Digital Communication using GNU Radio

Prof. Kumar Appaiah

Department of Electrical Engineering

Indian Institute of Technology Bombay

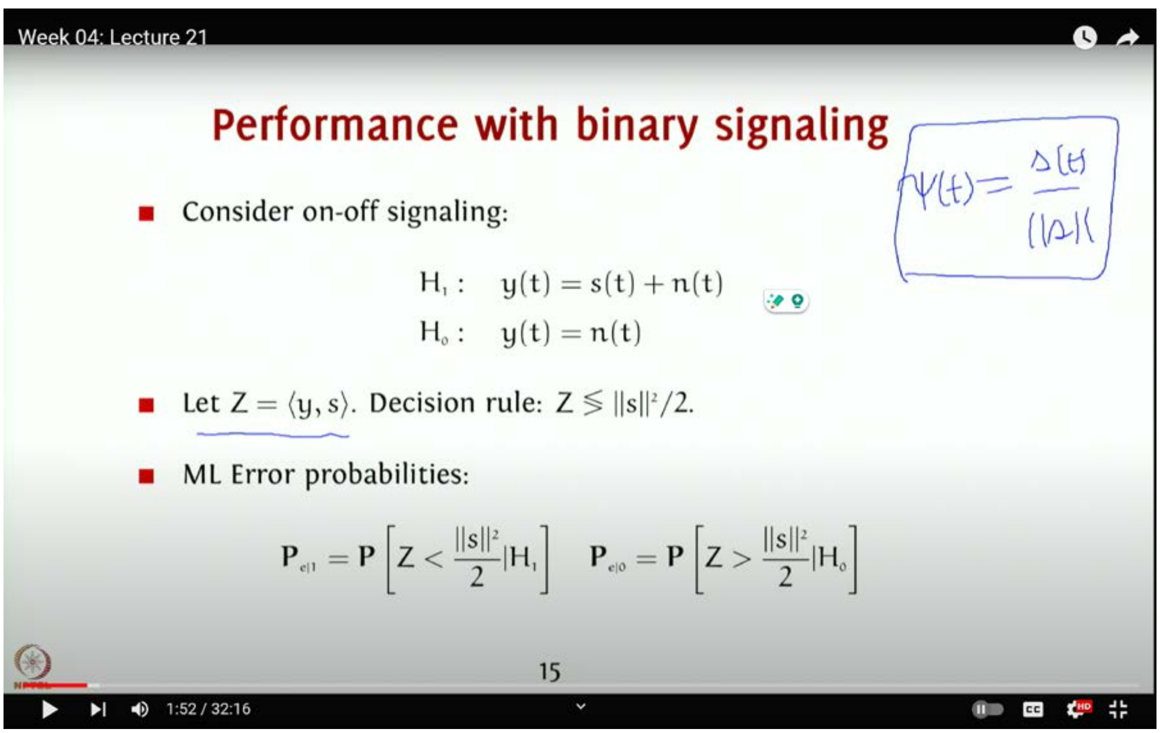
Week-04

Lecture-21

Detection and Optimal Decision for M-ary Signaling

Hello, and welcome back to this lecture series on Digital Communication using GNU Radio. I'm Kumar Appiah from the Department of Electrical Engineering at IIT Bombay. Today, we will be concluding our series on demodulation, at least from a theoretical standpoint. In this session, we will delve deeper into M-ary signal demodulation and subsequently explore the practical application of these methods using GNU Radio.

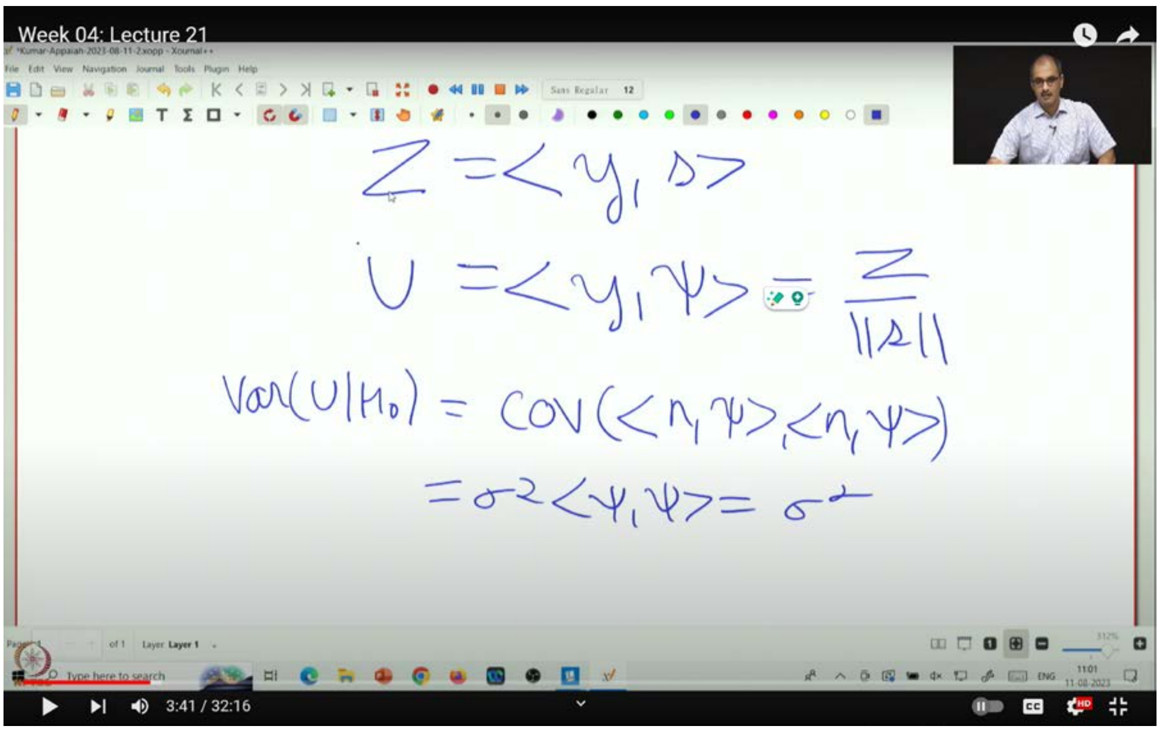
(Refer Slide Time: 01:52)



In our previous lecture, we had an extensive discussion on on-off signaling, specifically binary signaling where the transmitted signals are either 0 or S(t). Through this signaling method, we were able to analyze the optimal detection scheme and calculate the symbol error rate.

