Computational Self-Awareness in Musical Robotic Systems

David Thorvaldsen

Thesis submitted for the degree of
Master in Informatics: Robotics and Intelligent Systems
60 credits

Institute for Informatics
Faculty of mathematics and natural sciences

UNIVERSITY OF OSLO

Spring 2022

Computational Self-Awareness in Musical Robotic Systems

David Thorvaldsen

Computational Self-Awareness in Musical Robotic Systems

http://www.duo.uio.no/

Printed: Reprosentralen, University of Oslo

Abstract

MSc Thesis/Project Summary.

Acknowledgements

Tusen takk til alle som har støttet og hjulpet meg. Sånn faktisk.

Contents

1 Introduction **1**
- 1.1 Motivation . . . 1
- 1.2 Goal of the thesis . . . 1
- 1.3 Outline . . . 1

2 Background **2**
- 2.1 Taking inspiration from natural phenomena . . . 2
- 2.2 Oscillators and oscillator-synchronization . . . 3
  - 2.2.1 Phase-adjustment . . . 3
  - 2.2.2 Frequency-adjustment . . . 3

3 Tools and software **4**

4 Implementation **5**
- 4.1 The musical multi-robot collective . . . 5
  - 4.1.1 The node: the musical robot individual . . . 6
  - 4.1.2 Robot communication: the "fire"-signal . . . 6
  - 4.1.3 System target state: harmonic synchrony . . . 7
- 4.2 Synchronizing oscillator-phases . . . 7
  - 4.2.1 Mirollo-Strogatz's "standard" phase-updates . . . 7
  - 4.2.2 K. Nymoen's bi-directional phase-updates . . . 8
- 4.3 Synchronizing both oscillator -phases and -frequencies . . . 9
  - 4.3.1 Person X's low SA-level frequency-updates . . . 10
  - 4.3.2 K. Nymoen's mid SA-level frequency-updates . . . 10
  - 4.3.3 Thorvaldsen's high SA-level frequency-updates . . . 14
- 4.4 Performance measure . . . 14

5 Experiments and results **15**

6 Conclusions **16**

List of Tables

List of Figures

2.1 Synchronous fireflies at Congaree National Park, United States [1]. 2

4.1 Decentralized synchronization of phases achieved in a musical robot collective, consisting of M. J. Krzyzaniak and RITMO's Dr. Squiggles. 6
4.2 "Standard" Phase-Adjustment with Mirrollo-Strogatz's approach 8
4.3 Bi-directional Phase-Adjustment with K. Nymoen et al.'s approach 9
4.4 Error Measurement (4.4) plotted as a function of Phase 11

GJØR: [Finn ut hvordan man endrer Figure-navnene her til noe annet enn det de har som caption i hoved-teksten].

Chapter 1

Introduction

1.1 Motivation

Grunnene til å studere effektene av selv-bevissthet.

De viktigste bidragene av arbeidet summert (for å få det til å stå frem/ut bedre enn å bare "gjemme" det i den siste delen av oppgaven).

1.2 Goal of the thesis

BESKR.: ["to make the reader better understand what the thesis is about"—Jim, og "en rød tråd?"—Sigmund].

Spesifikke mål (goals/aims) med master-oppgaven/-prosjektet. Hva jeg vil vise/demonstrere til folk (leserne f.eks.) om self-awareness (SA) og hvordan.

Research Question 1:
Will performance in collective multi-robot systems increase as the level of Self-Awareness increases? Specifically, will increased levels of Self-Awareness in the individual agents/musical robots lead to the collective of individuals being able to synchronize to each other faster than with lower levels of Self-Awareness?

Research Question 2:
Will increased levels of Self-Awareness lead to more robustness and flexibility in terms of handling environmental noise and other uncertainties — specifically in the continued ability of musical robots to synchronize to each other efficiently despite these difficulties/challenges?

1.3 Outline

En fin (Eagle's-eye) oversikt over strukturen til hele dette dokumentet fra nå av, og utover.

Chapter 2

Background

Kulepunkter (bulletpoints) fra mulige inspirasjoner og referanser (pga. "Write a few lines summarizing relevant articles one comes across (which one is likely to refer to in the final report)" — Jims master-skrivingsdokument).

2.1 Taking inspiration from natural phenomena

The intriguing, diverse, and complex phenomena of nature have for long served as exciting inspirations to human engineers and researchers [ant-colonies, boids, swarms, beeclust]. One other such phenomena, studied and modelled, is the synchronously firing fireflies in forests, as can be seen an example of in Figure 2.1.

