Digital Communication using GNU Radio

Prof. Kumar Appaiah

Department of Electrical Engineering

Indian Institute of Technology Bombay

Week-06

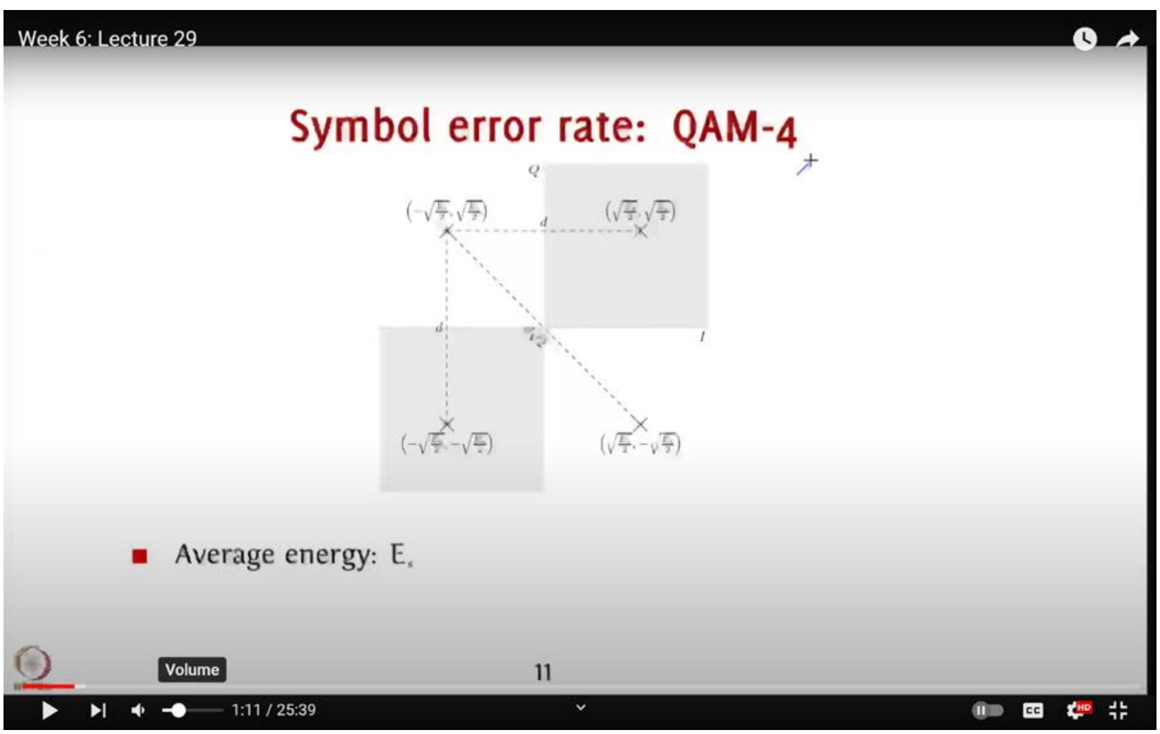
Lecture-29

Signal-to-Noise Ratio and Symbol Error Probability - Part 2

Welcome back to our lecture on Digital Communication using GNU Radio. I'm Kumar Appiah from the Department of Electrical Engineering at IIT Bombay. In this session, we will continue our exploration of bit errors and symbol errors across various constellations.

In our previous lecture, we discussed how to compute the energy of a constellation and how to scale the constellation points to achieve the desired energy. We specifically calculated the symbol error rates for both BPSK and PAM-4, and we gained a foundational understanding of how to approach similar calculations for other linear constellations.

(Refer Slide Time: 01:11)

