Digital Communication using GNU Radio

Prof. Kumar Appaiah

Department of Electrical Engineering

Indian Institute of Technology Bombay

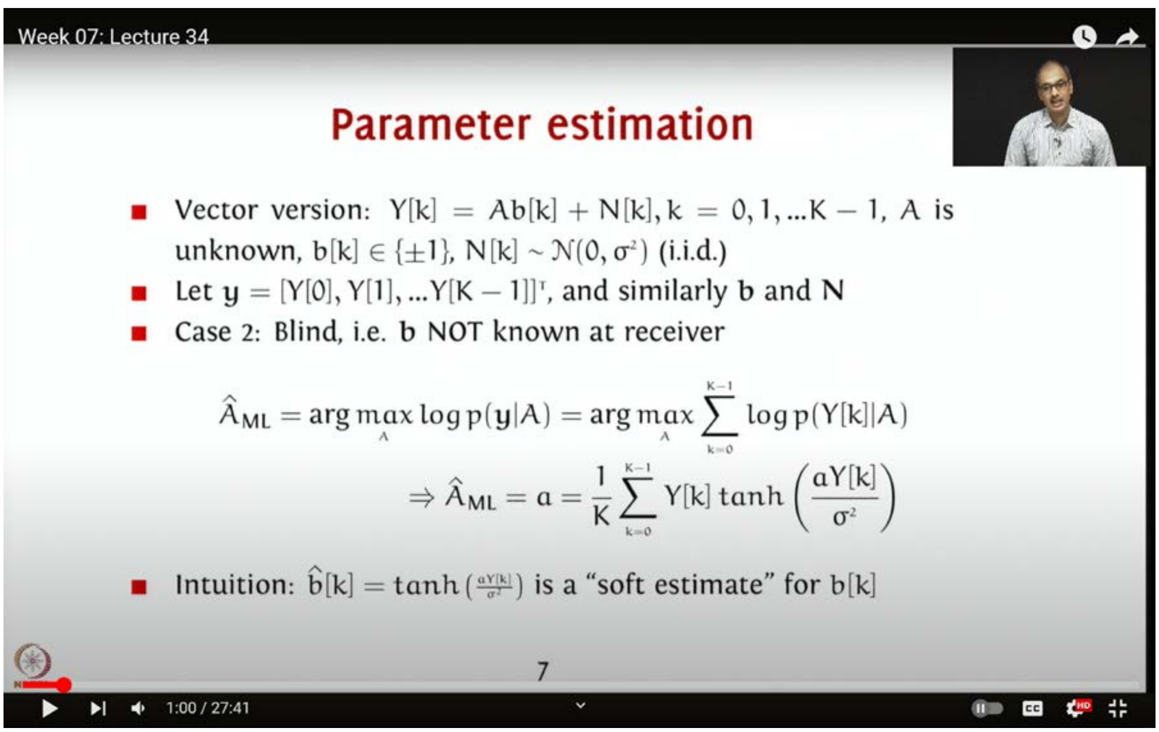
Week-07

Lecture-34

Parameter Estimation for Practical Receivers - Part 2

Hello, and welcome to this lecture on Digital Communication Using GNU Radio. I'm Kumar Appiah, and in this session, we will continue our exploration of synchronization, parameter estimation, and non-coherent communication, topics we began discussing in the previous lecture.

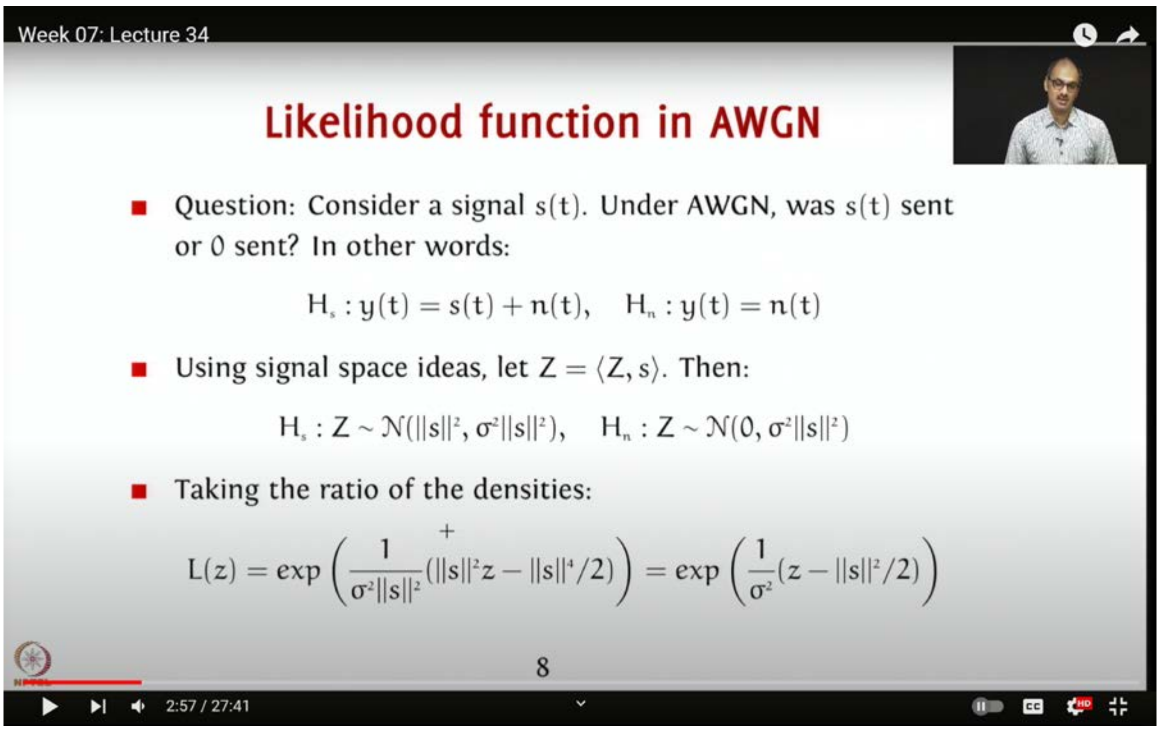
(Refer Slide Time: 01:00)



To recap, in our last session, we focused on parameter estimation, specifically dealing with the unknown amplitude of a transmitted signal. We discussed the approach of using known symbols, by employing a training mode, to achieve an estimate of the parameter a. This involved methods like weighted averaging based on the symbols sent.

