdots & \ddots & \vdots \\
\phi_{N-1} & \phi_{N-2} & \cdots & \phi_0
\end{bmatrix}
\begin{bmatrix}
h_1^N \\
h_2^N \\
\vdots \\
h_N^N
\end{bmatrix} = \begin{bmatrix}
\phi_1 \\
\phi_2 \\
\vdots \\
\phi_N
\end{bmatrix}
\]

(29)

where $h_k^N$ represent elements of $h_N^* = (h_1^N, \cdots, h_N^N)^T$

- Changing the equation to

\[
\begin{bmatrix}
\phi_1 & \phi_0 & \phi_1 & \cdots & \phi_{N-1} \\
\phi_2 & \phi_1 & \phi_0 & \cdots & \phi_{N-2} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
\phi_N & \phi_{N-1} & \phi_{N-2} & \cdots & \phi_0
\end{bmatrix}
\begin{bmatrix}
1 \\
-h_1^N \\
-h_2^N \\
\vdots \\
-h_N^N
\end{bmatrix} = \begin{bmatrix}
0 \\
0 \\
\vdots \\
0
\end{bmatrix}
\]

(30)
Including the prediction error variance for optimal linear prediction using the $N$ preceding samples:

$$\sigma_N^2 = \sigma_S^2 - c_1^T C_N^{-1} c_1 = \sigma_S^2 - c_1^T h_c^* = \phi_0 - h_1^N \phi_1 - h_2^N \phi_2 - \cdots - h_N^N \phi_N$$  \hspace{1cm} (31)

yields and additional row in the matrix

$$C_{N+1} = \begin{bmatrix}
\phi_0 & \phi_1 & \phi_2 & \cdots & \phi_N \\
\phi_1 & \phi_0 & \phi_1 & \cdots & \phi_{N-1} \\
\phi_2 & \phi_1 & \phi_0 & \cdots & \phi_{N-2} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
\phi_N & \phi_{N-1} & \phi_{N-2} & \cdots & \phi_0
\end{bmatrix}$$

$$\begin{bmatrix}
1 \\
-h_1^N \\
-h_2^N \\
\vdots \\
-h_N^N
\end{bmatrix} = \begin{bmatrix}
\sigma_N^2 \\
0 \\
0 \\
\vdots \\
0
\end{bmatrix}$$  \hspace{1cm} (32)

$→$ Augmented normal equation
One-Step Prediction III

- Multiplying both sides of the augmented normal equation with $\mathbf{a}_N^T$
  \[ \sigma_N^2 = \mathbf{a}_N^T \mathbf{C}_{N+1} \mathbf{a}_N \]  
  \( (33) \)

- Combing the equations for 0 to N preceding samples into one matrix equation yields
  \[
  \mathbf{C}_{N+1} \cdot \begin{bmatrix}
  1 & 0 & \cdots & 0 & 0 \\
  -h_1^N & 1 & \ddots & 0 & 0 \\
  -h_2^N & -h_1^{N-1} & \ddots & 0 & 0 \\
  \vdots & \vdots & \ddots & 1 & 0 \\
  -h_N^N & -h_{N-1}^N & \cdots & -h_1^1 & 1 
  \end{bmatrix} = \begin{bmatrix}
  \sigma_N^2 & X & \cdots & X & X \\
  0 & \sigma_{N-1}^2 & \ddots & X & X \\
  0 & 0 & \ddots & X & X \\
  \vdots & \vdots & \ddots & \sigma_1^2 & X \\
  0 & 0 & \cdots & 0 & \sigma_0^2 
  \end{bmatrix}
  \]

- Taking the determinant of both sides of the equation gives
  \[ |\mathbf{C}_{N+1}| = \sigma_N^2 \cdot \sigma_{N-1}^2 \cdot \ldots \cdot \sigma_0^2 \]  
  \( (34) \)

- Prediction error variance $\sigma_N^2$ for optimal linear prediction using the $N$ preceding samples
  \[ \sigma_N^2 = \frac{|\mathbf{C}_{N+1}|}{|\mathbf{C}_N|} \]  
  \( (35) \)
One-Step Prediction for Autoregressive Processes