Now, I'd like to make a small but important clarification regarding our earlier discussion. We determined our decision rule based on the quantity S, and using S, we plotted our probability density functions (PDFs) and identified the decision regions with respect to $|S|^2$, whether it should be to the left or right of $|S|^2$, and so on. However, there is a slight modification I want to introduce.

(Refer Slide Time: 03:41)



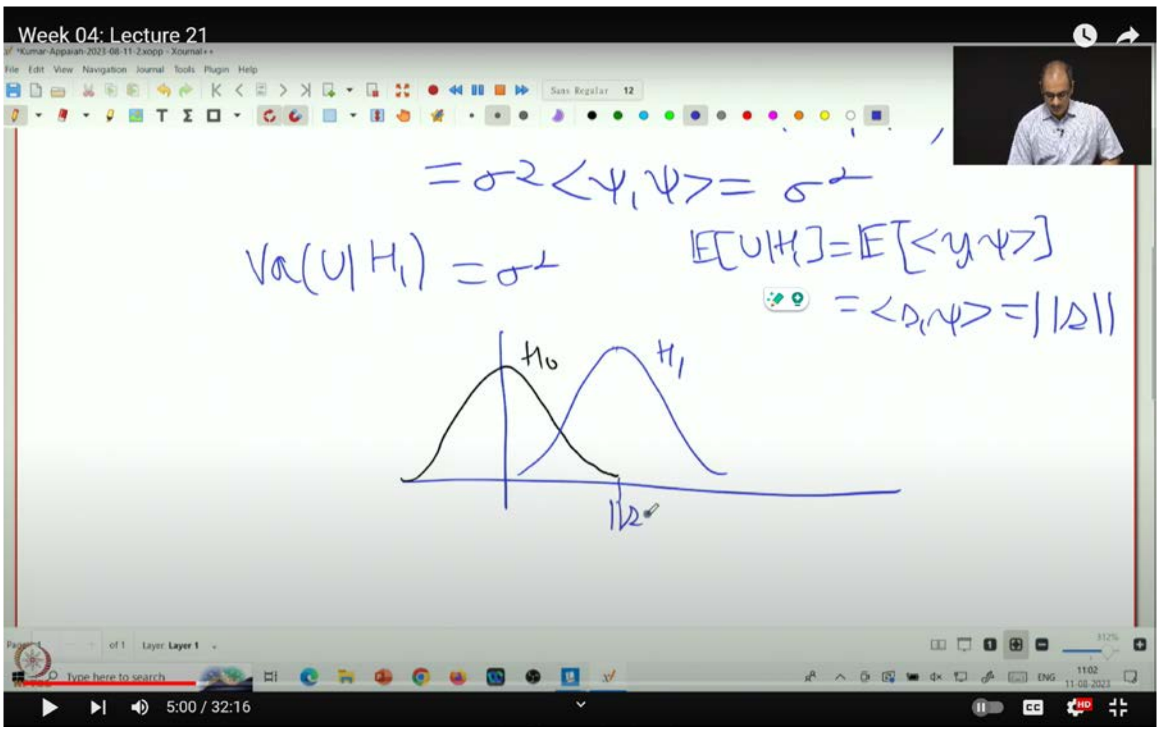
Earlier, we mentioned that ψ(t), a normalized version of S(t), can also be utilized as a decision mechanism. To provide some context, let's redefine our variable Z as $Y \cdot S$. Now, if we define U as the inner product $Y \cdot \psi$, we obtain a new metric U, which is essentially Z normalized by |S|. Although this modification doesn't alter the overall result, repeating the analysis with U reveals subtle differences.

For instance, when examining the relationship involving U, the expected value of Z given H_0 was $E[Z|H_0] = E[n \cdot S] = 0$, and the variance of Z given H_0 was $\text{Var}[Z|H_0] =$

$\text{Cov}(n \cdot S, n \cdot S)$. With the introduction of U, these relationships undergo an interesting change.

Specifically, the variance of U under H₀ becomes $\text{Var}[U|H_0] = \text{Cov}(\langle n, \psi \rangle, \langle n, \psi \rangle)$. Using the covariance formula $\text{Cov}(n \cdot v_1, n \cdot v_2) = \sigma^2 \langle v_1, v_2 \rangle$, this simplifies to $\sigma^2 \langle \psi, \psi \rangle = \sigma^2$. Similarly, the variance of U under H_1 also equals σ^2.

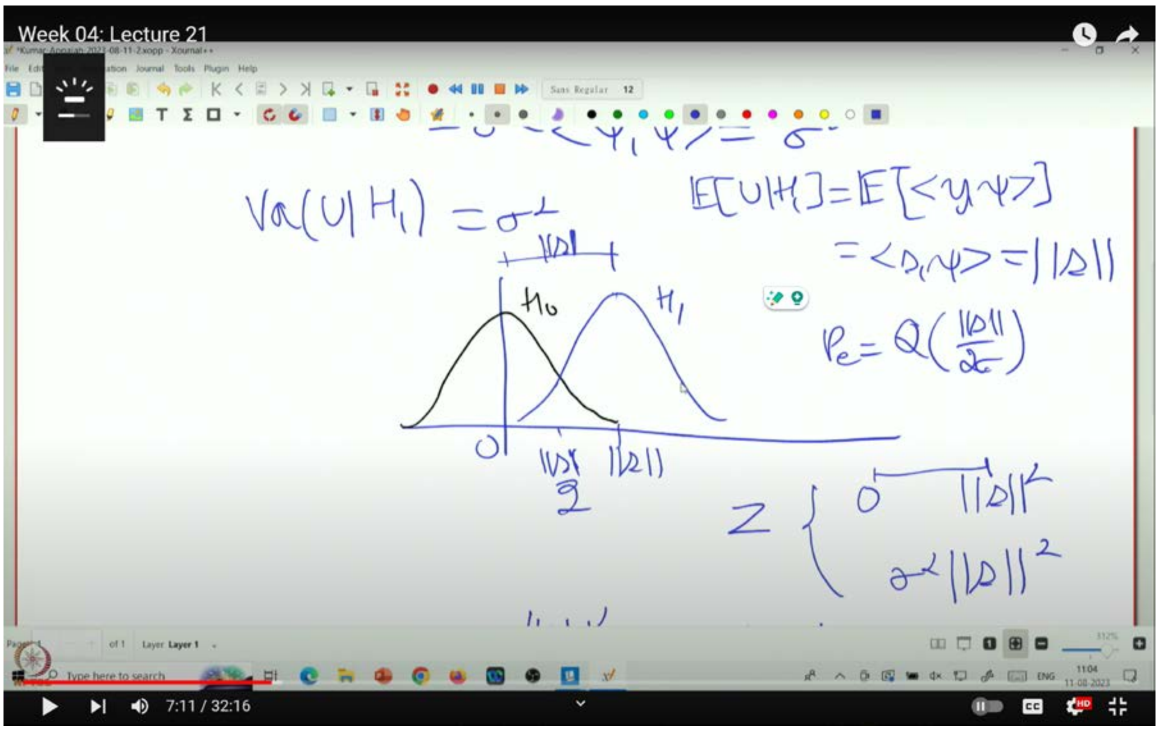
(Refer Slide Time: 05:00)