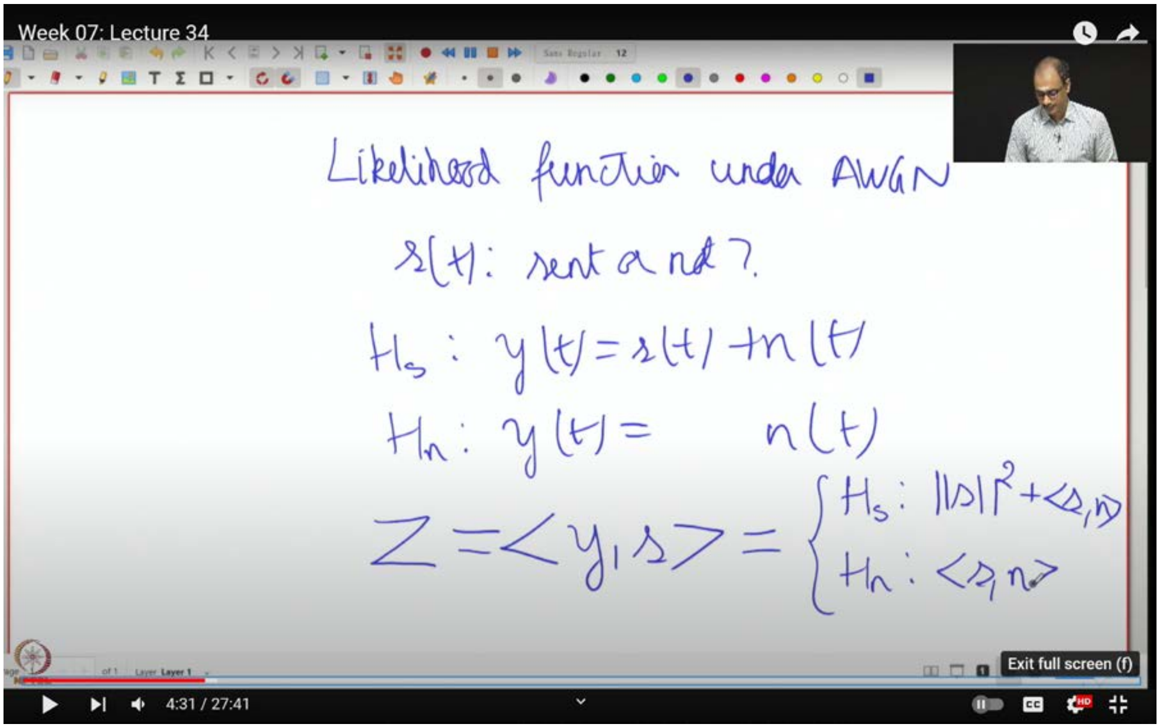
For scenarios where we don't have prior knowledge of the symbols, known as the blind estimation approach, the maximum likelihood estimation (MLE) method involves taking soft estimates. This means we collect received values and weigh them according to their reliability to derive an optimal estimate of the amplitude. Today, we will build upon these concepts and delve deeper into our discussion.

(Refer Slide Time: 02:57)



To address our problem, we will use the concept of the likelihood function. As we have previously discussed, the likelihood function helps us determine whether a particular signal was transmitted under the conditions of Additive White Gaussian Noise (AWGN). Specifically, we want to answer the question: Was the signal S(t) transmitted, or was it just zero? This problem is akin to binary signaling, where we previously focused on detecting symbols. However, in this case, we need to determine whether a specific signal S(t) was sent or not. We will use the hypothesis testing framework we applied in symbol detection.

(Refer Slide Time: 04:31)



Let's define our hypotheses:

- **Hypothesis H_S:** y(t) = S(t) + n(t), meaning the signal S(t) was transmitted, and n(t) represents the noise.
- **Hypothesis H_N:** y(t) = n(t), meaning no signal was transmitted, and only noise is received.

In the absence of noise, determining whether S(t) was sent is straightforward: you simply check for the amplitude of S(t). However, with noise present, the observed amplitude can be affected by the noise, making it less obvious whether S(t) was indeed transmitted. If S(t) has a very weak amplitude compared to the noise, distinguishing between the two hypotheses becomes challenging.

To tackle this problem, we will use the concept of the signal space. Let's write the likelihood function under AWGN:

- If S(t) was sent: y(t) = S(t) + n(t).

- If S(t) was not sent: y(t) = n(t).

Given only y(t), our goal is to decide whether S(t) was sent or not. We will compute our sufficient statistic Z, defined as the inner product of y(t) with S(t):

$$Z = \int y(t)S(t)\, dt$$

This integral will yield different results under the two hypotheses:

- Under H_S: $Z = |S|^2 + \langle S, n \rangle$, where $|S|^2$ represents the power of S(t) and $\langle S, n \rangle$ is the cross-term involving noise.
- Under H_N: $Z = \langle S, n \rangle$, which only includes the noise term.

To decide between H_S and H_N, we need to evaluate the likelihood function for each hypothesis and choose the one that maximizes it. This process ensures that we use the maximum likelihood approach to determine whether S(t) was transmitted.

Let's analyze the distribution of Z under the two hypotheses.

