MACHINE-MACHINE COMMUNICATION WITHOUT HIERARCHIES AND PROTOCOLS

CHRISTIAN FAUBEL
Academy of Media Arts Cologne
c.faubel@khm.de

Keywords
Analogue computation
Oscillators
Synchronisation
Neural Networks
Rhythm
Timing

In this paper I present two different implementations of analogue oscillators that synchronize when they are mutually coupled. I consider these as examples for machine—machine communication without hierarchies and protocols. Synchronization is the most fundamental indicator for the exchange of information. In these analogue oscillators this exchange is realized through a direct coupling, no encoding and no decoding has to be effectuated. Not one oscillator is dominating the communication, they mutually negotiate their timing. This capacity for non-hierarchical communication is a signature of analogue systems. They allow for information exchange without having to recur to symbols or to detect the presence of a signal, already the softest input may modify and modulate the behaviour of an analogue system. Presenting a digital implementation and approximation of an analogue system, this paper is clearly not meant to dismiss digital systems but to open up a perspective on the analogue as conceptual approach towards openness.

2016.
xCoAx
.org

Computation
Communication
Aesthetics
& X
Bergamo, Italy

1 INTRODUCTION

Obviously, nowadays the machines surrounding us are in communication and this communication is largely built upon a digital subface[1]. All the infrastructure for the communication between machines follows digital paradigms and protocols, from serial interfaces, over MIDI, to Open Sound Control or the Internet-Protocol to name some of the most obvious ones. Inherent to this type of digital communication is its closeness, it is only possible to understand the communication or to intervene in the communication between machines if the protocol is known. Intrinsic to every protocol is also a hierarchy, between those who define the protocols, those who implement them and those who follow the protocols. Inherent to the digital communication is furthermore the need for synchronization or as Trogemann and Scherffig write

> … From this general perspective the problem of synchronization is inherent to algorithmic computation, without being explicitly wanted the necessity for synchronization is a consequence of the nature of computation and thus deeply inscribed in the notion of the algorithm. (Trogemann and Scherffig 2013)

With this paper I want to propose mechanisms of machine -machine communication that do not require a protocol and that show synchronization not as a necessity for communication but as a consequence of communication, as an emergent signature of communication.

The mechanisms of machine-machine communication, considered in this paper, share that they are built upon an analogue subface, being characterized by a subtle and graded exchange of information between the machines. The oldest known example of machine-machine communication is that of synchronizing pendulums. It was first discovered in the 17th century by Christian Huygens (Bennet et al. 2002) who was developing maritime pendulum clocks. While at home due to illness he observed that the pendulum clocks, he had developed, were swinging in a perfect counter-phase relationship. Even when he disrupted this fixed phase relation, after some time the clocks would swing again in fixed counter-phase.

This effect of synchronizing pendulums can be reproduced using standard metronomes. Two metronomes or even more synchronize in phase when they are put on a common support platform that mediates the swing between the metronomes. The simplest setup for the experiment uses two empty cans as wheels and a small wood plank as a support base to place two metro-

1. Subface is a term coined by Frieder Nake, for describing the computable signs manipulated by digital systems in opposition to the perceivable signs on the surface (Nake 2008; 2013): 45.

nomes. After a short time the metronomes will synchronize. There is no hierarchy involved, the metronomes sway and are mutually influencing each other through the common support platform. The Ikeguchi Lab (Ikeguchi 2016a) has an impressive number of demonstrations from synchronizing metronomes to dripping water bottles and candles (Ikeguchi 2016b).

While the scope of this paper is on machine-machine communication examples of synchronization are also observable in organisms. Examples are the synchronisation of fireflies (Buck 1938) or the synchronization of a public when giving applause (Neda et. al. 2000). Such phenomena have inspired the development of a proto-musical system based on synchronisation of oscillators called crickets (Lambert 2012). The present paper points in a similar direction as the cricket system but the focus here is on the fact that the basic principle of synchronisation is based on the exchange of analogue or graded quantities.