In this context, if we were to plot the PDFs under hypotheses H_1 and H_2 for U, both would have the same height, indicating an identical variance. However, there is a crucial detail in calculating the mean value of U under H_1. The expected value $E[U|H_1]$ would be $E[\langle Y, \psi \rangle]$, where Y = S + n. Since $\langle S, \psi \rangle$ is equivalent to |S|, you will find that U under H_1 has a mean of |S| and a variance of σ^2. In essence, what we've done is scale the entire problem by |S|, which leads to these slight yet insightful changes in our analysis.

Now, let's discuss a subtle yet important advantage in this context. When calculating the probability of symbol error, you will arrive at the same result as before when using S, which is logical. However, an interesting observation here is that the distance, given by |S| divided by 2σ (where σ is the standard deviation of the noise), becomes the critical term in your Q-function. Recall that the probability of error, P_E, for equiprobable signaling was |S| over 2σ. This is determined by finding the distance between the means of the signals being transmitted, specifically, between 0 and S(t), which corresponds to a distance of |S| over 2σ along this axis.

(Refer Slide Time: 07:11)

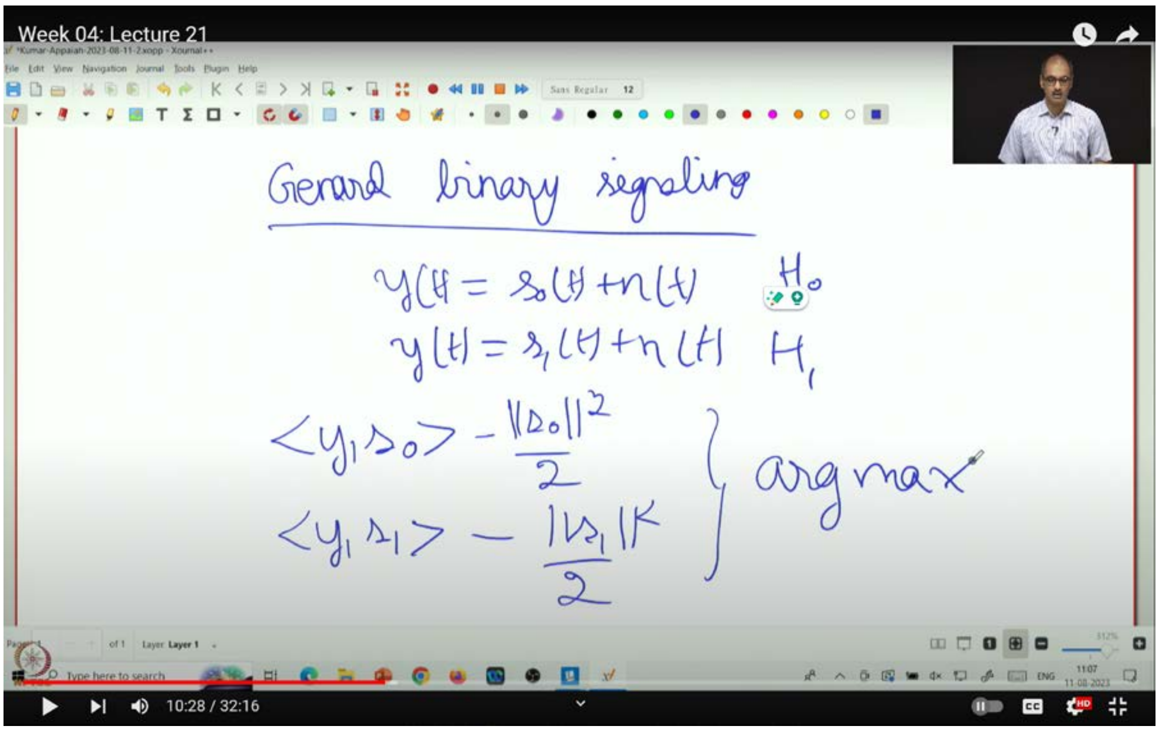


If we revisit our previous discussion on Z, you'll recall that the means were 0 and $|S|^2$, with the noise variance being $\sigma^2|S|^2$. If you compute the distance between these two means and divide it by the square root of the variance, you will again find the result $|S|^2$ over 2σ |S|, which is consistent with our earlier findings. This implies that whether you take the inner product $Z \cdot S$ or $Z \cdot \psi$ (or any scaled version of S), the result will be the same.

The only minor benefit of using the inner product $Y \cdot \psi$ is that it gives you a number already scaled by |S|, though the outcome remains unchanged. This is just a useful insight: the choice of taking the inner product $Y \cdot S$ is not unique; any scaled version of S will yield the correct result.

Now, let's shift our focus to a more general form of binary signaling, where we consider two signals, S_0 and S_1. Here, it's important to be cautious: simply writing S_0 and S_1 doesn't automatically imply that there are two dimensions involved. For instance, suppose S_0 is something like ψ_1 and S_1 is $-\psi_1$ or $4\psi_1$; in such cases, the dimensionality could be anywhere from one to two, depending on your choice of S_0 and S_1.

(Refer Slide Time: 10:28)