- Recall: AR($m$) process with mean $\mu_S$ and $a_m = (a_1, \cdots, a_m)^T$

$$S_n = Z_n + \mu_S(1 - a_m^T e_m) + a_m^T S^{(m)}_{n-1} \quad (36)$$

- Prediction using $N$ preceding samples in $h_N$ with $N \geq m$: define $a_N = (a_1, \cdots, a_m, 0, \cdots, 0)^T$

- Prediction error

$$U_n = S_n - h_N^T S_{n-1} = Z_n + \mu_S(1 - a_N^T e_N) + (a_N - h_N)^T S_{n-1} \quad (37)$$

- Subtracting the mean $E\{U_n\} = \mu_S(1 - a_N^T e_N) + (a_N - h_N)^T E\{S_{n-1}\}$

$$U_n - E\{U_n\} = Z_n + (a_N - h_N)^T (S_{n-1} - E\{S_{n-1}\}) \quad (38)$$

- Optimal prediction: covariances between $U_n$ and $S_{n-1}$ must be equal to 0

$$0 = E\{ (U_n - E\{U_n\} )(S_{n-k} - E\{S_{n-k}\} ) \}$$

$$= E\{ Z_n(S_{n-k} - E\{S_{n-k}\} ) \} + C_N(a_N - h_N) \quad (39)$$

yields

$$h_N^* = a_N \quad (40)$$
One-Step Prediction in Gauss-Markov Processes I

- Gauss-Markov process is a particular AR(1) process
  \[ S_n = Z_n + \mu_S (1 - \rho) + \rho \cdot S_{n-1}, \quad (41) \]
  for which the iid process \( \{Z_n\} \) has a Gaussian distribution

  → Completely characterized by its mean \( \mu_S \), its variance \( \sigma^2_S \), and the correlation coefficient \( \rho \) with \(-1 < \rho < 1\)

- Auto-covariance matrix and its inverse
  \[ C_2 = \sigma^2_S \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix} \quad C_2^{-1} = \frac{1}{\sigma^2_S (1 - \rho^2)} \begin{pmatrix} 1 & -\rho \\ -\rho & 1 \end{pmatrix} \quad (42) \]

- Auto-covariance vector
  \[ c_1 = \sigma^2_S \begin{pmatrix} \rho \\ \rho^2 \end{pmatrix} \quad (43) \]

- Optimum predictor \( h^*_2 = C_2^{-1} c_1 \)
  \[ h^*_2 = \frac{1}{1 - \rho^2} \begin{pmatrix} 1 & -\rho \\ -\rho & 1 \end{pmatrix} \begin{pmatrix} \rho \\ \rho^2 \end{pmatrix} = \frac{1}{1 - \rho^2} \begin{pmatrix} \rho - \rho^3 \\ -\rho^2 + \rho^2 \end{pmatrix} = \begin{pmatrix} \rho \\ 0 \end{pmatrix} \]

- First element of \( h^*_N \) is equal to \( \rho \), all other elements are equal to 0 (\( \forall N \geq 2 \))
Minimum prediction residual

\[ \sigma_U^2 = \frac{|C_2|}{|C_1|} = \frac{\sigma_S^4 - \sigma_S^4 \rho^2}{\sigma_S^2} = \sigma_S^2 (1 - \rho^2) \]  (44)

Prediction residual for filter \( h_1 \)

\[ U_n = S_n - h_1 S_{n-1} \]

Average squared error

\[ \sigma_U^2(h_1) = E\{U_n^2\} = \sigma_S^2(1 + h_1^2 - 2\rho h_1) \]

Note: obtain minimum MSE by

\[ \frac{\partial \sigma_U^2(h_1)}{\partial h_1} = \sigma_S^2(2h_1 - 2\rho) \overset{!}{=} 0 \]

also yields the result \( h_1 = \rho \)
Prediction Gain

- Prediction gain using \( \Phi_N = C_N/\sigma_S^2 \) and \( \phi_1 = c_1/\sigma_S^2 \)

\[
G_P = \frac{E\{S_n^2\}}{E\{U_n^2\}} = \frac{\sigma_S^2}{\sigma_U^2} = \frac{\sigma_S^2}{\sigma_S^2 - c_1^T C_N^{-1} c_1} = \frac{1}{1 - \phi_1^T \Phi_N^{-1} \phi_1},
\] (45)

