Codeword Error Rate and Hyper-Spherical TDECQ

Ahmad El-Chayeb – Keysight Technologies
David Leyba – Keysight Technologies

Addressing comment #1 against IEEE P802.3dj D2.0

Current definition for TDECQ points to clause 121.8.5.1 where TDECQ is calculated at a pre-FEC target SER. This definition is not a very good indicator of link performance   Re-define TDECQ and extend it to CER (codeword error ratio) to have better correlation with link performance. CER TDECQ definition need to be technically and economically feasible. A subsequent presentation will be provided at a later ad-hoc meeting.

Supporters

- Adee Ran, Cisco
- Ali Ghiasi, Ghiasi Quantum/Marvell
- Brian Welch, Cisco
- Hai-Feng Liu, HG Genuine
- John Calvin, Keysight Technologies
- John Johnson, Broadcom
- Marco Mazzini, Cisco
- Mark Kimber, Semtech
- Mike Dudek, Marvell
- Vasudevan Parthasarathy, Broadcom

AGENDA

1. Introduction
2. TDECQ in the Current Draft (IEEE P802.3dj D2.0)
3. Codeword Error Rate TDECQ
4. A Geometric Perspective – Rectangular TDECQ
5. Approximating CER TDECQ – Hyper-Spherical TDECQ
6. Experimental Data
7. Summary

AGENDA

1. Introduction
2. TDECQ in the Current Draft (IEEE P802.3dj D2.0)
3. Codeword Error Rate TDECQ
4. A Geometric Perspective – Rectangular TDECQ
5. Approximating CER TDECQ – Hyper-Spherical TDECQ
6. Experimental Data
7. Summary

Introduction

- The current definition of TDECQ is calculated at a theoretical pre-FEC target SER that assumes errors are uncorrelated and randomly distributed.
- TDECQ is intentionally pessimistic and is not intended, as defined, to correlate to link performance (BER/FLR) as demonstrated by chayeb_3dj_01_2505.
- chayeb_3dj_01_2505 documented some of the challenges of TDECQ and potential areas for further study.
- chayeb_3dj_01_2505 and ghiasi_3dj_03_2501 proposed a Codeword Error Rate TDECQ calculated at a target CER (codeword error rate) instead of the current SER (symbol error rate) target.
- This presentation proposes a Codeword Error Rate TDECQ that can be implemented on real-time and sampling oscilloscopes, as well as an approximation method that speeds up the calculation of CER TDECQ
- This presentation proposes a technically sound and economically viable implementation for a codeword error rate TDECQ referenced in comment ***#1*** against IEEE P802.3dj D2.0.

AGENDA

1. Introduction
2. **TDECQ in the Current Draft (IEEE P802.3dj D2.0)**
3. Codeword Error Rate TDECQ
4. A Geometric Perspective – Rectangular TDECQ
5. Approximating CER TDECQ – Hyper-Spherical TDECQ
6. Experimental Data
7. Summary

TDECQ in the Current Draft (P802.3dj D2.0)

Goal of TDECQ: Estimate the power penalty of a transmitter by finding the maximum intrinsic receiver noise that still achieves the desired error performance (currently specified as SER).

The methodology specified in the standards:

- Combines samples from each sample location, into histograms which represent an estimated pdf of the signal levels at the sample location, $f_{y_k}(y_k)$
- Let $n \sim N(0, \sigma^2)$ be a normal gaussian, then the SER is a function of σ defined as:

$$SER(\sigma) = \sum_{l=0}^{l=3} \sum_{k} f_{y_k}(y_k) \left[Q\left(\frac{y_k - P_{th(l-1)}}{\sigma} \right) + Q\left(\frac{P_{th(l)} - y_k}{\sigma} \right) \right]$$

- The maximum σ is found such that $SER(\sigma) \leq SER_{target}$

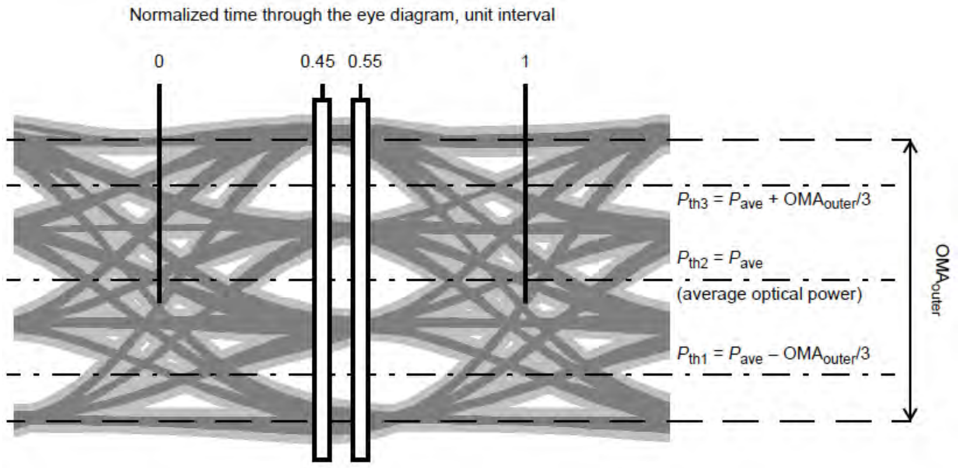


Figure 121–5—Illustration of the TDECQ measurement

IEEE 802.3bs – Clause 121.8.5.1

AGENDA

1. Introduction
2. TDECQ in the Current Draft (IEEE P802.3dj D2.0)
3. **Codeword Error Rate TDECQ**
4. A Geometric Perspective – Rectangular TDECQ
5. Approximating CER TDECQ – Hyper-Spherical TDECQ
6. Experimental Data
7. Summary

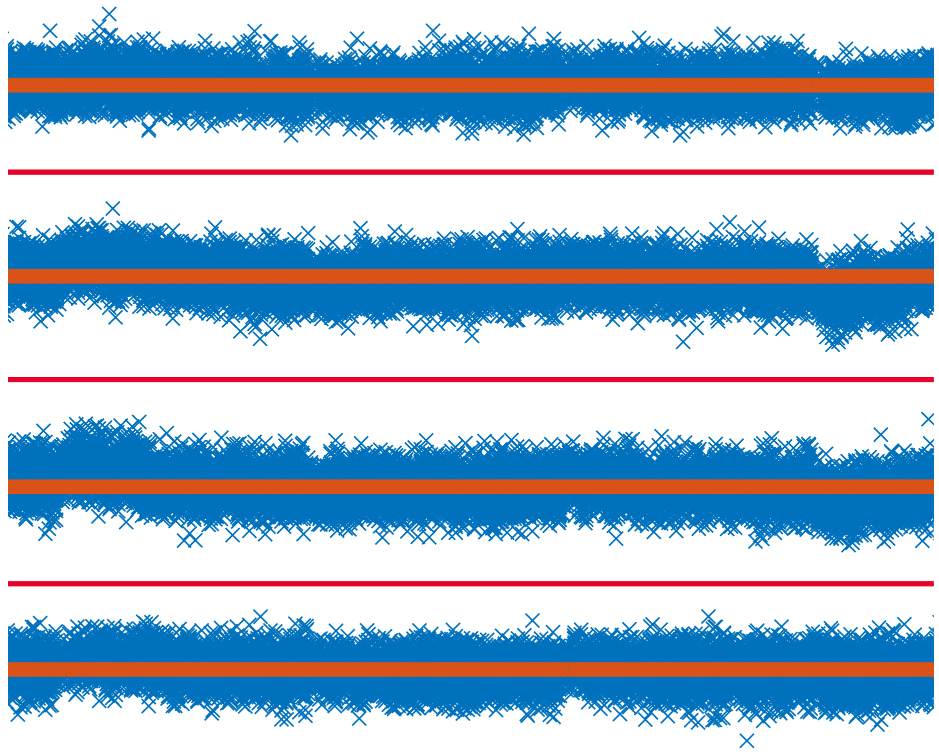
Codeword Error Rate TDECQ

Unwrapping the Histogram – Calculating the probability of error for each sample

- An alternative, but less efficient, methodology is to unwrap the level histograms as follows:

- Let:
  - i be the symbol index
  - x_i a sample from the target region of the symbol
  - l_i the symbol value (0,1,2,3)
  - N the number of symbols
  - $V(l_i)$ is the nominal level of the symbol
  - th_{l_i} is the optimal threshold below the l_i level

- The figure shows x_i in blue, $V(l_i)$ in orange, and th_{l_i} in red.



Codeword Errors

Unwrapping the Histogram

- We can define $SER(\sigma) = \frac{1}{N}\sum_{i=0}^{i=N-1}\left[Q\left(\frac{x_i - th_{l_i-1}}{\sigma}\right) + Q\left(\frac{th_{l_i} - x_i}{\sigma}\right)\right]$

- The above can be dissected to get a probability of error per symbol i assuming σ:

  - $P_{err,i}(\sigma) = SER_i(\sigma) = \left[Q\left(\frac{x_i - th_{l_i-1}}{\sigma}\right) + Q\left(\frac{th_{l_i} - x_i}{\sigma}\right)\right]$

  - $P_{err,i}(\sigma)$ is then the probability of error for the specific instance of x_i assuming receiver noise power σ

  - Each x_i includes ISI, transmitter and scope noise.

- If symbols are organized into codewords of length d and we assume that the FEC can correct up to K errors a probability of codeword error for a given σ can be calculated using $P_{err,i}(\sigma)$

Codeword Errors

Probability of k Errors in a Codeword of length *d*, $C_{k,d}(\sigma)$

- For a set of d symbols, $P_{err,i}(\sigma)$ is the corresponding probability of error of the ith symbol
- The number of errors in the codeword follows a Poisson Binomial distribution
  - An extension of the Binomial distribution, where the probabilities of each trial are not identical
- This distribution, $C_{k,d}(\sigma)$, can be calculated by convolving the individual symbol PMF's together:
  - The ith symbol has the PMF: $p_i(k) = \begin{cases} 1 - P_{err,i}(\sigma) & k = 0 \\ P_{err,i}(\sigma) & k = 1 \end{cases}$, where 1 indicates an error
  - $C_{k,d}(\sigma) = p_0(k) * p_1(k) * \cdots * p_{d-1}(k)$
- Once the PMF is calculated, the probability of a correctable codeword is found by summing the first K entries of $C_{k,d}(\sigma)$
- And the probability that the jth codeword is in error is:

$$P_{fail,j}(\sigma)=1-\sum_{k=0}^{K-1} C_{k,d}(\sigma)$$

Codeword Errors (con't)

- A d symbol codeword can be made up of any d distinct symbols from the pattern
  - ➢ There will be $M = \left\lfloor \frac{N}{d} \right\rfloor$ available blocks
  - ➢ The Codeword Error Rate is then the mean of $P_{fail,j}(\sigma)$ over all blocks:

$$CER(\sigma) = \frac{1}{M} \sum_{j=0}^{M-1} P_{fail,j}(\sigma)$$

- The codeword error rate can calculated for both inner FEC and outer FEC codewords.

Codeword Error Rate Complexity

- Assuming an inner FEC codeword that is 64 symbols long
  - An SSPRQ is $2^{16} - 1$ symbols
  - So, at least $M = 1023$ codewords need to be evaluated
  - Each codeword PMF requires 63 convolutions given a specific σ value
  - So, for each iteration using a σ value 64,449 convolutions are required

- The existing TDECQ method requires 1 convolution per σ value
  - There is additional computation to convert samples into histograms

- The complexity of the algorithm to exactly compute the Codeword Error Rate appears to be daunting

AGENDA

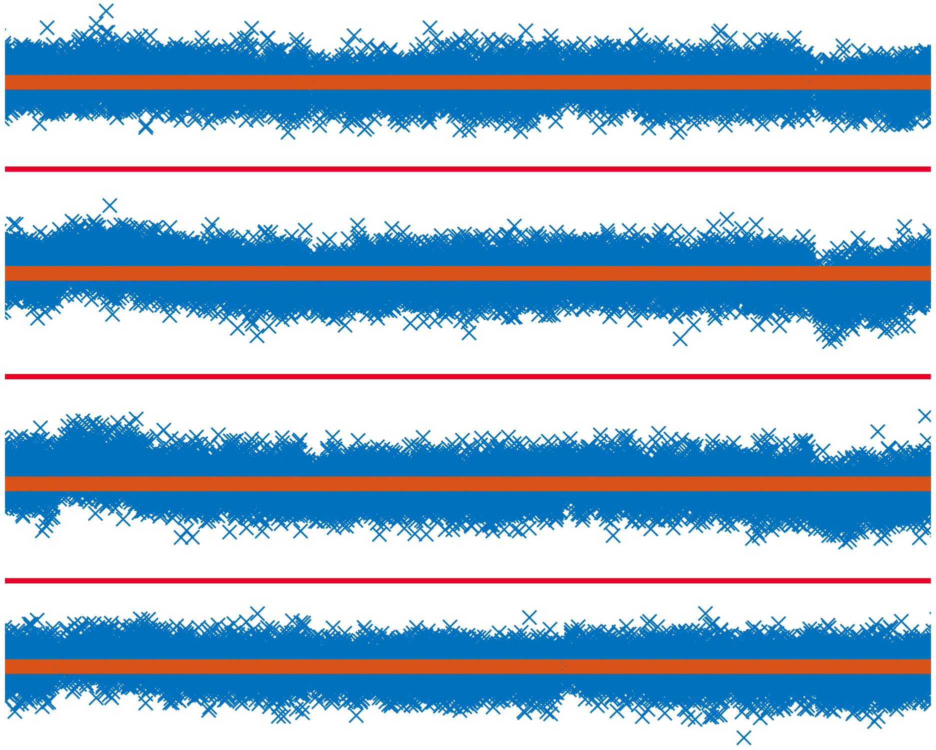
1. Introduction
2. TDECQ in the Current Draft (IEEE P802.3dj D2.0)
3. Codeword Error Rate TDECQ
4. **A Geometric Perspective – Rectangular TDECQ**
5. Approximating CER TDECQ – Hyper-Spherical TDECQ
6. Experimental Data
7. Summary

A Geometric Perspective – Rectangular TDECQ

Treat codewords as vectors in a d-dimensional space

- Let:
  - i be the symbol index
  - x_i a sample from the target region of the symbol
  - l_i the symbol value (0,1,2,3)
  - N the number of symbols
  - $V(l_i)$ is the nominal level of the symbol
  - th_{l_i} is the optimal threshold below the l_i level

