

Chapter 1

Experiments and results

BESKR.: [Et kapittel der du har satt opp (med hyper-parametere og miljøvariabler f.eks. i en oversiktlig tabell) eksperimenter, kjørt simulation-runs med disse verdiene, og viser hva resultatene ble. Resultater kan være **performance plots** (ift. harmonic synch.-times for diverse hyper-parametere og miljøvariabler (samt om robotene er homogene eller heterogene og isåfall hvordan), der plottet må lages spesifikt for hvert eksperiment, og kan f.eks. være boxplot eller tabeller med performance-/hsynctime-/simulation-time(s)-verdier), **harmonic synchrony detection plots** (altså hvordan harmonic synch. ble detectet i et simulation-run), **phase-&frequency-plots** (altså hvilke fase- og frekvensverdier robotene hadde iløpet av simulation-run'et, via `plot_PhaseFrequencyPlot_for_SimRun.py`), eller **synchrony-evolution plots** (der 'towards_k_counter'en iløpet av simulation-run'et plottes, via `plot_SynchronyEvolutionPlot_for_SimRun.py`)].

BESKR.: [Helst gode eksperimenter som motiveres og forklares, settes opp, og utføres m/resultater man diskuterer og analyserer. Det er bra å evaluere fra flere synspunkt med flere research methods og hvis tid].

This chapter presents the experiments set up and performed in the novel synchronization simulator in Unity, as presented in Chapter ??, for certain configurations of musical robot collectives. Effects of the individual musical robots' hyperparameters on the collective achievement and performance of achieving harmonic synchrony are presented. Some examples are hyperparameters which determines how much each musical robot will adjust itself after hearing a transmitted fire signal from a neighbouring robot; α for phase adjustment and β for frequency adjustment.

The main performance scores presented in this chapter will consist of synchronization times given in simulation time (s) (i.e. how long it takes robot collectives to reach the state of harmonic synchrony if they ever do), accompanied by the respective and corresponding error rates during the belonging simulation runs (i.e. the percentage of robot collectives out of e.g. 30 runs failing to reach harmonic synchrony before the maximum time limit of e.g. 5 simulation minutes). If a musical robot collective then never reaches the target state of harmonic synchrony within the maximum time limit, this simulation

run will be regarded as a “synchronization fail”, and we will then not regard its termination time (in simulation time seconds) as *harmonic synchronization time*—but simply that, that simulation run’s termination time.

1.1 Solving the ϕ synchronization problem

This is the section where experiments attempting to solve the first and simpler problem, namely synchronizing only the phases ϕ_i of all agents i , are presented and analyzed results for. These are then experiments where all musical robots have an equal and fixed frequency, only adjusting phases, in order to entrain to synchronize their phases to each other until reaching the target state of harmonic synchrony.

1.1.1 Reproducing baseline phase synchronization

In order to see that our developed synchronization simulator in Unity yields more or less the same results as Nymoen’s results, similar experiments as reported in their paper are performed here. These tell us whether differences in performance, in terms of synchronization times (sim s), is simply due to implementation differences, or actually because of the utilized synchronization methods and hyperparameters in question.

First off, Mirollo & Strogatz’s phase adjustment method (as presented in ??) is experimented with for the initial phase (ϕ) synchronization problem, for varying phase coupling constants α . Results can be seen in Figure 1.1.

