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

Autumn 2021

Computational Self-Awareness in Musical Robotic Systems

David Thorvaldsen

© 2021 David Thorvaldsen

Computational Self-Awareness in Musical Robotic Systems

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

Printed: Reprosentralen, University of Oslo

Abstract

GJØR: [Tenk igjennom og besvar så godt som mulig etterhvert:

- What Is *Self-Awareness*, according to some of history's most prominent thinkers, including central philosophers and psychologists? Or first things first, what Is *the self*? And what exactly is *awareness*? What implications does this have as it pertains to the domain and practice of computer science and engineering — especially when designing and describing computing systems? What lessons can designers and researchers of computing systems learn, what can they be inspired by, and which computational benefits/advancements in engineering can be made when exploring the answers to these questions?

].

INKL.: [Despite its elusive, and at times hard-to-tangibly-define, nature; the psychological concept of Self-Awareness have relatively recently served as a rich source of new inspiration, previously unknown or undiscovered for the computer engineer.]?

Acknowledgement(s$^?$)

Hei og takk til alle dere som støttet og hjalp meg igjennom denne reisen.

Contents

List of Tables

List of Figures

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

GJØR: [Tenk igjennom og besvar så godt som mulig etterhvert (kompilert fra Samuelsens MSc-thesis, og etterhvert Essay-kommentarer. Kan cross-checke med Tønnes og hun andres også):

- **Why is the thesis topic, and its outflowing proposed solutions/improvements, of relevance in the world today?**
  - History of field, how things have been done before — and why the situation/needs/requirements might have changed, or why these traditional/typical solutions may be ripe for improvements or better solutions? Why are these concerns/problems/factors of importance?
    - Demonstrate, illustrate, and explain these changes / this new situation so that the reader understands why your topic's contributions are necessary or needed.
  - What are the relevant real-world problems in need of solutions/improvement, where the thesis topic can provide such solutions/improvements?
  - Differentiate between what the "Background-/Related-works-proposed method" conributes with, and the "new proposed method" that you yourself want to try out (e.g. differentiate between ODA-loops and MAPE-K-loops, and endowing computational systems with *computational self-awareness* (and *self-expression*).
    - Explain why the "new proposed method" is needed/granted, maybe in relation to a lack or challenge with the original "Background-/Related-works-proposed method". Perhaps also mention the absence or "freshness" of this "new proposed method" in the history or field of the "Background-/Related-works-proposed method".

    "From Essay-comments": [

  - What constitutes the essential ideas behind the solutions to the problem?

– Have I written a short description of the problem / challenge for my thesis?

– Have I introduced (understandably and intuitively) Self-Awareness as an exciting and relevant field/source-of-inspiration-and-concepts?

– Have I presented motivations and arguable advantages of endowing a computational system with Self-Awareness (Se reMarkable'n)?

– Have I discussed or argued for the motivations for endowing, especially and in particular, music systems with *Computational Self-Awareness* — and then connected this with efforts like Nymoen et al.'s Firefliy-synchronization and/or Chandra et al.'s Solojam?].

].

"Fra Essay-Introduction": [

INKL.: [Designing and predicting all possible scenarios of a computing system at design-time is often hard, and sometimes impossible (in the case of unpredictable faults e.g.). If one wants to achieve coordination and continuous adaptation of a system or of system-components (for example in a collective) – some sort of intelligence might be necessary to endow it with. Endowing computing systems with Self-Awareness can be beneficial in several respects, including a greater capacity to adapt, to build potential for future adaptation in unknown environments, and to explain their behaviour to humans and other systems [SACS 17 Ch. 3]. Self-awareness concepts from psychology are inspiring new approaches for engineering computing systems which operate in complex dynamic environments [SACS 16 Ch. 2]. As we can see in various Music Technology Systems, this endowing can also give rise to interesting cooperative and coordinating behaviour.]**?**

INKL.: [The problem of this thesis will mainly consist of studying the effects differing Self-Awareness levels, varying collective-sizes, levels of task difficulty (like more complex behaviours, and limited communication) – have on usefulness, system dynamics, overall performance, and scalability (primarily within the domain of Musical Multi-Robot/-Agent collectives).]**?**].

"Fra Essay-Introduction-kommentarer": [

INKL.: [In this MSc. thesis, we will explore an exciting and relatively new translation of the concepts and notions regarding *self-awareness* – as they pertain to humans and animals especially – from the domain of Psychology, into the domain of Computation and Engineering.]**?**

INKL.: [We will in this project attempt to implement and explore whether, and indeed in what way (how), a computational system (like computational agents/agent-collectives in a computer-simulation, and/or even on physical musical robots) can exibit and display Self-Awareness (and corresponding Self-Expressive) capabilities, comparable to how they are perceived in humans and animals.]?

