Neural Networks for Signal Processing-I

Prof. Shayan Srinivasa Garani

Department of Electronic System Engineering

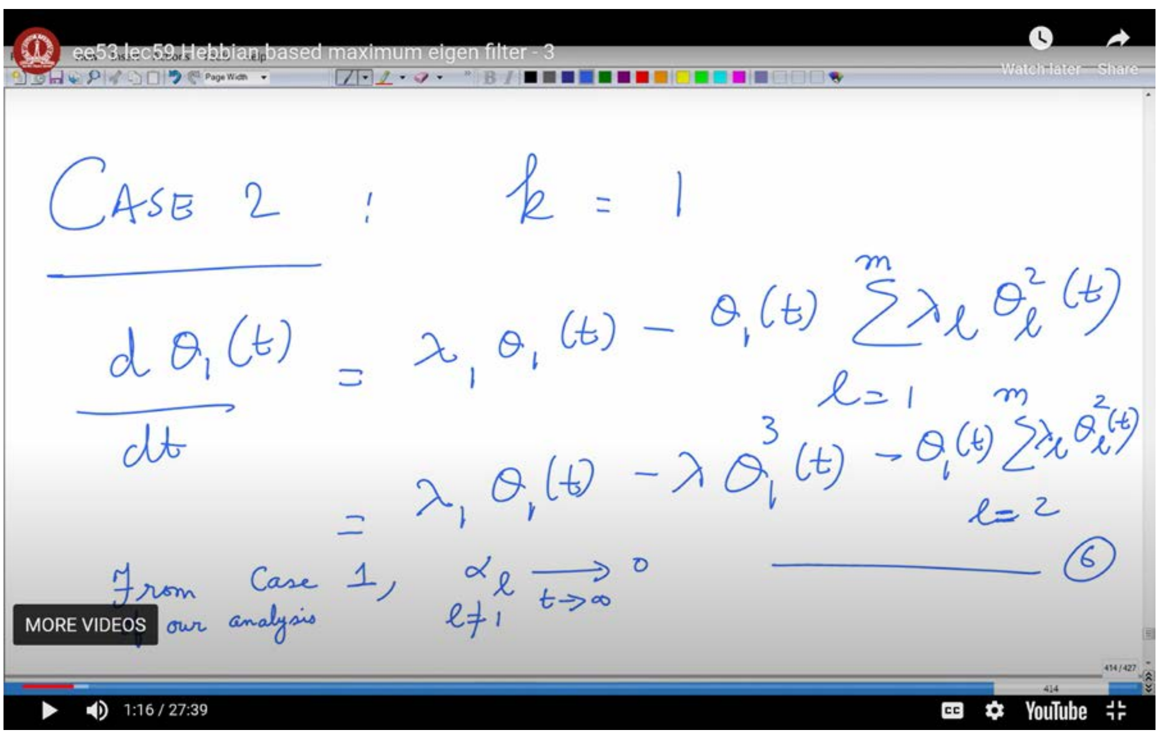
Indian Institute of Science, Bengaluru

Lecture – 59

Hebbian Based Maximum Eigen Filter – 3

In the previous lecture, we examined Case 1 of the stability analysis for the maximum Hebbian-based eigenfilter. Specifically, we considered all modes except the principal mode, demonstrating that these modes converge to zero asymptotically. This leaves us with the question: what happens to the principal mode? This is what we need to analyze next.

(Refer Slide Time: 01:16)



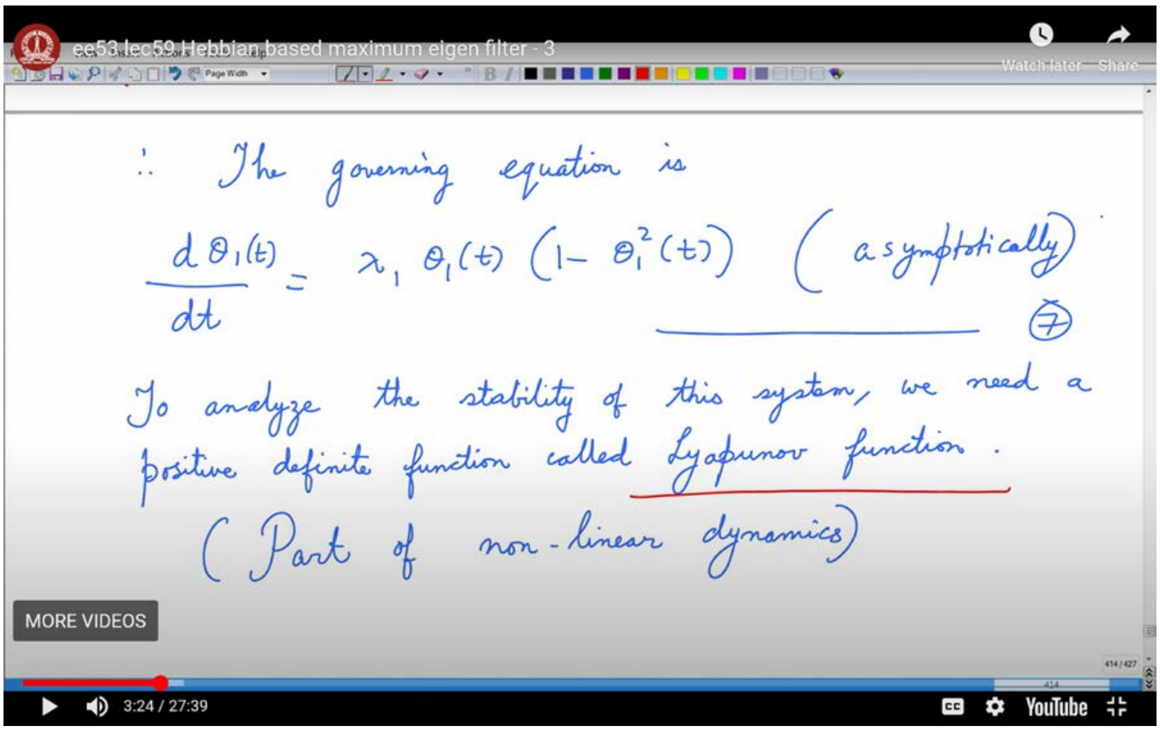
Let's focus on the case where k = 1. The governing differential equation for this scenario is:

$$\frac{d\theta_1(t)}{dt} = \lambda_1\theta_1(t) - \theta_1(t)\sum_{l=1}^{m}\lambda_l\theta_l^2(t)$$

