Kuroscillator: A Max-MSP Object for Sound Synthesis using Coupled-Oscillator Networks

Nolan Lem[1] and Yann Orlarey[2]

[1] Center for Computer Research in Music and Acoustics (CCRMA) Stanford University
[2] GRAME (Centre national de création musicale)
nlem@ccrma.stanford.edu

Abstract. This paper summarizes recent research using networks of coupled oscillators in real-time audio synthesis. We present two Max-MSP objects that synthesize the dynamics of these systems in real-time using a both an additive and rhythmic synthesis model to generate complex timbre and rhythmic content. This type of self-organizing system presents many useful avenues of exploration in the field of sound synthesis and rhythmic generation. These objects allow users of Max-MSP to synchronize different ensembles of sinusoidal oscillators in real-time which can then be used as a vehicle for creative sound design, composition, and sound art.

Keywords: coupled oscillators, audio synthesis, Max MSP, generative music

1 Introduction

Coupled oscillator networks are a type of dynamical systems that describe a wide variety of interactive phenomena relevant to a number of research fields. Among those pertinent to the natural world, coupled oscillator systems have been used to account for firefly synchronization, synchronous chorusing in animal populations, and the cortical rhythms that comprise human neural networks [1]. In terms of musical beat and perception, Large and colleagues have incorporated coupled oscillators networks in their computational models to characterize how we become entrained to different types of rhythmic stimuli [2].

Previous research in coupled oscillator networks as a generative sonic device has looked at how their system behavior can be exploited in terms of their potential for creating musically relevant content [3] [4]. This includes exploring different strategies for rhythmic generation, audio synthesis, and as a control signal to approximate many compositional techniques found in contemporary music and computer music [5].

2 System Dynamics: Ensembles of Kuramoto Oscillators

In order to better understand the dynamics of these systems, a brief introduction to the Kuramoto model is presented. Kuramoto proposed a model of limit-cycle oscillators that interact at the group level through their phase interactions [6]. Equation (1) shows the governing equation for such a system.

$$\dot{\phi}_i = \omega_i + \frac{K_i}{N} \sum_{j=1}^{N} sin(\phi_j - \phi_i) \tag{1}$$

where ϕ_i is the phase of the i_{th} oscillator and $\dot{\phi}_i$ is the derivative of phase with respect to time. ω_i is the intrinsic frequency of the oscillator, i, in a population of N oscillators. K_i is the coupling factor for each oscillator and the $sin(\phi_j - \phi_i)$ term is the phase response function that determines the interaction between each oscillator and the group. Typically, the range of intrinsic frequencies within the ensemble is taken from a Gaussian distribution, $g(\omega)$ at a center frequency, ω_c.

As K_i is increased, the oscillators with an ω_i closer to ω_c will begin to synchronize to the group by aligning their phases to other oscillators with similar frequencies. As more and more oscillators are recruited, synchrony emerges when $K_i > K_c$ where K_c is the point of critical coupling. Assuming a Gaussian distribution of intrinsic frequencies, Kuramoto was able to show that as the number of oscillators goes to infinity, $K_c = \frac{2}{\pi g(f_c)}$.

Much more complicated types of synchrony occur when we let the intrinsic frequencies and coupling coefficients, $\omega_i(t)$ and $K_i(t)$, take on different values as a function of time. For example, Abrams and Strogatz were able to show the existence of "chimera states" that exhibit unusual dynamics where phase-locked oscillators coexist with asynchronous ones [7]. These system dynamics have perceptual implication in the audio synthesis routines described in the following section.

2.1 Programmatic Design

The main challenge in designing a real-time synthesis scheme using the aforementioned model is accounting for the phase interactions that must occur at each sampling interval. The phase response term in Equation (1) shows how the model grows exponentially as a function of N ($O(N^2)$) for each oscillator in the ensemble. To reduce the number of calculations per sampling interval, we use Kuramoto's application of mean-field coupling to the oscillators phases to derive the complex order parameters shown in Equation (2)

$$Re^{j\psi} = \frac{1}{N} \sum_{i=1}^{N} e^{j\psi_i} \tag{2}$$

where R and ψ are defined as the phase coherence and average phase respectively [6]. We can represent quadrature component of the complex phasor as a 90° phase shifted version of the in-phase part. Now the oscillators are no longer explicitly coupled to one another because their average phase governs their behavior. This is shown in Equation (3).

$$\dot{\phi}_{\text{i}} = \omega_i + KR \sum_{j=1}^{N} sin(\psi - \phi_i) \tag{3}$$

From a computational standpoint, this has the benefit of reducing the number of calculations necessary to carry out the phase coupling adjustments and allows for a greater number of oscillators within each ensemble.

2.2 Faust Implementation

To create an interactive model, we utilized the functional programming language, Faust (Functional Audio Stream)[3], to implement the real-time signal processing. Faust is capable of being compiled into to a number of music programming related objects including Supercollider and Max MSP. Within Faust, the system is implemented by defining a series of "adjustable phasors" that can be modulated in terms of (instantaneous) phase and intrinsic frequency. By creating a feedback loop that calculates the average phase of the group of oscillators, the phase adjustments (shown in Equation (1) can be meaningfully applied to each term thereby allowing the oscillators phases to synchronize. This is shown in Fig 1 which shows the block diagram in Faust with four oscillators.

