Model Checking Biological Oscillators

Emanuela Merelli
Joint work with

Ezio Bartocci   Flavio Corradini   Luca Tesei

Department of Mathematics and Computer Science
University of Camerino - Italy

HSCB 2009, April 16^{th} 2009, San Francisco

Outline

Fireflies flashing in unison - Malaysia

Emerging Synchronization

- Spontaneous synchronization happens frequently in nature
- Electrons flowing, Fireflies flashing, Pacemaker cells firing, Crickets chirping, Planets orbiting, Neurons firing, Menstrual periods synchronizing, $\cdots$
- These phenomena has been studied by biologists, physicists, mathematicians, astronomers, engineers, sociologists
- They could be surprising, incredible, or driven by chance, but if there are the right conditions synchronization is actually inevitable!
- All phenomena have in common that at their base there are autonomous entities that exhibits a cyclic behavior: **oscillators**

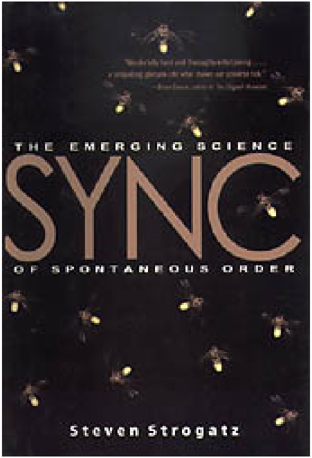
Coupled Oscillators

- Two or more oscillators are said to be coupled if some physical or chemical process allows them to influence one another
- Nature uses every available channel to make the oscillators interact
- The result of these interactions is often synchrony, in which all the oscillators begin to move as one
- Pulse-coupled oscillators interact during all individual firing
- Smooth-coupled oscillators continuously interact

History hints

- During the last century several studies have been done on the subject of synchronization
- Norbert Wiener, Arthur Winfree, Charles Peskin, Yoshiki Kuramoto, $\cdots$ defined mathematical models to capture the behavior of the single oscillators and the way they interact
- There exists mathematical proofs of inevitable synchronization, given the right conditions
- However, the working models make several simplifications, and there are difficult open problems

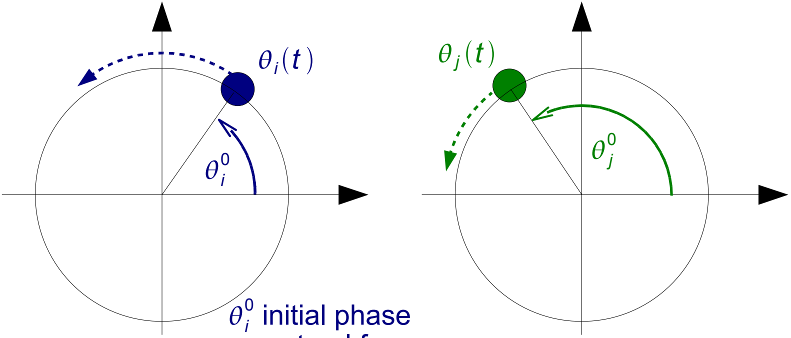
Emerging Synchronization



Our aim for this work

- Approach the collective synchronization with a Computer Science point of view
- Apply existing or new techniques and results in the field of heart arrhythmias: cardiac myocytes are oscillators
- This work is the first step towards the definition of regulation strategies for the treatment of heart fibrillation (a very larger project we are involved in)

Phase Oscillators



θ_i^0 initial phase
ω_i natural frequency

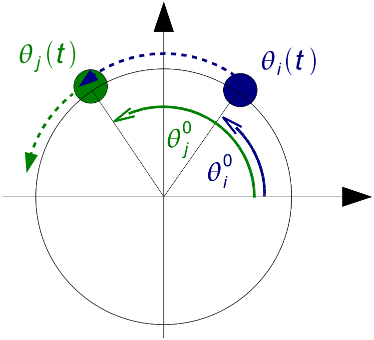
$$\theta_i(t) = \omega_i t + \theta_i^0$$

$$\dot{\theta}_i = \omega_i \qquad \ddot{\theta}_i = 0$$

$$\theta_j(t) = \omega_j t + \theta_j^0$$

$$\dot{\theta}_j = \omega_j \qquad \ddot{\theta}_j = 0$$

Coupled Oscillators



If there is an interaction, how do they interact?
What happens if there are N interacting oscillators?

