Digital Communication using GNU Radio

Prof. Kumar Appaiah

Department of Electrical Engineering

Indian Institute of Technology Bombay

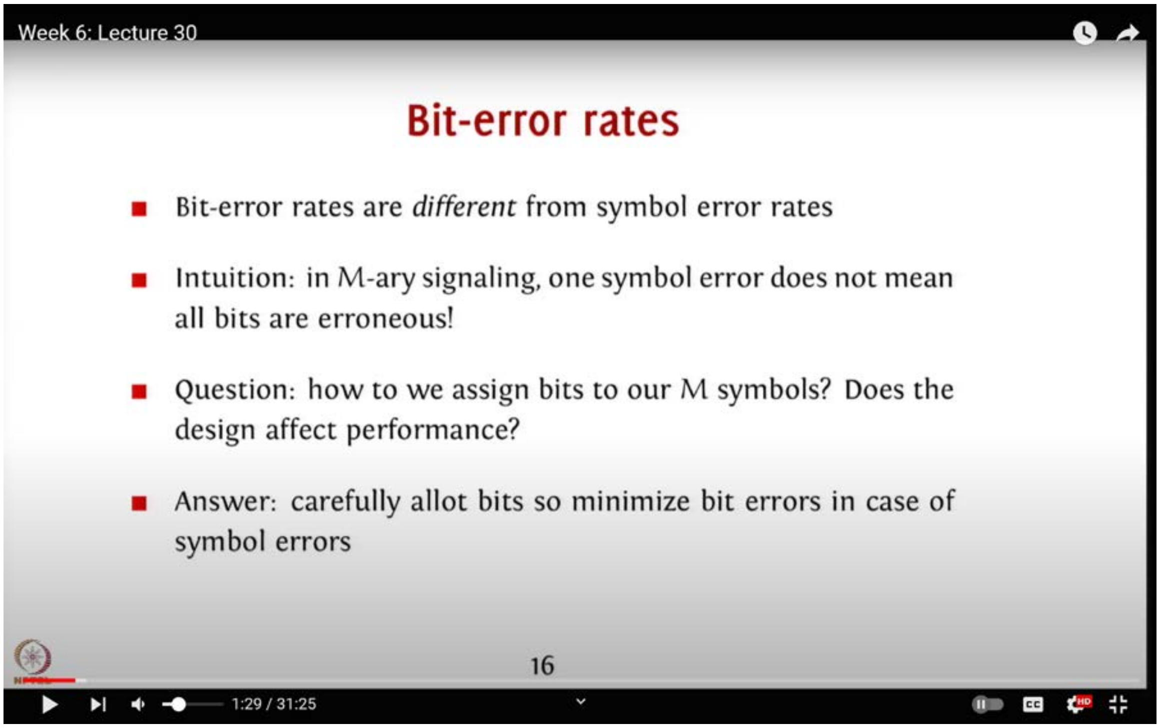
Week-06

Lecture-30

Symbol Error Rate and Bit Error Rate

Welcome to this lecture on Digital Communication using GNU Radio. My name is Kumar Appiah, and today, we will continue our exploration of demodulation, focusing on the concepts of symbol error rates (SER) and bit error rates (BER).

(Refer Slide Time: 01:29)

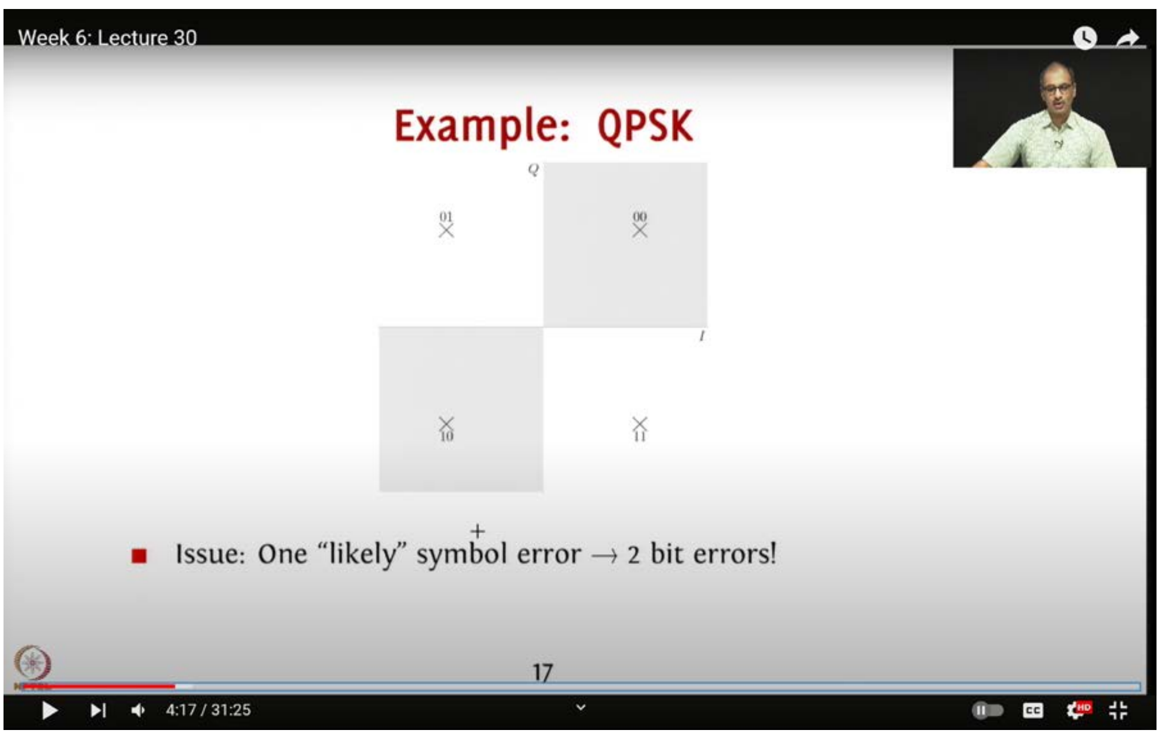


In our previous sessions, we delved deeply into the topic of symbol error rates, analyzing how to compute SER for various types of constellations. Specifically, we examined SER for BPSK, PAM-4, QPSK, and QAM-16, and discussed the similarities and differences between these modulation schemes. We also touched on the fact that for more complex constellations, methods like the union bound might be necessary to derive a useful approximation of the SER.

However, there's an important aspect that remains to be addressed, while we've discussed symbol error rates extensively, this course ultimately centers on the transmission of bits. This brings us to the key question: What is the difference between symbol error rates and bit error rates, and how do we compute bit error rates for a given constellation? This is the topic we will focus on today.

