Digital Communication using GNU Radio

Prof. Kumar Appaiah

Department of Electrical Engineering

Indian Institute of Technology Bombay

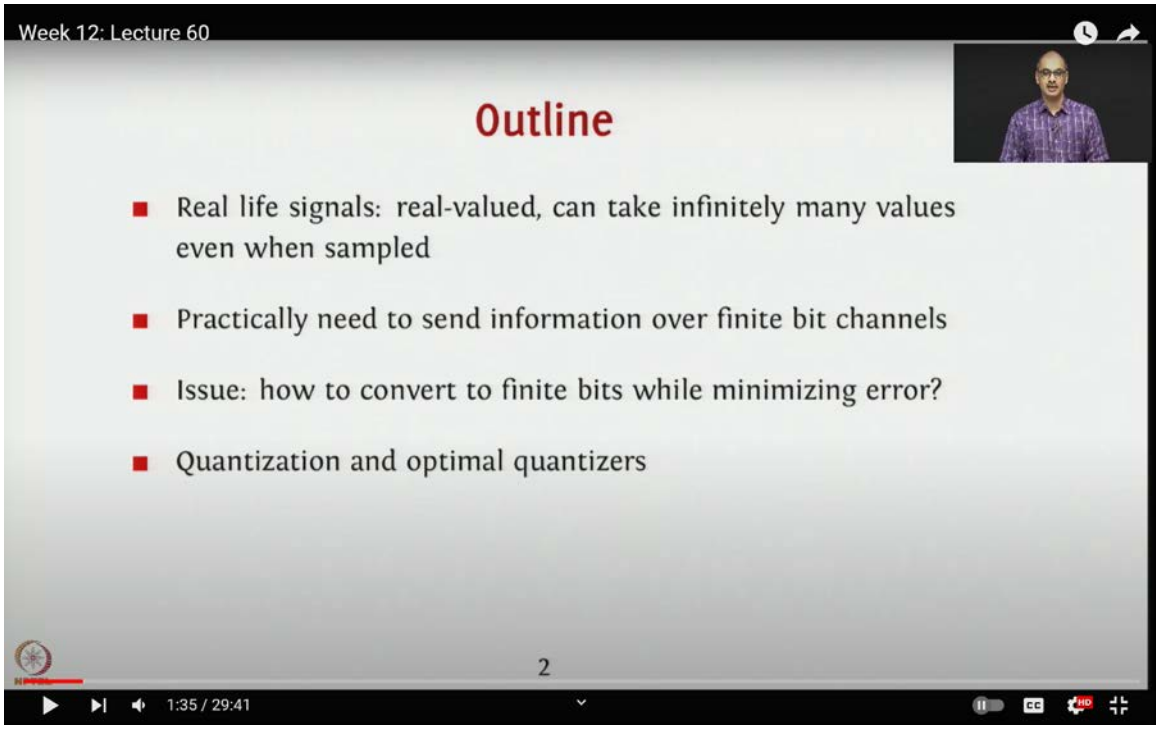
Week-12

Lecture-60

Quantisation

Hello, and welcome to this lecture on Digital Communication using GNU Radio. I'm Kumar Apaya from the Department of Electrical Engineering at IIT Bombay. Today, we will delve into the concept of quantization. Specifically, we will explore how real-world signals, which can take on an infinite number of values, are converted into a finite set of values that can be efficiently represented in bits. This conversion is crucial for enabling the digital communication techniques you have been learning about.

(Refer Slide Time: 01:35)

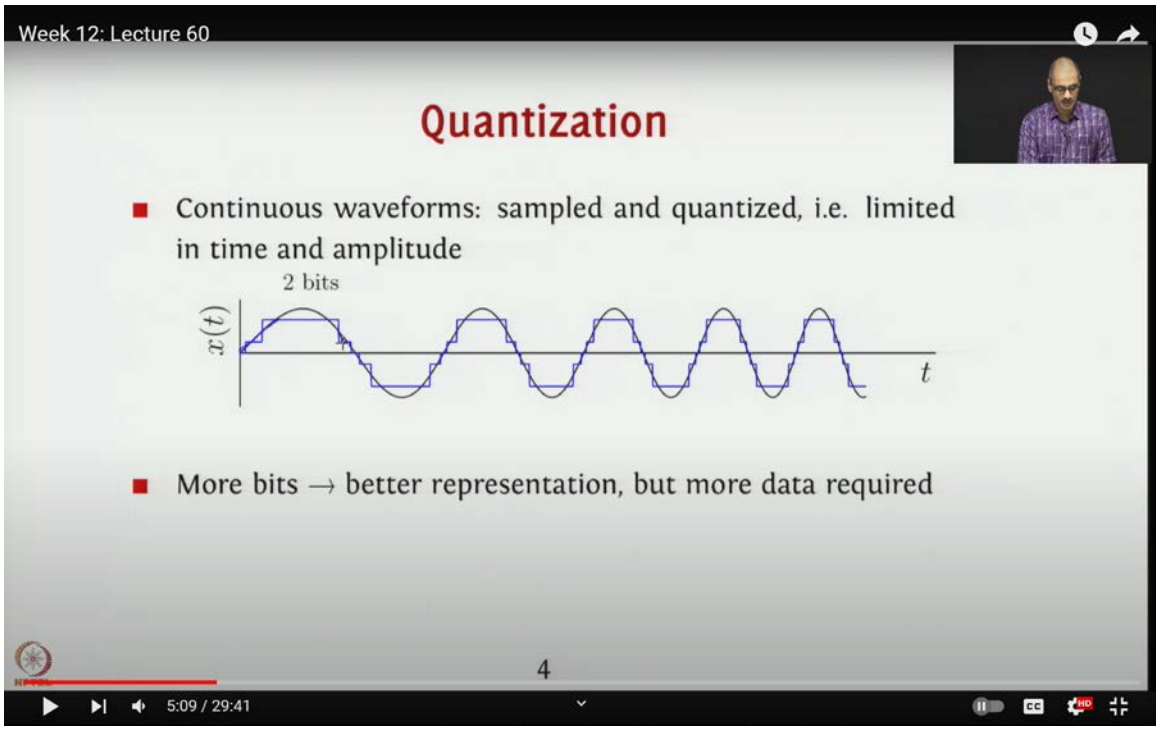


As we've discussed previously, real-life signals are indeed real-valued. For instance, when

you speak and measure the volume of your voice, it can be represented as a range of real numbers. Even if we confine this range to between -1 and 1, there are still infinitely many values within this range that your voice can assume.