Here, λ_i and θ_i represent the eigenvalues and eigenfunctions, respectively. Simplifying this equation, we get:

$$\frac{d\theta_1(t)}{dt} = \lambda_1\theta_1(t) - \theta_1(t)\sum_{l=2}^{m}\lambda_l\theta_l^2(t)$$

(Refer Slide Time: 03:24)

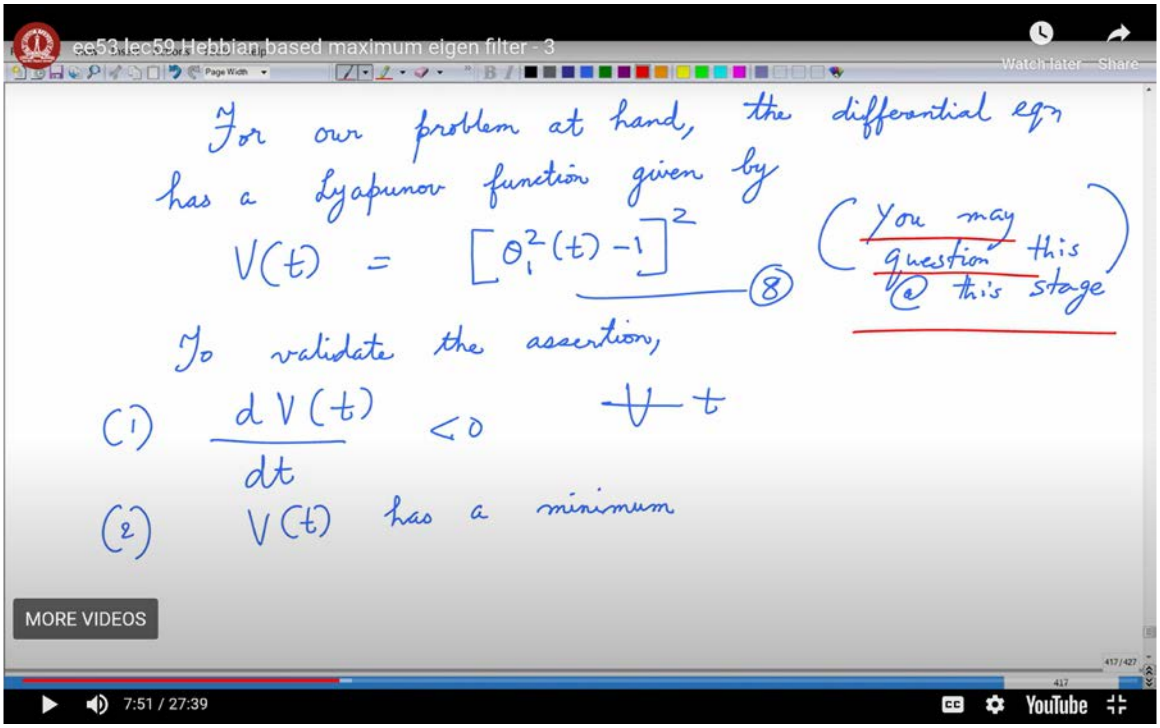


Since we know from Case 1 that α_l (for $l \neq 1$) approaches zero as t goes to infinity, all terms involving $\theta_l^2(t)$ for $l \neq 1$ vanish. Thus, our equation simplifies to:

$$\frac{d\theta_1(t)}{dt} = \lambda_1\theta_1(t)\left(1 - \theta_1^2(t)\right)$$

Now, to analyze the stability of this system, we need to use Lyapunov functions. For linear dynamical systems, stability is often assessed by examining the roots of the homogeneous equation. However, since we are dealing with a non-linear system, stability analysis is more complex and requires Lyapunov functions. These functions are positive definite and provide a way to assess the stability of non-linear systems.

(Refer Slide Time: 07:51)