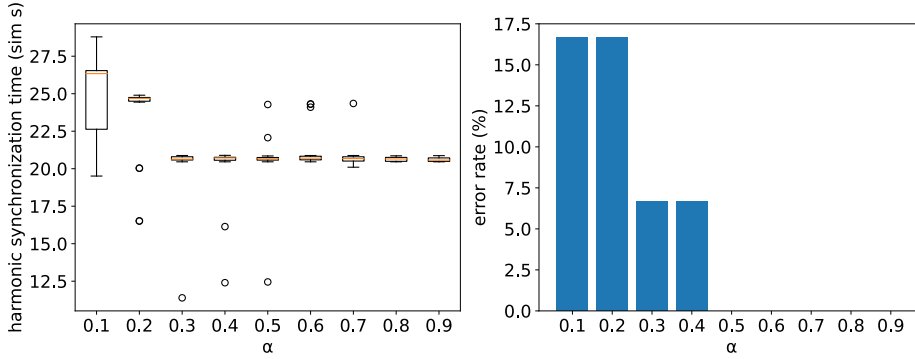


Figure 1.1: Harmonic synchronization times (s) for 6 robots with initially random and unsynchronized phases but equal and fixed frequencies (1Hz), for varying phase coupling constants α . 30 simulation runs per α . $\beta = 0$, $t_{ref} = 50ms$, $t_f = 80ms$, and $k = 8$. Maximum time limit was 5 simulation minutes.

1.1.2 Hyperparameter tuning

The specific values of hyperparameters to test synchronization times for were chosen based on an initial and identical but failed experiment (not reported) where the tested values for t_{ref}^{dyn} were showing some interesting effects for various robot collective sizes; and furthermore, after seeing, in the results found in this thesis for α already, which values of α could be interesting to investigate further.

The specific choices of hyperparameter values to test synchronization times for, when wanting to tune the phase coupling constant α and the dynamic refractory period variable $t_{ref_percentage_of_period}$, were chosen based on empirical hunches (gotten during observation of the synchronization simulator in Unity when playing with the hyperparameters) for what might be acceptable, according to experience, in order to facilitate stable and successful synchronization simulation runs—for varying robot collective sizes.

An hypothesis and aforementioned hunch is e.g. that larger robot collectives do not require as large of a phase coupling constant α in order to synchronize to each other compared to that which smaller robot collectives require; as we remember α tells each robot how much to adjust its oscillator phase when hearing a "fire" signal, and when more neighbours are present to transmit "fire" signals to a robot—a larger quantity of small adjustments can in theory be equivalent to a few large adjustments. This is something we want to investigate more quantitatively and thoroughly by the following experiment.

In the Experiment 6.1.2 results, the effects of tuning hyperparameters α and $t_{ref_percentage_of_period}$ for various musical robot collective sizes in the simpler phase (ϕ) synchronization problem are shown. Again, since we are solving the phase (ϕ) synchronization problem, now only phases are initially unsynchronized, and frequencies are fixed and constant (1Hz) throughout the simulation runs.

1.1.3 Comparing phase adjustment methods

Now, an experiment follows where we investigate the validity of the claimed [] benefits of performing bi directional phase adjustments—both inhibitory and excitatory—compared to simply adjusting phases in an excitatory way (i.e. only "pushing" other oscillators's phase further or higher when firing, never "holding" or "dragging" it back).

[FYLL INN](#)

1.2 Solving the ϕ & ω synchronization problem

This is the section for the experiments attempting to solve the second and harder problem of synchronizing both phases ϕ_i , as well as frequencies ω_i , for all agents i . These are then experiments where all musical robots originally have unequal and ever-changing phases and frequencies, adjusting both phases and frequencies in order to entrain to synchronize their phases and frequencies until reaching harmonic synchrony.

1.2.1 Reproducing baseline phase and frequency synchronization

Again, we want to here see whether we get more or less the same results in the Unity simulator as K. Nymoen et al. get in their firefly oscillator system.

Hence, we here present attempts made in the novel Unity synchronization simulator at recreating K. Nymoen et al.'s first results with their novel frequency synchronization method, which is utilizing, amongst other aspects, self awareness [2].