INKL.: [We will in this thesis project investigate methods and models for computational self-awareness in multi-robot systems, with application to the musical robotics domain.]?].

INKL.: [Engineering a computing system for a certain environment often requires some knowledge of said environement — both on the end of the creator of the computing system, as well as for the computing system in turn. This is at least the case in autonomous computing, where computing systems are supposed to be able to observe, learn, adapt, and act on their own — independently from their creator.]?

However, predicting all possible future states of complex, dynamic, and ever-changing environments is hard, and at times impossible. **INKL.:** [This calls for online and continuous learning, don't you think? How to best tackle this problem? Glad you asked. — With Self-Awareness of course. Because ...]?

GJØR: [Summer de viktigste bidragene av arbeidet her eller under 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

GJØR: [Kople tekst oppmot (de 2-3) *research-questions*'a mine her].

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

1.3 Outline

GJØR: [Skriv opp strukturen/oversikten (Eagle's-eye) av thesis-dokumentet her].

Chapter 2

Background

GJØR: [Skriv opp bulletpoints fra mulige inspirasjoner og referanser her. "Write a few lines summarising relevant articles one comes across (which one is likely to refer to in the final report)" - Jims master-skrivingsdokument)].

"Fra Essay-kommentarer": [

INKL.: [Often times, scientists have drawn inspiration from various scientific fields – particularly different fields from ones own – into their own field, for various reasons. Indeed, the translated concepts (from the one domain to the other) will most likely be accompanied with brand new ways to think about ones own domain, as well as other domains again interacting with it (hence having a real opportunity to start a "domino"-like chain-reaction of new ways to think about things emerging). These new ways to think about things (often in ones own domain) – apart from being interesting and intriguing – might be useful, both for ones own field but also for other fields again (especially if thinking long term). For example in the Multi-Agent Systems (MAS) field, it has been a common practice to study complex biological systems in nature, in order to translate these mechanisms into the technology- and engineering-domain (be it the *Ant Colony(?)*, or *Beeclust(?)*). Such bio-inspired algorithms have been – *and still are(?)* – some of the most widely used optimization algorithms throughout history.]**?**

INKL.: [(Fra Essay om 'path planning', 'EA's og 'multi-objective optimization'. Definitivt ikke kopier, men skriv om isåfall): "This essay attempts to give an overview over the fields of path-planning, evolutionary algorithms and multi-objective optimization, including pointers to recent work in these fields, especially where they intersect. In tradition with other literature relating to evolutionary algorithms, algorithms which are not population-based or otherwise based on principles similar to evolutionary algorithms are called 'classical' to separate them from evolutionary variants."]**?**].

2.1 Nymoen et al. sin Firefly-Synkronisering:

INKL.: [
Nymoen et al. [1] showed how one can, by endowing musical agents with self-awareness capabilities, achieve *harmonic synchrony* of phases and frequencies in pulse-coupled oscillators.

"Sigmund Kjøkken-Recall": [

BESKR.: [Kanskje denne seksjonen blir gjort overflødig av en god forklaring på Oscillatorer og synkronisering av Oscillatorer i 'Background'-kapittelet, og en god forklaring av akkurat hvordan dette oppnås i 'Implementation'-kapittelet?].

GJØR: [Demonstrer/Illustrer poenget bak fingrene og tidsaksen på bordet (ish det som er i figuren under for fase), og så det samme for frekvens-justering med f.eks. halve—eller noe annet—som start-frekvens; og at de da ender i *harmonisk synkroni*].

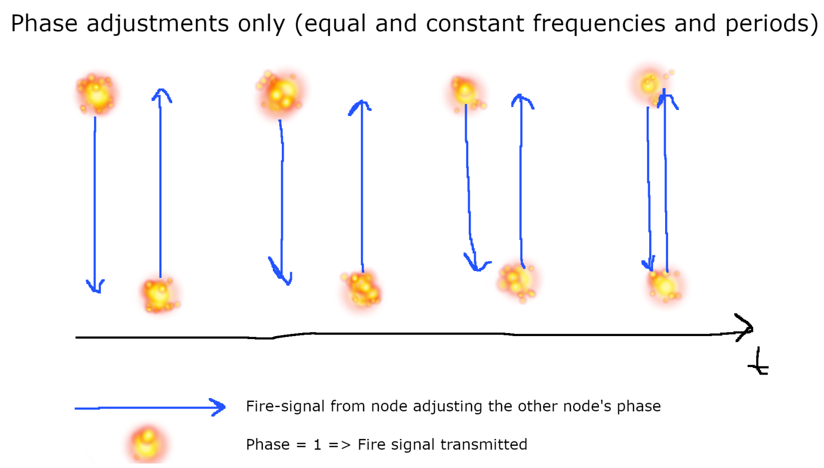


Figure 2.1: Fase-justering (som om man skulle tappet med fingrene på et bord)

].