- Prediction gain for optimal prediction in first-order Gauss-Markov process

\[
G_P(h^*) = \frac{1}{1 - \rho^2}
\] (46)

- Prediction gain for filter \( h_1 \)

\[
G_P(h_1) = \frac{\sigma_S^2}{\sigma_S^2(1 + h_1^2 - 2\rho h_1)} = \frac{1}{1 + h_1^2 - 2\rho h_1}
\]

- At high bit rates, \( 10 \log_{10} G_P \): SNR improvement achieved by predictive coding

![Graph showing the SNR improvement as a function of \( \rho \).]
Optimum Linear Prediction for $K = 2$

- The normalized auto-correlation matrix and its inverse follow as

$$\Phi_2 = \begin{pmatrix} 1 & \rho_1 \\ \rho_1 & 1 \end{pmatrix} \quad \Phi_2^{-1} = \frac{1}{1 - \rho_1^2} \begin{pmatrix} 1 & -\rho_1 \\ -\rho_1 & 1 \end{pmatrix}$$ (47)

- With normalized correlation vector

$$\phi_1 = \begin{pmatrix} \rho_1 \\ \rho_2 \end{pmatrix}$$ (48)

we obtain the optimum predictor

$$h_2^* = \Phi_2^{-1} \phi_1 = \frac{1}{1 - \rho_1^2} \begin{pmatrix} 1 & -\rho_1 \\ -\rho_1 & 1 \end{pmatrix} \begin{pmatrix} \rho_1 \\ \rho_2 \end{pmatrix} = \frac{1}{1 - \rho_1^2} \begin{pmatrix} \rho_1 - \rho_1 \rho_2 \\ -\rho_1^2 + \rho_2 \end{pmatrix}$$ (49)

- Result is identical to $h^*$ for the first-order Gauss-Markov source when setting $\rho_1 = \rho$ and $\rho_2 = \rho^2$

- For a source with $\rho_2 = \rho_1^2$: second coefficient doesn’t improve prediction gain → can be generalized to $N$th-order Gauss-Markov sources
Prediction for Speech Example

- Example for speech prediction: \( \rho_1 = 0.825, \rho_2 = 0.562 \rightarrow G_P(1) = 5.0 \text{ dB}, \ G_P(2) = 5.5 \text{ dB} \)
- Another speech prediction example
Prediction in Images: Intra Frame Prediction

- Past and present observable random variables are prior scanned samples within that image
- Derivations on linear prediction for zero-mean random variables (subtract $\mu_S$ or roughly 127 from 8-bit picture)
- Pictures are typically scanned line-by-line from upper left corner to lower right corner
- 1-D horizontal prediction:
  \[ \hat{S}_0 = h_1 \cdot S_1 \]
- 1-D vertical prediction:
  \[ \hat{S}_0 = h_2 \cdot S_2 \]
- 2-D prediction:
  \[ \hat{S}_0 = \sum_{i=1}^{3} h_i S_i \]
Prediction Example: Test Pattern

\[ \sigma^2_S = 4925.81 \] 
\[ (s - 127) \]

Horizontal Predictor
\[ h_1 = 0.953 \]
\[ h_2 = 0 \]
\[ h_3 = 0 \]
\[ \sigma^2_U(h) = 456.17 \]
\[ G_P = 10.33 \text{ dB} \]

Vertical Predictor
\[ h_1 = 0 \]
\[ h_2 = 0.932 \]
\[ h_3 = 0 \]
\[ \sigma^2_U(h) = 646.67 \]
\[ G_P = 8.82 \text{ dB} \]

2-d Predictor
\[ h_1 = 0.911 \]
\[ h_2 = 0.871 \]
\[ h_3 = -0.788 \]
\[ \sigma^2_U(h) = 109.90 \]
\[ G_P = 16.51 \text{ dB} \]
**Prediction Example: Lena**

256 × 256 center cropped picture

\[ \sigma_S^2 = 2746.43 \quad (s - 127) \]