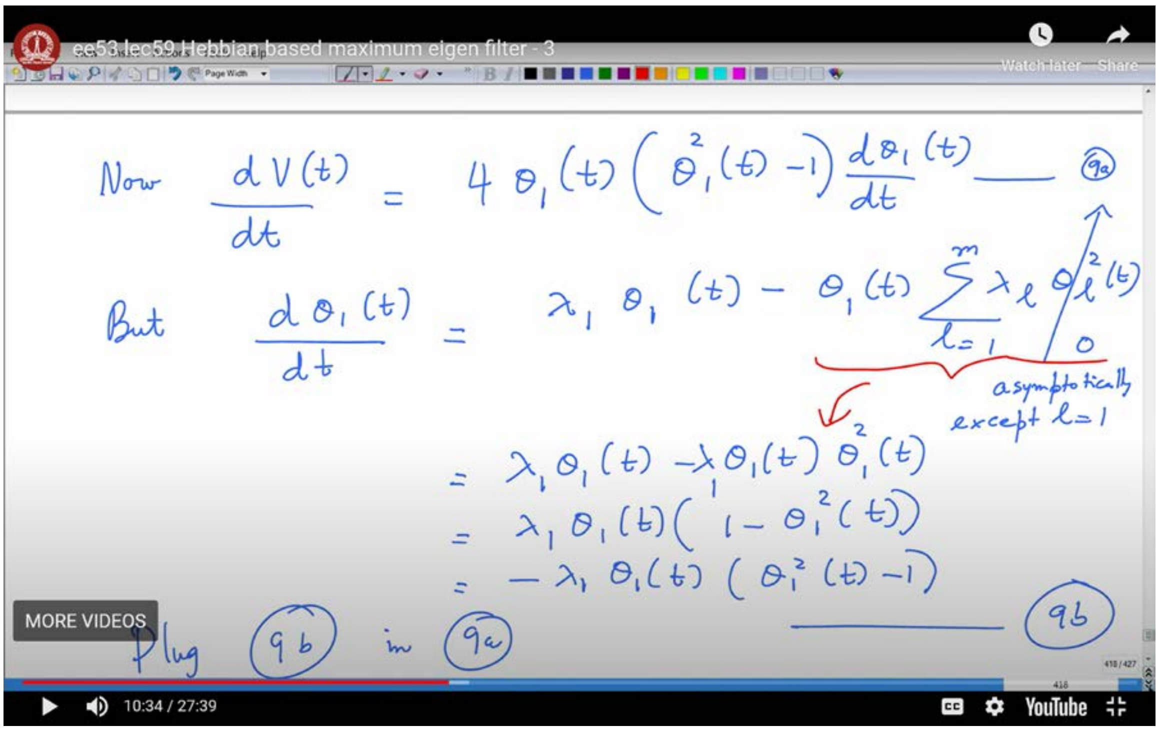


In this context, let S be the state vector of our autonomous non-linear system. The formal treatment of Lyapunov functions is covered in-depth in a graduate-level course on non-linear dynamics. For now, we will use a specific Lyapunov function relevant to our example and touch upon the necessary details.

Let V(t) be the Lyapunov function for the system. An equilibrium state, denoted by s_{eq}, is considered to be automatically stable if it satisfies the condition that $\frac{dV}{dt} < 0$ for all vectors s within a small neighborhood around s_{eq}, excluding the equilibrium point itself. Essentially, this means that the derivative of the Lyapunov function is negative in this neighborhood.

The challenge, however, lies in determining an appropriate Lyapunov function. Unlike linear systems, there is no universal Lyapunov function that works for all non-linear systems. Instead, finding a suitable Lyapunov function often relies on intuition and experience. Once a candidate Lyapunov function is identified, its validity must be tested to ensure it meets the necessary criteria for stability.

(Refer Slide Time: 10:34)



For the problem at hand, which involves analyzing the Hebbian-based maximum eigenfilter, we propose a Lyapunov function of the form:

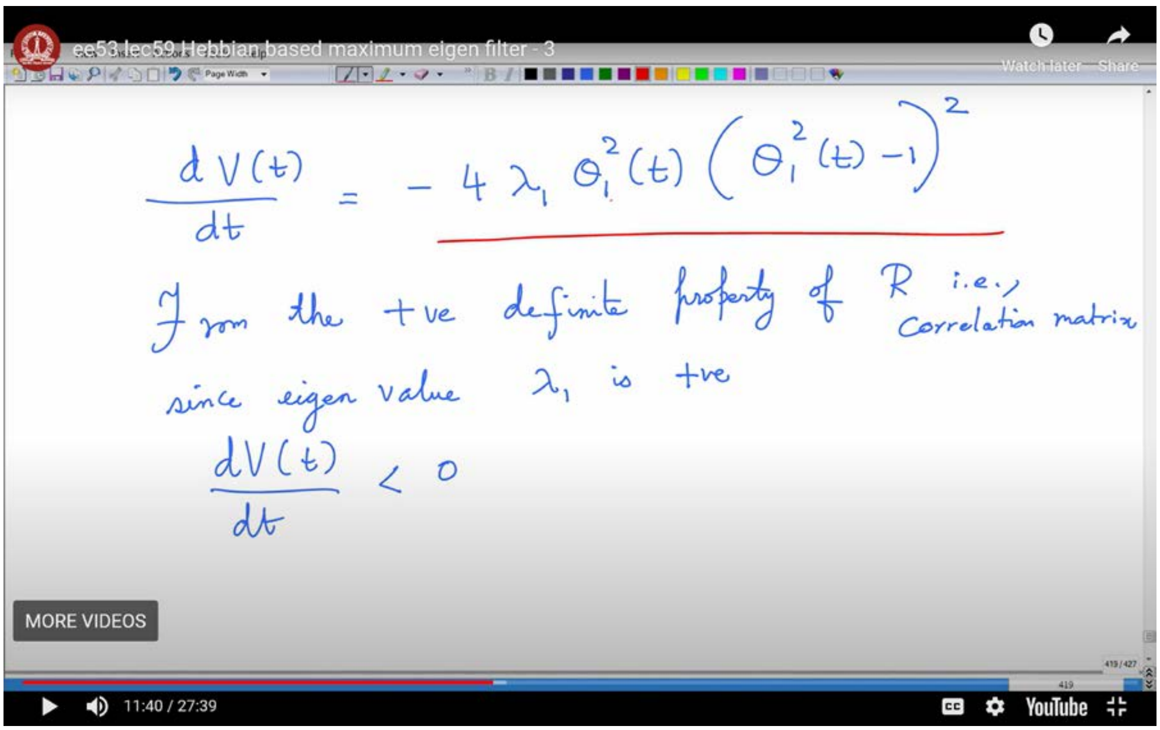
$$V(t) = (\theta_1^2(t) - 1)^2$$

At this point, you might wonder how this specific form for the Lyapunov function was chosen. As mentioned earlier, the choice of a Lyapunov function depends on the non-linear system under study. There is no one-size-fits-all function for all non-linear systems. In practice, one often relies on intuition to select a candidate function and then validates it against the required stability conditions.