I will present two oscillatory systems with synchronization that take their inspiration from synchrony as it is observed in the nervous system (Dayan and Abbott 2001). The first is an electronic realization of a very minimalistic artificial neural network based on the work by Hasslacher and Tilden (Hasslacher and Tilden 1995). The second system is digital approximation of a model of oscillatory neural circuits as they have been proposed and analysed by Shun-Ichi Amari (Amari 1977). In a third example I will show how these two systems can also synchronize with each other, bridging between the analogue and the digital subface.

2 SYNCHRONISATION OF ELECTRONIC ANALOGUE OSCILLATORS

The electronic analogue oscillator I propose is composed of two basic units (see Fig. 1). A basic unit approximates a biological neuron with two functional aspects: first similar to a neuron it only produces an output when stimulated to a sufficient level (Dayan and Abbott 2001). Second, it adapts to its input: on constant input it stops producing output. These two functional aspects are realized by coupling a differentiator circuit to an inverter. The differentiator is built by connecting a capacitor to a resistor, the inverter is then connected in between the capacitor and the resistor. The differentiating part is used to approximate the adaptation to an input, the inverter is used as non-linear, thresholding element.

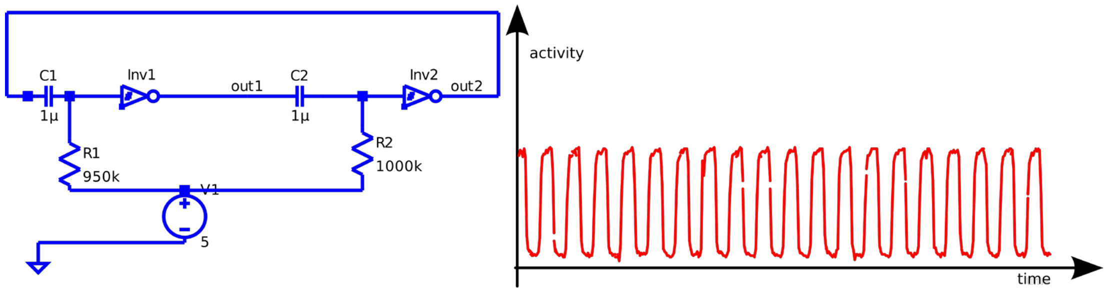


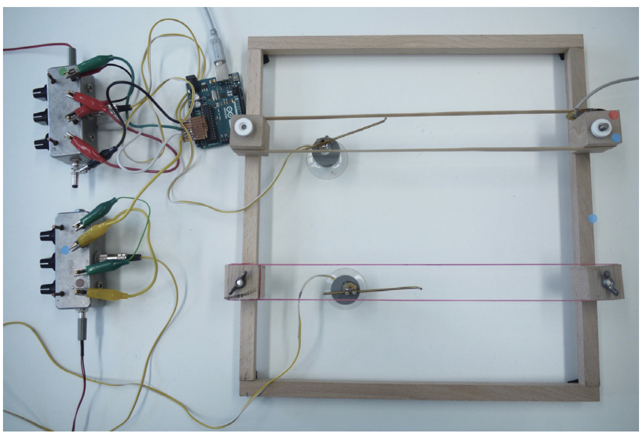
Fig. 1. The left image shows the schematic and the right plot shows a recording of a bi-core.

While a single unit does not do much, oscillatory patterns emerge when the units are connected into a loop (see Fig. 1). Such oscillatory behaviour emerges also in neural networks with very simple threshold models of integrate and fire neurons (Amari 1977; Wiener and Rosenblueth 1946).

The simplest recurrent network of basic units that produces oscillatory behaviour is a network of two units mutually coupled to each other, also referred to as bi-core (Hrynkiw and Tilden 2002). The bi-core can be used to directly drive dc-motors with a gearbox. When a motor is connected, the internal timing of the bi-core is influenced by the motor. I use this setup to create self-organized beats in a machine called the rhythm apparatus (Faubel 2013).