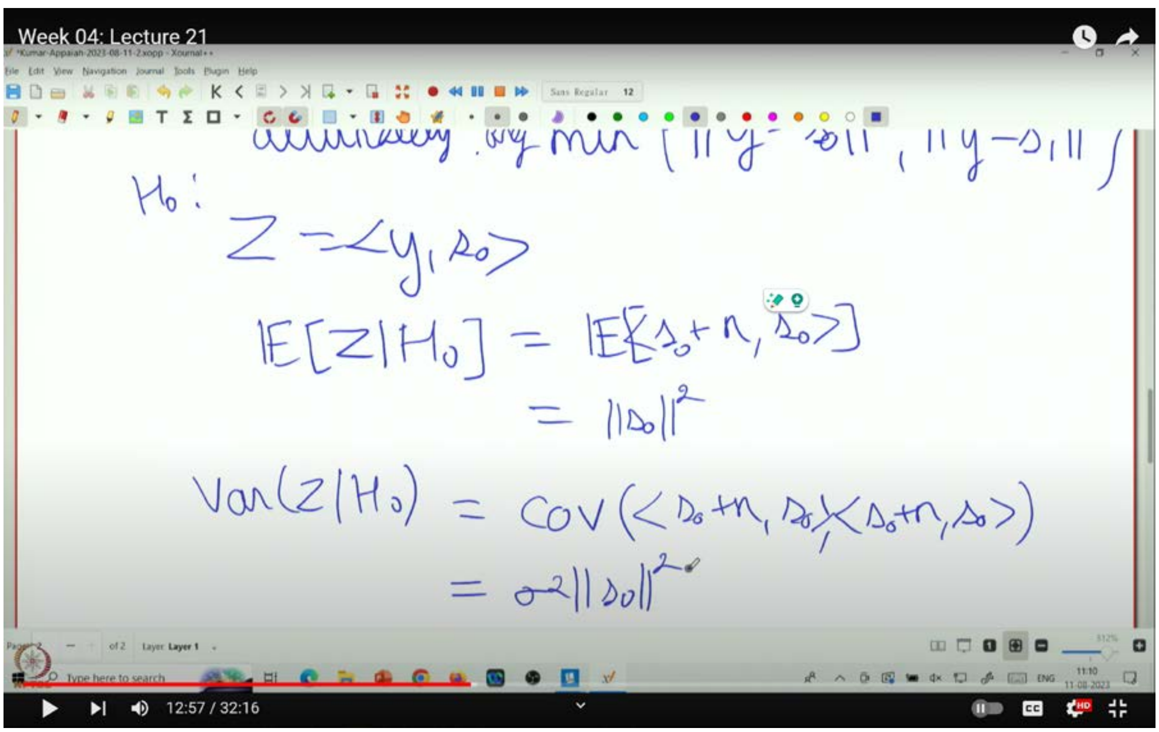
Let's analyze this general scenario. Similar to how we previously took the inner product with S in the on-off case, here we'll consider the inner product with S_0 and S_1.

For general binary signaling, let's express the problem as follows: if $Y(t) = S_0(t) + n(t)$ or $Y(t) = S_1(t) + n(t)$, it means that when you transmit 0, $S_0(t)$ is sent; when you transmit 1, $S_1(t)$ is sent. No surprises here, if you set S_0 to the zero signal, you end up with the same scenario as in the previous on-off signaling situation.

Now, let's determine the optimal decision region using the same tools as before. The process is nearly identical. Take the inner product $Y \cdot S_0 - |S_0|^2/2$. Why? Because this was our decision criterion. Previously, we minimized $|Y - S_i|^2$ and, after expanding, removed $|Y|^2$ and multiplied by a negative sign to maximize the resulting quantity. Similarly, here you must determine which value is larger; hence, find the argmax. If $\langle Y, S_0 \rangle - |S_0|^2/2$ is larger, conclude that 0 was sent. If $\langle Y, S_1 \rangle - |S_1|^2/2$ is larger, conclude that 1 was sent. It's straightforward.

Alternatively, you can minimize or find the argmin of the distance, which essentially results in the same outcome. In this context, you can refer to these specific variables as Z. For example, let $Z = \langle Y, S_0 \rangle$ under hypothesis H_0. To find the expectation of Z under H_0, it’s simple, this expectation will be $< S_0, S_0 >$ plus the expected value of the noise term $<n,S_0>$. Without diving into the complete details, this simplifies to $|S_0|^2$.

(Refer Slide Time: 12:57)



Let's delve into the variance of Z given the hypothesis H_0. As you might expect, the variance is closely related to the covariance. Here, you must be precise. Under hypothesis H_0, Y can be expressed as $S_0 + n$, where n is the noise component. Therefore, the variance is computed from the expression $\mathrm{Cov}(S_0 + n, S_0 + n)$.

If you expand this expression, the term involving $|S_0|^2$ will emerge, simplifying the calculations. A similar procedure applies if you were to evaluate the variance under a different hypothesis, say H_1. Often, $|S_0|^2$ is chosen to be equal to $|S_1|^2$, which simplifies the decision region, as both signals have the same energy.

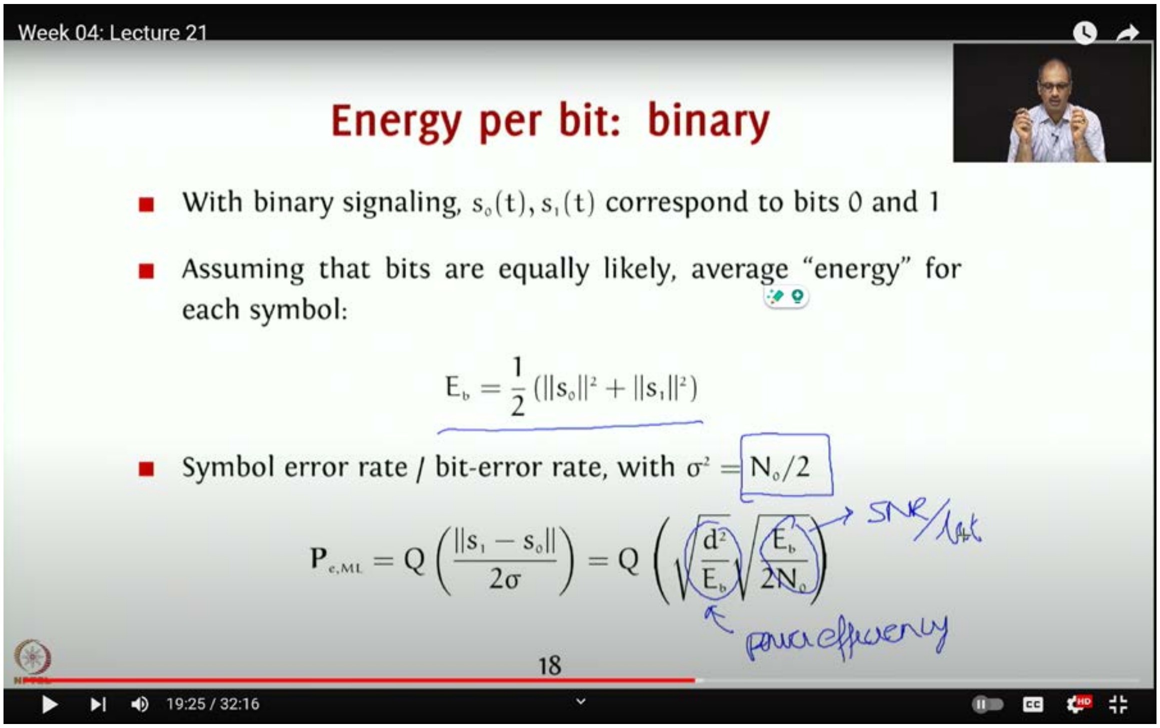
Although we're not going into a full evaluation here, let's summarize the decision-making process: the decision region is determined by comparing the inner products $\langle Y, S_0\rangle - |S_0|^2/2$ and $\langle Y, S_1\rangle - |S_1|^2/2$. If you simplify this by moving terms involving S_0 and S_1 to opposite sides, you get $\langle Y, S_1 - S_0\rangle$. The decision rule then becomes whether this quantity is greater or less than $|S_1|^2/2 - |S_0|^2/2$. This forms your decision region.