The first condition to validate is whether $\frac{dV(t)}{dt} < 0$ for all t and whether V(t) has a minimum. To verify this, we compute the derivative of V(t). Given $V(t) = (\theta_1^2(t) - 1)^2$, we use the chain rule to find:

$$\frac{dV(t)}{dt} = 4(\theta_1^2(t) - 1) \cdot \theta_1(t) \cdot \frac{d\theta_1(t)}{dt}$$

(Refer Slide Time: 11:40)



Here, $\theta_1(t)$ is a function of time, so applying the chain rule involves differentiating $(\theta_1^2(t) - 1)$ with respect to $\theta_1(t)$, which gives 2 $\theta_1(t)$. Hence:

$$\frac{dV(t)}{dt} = 4(\theta_1^2(t) - 1) \cdot \theta_1(t) \cdot \frac{d\theta_1(t)}{dt}$$

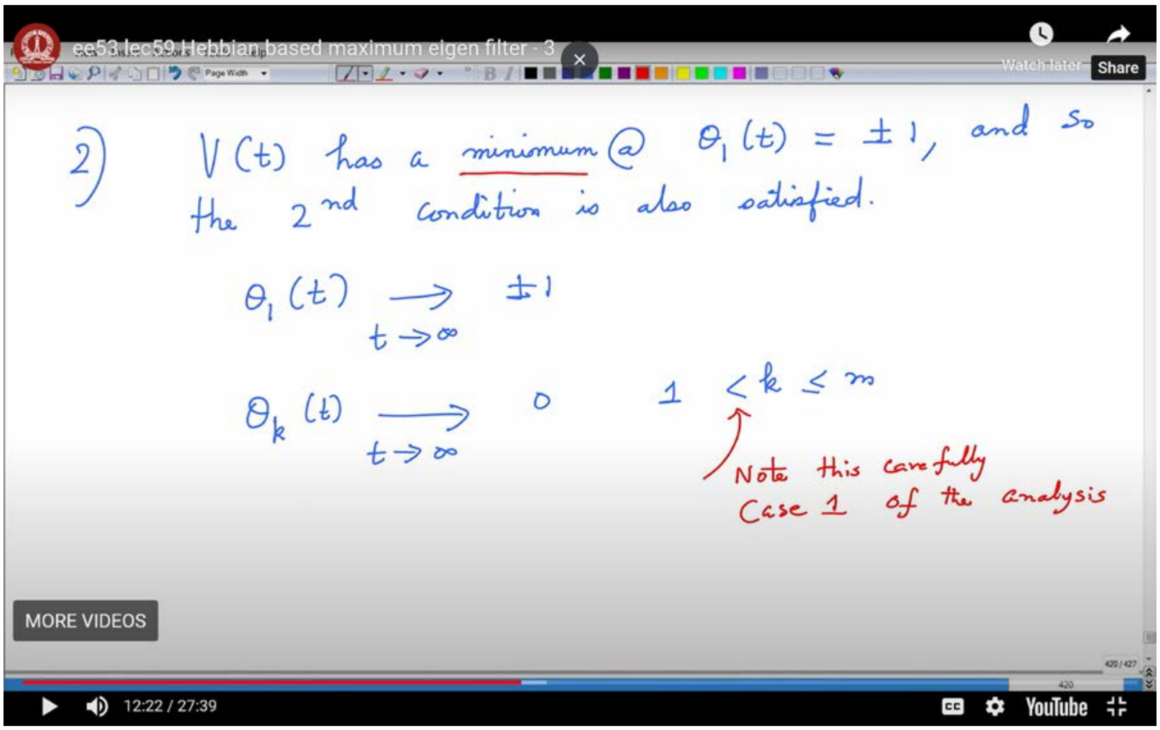
Next, we need the expression for $\frac{d\theta_1(t)}{dt}$, which we previously determined as:

$$\frac{d\theta_1(t)}{dt} = \lambda_1\theta_1(t) - \theta_1(t)\sum_{l=1}^{m} \lambda_l\theta_l^2(t)$$

From our earlier analysis, all $\theta_l(t)$ for $l \neq 1$ approach zero as t goes to infinity. Therefore, we only retain $\theta_1(t)$, and the summation terms for $l \neq 1$ drop out. This simplifies our expression to:

$$\frac{d\theta_1(t)}{dt} = \lambda_1\theta_1(t) - \lambda_1\theta_1(t)\theta_1^2(t)$$

(Refer Slide Time: 12:22)



Thus:

$$\frac{dV(t)}{dt} = 4(\theta_1^2(t) - 1) \cdot \theta_1(t) \cdot \left(\lambda_1\theta_1(t) - \lambda_1\theta_1^3(t)\right)$$

Simplifying further:

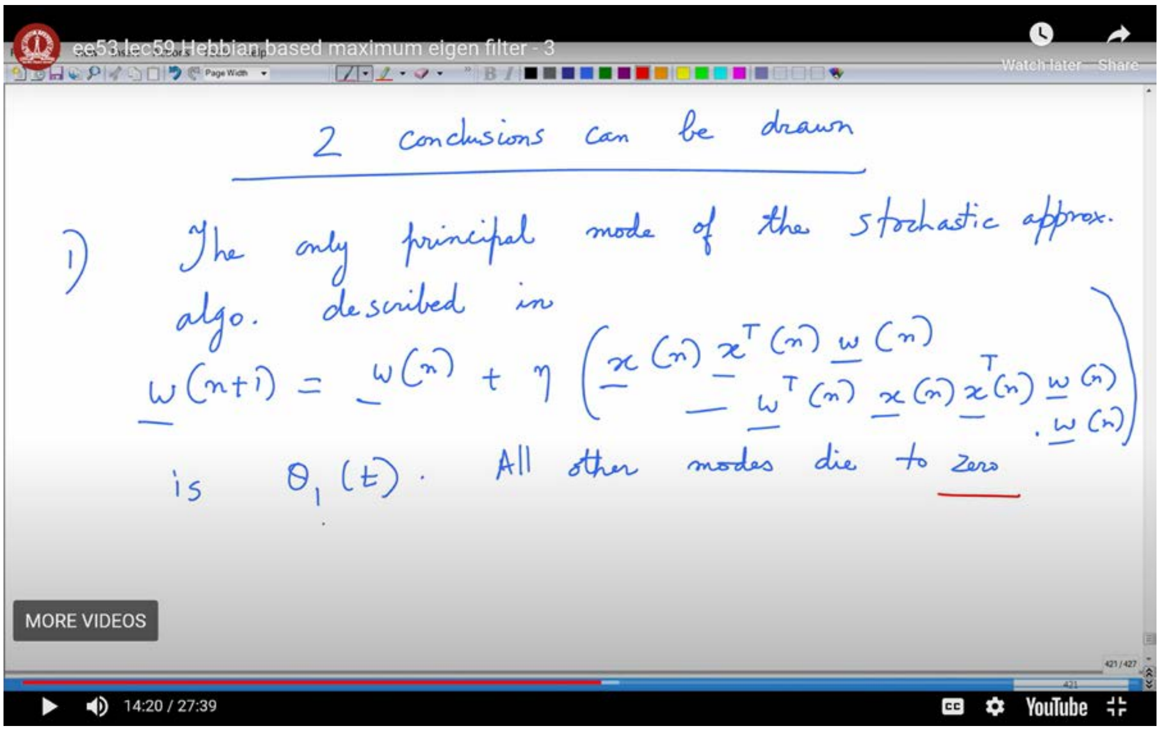
$$\frac{dV(t)}{dt} = -\lambda_1\theta_1(t)(\theta_1^2(t) - 1)^2$$

This final expression shows that $\frac{dV(t)}{dt}$ is negative as long as $\theta_1(t) \neq \pm 1$, confirming that V(t) satisfies the condition for stability in the vicinity of the equilibrium point.

In the previous discussion, we examined the differential equation $\frac{dV}{dt}$, which we referred to as equation 9B in our terminology. By substituting equation 9B into equation 9A and simplifying, we obtain:

$$\frac{dV}{dt} = -4\lambda_1 \theta_1^2(t)(\theta_1^2(t) - 1)^2$$

(Refer Slide Time: 14:20)

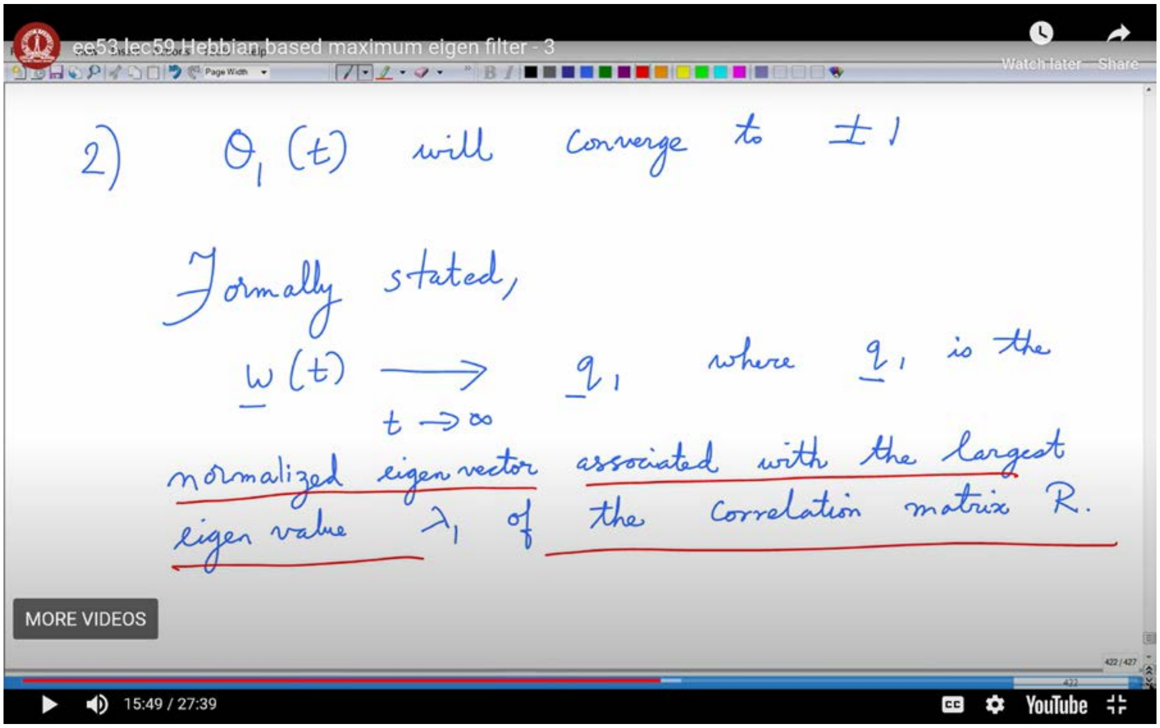


The data correlation matrix R is positive definite, which means that the eigenvalue λ_1, of particular interest here, is positive. We do not concern ourselves with the other eigencomponents, as all principal modes, except for the first mode, tend to zero. Consequently, since λ_1 is positive, $\frac{dV}{dt}$ is negative.

