



FREE PROJECT REPORT - M2 COMPUPHYS

Kuramoto Model

Niels FAUCHER

Under the supervision of Pr. Daniel BRUNNER,
Phd student Jacob PUHALO-SMITH and Dr. Anas SKALLI from the
FEMTO-ST Institute

Contents

1	Introduction	2
2	Alpha version	3
2.1	Basic Kuramoto model	3
2.2	Tools of analyze	4
2.3	Validation of the Basic Model	5
2.4	Upgraded Kuramoto model	6
2.4.1	Coupling Matrix K_{ij}	6
2.4.2	Adjacency Matrix A_{ij}	7
2.4.3	Interaction Matrix $\sin(\theta_j - \theta_i)$	7
2.5	Validation of the New Model	7
2.6	Conclusion Alpha Version	8
3	Beta Version	8
3.1	Testing the Parameters	8
3.1.1	Results Interpretation	11
3.2	Implementation of a Input	12
3.2.1	Equation of the Kuramoto model with a input	12
3.2.2	Validation of our Input Implementation	13
3.3	Conclusion Beta version	15
4	Discussion	16
A	Appendix	17
A.1	Validation of the Basic Model	17
References		20

Abstract

This report presents my work on the Kuramoto model, carried out as part of the Unite Free project. Guided by Daniel Brunner, a leading researcher in photonic AI and optical neural networks, I explored how this model can be used to study synchronization dynamics in coupled systems, particularly in vertical-cavity surface-emitting lasers (VCSELs). These lasers are key components in optical neural networks, and understanding their behavior under optical injection is essential for optimizing their performance. My project was divided into two main parts. First, I built and tested a basic Kuramoto model to gain insight into its behavior. Then, I extended the model to investigate how system parameters, external inputs, and coupling conditions influence synchronization. This work complements the efforts of a colleague working on training physical neural networks, as my simulations provide a numerical perspective on how these systems behave. Overall, this project contributes to a better understanding of synchronization in photonic systems and their potential in neural network applications.

Keywords : Kuramoto model, Photonic AI, Neural network, VCSELs, Coupled oscillator, Synchronization

1 Introduction

Understanding the dynamics of synchronization in coupled systems is a fundamental challenge in many fields of science and engineering. The Kuramoto model, introduced by Yoshiki Kuramoto in the 1970s, is a mathematical framework that provides deep insights into this phenomenon. It describes the collective behavior of coupled nonlinear oscillators and offers a simplified yet powerful approach to studying synchronization. Oscillatory systems appear in diverse contexts, from biological systems like neural networks and circadian rhythms to physical systems such as power grids and coupled laser arrays. The Kuramoto model, with its versatility and mathematical elegance, has become a cornerstone for analyzing these phenomena.

Despite its simplicity, the Kuramoto model continues to raise important questions. How do system parameters, such as coupling strength or the natural frequency distribution, influence the onset of synchronization? How do external inputs or noise affect the collective behavior of oscillators? Can the model be extended to describe more complex physical systems, such as coupled lasers or optical neural networks? Answering these questions requires not only theoretical insights but also numerical simulations and experimental validation.

One area where this model has particular relevance is in the emerging field of optical neural networks (ONNs) [Skalli et al.(2022)Skalli, Porte, Haghghi, Reitzenstein, Lott, and Brunner]. These systems leverage the speed, scalability, and energy efficiency of photonic hardware to implement artificial neural networks. Vertical-cavity surface-emitting lasers (VCSELs) [Skalli et al.(2022)Skalli, Porte, Haghghi, Reitzenstein, Lott, and Brunner] are promising candidates for ONNs due to their nonlinear dynamics and ability to operate in high-speed, parallel configurations. Understanding the synchronization behavior of coupled VCSELs under optical injection is crucial for optimizing their performance in neuromorphic computing tasks, such as pattern recognition.

In this report, we explore the Kuramoto model and its application to simulating coupled laser systems, specifically focusing on VCSELs. By examining the synchronization dynamics of these systems, we aim to bridge theoretical modeling with practical experimentation, providing insights into how physical networks achieve collective behavior. Our work not only explores fundamental questions about the Kuramoto model but also contributes to the development of scalable, high-performance optical neural networks.

The reader can access all the code used for plotting and analyzing the system in the "Technical Report," which provides a detailed explanation of how our code functions.

2 Alpha version

In this section, we will first introduce the basic Kuramoto model, the tools used to analyze the system's behavior, and some initial results. Then, we will explore a more realistic version of the model and examine how certain system parameters influence its behavior.

2.1 Basic Kuramoto model

The most basic is described as following : The model is described as a ensemble of N oscillators where the phases θ_i (avec $i = 1, 2, \dots, N$) evolve in time following this equation [Cumin and Unsworth(2022)]:

$$\frac{d\theta_i}{dt} = \omega_i + \frac{K}{N} \sum_{j=1}^N \sin(\theta_j - \theta_i), \quad i = 1 \dots N \quad (1)$$

where :

- $d\theta_i/dt$ represents the rate of change of phase of the oscillator i
- $\theta_j - \theta_i$ is the phase difference between oscillator j and i ,
- ω_i is the natural frequency of the oscillator i ,
- K is the coupling strength ,
- N is the numbers of oscillators.

Here is a 2D representation of a possible architecture of the model that represent the inter-connectivity between all the oscillators.