However, perfect accuracy in representation is not always necessary. It's often sufficient to create a simpler representation that retains most of the essential information. For example, a modern mobile phone converts your voice into a more manageable set of values. Similarly, the lecture you are receiving, including both the video and my voice, is represented with a limited set of values. Despite this simplification, you can still understand the content being conveyed.

(Refer Slide Time: 05:09)



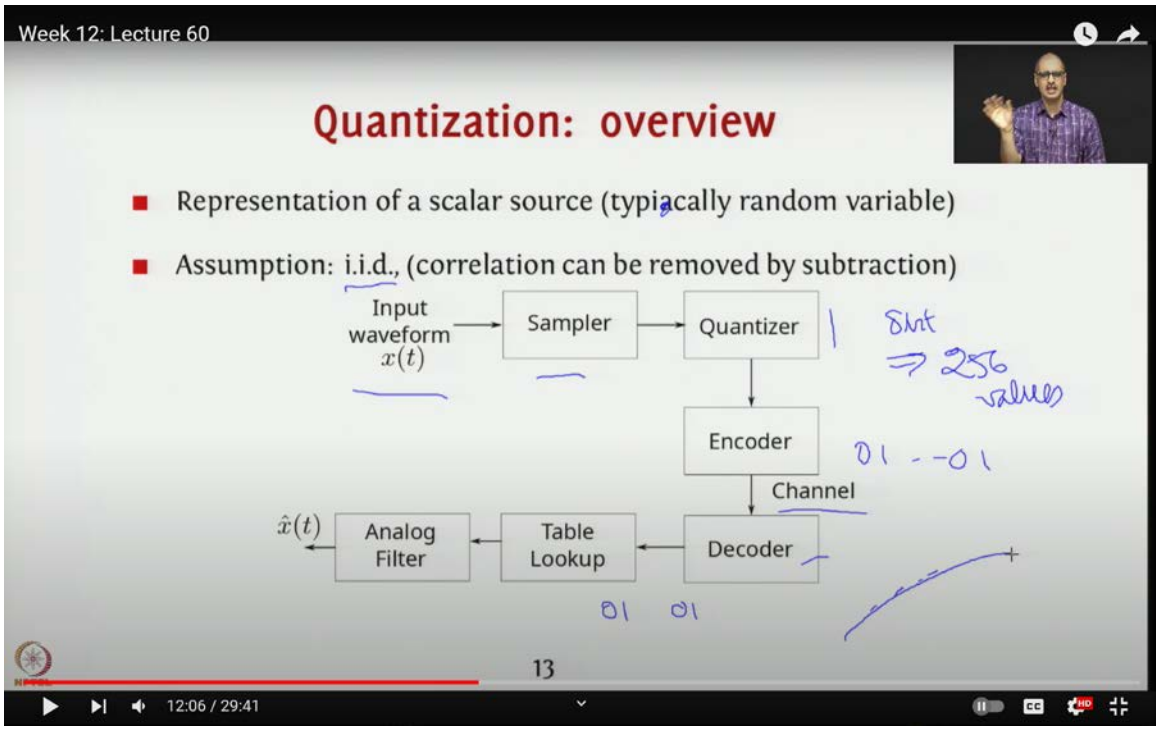
The practicality of this process arises from the need to transmit data over channels with finite bit rates. Channels, in reality, have limited capacities; even with allowances for errors, the amount of data that can be transmitted per second is finite. Thus, the challenge is to convert real-valued data into a finite number of bits while minimizing the error.

Minimizing error can take on various forms depending on the context. For instance, in the

context of speech, the goal might be to reduce errors to the point where the speech remains intelligible. In the case of images, the objective could be to capture only the most critical colors or details so that when the image is reproduced, it closely resembles the original, despite some information loss.

In the context of this lecture, our primary focus is on how to convert real-valued data into finite bits. This is the essence of quantization. Simply put, quantization involves restricting the set of values that a particular variable can assume. We will also explore optimal quantizers, which are designed to achieve the best possible performance given certain constraints.

(Refer Slide Time: 12:06)



Let's consider a straightforward example. Imagine you have a continuous waveform, a smooth, undulating signal. If I ask you to quantize this waveform, it means you need to limit the number of possible values it can take. For instance, let’s assume you have only one bit available to represent the signal at any given moment. In this case, you might choose to represent the values as either -0.5 or +0.5.

To illustrate, when the waveform's value is close to zero, you would represent it as zero because it is nearest to this value. In this example, we're not only using one bit but also incorporating zero as a level, so technically, it’s one bit plus zero. As the waveform value approaches +0.5, you would set it to +0.5. When the value crosses this limit and moves closer to zero, you revert to zero, and so on.

This process represents the waveform with just three levels of quantization. As you can see, this approach provides a relatively poor representation of the original waveform.

Now, let’s consider increasing the number of quantization levels. For instance, if you use two bits, you can represent five levels: 0, 1/4, 1/2, 3/4, and 1. This improved quantization provides a better approximation of the waveform. You can more accurately capture the sections where the signal rises and falls, although some distortion still occurs at the peaks.

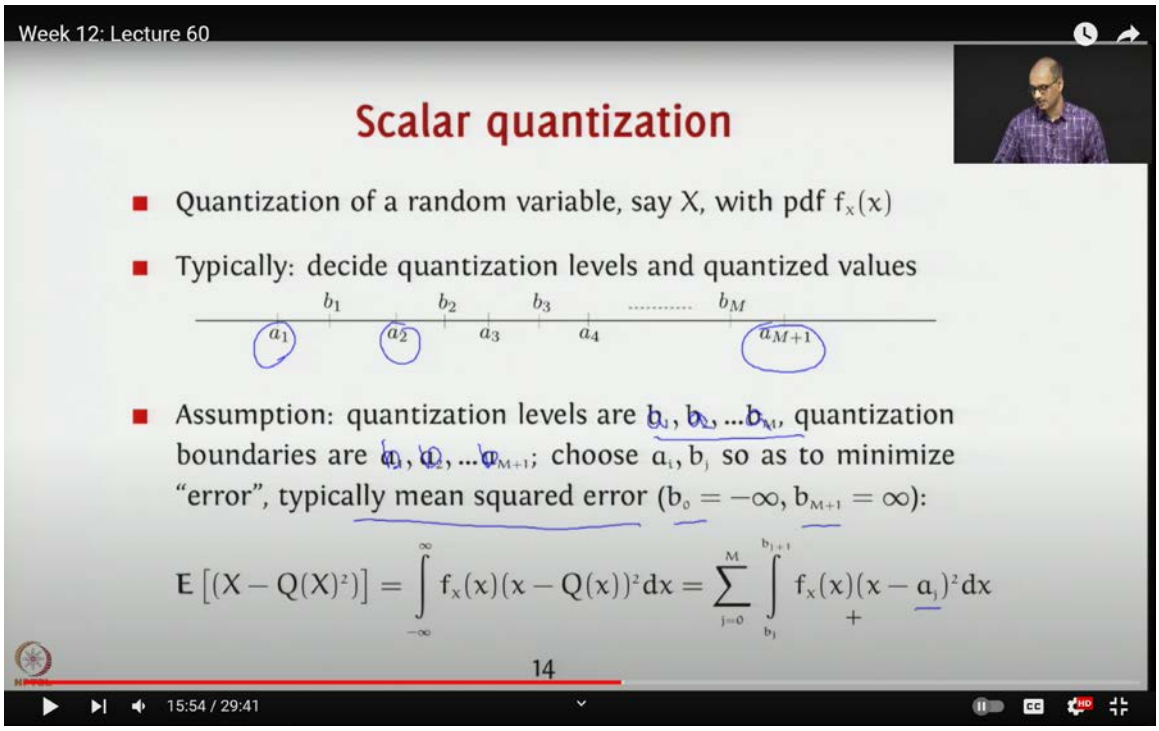
By further increasing the number of bits, say, to three, four, or five bits, the accuracy of the representation improves significantly. With more bits, you get more levels and a more precise approximation of the waveform. For example, at eight bits, you have 256 levels, and at ten bits, you have 1024 levels. As the number of levels increases, the waveform representation becomes much more accurate. If you closely inspect an eight-bit quantization, you will see that the waveform is represented with considerable fidelity, capturing its features with minimal jaggedness.