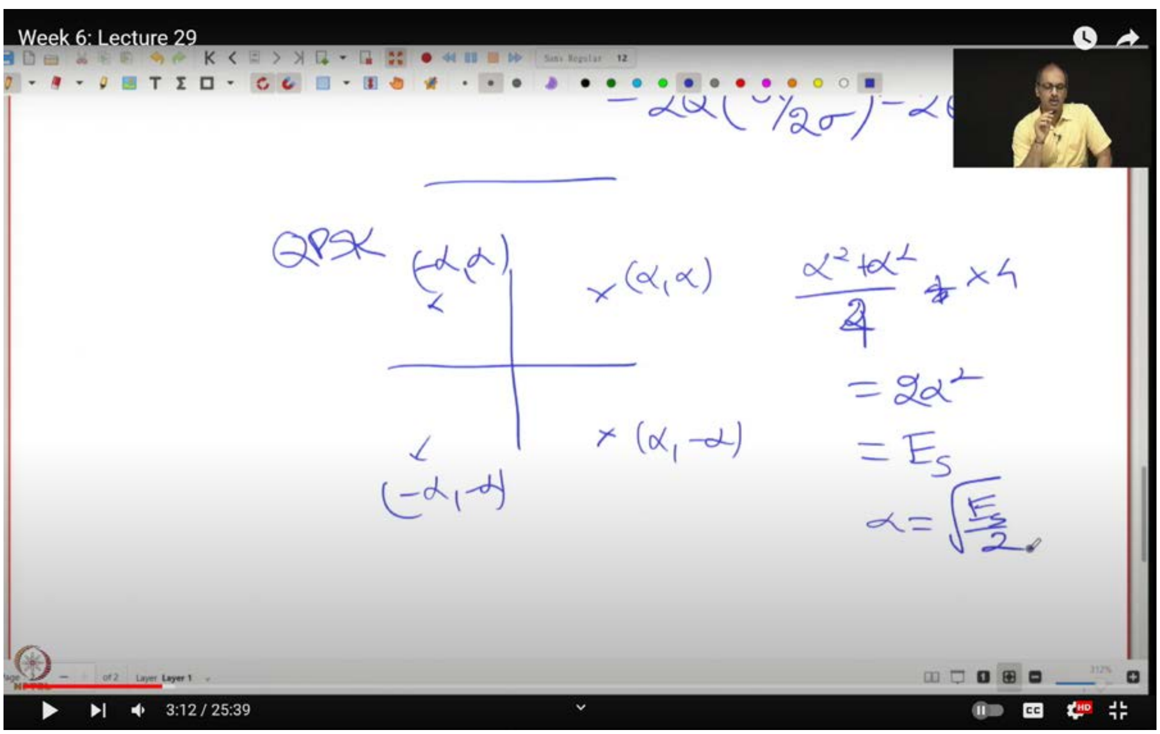


Today, we will start by examining the symbol error rate for QAM-4, which introduces us to constellations with complex symbols, unlike PAM-4 and BPSK, where the symbols are real.

Let’s delve into computing the average energy of a constellation, beginning with QPSK, also known as QAM-4. As discussed in the previous lecture, it is optimal to center the constellation in such a way that even if it is symmetric, positioning the center away from the origin can reduce the overall energy expenditure. Proper centering ensures that, for the same energy, you achieve a better bit error rate or symbol error rate.

So, let's continue by exploring QAM-4 in more detail, applying these principles to understand its performance and energy characteristics.

(Refer Slide Time: 03:12)



We’re familiar with the QPSK constellation, so let's take a closer look. Consider the points of the constellation: $\alpha + j\alpha$, $\alpha - j\alpha$, $-\alpha + j\alpha$, and $-\alpha - j\alpha$. All four points are equiprobable. The energy associated with sending a particular point, such as $\alpha + j\alpha$, is calculated as $\alpha^2 + \alpha^2$, which simplifies to $2\alpha^2$ because we square the magnitude of $\alpha + j\alpha$ and divide by 4 due to the probability of sending that point being 1/4. Therefore, multiplying by 4 for all points gives us $2\alpha^2$.

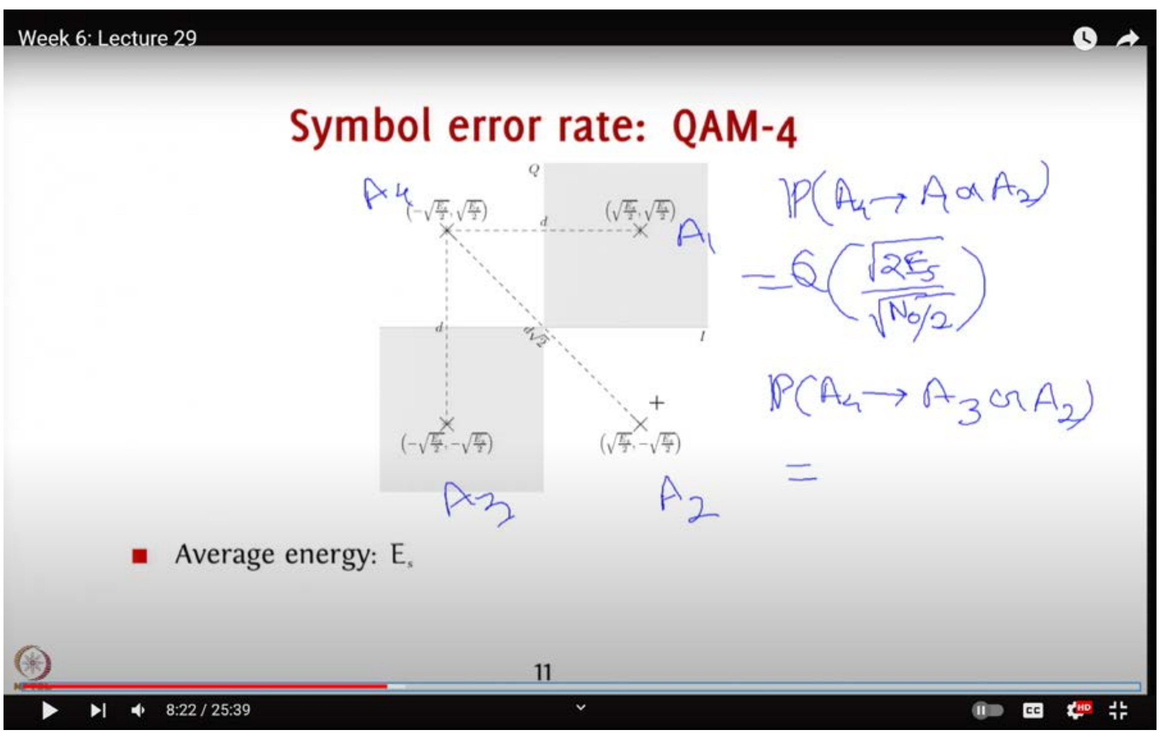
To set this equal to the desired signal energy E_s, we solve for α:

$$2\alpha^2 = E_s$$

$$\alpha = \sqrt{\frac{E_s}{2}}$$

Thus, the constellation points become $\pm\sqrt{\frac{E_s}{2}} \pm j\sqrt{\frac{E_s}{2}}$, which are exactly what we've illustrated. I've deliberately chosen these values to reflect an energy E_s, rather than unit energy.

(Refer Slide Time: 08:22)

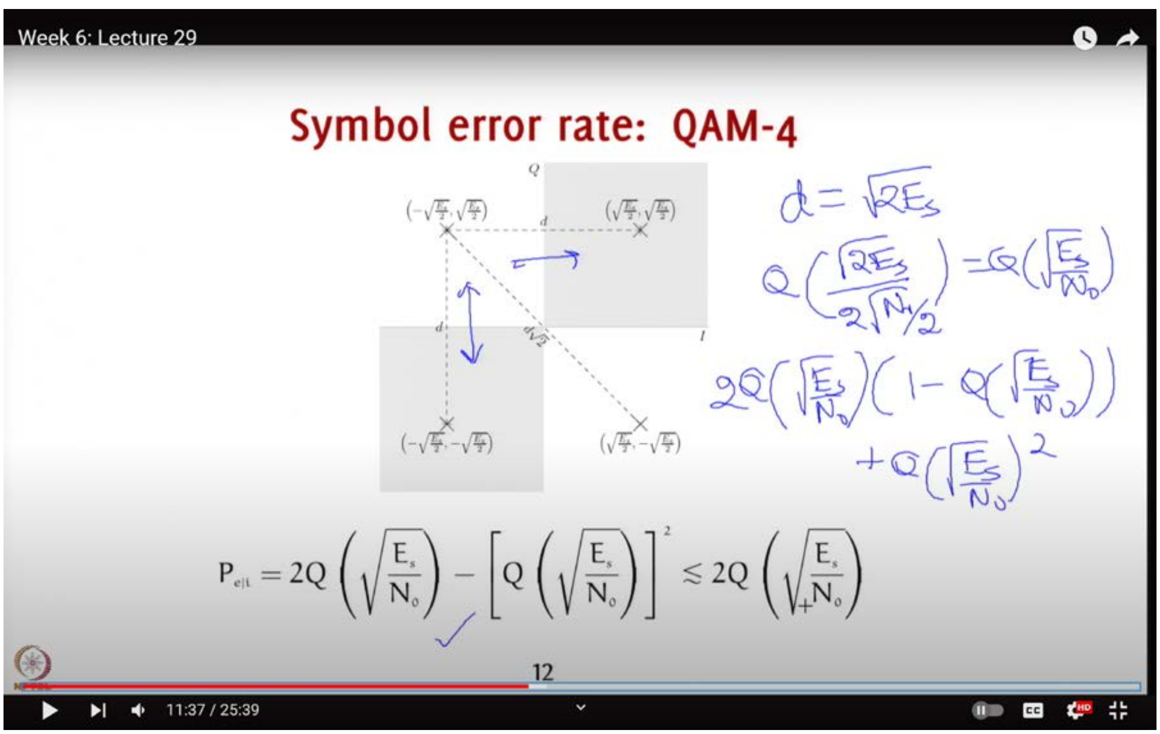


I have highlighted certain areas in grey and white for a reason. Suppose we transmit a particular constellation point. The noise affecting this point is complex additive white Gaussian noise (AWGN). You can imagine a two-dimensional Gaussian noise distribution centered on the constellation point, which means the noise can affect the point in any direction with equal probability.

This setup leads to several possible error events. For example, if the point is detected incorrectly as another, such as a_4 being mistaken for a_1, a_3, or a_2, these are considered error events. Specifically, in QAM-4, there are three distinct possible symbol error events. While it is possible to compute these error probabilities individually, a more efficient approach is available.