Examining the expression $\frac{dV}{dt}$, we see that it includes a negative sign and that all squared quantities, such as $\theta_1^2(t)$ and $(\theta_1^2(t) - 1)^2$, are positive. The only variable detail is λ_1, but due to the positive definite property, λ_1 is greater than zero. Therefore, all terms are positive except for the negative sign, confirming that $\frac{dV}{dt}$ is indeed less than zero.

The Lyapunov function V(t) has its minimum when $\theta_1(t)$ is $\pm$ 1. This scenario resembles a quadratic bowl where the saddle points are at $\theta_1 = 0$ and $\theta_1 = \pm 1$. We are particularly interested in the cases where θ_1 is ± 1 and ignore the zero case, as it is less significant. This analysis shows that V(t) has a minimum at these values.

(Refer Slide Time: 15:49)

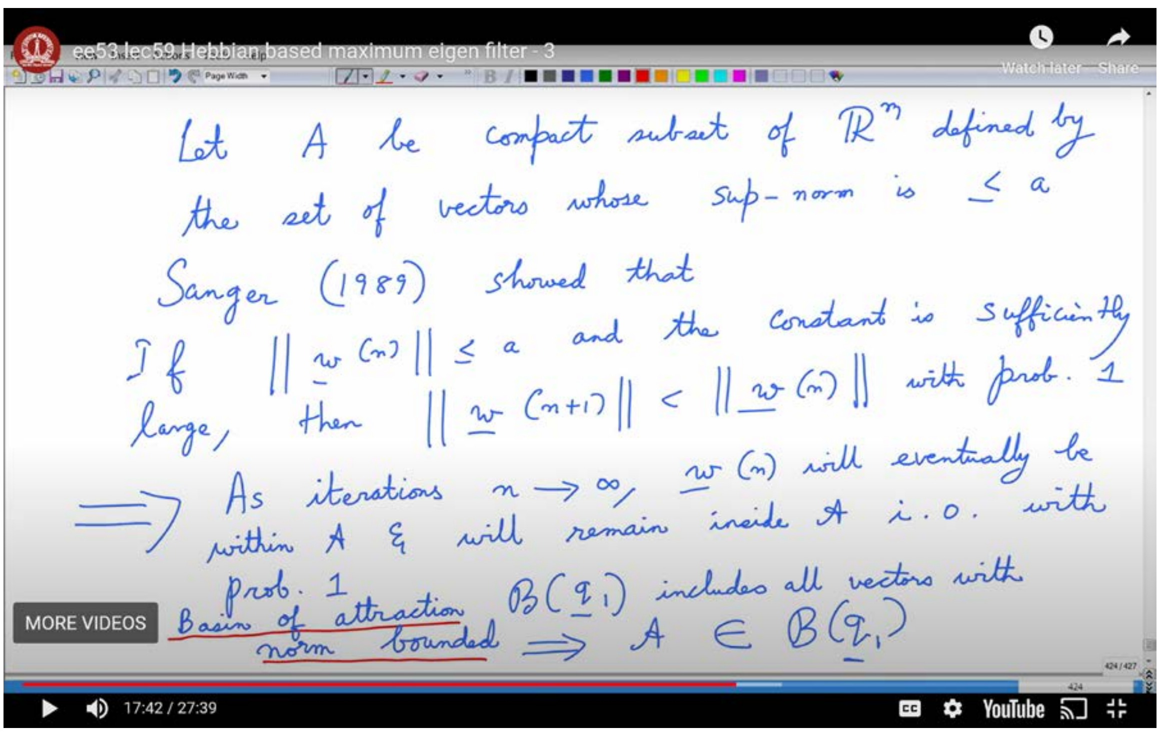


As a result, as t approaches infinity, $\theta_1(t)$ converges to ± 1, while all other modes $\theta_k(t)$ tend to zero. This indicates that the first component remains significant, and $\theta_1(t)$ approaches ± 1.

In practical terms, this implies that the maximum Hebbian-based maximum eigenfilter effectively extracts the first principal component. By analyzing the data correlation matrix and determining the eigenvalues and eigenvectors, we find that the algorithm, Oya's rule, essentially a normalized Hebbian update, adaptively extracts the first principal component. Thus, from the complex stochastic differential equation we started with, we see that $\theta_1(t)$ converges to ± 1, and the weight vector W(t) asymptotically converges to Q_1, where Q_1 is the normalized eigenvector associated with the largest eigenvalue λ_1 of the correlation matrix R. Feel free to pause and reflect on this outcome.

In our previous discussion, we covered the Hebbian update rule and its implications. Essentially, the Hebbian update involves adjusting the weight vector at time n+1 based on the weight vector at time n, with an additional term that is η times the product of the output and the input. Here, the input is a vector representing the presynaptic input, while y(n) is a scalar representing the postsynaptic output. This straightforward update, when normalized, leads the weight vector to converge to the normalized eigenvector corresponding to the largest eigenvalue of the data correlation matrix.