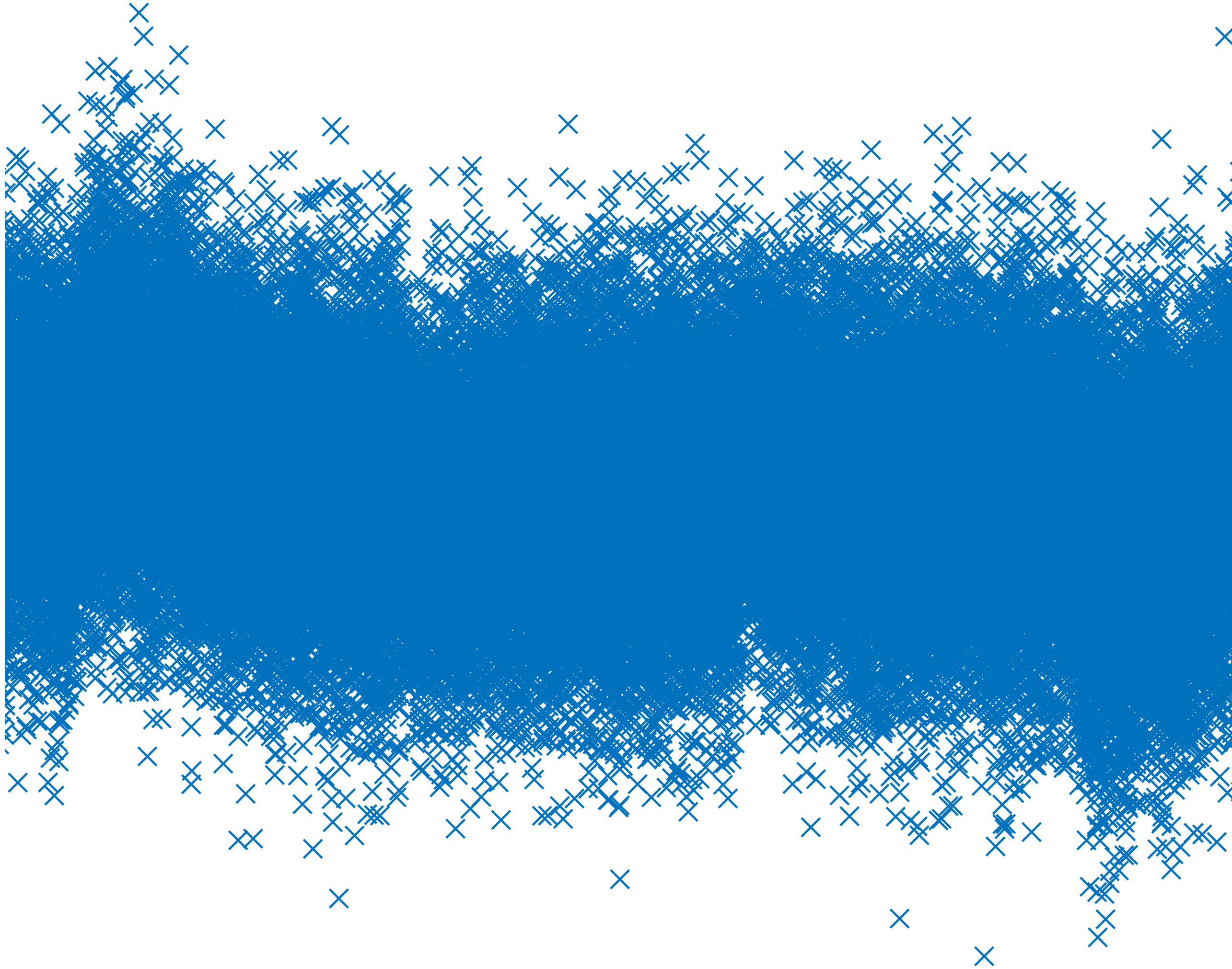
- The figure shows x_i in blue, $V(l_i)$ in orange, and th_{l_i} in red.



Defining the Residual Vector

Center analysis at the origin

- Define the residual, $\mu_i = x_i - V(l_i)$
- We can group the residuals to create a d-dimensional residual vector,
  - eg $\bar{\mu} = [\mu_0, \mu_8, \ldots \mu_{8(d-1)}]$
- The above example uses the interleaving of 8 symbol as specified in the standard for FECi.



A Geometric Perspective

The Noise Vector

- Let $\bar{n} \sim N(0, \Sigma)$ be a d-dimensional vector of the noise to be added, Σ is the covariance matrix of the added noise.
  - ➢ Assumed to be AWGN filtered by a 4th Order Bessel Thompson and any specified equalizers

- Combining the residual vector and noise vector results in the random variable $\bar{y} = \bar{\mu} + \bar{n}$
  - Where the residual is the mean and having a pdf, $f_{\bar{y}}(\bar{y}) = \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} e^{-\frac{1}{2}(\bar{y}-\bar{\mu})^T \Sigma^{-1} (\bar{y}-\bar{\mu})}$

- **Note:** If the data samples come from sufficiently separated UI and only linear equalizers are used then independence of noise samples can be assumed, i.e. $\Sigma \approx \sigma^2 \boldsymbol{I}$, where $\boldsymbol{I}$ is the identity matrix.

Probability of k Errors

- Can “easily” calculate the probability of 0 errors as:

$$P(0\ errors) = \int_R \frac{1}{\sqrt{(2\pi)^d|\Sigma|}} e^{-\frac{1}{2}(\bar{y}-\bar{\mu})^T\Sigma^{-1}(\bar{y}-\bar{\mu})} d\bar{y}$$

  - where R is a d-dimensional region centered at 0, where each dimension is bound by the upper and lower threshold (or +/- infinity) for that symbol.
  - **NOTE: Each dimension has its own limit based on l_i**

- For k>0, things are much more complicated
  - Need to take all regions R such that k dimensions are outside the thresholds and the other d-k dimensions are inside their respect thresholds.
  - If $\Sigma \approx \sigma^2 \boldsymbol{I}$ this results in the same results as the previous section

- For simplicity of analysis, assume that all thresholds are $\pm\frac{OMA}{6}$ from the nominal symbol level
  - Now, all symbol errors can be thought of as either too high or too low.