Figure 2.1: Synchronous fireflies at Congaree National Park, United States [1].

This phenomenon of fireflies synchronizing to each other has inspired scientists like Mirollo & Strogatz [2] and in later time Kristian Nymoen et al. [3], to model and replicate this natural phenomenon in human-engineered systems. It turns out that this is not a biological, but mathematical, problem. Given the periodic and repeating nature of the flashing/firing of the fireflies, modelling a firefly has been done by looking at each firefly as a periodic signal or oscillator. This work [2, 3] then ties into the broader work on synchronizing oscillators which has been subject to study for some time now []. What separates

Mirollo-Strogatz and K. Nymoen's approaches from these other and previous oscillator-synchronizing methods, is mainly that here the oscillators are *pulse-coupled* (which the fireflies also can be said to be), as opposed to the more "standard" and constraining *phase-coupled* oscillators.

2.2 Oscillators and oscillator-synchronization

GJØR: [Beskriv dette så godt at du kan snakke fritt om oscillatorers **faser** og **frekvenser** senere (i Implementation f.eks.), spesielt i tilfelle for noen ikke har vært borti det før, eller tatt et Signalbehandlings-kurs].

GJØR: [Skill på Pulse-coupled Oscillators, og Phase-coupled Oscillators].

Much of the terminology from [3] is used here. An oscillator i is characterized by its *phase* $\phi_i(t)$, which is—at the start of its periodic signal period—initialized to 0 and evolves towards 1 at a rate of $\frac{d\phi_i(t)}{dt}$ — which is also called the *frequency* of the oscillator. When oscillator i's phase is equal to 1 (i.e. when $\phi_i(t) = 1$, or when the periodic signal period is over), we say oscillator i has *phase-climaxed*.

2.2.1 Phase-adjustment

(If relevant and wanted)
Previous approaches to Phase-synchronization in oscillators (pulse- and/or phase-coupled).

2.2.2 Frequency-adjustment

(If relevant and wanted)
Previous approaches to Frequency-synchronization in oscillators (pulse- and/or phase-coupled) [fixed_freqs, fixed_range_freqs] where the oscillators's frequencies are either equal and fixed, or where frequencies are bound to initialize and stay within a fixed interval/range.

Chapter 3

Tools and software

BESKR.: ["which describes what you've used" — ish Kyrre].

BESKR.: [Kan flyttes til en egen seksjon hvis dette kapittelet ikke ville vært så stort (jf. 'ThesisChecklist' på Robin-wikien)].

BESKR.: [(Hentet fra Tønnes sin master, om Tools and engineering) En introduksjon til de forskjellige verktøyene og prosessene brukt iløpet av masteroppgaven. Fokuser på fysisk arbeid gjort, og ingeniør-delene av masteroppgaven, inkludert 3D-design av de fysiske robotene, valg av deler, simulering i systemer, og testingen, valideringen, og verifikasjonsmetoder brukt i oppgaven. Gjerne også en oversikts-tabell av verktøy og programvare brukt].

Software:

- Notepad++ v8.1.9.2 (64 bit). For writing my master's thesis and code.
- Unity Version 2021.2.0f1. Unity is originally a game-development platform, but can also be used to make **INKL.:** [realistic]**?** simulations containing physical rigid-bodies using the ¡bla.bla Rigidbody¿-physics engine.
- Python 3.10.0. For plotting and analysing data.

Chapter 4

Implementation

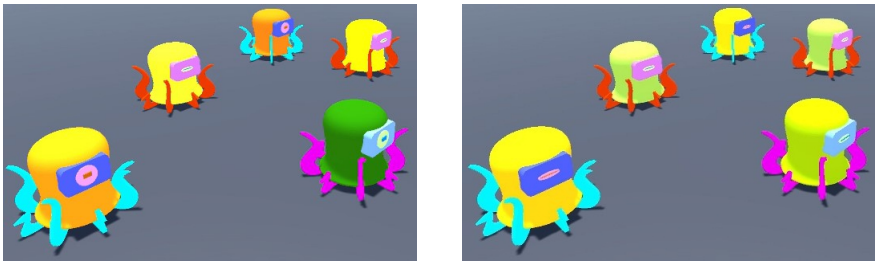
This chapter gives an overview of the developed musical multi-robot system. The main goal of the implemented system is to allow for a multi-robot (musical) collective to interact with each other in order to achieve emergent and coordinating/co-operative behaviour—synchronization specifically in our case—with varying degrees of difficulty and certainty in the environment and communication. More specifically, the goal with the design is to enable the robot collective to achieve so-called *harmonic synchronization* within a relatively short time. What is meant by *harmonic synchronization* will be expounded in Subsection 4.1.3. These goals firstly require of the agents/nodes the modelling of oscillators with their properties, like phase and frequency, as explained further in Subsection 4.1.1. To allow for interaction and communication between the agents, mechanisms so that the agents can transmit "fire"-signals, as well as listen for other agents's "fire"-signals, is necessary as well, and is presented in Subsection 4.1.2.