The Kuramoto's Model of Synchronization

- Kuramoto solved the problem for N interacting "smooth" oscillators
- They continuously interact by accelerating or decelerating
- The modification of the speed of each oscillator is a function of the current position of all the others
- If the given conditions are met they eventually synchronize
- The interaction also depends on a constant K that represents the "strength" of the communication

The Kuramoto's Model of Synchronization

By Kuramoto, the angular speed of the *i*-th oscillator is modified in this way:

$$\dot{\theta}_i = \omega_i + \frac{K}{N} \sum_{j=1}^{N} \sin(\theta_j - \theta_i)$$

The Kuramoto's Model of Synchronization

- The interacting dynamics can result in synchronization or not, depending on the initial conditions and on the parameters
- A primary basic condition: the natural frequencies of the N oscillators are *equal* or "*not too different*"

(chosen from a lorentzian probability density given by:

$$g(\omega) = \frac{\gamma}{\pi[\gamma^2 + (\omega - \omega_0)^2]}$$

where γ is the width and ω_0 is the median)

The Kuramoto's Model of Synchronization

- Kuramoto provided an equivalent measure of synchronization by defining r and ψ such that:

$$re^{i\psi} = \frac{1}{N}\sum_{j=1}^{N} e^{i\theta_j}$$

- r represents the phase-coherence of the population of oscillators
- If the oscillators are in sync, $r \approx 1$
- If the oscillators are completely out of phase with respect to each other the value of r is close to 0

Discretization

Fixed *dt* small enough:

$$\theta_i(t + dt) = \theta_i(t) + (\omega_i + \alpha_i(t))\, dt$$

where

$$\alpha_i(t) = \omega_i + \frac{K}{N} \sum_{j=1}^{N} \sin(\theta_j(t) - \theta_i(t))$$

we can simulate the dynamics as a sequence of discrete steps

Timed Automata

- A well-known formalism to express quantitative timed behaviors
- They use clock variables to measure time passing (dense time domain, typically $\mathbb{R}^{\geq 0}$)
- The timed traces of the automata are all the possible behaviors
- We define a subclass called Oscillator Timed Automata suitable to represents phase oscillators

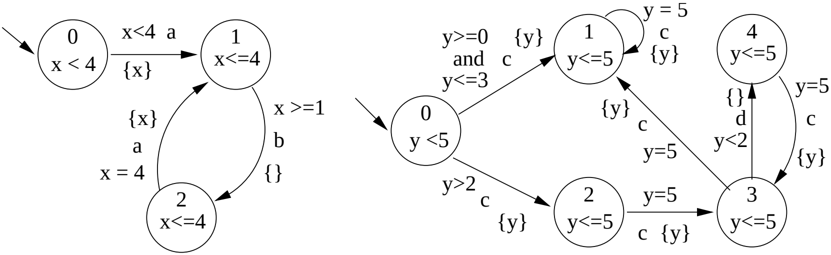
Oscillator Timed Automata

- There is a **distinguished action** that identifies the cyclic behavior: it is executed regularly over time
- The natural frequency ω_i becomes a **period** p_i, $\omega_i = \frac{2\pi}{p_i}$
- The initial phase θ_i^0 becomes an **initial delay** ϑ_i, $\theta_i^0 = \frac{2\pi}{p_i}\vartheta_i$
- We require that in every trace the distinguished action is executed every period
- No restriction on the structure or the complexity of the automata

E. Bartocci, F. Corradini, E. Merelli, L.Tesei *Model Checking Biological Synchronization* In the 2nd *From Biology To Concurrency International* Workshop (FBTC'08), Reykjavik, Island, July, 2008 - ENTCS 17341.