Probability of k Errors

Two-Dimensional Visualization

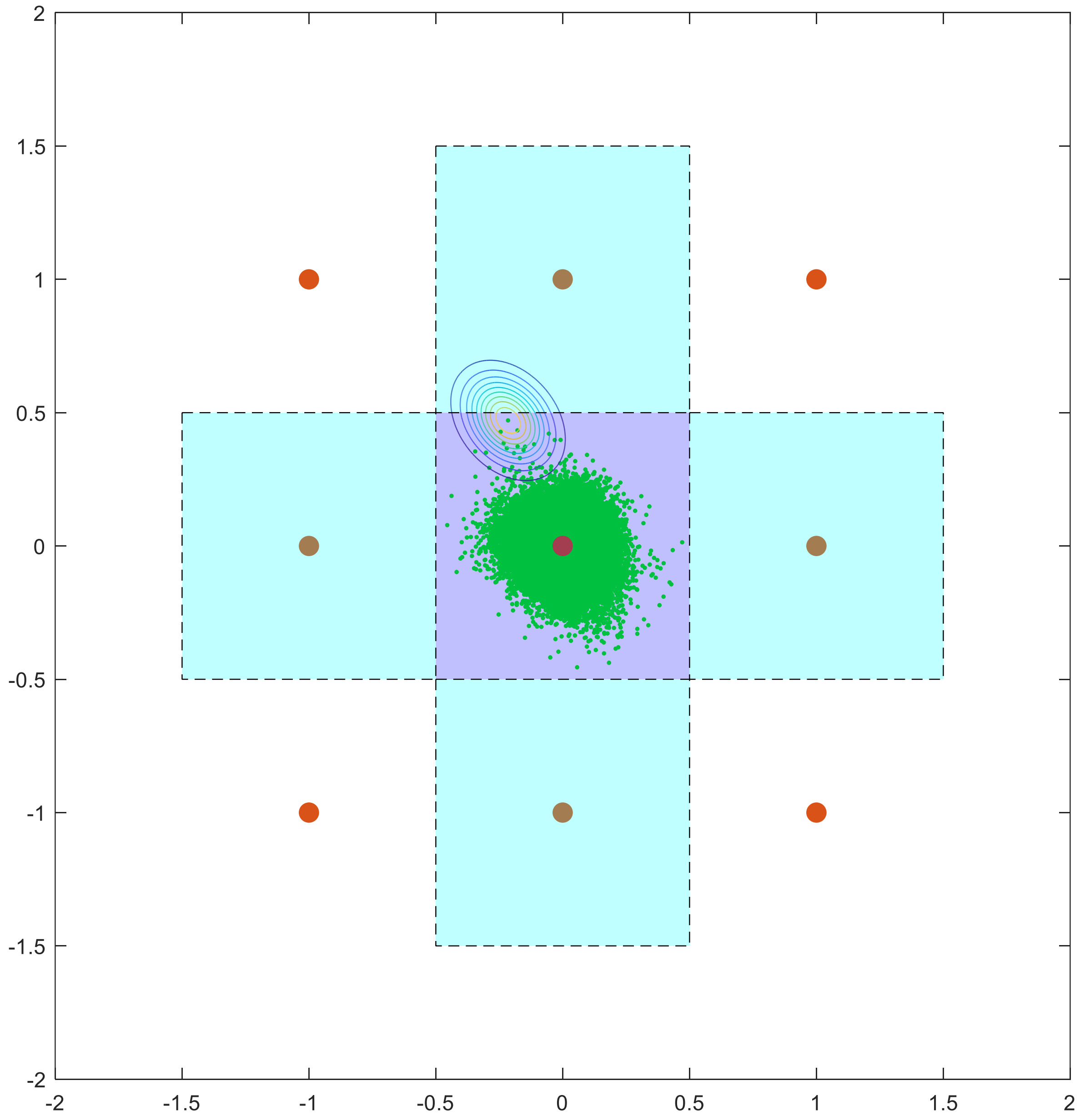
- The probability of exactly k errors is:

$$p_k = \sum_{c \in C_d(k)} \int_c \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} e^{-\frac{1}{2}(\bar{y}-\bar{\mu})^T \Sigma^{-1} (\bar{y}-\bar{\mu})} d\bar{y}$$

- Where $C_d(k)$ is a d-dimensional hypercube over the codeword space with exactly k symbol errors.

- Then the probability of k or more errors is

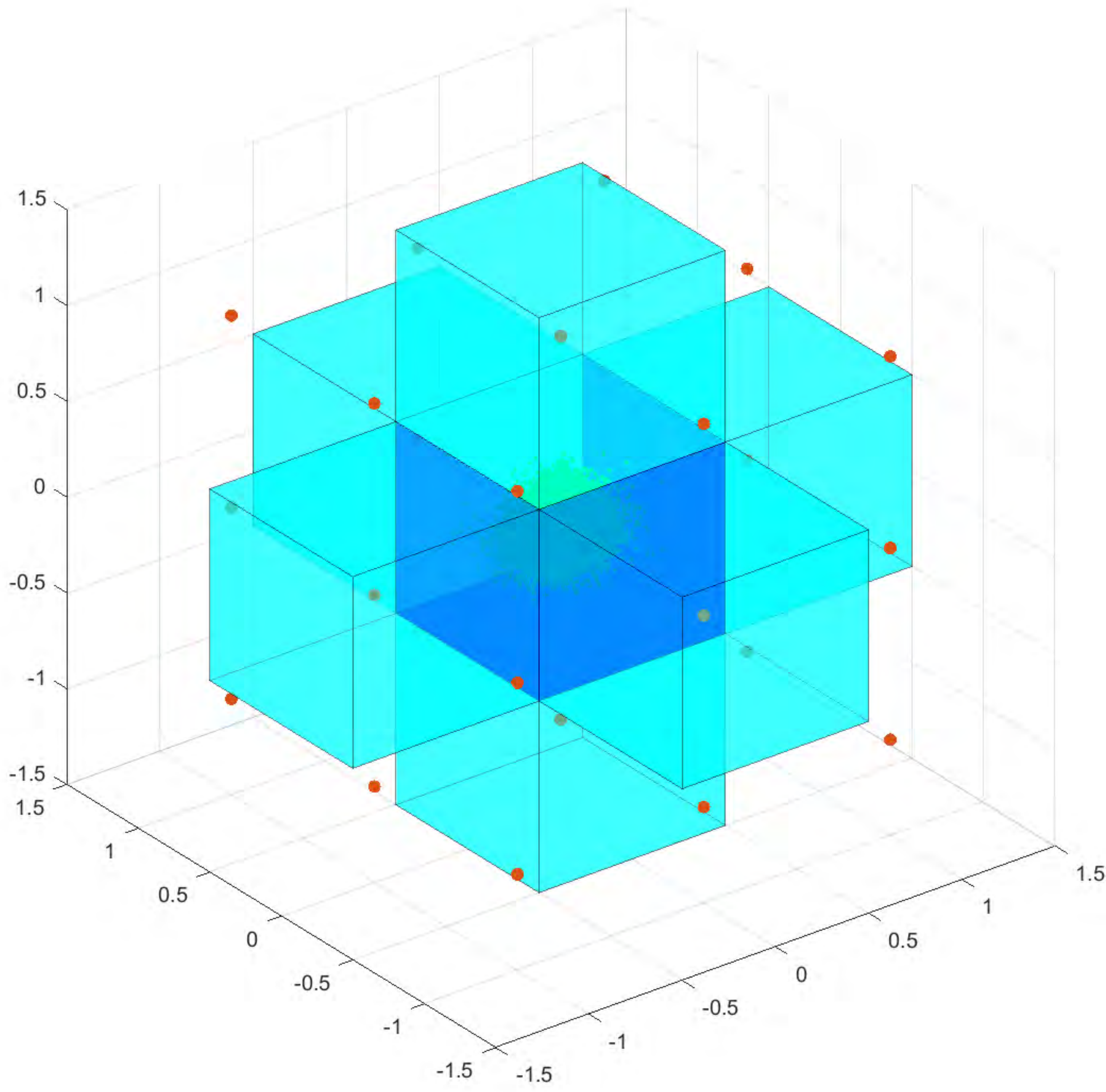
$$P_{fail} = 1 - \sum_{j=0}^{k-1} \mathrm{p_k}$$



Probability of k Errors

Three-Dimensional Visualization

- An example of our simplified model using just d=3 length codewords.
- The Red dots are the potential codewords
- Blue box indicates region of 0 errors per code word
- Cyan boxes indicate region with 1 error per code word
- Green dots are the residuals, $\bar{\mu}$, from data samples