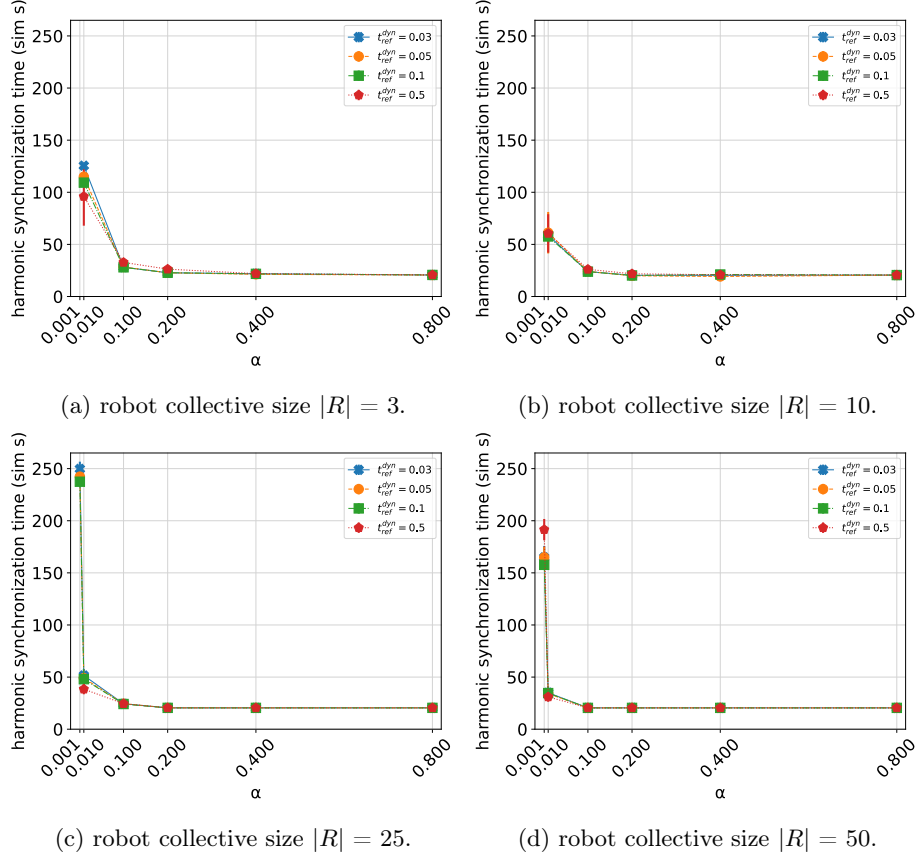


Figure 1.2: Hyperparameter tuning results: Performance scores are given in average \pm std simulation time (s), and ‘-’ entries imply that this type / configuration of musical robot collective did not once (out of 30 independent simulation runs) achieve harmonic synchrony during the maximum time limit. 30 simulation runs per α and $t_{ref_percentage_of_period}$ pair (although only simulation successes are included in the calculation of averages and standard deviations). $t_f = 80ms$, and $k = 8$. Maximum time limit was 5 simulation minutes.

Ordering by phase couplings Given that K. Nymoen et al. do not mention their β value in their frequency synchronization experiment where they order for different phase coupling values α , an *empirically decent* β value of 0.4 is chosen for this experiment. What *empirically decent* refers to in this case are K. Nymoen et al.’s findings in the results of their last experiment [2] where synchronization times for various β values were evaluated; deeming $\beta = 0.4$ to be a good value, with no further improvement in synchronization performance when β is increased further.

Here, K. Nymoen et al.’s self aware frequency adjustment method, implemented in our novel synchrony simulator in Unity, is experimented with for varying phase coupling constants α . This time not only oscillator phases are initially unsynchronized; oscillator frequencies are also unsynchronized to begin with. Hence, we here try to synchronize our musical robots in the phase

(ϕ) and frequency (ω) synchronization problem, using Nymoen’s phase adjustment method to synchronize phases, as well as Nymoen’s frequency adjustment method to synchronize frequencies. See the results in Figure 1.3.