There's also an intuitive understanding here: when you evaluate the probability of symbol error in this setup, you derive $Q\left(\frac{|S_1 - S_0|}{2\sigma}\right)$, where d = $|S_1 - S_0|$ is the distance between the pair of signals. This result aligns with our intuition because, in the on-off signaling example, where S_0 was essentially zero, you obtained $Q\left(\frac{|S|}{2\sigma}\right)$, where S_1 or S was the non-zero signal. Therefore, $Q\left(\frac{d}{2\sigma}\right)$ serves as a crucial check. Whenever you're dealing with binary signaling under Gaussian noise, the optimal detector will yield a probability of symbol error given by $Q\left(\frac{d}{2\sigma}\right)$, where σ is the noise variance, and d is the distance between the two signals.

It's essential to note that this result holds true only under the assumption of equiprobable signaling, where Maximum Likelihood (ML) detection coincides with Minimum Probability of Error (MPE) detection. If the signals are not equiprobable, the decision region shifts, and this needs to be considered carefully.

Therefore, the key idea is to identify the point where the greater-than or less-than condition holds, as this defines your decision region. Typically, this decision point lies at the midpoint between S_0 and S_1 if these signals are well chosen. In general binary signaling, the expression $Q\left(\frac{d}{2\sigma}\right)$ also applies to mRNA signaling scenarios.

(Refer Slide Time: 19:25)



Now, let's consider binary signaling with S_0 and S_1, which can be associated with the bits 0 and 1. We haven't yet discussed the energy consumption aspect. Energy in a signal, as you know from basic circuit theory, is the integral of the square of the signal over the interval from 0 to T. For our purposes, $|S_0|^2$ is the energy of S_0, computed as $\int_0^T S_0^2(t)dt$, and similarly for S_1.

When these two signals are transmitted with equal probability, since 0s and 1s occur half the time, the energy per bit, E_b, can be defined as $\frac{1}{2}(|S_0|^2 + |S_1|^2)$. Here, $|S_0|^2$ and $|S_1|^2$ correspond to $\int_0^T S_0^2(t)dt$ and $\int_0^T S_1^2(t)dt$, respectively.

An interesting conclusion arises when considering the symbol error rate or bit error rate in binary signaling. Both rates are the same when $\sigma^2 = \frac{N_0}{2}$, where N_0 is the noise power spectral density. We specifically choose $\frac{N_0}{2}$ as the noise variance per dimension, reflecting the standard assumption in communication theory.

The probability of error under maximum likelihood detection is given by $Q\left(\frac{d}{2\sigma}\right)$, where d represents the norm $|S_1 - S_0|$, and σ is the noise standard deviation. This can be further expressed as $Q\left(\sqrt{\frac{d^2}{E_b}} \cdot \sqrt{\frac{E_b}{2N_0}}\right)$. This expression is quite intriguing because the ratio $\frac{E_b}{N_0}$ acts as a signal-to-noise ratio (SNR) on a per-bit basis. In this context, $\frac{d^2}{E_b}$ can be interpreted as a measure of power efficiency.

To unpack this further, consider binary signaling where the bit 0 corresponds to signal S_0, and bit 1 corresponds to signal S_1. The average energy expenditure is $\frac{|S_1|^2}{2}$. Now, if you wish to increase the distance d between S_0 and S_1, you must increase the amplitude of S_1, effectively pushing it further away from S_0. This increase in amplitude, however, comes at a cost: as the amplitude grows, so does the energy consumption, whether measured in joules or in power (joules per second). Therefore, increasing d improves performance by reducing the bit error rate, but it requires more energy.

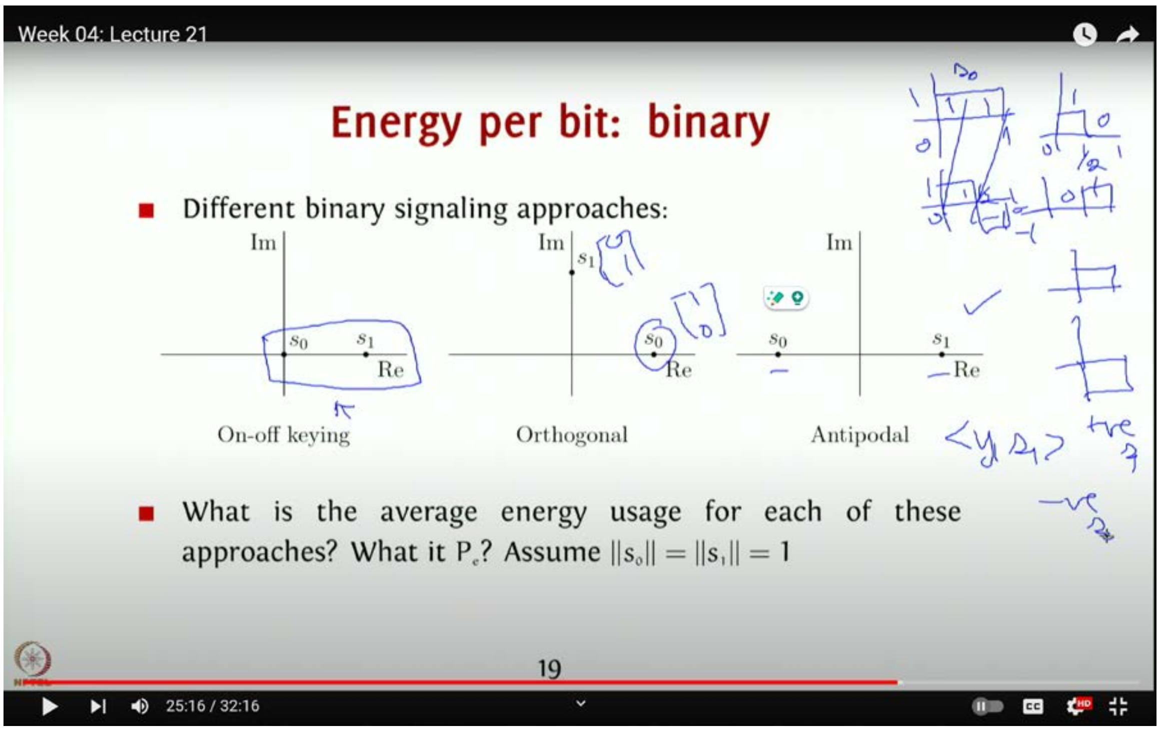
This trade-off is critical in designing communication systems. If you arrange your signal constellation points (e.g., S_0, S_1, S_2) such that they are spaced far apart, your system will exhibit excellent performance with a very low bit error rate. However, this comes with the caveat of increased energy consumption.