KEYSIGHT

Regions with k Errors

64-symbol Codeword Example - Inner FEC

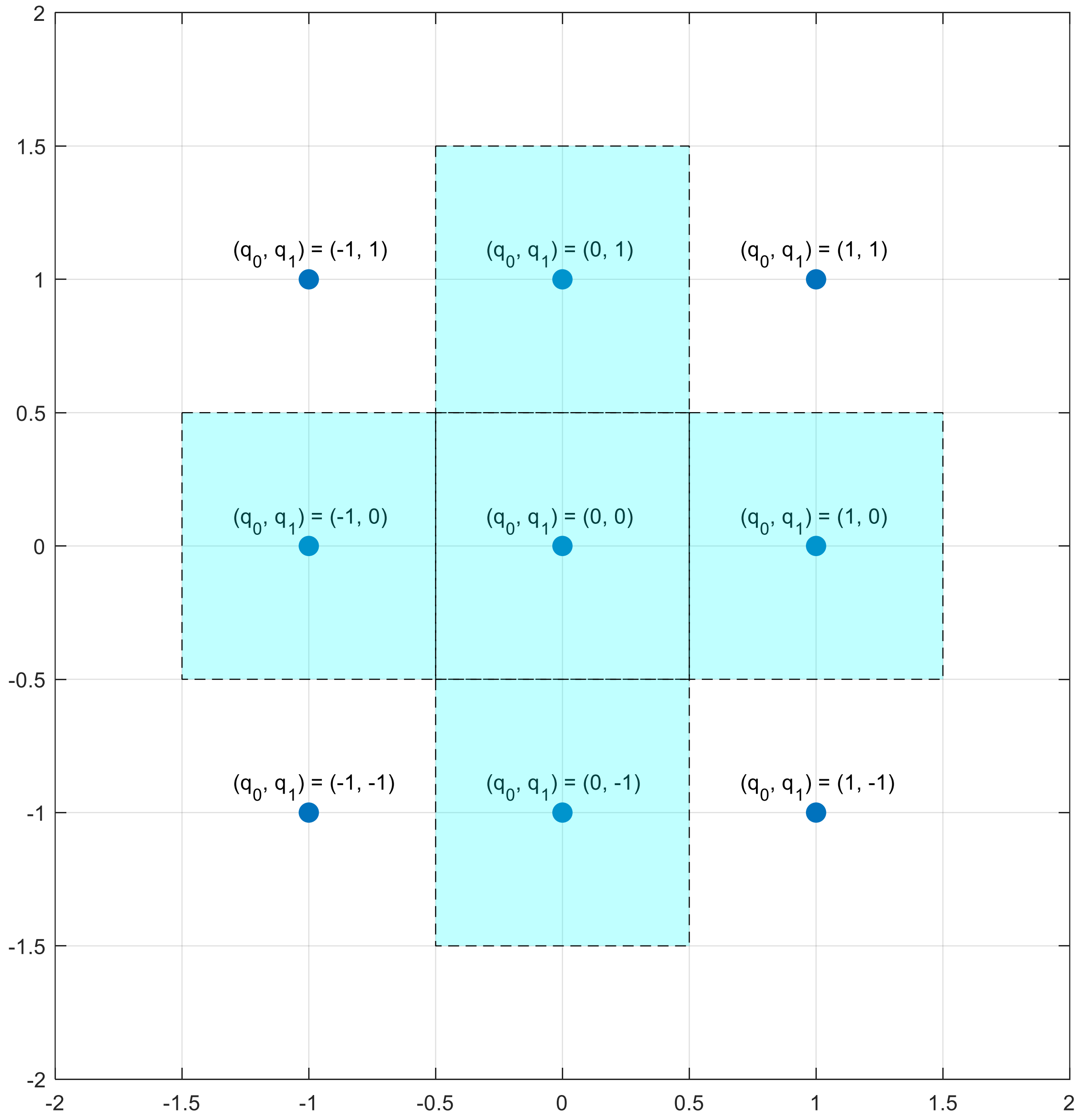
- Define a set of d-dimensional hypercubes, $C_d(k)$, all with sides of length $\frac{OMA}{3}$
- Each region is centered at the point

$$\frac{OMA}{3}[q_0, q_1, \ldots q_{d-1}], q_i \in \{-1,0,1\}$$

  where,

$$\sum_{i=0}^{d-1} |q_i| = k$$

- The number of such regions is $|C_d(k)| = 2^k \binom{d}{k}$
- 64 symbols and up to 3 errors results in over 333312 regions
- This appears to be an even more complex process