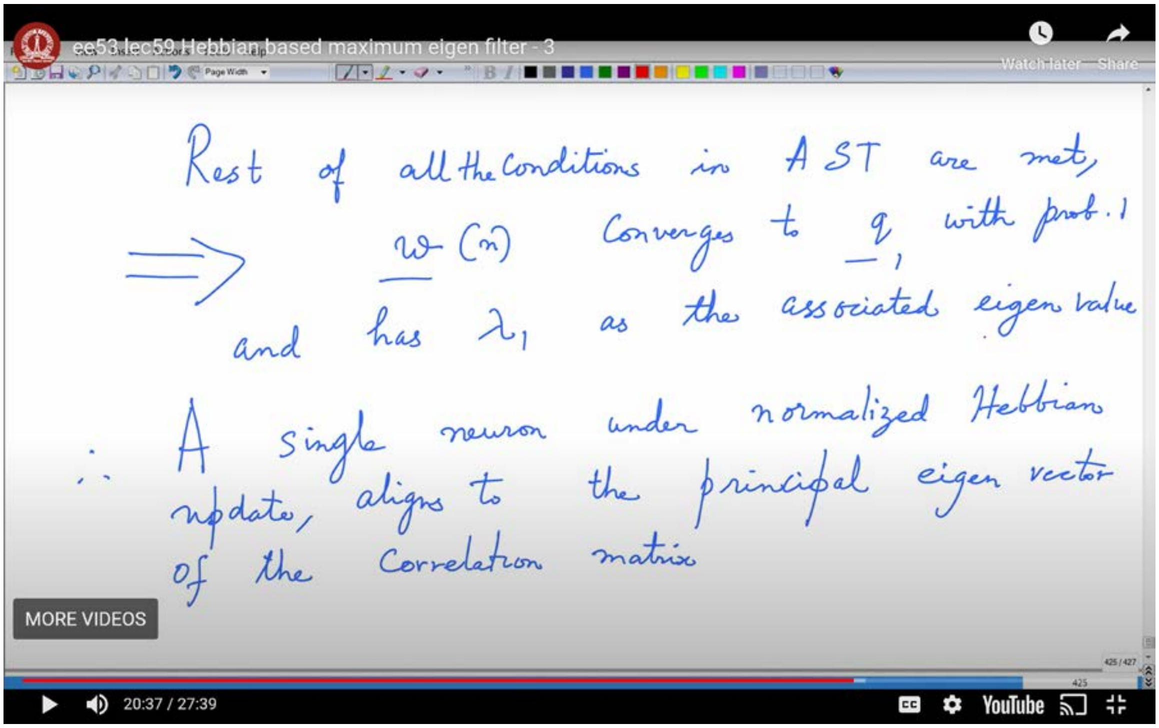
(Refer Slide Time: 17:42)



From condition 6 of the asymptotic stability theorem, which we previously discussed, we know that there exists a subset A of the set of vectors such that the weight vector converges to the eigenvector q_1 with probability 1, infinitely often. In other words, the trajectories of all these points will, with probability 1, eventually enter a compact set, or the basin of attraction, associated with this eigenvector.

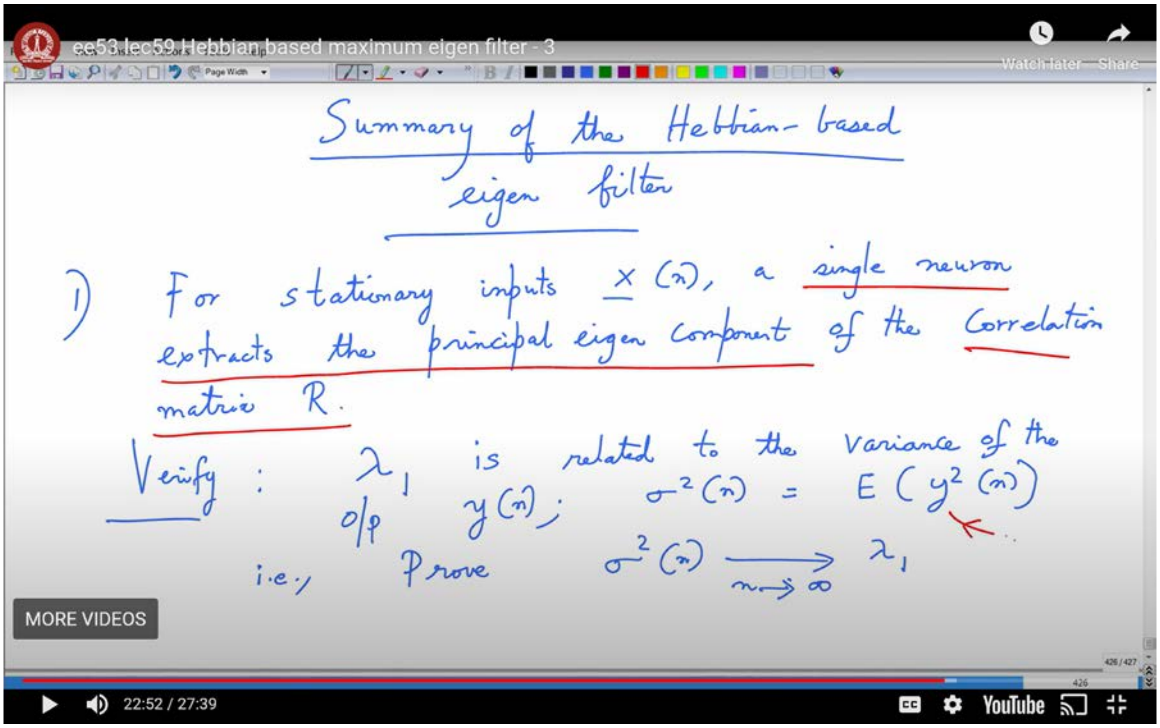
To satisfy condition 2 of the asymptotic stability theorem, normalization is required. Specifically, we implement a hard limit on the entries of the weight vector, ensuring that their magnitudes stay below a threshold A. This is expressed as $|W(n)| \leq A$, where A is a constant and $|\cdot|$ denotes the norm of the weight vector. Sanger demonstrated in 1989 that if this norm is bounded by A and A is sufficiently large, then with probability 1, the norm of the weight vector at time n+1 will be less than the norm at time n. This crucial result implies that, with a large number of iterations, the weight vector W(n) will almost certainly remain within this compact subset infinitely often.