Two bi-cores with motors will oscillate independently, but when they are linked with a cable so that one output connects to the other's output, their outputs will go in synchrony (see Fig. 2. for the experimental setup). This works also if the oscillating frequencies of the two bi-cores do not match. They will synchronize on an intermediate frequency. However using a variable resistor as link in between also allows for complex phase relationships, where they are in sync but they only meet every n-th beat of the faster going bi-core. This is visible in the data recording pictured in Fig. 3. The two bi-cores start off phase locked on the same frequency over time the coupling is lowered by increasing the resistance of the connecting cable. As a consequence, the faster bi-core can recover its Eigenfrequency, but it stays in sync with the slower bi-core, only matching every third beat.

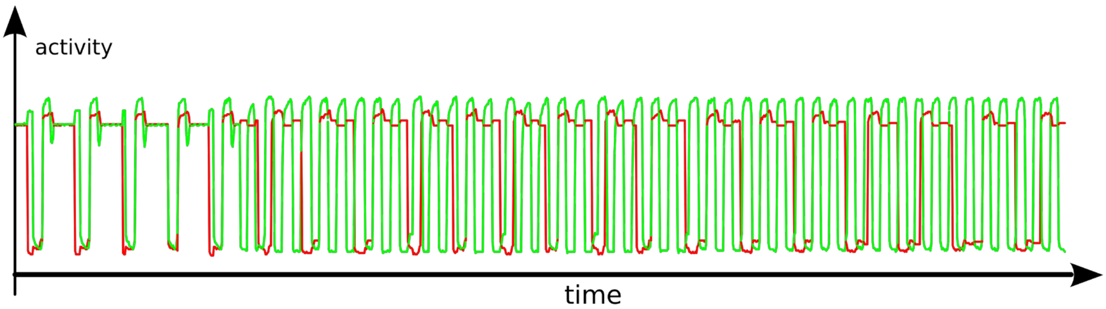
Fig. 2. Photo of the experimental setup, two bi-cores (aluminium cases) controlling two motors are connected with a variable resistor (black case), the Arduino is used for recording the data. a video of the experiments is available online[2].



2. Video is online here: https://vimeo.com/165357455

Fig. 3. Synchronization between two bi-cores, one (green) going at a faster beat, the other (red) at a slower beat. Over time the coupling is reduced. Initially the two bi-cores are in full sync, at lower coupling they synchronize on every third beat of the faster bi-core.

What is important to remark is that in order to synchronize these bi-cores only a single connection is needed, there is no difference between reading and writing. Current is flowing in both directions of the connecting cable, not one oscillator entrains the other but their coupling is mutual, they entrain each other. Varying the resistance of the cable controls the degrees of freedom of this coupling. The lower the resistance of the cable the stronger is the coupling,



3 SYNCHRONIZATION OF ANALOGUE OSCILLATORS REALIZED ON A DIGITAL SUBFACE

For the digital implementation of an analogue oscillator I use a model proposed by Shun-Ich Amari, the Amari oscillator. The Amari oscillator is based on the coupling of two neurons, an excitatory neuron is mutually coupled to an inhibitory neuron (Amari 1977). The excitatory neuron has self-excitation, once it becomes active it has a tendency to stay active. It also activates an inhibitory neuron, that when it is active will de-active the excitatory neuron. Once the excitatory neuron becomes deactivated, it will not activate the inhibitory neuron anymore, which is in turn becoming inactive. Without the negative input from the now inactive inhibitory neuron, the excitatory neuron will become active again and the cycle restarts. Both neurons are non-linear, which means that they have to reach a threshold level of input before they go into their active state[3].

The theoretical framework to describe such systems that change their behaviour in time is the dynamical systems theory. The oscillator's behaviour in time can be described by two equations, the first describing the rate of change of activation of the excitatory neuron $u(t)$, the second describing the rate of change of the inhibitory neuron $v(t)$:

$$\tau_u \dot{u}(t) = -u(t) + c_{uu}\sigma\big(u(t)\big) + c_{vu}\sigma\big(v(t)\big) - h_u$$
$$\tau_v \dot{v}(t) = -v(t) + c_{vv}\sigma\big(v(t)\big) + c_{uv}\sigma\big(u(t)\big) - h_v$$