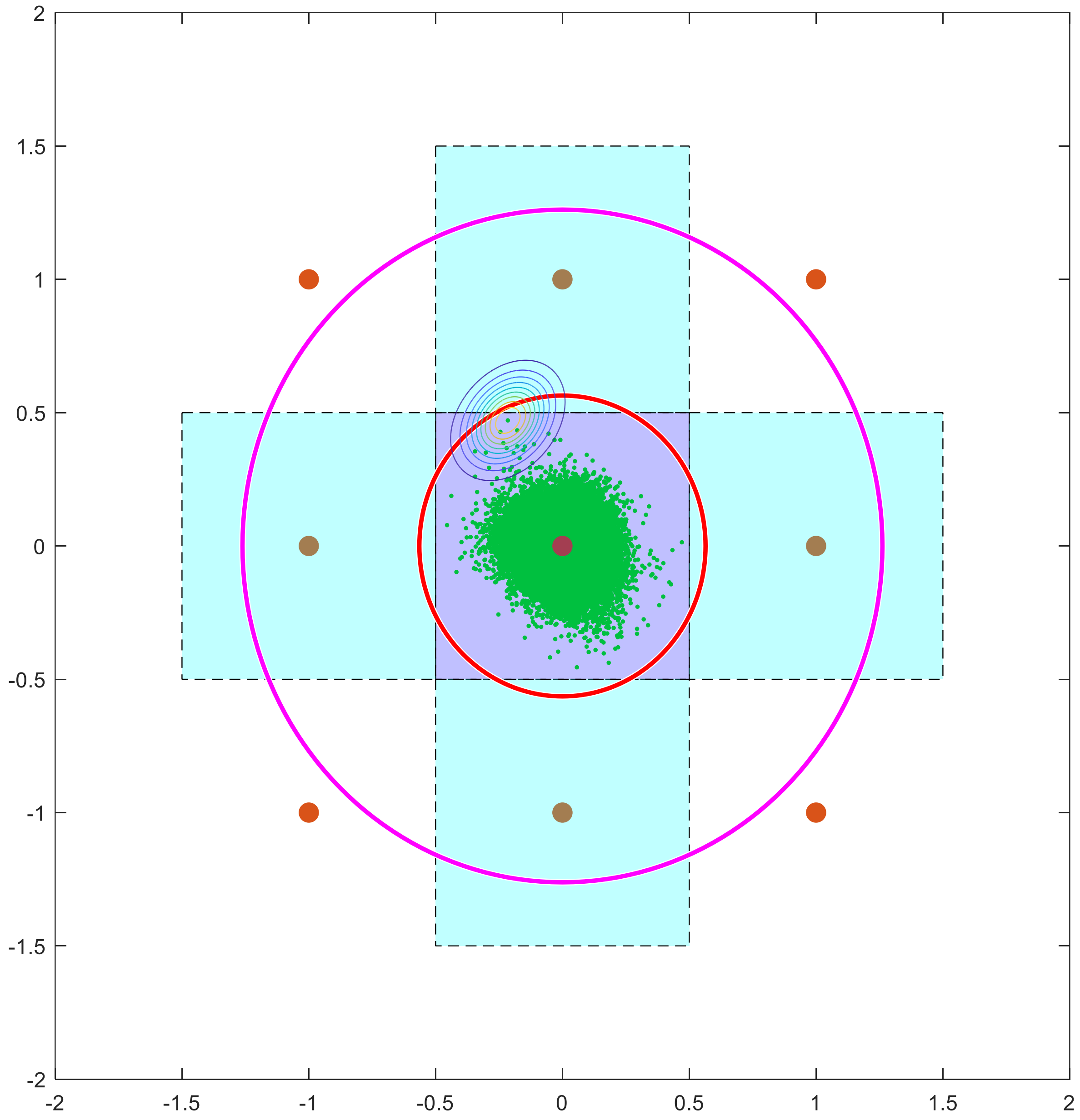
AGENDA

1. Introduction
2. TDECQ in the Current Draft (IEEE P802.3dj D2.0)
3. Codeword Error Rate TDECQ
4. A Geometric Perspective – Rectangular TDECQ
5. **Approximating CER TDECQ – Hyper-Spherical TDECQ**
6. Experimental Data
7. Summary

Can We Make it simpler?

Going Back to the Two-Dimensional Example

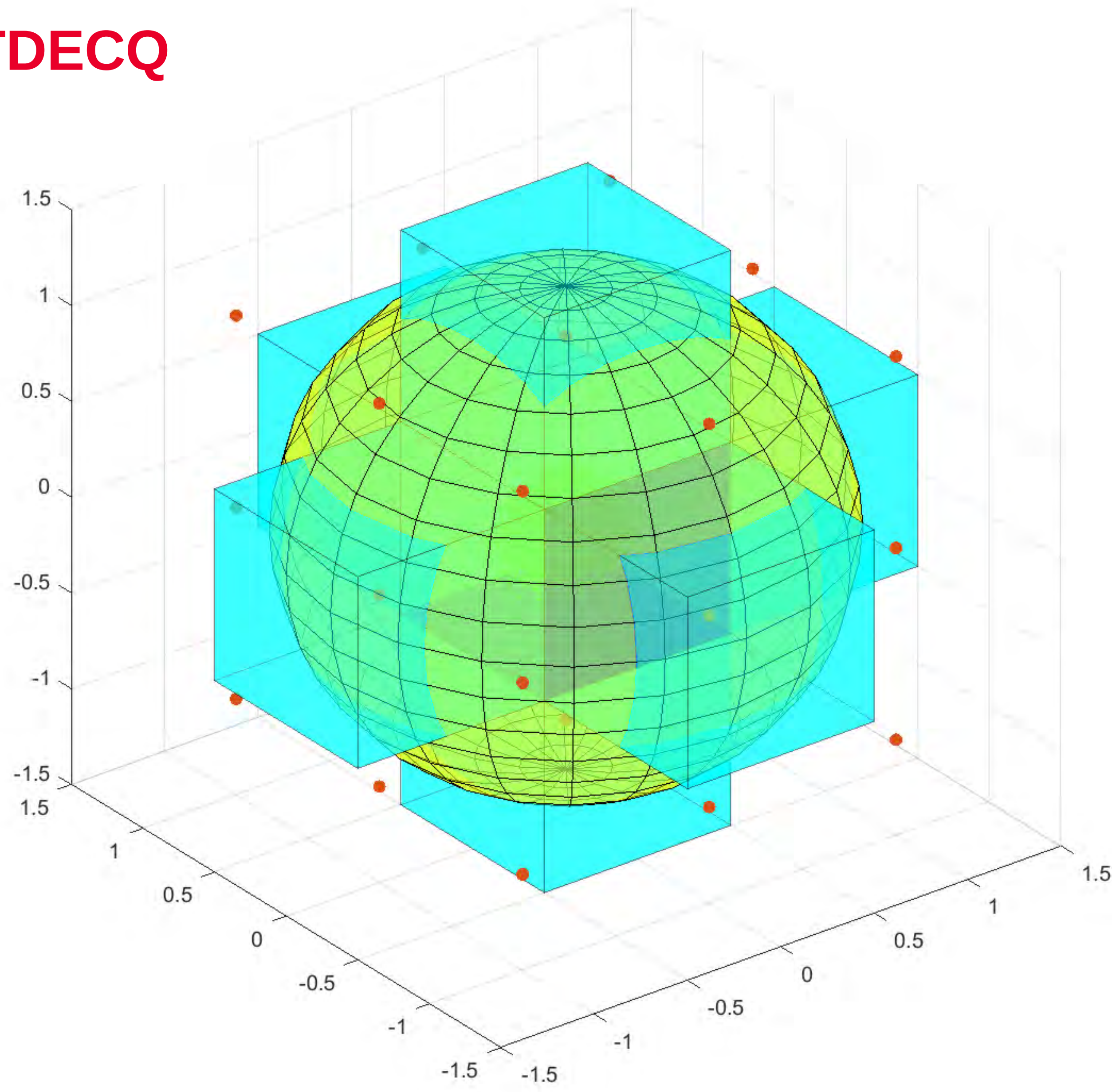
- Approximate the set of hypercubes as a single hypersphere
- Focus instead on the Euclidean distance from the ideal codeword location
- An estimate for the codeword error rate could be the probability that $\bar{y} = \bar{\mu} + \bar{n}$ is contained within a sphere of radius r_k centered at the ideal codeword location
- Essentially, CER $\approx 1 - \Pr\{\bar{y}^T \bar{y} \le r_k^2\}$



Approximating Codeword Error Rate TDECQ

Reduction to a Singular Dimension

- Define a new variable: $r^2 = \frac{\bar{y}^T \bar{y}}{\sigma^2}$
- For the case where $\bar{y} \sim N(\bar{\mu}, \Sigma)$, r^2 follows a generalized χ^2-distribution
- If $\Sigma \approx \sigma^2 \boldsymbol{I}$, then r^2 follows the non-central χ^2-distribution with d degrees of freedom and non-centrality parameter $\lambda = \frac{\bar{\mu}^T \bar{\mu}}{\sigma^2}$
- With cdf $F_{r^2}(r^2; d, \lambda)$
- Then an approximation for the probability of at most k errors per codeword can be written as: $F_{r^2}\left(\frac{r_k^2}{\sigma^2}; d, \lambda\right)$. Where r_k is the radius of the circle representing at most k errors.