In this context, the more data you are willing to allocate, the better your representation will be. While there are various techniques to scale the data to fit this representation more effectively, a general principle is that higher bit depths lead to better quality. For instance, CD-quality audio, as defined by industry standards, typically uses 16-bit quantization. In contrast, less ambitious quantization, such as that used in some mobile phones, might use 8-bit or 12-bit quantization. Generally, for audio quantization, 8 bits or 12 bits might be sufficient for speech, while music often requires 16 bits, and in stereo, this translates to 16 bits per channel.

These standards provide a rough guide to understanding quantization decisions. When you inspect the specifications for these audio formats, you can better grasp the rationale behind

the chosen bit depths.

(Refer Slide Time: 15:54)



Now, let's discuss how a quantizer is conceptualized. We aim to represent a scalar source, often modeled as a random variable, essentially, my speech is considered a random variable. While modeling speech as a random variable might seem complex, it's based on the idea that what I will say is not predetermined. Thus, it's modeled as a random variable. A more contentious assumption is that these random variables are independent and identically distributed (IID).

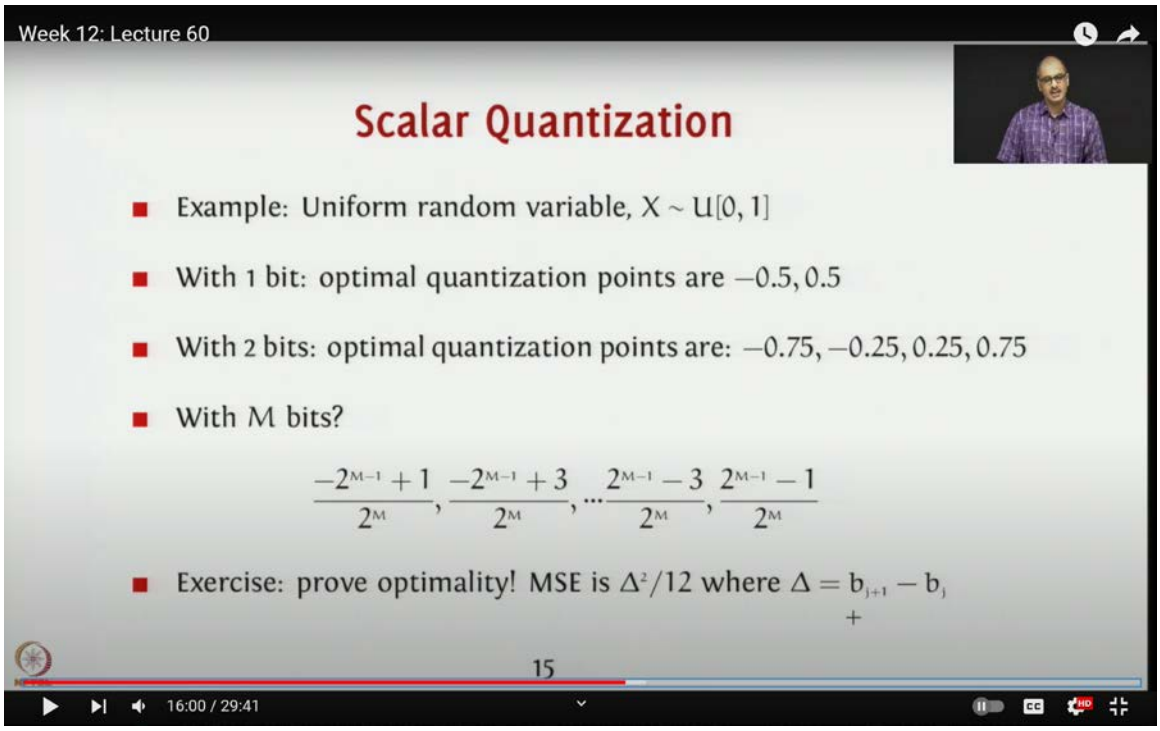
In reality, this assumption can be problematic. For instance, the sounds I produce from one millisecond to the next are highly correlated. The speech signal exhibits significant similarity over short intervals. However, to address this, techniques are used to remove the correlation. Essentially, you model the correlations, subtract them, and then reintroduce them at the receiver or reconstruction stage. This approach helps in approximating the IID assumption.

In practice, speech quantization and compression, as well as image quantization, frequently

utilize these techniques. The idea is to process the source signal by first removing correlations, sampling the data, and then reintroducing correlations during reconstruction.

The rate at which you sample depends on the type of signal being sampled. For instance, if you're sampling a signal with a frequency in the tens of hertz, such as machine vibrations, you can sample at hundreds of hertz.

(Refer Slide Time: 16:00)



If you are sampling a signal like speech, the sampling rate depends on the quality you desire. For standard quality, you might sample at around 8,000 samples per second (8 kHz), whereas for CD-quality audio, you need to sample at 40,000 or 44,000 samples per second (40 kHz or 44 kHz). This higher sampling rate ensures that you get a sufficiently accurate representation of the signal over time. Remember, quantization needs to address both time and amplitude. The sampling rate handles the time aspect, which is a key component of digital signal processing.