3. To get a better understanding of these equations they can be directly played with at http://christian.faubel.derstrudel.org/dynamical-systems-basics and http://christian.faubel.derstrudel.org/coupled-dynamical-systems

The two equations are the same, only the parameter values τ_u, c_{uu}, c_{vu}, h_u, t_v, c_{vv}, c_{uv}, h_v differ. The rate of change of activation of the excitatory neuron is a function of the current activation $u(t)$, it relaxes with relaxation rate τ_u to the resting level h_u in the absence of other input. The function $\sigma(u(t))$ is a non-linear threshold function, the sigmoidal function

$$\sigma(u) = \frac{1}{1+exp(-\beta u)},$$

where β controls the slope of the sigmoid. When the activation $u(t)$ becomes supra-threshold it is multiplied with coupling term c_{uu} and contributes to the rate of change $\dot{u}(t)$. The same holds for the sigmoid of the inhibitory neuron $\sigma(v(t))$, it contributes with the negative coupling term c_{vu}, when the inhibitory neuron becomes supra-threshold.

While these equations can't be fully solved analytically they can be approximated or played-out on digital hardware. The standard method to numerically solve dynamical systems is the Euler method, which is in a way similar to sampling data, with the difference that the data is not captured as with a sound-card but computed based on previous computation. The faster the computer the more precise is the approximation of the system to simulate. To play around with these oscillators I programmed sketches for Processing, Pure-data, P5js and the Arduino platform. These sketches/patches are made publicly available[4].

4. Downloads available at http://christian.faubel.derstrudel.org/code/neural-oscillators

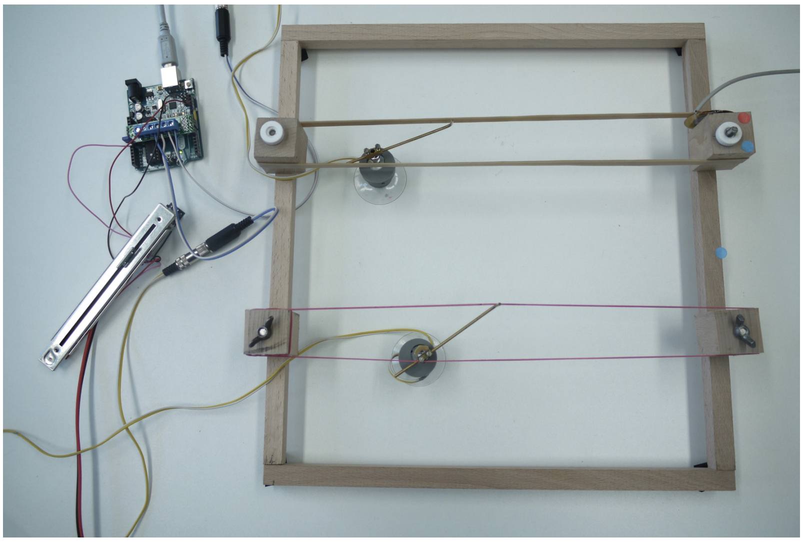


Fig. 4. Photo of the experimental setup, an Arduino controls two motors, each is driven by one oscillator computed on the Arduino, a video of the experiments is available online.

Two oscillators are coupled by mutually connecting the output of the excitatory neuron of the first oscillator to the inhibitory input of the second oscillator and vice versa. In this coupling the connection strength becomes a parameter. It can be used to switch the synchrony from in-phase relationship to a counter-phase relationship between the two oscillators (see Fig. 3 for the phase transition). Both in-phase and counter-phase are stable attractors in the phase space relationship of the two oscillators, the in-phase attractor being more stable. One can gradually vary the connection strength and observe how the system goes from one state of synchrony to the other. This behaviour of phase shift can also be explored in an online version[5].