An Algorithm

(Hyper)-Spherical TDECQ

- Let r_k be the target radius for up to k errors per codeword
- Let $\bar{\mu}_i$ be the ith vector of sampled residuals out of N total vectors
- Define the probability of a correctable codeword for a given σ as:

$$G(\sigma) = \frac{1}{N}\sum_{i=0}^{N-1} F_{r^2}\left(\frac{r_k^2}{\sigma^2}; d, \frac{\bar{\mu}_i{}^T \bar{\mu}_i}{\sigma^2}\right)$$

  - Find the maximum noise, σ_g, such that $\left(1 - \mathrm{G}\left(\sigma_g\right)\right) \leq CER_{target}$
  - Calculate $\sigma_s = \sqrt{C_{eq}^{-2}\sigma_g^2 + \sigma_c^2}$ to remove noise gain and include intrinsic channel noise

- $TDECQ = 10\log_{10}\left(\frac{\sigma_{ref}}{\sigma_s}\right)$, where σ_{ref} is the noise margin of an ideal transmitter
- Note: For d=1, this can be reduced to an equivalent calculation of traditional TDECQ

Additional Considerations

Choice of $\boldsymbol{r_k}$

- Choose $\boldsymbol{r_k}$ such that $1 - CER_{target} = F_{r^2}\left(\frac{r_k^2}{\sigma_{ref}^2}; d, 0\right)$
  - The probability of a successful codeword is equal to the probability of being within the hypersphere of radius $\frac{r_k^2}{\sigma_{ref}^2}$
  - $r_k = \sigma_{ref}\sqrt{F_{\chi_d}^{-1}(1 - CER_{target})}$

- σ_{ref} is calculated similarly to the existing method, assuming an idealized system
  - Assume symbol errors are iid with error probability P_e
  - $\sigma_{ref} = \frac{OMA}{6\sqrt{2}\,\mathrm{erfc}^{-1}\left(\frac{4}{3}P_e\right)}$
  - P_e can be derived from CER_{target} as the solution to $1 - CER_{target} = \sum_{i=0}^{k}\binom{n}{i}P_e^i(1 - P_e)^{n-i}$

Additional Considerations

Efficient Calculation of Non-Central Chi-Square CDF

- The non-central Chi-Square CDF, $F_{r^2}(r^2; d, \lambda)$, does not have a closed form solution
- “Exact” Numerical solutions exist, but are computationally complex
- Tables could be generated for pairs of values, (r, λ), to speed up computations
- For calculations in this presentation the following estimate[1] is used:

$$F_{r^2}(r^2; d, \lambda) \approx \frac{1}{2}\operatorname{erfc}\left[\frac{1}{\sqrt{2}}\left(r - \sqrt{\lambda} - \frac{d-1}{2\left(r - \sqrt{\lambda}\right)}\left(\log(r) - \log\left(\sqrt{\lambda}\right)\right)\right)\right]$$

1 - Fraser, D.A.S. & Wu, Jianrong & Wong, Augustine. (1998). An approximation for the noncentral chi-squared distribution. Communications in Statistics - Simulation and Computation. 27. 275-287. 10.1080/03610919808813480.

What does all this mean?

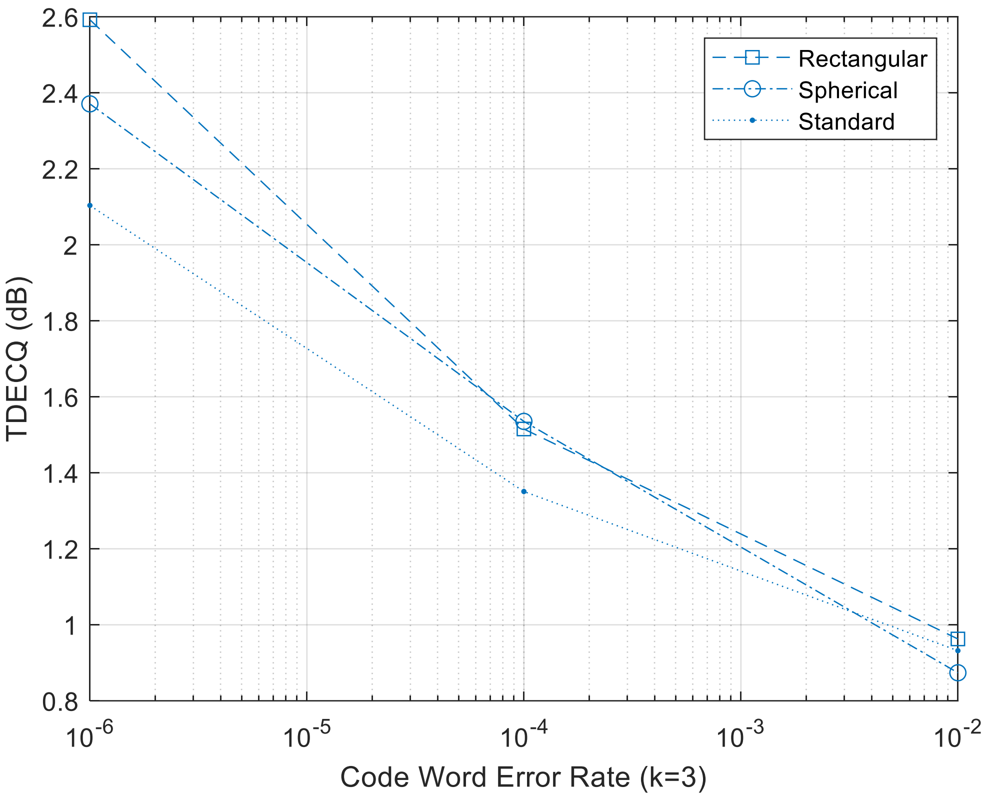
- The current definition of TDECQ calculates the power penalty at a target SER assuming errors are not correlated and randomly distributed.
- The current definition of TDECQ was not intended to correlate with link performance (BER/BER floor) raising questions around correlation between test methodology and link performance.
- Calculating the codeword error rate TDECQ improves the correlation between test methodology and link performance, yet it appears to be a daunting computation requiring 64,449 convolutions per σ.
- Rectangular TDECQ is a geometric approach that treats codewords as vectors in a d-dimensional space to calculate the probability of k errors in a codeword using a set of hypercubes to simplify the calculation of CER TDECQ, yet it is still too complex.
- Hyper-Spherical TDECQ is a technically sound and economically viable approximation that approximates the set of hypercubes as a single hypersphere and uses the Euclidean distance from the ideal codeword location to estimate the probability of k errors in a codeword.

