Digital Communication using GNU Radio

Prof. Kumar Appaiah

Department of Electrical Engineering

Indian Institute of Technology Bombay

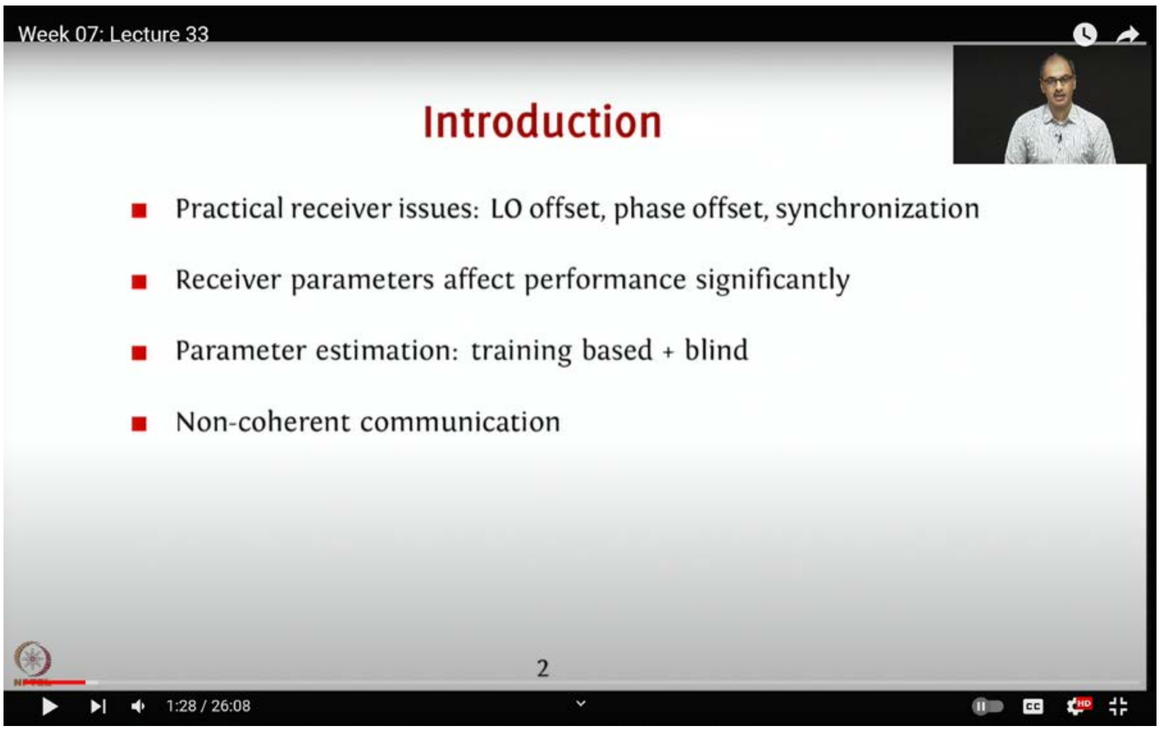
Week-07

Lecture-33

Parameter Estimation for Practical Receivers - Part 1

Welcome to this lecture on Digital Communication using GNU Radio. My name is Kumar Appaiah, and I am part of the Department of Electrical Engineering at IIT Bombay. In this lecture, we will focus on Synchronization and Non-Coherent Communication.

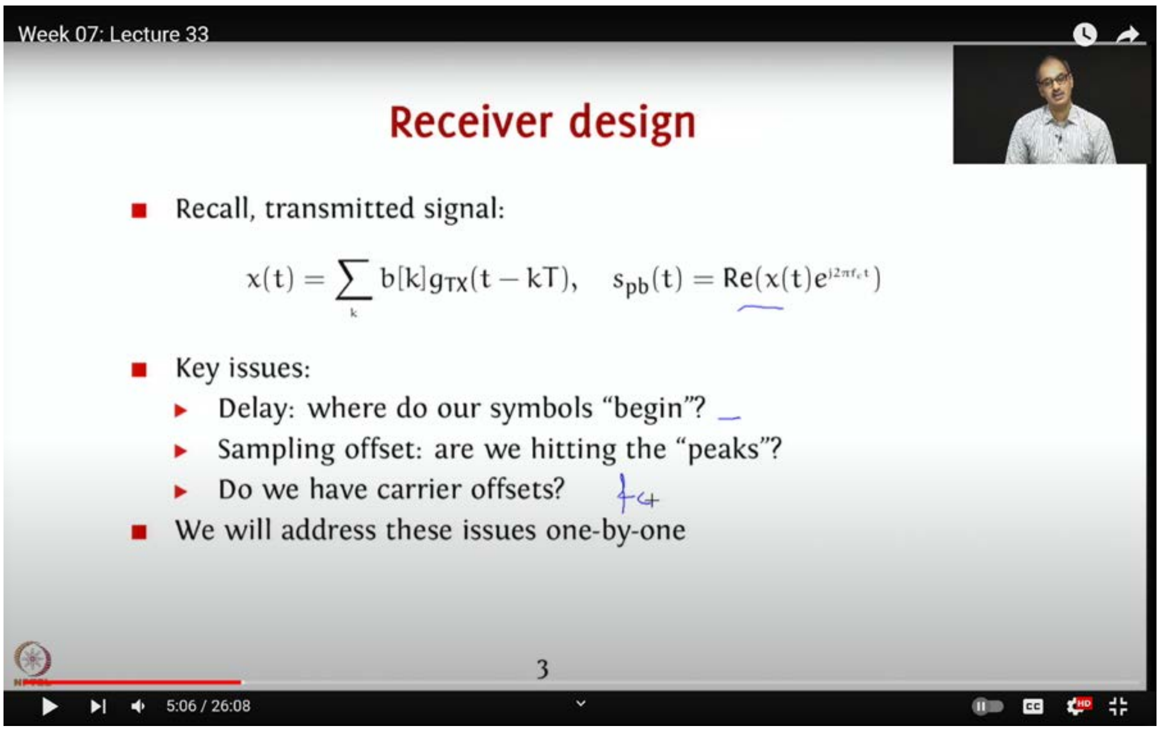
(Refer Slide Time: 01:28)