First, the system and the system components will be presented and introduced. Then, methods implemented for achieving the system target goal of *harmonic synchrony* in various synchronization objectives—firstly solely for oscillator-phases, then secondly for both oscillator-phases and oscillator-frequencies—will be described and presented.

4.1 The musical multi-robot collective

BESKR.: [about introducing, giving an overview, and describing the developed system (as well as the system sub-constituents — aka. what the system consists of). $\Longrightarrow$ In a nutshell answering: What is the system, and what does it consist of? Why?].

Envision that we have a multi-agent collective scenario consisting of musical robots modelled as oscillators, solely communicating through brief "fire"-like audio-signals—greatly inspired by K. Nymoen et al.'s synchronizing "fireflies" [3]. These agents are initially not synchronized in their firing of audio-signals, but as time goes, they are entraining to synchronize to each other by adjusting their phases and frequencies when or after hearing each other's audio-signals. An illustration of this is given in Figure 4.1.



(a) The agents firing asynchronously at first. Here, only the two Dr. Squiggles with red tentacles are firing simultaneously, but the rest are not.

(b) Seconds later, after having listened to each other's fire-event signals and adapted themselves accordingly, the agents are here firing synchronously.

Figure 4.1: Decentralized synchronization of phases achieved in a musical robot collective, consisting of M. J. Krzyzaniak and RITMO's Dr. Squiggles.

These aforementioned audio-signals to be expounded further in Subsection 4.1.2, also referred to as "fire"-signals, are transmitted whenever an agent's oscillator *peaks* (i.e. after its cycle or period is finished, having phase $\phi(t) = 1$)—or actually every second *peak*, due to the target system goal of *harmonic synchrony*. All agents have the ability to listen for such transmitted "fire"-signals from their neighbours, which they then will use as a trigger to adjust themselves according to some well-designed update-functions to be elaborated upon in Section 4.2 and 4.3.

4.1.1 The node: the musical robot individual

BESKR.: [about the individual node/agent with all its attributes and characteristics etc. (as e.g. an oscillator-component (cf. I.A, III.Intro., and 'Implementation' in Nymoen's firefly-paper))].

Notions like "agent", "node", "firefly", and "oscillator" will be used interchangeably throughout the thesis.

4.1.2 Robot communication: the "fire"-signal

No individual agent is able to adjust or modify the state of any other agent, only its own, and the only means of communication between the agents will be through these "fire"-signals.

- Short and impulsive audio sound/signal, representing a node's phase-climax and "fire"-/"flash"-event.
- Stigmergic co-ordination/communication.
- Facilitating PCO (pulse-coupled oscillators), different from phase-coupled oscillators with their differences.
- Aspekter fra Nymoens paper som presentert i Essay-oppsummeringen min.

4.1.3 System target state: harmonic synchrony

- The goal and target state of the system
- All agents/nodes having integer-relation frequencies (between each other), so that given a fundamental frequency $\omega_{i,0}$ for the agent i with the lowest frequency in the collective — all other agents have "legal" harmonic frequencies; meaning, all nodes have frequencies $\in \omega_{i,0} \cdot 2^{\mathbb{N}_0}$.
- Waveform-Spektrogram av en stemme eller en lyd med Harmonics i seg (f.eks. fra Prariedogs-fyren ellernoe) — som jeg kan sammenlikne med de musikalske nodenes Frekvens-Spektrogram ved siden av. Hvorfor? For å introdusere/presentere/sammenlikne/forklare target-goalet best mulig.

4.2 Synchronizing oscillator-phases

BESKR.: [about various methods/algorithms/approaches (the nitty gritty) implemented to achieve the target/goal state of harmonic synchrony. $\implies$ In a nutshell answer: What different but comparable methods did you use to achieve harmonic synchrony with Phase-Synchronization?].

