# Quantizer optimization for LPC based voice coder: Codec 2

Wojciech Kaczmarski, M17 Project

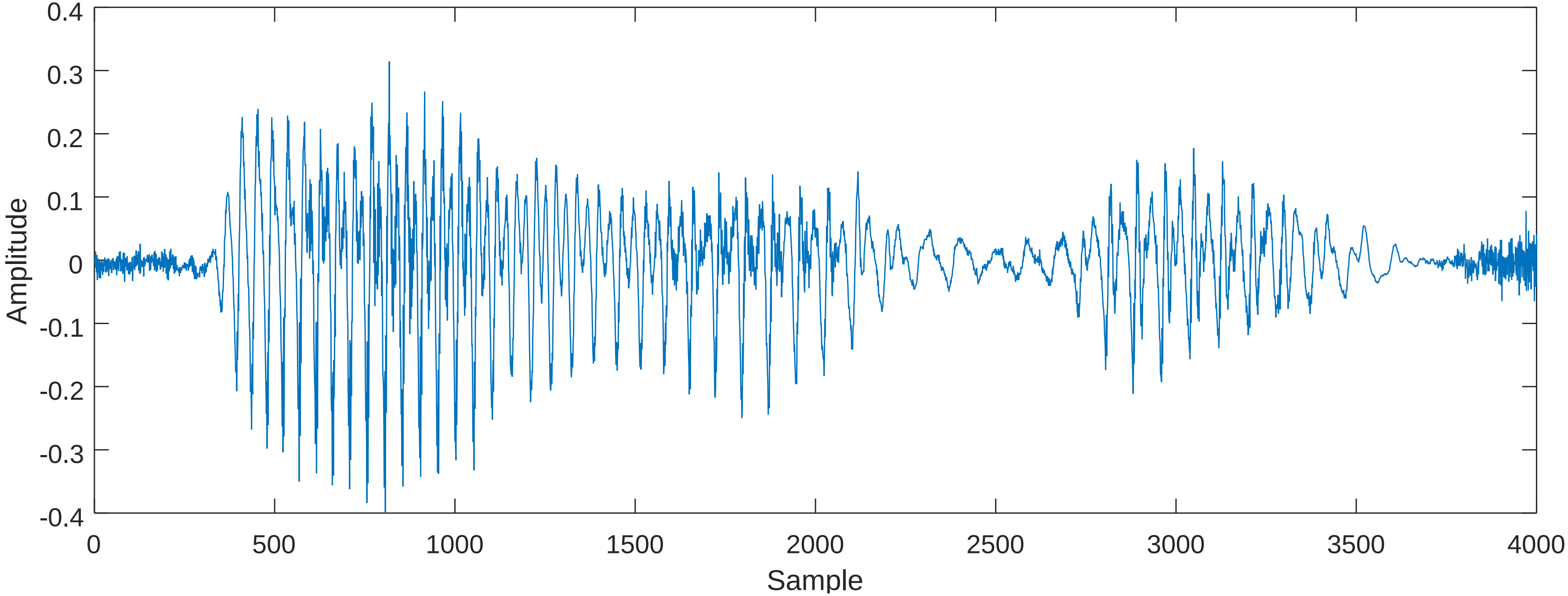
# Codec 2 - introduction

- A free to use and open-source, low bitrate voice codec
- Developed by David Rowe VK5DGR et al.
- We are going to focus on the full rate, 3200 bps mode