Now, let's discuss amplitude quantization. Here, we restrict the set of values that the vertical axis can take to a finite set of values, similar to what you saw in the previous

waveform example. If the signal's amplitude falls between these predefined values, it is mapped to the nearest available value. For instance, with an 8-bit quantizer, you have 256 possible amplitude values. The encoder then converts these values into a binary format, zeros and ones.

Although we're not covering modulation and other aspects in detail here, it's important to note that the quantized data is transmitted over a channel. After transmission, the data undergoes equalization and error correction processes, which are handled by the decoder. The decoder receives the binary data (e.g., 0, 1, 0, 0, 1) and performs a table lookup to map these binary sequences back to their corresponding amplitude values.

Finally, an analog filter is used to convert these sampled, quantized values back into a continuous waveform. For instance, if you have samples like 0, 1, 2, etc., the filter smooths out these quantized values and reconstructs a continuous time waveform, such as the original speech.

In practice, when I speak, my voice is captured by a microphone, sampled, and then quantized by the system. The quantized data is transmitted, for example, over the internet or stored on a disk. Upon reception, this quantized data is converted back to its amplitude values using the reverse process, and the analog filter reconstructs the continuous waveform from these discrete samples.

These quantized values are then converted into a waveform. In digital signal processing (DSP), this involves interpolating these values using an analog filter, which results in the speech you hear. While it is guaranteed that the speech you are listening to is not identical to the speech I am currently speaking, it is extremely close, so close that you can understand what I'm saying without missing nearly any words. The quantization is sufficiently accurate to ensure that every word I speak is comprehensible.

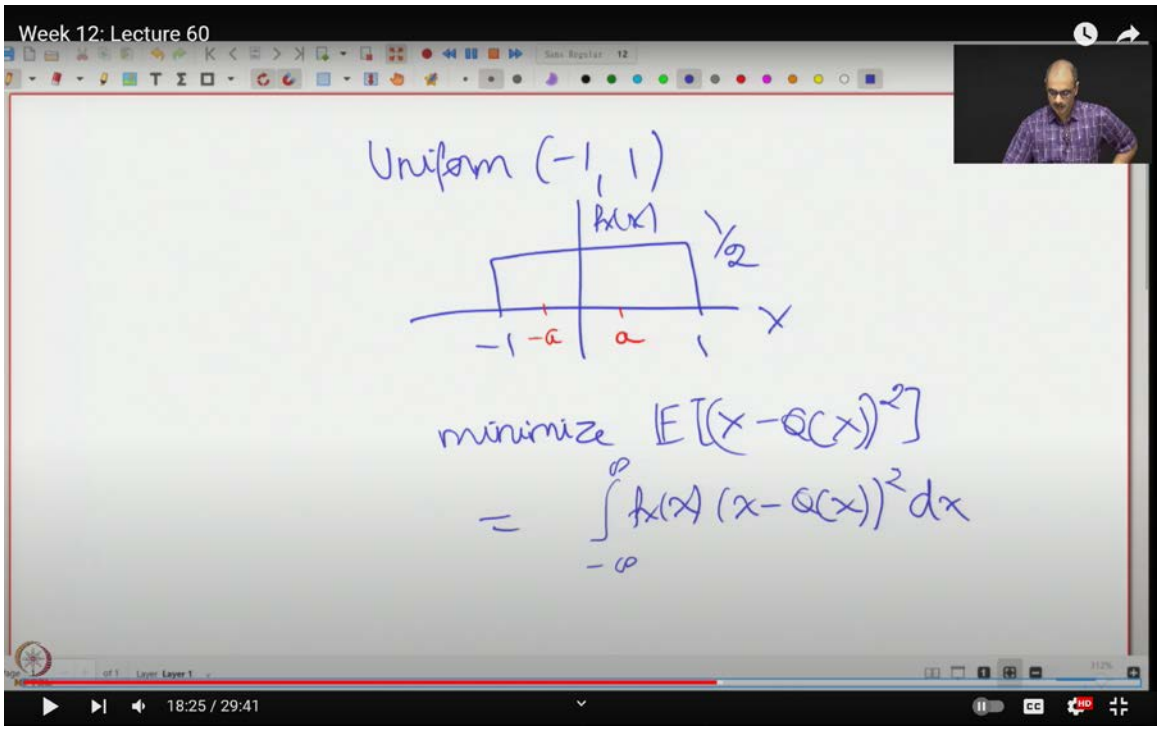
Now, let's delve into the specifics of scale and quantization. Suppose we want to quantize a random variable X with a probability density function (PDF) $f_X(x)$. Typically, the process involves determining quantization levels and corresponding quantized values.

Quantization values are denoted as $A_1, A_2, \ldots, A_{m+1}$, where these are the discrete values to

which the continuous signal will be mapped. Quantization levels are represented by decision regions B_1, B_2, B_3, …, B_m.

For example, if your signal falls to the left of B_1 (say, $B_1 = 0.5$), and the value you obtain is 0.4, which is less than 0.5, you would quantize this value to A_1. Conversely, if the value is 0.6, which is greater than 0.5, you would quantize it to A_2.

(Refer Slide Time: 18:25)



In this setup, B_1, B_2, B_3 are decision boundaries, and A_1, A_2, A_3, A_4 are the quantized levels. Note that the quantized levels do not have to be uniform, and the decision boundaries B_1, B_2, …, B_m need not be equally spaced. For instance, in voice quantization, certain values are more probable than others, so voice quantizers often have non-uniform gaps between quantization levels.

To minimize the quantization error, which is typically measured as the squared error, you need to carefully select both the quantization levels A_i and the boundaries B_j.

Minimizing squared error is a sensible approach because it relates to minimizing the power

or energy difference between the original signal and the quantized signal. Essentially, you aim to minimize the mean squared error between these two signals. Conventionally, the quantization boundaries are set at B_0 and B_{m+1} as $-\infty$ and $+\infty$ respectively. This ensures that the quantization boundaries B_1 through B_m are finite and properly enclose all possible values.

The objective is to minimize the expectation of the squared difference between the original signal X and its quantized version Q(X). The function Q(X) maps X to one of the quantization levels A_1, A_2, …, A_m, depending on the boundaries B_1, B_2, …, B_m. For example, if you choose $A_1 = 0$ and $B_1 = 0.5$, then Q(0.4) would be A_1, which is 0, while Q(0.6) would be A_2.