"Tante-Kjersti-inspirert (ish Stue-Recall)": [

The diverse and complex phenomena of nature have for long served as exciting inspirations to human engineering and research (cite ant colonies, boids & swarms, beeclust e.g.). One such phenomena studied and attempted modelled is the synchronous firing of fireflies in the rainforests.

GJØR: [Insert illustration/picture of synchronizing/synchronized fireflies firing in a dark forest here].

This has inspired scientists like Mirollo & Strogatz [], and in later time Kristian Nymoen, Kyrre Glette et al. [1], to attempt to model and "etterlikne" this natural phenomenon in human-engineered systems. This work ties into the work on synchronizing oscillators []? which has been subject to study for some time now. What separates Mirollo & Strogatz and K. Nymoen's approach from these previous ones, is that here the oscillators are *pulse-coupled*, as opposed to the more normal and constraining *phase-coupled* (explain?). Each modelled "firefly", or firing node, is here implemented and considered as an oscillator, characterized by its phase and frequency. **INKL.:** [Kinda, the job is to align sinusoidal waves, either by shifting an agent's phase "up", or "down".]**?**

INKL.: [No training of any neural networks or any model-data was needed to achieve synchrony in this case — and so far no machine learning is used — but instead we see an emergent *harmonic synchrony* in a collective, by endowing fairly simple agents with not too complicated update-functions. This is well known in the Multi-Agent Systems & Swarm Robotics literature []?.]**?**].]**?**

2.2 Oscillators and Oscillator-Synchronization

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

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

Chapter 3

Tools and Software

BESKR.: [”Det man har brukt” — 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]**.**

- 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.

Chapter 4

Implementation

GJØR: [**Svar på disse spørsmålene (kompilert fra Tønnes sin masteroppgave):** What does the chapter do? What's the main goal of the design/implementation/developed system? What do these goals require? Why did you make the design-choices you made? What advantages do these choices ensure/enable? Give a short chapter-outline (perhaps first about the design, my additional Self-Awareness component, manual choice of initial parameters, and then the benchmark or målestokk or performance measure I use to evaluate with).].

BESKR.: ["Det man har utviklet" — Kyrre].

GJØR: [Skriv opp Worklog-materiale (under "muy bien so far"-bokmerket) dandert i henhold til gode master-theses].

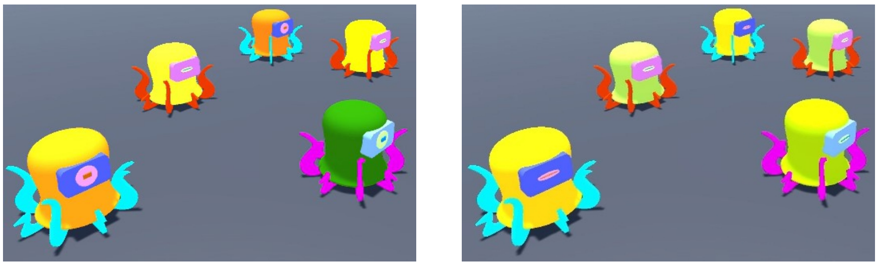
GJØR: [Se på Tønnes sin Masteroppgave og 'Archive History'-kommentarene nederst i .tex-fila for inspirasjon].

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 co-ordinating/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.

4.1 The baseline: Achieving Decentralized Harmonic Synchronization in (Oscillators $\vee$ Musical Robot Collectives)

BESKR.: [Nymoens ideer og algoritmer implementert i mitt system i Unity].

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 approach for achieving *decentralized harmonic synchronization in mobile music systems* [1]. 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. (NB! Dette skaper kanskje en misoppfatning om at alle noder må fyre samtidig for å være synkroniserte.. Husk at dette ikke er tilfellet med *harmonisk synkronisering*. **INKL.:** [denne figuren]**?** Husk også tvetydigheten med den orange Dr. Squiggle'n og den gule "fire"-color'en, og at du nå bruker .JPG (æsj))

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 in Subsection 4.1.4.