The first and crucial point to understand is that bit error rates are not the same as symbol error rates. The intuition behind this lies in the nature of M-ary signaling. In M-ary signaling, it's not always the case that a single symbol error results in all bits being incorrect. Let's consider an example to illustrate this:

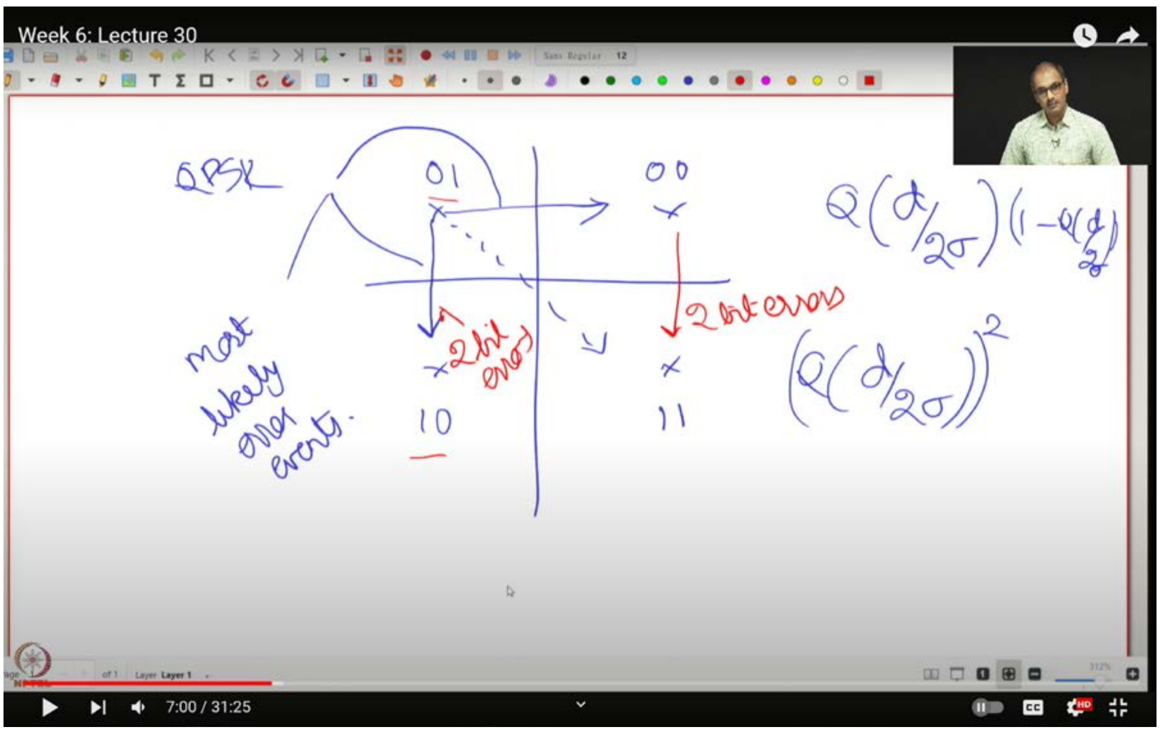
(Refer Slide Time: 04:17)



Take QPSK (Quadrature Phase Shift Keying), for instance. In QPSK, there are four possible symbols, each corresponding to a unique pair of bits. If you transmit the symbol corresponding to the bit pair "00", noise in the channel might cause this symbol to be received incorrectly as the symbol corresponding to "01". In this scenario, the original "00" is mistakenly detected as "01", which constitutes a symbol error, as we've discussed previously. However, of the two bits, only one is incorrect, the other remains correct. Thus, this symbol error leads to only a single bit error.

The key takeaway here is that a single symbol error does not necessarily imply that all bits within that symbol are erroneous. This distinction between SER and BER is fundamental to understanding how bit error rates are calculated and how they impact the overall performance of a communication system.

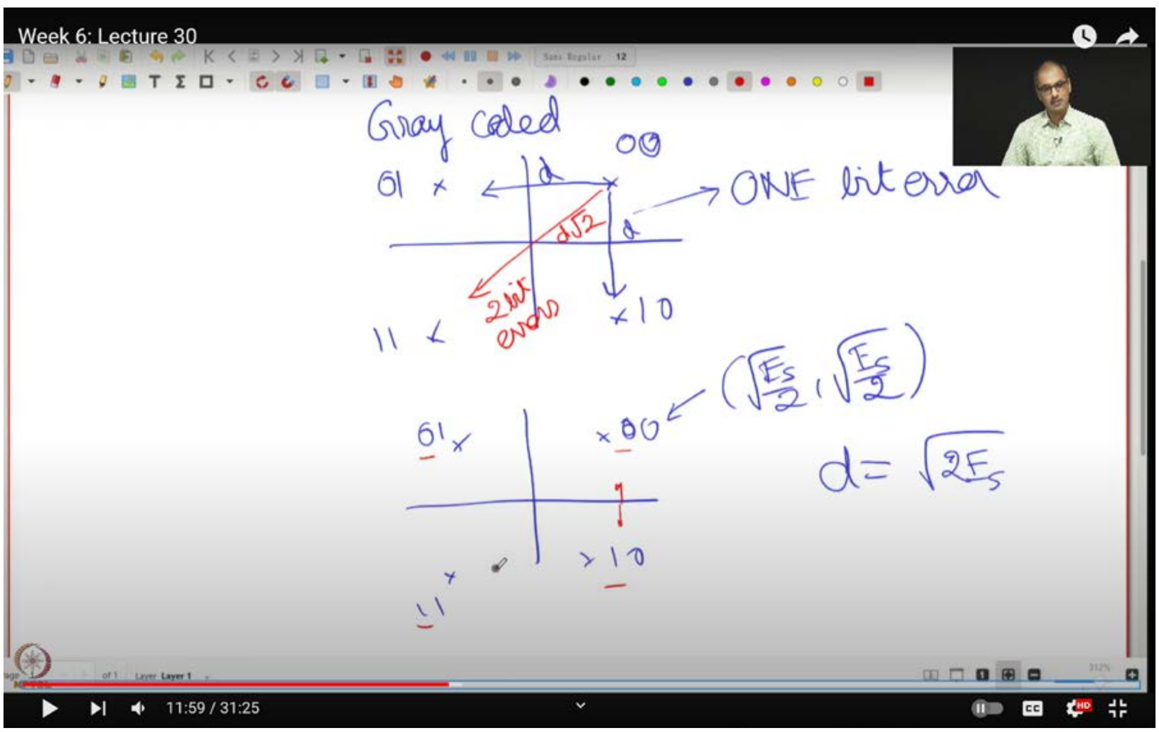
(Refer Slide Time: 07:00)



To truly understand the relationship between symbol error rates (SER) and bit error rates (BER), we need to extend our computation of SER to take into account the impact on BER. As mentioned earlier, BER, or bit error ratio, differs from SER due to the way bits are mapped to symbols. This brings us to an intriguing question: with M symbols available, how should we assign bits to these symbols? For instance, in the case of QPSK, which symbol should be mapped to "00", which one to "11", "01", or "10"? Does the specific arrangement of these bit-symbol mappings influence the overall performance? The short answer is: Yes, it does. We'll explore this further.