To minimize the squared error, you need to evaluate the integral:

$$\int_{-\infty}^{\infty} f_X(x) \cdot \big(x - Q(x)\big)^2 \, dx$$

You can break this integral into segments defined by the quantization boundaries:

$$\sum_{j=0}^{m} \int_{B_j}^{B_{j+1}} f_X(x) \cdot \big(x - A_j\big)^2 \, dx$$

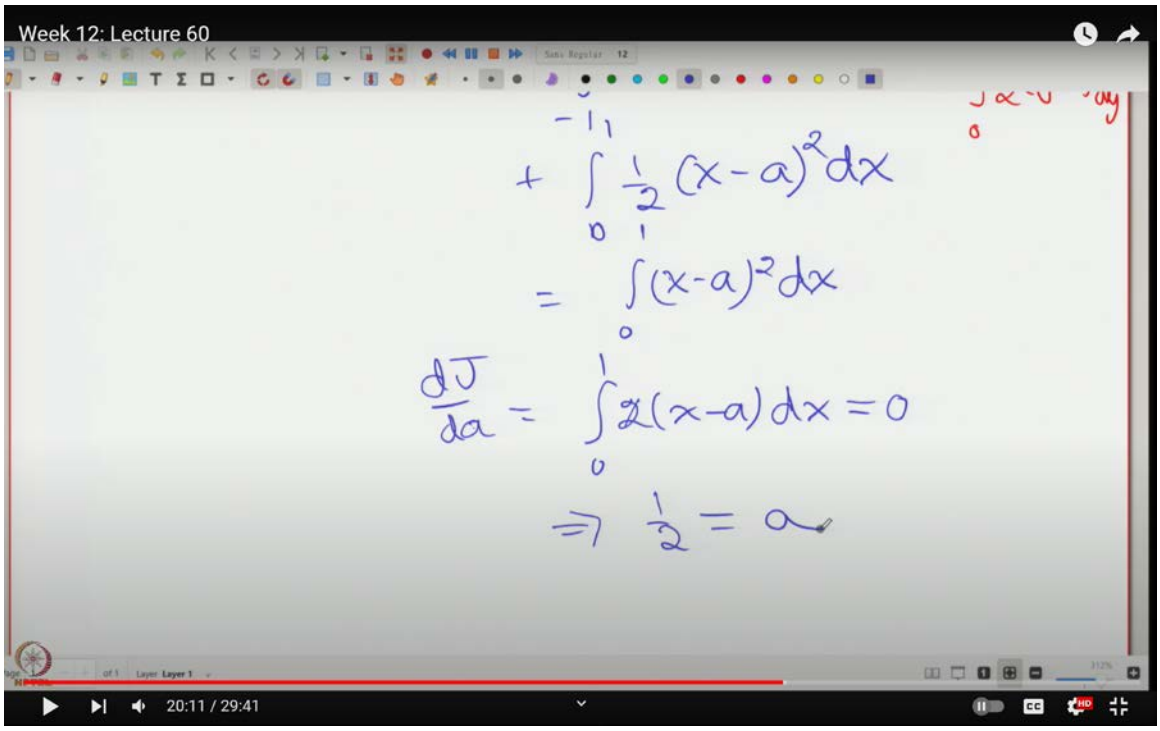
This involves minimizing the error within each boundary segment by adjusting the quantization levels A_j and boundaries B_j. This is the optimal quantization problem for the scalar case.

Let's consider a practical example with a uniform random variable, say from -1 to 1. The PDF, $f_X(x)$, is uniform in this range. To minimize the squared error with a one-bit quantizer (two levels), you need to choose the levels wisely.

For instance, if you choose levels only on the positive side or only on the negative side, you will incur significant error for values falling in the opposite range. Similarly, if the levels are not symmetrically placed, it will also lead to issues. Given the symmetry of the problem, choosing levels at -a and a is optimal. This is because the PDF is symmetric

around zero, and the random variable is equally likely to be positive or negative. Thus, the optimal strategy is to set the quantization levels at -a and a.

(Refer Slide Time: 20:11)



Verifying this mathematically is an exercise left for you, but intuitively, this approach minimizes the error effectively.

What I want to emphasize is that for a symmetric distribution, such as a uniform distribution over [-1, 1] or a Gaussian distribution, the optimal quantization levels must also be symmetric around the y-axis. Our goal now is to find the optimal quantization level, denoted as a, to minimize the expectation of $(X - Q(X))^2$.

To do this, we need to compute the integral:

$$\int_{-\infty}^{\infty} f_X(x) \cdot \big(x - Q(x)\big)^2 \, dx$$

For our uniform distribution between -1 and 1, this integral simplifies to:

$$\int_{-1}^{0} f_X(x) \cdot (x - A)^2 \, dx + \int_{0}^{1} f_X(x) \cdot (x - A)^2 \, dx$$

Given $f_X(x)$ is uniform, it is $\frac{1}{2}$ over this range. Therefore:

$$\int_{-1}^{0} \frac{1}{2} \cdot (x - A)^2 \, dx + \int_{0}^{1} \frac{1}{2} \cdot (x - A)^2 \, dx$$

To simplify, let's substitute y = -x in the integral from -1 to 0. This transforms it into:

$$\int_{0}^{1} \frac{1}{2} \cdot (y - A)^2 \, dy$$

Combining both integrals:

$$\int_{0}^{1} \frac{1}{2} \cdot (x - A)^2 \, dx + \int_{0}^{1} \frac{1}{2} \cdot (x - A)^2 \, dy = \int_{0}^{1} (x - A)^2 \, dx$$

To find the optimal a, we need to minimize this expression. Let J(a) denote the integral:

$$J(a) = \int_{0}^{1} (x - a)^2 \, dx$$

Differentiating J(a) with respect to a:

$$\frac{dJ}{da} = \int_{0}^{1} 2(x - a) \, dx$$

Solving this, we find:

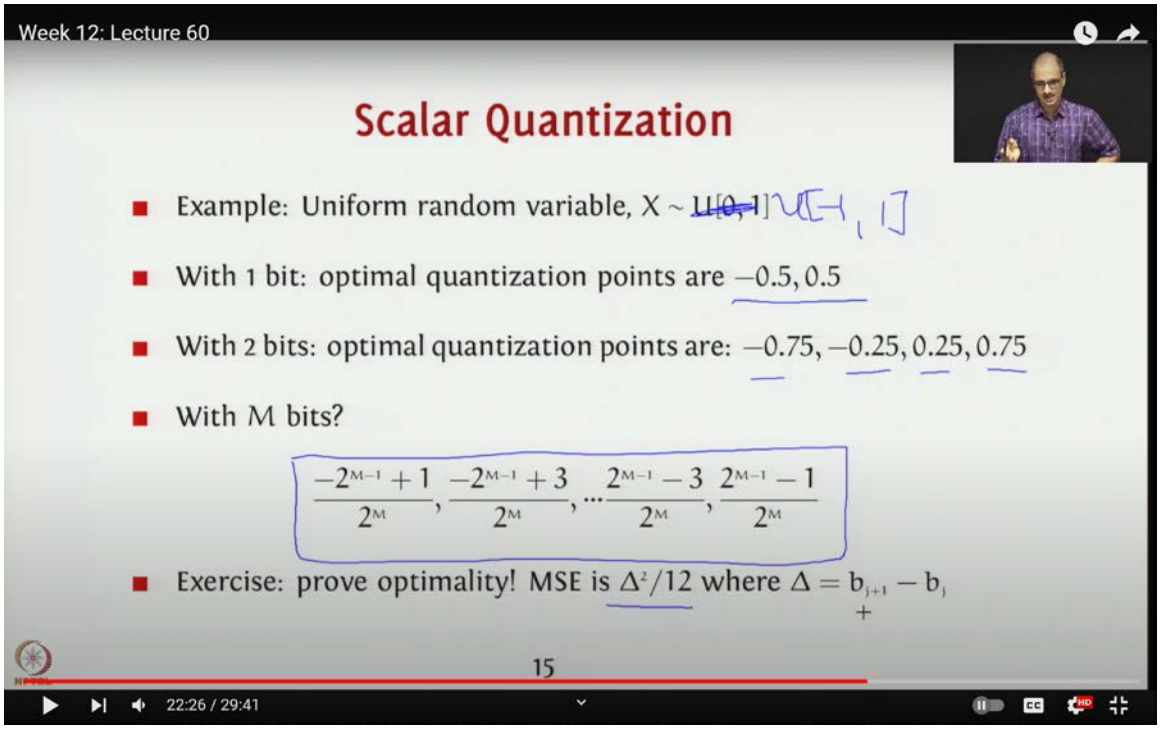
$$\frac{dJ}{da} = 2 \int_{0}^{1} (x - a) \, dx = 2 \left[\frac{x^2}{2} - ax \right]_0^1$$

$$= 2 \left(\frac{1}{2} - a \right)$$

Setting $\frac{dJ}{da} = 0$:

$$\frac{1}{2} - a = 0 \Rightarrow a = \frac{1}{2}$$

(Refer Slide Time: 22:26)



Thus, the optimal quantization levels are $-\frac{1}{2}$ and $\frac{1}{2}$. This result can be verified in GNU Radio, where choosing different levels would lead to higher errors.

For a two-bit quantization, the optimal levels are $-\frac{3}{4}, -\frac{1}{4}, \frac{1}{4}$, and $\frac{3}{4}$. For m bits, the levels are:

$$\text{Levels} = -\frac{2^{m-1}-1}{2^{m-1}}, \ldots, \frac{2^{m-1}-1}{2^{m-1}}$$

You can verify this pattern by considering the symmetry of the distribution and applying the principle of optimality. The mean squared error for this quantization is $\frac{\delta^2}{12}$, where δ is the difference between adjacent quantization boundaries or levels.

Let's review the results of our previous calculations. Evaluating the integral for the squared

error between x and $\frac{1}{2}$, we find:

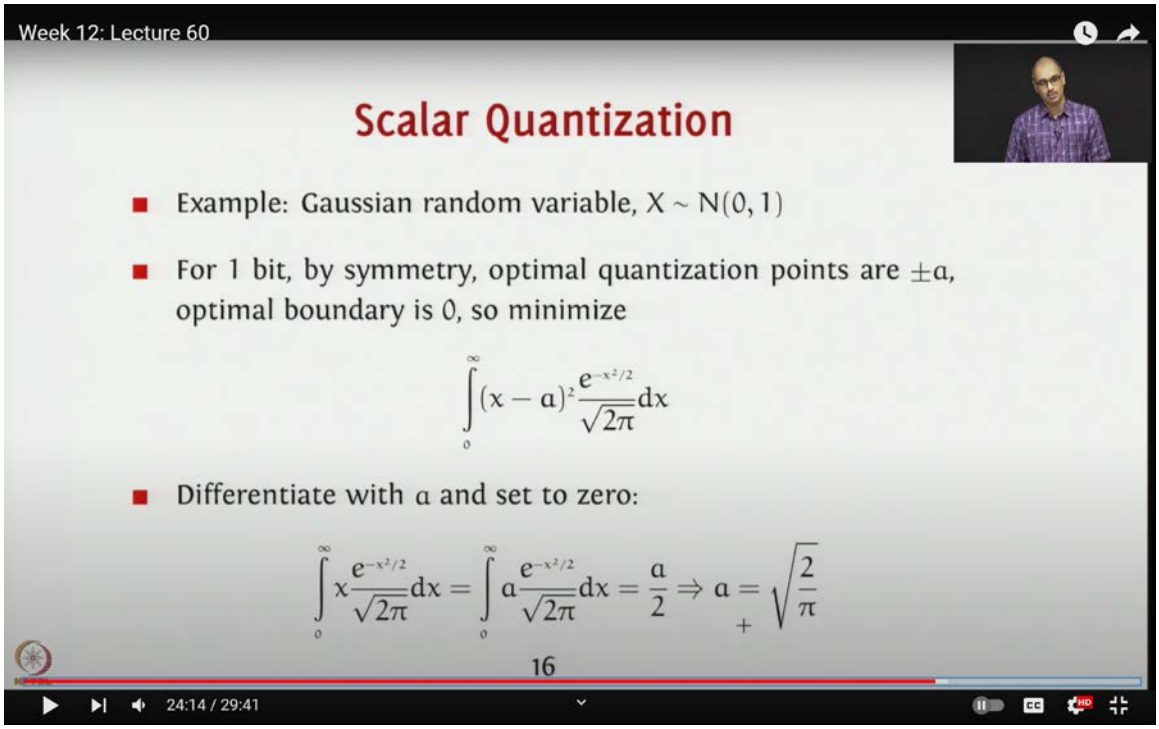
$$\int_0^1 \left(x - \frac{1}{2}\right)^2 dx$$

Substituting the limits into this integral, when x = 1, we get $\left(\frac{1}{2}\right)^3$, and when x = 0, we get $-\left(\frac{1}{2}\right)^3$ divided by 3. Adding these, we obtain:

$$\frac{1}{8} + \frac{1}{8} = \frac{1}{12}$$

This value, $\frac{1}{12}$, represents the minimum mean squared error for scalar quantization of a uniform [-1, 1] random variable. The term δ, which is the difference between adjacent quantization boundaries or levels, contributes to this error calculation.