Two Oscillator Timed Automata

(a) Distinguished action: *a*, Period 4
(b) Distinguished action: *c*, Period 5

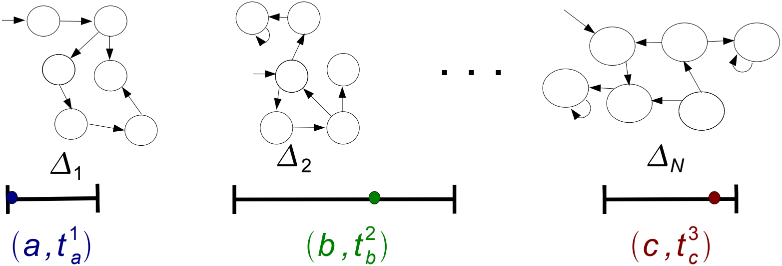


(a) (b)

Interaction Semantics

- We consider steps of activity of a Timed Automaton in which it lets Δ time units elapse
- We put in parallel (only interleaving) N oscillator timed automata T_i
- We consider a generic function *Interact* giving the **relative** duration Δ_i, w.r.t. dt, of a step of activity of T_i
- Δ_i depends on the chosen dt and on the current states of the T_is
- The semantics is a sequence of steps of activity whose action timestamps are *rescaled* to dt
- **The behaviors of the automata are perturbed over time, but their structure remains the same**

Relative times and rescaling



Relative Times:

$\Delta_i = Interact(i, dt, s)$

dt

Absolute Time

$(x, \frac{t_x^i}{\Delta_i} \cdot dt)$ Rescaling

Interaction Semantics

$$\frac{\forall i = 1, 2, \ldots N \quad \Delta_i = \mathcal{I}(i, dt, s_1, s_2, \ldots, s_N) \qquad s_i \xrightarrow[\Delta_i]{\ell_i} s_i' \quad \lambda_i = \mathcal{SC}(\mathcal{A}(\ell_i), \Delta_i, dt)}{\langle s_1, s_2, \ldots, s_N \rangle \xrightarrow[dt]{\lambda^1, \lambda^2, \ldots, \lambda^N} \langle s_1', s_2', \ldots, s_N' \rangle}$$

where $\mathcal{SC}$ is the re-scaling function:

$$\mathcal{SC}((a_0, t_0)(a_1, t_1) \cdots (a_k, t_k), \Delta, dt)$$

$$=$$

$$(a_0, \frac{t_0}{\Delta} \cdot dt)(a_1, \frac{t_1}{\Delta} \cdot dt) \cdots (a_k, \frac{t_k}{\Delta} \cdot dt)$$

Kuramoto Interaction function

- We add to each automaton T_i a new clock x_i measuring the time passed from the last occurrence of the distinguished action a_i
- let u_i be the current value of clock x_i
- $d_i = p_i - u_i$ is the time to elapse before the next a_i appears
- we have that $\theta_i(t) = \frac{u_i}{p_i} \cdot 2\pi$, modulo $2\pi h$ for some $h \in \mathbb{N}$
- in each instant t we can calculate, from these values, the value of the acceleration α_i, according to the Kuramoto model, to apply to T_i in the current dt step

KSL: Kuramoto Synchronization Logic

- Based on Linear Time Logic
- Models of the formulas are not Kripke structures, but (in)finite sequences of steps of activity of duration *dt*
- Suitable to express typical synchronization properties about populations of coupled oscillators

Atomic Propositions

$$p \quad ::= \quad \sum c_i v_i \sim b \mid r \sim b$$

where $c_i, b \in \mathbb{R}$, $\sim \in \{<, \leq, >, \geq, =\}$, r is the phase-coherence, and v_i are real variables measuring either "freezed" times $d_i^{(h)}$ or remaining times d_i to the next distinguished actions

Syntax

Formulas

$$\phi \quad ::= \quad T \mid F \mid p \mid \neg\phi \mid \phi \wedge \phi \mid X\phi \mid \phi \, U_{\prec m} \, \phi \mid D^{(h)}.\phi$$

where $\prec \in \{<, \leq\}$

- Until operator is always bounded (actually it is derivable from the Next operator)
- Usual shorthands can be defined: $\diamond_{\prec m}, \Box_{\prec m}, \Rightarrow, \cdots$