Another important factor is the SNR per bit, which effectively dictates the performance of your system, not the signal power or noise power individually. In other words, a system with very low noise can achieve a desirable bit error rate with minimal signal power.

To illustrate, consider a communication system where you require a bit error rate of 10^{-9}. If the noise variance is 1, you might need 100 joules of signal power. However, if the noise variance drops to 0.1, the required signal power may reduce to just 10 joules. Further reducing the noise variance to 0.01 might require only 1 joule. This demonstrates that it is the ratio of signal power to noise power, SNR, that fundamentally determines system performance, not the absolute values of these quantities.

For a practical example, consider a communication system operating in a narrow frequency band where the noise is very low. In such a scenario, the required signal power to achieve a certain bit error rate is also low. However, in a high-noise environment, perhaps due to high radiation levels, the required signal power must be significantly increased to maintain the same bit error rate. For instance, transmitting a signal across a room requires some power, but transmitting to a satellite, potentially hundreds of kilometers away, demands much more power to counteract the noise.

(Refer Slide Time: 25:16)



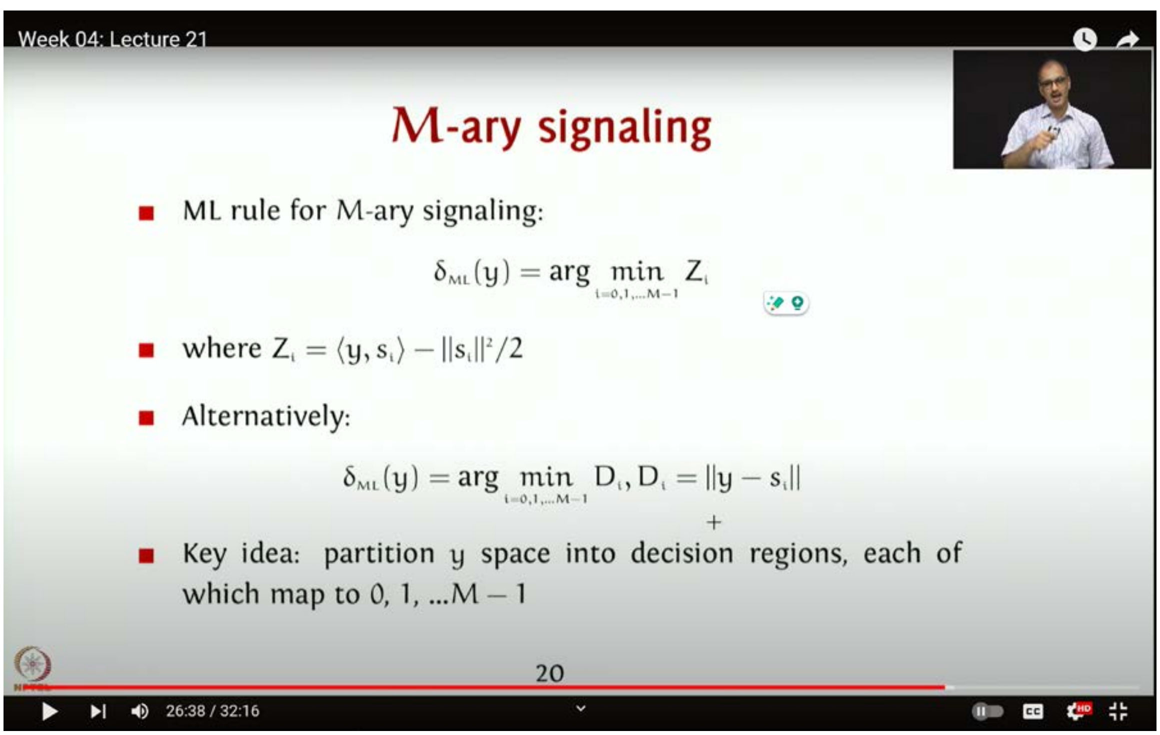
Therefore, the signal-to-noise ratio is the critical metric in determining system performance. The term $\frac{d^2}{E_b}$ represents power efficiency: the larger d is, the more power you consume. While a lower d improves power efficiency, you must still meet your bit error rate requirements.

Understanding binary signaling is foundational since M-ary signaling is essentially an extension of these principles. For instance, consider on-off keying where $S_0 = 0$ and $S_1 = 1$. Here, the error probability is given by $Q\left(\frac{|S_1|}{2\sigma}\right)$. However, signals can also be placed in different configurations. For example, S_0 and S_1 can be made orthogonal. Can you think of examples of binary orthogonal signaling? There are several, and exploring them further can provide deeper insights into efficient communication system design.

Let's consider a scenario where you have two signals, S_0 and S_1. Imagine that S_0 corresponds to the vector [1, 0] and S_1 to the vector [0, 1]. Clearly, these two vectors are orthogonal. If we visualize this, assume that the coordinates are labeled as 1, 0, -1, 0.5, and so on. In this context, it becomes apparent that these signals are orthogonal by nature.

Now, if you carefully select the basis vectors, such as [1, 1] and [1, -1], the orthogonality becomes even more evident. However, in this case, we're using the vectors [1, 0] and [0, 1], which still represent orthogonal signals. When you use orthogonal signals, you end up with a specific type of signal constellation. This constellation will influence the error performance, depending on how far S_0 and S_1 are from the origin, among other factors. Although the basic process of calculating inner products (like $\langle y, S_0 \rangle$ and $< y, S_1 >$) remains the same, the actual implementation may vary slightly depending on the scenario.