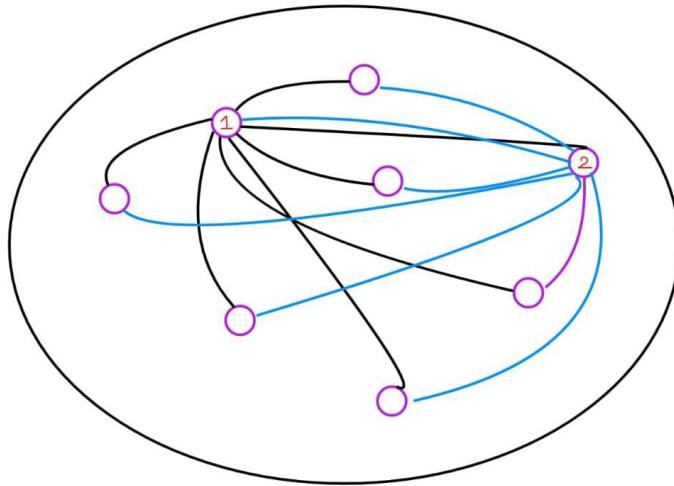


Figure 1: Possible architecture of the Oscillator Neural Network

This representation provides a simple illustration of the connectivity between the nodes in our system. For clarity, we have chosen to show only the connection between nodes 1 and 2. In Eq. 1, the term $\sin(\theta_j - \theta_i)$ represents the interactions between all nodes, while \mathbf{K} denotes the strength of each connection (in this case, all connections share the same strength).

2.2 Tools of analyze

As we define our base model, it is now necessary to introduce the tools that will help us analyze the results and understand the behavior of the system. One of the most important and interesting tools we use is the **phase coherence**. While it is not the only metric we consider—for example, we also plot the $\sin(\theta_j - \theta_i)$ over time—our primary focus will be on the coherence.

The phase coherence, often denoted by r , is defined as:

$$r = \frac{1}{N} \left| \sum_{j=1}^N e^{i\theta_j} \right| \quad (2)$$

where where:

- N is the total number of nodes (oscillators) in the system,
- θ_j represents the phase of the j -th oscillator,
- $e^{i\theta_j}$ maps the phase θ_j onto the unit circle in the complex plane, and
- $r \in [0, 1]$, with $r = 1$ indicating perfect synchronization (all oscillators are in phase) and $r = 0$ representing completely uncorrelated phases.

The coherence r provides a quantitative measure of the degree of synchronization in the system. It allows us to track how the phases of the oscillators evolve over time and how strongly they tend to align. This is particularly useful in systems like ours, where synchronization plays a key role in understanding the interactions between nodes.

In addition to phase coherence, we have other methods to analyze and verify the synchronization of our system. Although these methods are generally less precise in quantitatively measuring synchronization, they can provide valuable complementary insights into the dynamics of the oscillators.

For instance, in our simulation, we calculate **the vector sum of the $\sin(\theta)$ values** for all oscillators at each time step. This is defined as:

$$\text{Vector Sum} = \sum_{i=1}^N \sin(\theta_i), \quad (3)$$

where N is the total number of oscillators and θ_i represents the phase of the i -th oscillator. This measure gives a quick, qualitative sense of how the oscillators are behaving collectively. A large value indicates that many oscillators are roughly aligned in phase, while smaller values suggest more desynchronization.

We also analyze **the angular frequency dynamics** by computing the time derivative of the phases:

$$\frac{d\theta}{dt} = \frac{\theta(t + \Delta t) - \theta(t)}{\Delta t}. \quad (4)$$

This approach enables us to visualize the temporal evolution of the angular velocities of the oscillators. Although it is less focused on quantifying synchronization directly, it can reveal interesting patterns such as transitions between synchronized and desynchronized states or fluctuations in the system.

These additional tools also allow us to investigate in more detail how specific parameters impact the system. For example, by plotting the vector sum and angular frequency dynamics, we can observe the effects of variations in the standard deviation of the natural angular frequencies or changes in the coupling strength α . These graphs provide a finer understanding of how certain parameters influence the system's behavior, especially in cases where the global coherence metric might not capture all subtleties.

While phase coherence remains our primary tool for assessing synchronization, these additional metrics provide complementary information that can highlight subtle dynamics, transient behaviors, or local interactions within the system. Together, these tools enhance our understanding of the system's complexity and the factors influencing synchronization.

2.3 Validation of the Basic Model

In this section, we present the initial plots of our system, these plots correspond to the metrics and tools discussed above (sum of the sin, angular frequency and coherence), giving us a general idea of what we can expect from the Kuramoto model. In order to prove the reliability of our simulation we should try with different values of the coupling constant so **we tested 3 values of K [0.5, 1, 3]**. For the seek of clarity we will represent only the coherence of our system, but the reader can find all the necessary plot in the appendix.

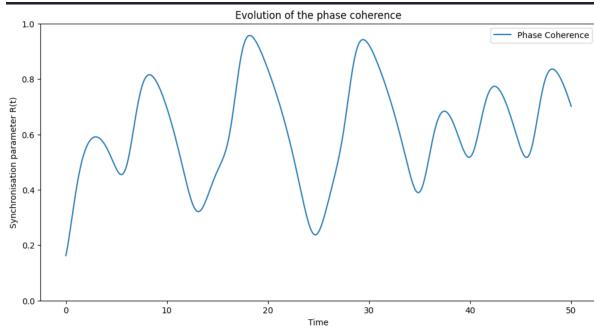


Figure 2: Coherence vs time for $\mathbf{K = 0.5}$

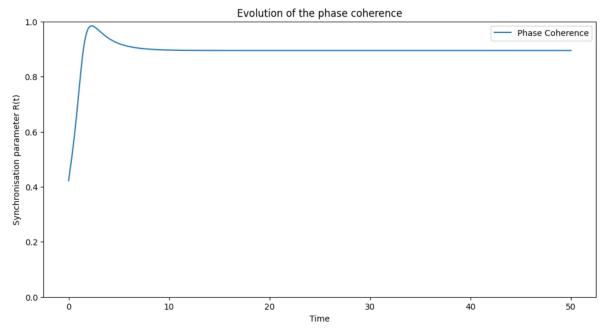


Figure 3: Coherence vs time for $\mathbf{K = 1}$

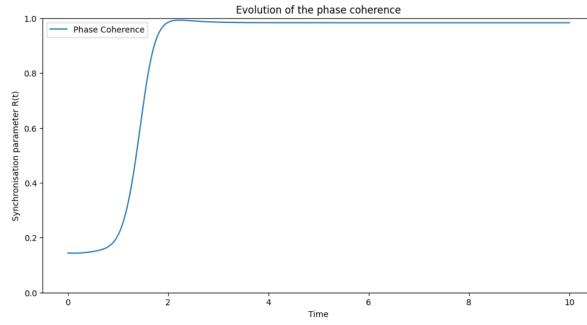


Figure 4: Coherence vs time for $\mathbf{K = 3}$

For the simulations, we used the following parameters that were chosen in an empirical way:

Parameters	Value
Numbers of nodes	5
Total time of the integration/simulation	[10, 50, 50]
Step time	0.01
Scaling factor / overall strength	[0.5, 1, 3]
Mean and std of the natural frequency	(1, 0.5)
Adjacency matrix	0.9

Table 1: Parameters for the testing of the basic model

As a first deduction, we can say that my initial results are quite promising, as they already demonstrate some expected behaviors of the Kuramoto model. In particular, **the phenomenon of synchronization depending on the K parameters** (alpha in the code) is clearly visible in all three plots. These initial findings suggest that the numerical simulations are working well and that the parameters we chose are appropriate for capturing the key features of the system. These will serve as a useful reference for our future work as we improve the model and refine the system.

2.4 Upgraded Kuramoto model

In this section, we introduce an upgraded version of the Kuramoto model, which includes several modifications that improve the realism and accuracy of our simulations. These upgrades involve adjusting the coupling and strength dynamics, introducing randomness in the connectivity between oscillators, and the coupling. We will present a detailed breakdown of the model's governing equation.

Governing Equation In our upgraded version, we introduce a coupling matrix K_{ij} , making the model more realistic by allowing the strength between oscillators to vary. The equation then becomes:

$$\frac{d\theta_i}{dt} = \omega_i + \alpha \sum_{j=1}^N K_{ij} A_{ij} \sin(\theta_j - \theta_i), \quad i = 1 \dots N \quad (5)$$

Where:

- α is a scaling factor that controls the overall coupling strength,
- A_{ij} is the adjacency matrix that dictates which oscillators are coupled, in the basic model we defined all the matrix as 1.
- K_{ij} represents the coupling matrix, which defines the interaction strength between the oscillators.

Let's now break down the three main parts of our new model :

2.4.1 Coupling Matrix K_{ij}

The coupling matrix K defines the interaction strength between oscillators and is represented as a square matrix. In our setup, the off-diagonal elements K_{ij} (for $i \neq j$) are assigned random values drawn from a uniform distribution in the range [0, 1], while the diagonal elements K_{ii} are fixed to 1, reflecting the self-coupling of each oscillator. To ensure system stability and properly scaled coupling dynamics during simulations, the matrix K is normalized by dividing all its elements by its largest eigenvalue. This normalization is a crucial step to maintain the numerical robustness and physical relevance of the model.

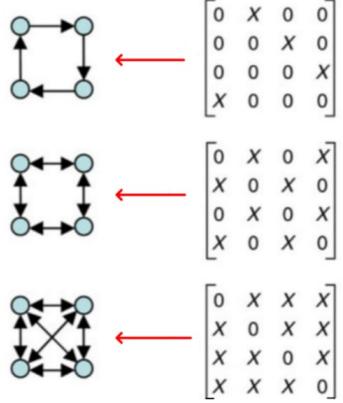


Figure 5: Representation of varying connectivity based on the percentage of matrix filling [Cumin and Unsworth(2022)]

2.4.2 Adjacency Matrix A_{ij}

The adjacency matrix A encodes the connectivity structure between oscillators, where A_{ij} indicates the presence (1) or absence (0) of a connection between oscillators i and j . This square matrix is generated probabilistically, with each off-diagonal element randomly assigned a value of 0 or 1. The diagonal elements are set to 1, representing self-connections. The purpose of A is to modulate the interaction dynamics by multiplying it element-wise with the interaction matrix and the term $\sin(\theta_j - \theta_i)$. This approach introduces randomly generated, non-symmetric connections, whose values are inherently random due to the probabilistic generation method. Here is a good way to represent it :

2.4.3 Interaction Matrix $\sin(\theta_j - \theta_i)$

The interaction matrix M_{ij} in the Kuramoto model characterizes the dynamic influence of phase differences between oscillators. Defined as $M_{ij} = \sin(\theta_j - \theta_i)$, it encapsulates how the relative phase difference $\theta_j - \theta_i$ modulates the interaction between oscillators i and j . This matrix evolves continuously throughout the simulation, adapting to changes in the phases θ_i and θ_j , thereby dynamically updating the network's interaction structure.

The diagonal elements of M are identically zero, as $\sin(\theta_i - \theta_i) = 1$, indicating the self-interaction. The off-diagonal elements represent the interaction strength and its nature: positive values signify attractive interactions, while negative values correspond to repulsive ones. For a network comprising n oscillators, M is an $n \times n$ matrix that symmetrically encodes the pairwise phase relationships, reflecting the dynamic interplay among the oscillators in the system.

2.5 Validation of the New Model

To validate the new model, we used the same testing parameters as in the initial version. However, we varied α to observe its effect on the system dynamics.

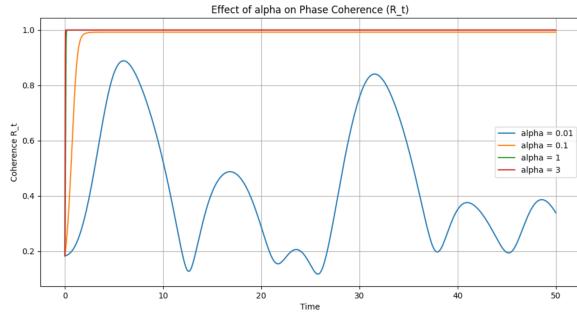


Figure 6: Coherence vs time, **Time = 50**
The above graphs are the coherence of the upgraded Kuramoto model with different K (alpha),
 $K = [0.01, 0.1, 1, 3]$

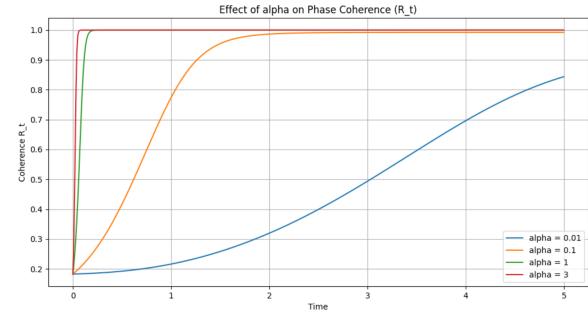


Figure 7: Coherence vs time - Zoom, **Time = 5**

To examine the influence of different values of α , we ran simulations using the same random seed (via `np.random.seed`) to ensure consistency. We tested the model across different time scales, starting with very small values of α to induce chaotic behavior. Even under these conditions, the system exhibited the expected transition from chaos to synchronization, confirming the validity of our upgraded equation.