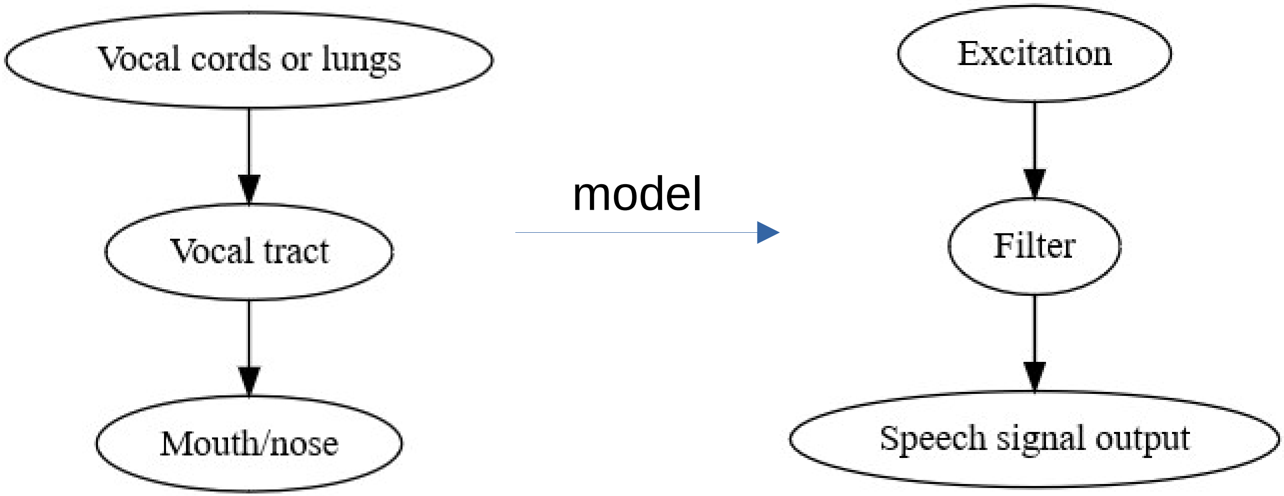
# Modeling speech

- Speech is a sound pressure wave generated by vocal chords and filtered by an anatomical structure called the vocal tract
- We can coarsely model the speech generation process using an excitation source and a filter
- Both the excitation source and filter change their paremeters over time