If we first assume constant and equal oscillator-frequencies in our agents, we can take a look at how the agents adjust their—initially random—phases in order to synchronize to each other. This is then in contrast to the case in Section 4.3 where heterogenous and randomly initialized oscillator-frequencies in the musical agents are used and synchronized.

The goal state of the agents is for now to fire/flash simultaneously, after having started firing/flashing at random initially. Note that this is a different goal from the final and ultimate goal of *harmonic synchrony*. This is actually not true — as firing/flashing simultaneously technically would be considered having achieved harmonic synchrony. Hence, this "sub-goal" is actually a sufficient and weaker goal w.r.t. the main goal of harmonic synchrony.

In order for the musical agents to synchronize to each other, they will have to—due to their heterogenous and randomly initialized phases—adjust or update their own phases according to some well-designed update-/adjustment-functions, as presented below.

When it comes to the temporality and timing of when these updating functions are used and applied; Musical agents's phases get updated/adjusted immediately as "fire"-/"flash"-events from neighbouring robots are perceived.

4.2.1 Mirollo-Strogatz's "standard" phase-updates

One approach having been used to achieve this in the past is Mirollo-Strogatz's "Standard" Phase-adjustment in oscillators [2], as illustrated in Figure 4.2.

Each musical node gets a new phase, $\phi(t^+) = P(\phi(t))$, accoring to the **phase update function** (4.1) upon perceiving a "fire"-event from one of the other musical nodes:

$$P(\phi(t)) = (1 + \alpha)\phi(t), \tag{4.1}$$

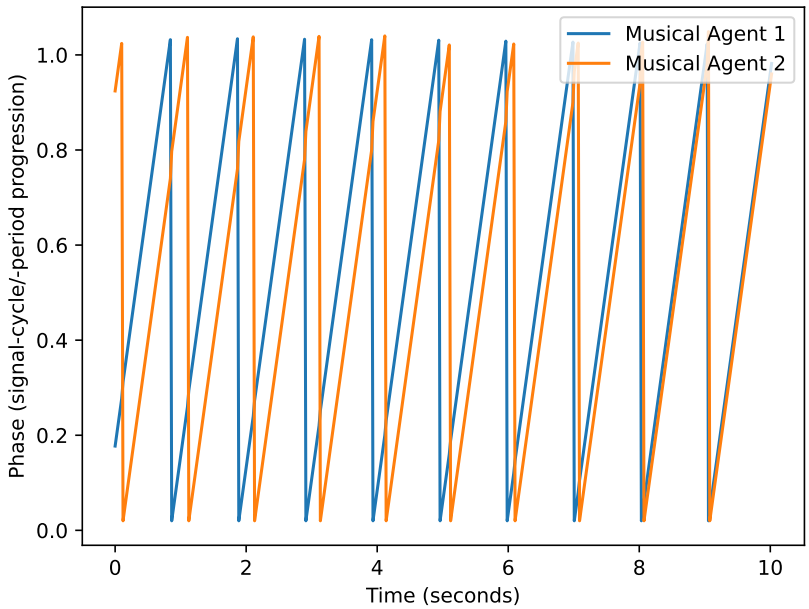


Figure 4.2: "Standard" Phase-Adjustment with Mirrollo-Strogatz's approach

where "α *is the pulse coupling constant, denoting the strength between nodes*" [3], and t^+ denotes the time-step immediately after phase-climax. So, if $\alpha = 0.1$ e.g., then a musical node's new and updated phase, immediately after hearing a "fire"-signal from another node, will be equal to $\phi(t^+) = P(\phi) = (1 + 0.1)\phi = 1.1\phi$. 110% of its old phase ϕ, that is. Hence, and in this way, the node would be "pushed" to fire sooner than it would otherwise (as nodes fire once they have reached phase-climax $\phi = 1$).

4.2.2 K. Nymoen's bi-directional phase-updates

This Phase-adjustment, as in Figure 4.3, works very similarly to the Phase-adjustment performed in the "standard" ***Mirollo-Strogatz*** approach presented earlier; the only difference being that now, nodes update their phases with the slightly more complex **phase update function** (4.2) when hearing a "fire"-event from one of the other musical nodes — allowing for both larger, but also smaller, updated phases:

$$P(\phi) = \phi - \alpha \cdot sin(2\pi\phi) \cdot |sin(2\pi\phi)| \tag{4.2}$$

The fact that new and updated phases can both be larger, but also smaller, compared to the old phases, is exactly what's meant by the Phase-adjustment being ***Bi-Directional***, or as the authors call it in the paper as using "*both excitatory and inhibitory phase couplings between oscillators*" [3].