Under hypothesis H_S, where S(t) is transmitted, the distribution of Z is:

$$Z \sim \text{Gaussian}(|S|^2, \sigma^2 |S|^2)$$

Here, $|S|^2$ is a constant representing the signal's power, and $< S, n >$ is a random variable. As we discussed earlier, $< S, n >$ is the projection of the white Gaussian noise onto the signal S. This random variable has a mean of 0 and a variance of $|S|^2 \sigma^2$, as we covered in our discussion of random variables in the signal space context.

Thus, the probability density function of Z under H_S is:

$$p(Z|H_S) \propto \exp\left(-\frac{(Z - |S|^2)^2}{2\sigma^2 |S|^2}\right)$$

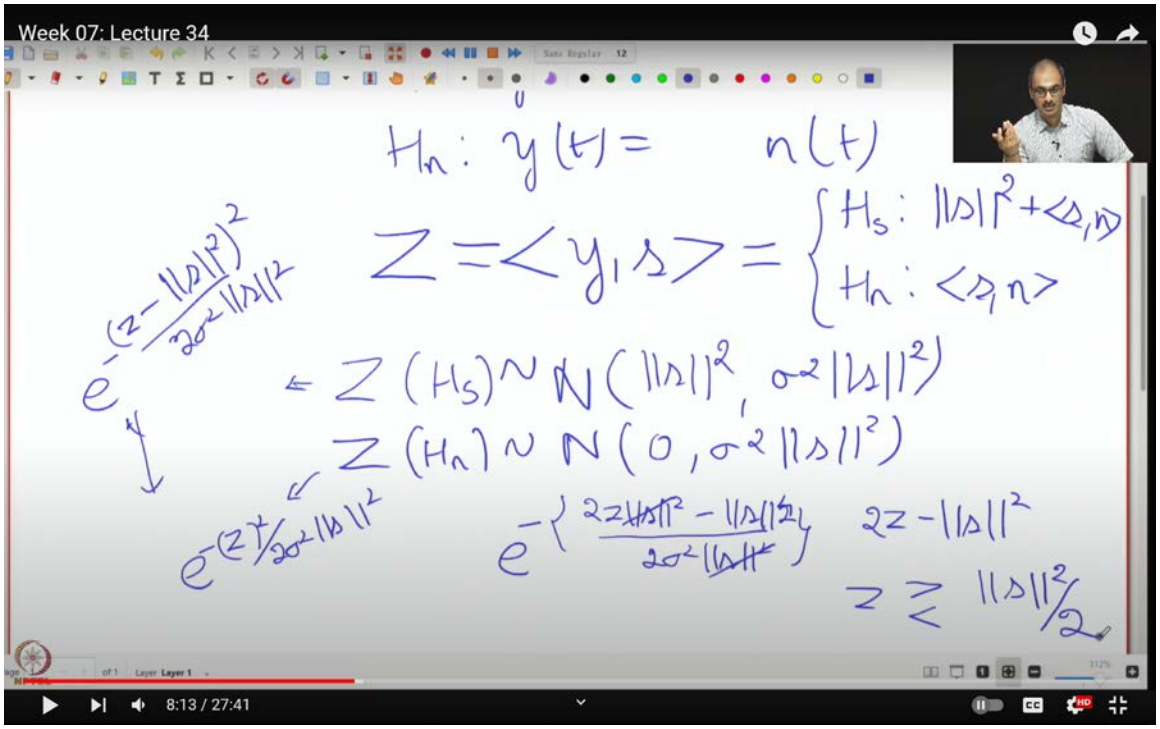
Under hypothesis H_N, where only noise is received, the distribution of Z is:

$$Z \sim \text{Gaussian}(0, \sigma^2 |S|^2)$$

The probability density function of Z under H_N is:

$$p(Z|H_N) \propto \exp\left(-\frac{Z^2}{2\sigma^2|S|^2}\right)$$

(Refer Slide Time: 08:13)



To decide which hypothesis is more likely, we need to compare these two distributions for the given Z. We can simplify this comparison by focusing on the likelihood ratio, which involves dividing one distribution by the other.

Thus, the likelihood ratio is:

$$\frac{p(Z|H_S)}{p(Z|H_N)} \propto \frac{\exp\left(-\frac{(Z-|S|^2)^2}{2\sigma^2|S|^2}\right)}{\exp\left(-\frac{Z^2}{2\sigma^2|S|^2}\right)}$$

Simplifying this, we get:

$$\frac{p(Z|H_S)}{p(Z|H_N)} \propto \exp\left(-\frac{(Z-|S|^2)^2 - Z^2}{2\sigma^2|S|^2}\right)$$

Expanding the numerator:

$$(Z-|S|^2)^2 - Z^2 = Z^2 - 2Z|S|^2 + |S|^4 - Z^2 = -2Z|S|^2 + |S|^4$$

So:

$$\frac{p(Z|H_S)}{p(Z|H_N)} \propto \exp\left(\frac{2Z|S|^2 - |S|^4}{2\sigma^2|S|^2}\right)$$

Further simplification gives:

$$\frac{p(Z|H_S)}{p(Z|H_N)} \propto \exp\left(\frac{2Z - |S|^2}{2\sigma^2}\right)$$

Thus, we just need to check the sign of $2Z - |S|^2$. If $2Z - |S|^2$ is positive, then H_S is more likely; if it is negative, then H_N is more likely. This simplifies our decision-making process and provides a clear criterion for hypothesis testing.