Throughout this lecture and the upcoming series, we will be addressing a critical assumption about communication systems. Previously, we examined the impact of noise on symbol detection. Now, we will delve into practical receiver challenges. Specifically, we will explore scenarios where the correct phase of the carrier or local oscillator is not available at the receiver, or where there is a frequency offset. What strategies can we employ in such situations? Additionally, we will need to estimate various parameters of the communication system.

In this lecture, we will cover these issues and complement the discussion with some hands-on experiments in GNU Radio.

(Refer Slide Time: 05:06)

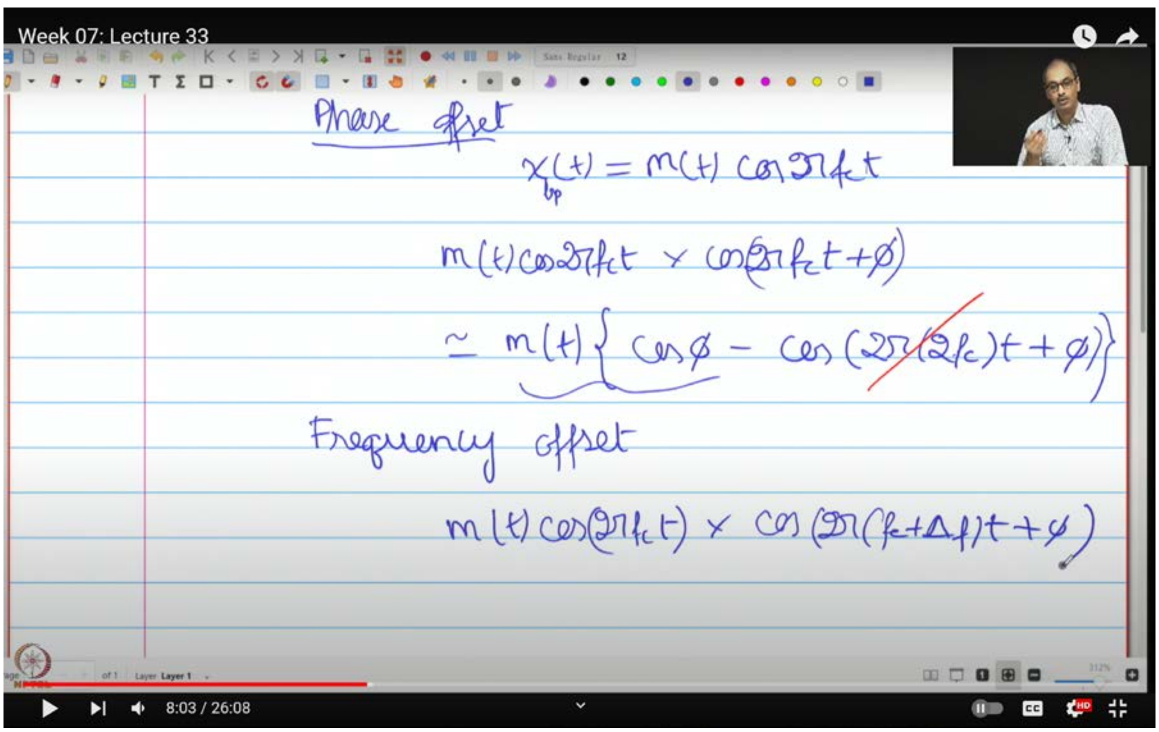


Several practical challenges at the receiver include local oscillator offset, where the carrier frequency at the receiver may differ from the expected frequency, phase offsets, and synchronization. Synchronization involves determining the exact start of the symbol so that we can accurately capture those symbols. We also need to ascertain the correct sampling location and the precise timing for sampling after matched filtering. These are crucial aspects that need to be addressed at the receiver end.