(Refer Slide Time: 24:14)



Now, let's consider a Gaussian random variable, specifically $X \sim N(0,1)$. While the

derivation can be generalized, we'll focus on the standard normal distribution for simplicity. For a symmetric distribution, the optimal quantization points for one-bit quantization are ± a, and the optimal boundary is zero, as discussed previously.

To minimize the squared error:

$$\int_0^\infty (x-a)^2 e^{-\frac{x^2}{2}} \frac{1}{\sqrt{2\pi}}\, dx$$

we can simplify by removing the $\frac{1}{\sqrt{2\pi}}$ factor. Differentiating the integral with respect to a, we get:

$$\frac{\partial}{\partial a}\left[\int_0^\infty (x-a)^2 e^{-\frac{x^2}{2}} \frac{1}{\sqrt{2\pi}}\, dx\right] = 2\int_0^\infty (x-a) e^{-\frac{x^2}{2}} \frac{1}{\sqrt{2\pi}}\, dx$$

Setting this derivative to zero:

$$\int_0^\infty x\, e^{-\frac{x^2}{2}} \frac{1}{\sqrt{2\pi}}\, dx = \int_0^\infty a\, e^{-\frac{x^2}{2}} \frac{1}{\sqrt{2\pi}}\, dx$$

Given that the integral of $xe^{-\frac{x^2}{2}}$ over $[0, \infty)$ is $\frac{1}{2}$, we find:

$$a = \sqrt{\frac{2}{\pi}}$$

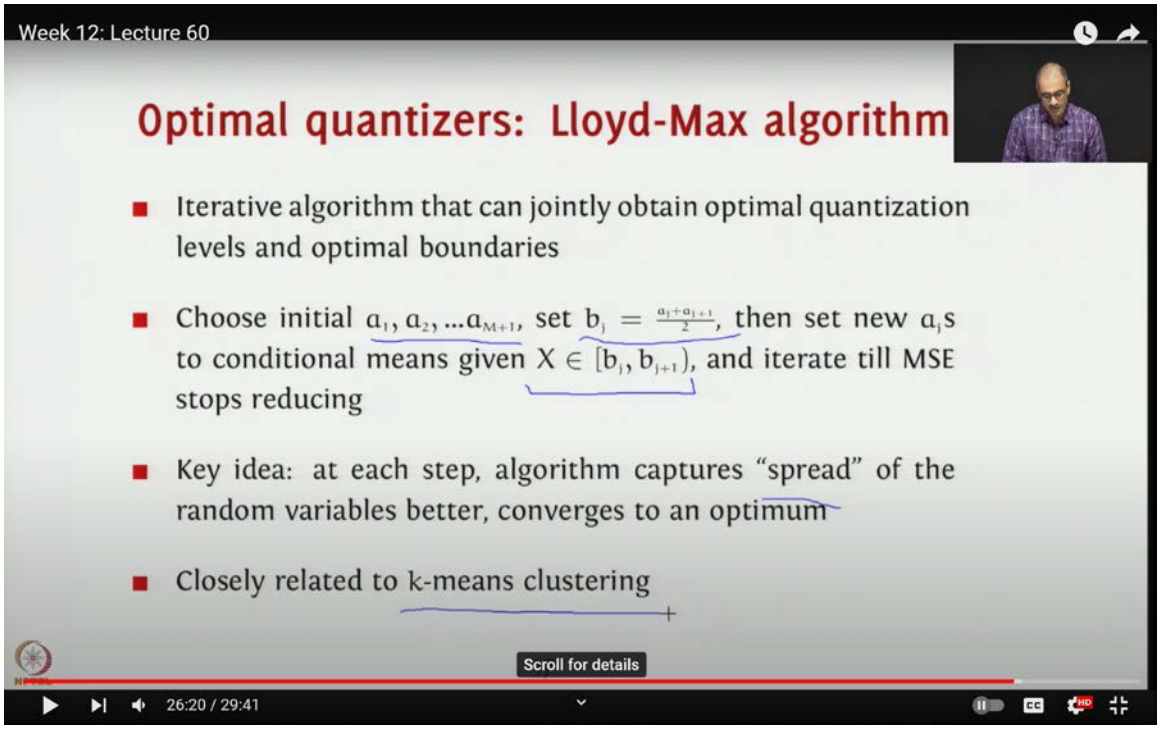
This result represents the optimal quantization level for a Gaussian distribution. In practice, using GNU Radio, we can set this optimal value and verify that it indeed minimizes the error.

For more complex optimal quantization, we use the Lloyd-Max algorithm, which provides a systematic approach to finding optimal quantization levels.

Let's briefly explore the Lloyd-Max algorithm, an iterative method used to determine both the optimal quantization levels and boundaries. The process starts with an initial set of quantization levels, say a_1, a_2, …, a_{m+1}, and initial boundaries b_j, which are typically set as

the midpoints between consecutive quantization levels, $b_j = \frac{a_j + a_{j+1}}{2}$.

(Refer Slide Time: 26:20)



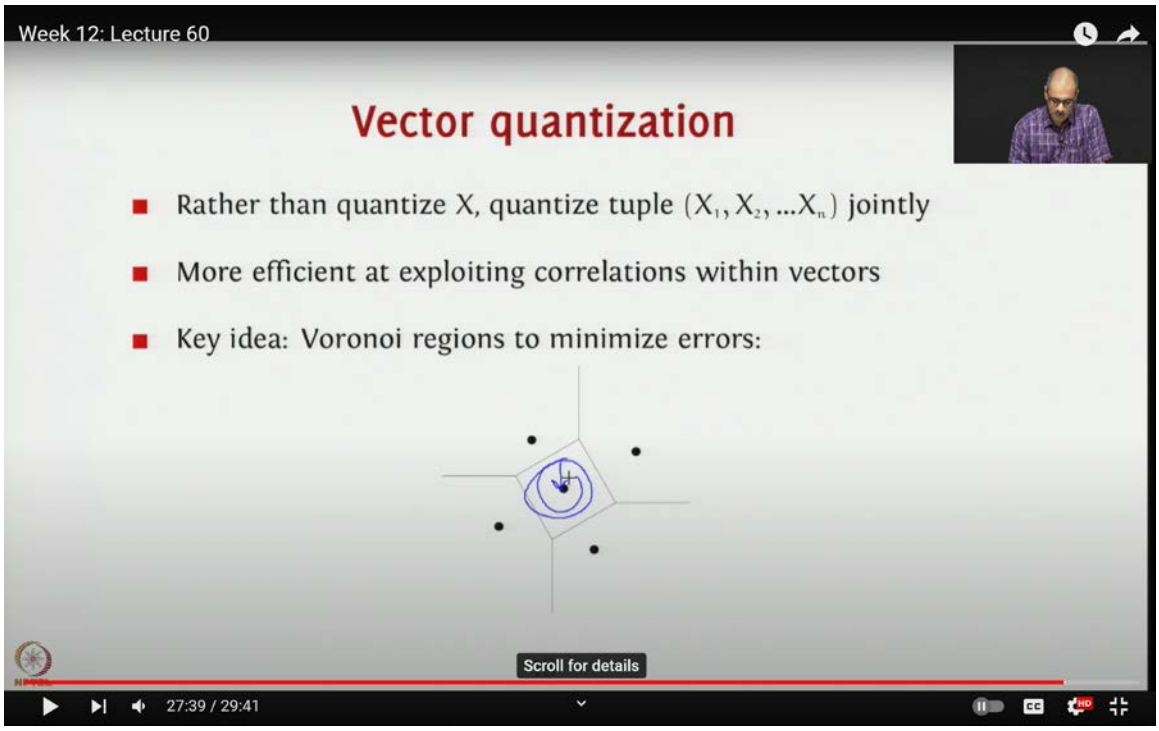
The next step is to update the quantization levels a_j. These new levels are set as the conditional means of the input values x that fall within the respective quantization intervals $[b_j, b_{j+1}]$. The reason we use the conditional mean is rooted in probability theory: the conditional mean minimizes the mean squared error. By setting a_j to this conditional mean, we ensure that each quantization level is optimally placed to reduce the error.