(Refer Slide Time: 20:37)



This basin of attraction includes all vectors that are bounded, and this implies that the compact subset A is part of this basin. This is particularly important because, unlike linear dynamical systems, where eigenmodes are scalar and not time-varying, non-linear dynamics involve attractor points. Tracking the trajectory of the weight vector will show that it enters the basin of attraction and converges to the attractor. Here, the basin of attraction around the eigenvector q_1 includes all bounded vectors, indicating that A is a subset of this basin.

(Refer Slide Time: 22:52)

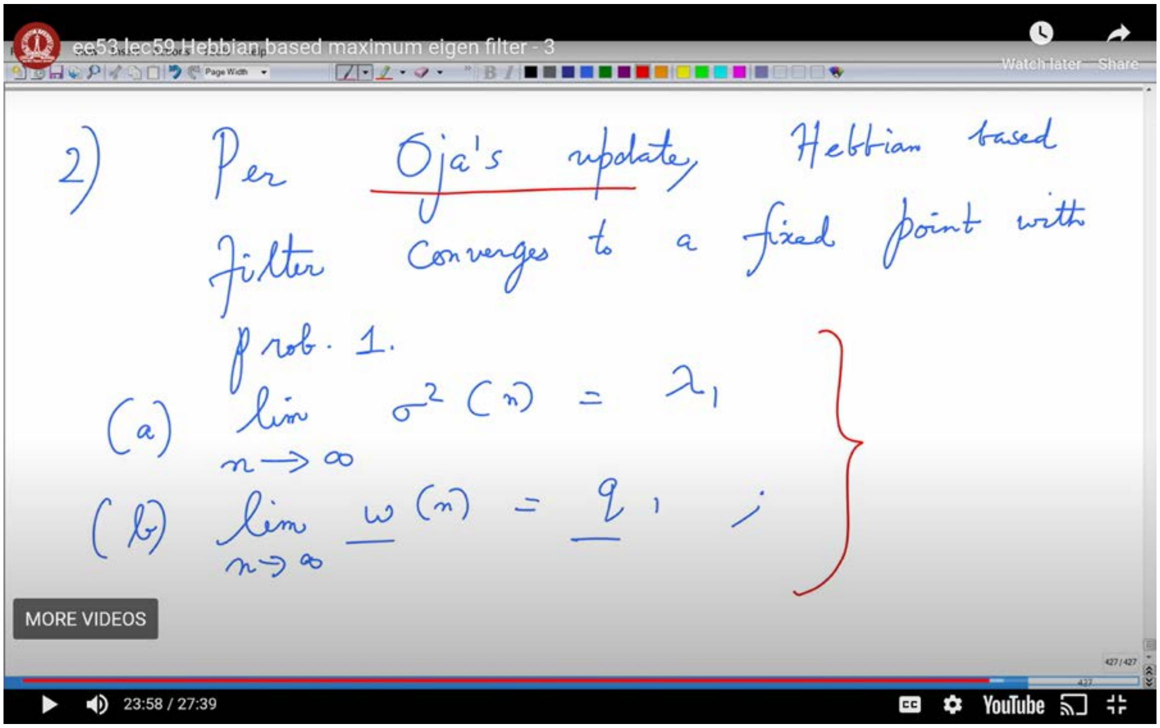


Verifying that all conditions of the asymptotic stability theorem are met confirms that the weight vector W(n) converges to q_1 with probability 1, with λ_1 as the associated eigenvalue. The key takeaway is that a single neuron with a normalized Hebbian update aligns with the principal eigenvector of the correlation matrix. In other words, it adaptively extracts the principal component of the matrix, as the weight vector converges to this eigenvector. The eigenvalue is related to the variance σ^2, which is connected to the expected value of the square of the output.

To summarize, we began with stationary inputs, meaning the statistical properties of the data remain constant over time, a crucial assumption for this analysis.

Otherwise, we would encounter time-varying correlation matrices, making the analysis significantly more complex. For stationary inputs x_n, a single neuron effectively extracts the principal eigencomponent of the correlation matrix R. As an exercise, I suggest considering how the variance of the output y_n, denoted σ^2_n, approaches λ_1 as n tends to infinity. Here, σ^2_n represents the variance of the output y_n, which is linked to both the input and the weight vector through their inner product.

(Refer Slide Time: 23:58)



In this context, W_n is a scalar, which is crucial. The expectation is taken over all realizations of the stochastic data vector. According to Oya's update rule, which normalizes the Hebbian-based rule, the Hebbian filter converges to a fixed point with probability 1. This results in two main observations: the largest eigenvalue λ_1 corresponds to the limit of σ^2_n as n approaches infinity, and the eigenvector associated with λ_1 is the asymptotic direction of the weight vector.

In summary, the Hebbian-based maximum eigenfilter, which operates through simple correlation-based updates, achieves a stable solution. The process involves a presynaptic input (the data vector), a postsynaptic output (the neuron's output), and a weight vector that couples the input with the neuron’s output. By applying the normalized Hebbian update rule, which adapts the weights, the solution stabilizes. We further explored this stability by expanding the normalization using a Taylor series, leading to a nonlinear difference equation. This was then related to a differential equation, and we analyzed the eigenmodes of the nonlinear system.

We examined two cases: one focusing on the principal mode and the other on the remaining modes not associated with k = 1. We found that all modes other than the dominant one decay to zero, leaving the dominant mode governed by nonlinear dynamics. Our analysis confirmed that this dominant mode represents a stable attractor.

Thus, adaptively, the weight vectors converge to the principal eigenvector of the data correlation matrix, and λ_1 becomes the variance of the neuron's output. To conclude, this study demonstrates that while traditional principal component analysis and KL transforms rely on linear methods, our approach delves into nonlinear versions, including Kernel PCA and Hebbian-based maximum eigenfilters.