Let's simplify this approach by considering the QPSK constellation in terms of its quadrants. If we look at a constellation point, say a_4, which is located in the fourth quadrant, and analyze the possibility of it being detected as another point in a different quadrant, such as a_1, a_2, or a_3, we can compute the probability of error more easily.

(Refer Slide Time: 11:37)



To determine whether a_4 is misdetected as a_1 or a_2, we need to understand how far the point has shifted from its original position. Essentially, we are interested in whether the point has moved sufficiently in the real part to fall into the region corresponding to a_1 or a_2. This means we need to assess whether the real part of the received value has shifted by more than $\frac{d}{2}$.

Here, $\frac{d}{2}$ represents the distance from the vertical axis (the imaginary axis) to the decision boundary. Therefore, the probability of a_4 being misdetected as a_1 or a_2 can be computed using the Q-function. Specifically, it is given by:

$$Q\left(\frac{d}{2\sigma}\right)$$

In this case, d is $\sqrt{E_s}/2 \times 2 = \sqrt{2E_s}$, and σ represents the standard deviation of the noise, which is $\sqrt{N_0/2}$ because we are considering only the real part of the signal. Thus, the expression for the probability of error is:

$$Q\left(\frac{\sqrt{2E_s}}{2 \cdot \sqrt{N_0/2}}\right)$$

By focusing on the real part of the signal and using this simplified approach, we can efficiently compute the probability of symbol errors in QPSK.

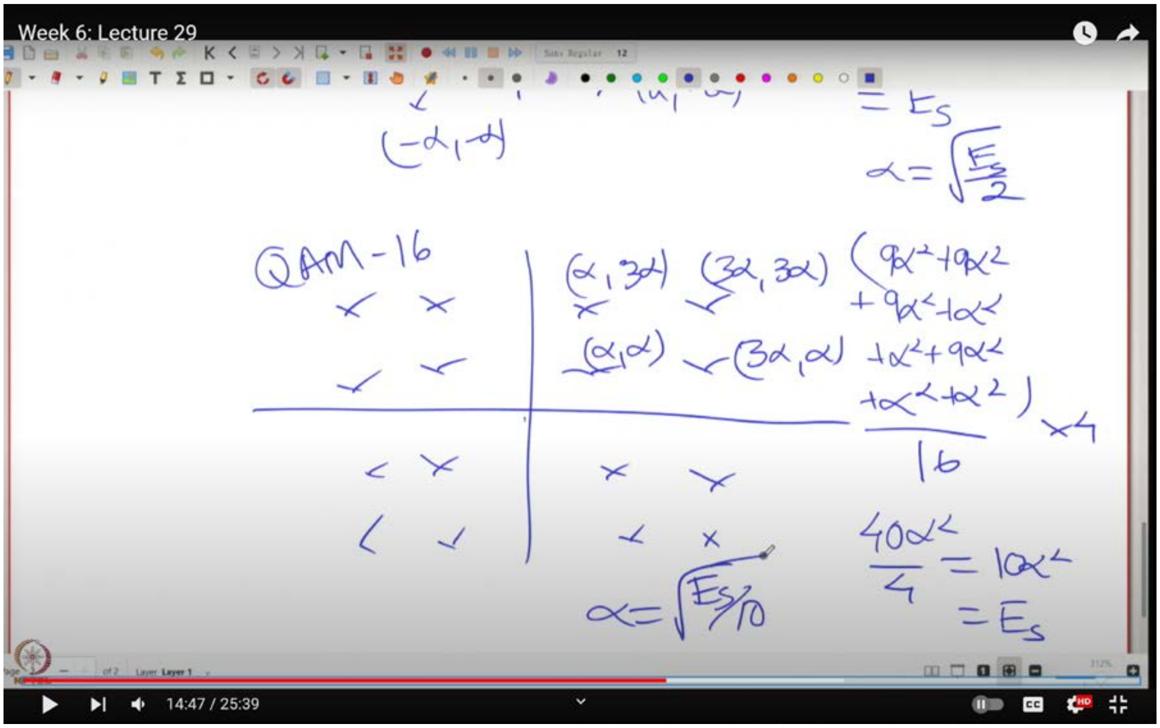
Similarly, if you compute the probability that a_4 is misdetected as a_3 or a_2, you will find that the process is quite straightforward and yields similar results. To determine the overall probability that a_4 is misdetected, you can apply basic probability principles. Specifically, if we denote the events of misdetection as E_1 and E_2, the probability of a_4 being in error can be calculated using the union of these events.

You can use the formula:

$$P(E_1 \cup E_2) = P(E_1) + P(E_2) - P(E_1 \cap E_2)$$

This formula accounts for the overlap between the events E_1 and E_2, so you subtract the probability of the intersection to avoid double-counting. Alternatively, you can also use the Q-function directly to compute this probability.