GJØR: [Insert a phase-/time-plot Figure with two Subplots. Subplot 1: phase-/time-plot when oscillators are unsynchronized. Subplot 2: phase-/time-plot when oscillators are synchronized].

GJØR: [Insert a frequency-/time-plot Figure with two Subplots. Subplot 1:

frequency-/time-plot when oscillators are unsynchronized. Subplot 2: frequency-/time-plot when oscillators are synchronized].

4.1.1 The node: the musical robot individual

BESKR.: [Om den enkelte noden/agenten med alle egenskaper den har osv. (som f.eks. en oscillator-komponent (jf. Nymoens Implementation-seksjon))].

4.1.2 Robot communication: the "fire"-signal

BESKR.: [Om kommunikasjonsmetoden til agentene: audio-/"fire"-signalet].

4.1.3 System target state: Harmonic Synchrony

BESKR.: [Om target-staten til systemet: harmonisk synkroni].
The goal and target state of the system is *harmonic synchrony*, as K. Nymoen et al. [1] coined it when ...

4.1.4 Update/Adjustment functions: Phase- & Frequency-Adjustment

4.1.4.1 Phase Adjustment

If we first assume constant and equal oscillator-frequencies in our agents, we can take a look at how the agents adjust their phase in order to synchronize to each other. This is in contrast to the case in 4.1.4.2 where heterogenous, often randomly initialized, oscillator-frequencies in the musical agents are implemented and utilized.

Two approaches were attempted in Unity, as presented in Nymoen et al.'s "*Firefly-paper*" [1]:

1) Mirollo-Strogatz's "standard" Phase Shifts

Each musical node updates its phase ϕ accoring to the **phase update function** (4.1) when hearing a "fire"-event from one of the other musical nodes:

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

where "*α is the pulse coupling constant, denoting the strength between nodes*" [1]. 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 $P(\phi) = (1 + 0.1)\phi = 1.1\phi$. 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.

GJØR: [Legg inn en figur liknende Nymoens om "standard" fase-justering].

2) Nymoen et al.'s Bi-Directional Phase Shifts

This Phase-adjustment works very similarly to the one in the ***Mirollo-Strogatz*** approach; The only difference being that now, nodes update their phase with the slightly more complex **phase update function** (4.2) when hearing a "fire"-event from one of the other musical nodes:

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

leading to the new and updated phases both being larger, but also smaller, compared to the old phases. This is 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*" [1].

The effects then of adjusting phases, upon hearing "fire"-events, according to this update-function (4.2) are that the nodes's phases now get decreased if ϕ is lower than 0.5, increased if ϕ is higher than 0.5, and neither—at least almost—if the phases are close to $0.5 = \frac{1}{2}$. This is due to the negative and positive sign of the sinewave-component in (4.2), as well as the last factor in (4.2) of $|sin(2\pi\phi)| \approx |sin(2\pi\frac{1}{2})| = |sin(\pi)| = |0| = 0$.

GJØR: [Legg inn en figur liknende den i Paragraph'en over (med Mirollo-Strogatz), bare med positive OG negative (evt. bare et negativt) fase-justeringer/-bidrag].

4.1.4.2 Frequency Adjustment

GJØR: [Presenter hva Frequency-adjustment er (hvis det ikke ble tydelig nok i slutten av Seksjon-4.1), bygg opp fremgangsmåten på hvordan man oppnår Frequency-adjustment (som i reMarkable-notatet 'Logs' → 'Simulations' → 'Frequency adjustment', og Worklog'en) — med passende figurer og illustrasjoner].

4.2 The new proposed algorithm: an additional Self-Awareness component

Chapter 5

Experiments and Results

BESKR.: [Skal følge opp *research questions*'a mine ved en diskusjon av til hvilken grad og på hvilke måter arbeidet har besvart dem].

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

Discussion/Conclusions

GJØR: [Se på Tønnes sin masteroppgave for inspirasjon].
GJØR: [(Hvis jeg heller vil bruke 'Conclusions' som tittel og dra inspirasjon fra Jim) Eller se på 'how to write a master thesis' av Jim om 'Conclusions'].

Bibliography

[1] Kristian Nymoen et al. “Decentralized Harmonic Synchronization in Mobile Music Systems”. In: *Grant agreement no. 257906 (EPiCS) from EU FP7* (2014).