The effects then of adjusting phases—upon hearing "fire"-events, according to this newest update-function (4.2)—are that the nodes's phases now get decreased if $\phi(t)$ is lower than 0.5, increased if $\phi(t)$ is higher than 0.5, and neither—at least almost—if the phases are close to 0.5. This is due to the negative

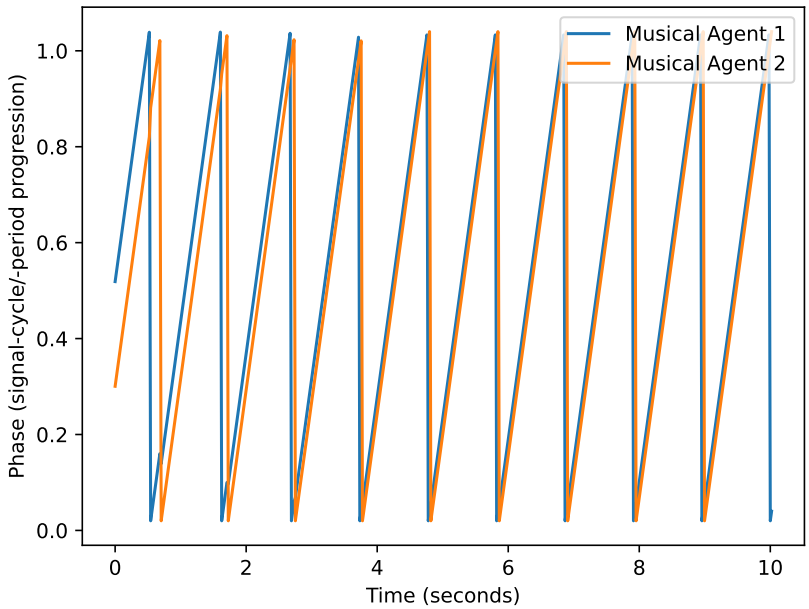


Figure 4.3: Bi-directional Phase-Adjustment with K. Nymoen et al.'s approach

and positive sign of the sinewave-component in Equation (4.2), as well as the last attenuating factor in it of $|sin(2\pi\phi)| \approx |sin(2\pi\frac{1}{2})| = |sin(\pi)| = |0| = 0$, then if we have $\phi(t) \approx 0.5 = \frac{1}{2}$.

4.3 Synchronizing both oscillator -phases and -frequencies

BESKR.: [about various methods/algorithms/approaches (the nitty gritty) implemented to achieve the target/goal state of harmonic synchrony. $\Longrightarrow$ In a nutshell answer: What different but comparable methods did you use to achieve harmonic synchrony with Frequency- **INKL.:** [& Phase-]**?**Synchronization?].

Now over to the slightly more complex issue of adjusting and synchronizing frequencies—as well as phases—in the agent-/oscillator-collective. We introduce randomly initialized, non-constant, and heterogenous oscillator-frequencies in our musical agents — allowing in the music collective the playing of rhythmic patterns like doubles, fourths, eights (FRA NYMOEN: subdivisions of a measure, i.e. quarter notes and 8ths. "Temporal components in music tend to appear in an integer-ratio relation to each other (e.g., beats, measures, phrases, or quarter notes, 8ths, 16ths)") etc. The agents are now then required to not only synchronize their phases to each other, but now also their initially different and random frequencies — so that frequencies are "legal" and *harmonically*

synchronized.

The goal state now will not be exactly equal frequencies and phases, but rather the goal state of *harmonic synchrony*.

In order for the musical agents to synchronize to each other, they will have to adjust or update their own phases and frequencies according to some well-designed update-/adjustment-functions. Three approaches/methods that was implemented to achieve this, and the goal state of *harmonic synchrony*, is presented now. Notice the increasing degree of *Computational Self-Awareness* endowed/utilized/included in the methods.

4.3.1 Person X's low SA-level frequency-updates

A simpler strategy/method for achieving Harmonic Synchronization including Frequeny-Adjustment, specifically with no Self-Awareness capabilities, if it exists.

4.3.2 K. Nymoen's mid SA-level frequency-updates

This approach to Frequency Adjustment stands in contrast to previous approaches to synchronization in oscillators [fixed_freqs, fixed_range_freqs] where the oscillators's frequencies are either equal and fixed, or where frequencies are bound to initialize and stay within a fixed interval/range.