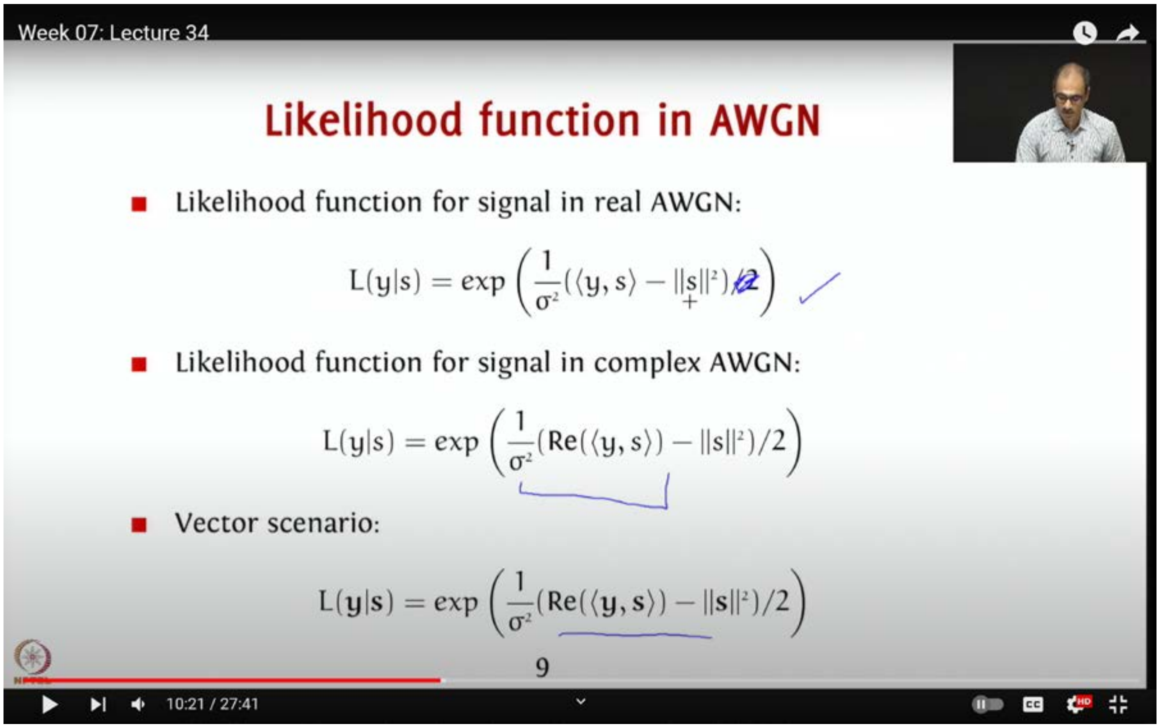
If Z is closer to $\frac{|S|^2}{2}$, it is more likely that S was sent. Conversely, if Z is closer to 0, it is more likely that S was not sent. This aligns with our discussion on binary signaling. Essentially, if you take the ratio of the densities, it becomes clear that $\frac{|S|^2}{2}$ serves as the decision boundary.

This method provides a way to determine whether S was sent or not. This approach is a fundamental component, and similar strategies can be employed for various other types of parameter estimation. It's crucial to remember how we utilized the sufficient statistic Z to evaluate the likelihood. This process was demonstrated with real additive white Gaussian noise (AWGN).

For complex AWGN, although I won't go through the details here, if you perform the same analysis with complex Gaussian noise, you will find that the decision criterion becomes:

$$\frac{1}{\sigma^2}\mathrm{Re}(\langle Y,S\rangle)-\frac{|S|^2}{2}$$

(Refer Slide Time: 10:21)

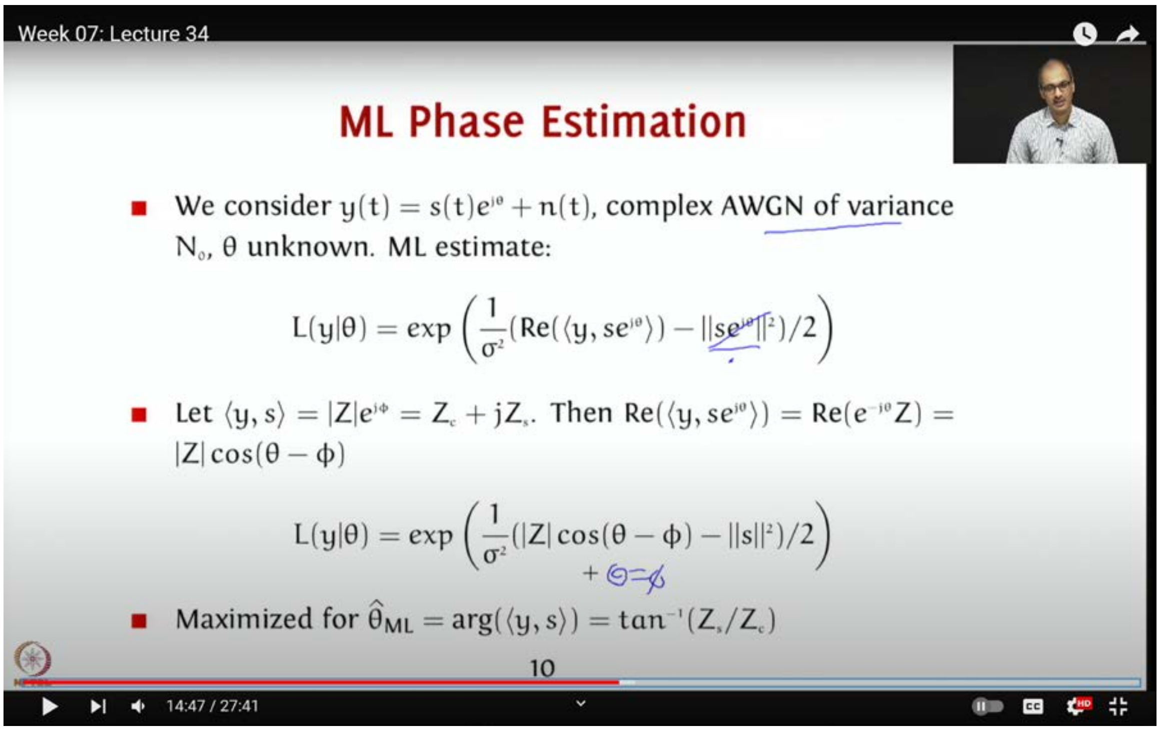


This makes sense because, when aligning the signal along the dimension of S, only that dimension matters. The component not aligned with S is irrelevant. Thus, the real part of the inner product $\langle Y,S\rangle$ is what truly matters.