**Horizontal Predictor**

\[
\begin{align*}
    h_1 &= 0.962 \\
    h_2 &= 0 \\
    h_3 &= 0
\end{align*}
\]

\[ \sigma_U^2(\mathbf{h}) = 212.36 \]

\[ G_P = 11.12 \text{ dB} \]

**Vertical Predictor**

\[
\begin{align*}
    h_1 &= 0 \\
    h_2 &= 0.977 \\
    h_3 &= 0
\end{align*}
\]

\[ \sigma_U^2(\mathbf{h}) = 123.61 \]

\[ G_P = 13.47 \text{ dB} \]

**2-d Predictor**

\[
\begin{align*}
    h_1 &= 0.623 \\
    h_2 &= 0.835 \\
    h_3 &= -0.48
\end{align*}
\]

\[ \sigma_U^2(\mathbf{h}) = 80.35 \]

\[ G_P = 15.34 \text{ dB} \]
Prediction Example: PMFs for Picture Lena

- Pms \( p(s) \) and \( p(u) \) change significantly due to prediction operation
- Entropy changes significantly
  (rounding prediction signal towards integer: \( E \{ [U_n + 0.5]^2 \} = 80.47 \))

\[
H(S) = 7.44 \text{ bit/sample} \quad H(U) = 4.97 \text{ bit/sample} \quad (50)
\]

- Linear prediction can be used for lossless coding: JPEG-LS
Asymptotic Prediction Gain

- Upper bound for prediction gain as $N \to \infty$
- One-step prediction of a random variable $S_n$ given the countably infinite set of preceding random variables $\{S_{n-1}, S_{n-2}, \cdots\}$ and $\{h_0, h_1, \cdots\}$

\[
U_n = S_n - h_0 - \sum_{i=1}^{\infty} h_i S_{n-i},
\]  

(51)

- Orthogonality criterion: $U_n$ is uncorrelated with all $S_{n-i}$ for $i > 0$
- But $U_{n-k}$ for $k > 0$ is fully determined by a linear combination of past input values $S_{n-k-i}$ for $i \geq 0$
- Hence, $U_n$ is uncorrelated with $U_{n-k}$ for $k > 0$

\[
\phi_{UU}(k) = \sigma_{U,\infty}^2 \cdot \delta(k) \quad \leftrightarrow \quad \Phi_{UU}(\omega) = \sigma_{U,\infty}^2
\]  

(52)

where $\sigma_{U,\infty}^2$ is the asymptotic one-step prediction error variance for $N \to \infty$
For one-step prediction we showed
\[ |C_N| = \sigma_{N-1}^2 \cdot \sigma_{N-2}^2 \cdot \sigma_{N-3}^2 \cdots \sigma_0^2 \] (53)

which yields
\[ \frac{1}{N} \ln |C_N| = \ln |C_N|^{\frac{1}{N}} = \frac{1}{N} \sum_{i=0}^{N-1} \ln \sigma_i^2 \] (54)

If a sequence of numbers \( \alpha_0, \alpha_1, \alpha_2, \cdots \) approaches a limit \( \alpha_\infty \), the average value approaches the same limit,
\[ \lim_{N \to \infty} \frac{1}{N} \sum_{i=0}^{N-1} \alpha_i = \alpha_\infty \] (55)

Hence, we can write
\[ \lim_{N \to \infty} \ln |C_N|^{\frac{1}{N}} = \lim_{N \to \infty} \frac{1}{N} \sum_{i=0}^{N-1} \ln \sigma_i^2 = \ln \sigma_\infty^2 \] (56)

yielding
\[ \sigma_\infty^2 = \exp \left( \lim_{N \to \infty} \ln |C_N|^{\frac{1}{N}} \right) = \lim_{N \to \infty} |C_N|^{\frac{1}{N}} \] (57)
Asymptotic One-Step Prediction Error Variance II

- Asymptotic One-Step Prediction Error Variance

\[ \sigma_{U,\infty}^2 = \lim_{N \to \infty} |C_N|^{\frac{1}{N}} \]