For now, let's focus on constellations where M is a power of 2, which simplifies the process of bit assignment. Take the example of a constellation with 16 symbols (QAM-16), where each symbol corresponds to 4 bits. The critical question arises: does the way we allocate bits to symbols really matter? The answer is that in many constellations, the bit allocation is crucial. You must carefully map bits to symbols to minimize the number of bit errors that occur when a symbol error happens. The goal is to ensure that most symbol error events result in the fewest possible bit errors.

(Refer Slide Time: 11:59)



Let's take a closer look at QPSK as an example. Imagine we have the following bit-symbol mapping: "00", "01", "10", and "11". Now, if we refer back to our discussion on symbol error rates, consider the symbol "01". The most likely error events for "01" occur with its nearest neighbors. In QPSK, these neighbors are "00" and "10", representing the cases where either the real part or the imaginary part of the symbol shifts by d/2. The probability of both the real and imaginary parts shifting simultaneously is lower, meaning that "01" transitioning to "11" is less likely. However, this isn't desirable because if "01" flips to "10", it results in two bit errors, a highly unfavorable outcome, as a single symbol error should ideally cause as few bit errors as possible.

Similarly, the symbol "00" transitioning to "11" would also cause two bit errors. Therefore, this mapping, where a highly probable symbol error event leads to two bit errors, represents a poor allocation of bits to symbols. The objective is to design the bit-symbol mapping in such a way that even when symbol errors occur, the number of resulting bit errors is minimized, thereby enhancing the overall robustness of the communication system.

The issue with this particular bit allocation, or mapping of bit pairs to the four QPSK symbols, is that it results in highly probable symbol error events corresponding to a significant number of bit errors. This is a situation we absolutely want to avoid. In this scenario, the most likely symbol error leads to two bit errors, which is clearly undesirable. So, how do we rectify this?

The solution lies in adopting a different approach, using what is known as Gray coding. Gray coding, named after the scientist who introduced it, is a method where we ensure that all probable symbol error events result in the minimum number of bit flips. Let’s consider the Gray coding scheme with bit allocation like this: "01", "00", "11", "10".

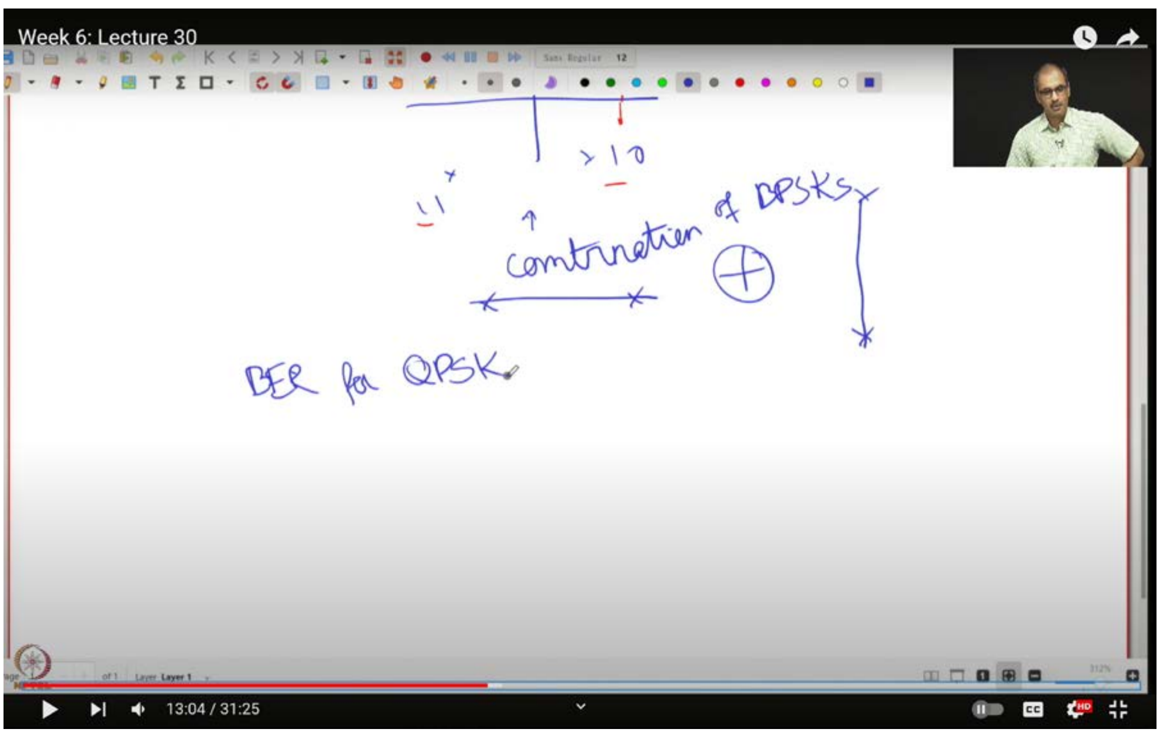
Now, let's evaluate the likely error events in this setup. For instance, when considering the symbol "01", the most likely transitions are to "00" or "11", and these correspond to only one bit error. This is a significant improvement because, in the most probable transitions, we are limiting the bit errors to just one per symbol error. While there is a less likely scenario where a symbol error could lead to two bit errors, this is acceptable given its lower probability.

To put it in perspective, consider the distances: if the distance between the nearest neighbors is denoted as d, then the distance to the next most likely symbol error is $d\sqrt{2}$, which is significantly less probable due to its appearance in the exponential function within the Q function. Gray coding ensures that in the likely symbol error events, only one bit error occurs, and in the least likely events, two bit errors might occur, but this is a rare scenario and far better than the previous bit allocation scheme.

If you wish to calculate the bit error probability for this Gray-coded constellation, you can certainly do so. However, let's go through it together. We'll start fresh, labeling the symbols as "01", "00", "11", and "10". Recall that each of these constellation points was chosen to have coordinates of $\frac{\sqrt{E_s}}{2}, \frac{\sqrt{E_s}}{2}$, which was done to ensure that the average energy of the constellation equals E_s. If you position all the points at a distance of $\sqrt{E_s}$ from the origin, the average energy of this constellation will indeed be E_s, which is our goal.