Introducing this new equation naturally leads us to the next stage of analysis: investigating how all parameters influence the system's dynamics. With this modification, we effectively increase the "dimensions of control" in the model, providing a more flexible framework to explore complex behaviors.

2.6 Conclusion Alpha Version

In the alpha version, we defined the foundational equation as a starting point before transitioning to an upgraded, more realistic version of the model. Throughout the implementation, the system consistently demonstrated behavior aligned with the expected dynamics of the Kuramoto model.

We emphasize that the parameters used during these tests were carefully selected to validate both the correctness of our implementation and the robustness of the model itself. Up to this stage, our system has performed as expected, providing confidence to proceed to the next phase of development.

3 Beta Version

In this section, we address two primary objectives. The first is a natural continuation of the alpha version, where we test the updated system under varying parameter values to evaluate its robustness and behavior. The second objective involves integrating an external input into the system and analyzing the phase-locking correlation as a function of the coupling strength (β) and the input frequency.

3.1 Testing the Parameters

To precisely assess the influence of specific parameters, we define a baseline set of parameters that will serve as a reference. This ensures that any observed changes in the system dynamics are attributed solely to the variation of the studied parameters, rather than unintended interactions with others.

Two key metrics will be used to quantify differences during these tests: the synchronization time and the degree of synchronization.

Let us begin by establishing **the baseline parameter set**. For the number of nodes, we will maintain a value of 5. The integration time will be adjusted as needed to ensure optimal graph representation and effective analysis,

while the time step (Δt) will remain constant. Regarding the standard deviation (std) of the natural frequencies, we previously used a value of approximately 0.5. However, since our goal is to simulate the physical phenomenon of coupled lasers, we need to align with the characteristics of this system, where the std of the frequencies is typically no more than 10% of the mean. Thus, we will set the std to 0.01 to reflect these physical constraints. For the adjacency matrix, we will use a connectivity probability of 0.9, meaning 90% of the matrix elements will be filled with 1. Finally, the value of α will be set to 0.1 to observe the phase transition from chaotic to ordered behavior.

Parameters	Value
Numbers of nodes	5
Total time of the integration/simulation	50
Step time	0.01
Scaling factor / overall strength	0.1
Mean and std of the natural frequency	(1, 0.01)
Adjacency matrix	0.9

Table 2: Base set of parameters for the testing with the upgraded model

Regarding the parameters we are varying, we use a straightforward yet effective method to explore their influence on the system. As mentioned earlier, in our simulations, coherence is achieved 99% of the time. However, the key difference lies in the time it takes to reach this coherence. To investigate this, we run simulations while systematically varying the selected parameters and record the time required to achieve a coherence level of 90%.

Fitting the Results to Models: To analyze the relationship between each parameter and the time to reach coherence, we fitted the experimental data to different models:

- A power-law model ($y = a \cdot x^b$) was tested for N nodes and alpha, as these parameters showed non-linear dependencies on a log-log scale.
- A linear model ($y = a \cdot x + b$) was applied for matrix_adj, as the relationship appeared approximately linear. The fitting process was performed using the `curve_fit` function from `scipy`, and the results were plotted along with the experimental data. This approach allows us to directly observe how each parameter impacts the system's dynamics.

Below are the results:

- For the numbers of nodes & the scaling factor:

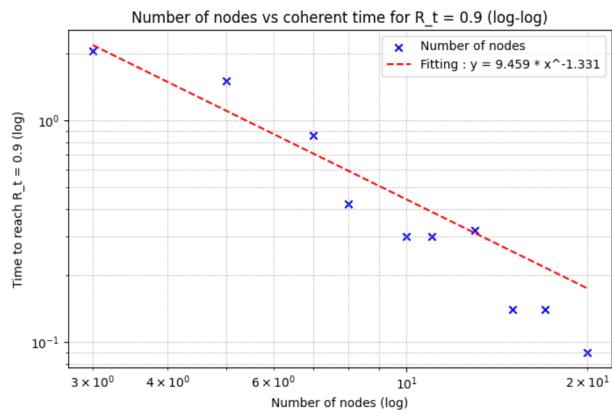


Figure 8: Number of Nodes vs Coherent time

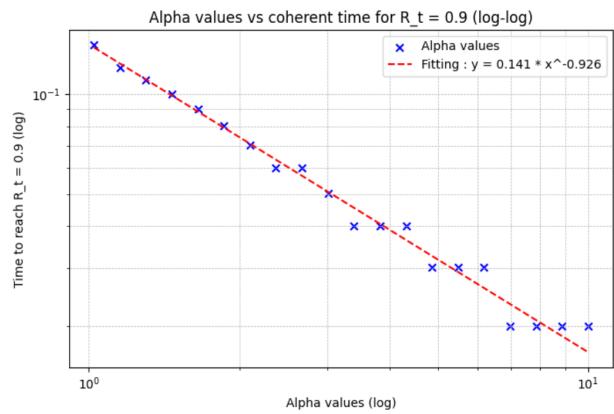


Figure 9: Alpha values vs Coherent time

- For the adjacency matrix & the std of the natural frequency :