In a vector scenario, where $Y_1,Y_2,Y_3,\ldots$ represent S(t) plus independent and identically distributed (i.i.d.) noise samples, you would combine the distributions of these measurements. You use the fact that the noise values are i.i.d. to effectively combine them. The goal remains to maximize this combined function. This essentially translates to finding the real part of $\langle Y,S\rangle$ and determining if it falls to the left or right of $\frac{|S|^2}{2}$. Please make sure to correct this to $\frac{|S|^2}{2}$.

Let's now address a practical problem of significant importance: maximum likelihood phase estimation. To recap from the previous lecture, we discussed that when there is a phase offset φ, the I-component of the signal, denoted as M(t), is affected by a multiplication with cos φ. Similarly, the Q-component will be affected by a multiplication with sin φ. Thus, a phase offset essentially multiplies your signal S(t) by $e^{j\theta}$, where θ represents the unknown phase we need to estimate.

(Refer Slide Time: 14:47)



In other words, at the transmitter, the signal is $e^{j2\pi f_c t}$, while at the receiver, it is $e^{j2\pi f_c t+\theta}$. This results in a situation where you have cos(2πf_ct + θ) and sin(2πf_ct + θ) at the receiver, as opposed to just cos(2πf_ct) and sin(2πf_ct). This phase offset θ introduces a complication in the phase estimation process.

Consequently, the received signal Y(t) is given by S(t) $e^{j\theta}$. A point to note is that in various references, you might find slightly different formulations: some state that the receiver has cos(2πf_ct) and sin(2πf_ct), while the transmitter has cos(2πf_ct + θ) and sin(2πf_ct + θ). This is a minor difference. By accounting for the offset appropriately at one location, you will end up with the same results.

In this context, we are dealing with complex additive white Gaussian noise (AWGN) with variance σ^2, and our objective is to estimate the unknown phase θ.

Based on our previous discussion, we have the following formulation:

$$\text{Re}\langle Y, Se^{j\theta}\rangle - |Se^{j\theta}|^2/2$$

For simplicity, let's denote $\langle Y, S>$ as $|Z|e^{j\varphi}$, where $Z_c + jZ_s$ represents the real and imaginary components of Z, respectively.

Thus, the real part of $\langle Y, Se^{j\theta}>$ is:

$$\text{Re}\left(e^{-j\theta} \cdot Z\right)$$

Since $\langle Y, S>$, or $|Z|e^{j\varphi}$, can be expressed as $Z_c + jZ_s$, the real part of $\langle Y, Se^{j\theta}>$ simplifies to:

$$\text{Re}\left(e^{-j\theta} \cdot Z\right)$$

Note that this term's norm does not depend on θ because $|Se^{j\theta}|$ is equivalent to |S|. We are therefore interested in:

$$\text{Re}\left(e^{-j\theta} \cdot Z\right)$$

We examine this for different values of θ. For Z with a phase of φ, this becomes:

$$\text{mod}\, Z \cdot cos(\theta - \varphi)$$

So the likelihood function $L(\, Y \mid \theta \,)$ is given by:

$$L(\, Y \mid \theta \,) = \exp\left(\frac{1}{\sigma^2}[|Z|cos(\theta - \varphi) - |S|^2]\right)$$

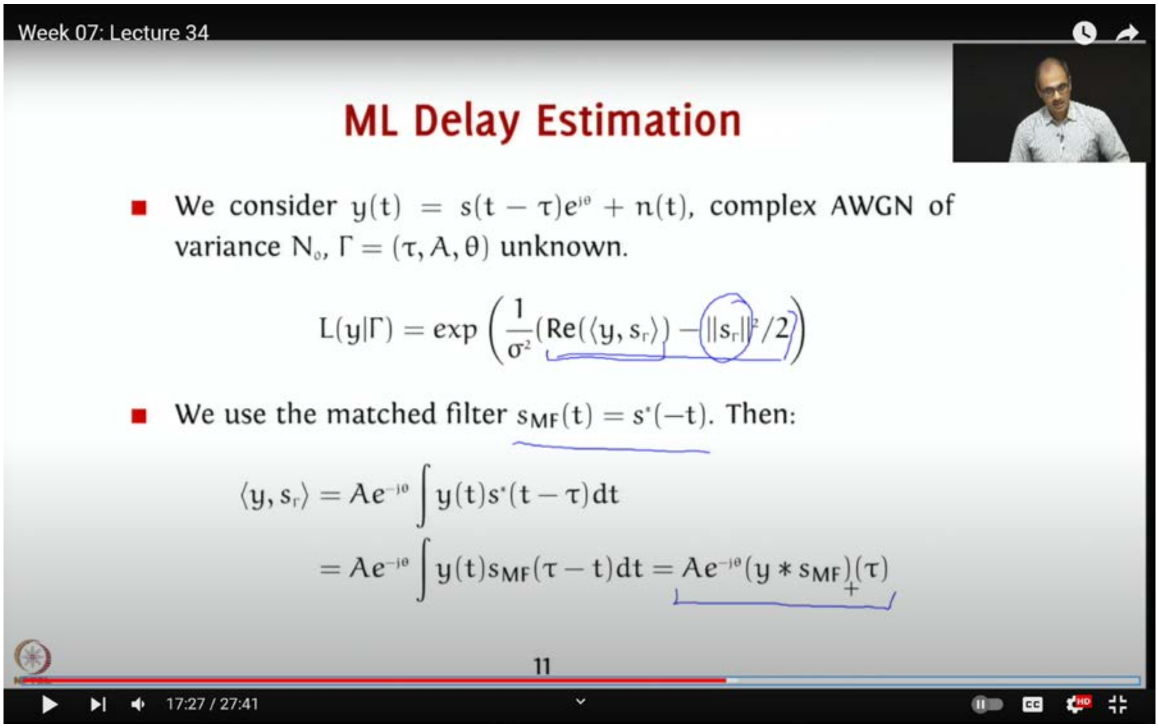
As mentioned earlier, |S| does not depend on φ, so we need to maximize this function with respect to θ. The optimal choice is θ = φ, where φ is the argument of Z. Specifically, φ is given by:

$$\varphi = \tan^{-1}\left(\frac{Z_s}{Z_c}\right)$$

Thus, the optimal θ is the angle of Z, which is $\frac{Z_s}{Z_c}$. Intuitively, this makes sense because to find the phase offset, you essentially need to rotate S to align with Y in a way that maximizes the inner product. This concept can be easily verified using GNU Radio.

Next, we will discuss maximum likelihood delay estimation.

(Refer Slide Time: 17:27)



In the case of maximum likelihood delay estimation, we start with the expression:

$$Y(t) = S(t - \tau)e^{j\theta} + n(t)$$

where n(t) represents complex AWGN with variance σ^2. Our objective here is to estimate three parameters simultaneously: the amplitude a, the phase θ, and the delay τ. This creates a three-pronged problem, as we need to determine all three parameters.

To approach this, we parameterize the problem by defining a tuple or vector Γ that includes τ, a, and θ. Intuitively, we evaluate different values of S for various combinations of τ, a, and θ, and compute the overlap integral to find the combination that maximizes this integral.

Unfortunately, this involves searching through a vast parameter space. To make this process more efficient, we use a matched filter approach. The matched filter essentially performs an operation similar to the integral we discussed. Specifically, we have:

$$\langle Y, S(\Gamma)\rangle = \mathrm{A}e^{-j\theta}$$

Here's how it works:

1. We apply a matched filter, which involves the convolution of Y(t) with $e^{j\theta}$, effectively removing the phase term $e^{-j\theta}$.

2. We then account for the delay τ and scale the result by the amplitude a.

This process gives us:

$$\mathrm{A}e^{-j\theta}\int Y(t)S^{*}(t-\tau)\ dt$$

where $S^{*}(t-\tau)$ denotes the complex conjugate of S(t - τ).

At this stage, we set aside the norm $|S(\Gamma)|$ for a couple of reasons:

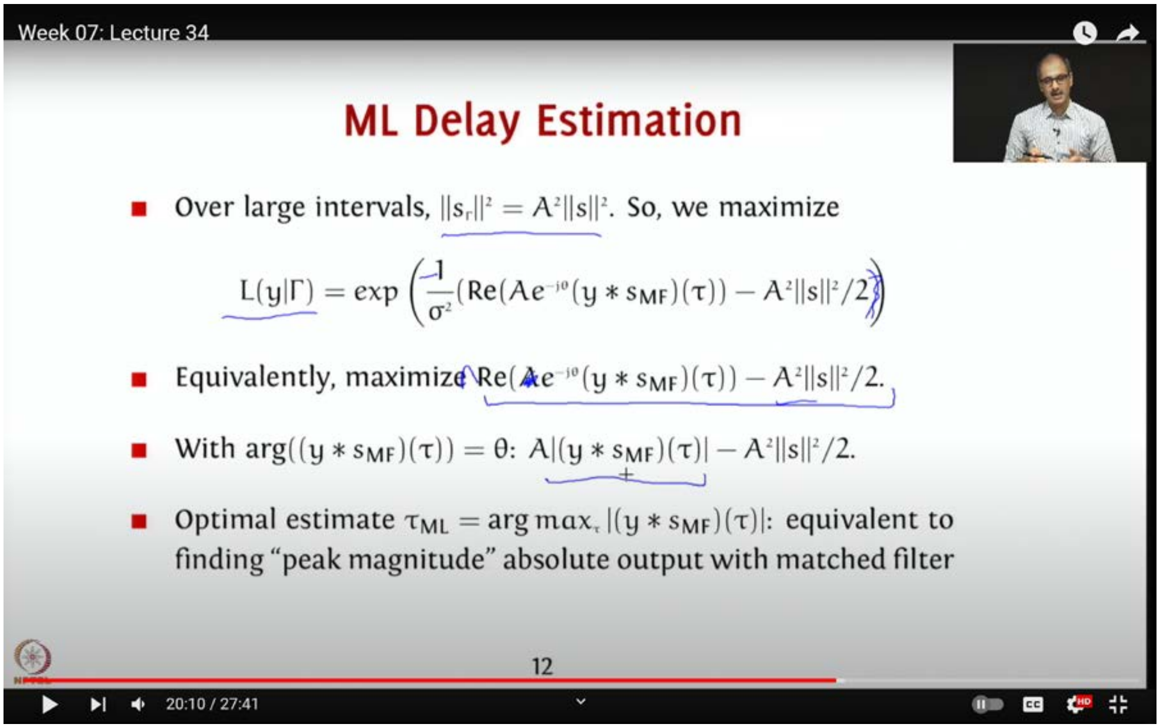
- The norm $|S(\Gamma)|$ does not depend significantly on the delay τ or the phase θ. Specifically, delaying S(t) or adding a phase shift does not change its norm.
- Therefore, our focus is on maximizing the integral:

$$\mathrm{A}e^{-j\theta}\int Y(t)S^{*}(t-\tau)\ dt$$

Here, we are optimizing over a, θ, and τ to find the maximum likelihood estimate for these parameters.

Our primary goal here is to estimate the delay τ. We previously discussed methods for finding the amplitude a and the phase θ, but now we focus on estimating τ. Over large intervals, we assume that the norm of S(Γ) is $A^2|S|^2$, where $S(\Gamma)$ represents S modified to include an amplitude scaling factor a, a delay τ, and a phase shift $e^{j\theta}$.