# Speech signal



# Excitation and filtering

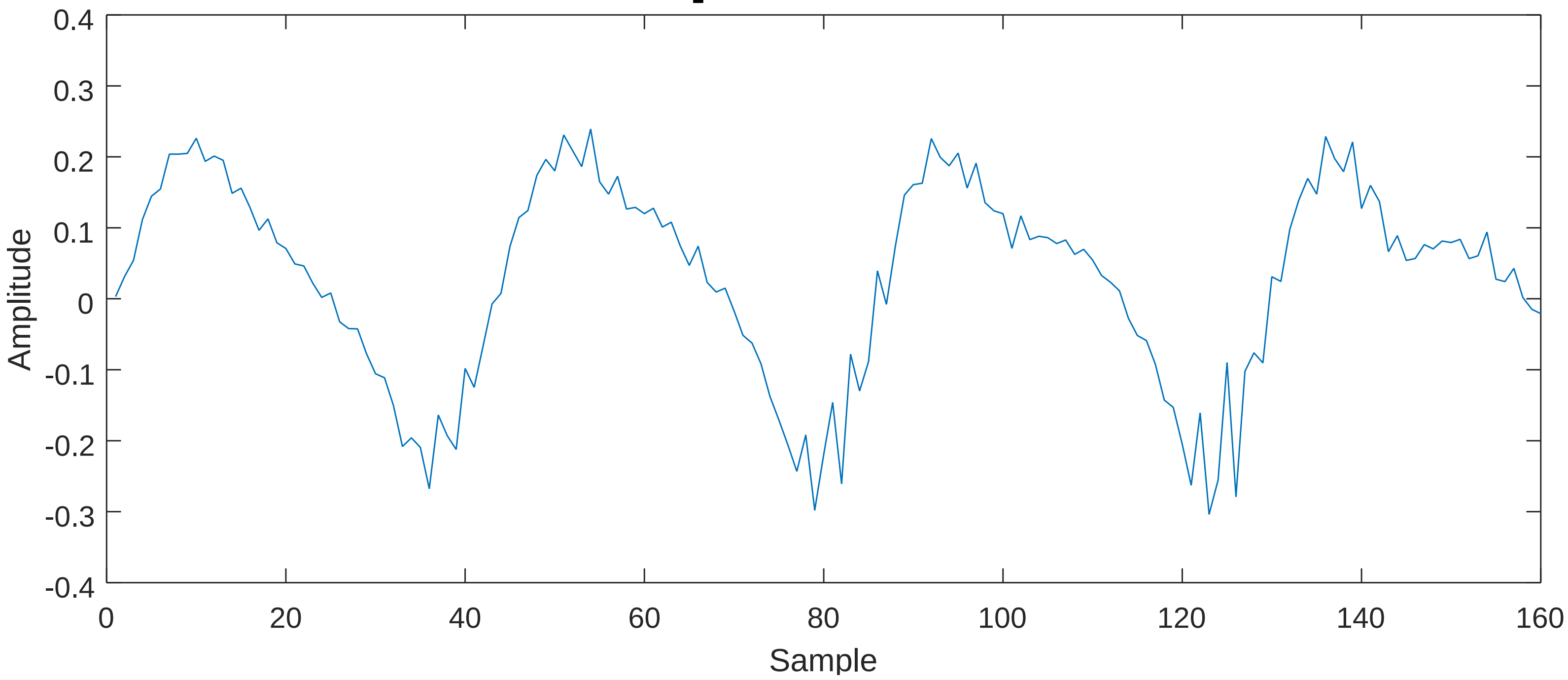


# Excitation and filtering

- The filter can be modeled with an all-pole digital filter of order *p*:

$$H(z) = \frac{1}{A(z)} = \frac{1}{1 + \sum_{k=1}^{p} a_k z^{-k}}$$

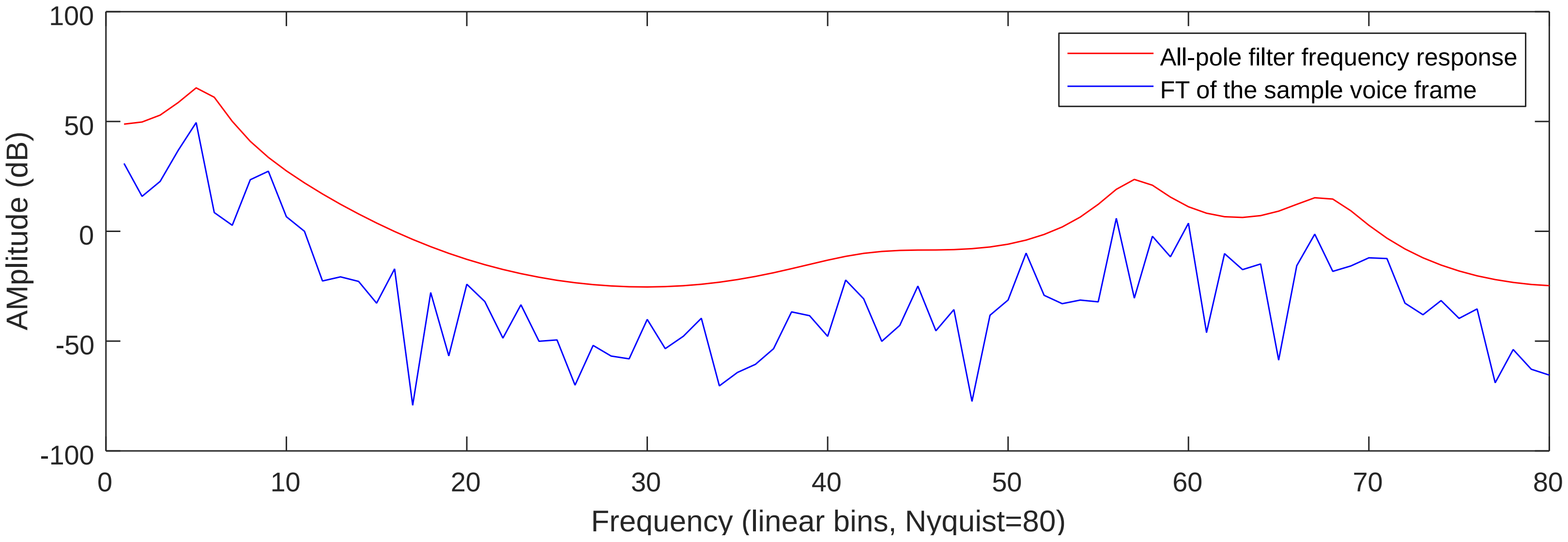
# 20 ms speech frame



# Excitation and filtering

- Codec 2 obtains the filter coefficients $a$ for each frame by computing $p+1=11$ autocorrelation values and solving a set of linear equations