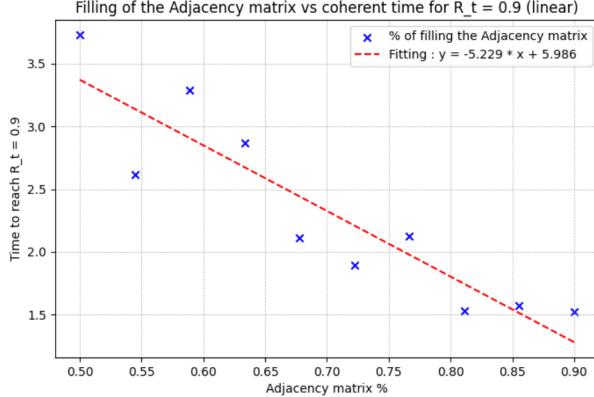


Figure 10: Percentage of matrix filling vs Coherent time

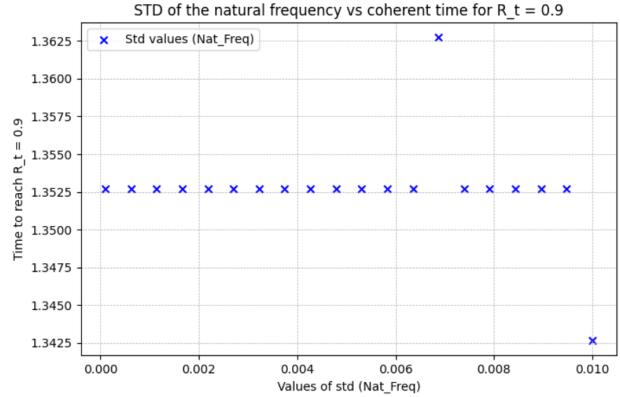


Figure 11: Std of the Natural Frequency vs Coherent time

3.1.1 Results Interpretation

Our results highlight several interesting features. First, we observe that both the number of nodes and the scaling factor (α) follow a power-law relationship (represented in a log-log graph), while the adjacency matrix density shows a linear fit. Similarly, the standard deviation of natural frequencies ω_i exhibits a linear behavior, but with a notable exception: for our chosen parameters, once the standard deviation exceeds 10% of the mean, it no longer significantly affects the system's behavior in terms of the time to reach coherence. This suggests a saturation effect for this parameter.

For the linear fit of the adjacency matrix density, the results clearly show that increasing the percentage of matrix connections reduces the time required to reach coherence. This is expected, as higher connectivity improves the synchronization of oscillators, facilitating faster convergence to a coherent state.

Regarding the variables that follow a power law, we know that such behavior indicates scale invariance, meaning the system's dynamics are consistent across different scales. On a log-log plot, this relationship appears as a straight line, making it easier to interpret the data. Power-law behavior is often seen in complex systems with many interacting components, such as networks, and can signify self-organized criticality, where the system operates near a critical state. This behavior highlights the system's ability to be sensitive to small changes at lower values while producing significant outputs at higher values, emphasizing both complexity and adaptability.

- **Case 1: Number of Nodes (N)** The relationship $y = 9.46 \cdot n^{-1.33}$ indicates a strong negative scaling, where increasing the number of nodes significantly decreases the time to reach coherence. The large exponent magnitude (-1.33) implies that the system becomes much less sensitive as the number of nodes grows, likely due to saturation effects in highly connected networks.

- **Case 2: Scaling Factor (α)** The relationship $y = 0.14 \cdot \alpha^{-0.92}$ shows a slower decrease in time to reach coherence as α increases. The smaller exponent magnitude (-0.92) indicates that the influence of α is more sustained across scales compared to n , suggesting that this factor has a more gradual effect on the system dynamics.

Summary of the influence of parameters

In conclusion, our results demonstrate that the power-law behavior of certain parameters reflects the inherent complexity of the system. The number of nodes has a pronounced diminishing effect on the time to reach coherence, with larger networks achieving synchronization much faster. In contrast, the scaling factor shows a more gradual impact, maintaining its influence across different scales. For the parameters with linear relationships, such as

adjacency matrix density, we observed that increased connectivity significantly reduces the time to reach coherence. Additionally, the standard deviation of natural frequencies exhibits a saturation effect, where values exceeding 10% of the mean no longer influence the system's behavior.

3.2 Implementation of a Input

The next step in our work is to introduce an input to the system, as this is essential for achieving our ultimate goal of constructing a neural network. A neural network requires an input to process and propagate information, making this a critical component of our model. In this section, we will present the formula for the input, explain the modifications it introduces to the current model, and discuss the theoretical implications of its role in the system dynamics.

3.2.1 Equation of the Kuramoto model with a input

$$\frac{d\theta_i}{dt} = \omega_i + \alpha \sum_{j=1}^N K_{ij} A_{ij} \sin(\theta_j - \theta_i) + \beta \sin(\theta_{\text{input}}(t) - \theta_i), \quad i = 1 \dots N \quad (6)$$

This equation is the same as the previous upgraded model, with the addition of an input term. The parameter β serves a role analogous to α , but specifically governs the strength of the interaction between the oscillators and the external input. The term involving $(\theta_{\text{input}}(t) - \theta_i)$ represents the interaction between the input signal and each oscillator, capturing how the input influences the dynamics of the entire system.