Receiver parameters have a substantial impact on performance, in addition to noise. If you do not sample at the correct location or have the precise carrier frequency and phase, you risk encountering significant errors because the receiver is not accurately capturing the signal. Therefore, one critical aspect we will explore is parameter estimation. This involves determining how to estimate the amplitude of the signal at the receiver, as well as the delay, phase offset, and frequency offset. These challenges fall under parameter estimation because there are multiple unknowns that need to be resolved amidst noise. Our goal is to devise a statistically sound method for obtaining the best estimates in the presence of noise.

Additionally, we will investigate non-coherent communication, where we do not need to know the exact phase or frequency of the local oscillator. Instead, we will explore whether communication can proceed effectively if the phase and frequency are only approximately correct.

(Refer Slide Time: 08:03)

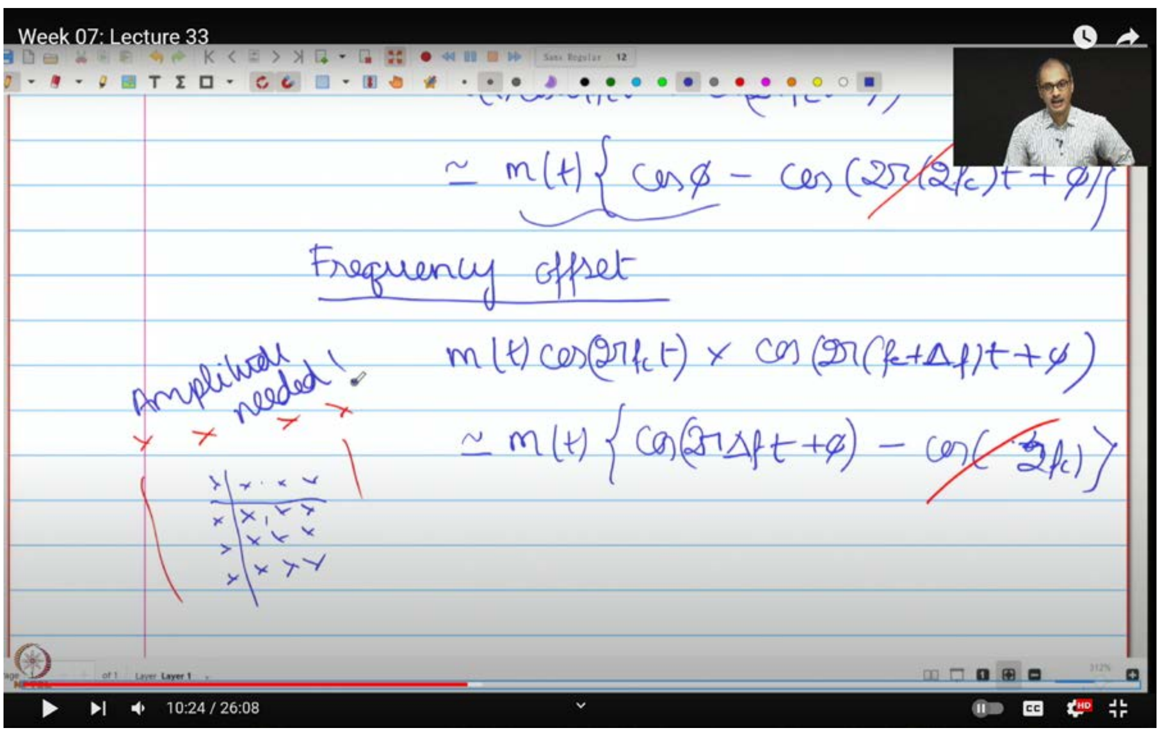


For receiver design, we will simplify some assumptions to convey the core concepts. Suppose the transmitted signal is represented as $\sum b_k g_{tx}(t-kT)$, where b_1, b_2, b_3, … are symbols from your constellation, such as QPSK or BPSK. Here, $g_{tx}(t-kT)$ represents the pulse shaping function, which could be a sinc function, a root-raised cosine, or even a rectangular pulse, chosen based on constraints like power and bandwidth.

The bandpass version of this signal is given by the real part of $x(t)e^{j2\pi f_c t}$. Essentially, you upconvert the signal by multiplying x(t) with $e^{j2\pi f_c t}$. This results in the waveform $x(t) \times e^{j2\pi f_c t}$, where x(t) is split into its in-phase (I) and quadrature (Q) components: $x_c(t) \times cos(2\pi f_c t)$ and $x_s(t) \times sin(2\pi f_c t)$, respectively. This waveform is centered around f_c with a bandwidth of $\pm \frac{W}{2}$, assuming the baseband bandwidth of x(t) is W.