# DFT vs. $|H(z)|$

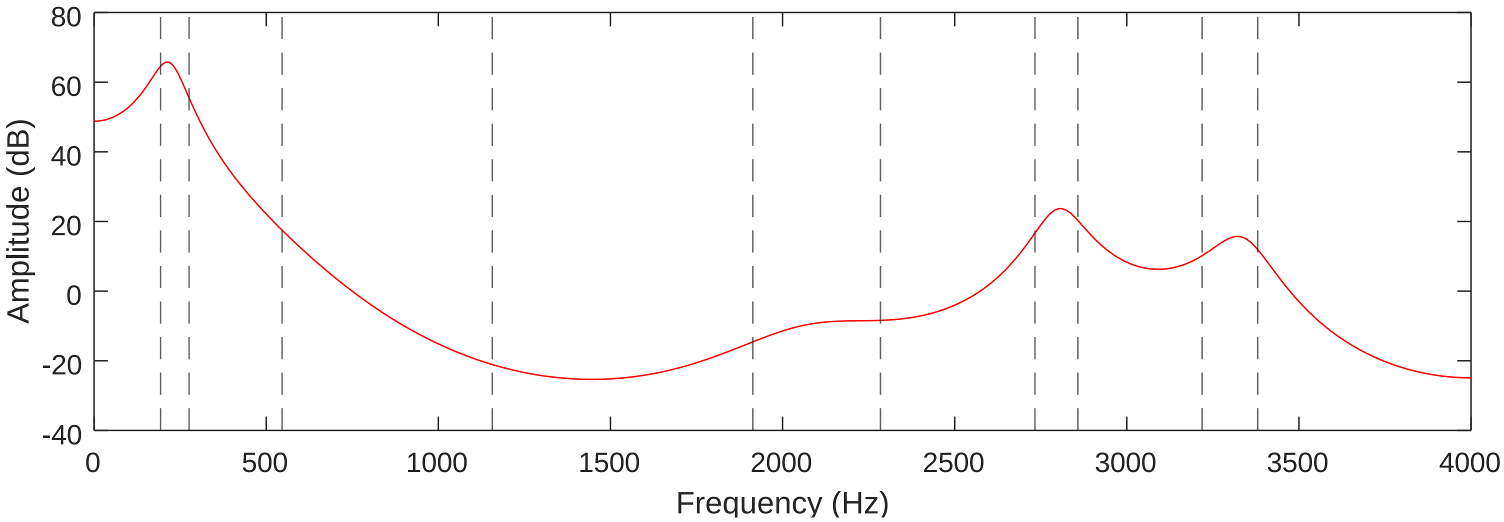


100
50
0
-50
-100
AMplitude (dB)
0
10
20
30
40
50
60
70
80
Frequency (linear bins, Nyquist=80)
All-pole filter frequency response
FT of the sample voice frame

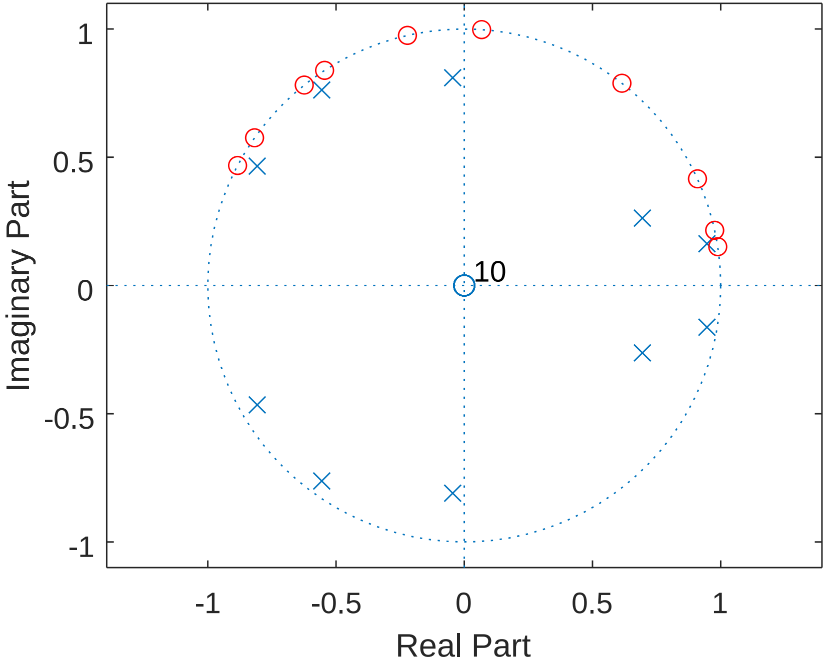
# Polynomial to LSF

- LSF – Line Spectral Frequency(ies)
- A slight error in any value *a* of the *A(z)* polynomial representation of the all-pole filter might be catastrophic
- Rationale for the conversion: LSF representation is much less susceptible to errors