Here is a 2D scheme to better understand the architecture with a input :

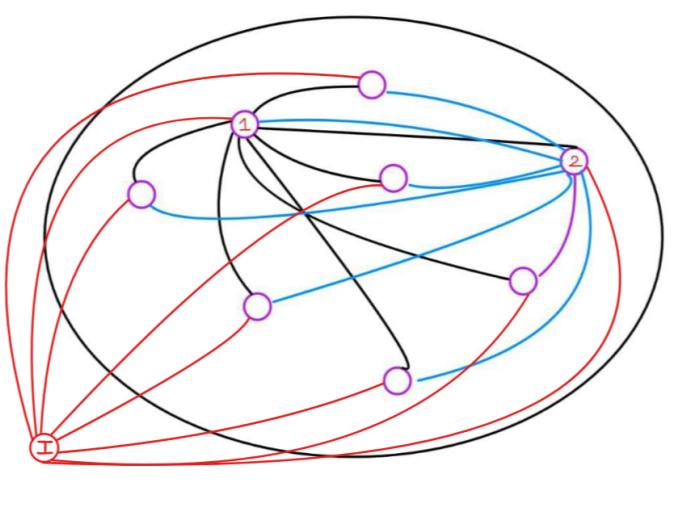


Figure 12: Possible architecture of the Oscillator Neural Network with a Input

The input applied to the i -th oscillator. The input function can take various forms depending on the application, such as a constant signal, a periodic waveform, or a more complex time-dependent signal. In our case our input is defined as a function that evolves in time like that :

$$\theta_{\text{input}}(t) = \text{amplitude} \cdot \sin(2\pi \cdot freq_{\text{input}} \cdot t) \quad (7)$$

From a **theoretical perspective**, we expect the oscillators to phase-lock to the input under certain conditions, indeed research published a article [Bronsiki and Wang(2020)] on the phase locking nature of the Kuramoto model, driven by external forces, providing insights into the conditions under which such synchronization occurs. Here we could expect phase locking to occurs when the oscillators synchronize their phases or frequencies with the input signal.

The input serves as a control mechanism that can drive the entire system. When sufficiently strong, it aligns the dynamics of all oscillators, creating a unified response. This behavior is important because it enables us to direct the system toward desired states or outputs, mimicking the processing capabilities of a neural network.

3.2.2 Validation of our Input Implementation

To set our validation we tested the newest model for different value of β . We want to precise that for theses results the simulation was done with a amplitude of 1 and a input frequency of 0.5.

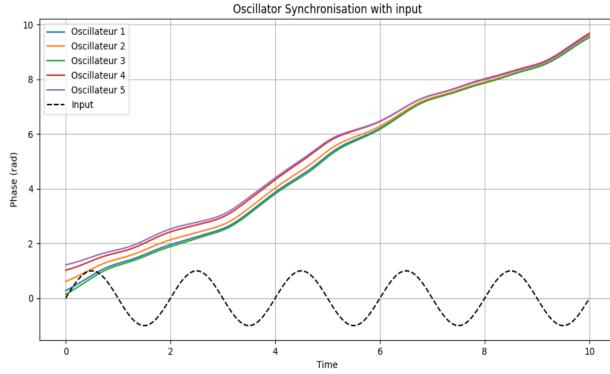


Figure 13: Phase Locking vs Time for $\beta = 0.5$

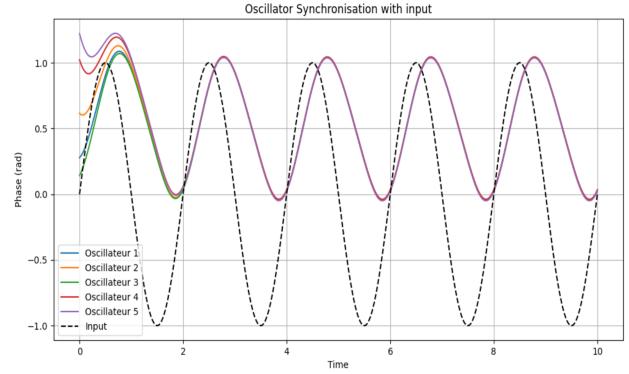


Figure 14: Phase Locking vs Time for $\beta = 2.5$

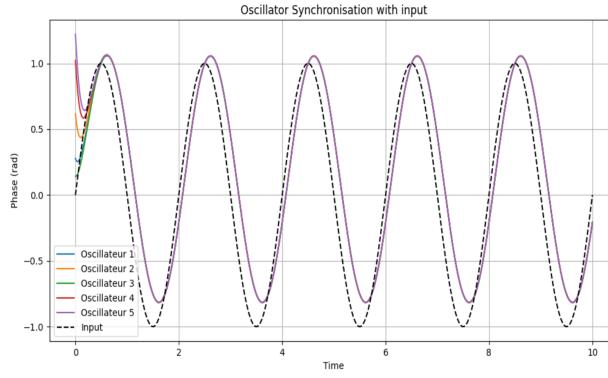


Figure 15: Phase Locking vs Time for $\beta = 8.5$

First, we observed very small β value induced a behavior as is the input wasn't present, so this kind of behavior is encouraging as we should still be able to have "cut off" the input. When we increase the β value, the behavior follows what we expected (as we discussed before) —locking onto the input phase.

By running more simulations with different input frequencies, we gained information and observed that the input frequency also affects the phase locking. To characterize this behavior, we defined a phase locking value and plotted a 2D map to represent the influence of both β and the input frequency.

To quantify the phase locking between the oscillators and the input signal, we utilized the Phase Locking Value (PLV). The PLV is a measure of how closely the phases of the oscillators are synchronized with the input signal. It provides insight into the strength of the phase coupling between the oscillators and the external input.

The PLV for each oscillator i can be calculated using the following formula:

$$\text{PLV}_i = \left| \frac{1}{T} \sum_{t=1}^T e^{j(\theta_i(t) - \theta_{\text{input}}(t))} \right| \quad (8)$$

where:

- $\theta_i(t)$ is the phase of oscillator i at time step t ,
- $\theta_{\text{input}}(t)$ is the phase of the input signal at time step t ,
- T is the total number of time steps.

The average PLV across all oscillators is then given by:

$$\text{PLV} = \frac{1}{N} \sum_{i=1}^N \left| \frac{1}{T} \sum_{t=1}^T e^{j(\theta_i(t) - \theta_{\text{input}}(t))} \right| \quad (9)$$

where N is the number of oscillators.