In order to achieve this goal of *harmonic synchrony* in conjunction with—or rather through—frequency adjustment, we have to go through a few steps to build a sophisticated enough update-function able to help us achieve this.

When it comes to the temporality and timing of when these update functions are used and applied; Musical agents's phases get updated/adjusted immediately as "fire"-/"flash"-events are perceived, whereas agents's frequencies do not get updated until the end of their oscillator-cycle (i.e. when having a phase-climax $\phi(t) = 1$). This is also the reason why frequencies are updated discretely, not continuously. So-called H-values however, being "contributions" with which the frequencies are to be updated according to, are immediately calculated and accumulated when agents are perceiving a "fire"-/"flash"-event — and then finally used for frequency-adjustment/-updating at phase-climaxes.

Each agent i update their frequency, on their own phase-climax (i.e. when $\phi_i(t) = 1$), according to the frequency-update function $\omega_i(t^+)$:

$$\omega_i(t^+) = \omega_i(t) \cdot 2^{F(n)}, \tag{4.3}$$

where t^+ denotes the time-step immediately after phase-climax, $\omega_i(t)$ is the old frequency of the agent at time t, and $F(n) \in [-1, 1]$ is a quantity denoting how much and in which direction a node should update its frequency after having received its nth "fire"-signal.

This is how we obtain the aforementioned $F(n)$-quantity:

4.3.2.1 Step 1: the "in/out-of synch" error-measurement/-score, $\epsilon(\phi(t))$

Describing the error measurements at the n-th "fire"-event, we introduce an Error Measurement function.

The Error Measurement function (4.4), plotted in Figure 4.4, is calculated immediately by each agent i, having phase $\phi_i(t)$, when a "fire"-event signal from another agent is detected by agent i at time t.

$$\epsilon(\phi_i(t)) = sin^2(\pi\phi_i(t)) \tag{4.4}$$

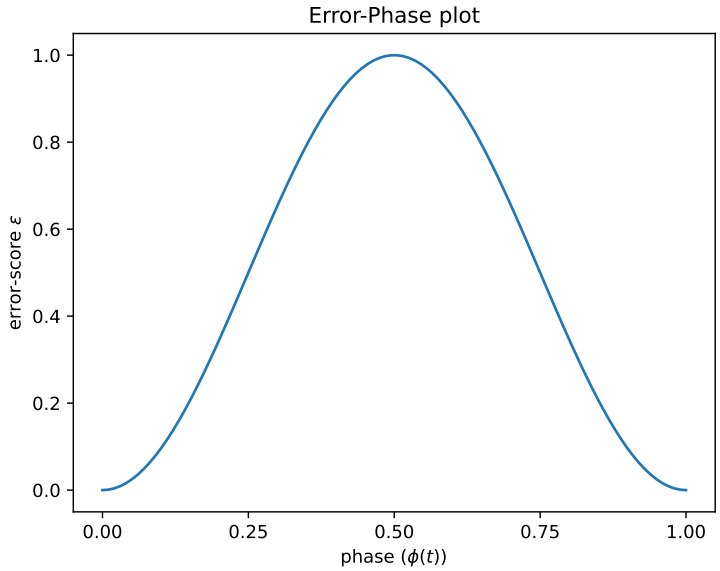


Figure 4.4: Error Measurement (4.4) plotted as a function of Phase

As we can see from this Error-Function, the error-score is close to 0 when the agent's phase $\phi_i(t)$ is itself close to 0 or 1 (i.e. the agent either just fired/flashed, or is about to fire/flash very soon). The error-score is the largest when an agent perceives a "fire"-signal while being half-way through its own phase (i.e. having phase $\phi(t) = 0.5$). We could also then ask ourselves, does not this go against the main/target goal of the system, being *harmonic synchrony* — if agents are allowed to be "half as fast" as each other?

We could imagine a completely "legal" and harmonically synchronous scenario where two agents have half and double the frequency of each other. The agent with half the frequency of the faster agent would then have phase $\phi(t) = 0.5$ when it would hear the faster agent "fire"/"flash" — leading to its Error-score $\epsilon(0.5) = sin^2(\pi/2) = 1$, which then makes it seem like the slower agent is maximally out of synch, when it is actually perfectly and harmonically synchronized. ??? **INKL.:** [This calls out for an attenuating mechanism in our frequency update function, in order to "cancel out" this contribution so that perfectly harmonically synchronized agents will not be adjusted further despite their high Error-measurement.]**?**

This Error-Measurement/-Score forms the basis and fundament for the first component of Self-awareness, being the *self-assessed synchrony-score* $s(n)$.