5. http://christian.faubel.derstrudel.org/two-oscillators

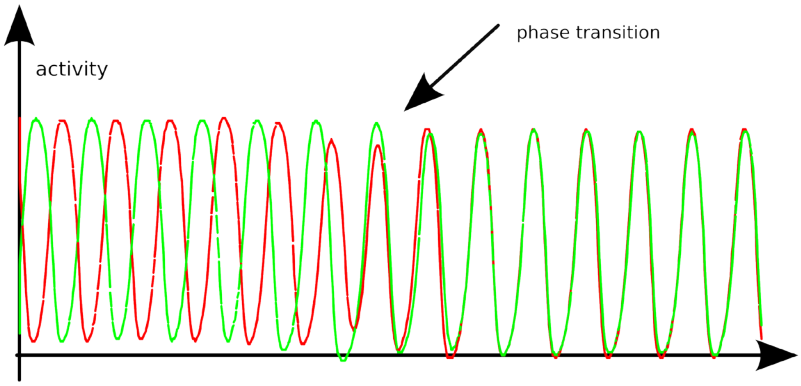


Fig. 5. The Plot shows a recording from two Amari oscillators on an Arduino platform. Two Amari oscillators start in counter-phase, then their coupling strength is modified and the system moves to in-phase synchrony.

The fact that the systems are showing two discrete stable states is also an example where at the surface a phenomenon may look digital, the system is either in in-phase or in counter-phase, as if there were only these two options, while at the subface the system is analogue and the parameter to control the phase shift is a graded one.

When both oscillators are set to run at different oscillation-speeds through different values of their relaxation rate, the mutual coupling can lead to interesting rhythmic coordination as is demonstrated by another online script[6].

6. http://christian.faubel.derstrudel.org/rhythmic-oscillators

3.1 ASSESSMENT OF INFORMATION TRANSFER

To assess the degree of information transfer the transfer entropy between the time-series produced by two coupled oscillators is calculated. The transfer entropy is a measure to evaluate the statistical coherence between systems evolving in time (Schreiber 2000). The transfer entropy is computed based on the Java Information Dynamics Toolkit (Lizier 2014). In the decoupled condition the transfer entropy is significantly lower than in the coupled condition (0.002 vs. 0.03). In the coupled condition the transfer entropy from the first to the second oscillator is not significantly different from the transfer entropy from the second to the first oscillator (0.032 vs. 0.030). This shows that the coupling is really mutual, it is not that one oscillator is entraining the other but they are mutually coupled each is influencing the other to the same degree.

3.2 COUPLING OF MULTIPLE OSCILLATORS

The synchronisation of Amari oscillators is not restricted to only two oscillators, larger numbers of oscillators will also synchronize. The connectivity between the oscillators may be random or all to all connections, both produce synchronized behaviour. This coupling of multiple oscillators is demonstrated in p5js script that can be viewed online[7] as well. The activity of the oscillators is rendered into an ever changing display and is made audible by using each oscillator to modulate the amplitude of a sound generator. Ten tiles on top represent the activity of each oscillator, the lighter colour corresponds to higher activity. The big tile below is a slowly varying oscillator modulating the overall coupling strength. The row below shows the phase plot of the excitatory versus the inhibitory neuron of each of the oscillators. The last row shows the evolution of each oscillator in time (see Fig. 6.).

7. http://christian.faubel.derstrudel.org/code/ten-synced-oscillators

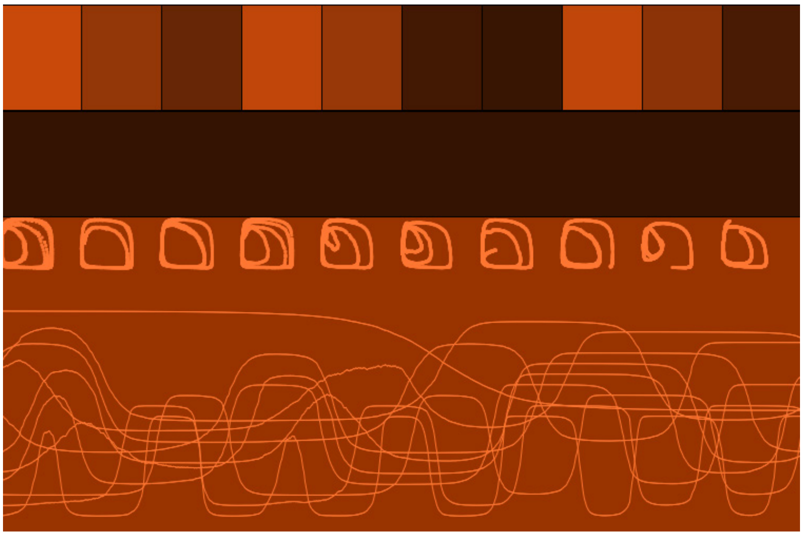


Fig. 6. Screenshot from a P5js implementation with 10 coupled oscillators.