Semantics

- Structure $\mathcal{M} = (\otimes, \times, \mathcal{D}, \mathcal{D}^{(*)}, K, r)$
- $\otimes = \{\omega_1, \cdots, \omega_n\}$ is the vector of *frequencies* (periods) of the oscillator timed automata
- $\times = \{\theta_1, \cdots, \theta_n\}$ is the vector of the *initial phases* (initial delays)
- $\mathcal{D} = \{d_1, \cdots, d_n\}$ is the vector of the *remaining times* to accomplish the distinguished actions
- $\mathcal{D}^{(*)}$ are the set of vectors of the *stored remaining times* during checking/simulation
- $K \in \mathbb{R}$ is the *interaction constant* between the oscillators automata
- r is the *phase coherence* calculated at current time

Semantics

$$
\begin{aligned}
\mathcal{M}, t \models T & \Leftrightarrow \mathcal{M}, t \not\models F \\
\mathcal{M}, t \models \sum_i c_i \cdot v_i \sim b & \Leftrightarrow \sum_i c_i \cdot \mathcal{M}(v_i) \sim b \\
\mathcal{M}, t \models r \sim b & \Leftrightarrow \mathcal{M}(r) \sim b \\
\mathcal{M}, t \models \neg\phi & \Leftrightarrow \mathcal{M}, t \not\models \phi \\
\mathcal{M}, t \models \psi \wedge \phi & \Leftrightarrow \mathcal{M}, t \models \psi \text{ and } \mathcal{M}, t \models \phi \\
\mathcal{M}, t \models X\, \phi & \Leftrightarrow \mathcal{M}, t+1 \models \phi \\
\mathcal{M}, t \models \psi\, U_{\prec m}\, \phi & \Leftrightarrow \exists s_2 : 0 \leq s_2 \prec m \text{ such that} \\
& \qquad \mathcal{M}, t+s_2 \models \phi \text{ and} \\
& \qquad \mathcal{M}, t+s_1 \models \psi \text{ for } s_1 = 0, \cdots, s_2 - 1 \\
\mathcal{M}, t \models D^{(h)}.\phi & \Leftrightarrow \mathcal{M}_{[\mathcal{D}^{(*)} := \mathcal{D}^{(*)} \cup \{d_1^h(t), \cdots, d_n^h(t)\}]}, t \models \phi
\end{aligned}
$$

Example

- Using the phase-coherence parameter r, it is possible to measure the collective behavior of a population of coupled oscillators
- Kuramoto showed that if the interaction constant K is greater than a certain threshold K_c **and** the oscillators have the same frequency, after a certain amount of time they become perfectly synchronized, i.e. $r = 1$

Given a population of N oscillator timed automata with same frequencies, "they become perfectly synchronized within 10s"

$$\phi_{psynch} = \Diamond_{\leq 10s}\, r = 1$$

Example

- If the oscillators have slightly different frequencies and K is greater than a certain threshold K_c, the phase-coherence parameter r becomes approximately 1
- Given a system of N oscillator timed automata with slightly different frequencies we can choose an $\epsilon > 0$ small enough

"within 10s, they become synchronized with an approximation of ϵ"

$$\phi^{\epsilon}_{psynch} = \Diamond_{\leq 10s} \Box_{\leq 5s} \, r > 1 - \epsilon$$

Example

After a system of oscillators starts to synchronize, the population may split into:

- a synchronized group, called *locked*, in which all oscillators advance with the average frequency
- a desynchronized group, called *drifted*, whose natural frequencies are too extreme to be entrained

Given a set F of indexes, "the oscillators in F become locked within 10s"

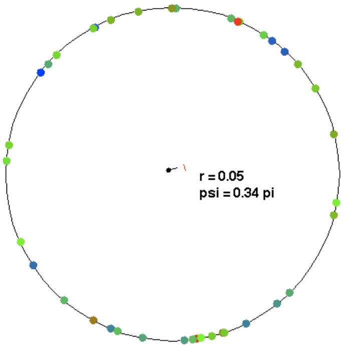
$$\phi^F_{locked} = \Diamond_{\leq 10s} D^{(1)}.X \bigwedge_{i,j \in F} (d_i - d_j) - (d_i^{(1)} - d_j^{(1)}) = 0$$