Key issues of concern include delay, such as determining the exact start of the symbols and the appropriate sampling point. Accurate peak detection is crucial; otherwise, you will encounter a high rate of symbol errors.

(Refer Slide Time: 10:24)



Sampling offset can lead to symbol errors if you don't sample precisely at the correct point. However, an equally important issue is the carrier offset. Specifically, are you receiving the carrier frequency f_c accurately at the receiver? Are you capturing the phase correctly? If either the phase or the frequency is incorrect, you will encounter significant issues. Let's perform a brief exercise to illustrate this.

Assume you have a symbol represented as M(t). We'll refer to this as phase offset. For simplicity, let's consider the signal x(t) as a real signal, $x_{bp}(t) \cdot cos(2\pi f_c t)$. At the receiver, the signal might be $cos(2\pi f_c t + \theta)$, where we're ignoring the quadrature (Q) component and focusing on the real part for simplicity. The concept remains similar.

Now, consider the product of $M(t) \cdot cos(2\pi f_c t)$ with $cos(2\pi f_c t + \phi)$. Using the trigonometric identity $cosa \cdot cosb = \frac{1}{2}[cos(a-b) + cos(a+b)]$, we get:

$$M(t) \cdot cos(2\pi f_c t) \cdot cos(2\pi f_c t + \phi) = \frac{1}{2} M(t)[cos(\phi) + cos(2\pi f_c t + \phi)]$$

Since the term $2\pi f_c t$ is twice the carrier frequency, it can be filtered out using a low-pass filter. What remains is:

$$M(t) \cdot cos(\phi)$$

This result is problematic if the phase φ is unknown. If φ is close to $\frac{\pi}{2}$, it will nullify the signal. Even if φ is not exactly $\frac{\pi}{2}$ or zero, the signal-to-noise ratio (SNR) will degrade by a factor of $cos^2(\varphi)$. This phase offset issue can significantly impact performance.

Similarly, frequency offset introduces additional complications. Suppose at the receiver, the local oscillator frequency is $f_c + \Delta f$ instead of the expected f_c. The received signal is:

$$M(t) \cdot cos(2\pi f_c t) \cdot cos\big((2\pi f_c + \Delta f)t + \varphi\big)$$

Using the trigonometric identity $cosa \cdot cosb = \frac{1}{2}[cos(a-b) + cos(a+b)]$, we get:

$$M(t) \cdot cos(2\pi f_c t) \cdot cos(2\pi(f_c + \Delta f)t + \varphi)$$
$$= M(t) \cdot \frac{1}{2}[cos(2\pi \Delta f t + \varphi) + cos(2\pi f_c t + 2\pi(f_c + \Delta f)t + \varphi)]$$

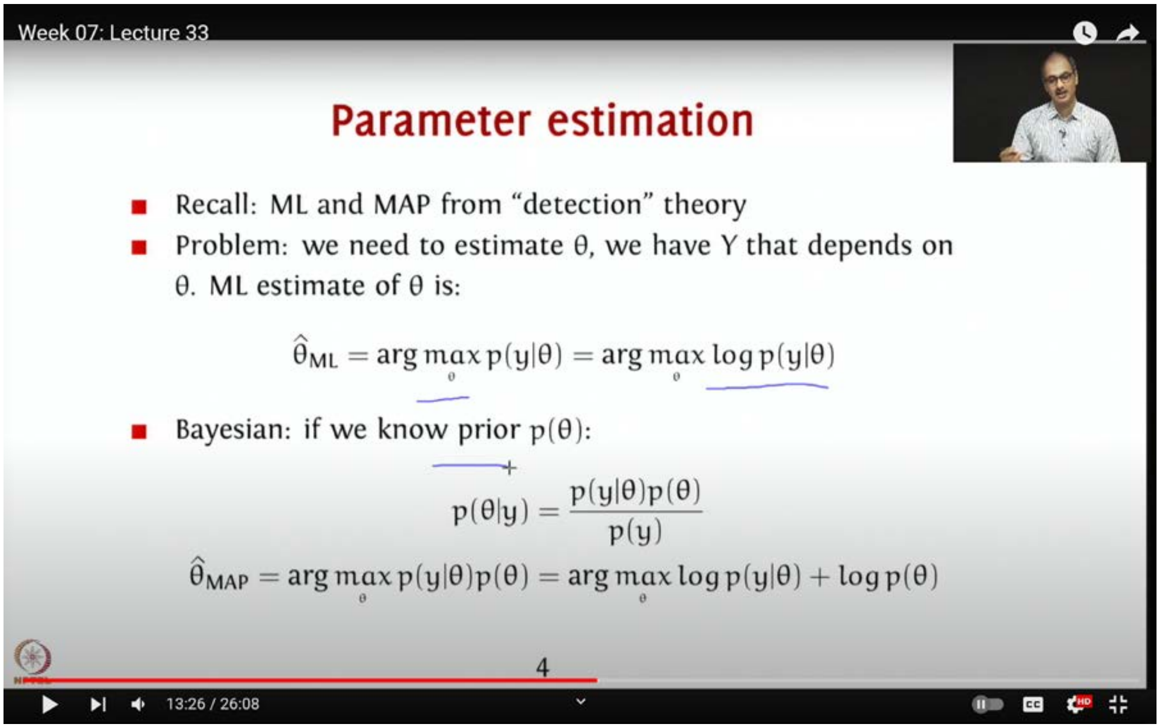
Ignoring the constant term and the factor of 2, which can be filtered out, we are left with:

$$M(t) \cdot cos(2\pi \Delta f t + \varphi)$$

Here, Δf represents the frequency offset, and it can cause additional issues similar to phase offset. Both phase and frequency offsets need to be carefully managed to avoid significant degradation in the performance of the communication system.

In this scenario, accurately determining Δf and φ is crucial. If, for example, Δf is close to 50 or 60 Hertz, your signal will exhibit oscillations. Ideally, you want to recover M(t) at the receiver to extract the symbols, but with the additional modulated term, some parts of M(t) will be amplified while others are diminished. This variation makes it nearly impossible to correctly identify the symbols unless you adjust Δ f and correct for φ. In other words, dealing with phase and frequency offsets poses a significant challenge in retrieving your symbols, which must be resolved by accurately estimating these parameters.

(Refer Slide Time: 13:26)



Additionally, if M(t) undergoes a scaling factor A, this scaling factor must also be determined. For instance, if your constellation is QAM-16 and the scaling alters the regions, all your decisions will be incorrect if the constellation is scaled improperly. Therefore, you need to account for amplitude scaling as well. To summarize, you must determine several parameters: the sampling point, the frequency offset, and the amplitude. These steps are essential for successful parameter estimation.

Let’s move on to an introduction to parameter estimation. Recall that maximum likelihood (ML) and maximum a posteriori (MAP) estimation were used to determine which symbol was transmitted. In ML estimation, we identified the symbol that maximized the likelihood function given the received signal Y. In the case of MAP estimation, we found the symbol with the highest posterior probability, given Y. For equiprobable symbols, ML was used to maximize the probability of receiving Y.

The distinction between parameter estimation and symbol detection lies in the nature of the problem. In symbol detection, the symbols come from a finite set, such as a QPSK or QAM-16 constellation. For QAM-16, you would evaluate 16 possible symbols to find the one that maximizes the likelihood or minimizes the distance metric. Parameter estimation, on the other hand, involves estimating continuous parameters like Δ f and φ, which affect the signal's recovery and quality.

In parameter estimation, the goal is to find the value of a parameter, θ, that best describes a given observation. This parameter could represent various aspects such as amplitude, phase, delay, etc. We are interested in estimating θ based on the observation Y, where θ is a continuous variable. For instance, in the case of phase estimation, θ might range from $-\pi$ to π or from 0 to 2π. This means we are selecting a value from an infinite set of possible values.

This distinction is what defines the problem as one of estimation. Unlike detection, where we choose from a finite set of symbols, estimation involves determining a specific value from a potentially infinite range. In the context of maximum likelihood (ML) estimation, $\widehat{\theta_{\mathrm{ML}}}$ is found by maximizing the probability $P(Y \mid \theta)$. Essentially, we vary θ to find the one that maximizes this probability. Since probabilities are non-negative, we can take the logarithm of the likelihood function. Using the logarithm is helpful because it simplifies calculations, particularly with functions like the Gaussian, where the exponential function complicates matters.

In Bayesian estimation, if we have a prior distribution P(θ), we update our estimate based on this prior knowledge. For example, if we are estimating phase and know it is likely between -π/2 and π with high probability, this prior can be incorporated. The posterior probability $P(\theta \mid Y)$ is proportional to $P(Y \mid \theta) \times P(\theta)$. If P(θ) is uniform for all values, the Bayesian approach simplifies to ML estimation. This is consistent with detection theory, where MAP and ML approaches align for equiprobable symbols.

So, remember, using logarithms can simplify complex expressions, especially when dealing with functions involving exponentials. As an example of a parameter estimation problem, let's consider determining the amplitude A.

Let's consider a simplified example of Binary Phase Shift Keying (BPSK) communication. In BPSK, the symbol B can be either +1 or -1. Suppose you have a system where you receive a number Y, and you need to estimate the amplitude A while being affected by Gaussian noise with zero mean and variance σ^2.

In this scenario, the problem is essentially a reformulation of the BPSK detection problem. Specifically, if B is +1, then Y follows a Gaussian distribution with mean A and variance σ^2. Conversely, if B is -1, then Y follows a Gaussian distribution with mean -A and variance σ^2.