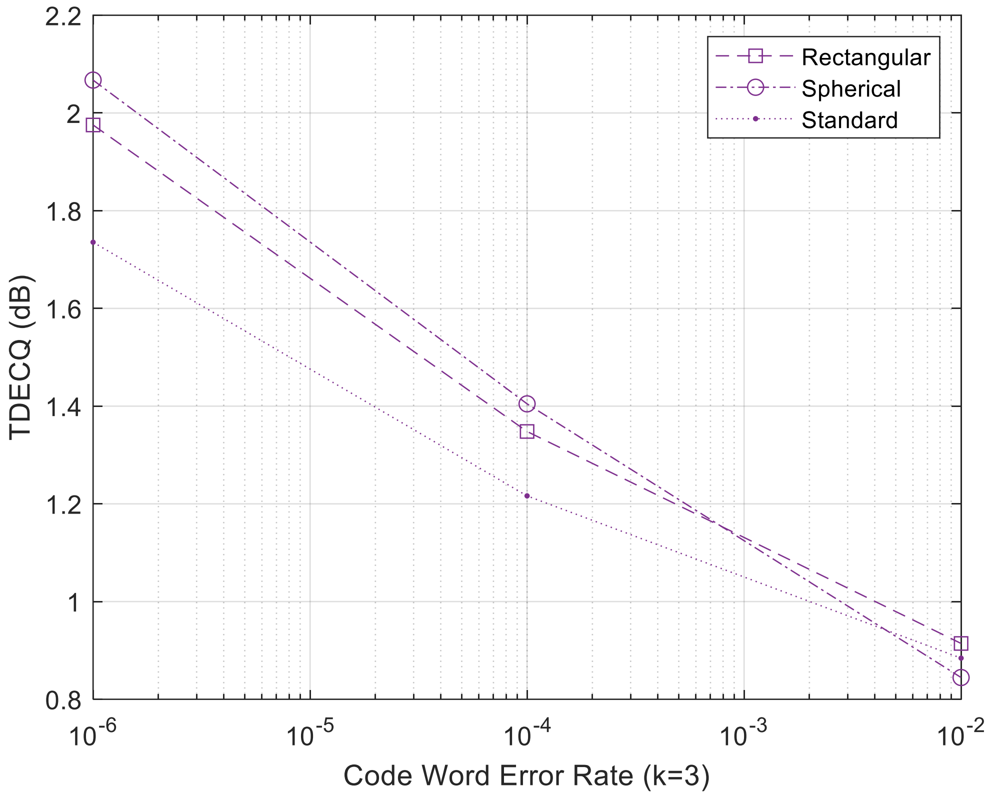
AGENDA

1. Introduction
2. TDECQ in the Current Draft (IEEE P802.3dj D2.0)
3. Codeword Error Rate TDECQ
4. A Geometric Perspective – Rectangular TDECQ
5. Approximating CER TDECQ – Hyper-Spherical TDECQ
6. **Experimental Data**
7. Summary

Comparison of Methods

Two Different Waveforms



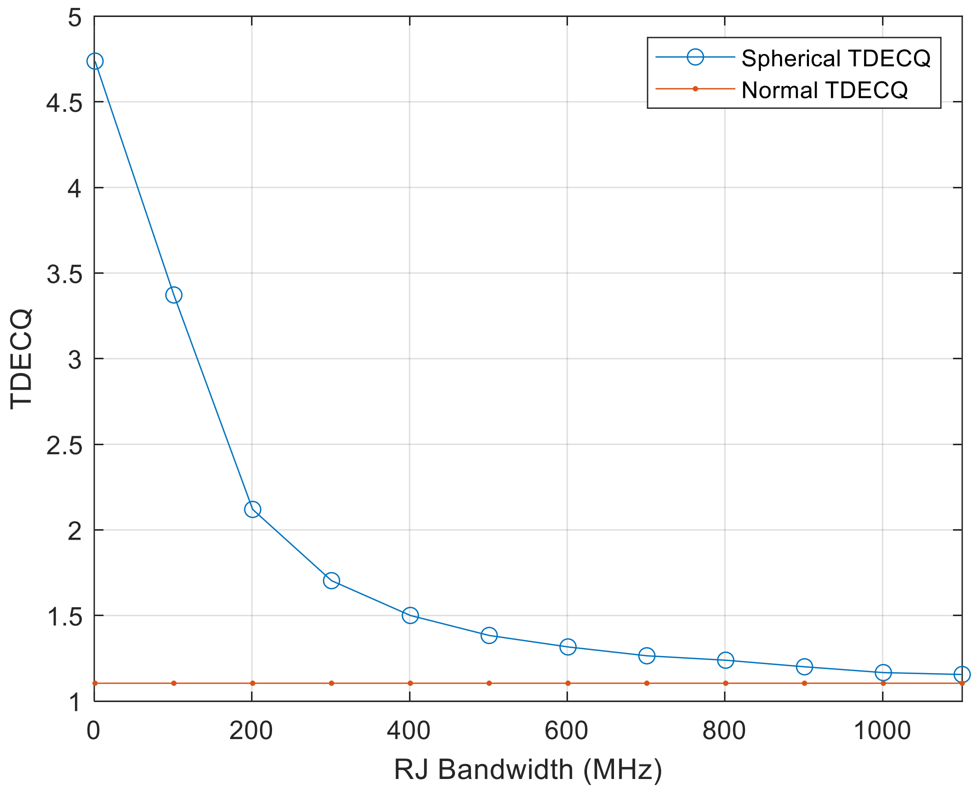


In this example, TDECQ (standard, rectangular and hyper-spherical) was calculated on two sampling scope waveforms. The horizontal scale shown is the codeword error rate of the inner FEC assuming up to 3 symbol errors can be corrected and that the full inner FEC interleaving is applied before encoding for RS.

Bandlimited RJ

Simulated Test Case

- ran_3dj_02a_2407 and oif2024.449.02 demonstrated receivers with performance issue related to bandlimited random jitter
- Using a captured real signal and applying 400fs of bandlimited RJ in MATLAB
- Comparing standard TDECQ and Hyper-Spherical TDECQ, we can see that Hyper- Spherical TDECQ is sensitive to the lower frequency bandlimited RJ
- NOTE: On a Sampling Oscilloscope, the spectrum of the RJ will be aliased, and will appear uncorrelated.



What are the next steps?

- Early experimental data shows that Hyper-Spherical TDECQ is an acceptable approximation at the target codeword error rate.
- Simulated data also shows that real-time oscilloscope Hyper-Spherical TDECQ would capture the band limited jitter issue demonstrated in oif2024.449.02 .
- A FlexDCA beta implementation of Rectangular TDECQ and Hyper-Spherical TDECQ will be made available to individuals who wish to test it. More contributions with experimental data are highly encouraged.
- We are looking to gage the interest of task force members to pursue the direction of investigating a CER TDECQ implementation that better correlates to link performance through a directional straw poll.
- The current IEEE P802.3dj D2.0 points to clause 121.8.5.1 of the IEEE 802.3bs spec. Adopting the new test methodology could either replace the current TDECQ measurement or be defined as an additional metric. Implementation details to follow based on feedback from the working group members.

AGENDA

1. Introduction
2. TDECQ in the Current Draft (IEEE P802.3dj D2.0)
3. Codeword Error Rate TDECQ
4. A Geometric Perspective – Rectangular TDECQ
5. Approximating CER TDECQ – Hyper-Spherical TDECQ
6. Experimental Data
7. **Summary**

Summary

- TDECQ is doing exactly what it was designed to do, and that is to compute a power penalty against a well defined minimum viable reference receiver to ensure interoperability.
- The functional receiver test proposed by Chris Cole could complement TDECQ to verify link performance and guard against error bursts and correlated errors, however, it cannot guarantee interoperability unless a bronze reference receiver is defined and used (something that we have moved away from since we adopted PAM4 signaling).
- The goal of IEEE specs is to ensure interoperability. While a functional receiver test can be used to verify link performance, a penalty-based metric (SER TDECQ or CER TDECQ) is required for interoperability.
- This presentation proposes a Codeword Error Rate TDECQ calculation (Rectangular TDECQ) that can be implemented on real-time and sampling oscilloscopes, as well as a technically sound and economically viable approximation (Hyper-Spherical TDECQ) that can address some of the challenges associated with optimizing transceiver performance.
- Hyper-Spherical TDECQ implemented on a sampling scope will capture correlated errors synchronous to the pattern (ISI) which are not captured by the current definition of TDECQ (Standard TDECQ). Hyper-Spherical TDECQ implemented on a real-time oscilloscope will additionally capture correlated errors that are asynchronous to the pattern (eg, jitter).
- Early experimental data shows that Hyper-Spherical TDECQ is an acceptable approximation at the target codeword error rate. A beta SW implementation of CER TDECQ (Rectangular and Hyper-Spherical TDECQ) will be made available to individuals who wish to test it. Contributions with experimental data are highly encouraged.

Thank you