(Refer Slide Time: 14:47)



For a simpler approach, consider the fact that if the real part of the constellation point needs to shift by d/2 or more to fall into a different region, you are essentially looking at how far the point moves in relation to the decision boundaries. The imaginary part should not shift by d/2 or more. Therefore, you need to compute the probability based on these shifts in both dimensions.

In a more direct calculation, the probability of error can be expressed as:

$$2 \times Q\left(\frac{\sqrt{E_s}}{N_0}\right) - Q^2\left(\frac{\sqrt{E_s}}{N_0}\right)$$

The subtraction accounts for the common area where both types of misdetections overlap. There might be slight scaling differences in the calculations, but this formula provides a reliable estimate of the symbol error rate.

To guide you through the calculation for symbol error probability, consider the following. Let's denote d as the distance, where in this case $d = \sqrt{2E_s}$. For the correct calculation of the error probability, we use:

$$Q\left(\frac{d}{2\sigma}\right)$$

Here, d is $\sqrt{2E_s}$, so for this particular event, the probability is:

$$Q\left(\frac{\sqrt{2E_s}}{2\sqrt{N_0/2}}\right) = Q\left(\frac{\sqrt{E_s}}{\sqrt{N_0}}\right)$$

This gives the probability that the real part of the signal moves by d/2. The imaginary part should not move by the same amount. To account for this, we need to calculate:

$$Q\left(\frac{\sqrt{E_s}}{N_0}\right) \times \left(1 - Q\left(\frac{\sqrt{E_s}}{N_0}\right)\right)$$

For the scenario where both the real and imaginary parts experience similar shifts, you will add:

$$Q\left(\frac{\sqrt{E_s}}{N_0}\right)$$

Expanding this, the total probability expression becomes:

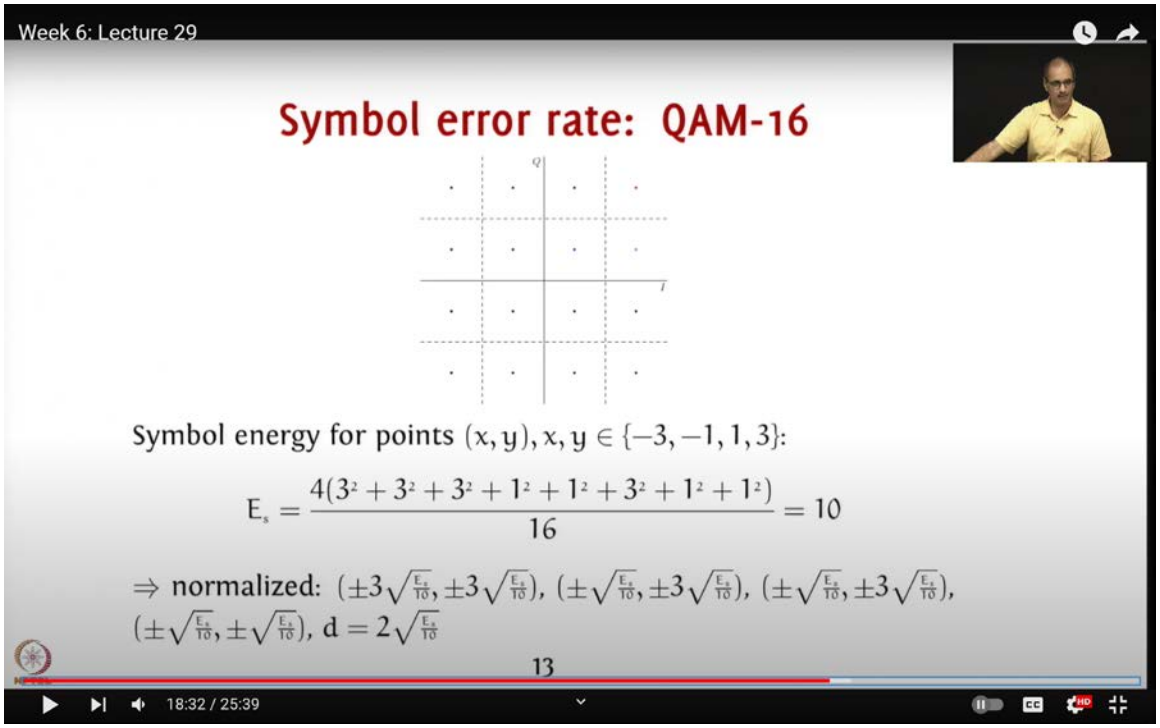
$$2Q\left(\frac{\sqrt{E_s}}{N_0}\right) - 2\left(Q\left(\frac{\sqrt{E_s}}{N_0}\right)\right)^2$$

This formula represents the probability of symbol error for QAM-4. For high SNR scenarios, it can be approximated as:

$$2Q\left(\frac{\sqrt{E_s}}{N_0}\right)$$

Now, let’s consider QAM-16, which is a more complex constellation with 16 symbols. Unlike QAM-4, where increasing the number of symbols leads to higher error rates, QAM-16 is widely used, especially when there is a sufficient SNR. It allows for denser packing of symbols compared to PAM-4.