# Polynomial to LSF



# Polynomial to LSF



# LSF encoding

- Codec 2 uses quantized delta encoding for the LSF parameters
- LSF1 is mapped ‘as-is’
- Every next LSF is encoded as a frequency shift from the previous one

# LSF encoding

- Eg. LSF1=145 Hz, the best entry is then 150

dlsp1
25
50
75
100
125
150
175
200

# LSF encoding

- LSF2=560 Hz, 560-150=410

| dlsp1 | dlsp2 |
|---|---|
| 25 | 300 |
| 50 | 325 |
| 75 | 350 |
| 100 | 375 |
| 125 | 400 |
| 150 | 425 |
| 175 | 450 |
| 200 | 475 |

# LSF encoding

- This process continues up to the last LSP

| dlsp1 | dlsp2 | | dlsp10 |
|---|---|---|---|
| 25 | 300 | | 400 |
| 50 | 325 | | 425 |
| 75 | 350 | | 450 |
| 100 | 375 | . . . | 475 |
| 125 | 400 | | 500 |
| 150 | 425 | | 525 |
| 175 | 450 | | 550 |
| 200 | 475 | | 575 |

# Quantizer improvements

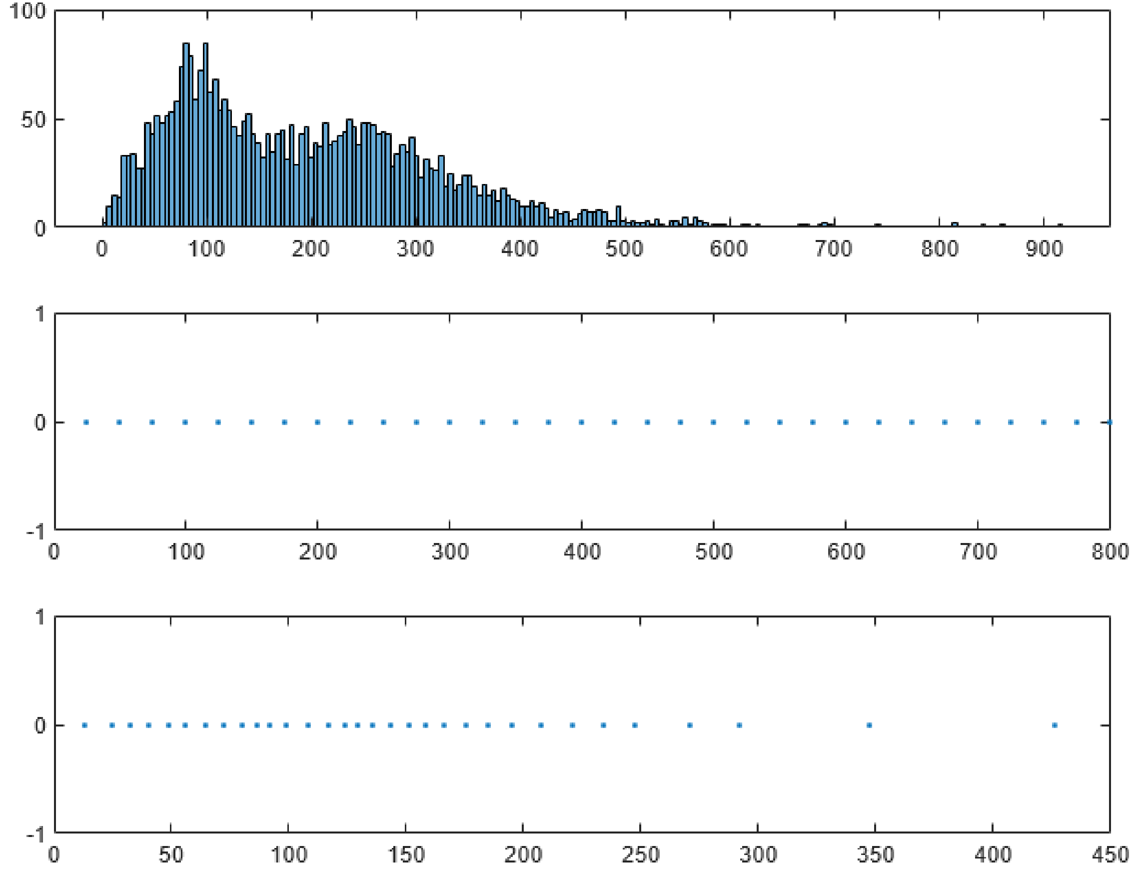
- Two approaches: improve the scalar quantizer – make quanta fit the actual data and its PDF
- Use vector quantizer instead

