Quantization and Erasures on Frame Representations

Petros Boufounos
petrosb@rice.edu

Rice ECE presentation
Oct. 3, 2006
Motivation

- Frame representations are overcomplete generalizations to basis expansions.
  - Frame representations are often used in signal processing applications to provide robustness to errors.
  - Oversampling is one example, and the main motivation for the thesis.

- Goal: Exploit the redundancy of frame representations to develop efficient algorithms that provide robustness to quantization and erasures.

- Main Tool: Error compensation using projections.
Overview

• Background

• Quantization and Sigma-Delta noise shaping.

• Erasure Compensation using projections

• Puncturing of Dense Representations

• Summary
Basis Expansions

Analysis: \( a_k = \langle x, \overline{b}_k \rangle \)

Synthesis: \( x = \sum_k a_k b_k \)
Frame Representations

Analysis: \( a_k = \langle x, \overline{f}_k \rangle \)

Synthesis: \( x = \sum_k a_k \overline{f}_k \)

\( N \) dimensional signal space, \( M \geq N \) frame vectors/coefficient
Redundancy \( r = M/N \)
Examples of Frames and Frame Expansions

Matrix Operations in $\mathbb{IR}^N$:

Analysis:
\[
\begin{bmatrix}
-\bar{f}_1 \\
\vdots \\
-\bar{f}_M
\end{bmatrix}
\begin{bmatrix}
x \\
\vdots \\
\bar{a}_M
\end{bmatrix} \Leftrightarrow a_k = \langle x, \bar{f}_k \rangle
\]

Redundancy:
\[r = M/N\]

Synthesis:
\[
\begin{bmatrix}
f_1 \\
\vdots \\
f_M
\end{bmatrix}
\begin{bmatrix}
a_1 \\
\vdots \\
a_M
\end{bmatrix} = \begin{bmatrix} x \end{bmatrix} \Leftrightarrow x = \sum_k a_k f_k
\]

$r$-times Oversampling:

Analysis:
\[
a_k = \int_{-\infty}^{\infty} x(t) \frac{1}{rT} \text{sinc} \left( \frac{r}{T} t - k \right) dt \Leftrightarrow a_k = \langle x, \bar{f}_k \rangle
\]

Synthesis:
\[
x(t) = \sum_k a_k \text{sinc} \left( \frac{r}{T} t - k \right) \Leftrightarrow x = \sum_k a_k f_k
\]
The Frame Operators

Signal Space $\mathcal{W}$
Frame: $\{f_k \in \mathcal{W} | k = 1, \ldots, M\}$
$\dim(\mathcal{W}) = N$

Coefficient Space $\mathcal{H}$
Orthonormal Basis: $\{b_k \in \mathcal{H} | k = 1, \ldots, M\}$
$\dim(\mathcal{H}) = M \geq N$

Analysis Operator: $F : \mathcal{W} \rightarrow \mathcal{H}$ s.t. $F(x) = \sum_k \langle x, f_k \rangle b_k = \sum_k a_k b_k$

Synthesis Operator: $S : \mathcal{H} \rightarrow \mathcal{W}$ s.t. $S(x) = \sum_k \langle x, b_k \rangle f_k = \sum_k a_k f_k$
Overview

• Background

• **Quantization and Sigma-Delta noise shaping.**

• Erasure Compensation using projections

• Puncturing of Dense Representations

• Summary
Scalar Quantization

$L$ level quantizer: $\sim \log_2(L)$ bits per coefficient

Additive noise model: $e_k$ uncorrelated, uniform in $\pm \frac{\Delta}{2}$

$$\sigma_e^2 = \frac{\Delta^2}{12} \propto \frac{2^{-2b}}{12}$$
Quantization of Orthonormal Basis Expansions

\[
\begin{align*}
\mathbf{x} & \rightarrow \langle \mathbf{x}, \mathbf{b}_k \rangle \\
& \rightarrow a_k \rightarrow Q(\cdot) \rightarrow \hat{a}_k \rightarrow \sum_k \hat{a}_k \mathbf{b}_k \\
& \rightarrow \hat{\mathbf{x}} = \sum_k \hat{a}_k \mathbf{b}_k
\end{align*}
\]

\[
\log_2 L \text{ bits per coefficient} \\
M \text{ expansion coefficients} \implies M \log_2 L \text{ bits used}
\]
Quantization of Frame Representations

\[ x \rightarrow \langle x, f_k \rangle \rightarrow a_k \rightarrow Q(\cdot) \rightarrow \hat{a}_k \rightarrow \text{Synthesis} \rightarrow \hat{x} \]

Very few quantization cells are intersected.
Bounds on Scalar Quantization

Number of cells intersected $I(M,N,L)$:

$$I(M,N,L) \leq (2L)^N \binom{M}{N} \leq \left( \frac{2LMe}{N} \right)^N = (2Lre)^N$$

$$\text{dim}(\mathcal{W}) = \text{rank}(F) = N$$

$$\text{dim}(\mathcal{H}) = M, \quad r = \frac{M}{N}$$

Bit-use efficiency:

$$\frac{\text{bits necessary}}{\text{bits used}} \leq \frac{\log_2 (I(M,N,L))}{N \log_2 L} \leq \frac{\log_2 (2Lre)}{r \log_2 L},$$