- Determinant of $N \times N$ matrix: product of its eigenvalues $\xi_i^{(N)}$

\[
\lim_{N \to \infty} |C_N|^{\frac{1}{N}} = \lim_{N \to \infty} \left( \prod_{i=0}^{N-1} \xi_i^{(N)} \right)^{\frac{1}{N}} = 2 \left( \lim_{N \to \infty} \sum_{i=0}^{N-1} \frac{1}{N} \log_2 \xi_i^{(N)} \right) \tag{58}
\]

- Apply Grenander and Szegö’s theorem

\[
\lim_{N \to \infty} \frac{1}{N} \sum_{i=0}^{N-1} G \left( \xi_i^{(N)} \right) = \frac{1}{2\pi} \int_{-\pi}^{\pi} G \left( \Phi(\omega) \right) d\omega \tag{59}
\]

- Expression using power spectral density

\[
\sigma_{U,\infty}^2 = \lim_{N \to \infty} |C_N|^{\frac{1}{N}} = 2 \frac{1}{2\pi} \int_{-\pi}^{\pi} \log_2 \Phi_{SS}(\omega) d\omega \tag{60}
\]
Asymptotic Prediction Gain

- Prediction gain $G_P^\infty$

$$G_P^\infty = \frac{\sigma^2_S}{\sigma^2_{U,\infty}} = \frac{1}{2\pi} \int_{-\pi}^{\pi} \Phi(\omega) \, d\omega \quad \leftarrow \text{Arithmetic mean}$$

$$= \frac{1}{2} \int_{-\pi}^{\pi} \log_2 \Phi(\omega) \, d\omega \quad \leftarrow \text{Geometric mean}$$  \hspace{1cm} (61)

- Result for first-order Gauss-Markov source (can also be computed differently)
Combining prediction with quantization requires simultaneous reconstruction of predictor at coder and decoder → use of quantized samples for prediction

Re-drawing yields block-diagram with typical DPCM structure
DPCM Codec

- Redrawing with encoder $\alpha$, mapping from index to bit stream $\gamma$, and decoder $\beta$ yields DPCM encoder.

- DPCM encoder contains DPCM decoder except for $\gamma^{-1}$.
DPCM and Quantization

- Prediction $\hat{S}_n$ for a sample $S_n$ is generated by linear filtering of reconstructed samples $S'_n$ from the past

$$
\hat{S}_n = \sum_{i=1}^{K} h_i S'_{n-i} = \sum_{i=1}^{K} h_i (S_{n-i} + Q_{n-i}) = h^T \cdot (S_{n-1} + Q_{n-1})
$$

(62)

with $Q_n$ being the quantization error between the reconstructed samples $S'_n$ and original samples $S_n$

- Prediction error variance (for zero-mean input) is given by

$$
\sigma^2_U = E\{U_n^2\} = E\{(S_n - \hat{S}_n)^2\} = E\{ (S_n - h^T S_{n-1} - h^T Q_{n-1})^2 \}
$$

$$
= E\{S_n^2\} + h^T E\{S_{n-1} S_{n-1}^T\} h + h^T E\{Q_{n-1} Q_{n-1}^T\} h
$$

$$
- 2h^T E\{S_{n-1} S_{n-1}\} - 2h^T E\{S_{n} Q_{n-1}\} + 2h^T E\{S_{n-1} Q_{n-1}^T\} h
$$

(63)

- Defining $\Phi = E\{S_{n-1} S_{n-1}^T\} / \sigma^2_S$ and $\phi = E\{S_{n} S_{n-1}\} / \sigma^2_S$ we get

$$
\sigma^2_U = \sigma^2_S \left( 1 + h^T \Phi h - 2h^T \phi \right)
$$

$$
+ h^T E\{Q_{n-1} Q_{n-1}^T\} h - 2h^T E\{S_{n} Q_{n-1}\} + 2h^T E\{S_{n-1} Q_{n-1}^T\} h
$$

(64)
DPCM for a First-Order Gauss-Markov Source

- Calculate $R(D)$ for zero-mean Gauss-Markov process with $-1 < \rho < 1$ and variance $\sigma_S^2$:
  \[ S_n = Z_n + \rho \cdot S_{n-1} \] (65)

- Consider a one-tap linear prediction filter $h = (h)$

- Normalized auto-correlation matrix $\Phi = (1)$ and cross-correlation $\phi = (\rho)$