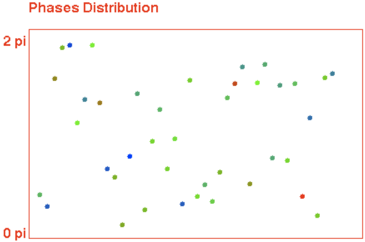
In a set of locked oscillators no changes happen to the relative remaining times in two subsequent simulation steps

Model Checking

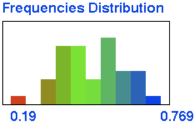
- The algorithm follows recursively the structure of the formula to be checked
- All Until operators are simulated with Next
- When a Next operator is visited a step of activity is performed and $\mathcal{M}$ is updated
- When a $D^{(h)}.$ operator is visited the current values of d_i are stored in $\mathcal{M}$ with the index h
- A diagnostic trace is returned if the formula is false
- The complexity of the algorithm is linear in the length of the formula to be checked

Simulator / Model Checker

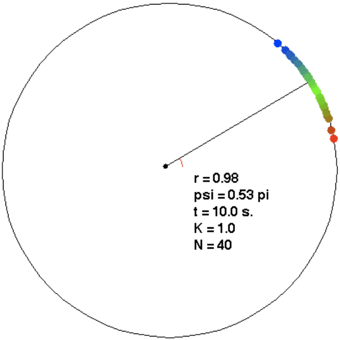


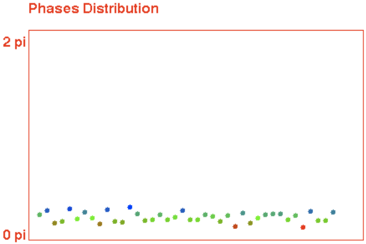


locked[all] = False
synchronized~[0.3] = False

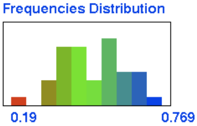


Simulator / Model Checker





locked[all] = True
synchronized~[0.3] = True



Summary

- We have identified a subclass of Timed Automata to model oscillators
- We have defined a generic interaction semantics for describing their collective behavior
- We have instantiated the semantics to the Kuramoto model of synchronization for smooth coupled oscillators
- We have introduced the logic KSL to express properties of populations of oscillators
- We have provided a model checker for KSL

Ongoing work

- Implementation of other models of synchronization, e.g. Peskin's model for **pulse** coupled oscillators
- Relaxing the strict (punctual) definition of oscillating behavior by allowing the distinguished actions to occur in an **interval** of time
- Improvement the prototype of the simulator/model checker. A demo is at `http://cosy.cs.unicam.it/kuramoto/`

Future work

KSL is a good starting point towards the definition of **regulation strategies** on populations of oscillators:

- We could insert artificial oscillators, which we can control, in the population
- We specify our desired property
- A properly modified model checking procedure might calculate the behaviors of the artificial oscillators to make the system satisfy the desired property

We are working on an application in cardiac myocytes networks to treat arrhythmia-related diseases (Radu, please could you explain our StonyCam project)

Prospective Work with Hybrid Automata

- Define the interaction semantics among **Hybrid Automata**

Open Issues

- could be I/O Hybrid Automata another way to formalize this population of oscillators?
- could the semantics be parameterized?
- what could be the added value in using this formalization? (being the Simulator/Model Checker in any case discretized and due to the undecidability for Hybrid Automata
- from theoretical point of view can be interesting to compare the two approches on defining semntics?

Anyone interested in collaborating with us is invited to contact us and visit our laboratory in Camerino ...

Thank you!


University of Camerino
UNICAM
Università di Camerino
1336
Thanks!

Acknowledgements

LitBio: Laboratory of Interdisciplinary Technologies in Bioinformatics `http://www.litbio.org/`

CoSy Research group: `http://cosy.cs.unicam.it`