Quantization Error Reduction Rate:

$$\varepsilon^2 \rightarrow \Omega(1/r^2)$$

Simple scalar quantization is inefficient.
Oversampling and Quantization

\[
x(t) \xrightarrow{\text{LPF}} C/D \xrightarrow{\hat{a}_k} \xrightarrow{\text{Q( )}} \text{D/C} \xrightarrow{\hat{D}(t)} \hat{x}(t)
\]

Oversampling (analysis)
Quantization
Interpolation (synthesis)

Using the additive noise model, \( e_k \) uncorrelated, uniform in \( \pm \frac{\Delta}{2} \):

\[
X(j\Omega) \xrightarrow{\text{LPF}} A(e^{j\omega}) \xrightarrow{\text{Q( )}} \hat{A}(e^{j\omega}) \xrightarrow{\text{D/C}} \hat{X}(j\Omega)
\]

Tradeoff: Gain 1 bit for each 4 times oversampling
Similar tradeoff for arbitrary frame expansions
Can we extend noise shaping to arbitrary frames?

Optimal choice $c = \text{sinc} \left( \frac{\pi}{r} \right) (\approx 1 \text{ for } r \geq 4)$
Error compensation using projections

\[ \mathbf{x} = a_1 \mathbf{f}_1 + a_2 \mathbf{f}_2 + a_3 \mathbf{f}_3 \]

1. Quantization
\[ \mathbf{x} = \hat{a}_1 \mathbf{f}_1 + a_2 \mathbf{f}_2 + a_3 \mathbf{f}_3 - e_1 \mathbf{f}_1 \]

2. Compensation using projection
\[ a'_2 = a'_2 - e_1 c_{1,2} \]
\[ \mathbf{x} = \hat{a}_1 \mathbf{f}_1 + a'_2 \mathbf{f}_2 + a_3 \mathbf{f}_3 - e_1 (\mathbf{f}_1 - c_{1,2} \mathbf{f}_2) \]

Incremental error: \(-e_1 (\mathbf{f}_1 - c_{1,2} \mathbf{f}_2)\)

\[ c_{1,2} = \frac{\langle \mathbf{f}_1, \mathbf{f}_2 \rangle}{\| \mathbf{f}_2 \|^2} \]

Compensation linear in the error.
Projection coefficients \(c_{i,j}\) can be pre-computed.
Higher Order Projections

\[ \hat{a}_1 \mathbf{f}_1 - e_1 \mathbf{f}_1 \]

\[ \mathbf{x} = a_1 \mathbf{f}_1 + a_2 \mathbf{f}_2 + a_3 \mathbf{f}_3 \]

1. Quantization:

\[ \hat{a}_k = Q(a'_k) = a'_k + e_k \]

2. Projection:

\[ a'_{k+1} = a_{k+1} - c_{k,k+1}e_k \]

\[ a'_{k+p} = a_{k+p} - c_{k,k+p}e_k \]

Solution might not be unique; all have the same error.
1. Quantization:
\[ \hat{a}_k = Q(a'_k) = a'_k + e_k \]

2. Projection (Coefficient Update):
\[
\begin{align*}
\hat{a}'_{k+1} &= a_{k+1} - c_{k,k+1} e_k \\
\vdots \\
\hat{a}'_{k+p} &= a_{k+p} - c_{k,k+p} e_k
\end{align*}
\]

Average and worst case error is reduced.
Noise Shaping Example

- Random points on the plane, uniform inside the unit circle.
- Optimal ordering (one of many) is: \( \{f_1, f_4, f_7, f_3, f_6, f_2, f_5\} \)
Simplified D/A Converters

\[ x[n] \xrightarrow{\uparrow r} \text{LPF} \xrightarrow{a_k} \Sigma \Delta \xrightarrow{\hat{a}_k} \text{D/C} \xrightarrow{\text{LPF}} \hat{x}(t) \]

Oversampling (analysis) \hspace{2cm} Quantization \hspace{2cm} Interpolation (synthesis)

\[ x[n] \xrightarrow{\uparrow r \ \text{Gain} \ r} \Sigma \Delta \xrightarrow{\hat{a}_k} \text{D/C} \xrightarrow{\text{LPF}} \hat{x}(t) \]

\[ x[n] \xrightarrow{\uparrow r_1} \text{LPF} \xrightarrow{\uparrow r_2} \Sigma \Delta \xrightarrow{\hat{a}_k} \text{D/C} \xrightarrow{\text{LPF}} \hat{x}(t) \]