(Refer Slide Time: 20:10)



The phase shift $e^{j\theta}$ does not affect the norm, and the delay τ is irrelevant in the norm calculation over large intervals because shifting S(t) does not change its norm. Therefore, the likelihood function $L(Y \mid \Gamma)$ simplifies as follows:

$$L(Y \mid \Gamma) \propto \mathrm{Re}\left(Ae^{-j\theta}\langle Y, \mathrm{mf}\rangle(\tau)\right) - \frac{A^2|S|^2}{2}$$

Here, mf_τ denotes the matched filter response for the delay τ.

To maximize the likelihood function with respect to τ, we note that the term $\frac{A^2|S|^2}{2}$ does not affect the optimization process for τ because it is constant for a given τ. Thus, we focus on maximizing:

$$\mathrm{Re}\left(\mathrm{A}e^{-j\theta}\langle Y, \mathrm{mf}\,\rangle(\tau)\right)$$

Since a is a positive number, it can be factored out of the maximization problem. Therefore, the key step is to maximize the real part of:

$$\mathrm{Re}\left(\mathrm{A}e^{-j\theta}\langle Y, {}^*\mathrm{mf}\,(\tau)\rangle\right)$$

where the term involving a can be considered separately as it does not impact the maximization over τ.

To maximize the real part of $e^{-j\theta}\langle Y, \mathrm{mf}_\tau\rangle$, the process is quite intuitive. Essentially, if you align the phase of $\langle Y, \mathrm{mf}_\tau\rangle$ with θ, you simplify the expression to its modulus. In other words, by choosing mf_τ such that its phase matches θ, you effectively maximize the term. Therefore, to find the maximum value, you simply need to identify the peak value of $\langle Y, \mathrm{mf}_\tau\rangle$ as a function of τ.

Intuitively, this means you shift the matched filter across different delays and calculate the overlap integrals, finding the delay where this integral reaches its maximum value.

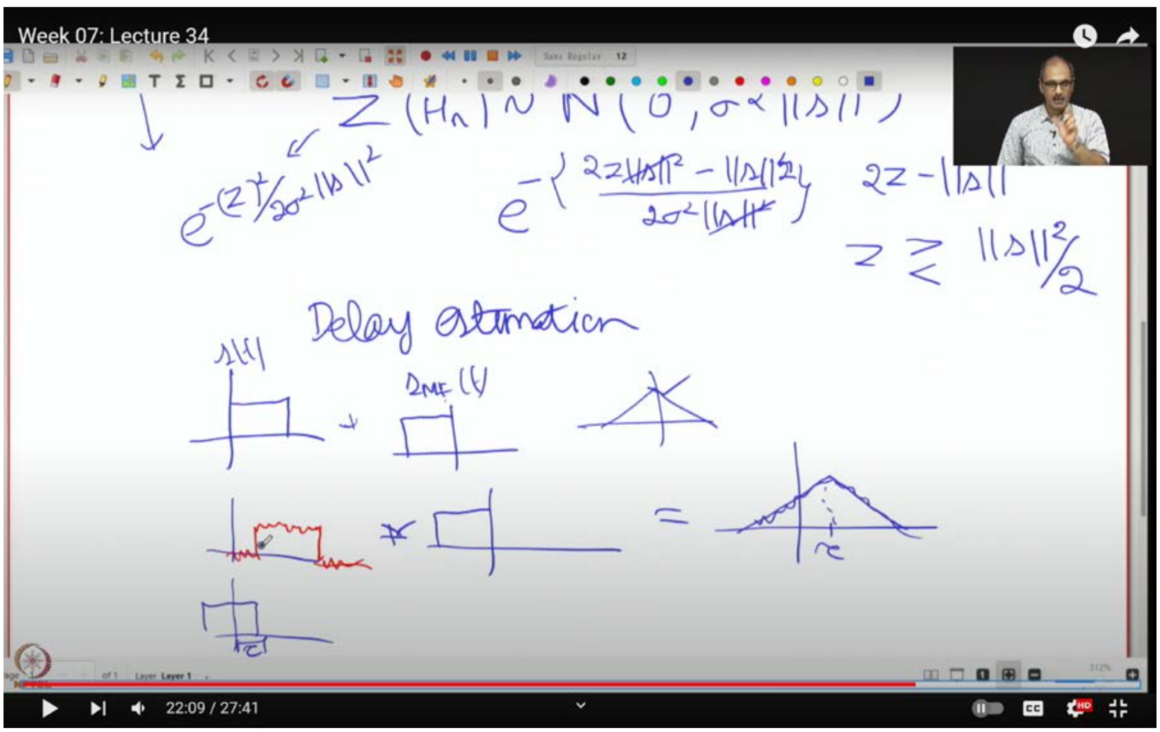
To illustrate this concept, consider a practical example with Gaussian noise. Suppose your template signal is represented by a waveform, and the observed signal exhibits variations due to noise. The matched filter is essentially a time-reversed version of the template signal.

If you convolve the observed signal with this matched filter, you should expect a peak in the resulting signal. This peak corresponds to the delay where the matched filter aligns best with the observed signal. In practical terms, if you shift the matched filter, you will eventually find that the peak value of the convolution output indicates the delay τ.

Even with noise, which may cause some waviness, finding this peak through shifting and convolving will yield the delay. This method, while potentially affected by noise, remains effective for estimating delay, and we will also verify this approach using GNU Radio.

If you introduce a random delay and then perform matched filtering, you can determine the delay by analyzing how the output of the matched filter is shifted. Essentially, instead of manually multiplying the signal by the matched filter at various points and integrating, which is essentially convolution, you can simplify the process. By convolving the signal with the matched filter and identifying the peak amplitude in the resulting output, you can directly determine the delay. This method works effectively for a single signal. However, if you have a signal composed of multiple pulses, you will need to make slight adjustments, but the core intuition remains the same: the highest correlation peak indicates the delay.