Now, the distance between these nearest neighboring constellation points is $d = 2\sqrt{E_s}/2 = \sqrt{2E_s}$. With this in mind, we can proceed to find the bit error rate, leveraging the advantages offered by Gray coding in this setup.

(Refer Slide Time: 13:04)

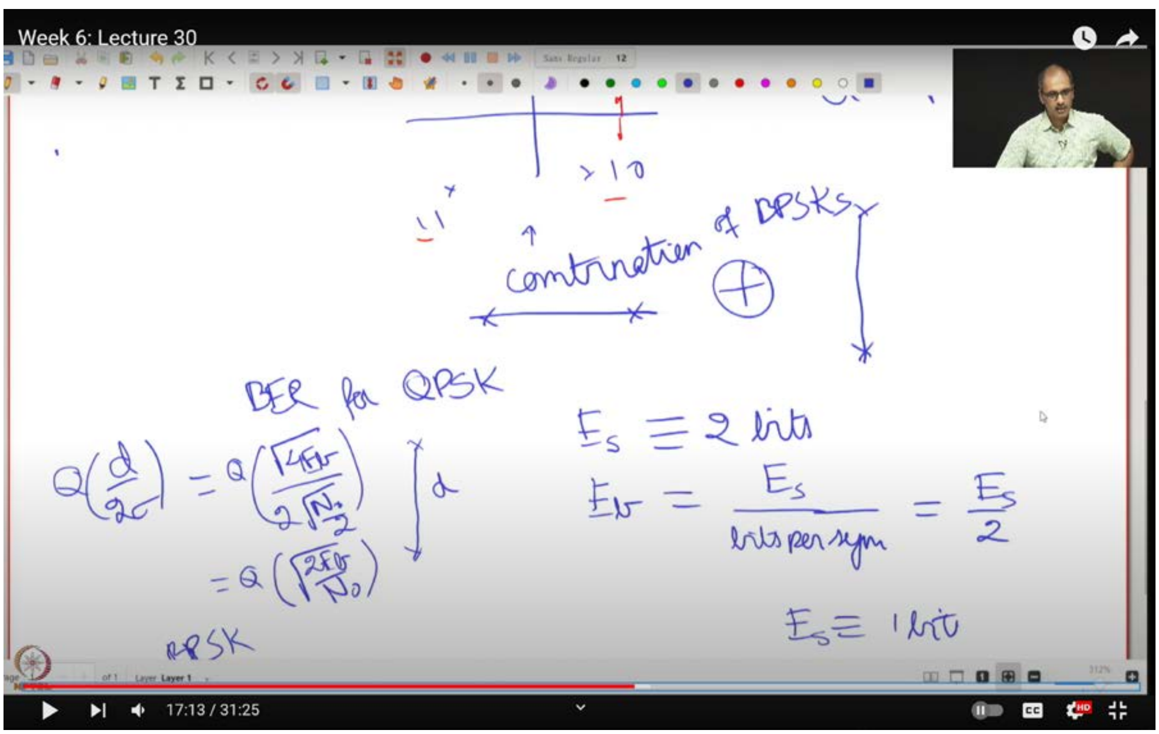


Let's focus on just one aspect, specifically, the Most Significant Bit (MSB). There are several approaches you can take here, but there's a clever trick that can simplify things.

The trick is to observe that the MSB is always zero whenever the received point is above the x-axis, and it becomes one when the received point is below the x-axis. In other words, if your point is located above the x-axis, the MSB is zero, and if it's below, the MSB is one, regardless of its exact position along the x-axis.

This is a particularly neat observation because it allows you to make optimal decisions for one of the bits independently of the other. I briefly mentioned this in the previous lecture, noting that it's something you might expect. The reasoning behind this is that what we essentially have is a BPSK on the x-axis and another BPSK on the y-axis. These are combined to form the QPSK constellation. By viewing QPSK as a combination of two BPSKs, it becomes much easier to grasp and appreciate the similarity between these schemes, and you can straightforwardly compute the bit error rate or symbol error rate.

(Refer Slide Time: 17:13)

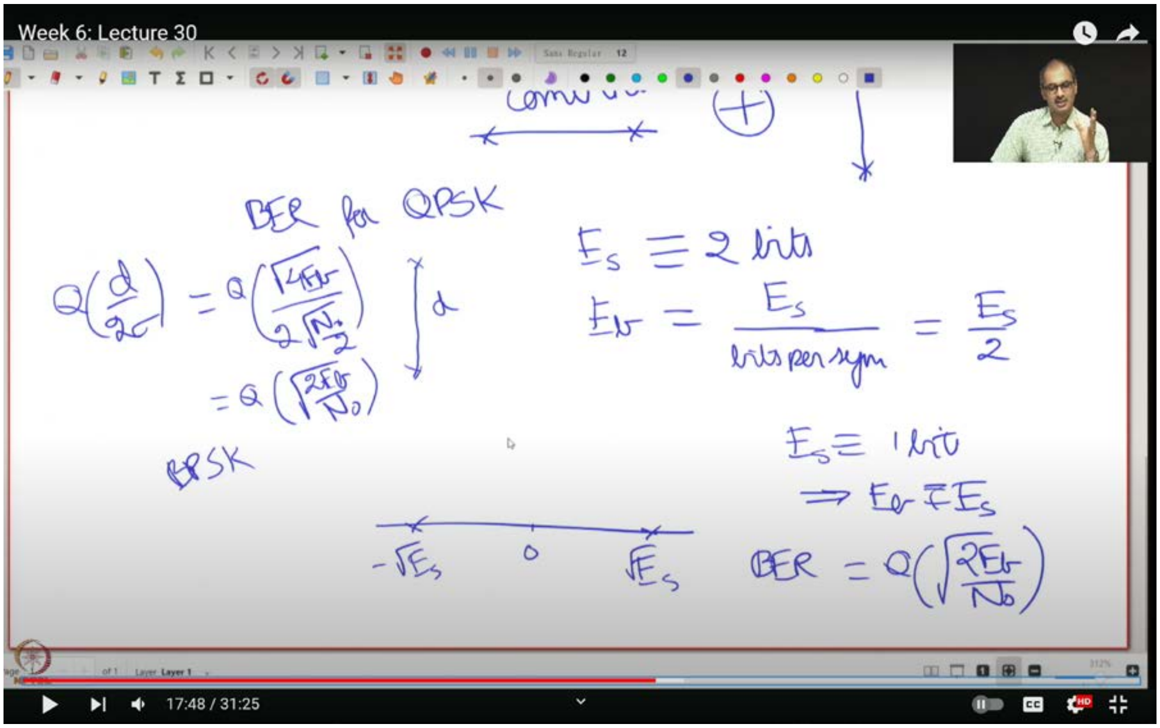


So, let's proceed by considering QPSK as a combination of BPSKs. Now, let's compute the bit error rate for QAM-4, or equivalently, QPSK. But before we dive into that, there's an important concept we need to address: bit energy.