Mathematically, if B is known, the probability density function of Y given A and B = +1 is:

$$P(Y \mid A, B = +1) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(Y - A)^2}{2\sigma^2}\right)$$

Similarly, if B is -1, the probability density function is:

$$P(Y \mid A, B = -1) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(Y + A)^2}{2\sigma^2}\right)$$

To estimate A, we perform maximum likelihood estimation (MLE). We find the value of A that maximizes the likelihood function. Intuitively, if we have one sample, the MLE is straightforward: without noise, Y would exactly match A or -A depending on whether B is +1 or -1.

If the value of B is known, which corresponds to training-based estimation, the receiver can directly use the value of B to determine A. For instance, if the receiver is informed that B is +1, then it knows to expect A as the signal. Similarly, if B is -1, the receiver expects -A. In the presence of noise, the maximum likelihood estimate of A will still be the value that best matches the received Y, since noise affects Y but does not change the basic principle of estimation.

When attempting to estimate the amplitude A in BPSK communication, your best guess is typically the mean of the received signal. Thus, if B is known, you can estimate A directly: if B is +1, then $\hat{A}$ is Y; if B is -1, then $\hat{A}$ is -Y.

However, if B is unknown, you need to average over the possible values of B. In practice, if the received value is significantly high, say, +7 and the noise variance is known to be 1, you can reasonably infer that B was +1. Thus, you can estimate A to be close to 7, possibly 7.1, depending on the exact received value.

Conversely, if you receive a very high negative value, such as -6, you would estimate A to be close to +6. This is because a high negative value makes it very unlikely that B was +1. The problem arises when the received signal is close to zero, as it becomes difficult to determine whether B was +1 or -1.

In cases where the received value is near zero, the situation becomes more uncertain. The hyperbolic cosine function, cosh, helps address this uncertainty. Simplifying the expression involving $e^{-\frac{(Y-A)^2}{2\sigma^2}}$ and $e^{-\frac{(Y+A)^2}{2\sigma^2}}$, you'll find that the terms reduce to a form involving cosh, which captures the uncertainty when the received signal is close to zero compared to the noise variance σ.

Thus, if the received value is several standard deviations away from zero, you can confidently estimate B as +1 or -1. However, relying on a single sample for estimation is generally insufficient and prone to error.

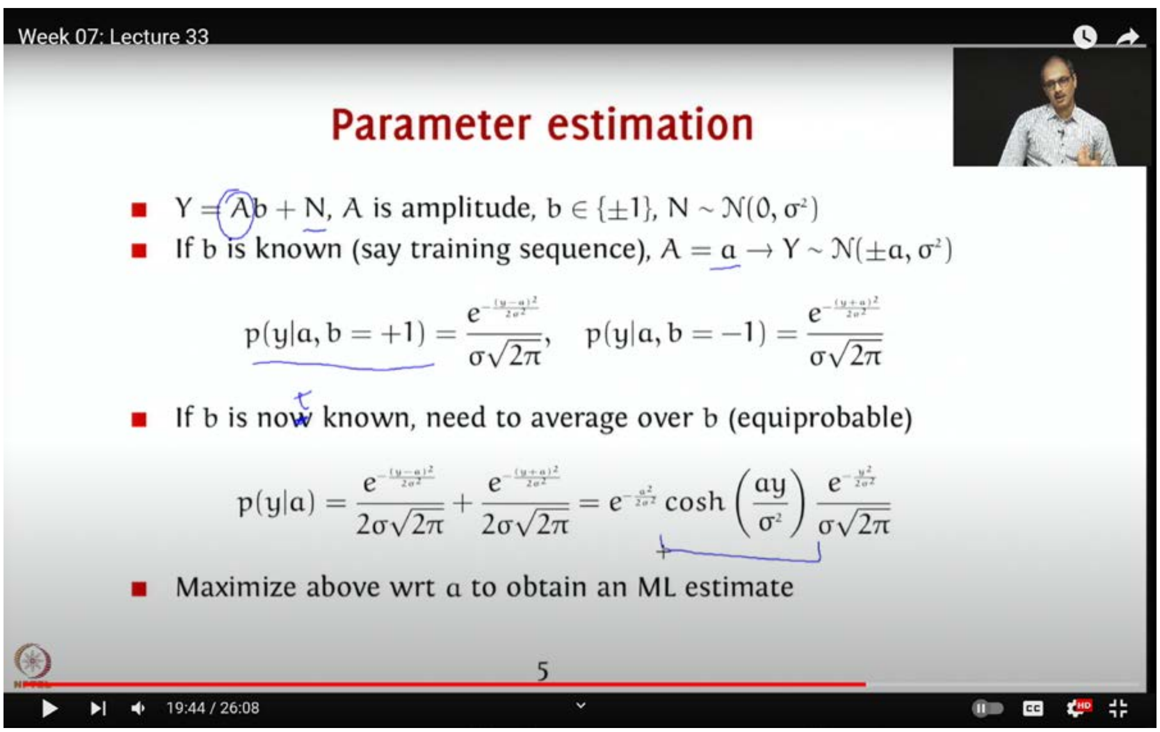
Consider the scenario where you are using your cell phone, which is calibrating itself to the base station. When you are in close proximity to the base station, you can receive a clear signal with high signal-to-noise ratio (SNR). However, if you are inside a building or a tunnel, the SNR decreases because the noise level is relatively high compared to the signal.