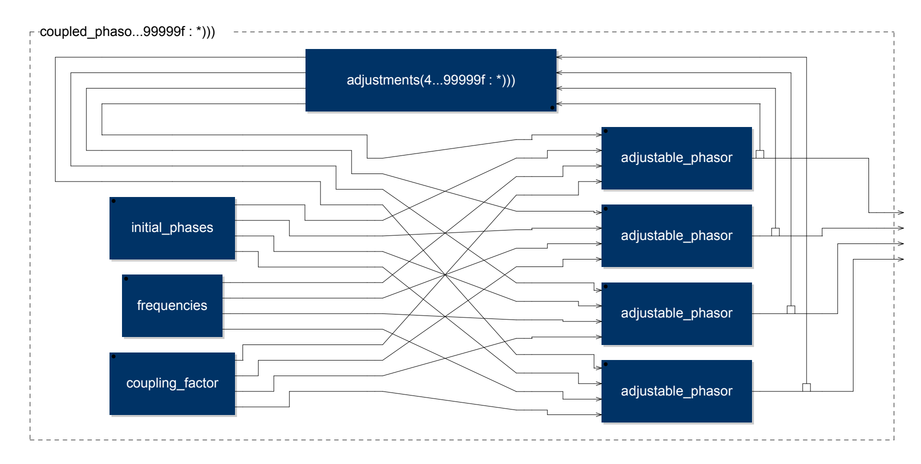

Fig. 1. Faust block diagram.

In the simple additive synthesis scheme, these phasors are applied as arguments into sine functions at the output to generate a bank of N phase-coupled sine waves. To use the oscillators to generate rhythms, we use the trajectory of

[3] `https://faust.grame.fr`

each phasor to trigger audio events upon each zero crossing ($\phi_i < \phi_{i-1}$) they encounter. We also must significantly decrease the distribution of ω_i to fall within a range normal for beat perception (0.25 Hz to 30 Hz). In order for the user to be able to interact with the model, each oscillator's coupling and intrinsic frequency are made variable.

3 Kuroscillator-rhythm and Kuroscillator-audio objects

The Kuroscillator max objects can be found in the directory listed below[4]. Figure 2 shows the *Kuroscillator-rhythm* and *Kuroscillator-audio* objects within the Max MSP programming environment.

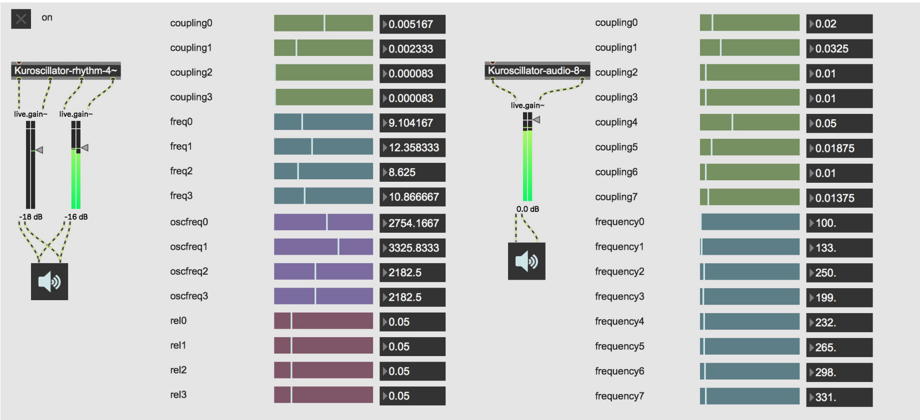

Fig. 2. *Kuroscillator-rhythm* object with 4 oscillators (left) and *Kuroscillator-audio* object with 8 oscillators (right).

The intrinsic frequencies of oscillators in both objects are limited to the range of 0.25 Hz to 30 Hz. In the Kuroscillator-rhythm object, the user can also modify the frequency of the audio oscillator's triggered note (33 Hz $< oscfreq <$5 kHz) and the length of the ASR envelope that gets applied to it. The Kuroscillator-rhythm object generates N individual outputs so they can be used as a control signal within Max MSP whereas the Kuroscillators-audio object is mixed down to two channels at the output. These constraints can be modified in the Faust source code that is located in the code directory.