4.3.2.2 Step 2: The first self-awareness component, s(n)

This aforementioned self-assessed synchrony-score, $s(n)$, is in fact simply the median of a list (was originally in K. Nymoen et al.'s approach a running median filter over the discrete error measurement function) containing m Error-scores ϵ. Such a list is easily implemented in C# by declaring a List<float> called *errorBuffer* e.g. (i.e. *errorBuffer* is a list containing floating point values):

$$errorBuffer = \{\epsilon(n), \epsilon(n-1), ..., \epsilon(n-m)\}, \tag{4.5}$$

then leading to:

$$\begin{aligned} s(n) &= median(errorBuffer) \\ &= median(\{\epsilon(n), \epsilon(n-1), ..., \epsilon(n-m)\}) \in [0, 1], \end{aligned} \tag{4.6}$$

where n is the latest observed "fire-event", and m is the number of the last observed "fire"-events we would like to take into account when calculating the self-assessed synch-score.

If we then have a high $s(n)$-score, it tells us that the median of the k last error-scores is high, or in other words that we have mainly high error-scores — indicating that this agent is out of synch. Conversely, if we have a low $s(n)$-score, indicating mainly low error-scores for the agent — then we have an indication that the agent is in synch, hence leading to low error scores, and in turn low $s(n)$-scores.

In other words, we then have a way for each agent to assess themselves how much in or out of synch they believe they are compared to the rest of the agents. This is then the first degree/aspect of (public?) Self-awareness in the design.

4.3.2.3 Step 3: frequency update amplitude- & sign-factor, $\rho(n)$

Describing the amplitude and sign of the frequency-modification of the n-th "fire-event" received. It is used to say something about in which direction, and in how much, the frequency should be adjusted.

$$\rho(\phi) = -sin(2\pi\phi(t)) \in [-1, 1] \tag{4.7}$$

For example, if a node i has phase $\phi_i(t) = 1/4$, it gets a value $\rho(1/4) = -sin(\pi/2) = -1$; meaning, the node's frequency should be decreased (with the highest amplitude actually) in order to "slow down" to wait for the other nodes. Conversely, if a node j has phase $\phi_j(t) = 3/4$, it gets a value $\rho(3/4) = -sin(3/2\pi) = -(-1) = 1$; meaning, the node's frequency should be increased (with the highest amplitude) in order to getting "pushed forward" to catch up with the other nodes.

Acts as an attenuating factor, when $\phi(t) \approx 0.5$, in the making of the H-value — supporting the goal of *harmonic synchrony*.

4.3.2.4 Step 4: the H-value, and the H(n)-list

The following value, being "frequency-update-contributions", is then (as previously mentioned) calculated immediately when the agent perceives another agent's "flashing"-signal:

$$H(n) = \rho(n) \cdot s(n) \tag{4.8}$$

Here we then multiply a factor $\rho(n)$ representing how much, as well as in which direction, the node should adjust its frequency, together with a factor $s(n) \in [0, 1]$ of the adjusting node's self-assessed synch-score. To recall, the self-assessed synch-score $s(n)$ tells an adjusting node how in- or out-of-synch it was during the last m perceived "fire"-/"flash"-events — where $s(n) = 0$ signifies a mean of 0 in error-scores, and $s(n) = 1$ signifies a mean of 1 in error-scores. So then if this H-value is to be used to adjust the nodes's frequencies with, the frequency will then be adjusted in a certain direction and amount (specified by $\rho(n)$) — given that the node is *enough* "out of synch"/"unsynchronized" (in the case $s(n)$ is considerably larger than 0).

The H-value says something about how much "out of phase" the node was at the time the node's nth "flashing"-signal was perceived (and then followingly how much it should be adjusted, as well as in which direction after having been multiplied together with a sign-factor $\rho(n)$), given then that this H-value also consists of the *self-assessed synch score s(n)* — which again simply was the median of the list of error-scores, *errorBuffer* in our case.

We could look at this H-value as representing the direction and amplitude of the frequency adjustment weighted by the need to adjust (due to being out of synch) at the time of hearing "fire"-/"flash"-event n. Or in other words, this H-value is then the n-th contribution with which we want to adjust our frequency with.