In such situations, it is beneficial to use multiple symbols for better estimation. This leads us to a vector-based approach where the received signal can be represented as:

$$y_k = b_k + n_k$$

where k ranges from 0 to K-1, and A is the unknown parameter we want to estimate.

(Refer Slide Time: 19:44)



This problem is analogous to the single-symbol case but is extended to multiple measurements. In this vector form, you collect multiple observations: $y_0, y_1, \ldots, y_{K-1}$. Since the noise realizations n_k are independent and identically distributed Gaussian variables, averaging these observations will help in reducing the impact of noise and improving the estimate.

For example, if b_k can be either +1 or -1, you adjust the sign of the corresponding y_k based on the value of b_k. In an extreme case where all b_k are +1, the received signals would be:

$$y_0 = A + n_0$$

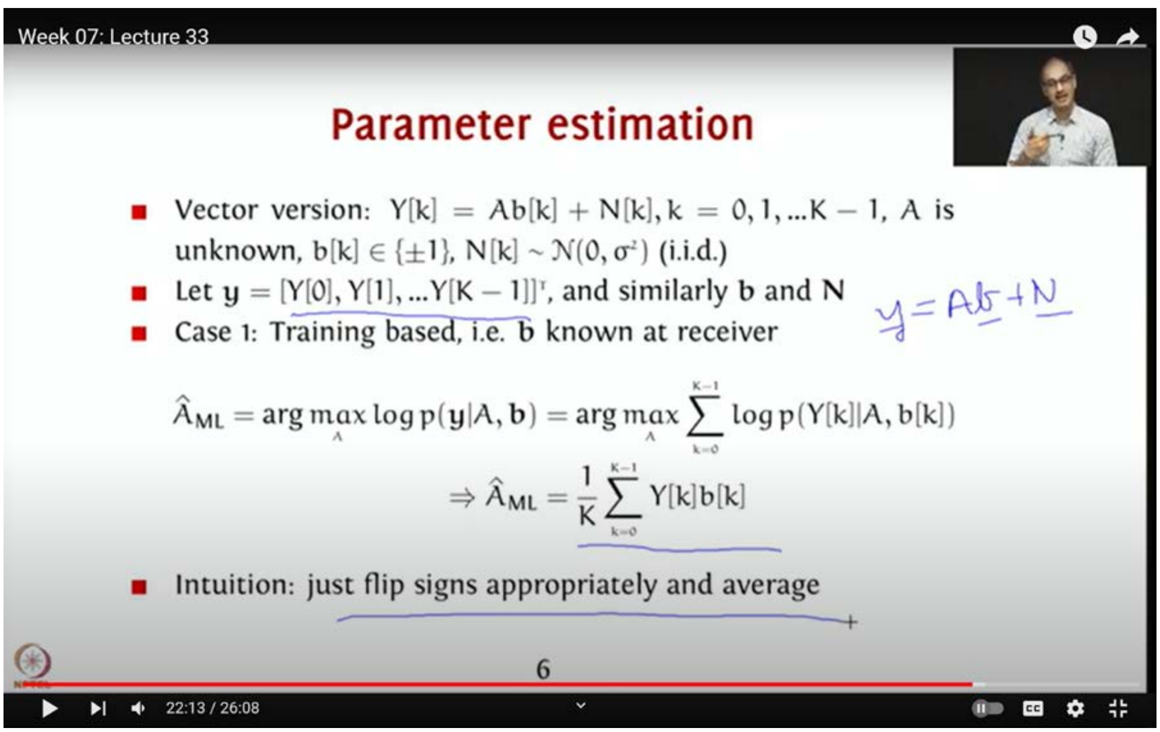
$$y_1 = A + n_1$$

$$y_2 = A + n_2$$

$$\vdots$$

$$y_{K-1} = A + n_{K-1}$$

(Refer Slide Time: 22:13)