# Codebook generation

- OpenSLR - free speech corpora for codebook generation
- TED-LIUM(v2) has been successfully used for OpenACELP

# Codebook generation

- Centroid search – Linde-Buzo-Gray (LBG) or k-means algorithm

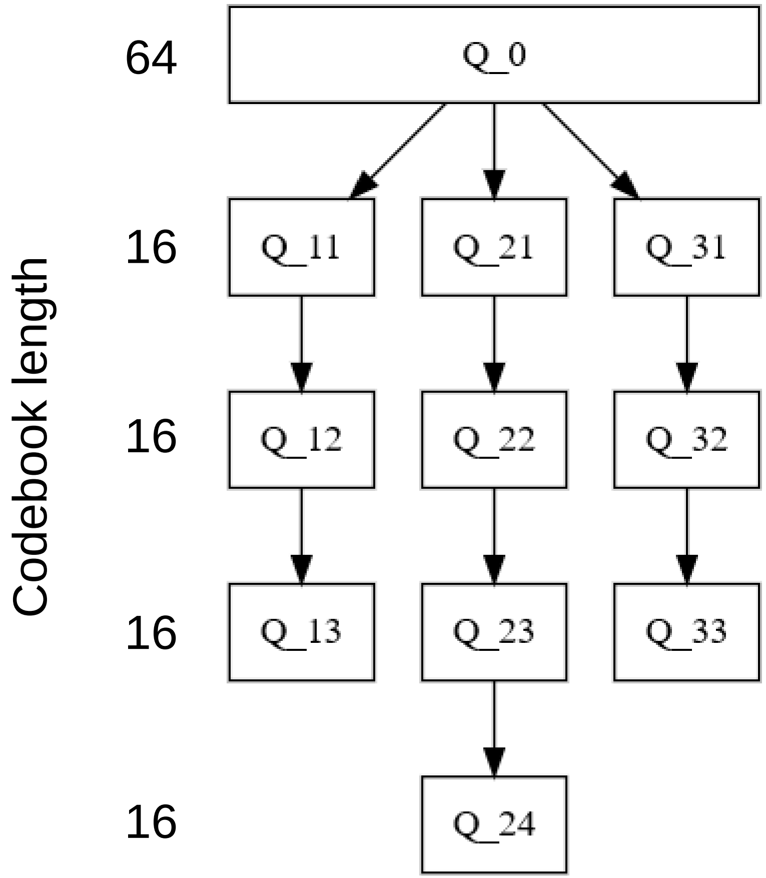

PDF
LSF2-LSF1 difference distribution
Codec 2 reference codebook
PDF optimized codebook
Frequency (Hz)

# Vector quantizer

- Proposal: use Multistage Vector Quantizer with Vector Split
- Use a coarse codebook for all 10 LSFs
- Split LSFs into subvectors of length {3,3,4} and use MSVQ to obtain better approximations of the input vector

# Vector quantizer

- Bit allocation for the quantizer:
  6 bits for the coarse codebook
  4 bits per stage for MSVQ
- Use 3 stages for LSF1..3, 4 for LSF4..6 and 3 for LSF7..10
- Total of 46 bits per frame (Codec 2 uses 50)