In QPSK, each constellation point has an energy of E_s, which corresponds to two bits. So, what is the energy per bit? Since E_s corresponds to one constellation point and that point represents two bits, we define a new quantity, E_b, as $E_b = \frac{E_s}{\text{bits per symbol}}$, which in this case is $E_b = \frac{E_s}{2}$.

Now that we've established that $E_b = \frac{E_s}{2}$, let's determine the bit error rate for this configuration. But before we do that, let's consider the simpler case of BPSK. For BPSK, the process is straightforward. Imagine the BPSK constellation with points at $-\sqrt{E_s}$ and $\sqrt{E_s}$, where E_s corresponds to one bit. Therefore, in BPSK, $E_b = E_s$, and the bit error rate equals the symbol error rate because one bit corresponds to one symbol.

(Refer Slide Time: 17:48)



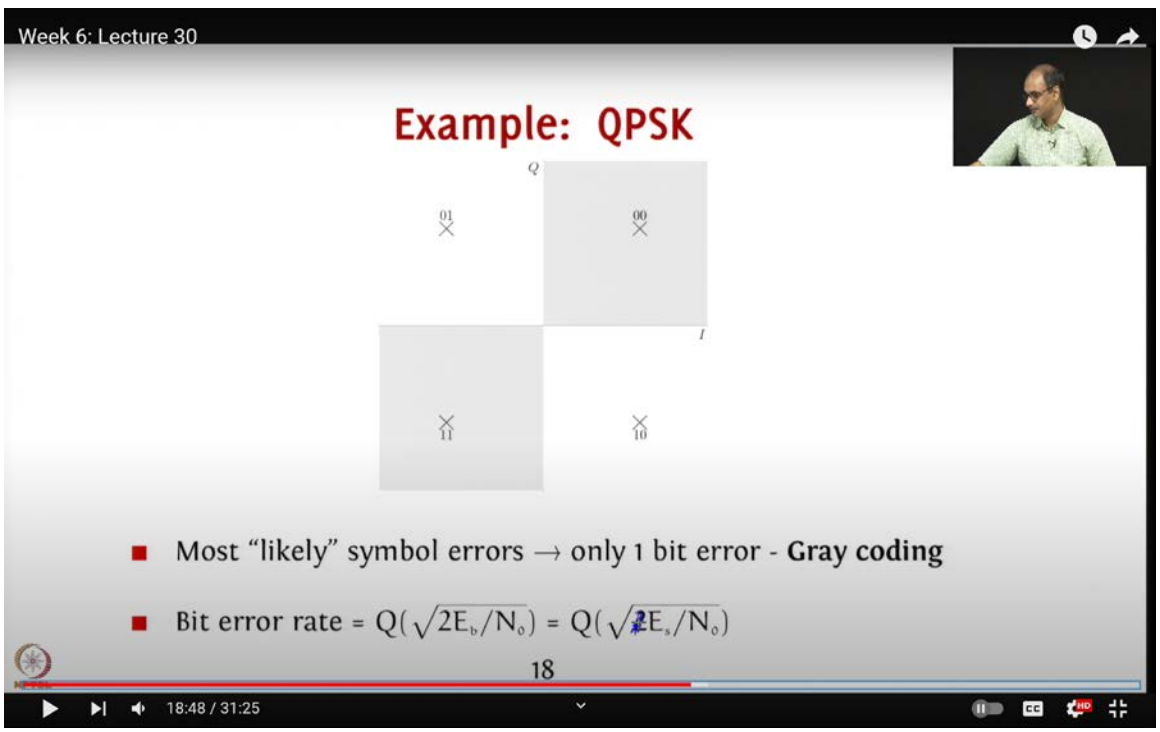
Thus, the bit error rate for BPSK can be expressed as $Q\left(\sqrt{\frac{2E_b}{N_0}}\right)$. This is because, in BPSK, a symbol error directly results in a bit error, making the symbol error rate and bit error rate equivalent.

However, for QPSK, the situation is a bit more complex and requires a different approach. If you observe that only the imaginary component determines the MSB, then you only need to focus on the projection along the y-axis to decide the MSB's value. As you recall, Gaussian noise plays a role in this scenario. Let's say that in QPSK, the distance D between constellation points is important. Our method has been to use $Q\left(\frac{D}{2\sigma}\right)$.

For QPSK, the distance D is $\sqrt{2E_s}$. So, applying this, we write:

$$Q\left(\frac{D}{2\sigma}\right) = Q\left(\frac{\sqrt{2E_s}}{2\sigma}\right)$$

(Refer Slide Time: 18:48)



Now, recall that $E_s = 2E_b$, so $D = \sqrt{2E_s}$ becomes $D = \sqrt{4E_b}$. Plugging this in, we get:

$$Q\left(\frac{\sqrt{4E_b}}{2\sigma}\right) = Q\left(\sqrt{\frac{2E_b}{N_0}}\right)$$

This simplifies to the expression for the bit error rate in QPSK. Therefore, through this approach, we've aligned the QPSK bit error rate with that of BPSK, ensuring consistency in the calculation.

Let's break this down to understand the nuances of BPSK and QPSK, and address the apparent confusion about the bit error rate expressions.

Initially, we observe that the bit error rate for BPSK is given by $Q\left(\sqrt{\frac{2E_b}{N_0}}\right)$, and surprisingly, for QPSK, it also appears as $Q\left(\sqrt{\frac{2E_b}{N_0}}\right)$. This might seem perplexing at first glance, but the key lies in the difference in how power, E_s, is allocated in these two modulation schemes.