- Prediction error variance
  \[ \sigma_U^2 = \sigma_S^2 (1 + h^2 - 2h\rho) + h^2 E \{ Q_{n-1}^2 \} \]
  \[ -2hE \{ S_nQ_{n-1} \} + 2h^2 E \{ S_{n-1}Q_{n-1} \} \] (66)

- Using $S_n = Z_n + \rho \cdot S_{n-1}$, the second row in above equation becomes
  \[ -2hE \{ S_nQ_{n-1} \} + 2h^2 E \{ S_{n-1}Q_{n-1} \} \]
  \[ = -2hE \{ Z_nQ_{n-1} \} - 2h\rho E \{ S_{n-1}Q_{n-1} \} + 2h^2 E \{ S_{n-1}Q_{n-1} \} \]
  \[ = -2hE \{ Z_nQ_{n-1} \} + 2h(h - \rho)E \{ S_{n-1}Q_{n-1} \} \] (67)

- With setting $h = \rho$, we have
  \[ E \{ Z_nQ_{n-1} \} = 0 \quad 2h(h - \rho)E \{ S_{n-1}Q_{n-1} \} = 0 \] (68)
Combination of DPCM with ECSQ for Gauss-Markov Processes

- Expression for prediction error variance simplifies to
  \[
  \sigma^2_U = \sigma^2_S (1 - \rho^2) + \rho^2 E \{Q^2_{n-1}\} 
  \]  
  (69)

- Model expression for quantization error \( D = E \{Q^2_{n-1}\} \) by an operational distortion rate function
  \[
  D(R) = \sigma^2_U \cdot g(R) 
  \]  
  (70)

- Example: Assume ECSQ and with that \( g(R) \) as
  \[
  g(R) = \frac{\varepsilon^2 \ln 2}{a} \log_2(a \cdot 2^{-2R} + 1) 
  \]  
  (71)

  with \( a = 0.9519 \) and \( \varepsilon^2 = \pi e / 6 \)

- Expression for prediction error variance becomes dependent on rate
  \[
  \sigma^2_U = \sigma^2_S \cdot \frac{1 - \rho^2}{1 - g(R) \rho^2} 
  \]  
  (72)
Computation of Operational Distortion Rate Function for DPCM

- Operational distortion rate function for DPCM and ECSQ for a first-order Gauss-Markov source

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

(73)

- Algorithm for ECSQ in DPCM coding
  1. Initialization with a small value of \( \lambda \), set \( s'_n = s_n \), \( \forall n \) and \( h = \rho \)
  2. Create signal \( u_n \) using \( s'_n \) and DCPM
  3. Design ECSQ \((\alpha, \beta, \gamma)\) using signal \( u_n \) and the current value of \( \lambda \) by minimizing \( D + \lambda R \)
  4. Conduct DPCM encoding/decoding using ECSQ \((\alpha, \beta, \gamma)\)
  5. Measure \( \sigma_U^2(R) \) as well as rate \( R \) and distortion \( D \)
  6. Increase \( \lambda \) and start again with step 2

- Algorithm shows problems at low bit rates: instabilities
Comparison of Theoretical and Experimental Results I

- For high rates and Gauss-Markov sources, shape and memory gain achievable
- Space-filling gain can only be achieved using vector quantization
- Theoretical model provides a useful description

**Distortion-Rate Function** $D(R)$

$$D(R) = \sigma^2_U(R) g(R)$$

- **EC-Lloyd and DPCM**
- **EC-Lloyd (no prediction)**

$$D(R) = \sigma^2_S g(R)$$

- **Space-Filling Gain: 1.53 dB**
- **$G_P^\infty = 7.21$ dB**

SNR [dB]

Bit Rate [bit/sample]
Comparison of Theoretical and Experimental Results (I)

- Prediction error variance $\sigma_U^2$ depends on bit rate
- Theoretical model provides a useful description

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

\[
\sigma_U^2(\infty) = \sigma_S^2 \cdot (1 - \rho^2)
\]
Adaptive Differential Pulse Code Modulation (ADPCM)

- For quasi-stationary sources like speech, fixed predictor is not well suited
- ADPCM: Adapt the predictor based on the recent signal characteristics
- Forward adaptation: send new predictor values - additional bit rate
Forward-Adaptive Prediction: Motion Compensation in Video Coding