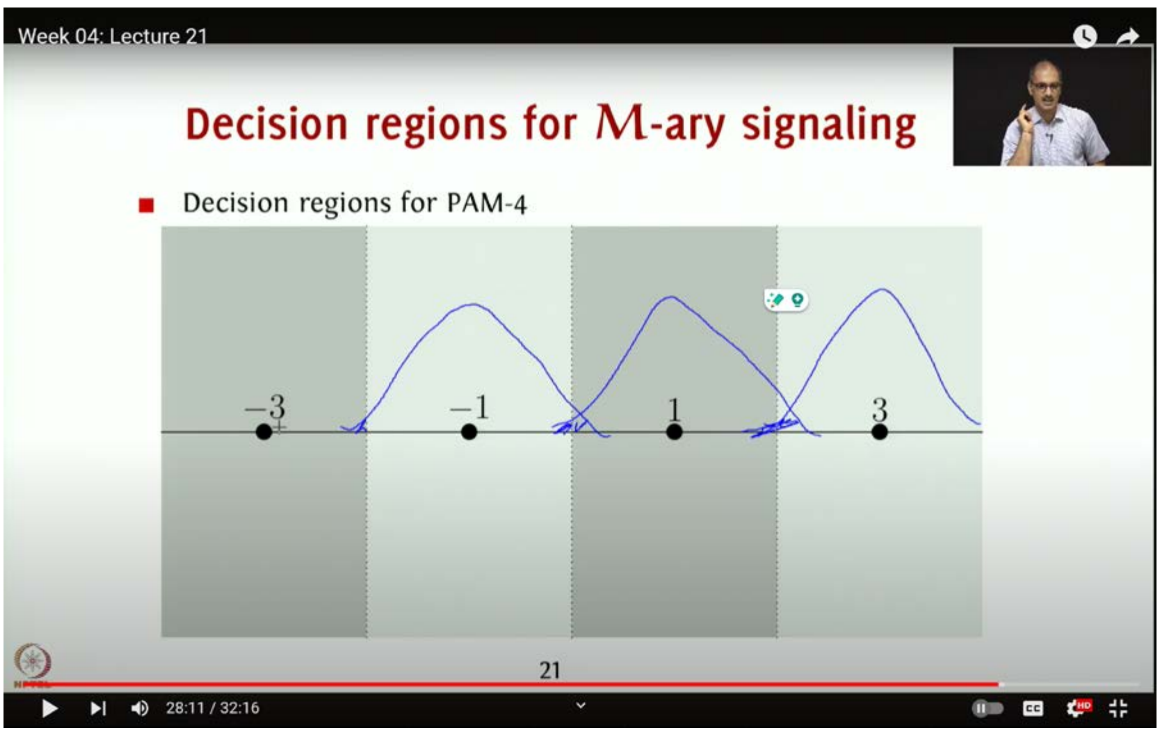
(Refer Slide Time: 26:38)



One particularly interesting case is antipodal signaling. This type of signaling can be seen as an extension of on-off keying. Here, the signals S_0 and S_1 are typically equidistant from the origin but in opposite directions. The dimension of the signaling in this case is exactly one, similar to on-off keying. You can choose various signal forms, such as a triangular pulse or a sinc function, where one signal is the negative of the other.

In binary signaling, a fascinating simplification occurs. If you write out the detector as $< y, S_0 >$ and $< y, S_1 >$ and work through the math, you'll find that, assuming equally probable signaling, the decision boundary is simply whether the result is positive or negative relative to zero. You just take the sign of $z = < y, S_1 >$. If the result is positive, the signal is S_1; if negative, it's S_0. This leads to a very elegant detection method where the error probability is given by $Q\left(\frac{d}{2\sigma}\right)$. You can also verify the average energy usage for these cases, which is an important consideration in practical systems.

(Refer Slide Time: 28:11)

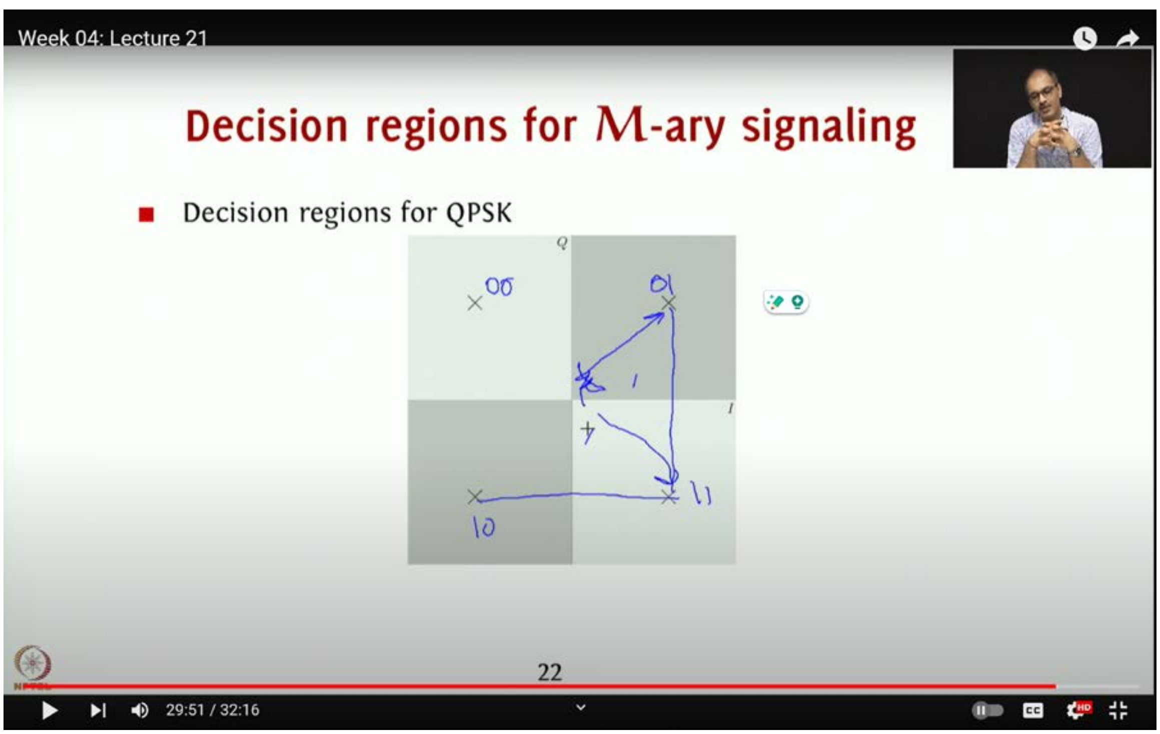


Moving on to M-ary signaling, the maximum likelihood (ML) detection rule doesn't change significantly from the binary case. The decision rule becomes $\delta_{ML}(y) = \arg\min_i \{\langle y, S_i\rangle -$

$\frac{|S_i|^2}{2}\}$, where i runs from 1 to M-1. The key difference is that there are now more signal points S_i. Alternatively, you could express this as finding the signal point with the minimum Euclidean distance from the received signal y, i.e., $\arg\min_i |y - S_i|$. This is essentially finding the signal point closest to the received signal in terms of squared Euclidean distance.