4. SYNCHRONISATION ACROSS ANALOGUE AND DIGITAL SUBFACES

As the digital implementation of the Amari oscillator is conceptually analogue, it can be synchronised with analogue hardware. To connect the bi-core oscillator with the Amari oscillator, the bridging takes place on two levels, bridging from analogue to the digital subface and vice-versa and bridging the two analogue subfaces. The former is done with the classical components of the Arduino platform, the analogue Pulse-Width-Modulated (PWM) output and the Analogue-Digital transducer of the analogue ports (see Fig. 7 for the setup). The latter is done by using the analogue input value associated with the bi-core as input to the excitatory neuron of the Amari oscillator and by transforming the PWM output from the Arduino to a continuous activation by low-pass filtering it with resistor-capacitor pair. The strength of the mutual coupling is controlled with an additional potentiometer.

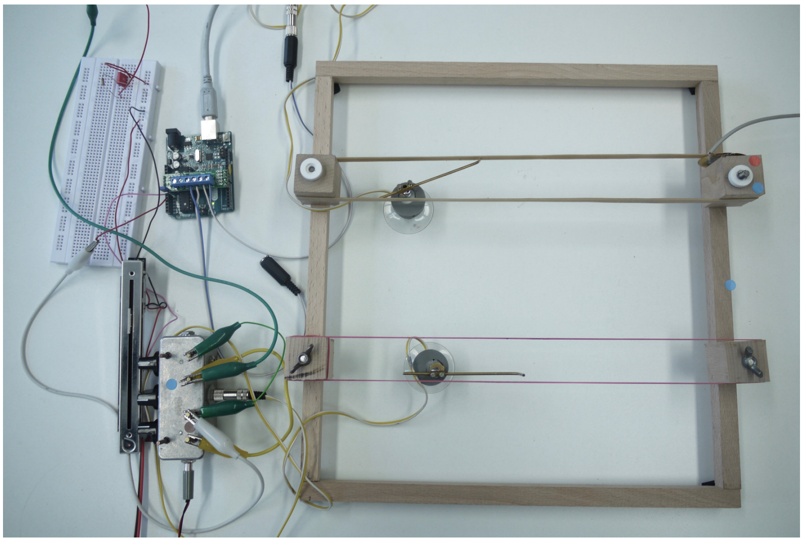


Fig. 7. Hardware setup for connecting the Arduino and the bi-core.

Fig. 8. shows the activation of the outputs from a bi-core and the output of the Amari oscillator, when the connection between one output of the bi-core and the excitatory neuron of the Amari oscillator is established by controlling the connection strength, both oscillators synchronize with a minimal but constant phase delay.

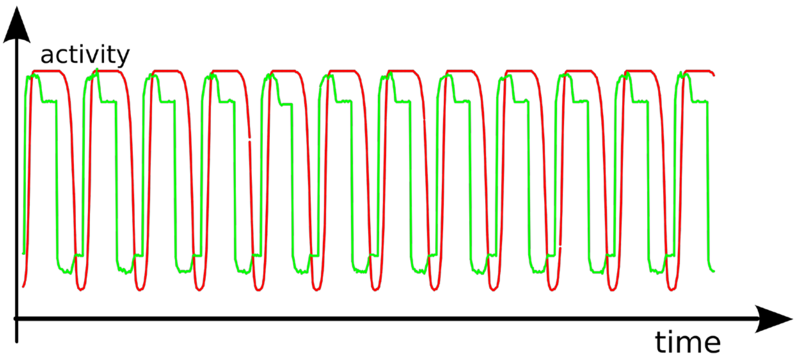


Fig. 8. The plot shows the activation of the Amari oscillator (red) and the bi-core, both are in synchrony.