Especially interesting cases are when we have $\phi(n) \approx 0.5 \implies \rho(n) \approx \pm 0$, as well as the last m Error-scores $\epsilon(n)$ being close to 0, also leading to $s(n) \approx 0$. In both of these two cases the entire frequency-adjustment contribution H would be cancelled out, due to harmonic synchronization (legally hearing a "fire"-/"flash"-event half-way through ones own phase) in the first case, and due to not being out of synch in the latter (having low Error-Measurements). Cancelling out the frequency adjustment contribution in these cases is then not something bad, but something wanted and something that makes sense. If these H-values then are cancelled out or very small, it is indicative of that nodes are already in *harmonic synchrony*, and hence should not be "adjusted away" from this goal state. On the other side, if these H-values then are different (e.g. closer to -1 and 1), it is indicative of that nodes are not yet in *harmonic synchrony*, and that they hence should be "adjusted closer" to the goal state.

All the calculated H-values are in my implementation accumulated and stored in an initially empty C#-list (of floats), referred to as $H(n)$, at once they are calculated. The $H(n)$-list is then consecutively "cleared out" or "flushed" when its H-values have been used for the current cycle/period's frequency adjustment (i.e. at the phase climax, when $\phi(t) = 1$), and is then ready to accumulate new H-values during the next cycle/period.

4.3.2.5 The final step: the frequency update function, $\omega_i(t^+)$

Putting it all together.

When an agent i then has a phase-climax ($\phi_i(t) = 1$), it updates its frequency to the new $\omega_i(t^+)$ accordingly:

$$\omega_i(t^+) = \omega_i(t) \cdot 2^{F(n)}, \tag{4.9}$$

where t^+ denotes the time-step immediately after phase-climax, and $F(n)$ is found by:

$$F(n) = \beta \sum_{x=0}^{k-1} \frac{H(n-k)}{k}, \tag{4.10}$$

where β is the frequency coupling constant, k is the number of heard/received "fire-event"s from the start of the last cycle/period to the end (i.e. the phase-climax, or *now*) — and the rest of the values are as described above.

This $F(n)$-value then, as we see in Equation (4.10), is a weighted average of all the node's $H(n)$-values accumulated throughout the node's last cycle.

4.3.3 Thorvaldsen's high SA-level frequency-updates

Min nye proposede algoritme for å oppnå Harmonic Synchronization med Frequency-Adjustment, som inneholder flere tilleggs- Self-Awareness-komponenter (e.g. Belief-awareness og/eller Expectation-awareness), sammenliknet med K. Nymoens tilnærming (jf. **Thorvaldsen's improved approach** og det jeg og Kyrre snakka om under 'Mid-November'-recall-notatet på 'reMarkable - Meetings - Kyrre' på reMarkablen). Jeg vil "beste" Kristian.

4.4 Performance measure

BESKR.: ["where I present the method used to evaluate the performance (detecting harmonic synchrony) of the presented and newly proposed algorithms/methods" — Samuelsen's MSc-thesis?].

GJØR: [Vurder om du skal droppe denne seksjonen og slå sammen med 4.1.3].

Chapter 5

Experiments and results

GJØR: [Vurder å dele opp som Tønnes: Først 1) Evolusjonære/Simulator[?] eksperimenter og resultater, så 2) Fysiske eksperiment og resultater].

GJØR: [Legg inn performance-plot av initielt Simulator-eksperiment for synkroniseringstider for f.eks. Mirollo-Strogatz vs. Nymoen et al.'s fase-justering].

Chapter 6

Conclusions

BESKR.: [where I shall follow-up on my *research questions* by a discussion of to which degree—and in what ways—the thesis-/project-work has answered them].

GJØR: [Se på (for kapittel-inspirasjon):

- Tønnes . MSc-thesis . Discussion-Ch.
- Jim . 'how to write a master thesis.pdf' . 'Conclusions'.

].

Bibliography

[1] @columbiasc and @_flashnick. *RoadTrip: Synchronous Fireflies at Congaree National Park*. URL: `https://avltoday.6amcity.com/synchronous-fireflies-congaree-national-park/` (visited on 01/26/2022).

[2] Renato E. Mirollo and Steven H. Strogatz. "Synchronization of pulse-coupled biological oscillators". In: *SIAM Journal on Applied Mathematics* 50.6 (1990). Publisher: SIAM, pp. 1645–1662.

[3] Kristian Nymoen et al. "Decentralized harmonic synchronization in mobile music systems". In: *2014 IEEE 6th International Conference on Awareness Science and Technology (iCAST)*. IEEE, 2014, pp. 1–6.