(Refer Slide Time: 22:09)

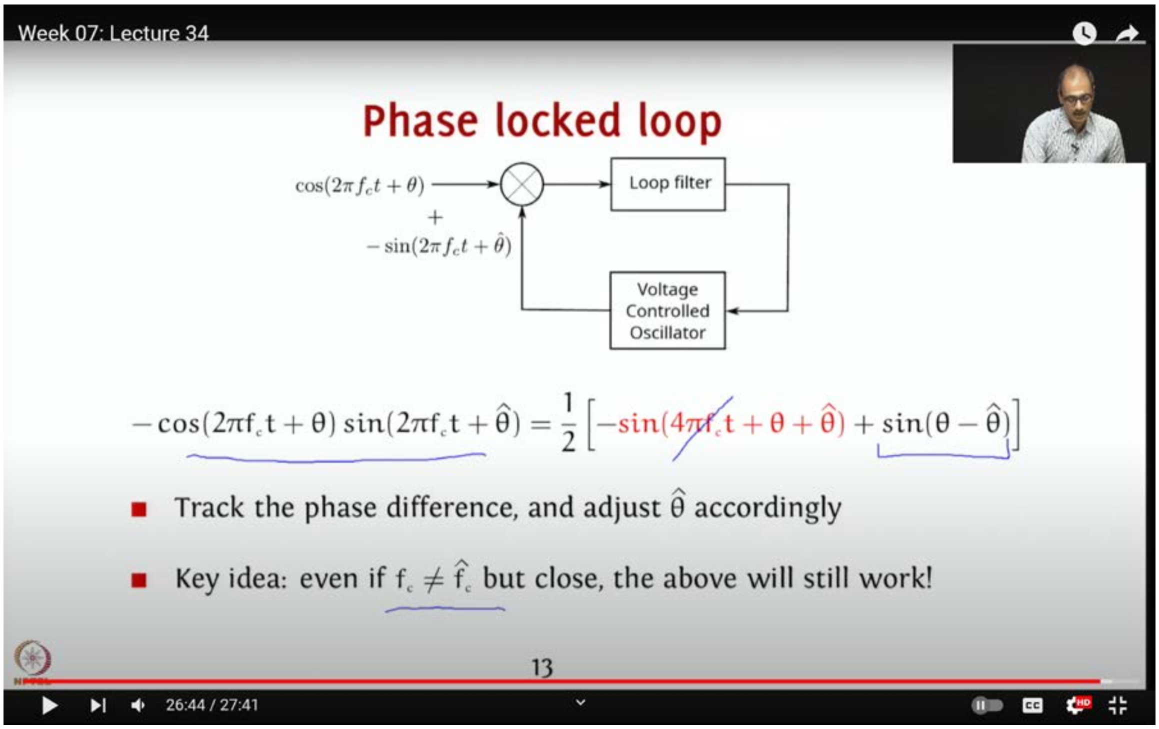


When it comes to tracking frequency offsets, the process becomes a bit more complex. Let's consider an intuitive approach: Suppose you have a signal represented as $\cos(2\pi f_c t + \theta)$. In a feedback-based system with a loop filter and a voltage-controlled oscillator, you can track frequency offsets. For simplicity, let's assume the frequency is constant.

If you multiply $\cos(2\pi f_c t + \theta)$ by $cos\left(2\pi fct + \hat{\theta}\right)$ and focus on the phase offset, you can use the cosine angle addition formula to simplify the result. The product yields

$\frac{1}{2}\left[cos\left(-4\pi f_c t + \theta - \hat{\theta}\right)\right]$, which can be simplified to $sin\left(\theta - \hat{\theta}\right)$ after applying a low-pass filter.

(Refer Slide Time: 26:44)



If θ and $\hat{\theta}$ are very close, then $sin\left(\theta - \hat{\theta}\right) \approx \theta - \hat{\theta}$. The loop filter works to track this phase offset $\theta - \hat{\theta}$ and continually adjust it. The voltage-controlled oscillator changes the frequency based on the voltage applied, thus providing a cosine or sine wave that reflects the applied voltage.

Given that $sin\left(\theta - \hat{\theta}\right)$ approximates $\theta - \hat{\theta}$ for small changes, if you repeatedly measure $\theta - \hat{\theta}$ over time and divide by t, you essentially calculate the frequency offset. In other words, a phase-locked loop (PLL) intuitively tracks the phase difference between θ and $\hat{\theta}$, and over time, averages this phase difference. This averaging effectively provides the frequency offset, allowing the PLL to track the frequency.

However, this PLL method works optimally with a pure carrier signal, such as a simple sine or cosine wave. It remains effective even in the presence of noise due to the low-pass filtering and averaging. When dealing with practical signals that have data modulated onto them, you need to be cautious. The PLL may need modification or adaptation to handle the data stream properly.

In training-based approaches, there are two common methods: you can either send a bare carrier signal or a carrier with higher power. This allows the PLL to be trained at specific locations, and the locked frequency is maintained until drift occurs. Typically, clock offsets range from 10 to 100 parts per million (ppm). For instance, a 10 ppm offset at 1 GHz corresponds to $\frac{10}{10^6} \times 1$ GHz, which translates to an offset in the range of kilohertz.

Phase-locked loops are designed to search and track frequencies effectively. Once the PLL is close to the target frequency, it locks onto it, even if the actual frequency f_c differs slightly from f_c'. The PLL corrects the phase offset and averages the phase changes to provide the necessary frequency compensation. This is the core intuition behind how phase-locked loops function.

In our next class, we will explore the phase-locked loop from an optimization perspective and delve into differential modulation techniques. Thank you.