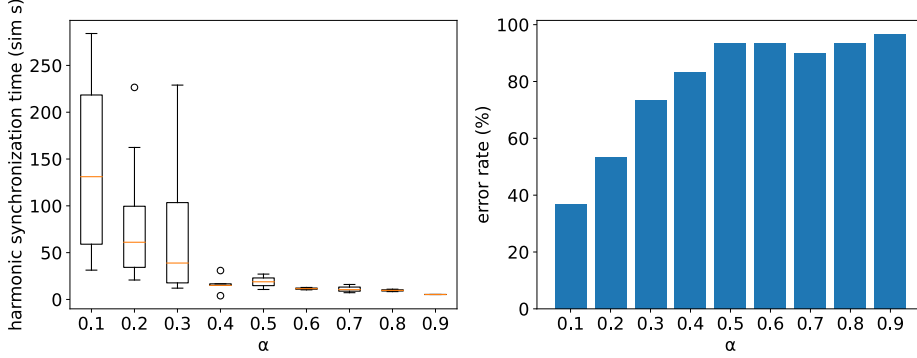


Figure 1.3: Harmonic synchronization times (sim s) for 6 robots with both initially unsynchronized phases *and* frequencies for varying phase coupling constants α . 30 simulation runs per α . $\beta = 0.4$, $t_{ref} = 50ms$, $m = 5$, $t_f = 80ms$, and $k = 8$. Maximum time limit was 5 simulation minutes.

Ordering by phase couplings but with more stable hyperparameters

When seeing how poorly the robot collectives managed to achieve harmonic synchrony in experiment [], we change the hyperparameters slightly in the hopes of achieving harmonic synchrony more frequently, or in other words more stable synchronization simulation runs. The reason is mostly that it would be beneficial to actually see whether we observe similar patterns as in Nymoen’s results—something which becomes impossible when robot collectives nearly never achieve harmonic synchrony.

The frequency coupling constant of $\beta = 0.6$ could potentially be another supposed good β value, at least solely judging by Nymoen’s results, and hence we select this value for this hopefully stable phase and frequency synchronization experiment.

By initial trial and error, it also became apparent that phase *and* frequency synchronization was not always stable—i.e. the robot collective not managing to achieve harmonic synchrony within the maximum time limit—for collective sizes of 6 or more. It did however become apparent that phase *and* frequency synchronization *was* stable for smaller musical robot collective sizes, like 2 or 3.

Hence, a similar experiment to as in [] is set up in the Unity synchrony simulator, and is run similarly as before. So still, the phase (ϕ) and frequency (ω) synchronization problem is to be experimented for, given different phase coupling constants α , with Nymoen’s both phase and frequency adjustment methods—only this time with $\beta = 0.6$ and $collsize = 3$ instead. See results in Figure 1.4.

Ordering by frequency couplings Now, we perform the same phase & frequency synchronization experiment as in Figure 1.3, except that this time we will fix the phase coupling constant α and instead test how the individual

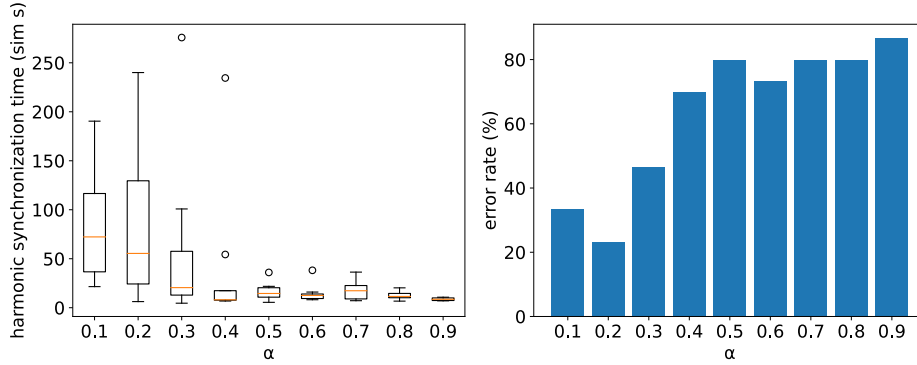


Figure 1.4: Harmonic synchronization times (sim s) for 3 robots with both initially unsynchronized phases *and* frequencies for varying phase coupling constants α . 30 simulation runs per α . $\beta = 0.6$, $t_{ref} = 50ms$, $m = 5$, $t_f = 80ms$, and $k = 8$. Maximum time limit was 5 simulation minutes.

musical robots’s various frequency coupling constants β affect the performance of the musical robot collective. These results are shown in Figure ??