- Since predictor values are sent, extend prediction to vectors/blocks
- Use statistical dependencies between two pictures
- Prediction signal obtained by searching a region in a previously decoded picture that best matches the block to be coded
- Let $s[x, y]$ represent intensity at location $(x, y)$
- Let $s'[x, y]$ represent intensity in a previously decoded picture also at location $(x, y)$

$$J = \min_{(dx, dy)} \sum_{x,y} (s[x, y] - s'[x - dx, y - dy])^2 + \lambda R(dx, dy) \quad (74)$$

- Predicted signal is specified through motion vector $(dx, dy)$ and $R(dx, dy)$ is its number of bits
- Prediction error $u[x, y]$ is quantized (often using transform coding)
- Bit rate in video coding is sum of motion vector and prediction residual bit rate
Backward Adaptive DPCM

- Backward adaptation: use predictor computed from recently decoded signal
  - No additional bit rate
  - Error resilience issues
  - Accuracy of predictor

![Diagram of Backward Adaptive DPCM]

\[
\begin{align*}
S_n & \quad U_n \\
\alpha & \quad I_n \\
\beta & \quad B_n \\
\gamma & \quad \text{Channel} \\
\hat{S}_n & \quad U'_n \\
P & \quad \text{APB} \\
\end{align*}
\]
Adaptive Linear Prediction

- Computational problems when inverting $\Phi$ for computing $h_* = \Phi^{-1}\phi$
- Gradient of objective function $\frac{d\sigma^2_U(h)}{dh} = \sigma^2_U(\Phi h - \phi)$
- Instead of setting $\frac{d\sigma^2_U(h)}{dh} \neq 0$ which leads to matrix inversion, approach minimum by iteratively adapting prediction filter
- Steepest Descent Algorithm: Update filter coefficients in the direction of negative gradient of objective function

$$h[n + 1] = h[n] + \Delta h[n] = h[n] + \kappa \cdot (\phi - \Phi h[n]) \quad (75)$$
Least Mean Squared (LMS) Algorithm

- LMS is a stochastic gradient algorithm [Widrow, Hoff, 1960] approximating steepest descent
- LMS proposes simple current-value approximation

\[
\Phi \cdot \sigma^2_S = E \left\{ S_{n-1} \cdot S^T_{n-1} \right\} \approx s_{n-1} \cdot s^T_{n-1} \tag{76}
\]

\[
\phi \cdot \sigma^2_S = E \left\{ S_{n-1} \cdot S \right\} \approx s_{n-1} \cdot s_n \tag{77}
\]

- Update equation becomes

\[
h[n + 1] = h[n] + \kappa \cdot (s_{n-1}s_n - s_{n-1}s^T_{n-1}h[n])
\]

\[
= h[n] + \kappa \cdot s_{n-1} \cdot (s_n - s^T_{n-1}h[n]) \tag{78}
\]

- Realizing that prediction error is given as \( u_n = s_n - s^T_{n-1}h[n] \)

\[
h[n + 1] = h[n] + \kappa \cdot s_{n-1} \cdot u_n \tag{79}
\]

- LMS is one of many adaptive algorithms to determine \( h \), including [Itakura and Saito, 1968, Atal and Hanauer, 1971, Makhoul and Wolf, 1972]
  - Autocorrelation solution
  - Covariance solution
  - Lattice solution
Linear Predictive Coding of Speech

- Speech coding is done using source modeling
- All-pole signal processing model for speech production is assumed
- Speech spectrum $S(z)$ is produced by passing an excitation spectrum, $V(z)$, through an all-pole transfer function $H(z) = \frac{G}{A(z)}$

$$S(z) = H(z) \cdot V(z) = \frac{G}{A(z)} \cdot V(z)$$  \hspace{1cm} (80)

where, $A(z) = 1 - \sum_{k=1}^{P} a_k \cdot z^{-k}$

- Corresponding difference equation

$$s[n] = \sum_{k=1}^{P} a_k \cdot s[n - k] + G \cdot v[n]$$  \hspace{1cm} (81)