5 CONCLUSION AND OUTLOOK

I have shown three examples of machine-machine communication that show synchrony as an emergent effect of mutual information exchange. All examples are analog oscillators, the first is implemented with analog electronics, the second is conceptually analog but implemented on digital hardware. The third is a mix of both types showing how to bridge analog and digital subfaces.

I specifically chose this second example to show that an analog approach can be realized on digital hardware and pertain the special quality of analog systems, which is an openness to sub-threshold input. This openness manifests itself in the fact that the oscillators synchronize based on a very soft mutual exchange of activation. On the other hand the second system also shows how a dichotomy of two separate states, namely in-phase and counter-phase oscillation, can emerge as distinct stable states out of a continuum of possible states. I see this as an example where the surface is digital either in-phase our counter-phase, while the subface is analog (the continuous activation of the neural variables).

The effect of synchrony is in all examples a result of mutual coupling where cause and effect are in a circular relationship. The degree to which the systems synchronize depends on the Eigen frequencies of the oscillators and on the coupling strength. This type of soft-coupling presents an alternative to established hierarchical methods of synchronization, such as the master-clock principle. With the bridging from the analog to the digital subface it becomes possible to link these oscillators to the digital protocols such as the MIDI protocol, effectively linking between analog and digital hardware.

With the examples of oscillator synchronization presented in this paper I hope to establish an appreciation of the analog as a conceptual paradigm that allows for an openness and flexibility in process communication.

REFERENCES

Amari, S.-i. *Dynamics of pattern formation in lateral-inhibition type neural fields.* Biological cybernetics 27, 2 (1977), 77 – 87.

Bennett, M., Schatz, M. F., Rockwood, H., and Wiesenfeld, K. *Huygens's clocks.* Proceedings: *Mathematics, Physical and Engineering Sciences (2002), 563 – 579.*

Buck, J. B. *Synchronous rhythmic flashing of fireflies.* The Quarterly Review of Biology 13, 3 (1938), 301 – 314.

Cometto-Muniz, J., and Cain, W. *Olfactory adaptation.* Handbook of Olfaction and Gustation. New York: Marcel Dekker (1995).

Dayan, P., and Abbott, L. *Theoretical neuroscience*, vol. 31. MIT press Cambridge, MA, 2001.

Hasslacher, B., and Tilden, M. *Living machines.* Robotics and Autonomous Systems 15, 1 (1995), 143 – 169.

Ikeguchi Lab. http://www.hisenkei.net/en/index.html.

Ikeguchi Lab youtube videos. http://www.youtube.com/user/ikeguchi-lab/videos.

Lakens, D. *Movement synchrony and perceived entitativity.* Journal of Experimental Social Psychology 46, 5 (2010), 701 – 708.

Lambert, A. *A stigmergic model for oscillator synchronization and its application in music systems.* In ICMC 2012 (2012).

Lizier, J. T. Jidt. *An information-theoretic toolkit for studying the dynamics of complex systems.* arXiv preprint arXiv:1408.3270 (2014).

Nake, F. *On the impossibility of avoiding aesthetics in human-computer interaction.* In Dagstuhl Seminar Proceedings (2008), Schloss Dagstuhl-Leibniz-Zentrum für Informatik.

Néda, Z., Ravasz, E., Brechet, Y., Vicsek, T., and Barabasi, A.-L. *Self-organizing processes: The sound of many hands clapping*. Nature 403, 6772 (2000), 849 – 850.

Scherffig, L., and Trogemann, G. *Über das Warten beim Rechnen – Synchronisationsstrategien in parallelen und interaktiven Systemen.* Kulturtechniken der Synchronisation (2013), 231 – 267.

Schreiber, T. *Measuring information transfer.* Physical review letters 85, 2 (2000), 461.

Wiener, N., and Rosenblueth, A. *The mathematical formulation of the problem of conduction of impulses in a network of connected excitable elements, specifically in cardiac muscle.* Archivos del instituto de Cardiologia de Mexico 16, 3 (1946), 205.