Again, K. Nymoen et al. does not specify exactly the phase coupling constant they use in this latter experiment when testing their firefly-inspired synchronization system for various β values. Hence, the now fixed phase coupling constant α is here selected by reusing the α value found in the similar experiment presented in Figure ?? to yield the lowest error rate in the musical robot collective when synchronizing to each other. This then gives us a fixed $\alpha = 0.2$.

FYLL INN

Ordering by frequency couplings but with more stable hyperparameters When again seeing how poorly the robot collectives managed to achieve harmonic synchrony in experiment [], we change the hyperparameters slightly in the hopes of achieving harmonic synchrony more frequently, or in other words more stable synchronization simulation runs. The reason is the same as for the previous experiment where collective size was reduced; mostly in order to see whether we observe similar patterns as in Nymoen’s results.

As we saw in the other stabilizing experiment [], smaller musical robot collective sizes like 3 seemed to definitely make robot collectives achieve harmonic synchronization more frequently and also quicker. Hence we also will use collective size 3 here.

So still, the phase (ϕ) *and* frequency (ω) synchronization problem is to be experimented for, given different frequency coupling constants β , with Nymoen’s both phase and frequency adjustment methods—only this time with *collsize* = 3 instead. See results in Figure ??.

FYLL INN

1.2.2 Hyperparameter tuning

Here we tune the hyperparameters β and m for several robot collective sizes according to performance scores. The reason for tuning exactly these two hyperparameters, the frequency coupling constant β and the error score list length

(or memory length) m , is again since they seem the most significant for the synchronization performance.

1.2.3 Increasing degree of self awareness

It has previously, although more mathematically and not so much experimentally, been attempted to reduce the connectivity assumptions in pulse coupled oscillators [3] and to analyze the effects of it.

Here the hypothesis of whether increasing musical robots's degrees of self awareness will affect the synchronization performance or not, is investigated experimentally. Exactly what is meant by an increasing degree of self awareness specifically refers to the robots's *self awareness scope* [1]. This is tested in Unity for the more challenging $\phi & \omega$ problem of harmonically synchronizing both phases and frequencies. Perhaps a larger self awareness scope, meaning more knowledge about the social environment, will lead to the robots having a better "overview" of the environment; hence leading to shorter simulation time (s) before reaching the goal state of harmonic synchrony. Or perhaps hearing more "fire" signals on average simply will be disturbing to the robots and hence disturb and slow down their entrainment towards harmonic synchrony. This experiments attempts to answer questions like these by for the following three scenarios evaluating collective synchronization performance:

1. **Minimal self awareness scope:** Each individual robot only hears the nearest neighbouring robot's "fire" signals []. In this case, robots have limited and more local knowledge.
2. **Radial self awareness scope:** Each individual robot hears neighbouring robots's "fire" signals within a radius d around it []. In this case, robots have more knowledge but also still locally.
3. **Global self awareness scope:** Each individual robot hears all other neighbouring robots's "fire" signals []. In this case, robots have maximum and global knowledge when it comes to awareness of other neighbouring robots.

In this way, the degree to which robots are self aware of or communicating with other robots is increasing. The effects of increasing the degree of self awareness in this sense are shown in [].

Bibliography

- [1] Peter Lewis et al. “Towards a Framework for the Levels and Aspects of Self-aware Computing Systems”. In: *Self-Aware Computing Systems*. Ed. by Samuel Kounev et al. Cham: Springer International Publishing, 2017. Chap. 3, pp. 51–85. ISBN: 978-3-319-47474-8. DOI: 10.1007/978-3-319-47474-8_3.
- [2] 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.
- [3] Anton V. Proskurnikov and Ming Cao. “Synchronization of pulse-coupled oscillators and clocks under minimal connectivity assumptions”. In: *IEEE Transactions on Automatic Control* 62.11 (2016). Publisher: IEEE, pp. 5873–5879.