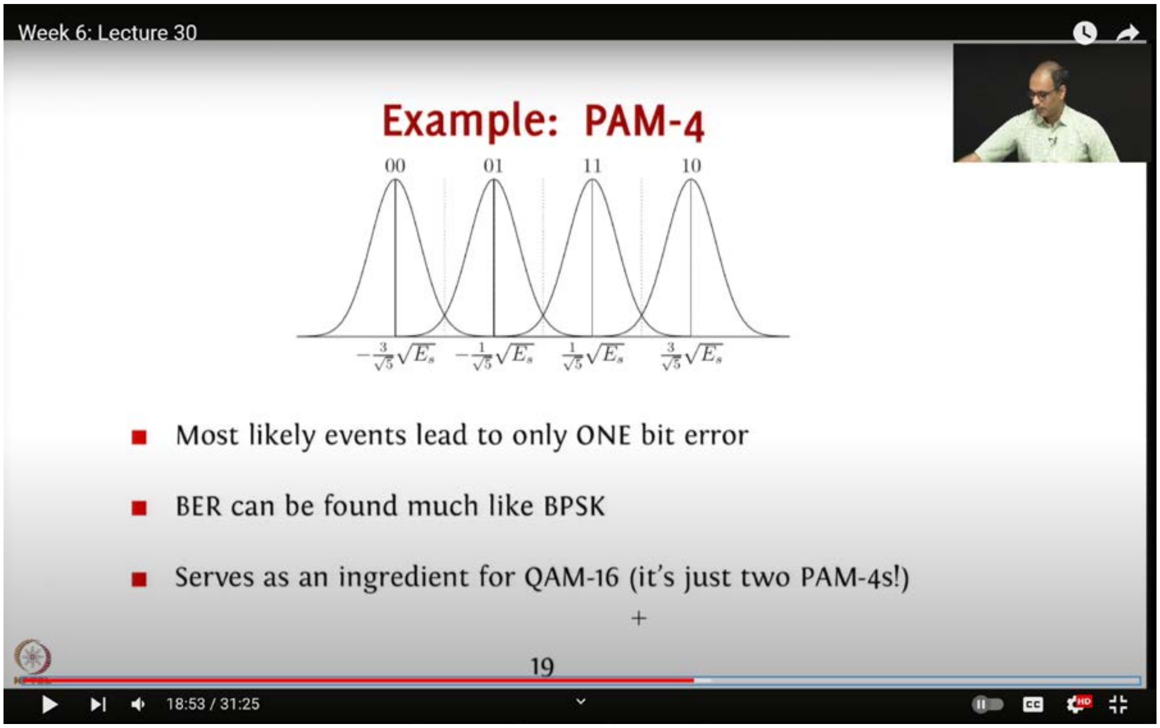
For BPSK, E_s is equivalent to E_b, meaning the energy per symbol is equal to the energy per bit. On the other hand, in QPSK, E_s is actually $2E_b$. This difference arises because, in QPSK, each symbol carries two bits, and therefore, the power is divided between both the real and imaginary components. In BPSK, the noise affects only the real component, whereas, in QPSK, both the real and imaginary components are subject to noise.

Now, considering $Q\left(\sqrt{\frac{2E_b}{N_0}}\right)$ as the bit error rate probability for the MSB in QPSK, it should be noted that due to symmetry, the LSB will have the same error probability. Thus, the bit error probability for QPSK can be computed using this understanding.

However, a critical point to correct is that the expression $Q\left(\sqrt{\frac{2E_b}{N_0}}\right)$ should actually correspond to $Q\left(\sqrt{\frac{E_s}{N_0}}\right)$ for QPSK, given that $E_s = 2E_b$. While the formula appears similar in form, the crucial difference is the relationship between E_s and E_b. For BPSK, E_s equals E_b, but for QPSK, E_s equals $2E_b$, and this distinction must be emphasized.

Next, let's delve into the case of PAM-4 modulation. But before we proceed, a quick recap on the symbol error rate for QPSK is warranted. Recall that the probability of making a bit error, given by $Q\left(\sqrt{\frac{E_s}{N_0}}\right)$, can be used to compute the symbol error probability. A symbol error occurs when at least one bit error has occurred. This can be computed by considering the event where no bit error occurs and subtracting it from one, thus giving us the probability of at least one bit error.

(Refer Slide Time: 18:53)



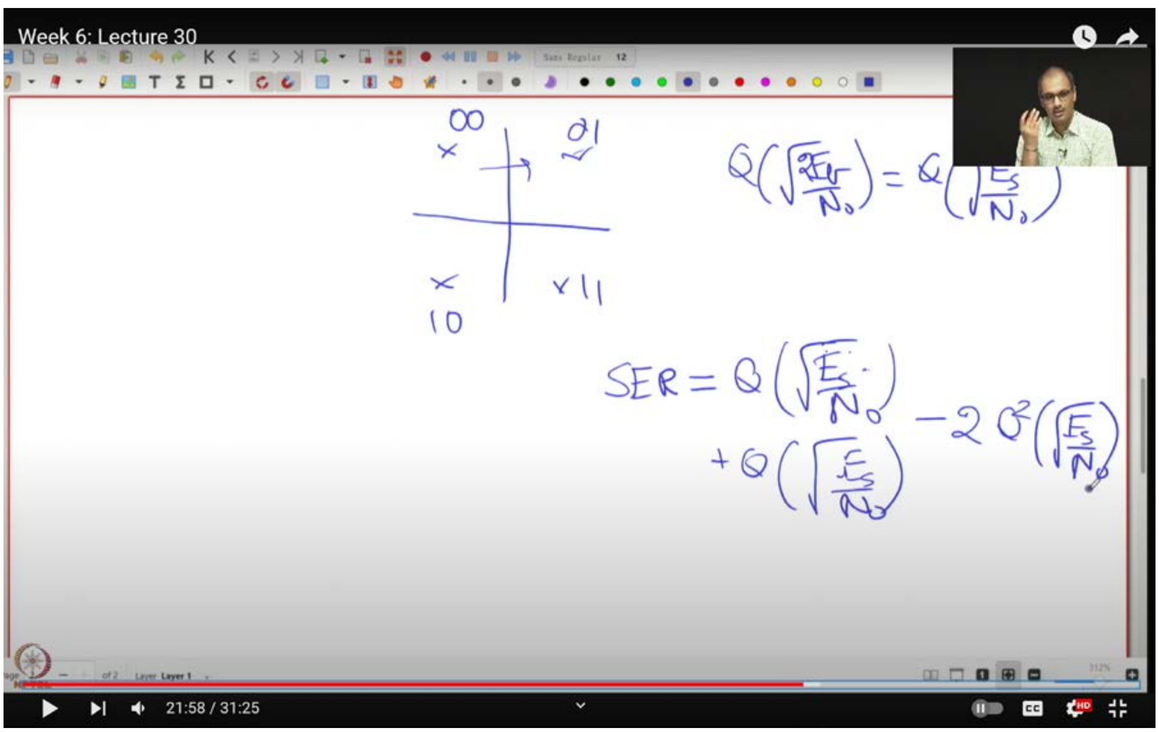
In the context of PAM-4, let's explore what happens. PAM-4 assigns pairs of bits to each symbol, but it's important to be cautious in this assignment. For instance, assigning bits as 00, 01, 10, and 11 may seem straightforward, but if two symbols are close neighbors in the constellation, a single symbol error can result in two bit errors. This is particularly problematic because such symbol errors are more likely to occur when symbols are close together.

Therefore, this initial assignment is not ideal. Instead, consider reassigning the bits, perhaps as 11, 10, 01, and 00. In this arrangement, when a symbol error occurs between close neighbors, it results in only one bit error, which is a more desirable outcome.