(Refer Slide Time: 18:32)



Interestingly, QAM-16 can be related to PAM-4. For example, QAM-4 can be visualized as a combination of two BPSK systems: one determining the position along the real axis and the other along the imaginary axis. Similarly, QAM-16 can be seen as a combination of four PAM-4 systems. Here, one PAM-4 constellation determines the vertical position, while another determines the horizontal position.

Let's quickly compute the symbol energy for QAM-16. In a QAM-16 constellation, there are 16 points arranged as follows: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, and so on. For simplicity, and because we're centering our analysis around the origin, I'll sketch these points roughly. Assume we have the points (3α, 3α), (3α, α), (α, 3α), and (α, α).

Our goal is to determine the value of α such that the average energy of all constellation points equals E_s. To do this, we calculate the energy of each point, then average it. Here's how we do it:

1. Calculate the Energy for Each Point:

- For $(3\alpha, 3\alpha)$: The energy is $(3\alpha)^2 + (3\alpha)^2 = 18\alpha^2$.
- For $(3\alpha, \alpha)$: The energy is $(3\alpha)^2 + \alpha^2 = 10\alpha^2$.
- For $(\alpha, 3\alpha)$: The energy is $\alpha^2 + (3\alpha)^2 = 10\alpha^2$.
- For (α, α): The energy is $\alpha^2 + \alpha^2 = 2\alpha^2$.

Since each point contributes to the average energy, and considering symmetry, we multiply each energy term by 4 (the number of unique points in the quadrant).

2. Compute the Average Energy:

- Total energy = $4 \times \left(\frac{18\alpha^2+10\alpha^2+10\alpha^2+2\alpha^2}{16}\right)$
- Simplify: Total energy = $\frac{40\alpha^2}{4} = 10\alpha^2$

To match this to E_s, we set $10\alpha^2 = E_s$, so:

$$\alpha = \sqrt{\frac{E_s}{10}}$$

Thus, the constellation points should be scaled by $\frac{\sqrt{E_s}}{\sqrt{10}}$. Hence, the coordinates of the points in the normalized constellation are $\pm 3\frac{\sqrt{E_s}}{10}$ and $\pm 1\frac{\sqrt{E_s}}{10}$.

While the constellation might appear complex, it offers a significant advantage in terms of data transmission. Each symbol in QAM-16 can encode 4 bits, compared to PAM-4 where each symbol encodes only 2 bits. This density of information per symbol enhances the efficiency of data transmission.

I’ve highlighted three specific points in different colors: red, light blue, and violet. These are the critical points for computing the symbol error rate. This is because, due to symmetry, the other points in the constellation will have similar error characteristics. For instance, points that are symmetrically placed relative to the origin will have analogous error probabilities.

When analyzing the symbol error rate for QAM-16, it’s helpful to recognize that all eight points in a given quadrant are similar, meaning they will experience similar symbol errors. For simplicity, you only need to compute the error rate for three distinct points, though I will explain how to approach this.

Consider a distance d between points in the constellation. In this context, d is defined as $2\sqrt{\frac{E_s}{10}}$, and all points are spaced by this distance. Now, let's examine the error events for these points:

1. For the first point:

Errors occur if the received signal moves in any of the directions that are equidistant from the point. This scenario resembles QPSK closely. You can compute the error events for this point using the same approach as you would for QPSK, adjusting for the different distance d.