This measure provides a comprehensive understanding of how well the phases of the oscillators are locked to the input signal, giving us valuable insights into the synchronization dynamics in our study. The value of the heat-map are based on the mean of the N PLV value, so the Eq.9

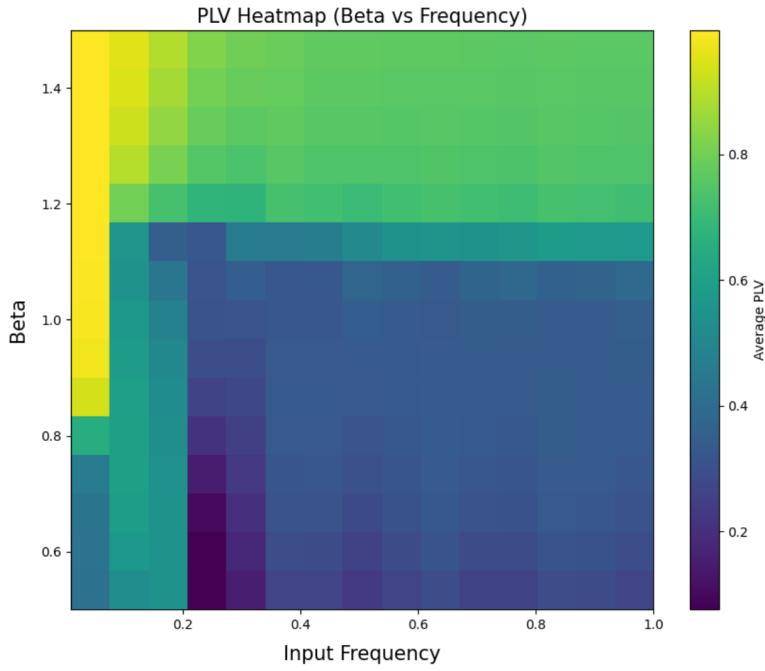


Figure 16: Phase Locking value depending on the β and Input frequency

Heat-map Interpretation :

The initial observation is that the beta parameters will exert a more significant influence on the overall locking phenomenon, as they act as a scaling factor affecting all values of the input signal. Concerning the input frequency, we noted an increase in PLV even for low values of beta when the frequency was below 0.2. However, above this threshold, it becomes increasingly challenging to achieve further increases solely through variations in input frequency.

In conclusion, these findings suggest that while both beta parameters and input frequency play crucial roles, their effects are more pronounced at different regimes. Specifically, for low beta values, a higher input frequency can enhance locking; however, this relationship diminishes as the frequency surpasses 0.2, highlighting the need for adjustments in beta to achieve significant improvements in locking performance.

3.3 Conclusion Beta version

In conclusion, our results demonstrate that the number of nodes significantly affects the time to achieve synchronization, with larger networks reaching coherence much faster. In contrast, the scaling factor impacts the system gradually but consistently across different scales. For linear parameters such as adjacency matrix density, increased connectivity substantially reduces the time required for coherence. Additionally, the standard deviation of natural frequencies exhibits a saturation effect, where values exceeding 10% of the mean no longer significantly influence the system's behavior.

Regarding β and the input frequency, for low values of beta, an increase in input frequency enhances locking. However, this relationship diminishes as the input frequency surpasses 0.2, highlighting the need for adjustments in beta to achieve significant improvements in locking performance

4 Discussion

This work highlights the utility of the Kuramoto model in understanding synchronization dynamics, particularly in VCSELs, which are promising for optical neural networks. The initial phase focused on building and validating a robust model, ensuring that the system accurately reflected the expected behavior of coupled oscillators. This provided a solid foundation for exploring more complex dynamics in the second phase.

In the later stage, we uncovered how factors like network size, connectivity, and external inputs influence synchronization. Larger networks synchronized faster, while increased connectivity and careful tuning of parameters like β and input frequency further enhanced performance. These findings provide valuable insights for optimizing VCSEL-based networks, emphasizing the importance of parameter selection in real-world applications.

While this project has laid the groundwork, there is room to push it further. A natural next step would be to complete the neural network and train it on specific tasks, allowing us to test its computational efficiency and scalability. Such work could bridge the gap between theoretical modeling and practical implementation, advancing the development of efficient photonic neural networks capable of outperforming traditional systems in certain fields.

Acknowledgments

I would like to thank Professor Daniel BRUNNER for giving me the opportunity to work on such an interesting topic. His guidance really helped me deepen my curiosity and understanding of PNNs. I'm also grateful for the time and support of Dr. Anas SKALLI and PhD student Jacob PUHALO-SMITH , who were always there to help me with the day-to-day aspects of my project.

Finally, I'd like to thank my colleague Léo BECHET for suggesting I reach out to Professor Brunner to collaborate on this project.

A Appendix

A.1 Validation of the Basic Model

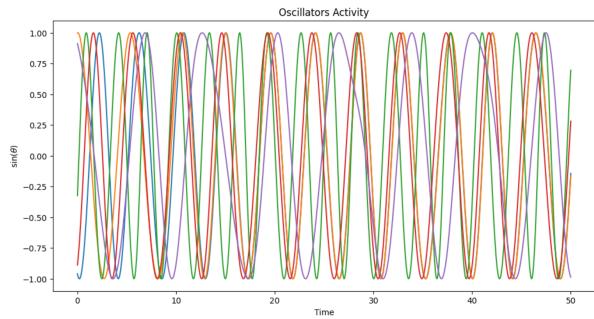


Figure 17: Time evolution of Oscillator Activity

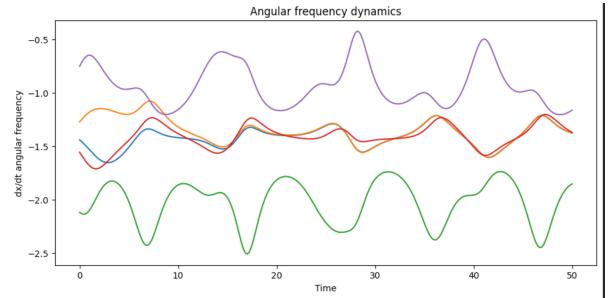


Figure 18: Time evolution of the Angular Frequency

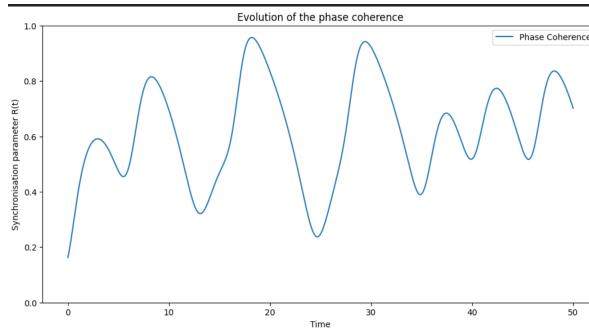


Figure 19: Time evolution of the Coherence

Tools to study the Kuramoto model with natural frequencies (mean 1, std 0.5) and $K = 0.5$