So, careful consideration of the power allocation in modulation schemes like BPSK and QPSK is crucial for understanding their error rates. Similarly, in PAM-4, proper bit assignment is essential to minimize the impact of symbol errors on the overall bit error rate.

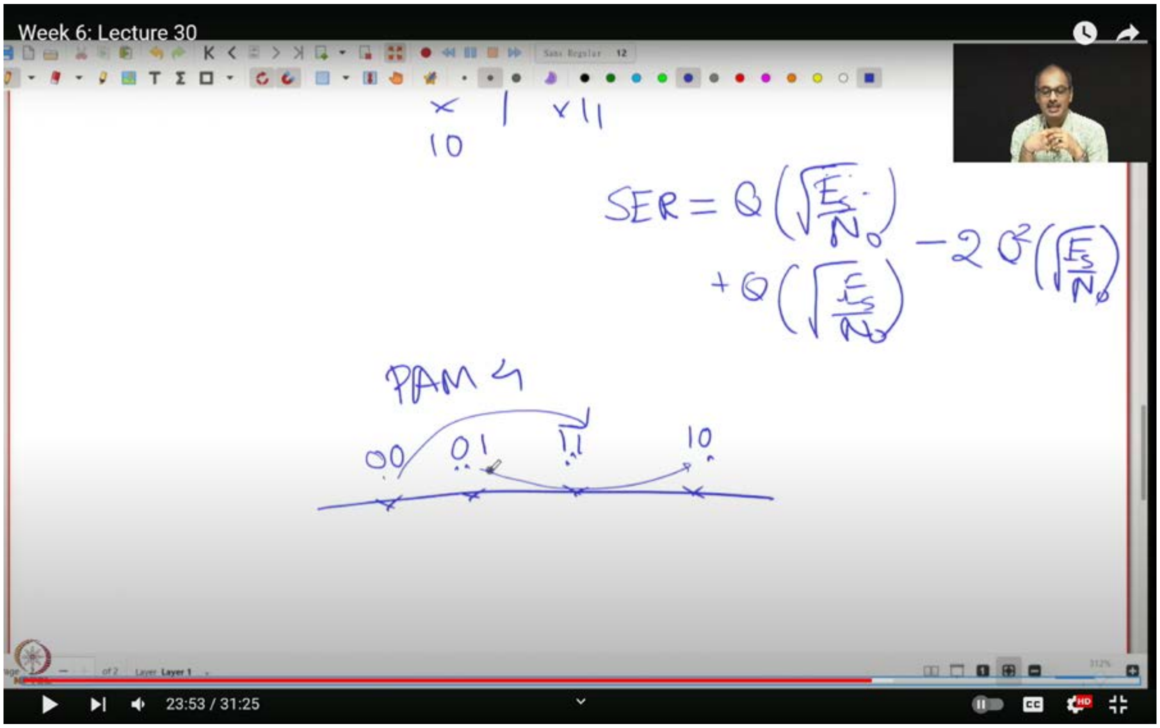
In this scenario, the bit error rate (BER) is minimized due to the implementation of Gray coding, which ensures that a flip of a single bit or symbol results in only one bit error in the most likely symbol flip events. Of course, one might argue, "What if a transition occurs from one symbol to a more distant symbol?" Indeed, in such cases, two bits might be incorrect. However, these errors are less probable because the symbols are much farther apart, essentially twice as far, making these events significantly less likely to occur. Therefore, when using PAM-4 modulation, employing Gray coding is essential. As demonstrated with the sequence 00, 01, 11, 10, Gray coding ensures that the most likely errors lead to only one bit error.

(Refer Slide Time: 21:58)



To calculate the BER for PAM-4, the approach is somewhat similar to that of BPSK, but with a slight complexity. The issue arises because you need to calculate the BER separately for different pairs of symbols. For instance, consider the transition from 00 to 01. To determine the probability of this specific error, you would need to calculate the probability of the 00 symbol landing in the 01 region, which is where the least significant bit (LSB) flips. This involves integrating the Gaussian error function over this specific region.

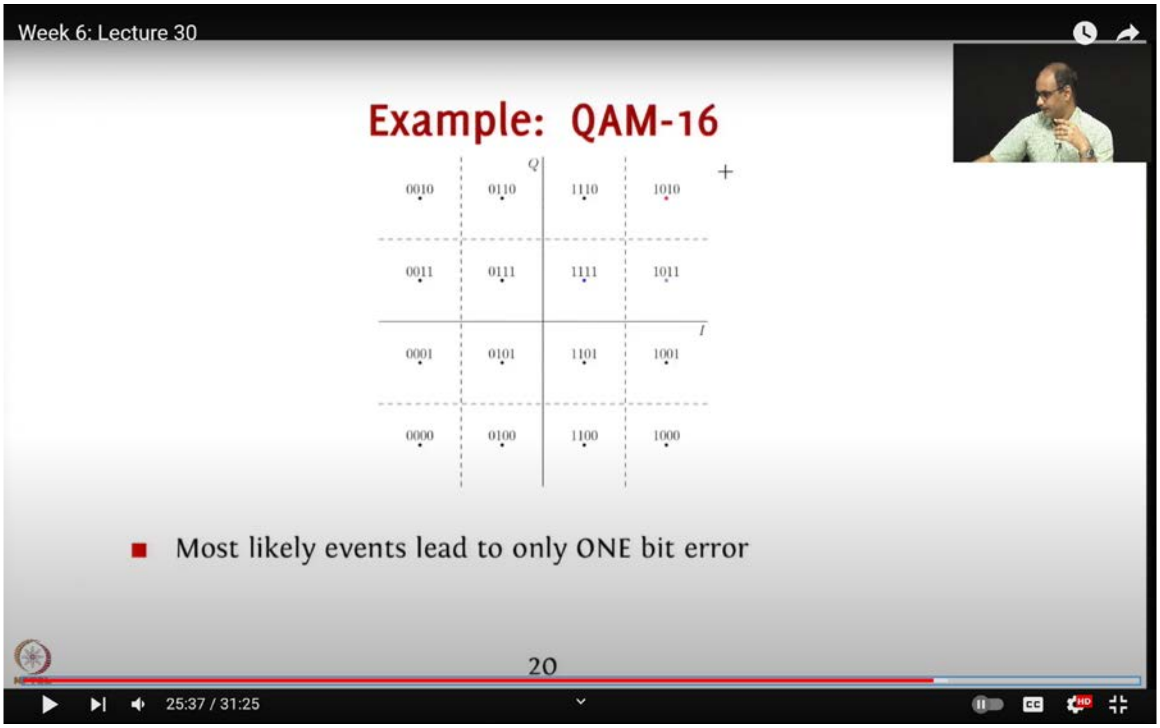
(Refer Slide Time: 23:53)



However, for other symbol pairs, the situation becomes a bit more complicated. For instance, if the symbol 10 transitions to either 00 or 11, there are two possible error events to consider. Moving left from 10 to 00 flips the MSB, while moving right from 10 to 11 keeps the MSB intact but changes the LSB. You must be meticulous in defining the exact integration regions for these transitions. Nevertheless, if you find this too complex to calculate analytically, you can always run a simulation to estimate the BER for PAM-4.