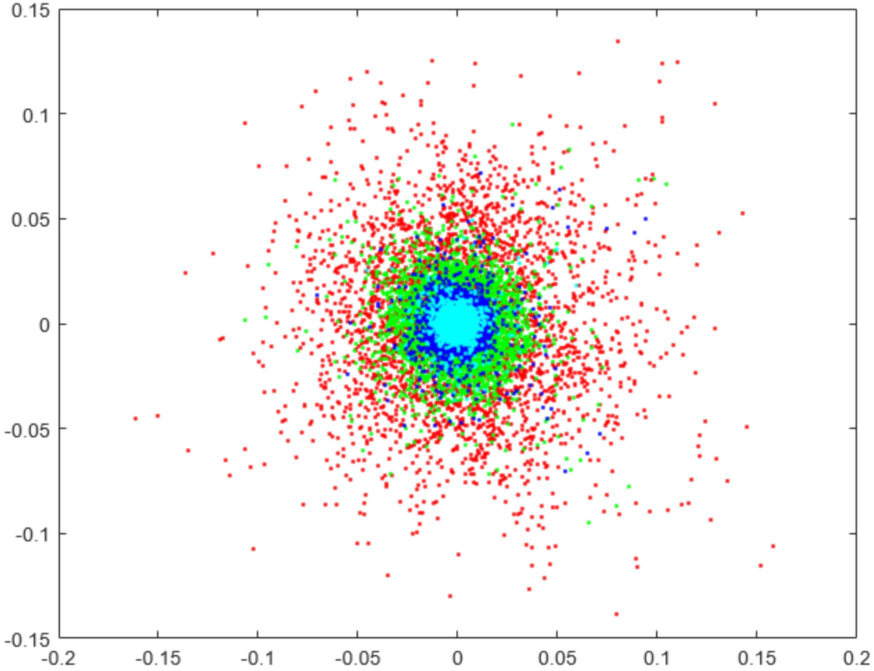

Codebook length
64
16
16
16
16
Q_0
Q_11
Q_21
Q_31
Q_12
Q_22
Q_32
Q_13
Q_23
Q_33
Q_24

# Vector quantizer

- The resulting reconstructed vector is given by

$$v = Q_0(i_0) + \sum_{k=1}^{n_1} Q_{1k}(i_{1k}) + \sum_{k=1}^{n_2} Q_{2k}(i_{2k}) + \sum_{k=1}^{n_3} Q_{3k}(i_{3k})$$