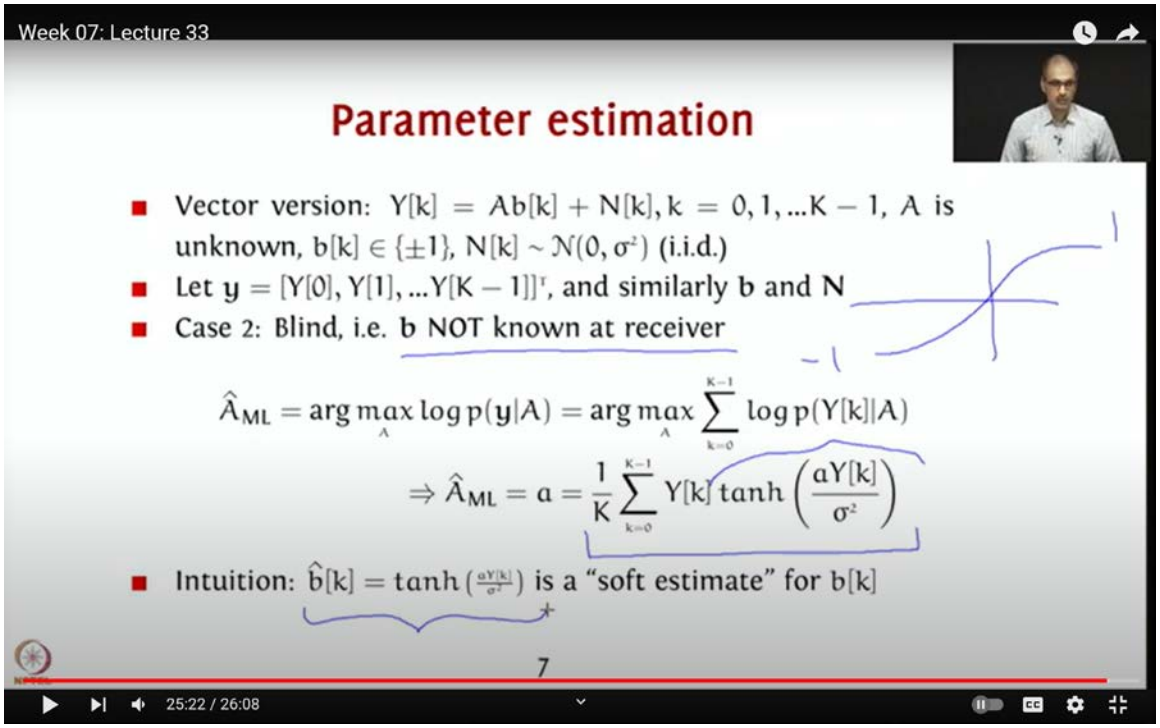
The optimal approach here is to average the observations. Averaging effectively reduces the noise variance and provides a more accurate estimate of A. To formalize this, you represent y as a column vector consisting of the stacked values $y_0, y_1, \dots, y_{K-1}$. Similarly, create vectors for b and n. The equation can then be expressed as:

$$y = A \cdot b + n$$

In this framework, $\bar{y}$ is a Gaussian random vector with mean $A \cdot b$ and variance $\sigma^2 \cdot I$. Given the vector b, you can perform a weighted average to estimate A. This approach is straightforward: if you multiply y by b^T, you obtain a sufficient statistic for estimating A. This makes the estimation process efficient and accurate.

The core idea here is straightforward: flip the signs of the samples and average them appropriately. Each sample carries equal weight because the noise is independent and identically distributed, so averaging effectively reduces the noise impact and provides a reliable estimate.

(Refer Slide Time: 25:22)



However, in practical scenarios, such as when a transmission is sent to multiple users, you may not know the value of b_k for each receiver. For instance, imagine ten people receiving a signal simultaneously. It's impractical to calibrate each user individually, as it would delay the transmission for others. Instead, the system must be designed to estimate the parameter A without prior knowledge of b_k. This approach is known as blind estimation.

In blind estimation, the problem is formulated similarly to the previous case, but now you need to average over all possible values of b_k. Essentially, you need to marginalize over b_k to eliminate it from the equations. Although I won't delve into the details in this lecture, this process leads to a form resembling the earlier approach.

Previously, you had an expression involving:

$$\frac{1}{K}\sum_{k=0}^{K-1} y_k b_k$$

which involved flipping the signs and averaging. In the blind estimation case, this transforms into a form involving the hyperbolic tangent function:

$$\tanh\left(\frac{Ay_k}{\sigma^2}\right)$$

The hyperbolic tangent function provides a soft estimate of b_k. To illustrate, the tanh function is shaped like this:

- It approaches -1 as its argument goes to $-\infty$
- It approaches 1 as its argument goes to $+\infty$

When you receive a highly negative or highly positive y, the tanh function assigns a higher weight to these values. For instance, if y is several times the noise standard deviation σ, it contributes more to the estimate. Conversely, if y is close to zero, the tanh function provides less weight. This behavior allows the function to act almost like a soft estimate for b_k while simultaneously estimating A.

In summary, the tanh function and similar sigmoid functions are useful in blind estimation because they provide a soft, adaptive estimate of b_k while estimating A, effectively handling the uncertainty in the value of b_k.

If a sample is not reliable, you won't assign it much weight. To put it simply, if you use the hyperbolic tangent function to estimate $\widehat{b_k}$, then if the amplitude is reliably high, whether positive or negative, you'll give that sample more weight. This is the fundamental approach we're using here.

For now, let's conclude this lecture. In the next one, we will continue our exploration of parameter estimation and expand our discussion to include the estimation of other parameters, such as phase, frequency, delay, and so forth. Thank you for your attention.