It's also worth noting that PAM-4 serves as a fundamental building block for QAM-16. When working with QAM-16, the goal is to allocate bits in such a way that the most likely symbol error events result in only one bit error. In this case, using Gray coding ensures that this requirement is met.

(Refer Slide Time: 25:37)



Let's review the bit allocation strategy in Gray coding for QAM-16. Suppose we start with the symbol 0000. The two nearest neighbors might be 0001 and 0100. The objective is to ensure that each transition leads to exactly one bit error. For example, moving from 0000 to 0001 flips only the LSB, and moving from 0000 to 0100 flips only one other bit. This approach ensures that the most probable error events lead to minimal bit errors.

Finally, let’s consider another symbol, say 0100. Its nearest neighbor could be 0101, where the LSB flips, or 1100, where only the MSB flips. Again, the Gray coding approach ensures that each of these transitions results in only one bit flip, minimizing the BER for QAM-16.

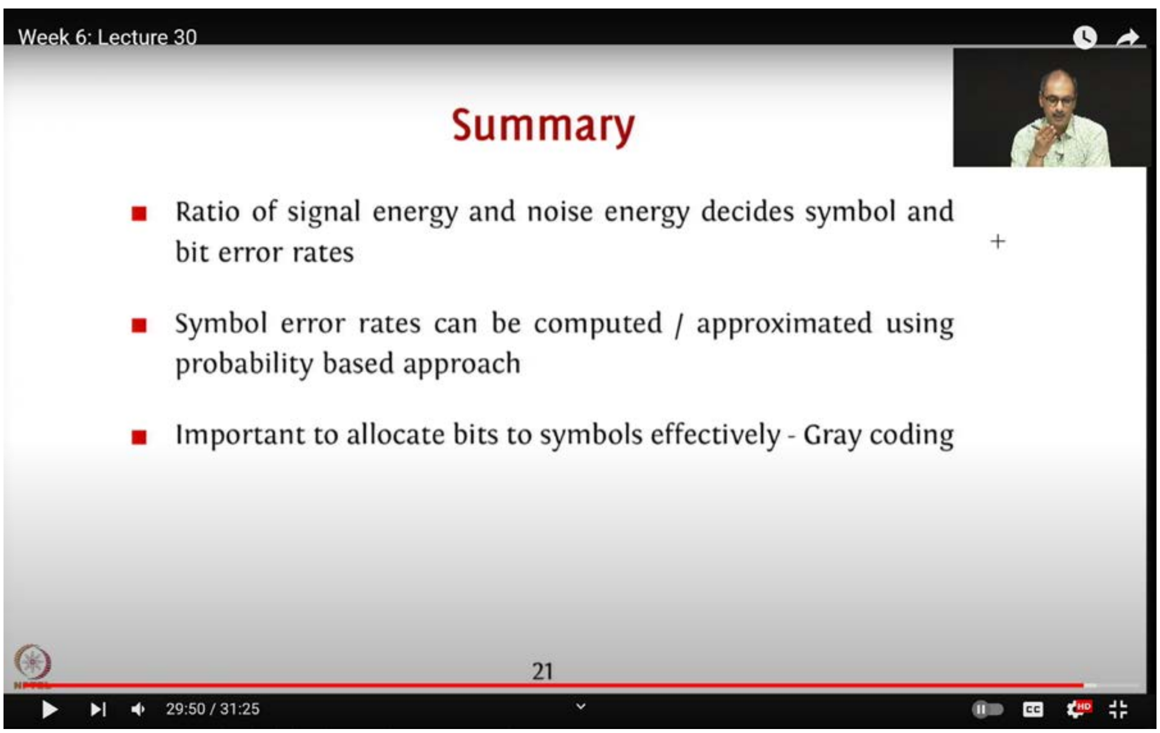
Here, the strategic use of Gray coding in modulation schemes like PAM-4 and QAM-16 effectively minimizes the BER by ensuring that the most probable symbol errors cause only a single bit error. This careful bit allocation is crucial in optimizing the performance of digital communication systems.

Next, let's examine this particular symbol, 0101. If we look below it, there's only one bit difference. To the left, there's only one bit difference as well. Moving to the right, from

0101 to 1100, there's again just one bit flip. Finally, checking above, the transition from 0111 to 0101 also results in a single bit change. This configuration ensures that all four nearest neighbors for this symbol, as well as the adjacent neighbors for other symbols, involve only one bit flip. This is an extension of the Gray coding concept applied to PAM-4.

Another way to look at this is through the most significant bit (MSB). For instance, the MSB is 0 on the left and 1 on the right, which provides a clear structure for the constellation. Similarly, when you examine the least significant bit (LSB), a pattern emerges: 0110, 0110, and so on. This allocation of bits to symbols is not unique, you can create your own Gray-coded constellation that performs just as well. The mapping you choose is flexible, as long as it maintains the Gray coding properties, meaning it will be equivalent in performance and functionality.

(Refer Slide Time: 29:50)



When computing bit error rates (BER), you can approach it by analyzing the MSB. Since 0 is on the left and 1 is on the right, you can use a BPSK-like method to calculate the BER for the MSB. However, for the other bits, such as the third bit, where the pattern might be 11 here and 00 there, you need to carefully decode the pattern. While this can be cumbersome, it's entirely doable. If you prefer not to go through this manual computation, or if your scenario doesn't require it, you can always perform a simulation to estimate the BER for the signal-to-noise ratio (SNR) you're interested in.

One important point to note here is that $4E_b$ corresponds to E_s, a factor to keep in mind during calculations.

To summarize, the ratio of signal energy to noise energy determines both symbol and bit error rates. You can compute symbol error rates using precise methods or approximate them using probability-based approaches like the union bound or nearest neighbor approximations. When it comes to bit error rates, the process is similar, but you must first consider how bits are allocated to symbols. Proper allocation is crucial to minimizing bit errors, particularly for the most likely symbol errors. Gray coding is an excellent strategy for achieving this, as it ensures that the most probable symbol errors result in only one bit error, thereby reducing the BER.

Up to this point, we've focused on the impact of noise on symbols, symbol error rates, and bit error rates. However, we've assumed an idealized communication system without any impairments other than noise. In the next lecture series, we'll explore how to relax these ideal assumptions about our transmitter-receiver system and their pairing, moving towards a more realistic communication system model. Thank you.