Simplified D/A structure:
Tunable DACs and ADCs

**Tunable DAC:**

\[
x[n] \xrightarrow{+} x[n] \xrightarrow{-} H_f(z) \xrightarrow{-} Q() \xrightarrow{+} \hat{x}[n] \xrightarrow{-} \hat{x}[n] \xrightarrow{+} D/C: \sum_k \hat{x}[k] \delta[n-k] \xrightarrow{Tunable h(t)} \hat{x}(t)
\]

**Tunable ADC:**

\[
x(t) \xrightarrow{+} x(t) \xrightarrow{-} c_p \xrightarrow{z^{-1}} \ldots \xrightarrow{c_2} \xrightarrow{c_1} \xrightarrow{z^{-1}} \xrightarrow{ADC} \xrightarrow{Tunable h[n]} \xrightarrow{DAC} \xrightarrow{z^{-1}} \xrightarrow{z^{-1}} \xrightarrow{-} \hat{x}[n]
\]
Overview

• Background

• Quantization and Sigma-Delta noise shaping.

• **Erasure Compensation using projections**

• Puncturing of Dense Representations

• Summary
Erasures

1. Erasure:
   \[ e_k = 0 \Rightarrow \hat{a}_k = 0 \]

2. Projection:
   \[ a'_{k+1} = a_{k+1} + c_{k,k+1}a_k \]
   \[ \vdots \]
   \[ a'_{k+p} = a_{k+p} + c_{k,k+p}a_k \]

The transmitter should know the erasure occurred.
Pre-compensation with Post-correction

Algorithm:

1. **Project** as if an erasure will occur.
2. **Transmit**, subject to erasures.
3. **Undo** the projection if erasure does not occur. Otherwise **do nothing**.

Input-output behavior is the same.
Only the receiver needs to be aware of the erasure.
System Stability (LTI case)

Necessary condition:
System\[\frac{1}{1 - \sum_{k=1}^{p} q c_k z^{-k}}\]
is stable

Sufficient condition:
System\[\frac{1}{1 - \sum_{k=1}^{p} \sqrt{q} |c_k| z^{-k}}\]
is stable

Sub-optimal compensation is sometimes necessary for stability.
Simulation with the Oversampling Frame

- Input $a_k$: 0-mean, unit variance, white Gaussian process.
- Erasures $e_k$: i.i.d, $P(\text{erasure}) = q$
- Oversampling frame with ratio $r=4$ and 8, approximated with 4096 tap Hamming window FIR filter
Simulation with Oversampled Filterbank

- $h_l[n]=$Length 4 Harmonic Frame, $M=1$, $L=8$, $r=8$
- Input $a_k$: 0-mean, unit variance, white Gaussian process.
- Erasures $e_k$: i.i.d, $P$(erasure)$=q$
Overview

• Background

• Quantization and Sigma-Delta noise shaping.

• Erasure Compensation using projections

• **Puncturing of Dense Representations**

• Summary
Puncturing of Dense Representations

Starting with a dense overcomplete frame representation:

\[ \{a_1, a_2, \ldots, a_M\}, \text{ such that } x = \sum_{k=1}^{M} a_k f_k, \quad x \in \mathcal{W}, \quad \dim(\mathcal{W}) = N \]

Erase coefficients to produce a sparse approximation of \( x \), projecting the erasure error to the remaining ones:

\[ \{\hat{a}_k \mid k \in S\}, \text{ such that } \hat{x} = \sum_{k \in S} \hat{a}_k f_k \approx x, \quad \varepsilon = x - \hat{x}, \]

with \( S \) being a sparse subset of indices (i.e. \( |S| < N \)).

Puncturing algorithm:

1. **Select** coefficients to be erased at iteration \( i \)
2. **Project** the erasure error to **all** the remaining ones
3. **Iterate** from 1, or stop.
Puncturing Algorithm Properties

• The schedule is dynamically determined.
  – The choice of the schedule affects the performance.
  – Can be determined by the error introduced at each step.

• First $M-N$ coefficients can be erased without error.
  – The sequence of erasures can affect subsequent performance.

• Each iteration performs an orthogonal projection.

• Error properties
  – The total error $\epsilon$ is orthogonal to the remaining data
  – The incremental error $\epsilon_i$ forms an orthogonal set:
    \[
    \epsilon = \sum_i \epsilon_i, \quad \langle \epsilon_i, \epsilon_j \rangle \propto \delta_{i,j}
    \]
    \[
    \Rightarrow \|\epsilon\|^2 = \sum_i \|\epsilon_i\|^2
    \]

General algorithm. Details depend on the application.
Summary

• Error compensation using projections
  – Principle can be used for several types of errors.

• Generalization of Sigma-Delta noise shaping.
  – Improves quantization performance over scalar quantization.

• Erasure Compensation using projections
  – Provides an efficient, causal, system to compensate at the transmitter, even if the transmitter is not aware of the erasure occurrence.

• Puncturing of Dense Representations
  – General algorithm that can be applied in different applications.
Acknowledgements

• Al Oppenheim

• George Verghese

• Vivek Goyal

• DSP Group at MIT
Questions/Comments?