- When input $v[n]$ is train of impulses, it produces voiced speech
- When $v[n]$ is noise-like, it produces unvoiced speech (e.g. sounds like 'f', 's', etc)
Prediction in Speech

- Prediction based on LPC is called Short Term Prediction (STP) as it generally operates on recent speech samples (e.g. around 10 samples)

\[
\hat{s}[n] = \sum_{i=1}^{N} a_i \cdot s[n - i]
\] (82)

- After STP, resulting prediction error

\[
u[n] = s[n] - \hat{s}[n]
\] (83)

still has distant sample correlation (known as Pitch)

- Pitch is predicted by Long Term Prediction (LTP) by block matching using cross-correlation

\[
R(l) = \frac{\sum_{n=0}^{N-1} u[n] \cdot u[n - l]}{\sum_{n=0}^{N-1} u[n - l] \cdot u[n - l]}
\] (84)

Location \( l_{opt} \) that maximizes cross-correlation is called lag

- Signal block at lag is subtracted from \( u[n] \) - resulting signal is called excitation sequence
Prediction in Speech: Code Excited Linear Prediction (CELP)

- Instead of quantizing and transmitting the excitation signal, CELP attempts to transmit an index from a codebook to approximate the excitation signal.
- One method would be to Vector Quantize the excitation signal to best match a codebook entry.
- But since this signal passes through LPC synthesis filter, the behavior after filtering might not necessarily be optimal.
- Analysis-by-Synthesis (AbS) approach: Encoding (analysis) is performed by optimising the decoded (synthesis) signal in a closed loop.
- At encoder, excitation sequences in codebook are passed through synthesis filter and the index of best excitation is transmitted.
CELP and AbS

Analysis-by-Synthesis Loop

Speech Generator

Frame Buffer

Speech

Spectral Analysis

Coding Error

Perceptually weighted Error

Minimization Procedure

Synthesized Speech

1
2
K

Vector Excitations

Scaling

Synthesis Filter

α
Transmission Errors in DPCM

- For a linear DPCM decoder, the transmission error response is superimposed to the reconstructed signal $s'$
- For a stable DPCM decoder, transmission error response decays
- Finite word-length effects at decoder can lead to residual errors that do not decay (e.g. limit cycles)
- Below: (a) error sequence (BER of 0.5%) (b) error-free transmission (c) error propagation
Transmission Errors in DPCM for Pictures

Example: Lena, 3 b/p (fixed code word length)

1D pred. hor. $a_H = 0.95$, 1D pred. ver. $a_V = 0.95$, 2D pred. $a_H = a_V = 0.5$
Transmission Errors in DPCM for Motion Compensation in Video Coding

- When transmission error occurs, motion compensation causes spatio-temporal error propagation
- Try to conceal image parts that are in error
- Code lost image parts without referencing concealed image parts helps but reduces coding efficiency – intra block

Use "clean" reference picture for motion compensation
Summary on Predictive Coding

- Prediction: Estimation of random variable from past or present observable random variables
- Optimal prediction – only in special cases
- Optimal linear prediction – simple and efficient
- Wiener-Hopf equation for optimal linear prediction
- Gauss-Markov process of order $N$ requires predictor with $N$ coefficients that are equal to correlation coefficients
- Non-matched predictor can increase signal variance
- Optimal prediction error is orthogonal to input signal
- Optimal prediction error filter operates as a whitening filter
Summary on Predictive Coding (cont’d)

- Differential pulse code modulation (DPCM) is a structure for the combination of prediction with quantization.
- In DPCM, prediction is based on quantized samples.
- Simple and efficient: combine DPCM and ECSQ.
- Extension of Entropy-Constrained Lloyd algorithm towards DPCM.
- For Gauss-Markov sources, EC-Lloyd for DPCM achieves shape and memory gain.
- Adaptive DPCM: forward and backward adaptation.
- Forward adaptation requires transmission of predictor values.
- Backward adaptation poses problems of error resilience and accuracy questions.
- Adaptive linear prediction using steepest descent algorithm: LMS, autocovariance, covariance, and lattice solutions.
- Transmission errors cause error propagation in DPCM.
- Error propagation can be mitigated by interrupting erroneous prediction chains.