# MSVQ convergence

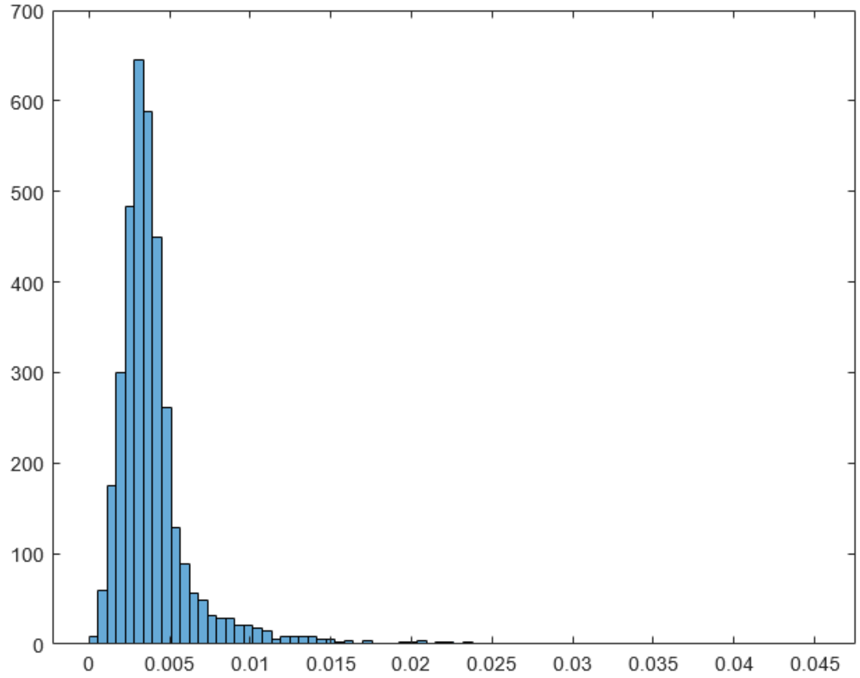



2D cross sections of the training sets used for Q_2x codebook generation

Axes: LSFs in cosine domain

# MSVQ error distribution

PDF




Error distribution for the Q_2 codebook

Horizontal axis: absolute error bins, Euclidean distance

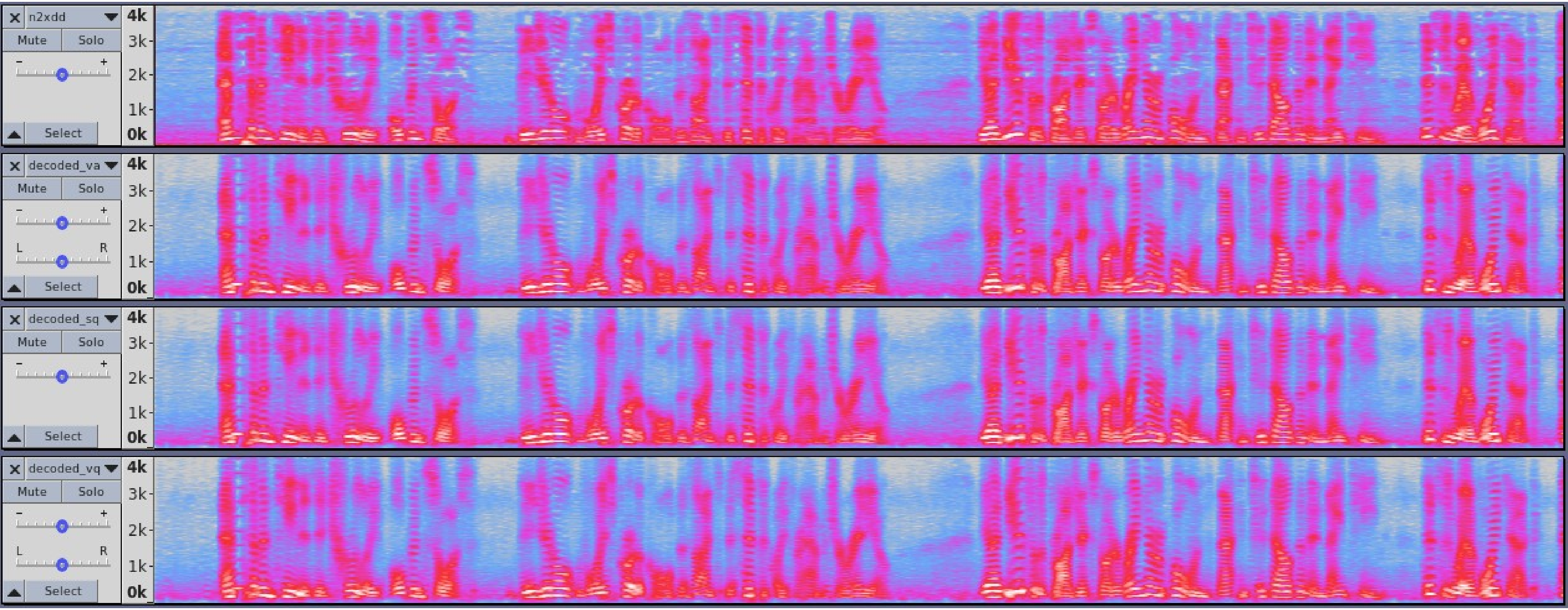
$$PDF\left(\|v - v_{inp}\|_2\right)$$

# Improving SQ - results

- Computational complexity is the same
- SD is reduced, but the change is likely to be imperceptible
- Bitrate doesn’t change

# Switch to VQ - results

- Saves 200 bps
- Cost – more multiplications to perform for the codebook search
- SD rises, but the change should be negligible (assuming a simple handheld transceiver is used – low quality reproduction)



n2xdd
Mute
Solo
Select
4k
3k
2k
1k
0k
decoded_va
Mute
Solo
L
R
Select
4k
3k
2k
1k
0k
decoded_sq
Mute
Solo
Select
4k
3k
2k
1k
0k
decoded_vq
Mute
Solo
L
R
Select
4k
3k
2k
1k
0k

# Thank you