The user defines which *type* of coupled oscillator object (*rhythm* or *audio*) and how many oscillators (N) in the ensemble using the convention "Kuroscillator-*type*-N" (where $N \leq 30$). Max MSP then generates the object and allows the user to interact with the coupling and intrinsic frequency by sending Max style messages to the Kuroscillator objects. These parameters are addressable: for

[4] `https://bitbucket.org/no_lem/kuroscillators/src/master/`

the Kuroscillator-audio object, the user can send messages using the format "./Kuroscillator-type-N/coupling$i$ K_i" and ""./Kuroscillator-type-N/frequency$i$ $frequency$"" to modify the couplings and intrinsic frequencies respectively. For the Kuroscillator-rhythm object, the user can send additional Max messages of the form "./Kuroscillator-type-N/oscfreq$i$ *oscillator frequency*" and "./Kuroscillator-type-N/rel$i$ *sustain-time(sec)*". These parameters allow the user to explore different system states and bifurcations that are generated in real time. Assuming an intrinsic frequency distribution that is Gaussian (unimodal with respect to a center frequency), the oscillators can self-synchronize to the mean frequency of the distribution when they approach the critical coupling coefficient. Figure 3 shows the output of the *Kuroscillator-additive-50* object which shows a spectrum of a system of 50 oscillators synchronizing over a period of around 6 seconds.

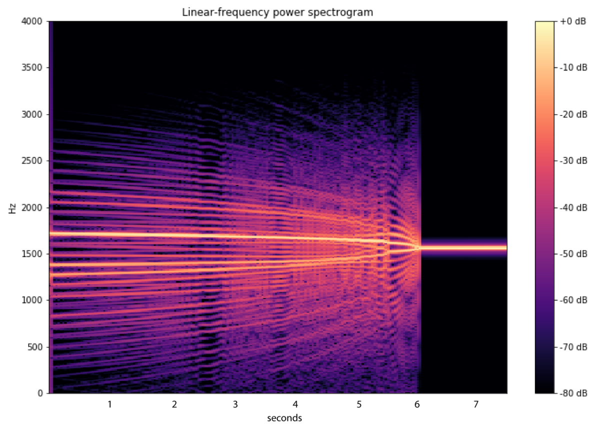

Fig. 3. Synchronizing frequencies of *Kuroscillator-additive* object with 50 oscillators.

Due to the flexibility of the programming environment, Max MSP is well suited for interaction with the model since the user can utilize the plethora of control objects in the Max tool kit. The examples directory hosted on the project site includes several Max example patches that show interesting system states of synchrony by allowing control signals to modulate system parameters over time. These highlight the versatility of this model in producing sonic phenomena that approximates many pre-existing techniques already in the field of computer music namely (granular synthesis) and contemporary composition (temporal canonization and phasing).

4 Conclusions

These coupled oscillator objects allow for real-time interaction with this particular self-organizing dynamical system. Because these systems contain a plethora of unusual output behaviors, Max MSP facilitates creative sonic exploration by allowing users to interact with system behavior using its output sound as a form of auditory feedback. Besides the sonic states characterized by "full synchrony" (in which all oscillators synchronize to a mean frequency), there exists a number of quasi-periodic states that emerge when oscillators take on different coupling coefficients and intrinsic frequency distributions [8]. In general, it is likely that the audio generated by these objects are perhaps best suited for types of music that are more oriented toward experimental music practices such as procedural, minimalist, and drone musical genres.

The Kuramoto model is just one particular type of coupled oscillator system that allows for self-synchronizing behavior. There are several other types of coupled systems that would be interesting candidates for synthesis: these include pulse-coupled oscillators (Mirollo-Strogatz oscillators), Van der Pol oscillators, and Stuart-Landau oscillators each one comprised by their own unusual dynamics [9] [3]. Future research in the sonifying these dynamical systems would be enhanced by integrating extant psychoacoustic models that allow us to better perceive their system dynamics, particularly those that have been documented in existing research.

References

1. Ravignani, A., Bowling, D., Fitch, W. T.: Chorusing, synchrony and the evolutionary functions of rhythm. In: Frontiers in Psychology. 5, 1–15 (2014)
2. Large, E., Herrera, J. A., Velasco, M. J.: Neural Networks for Beat Perception in Musical Rhythm. In: Frontiers in Systems Neuroscience, pp.1–14 (2015)
3. Lambert, A. a Stigmergic Model for Oscillator Synchronisation and Its Application in Music System. In: Proceedings of the International Computer Music Conference, pp. 247–252. Ljublana (2012)
4. Collins, N. Errant sound synthesis. In: Proceedings of International Computer Music. (2008)
5. Lem, N.: Sound in Multiples: Synchrony and Interaction Design using Coupled-Oscillator Networks. In: Proceedings International Conference Sound and Music Computing. Malaga, Spain (2019)
6. Kuramoto, Y.: Self-entrainment of a population of coupled non-linear oscillators. In: International Symposium on Mathematical Problems in Theoretical Physics, Lecture Notes in Physics, pp. 420–422. Springer, Berlin (1975)
7. Abrams, D., Strogatz, S.: Chimera States for Coupled Oscillators. In: Physical Review Letters, pp. 1–4. The American Physical Society, (2004)
8. Pikovsky, A., Rosenblum, M.: Dynamics of globally coupled oscillators: Progress and perspectives. In: Chaos. (2015)
9. Mirollo, R., Strogatz, S. Synchronization of Pulse-Coupled Biological Oscillators. In: SIAM Journal of Applied Mathematics, pp. 1645–1662.(1990)