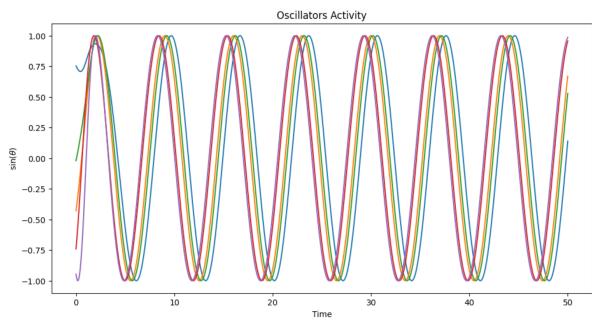


Figure 20: Time evolution of Oscillator Activity

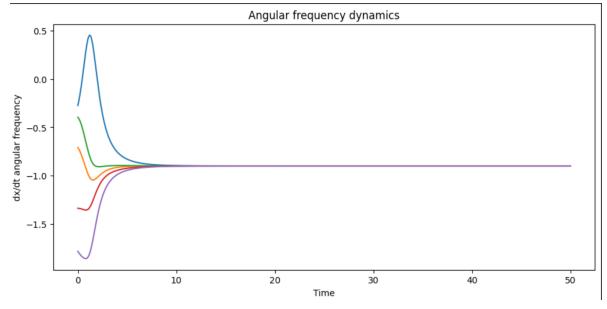


Figure 21: Time evolution of the Angular Frequency

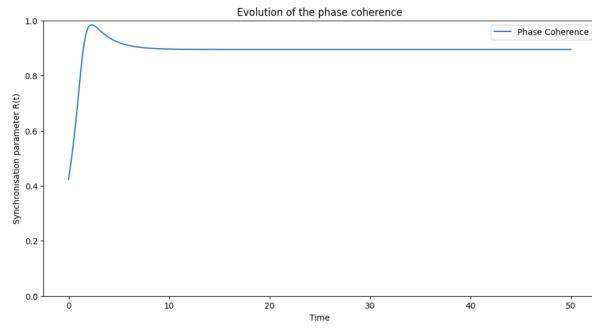


Figure 22: Time evolution of the Coherence

Tools to study the Kuramoto model with natural frequencies (mean 1, std 0.5) and $K = 1$

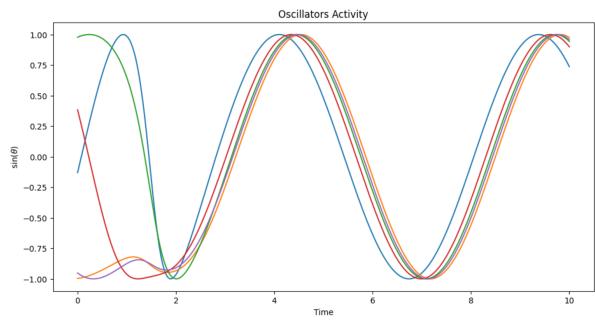


Figure 23: Time evolution of Oscillator Activity

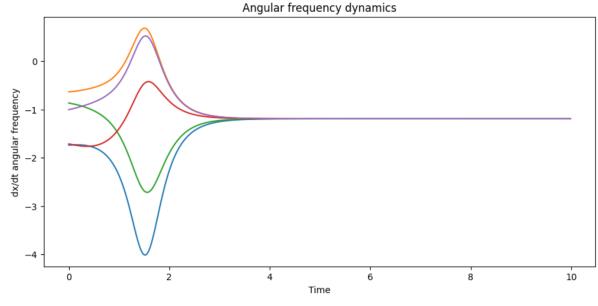


Figure 24: Time evolution of the Angular Frequency

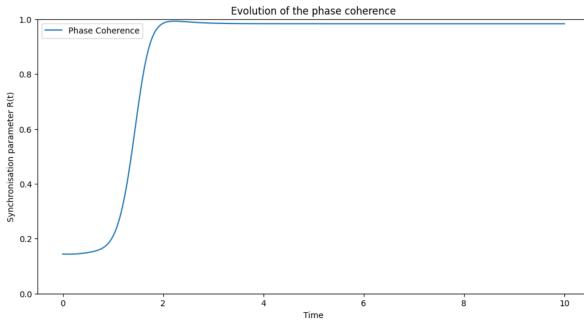


Figure 25: Time evolution of the Coherence

Tools to study the Kuramoto model with natural frequencies (mean 1, std 0.5) and $K = 3$

References

- [Bronski and Wang(2020)] Jared C. Bronski and Lan Wang. Partially phase-locked solutions to the kuramoto model. 2020. Available online: May 14, 2020.
- [Cumin and Unsworth(2022)] D. Cumin and C.P. Unsworth. Generalising the kuramoto model for the study of neuronal synchronisation in the brain. *Journal of Computational Neuroscience*, 12(7):2793, 2022. Received 2 May 2006, Accepted 5 December 2006, Available online 23 January 2007.
- [Skalli et al.(2022)Skalli, Porte, Haghghi, Reitzenstein, Lott, and Brunner] A. Skalli, X. Porte, N. Haghghi, S. Reitzenstein, J.A. Lott, and D. Brunner. Computational metrics and parameters of an injection-locked large area semiconductor laser for neural network computing. *Optical Materials Express*, 12(7):2793, 2022. Invited, Email: anas.skalli@femto-st.fr.