2. For the second point:

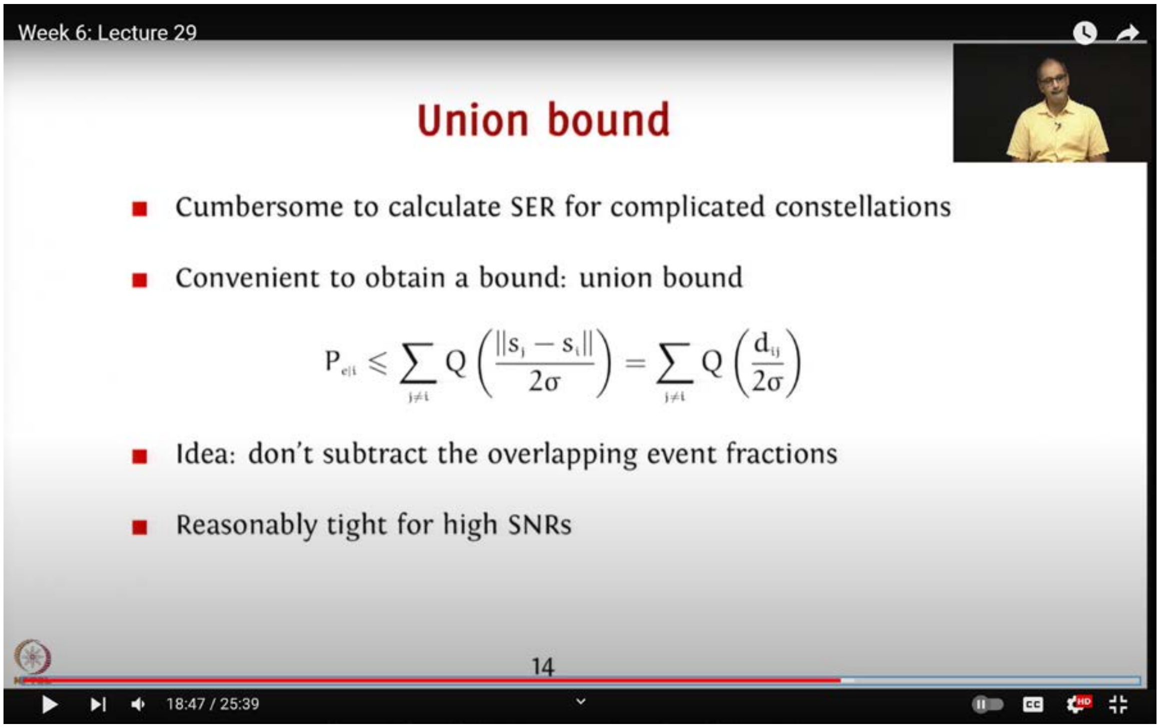
This point behaves similarly to PAM-4. Errors occur if the received signal moves to any of the four adjacent regions. The computation is straightforward and follows the same principles as those used for PAM-4.

3. For the third point:

This point has four decision boundaries, meaning errors can occur if the signal moves across any of these boundaries. The approach involves calculating errors along both the real and imaginary components, similar to how you handle the middle point in PAM-4, and then combining these results.

Therefore, while calculating the symbol error rates for QAM-16 can be complex due to the multiple decision boundaries and overlapping regions, the fundamental principles are the same as for simpler constellations.

(Refer Slide Time: 18:47)



For high SNR scenarios, detailed calculations might be cumbersome. In such cases, using an approximation like the union bound can be very useful. The union bound provides an upper limit on the probability of error by summing the probabilities of error for each constellation point. Specifically, it is given by:

$$P(\text{error}) \leq \sum Q\left(\frac{|S_j - S_i|}{2\sigma}\right)$$

Here, Q is the Q-function, S_i represents the transmitted symbol, and S_j represents a potentially erroneous symbol. This method simplifies the calculation by bounding the probability of error rather than computing it precisely, making it a practical tool for handling more complex constellations like QAM-256, QAM-1024, or QAM-4096.

In simpler terms, what they are suggesting is to use the union bound without worrying about overlapping errors. In the case of QPSK, for example, we previously considered $P(E_1) + P(E_2) - P(E_1 \cap E_2)$ to account for overlapping error events. However, with the union bound, we simplify this by ignoring the intersection term.

So, instead of subtracting out the overlap, the union bound approach involves summing the probabilities of individual error events. Specifically, you compute $Q\left(\frac{D_{ij}}{2\sigma}\right)$ for each pair of constellation points. This means you calculate the probability of error assuming each pair of points could be in error independently, without accounting for the intersection of those error events.

To illustrate, imagine your constellation points can potentially be in error if the received signal moves in various directions. The union bound method involves:

1. Calculating $Q\left(\frac{D_{ij}}{2\sigma}\right)$ for each possible pair of constellation points.

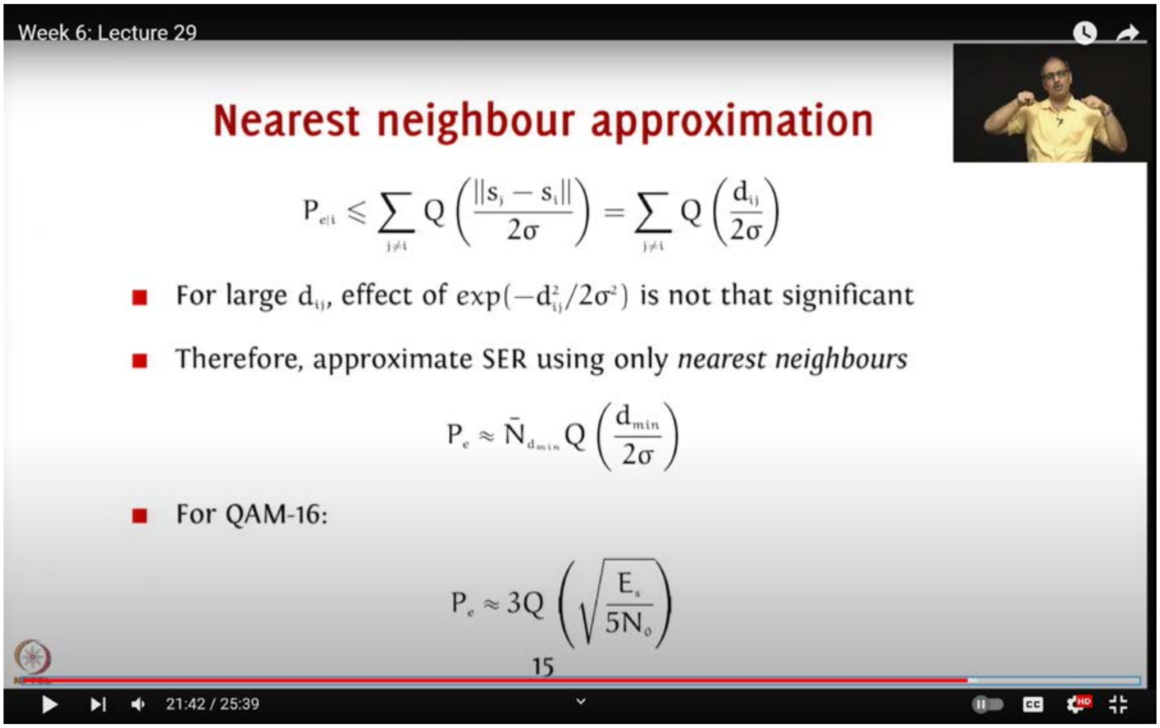
2. Summing these probabilities across all pairs, assuming each pair of points is considered in isolation, without worrying about overlapping regions.

In essence, you are adding up the probabilities of errors for each possible pair of points and ignoring the complexity of overlapping error regions. This approach simplifies the calculation by avoiding the need to account for overlapping error events, making it more manageable while still providing a useful upper bound on the error probability.

The rationale behind this approach is grounded in the properties of the Q function. Recall that the Q function has an exponential decay, meaning $Q\left(\frac{D_{ij}}{2\sigma}\right)$ decreases exponentially as D_{ij} becomes large. When D is very large, the term $e^{-\frac{D^2}{2\sigma^2}}$ becomes exceedingly small. Consequently, in many cases, only a few terms in the summation are significant, while the rest contribute negligibly to the overall error probability.

For example, in the case of QAM-16, some errors will have a considerable impact, while others will be so distant that they don't significantly affect the probability of error. This is particularly true when the distance between points is several times greater than the distance between nearest neighbors. Hence, the union bound simplifies the calculation by allowing us to focus on the most impactful terms.

(Refer Slide Time: 21:42)



The union bound provides a reasonably tight approximation, especially for high signal-to-noise ratios (SNRs). In high SNR scenarios, even small changes in distance become insignificant because the Q function's values for large arguments are very small. Thus, approximating the symbol error rate using the nearest neighbors becomes effective.

To apply the nearest neighbor approximation, you:

1. Identify the nearest neighbors for each constellation point.

2. Calculate the average number of nearest neighbors.

3. Multiply this average by $Q\left(\frac{D_{\min}}{2\sigma}\right)$, where $D_{\min}$ is the minimum distance between nearest neighbors.

For instance, in the QAM-16 constellation, some points might have two nearest neighbors, others three, and some four. By averaging these values and using $D_{\min}$, you get an approximation for the symbol error rate. In the example provided, this yields an error probability of approximately $3 \cdot Q\left(\frac{\sqrt{E_S}}{\sqrt{5N_0}}\right)$.

This approximation works well for high SNR scenarios but remember that it is not an exact bound but rather an estimate. At lower SNR levels, this approximation may not be as accurate.

To summarize, one practical approach for calculating the symbol error rate in various scenarios is to simplify the process by using a simulation-based method, specifically the Monte Carlo method. This involves generating numerous constellation points, applying Gaussian noise to each, and then determining the fraction of symbols that are correctly and incorrectly detected. This method will be employed in our analysis of Bluetooth radio systems, as it effectively estimates the symbol error rate without the need for complex calculations.

Monte Carlo simulations are particularly useful when calculating accurate symbol error rates is complex and cumbersome. They provide a straightforward and practical solution for assessing performance.

Let’s review what we’ve covered so far. We’ve calculated the symbol error rate for several key constellations, considering how the arrangement and spacing of constellation points, along with the noise, influence the error rate. Whether you use a real or complex constellation, the configuration and minimum distance between points are crucial factors in determining the symbol error rate.

For more intricate constellations, you can apply the union bound and nearest neighbor approximation to estimate the symbol error rate. In the next lecture, we will build on this knowledge and explore bit error rates. This is important because not all symbol errors translate directly into bit errors, and we’ll examine these differences in detail. Thank you.