Once new a_j values are computed, we recalculate the boundaries b_j as the midpoints of the updated quantization levels. This process of updating a_j and b_j is repeated iteratively. Over time, the changes in a_j and b_j become minimal, and the mean squared error stabilizes, indicating that an optimal quantizer has been found.

The Lloyd-Max algorithm, developed over 30 to 40 years ago, has proven to be effective in various scenarios by progressively minimizing mean squared error. It captures the distribution of the random variable by repeatedly adjusting the quantization levels to

achieve the best possible MSE.

(Refer Slide Time: 27:39)



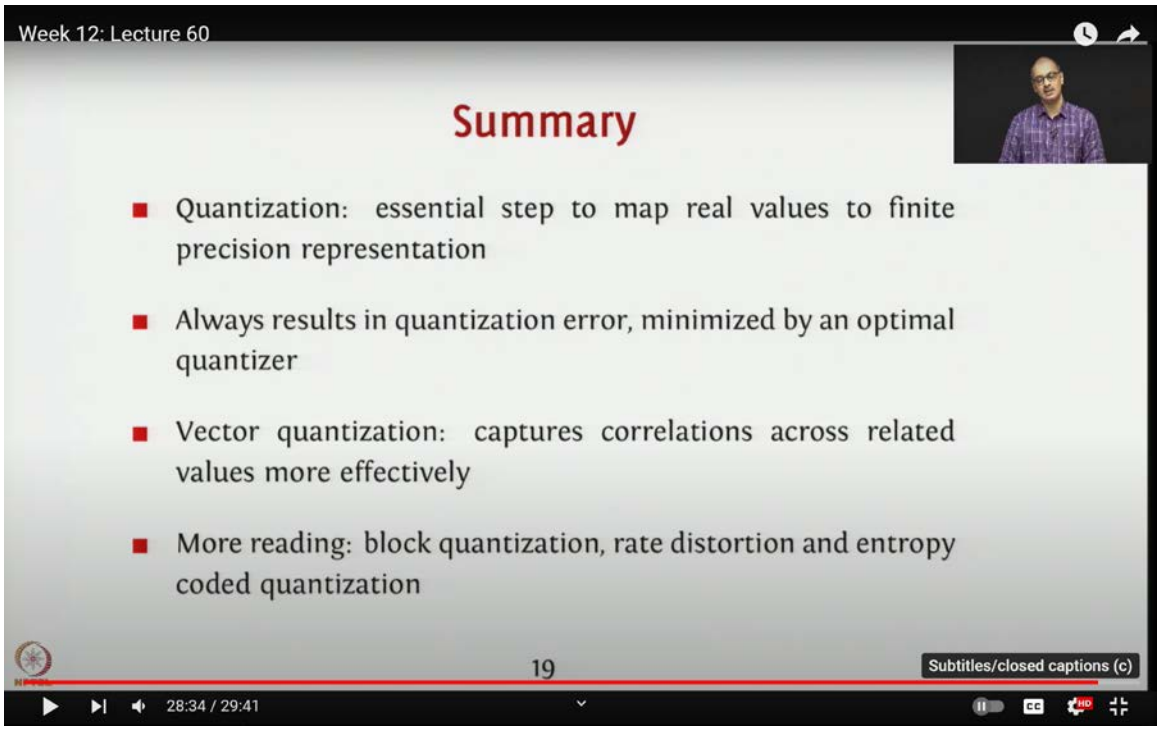
Interestingly, the Lloyd-Max algorithm is closely related to k-means clustering, a popular technique in pattern recognition and machine learning. In k-means clustering, the algorithm also involves finding centroids that minimize the distance to points in clusters, which parallels the Lloyd-Max method's use of conditional means.

Additionally, instead of quantizing individual scalars, you can extend the approach to quantize tuples or groups of random variables, x_1, x_2, …, x_n, jointly. This method takes advantage of correlations among the variables. For instance, in image processing, after transforming an image into vectors, quantizing these vectors as a whole, rather than individually, preserves the inherent correlations and leads to more efficient quantization. This approach results in what are known as Voronoi regions: areas within which all points are mapped to a single quantization point.

Voronoi regions play a crucial role in determining where to quantize, ensuring that at the optimal point, errors such as squared error are minimized. Vector quantization is widely

utilized in scenarios involving large datasets with inherent correlations. This method effectively captures and leverages those correlations to improve quantization accuracy.

(Refer Slide Time: 28:34)



In summary, quantization is a fundamental process for converting continuous values into finite precision representations. However, it's important to remember that quantization error is an inherent aspect of practical signals. An optimal quantizer is designed to minimize this quantization error. While scalar quantization is suitable for many applications, vector quantization is preferred when dealing with groups of correlated data. This approach allows for quantization in higher-dimensional spaces, enhancing overall efficiency and accuracy.

For those interested in exploring this topic further, there are advanced areas such as block quantization, rate distortion theory, and entropy-coded quantization. These topics are extensively covered in sources related to compression, information theory, and related fields. In our next lecture, we will simulate these operations using GNU Radio and analyze how quantization errors appear in histograms. Thank you.