The concept can be extended to practical modulation schemes like 4-PAM (Pulse Amplitude Modulation). In this case, the signaling is one-dimensional, with possible signal values S_0, S_1, S_2, S_3 corresponding to scaled versions of the same basic signal, such as -3, -1, 1, and 3. The optimal decision regions are then defined by boundaries such as to the left of -2 being associated with -3, and to the right of -2 being associated with -1, and so on. To determine the symbol error rate, you place a Gaussian distribution over the signal points and calculate the probability that the noise causes the received signal to fall into an incorrect region.

(Refer Slide Time: 29:51)

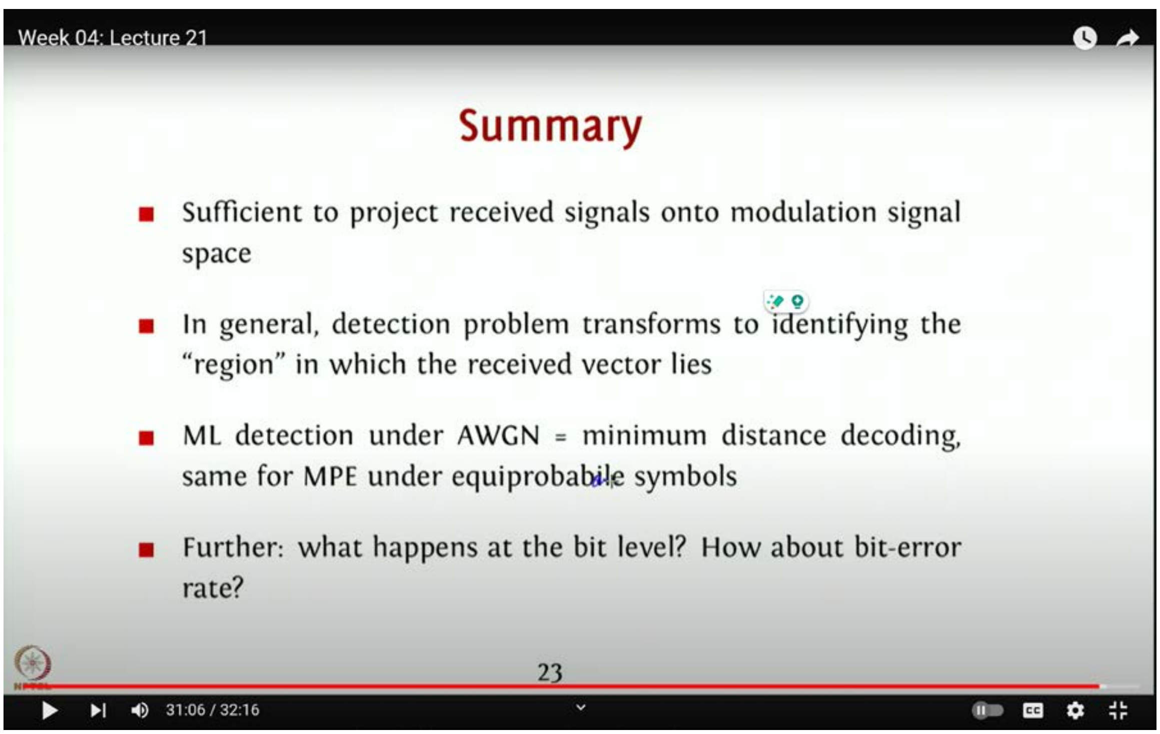


However, the calculation of the symbol error rate for certain points, like 1, requires you to consider errors from multiple regions, leading to a combination of Q functions. For equiprobable signaling, this calculation can be simplified because the error rates for symmetric signal points (e.g., 1 and -1) are identical. Similarly, the error rates for the outermost points, such as -3 and 3, will also be the same.

When working with signals like those at 3 and -3, you can directly use the cue. However, for signals at 1 and -1, you'll need to use two cues because the signal could deviate to either side. It's important to exercise caution here, and this is an exercise you should try on your own to solidify your understanding.

Moving on to QPSK (Quadrature Phase Shift Keying), the decision regions become quite fascinating. These regions are divided into four parts. Why is that? Imagine you're standing at a particular point where the signal is received. If you're here, this point is obviously the closest. But if you shift slightly, another point becomes the nearest. So, each decision region corresponds to the area where a specific signal point is closest.

(Refer Slide Time: 31:06)



Now, depending on how you view this, QPSK can be interpreted in two ways. If you consider it in terms of real numbers, it appears two-dimensional. But if you think in terms of complex numbers, it's one-dimensional. However, analyzing it as a combination of real signals simplifies the problem. You must decide whether you're above or below the x-axis, as well as whether you're to the left or right of the y-axis.

This leads to an intriguing concept regarding bit assignment. For instance, consider a bit assignment like 00, 01, 11, and 10. Notice how the bit 0 is consistently located in the upper half of the plane, while bit 1 is in the lower half. So, determining whether you're above or below the x-axis reveals the most significant bit. Similarly, the least significant bit remains the same to the left and right of the y-axis. Thus, identifying whether you're to the left or right of the y-axis informs you about the least significant bit.

In fact, one of the key insights here is that QPSK can be viewed as two BPSK (Binary Phase Shift Keying) schemes combined when you analyze it from a bit perspective. This is something you'll explore and confirm in future classes.

To summarize what we've covered in recent lectures, we delved deeply into how demodulation works and how it enables the detection of transmitted signals. We first established that it's sufficient to project the received signals onto the modulation signal space. In general, the detection problem then reduces to identifying the region in which the received vector lies. This outcome naturally follows from the optimal decision being based on minimizing the distance to the correct signal point.

For example, in QPSK, you observed that each region corresponds to the point with the minimum distance. Identifying where your received signal lies within these regions automatically partitions your decision space.

When using maximum likelihood detection under additive white Gaussian noise, this minimum distance decoding is optimal. The same holds true for minimum probability of error (MPE) under equiprobable signaling. When symbols are equally probable, MPE and ML are equivalent.

In the next phase, we’ll take a brief detour into a radio-related exercise before returning to what happens at the bit level. While symbol errors are important, bit errors are the focus for most digital communication systems. How do bit errors occur, and how can we characterize them as opposed to symbol errors? We’ll explore these questions in upcoming lectures. Thank you.