POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Executive Summary of the Thesis

Emergent orchestra: a framework for bio-inspired musical robot swarms

Laurea Magistrale in Music and acoustic engineering Engineering - Ingegneria Industriale e dell'Informazione

Author: Pierluigi Tartabini

Advisor: Prof. Fabio Antonacci

Co-advisor: Andreagiovanni Reina

Academic year: 2025-2026

---

1. Introduction

Can music emerge from a swarm of robots, each playing a single note while coordinating with others? This thesis presents a simulation framework in which a group of robots self-organizes as an orchestra and sustains a musical performance without a central leader. The system is inspired by principles of collective decision-making and bio-inspired collective behavior, modelling how agents interact and coordinate their choices in order to produce coherent musical output.

The framework simulates an experimental setting in which an arbitrary number of robots is placed on a virtual stage: each robot plays one note per musical measure, characterized by attributes such as *volume*, *pitch*, *onset timing*, and *instrumental timbre*. Initially, robots explore these parameters randomly, then iteratively adapt them based on what they listen from others, to progressively converge toward coordinated musical behavior without centralized control.

Music emerges across four dimensions: harmony, rhythm, beat synchronization and timbre distribution. Each robot is equipped with a dedicated module for each dimension, operating in parallel to analyze perceived musical features and update the note to be played in the next iteration.

1.1. Contribution

The original idea underlying this framework builds upon the research conducted by Lluc Bono Rosselló, Muhanad Alkilabi, Elio Tuci, Hugues Bersini, and Andreagiovanni Reina at the Université Libre de Bruxelles [4]. In that research, the experiment was carried out using Thymio robots, equipped with loudspeakers for sound production; the robots relied on collective decision-making processes and self-organized as an orchestra through the interaction of three modules.

The contribution of this thesis can be summarized as follows:

- **Development of a simulation framework** for exploring and testing the musical properties of the Emergent Orchestra. Compared to the previous ARGoS-based simulator, the proposed framework is more accessible and less dependent on robotics-specific implementation details;
- **Enhancement of the musical output** through the introduction of an additional

module within each agent, expanding the range of musical possibilities available to the swarm, in order to generate various musical outputs.
- **Integration of bio-inspired collective behavior algorithms** into the robots' control logic. Principles of *collective decision-making* observed in animal systems are translated into computational mechanisms that enable robots to coordinate musically without centralized control: this relevant aspect is underlined by the term **bio-inspired** in the thesis title.

The most common example of *collective decentralized organization* can be observed in ants which, when faced with gaps too wide to cross, connect to one another forming chains in order to create a bridge to overcome the obstacle [6]: analysis of firelflies and ant behaviour will be peculiar for two of the four modules of the framework.

These concepts constitute the central research focus of the Centre for the Advanced Study of Collective Behaviour (CASCB), where this thesis was carried out under the supervision of Andreagiovanni Reina and Lluc Bono Rossello from October 2024 to June 2025. The Centre was founded by the Max Planck Institute of Animal Behavior and the University of Konstanz in Germany; by integrating behavioral, neural, and technological perspectives, it adopts a quantitative approach aimed at advancing fundamental understanding of collective behavior.

1.2. Objectives

The objectives of this thesis arise from its inherently multidisciplinary nature, combining music, swarm robotics and biology. As a result, it pursues both scientific dissemination and artistic expression.

- **Public outreach**: encourage public engagement with scientific and artistic concepts through an accessible and immersive musical experience, that supports intuitive learning. Audience observes the progressive development of the musical performance, understanding fundamental musical concepts (such as harmony and rhythm), while simultaneously gaining insight into principles of bio-inspired collective behavior. The performance emerges from the swarm's gradual self-coordination and stabilization, allowing listeners to perceive, through sound, the dynamics of collective interaction and self-organization that sustains the system.
- **Understanding musical complexity**: unlike conventional performances, which present a refined final product, emergent music reveals the gradual unfolding of both the musical output and the underlying decision-making process thanks to a **bottom-up** development. Thanks to this approach, audience can observe how musical layers progressively emerge and stratify over time.

2. Design and Implementation

This section provides an overview of the framework's design and describes the implementation of the robots' behavioural logic.

The framework allows the user to customize simulation parameters through a configuration file, including the number of robots, their velocity, the simulation duration, the size of the stage, the BPM of the generated music, which modules are activated, and whether a final video of the simulation should be produced.

If video generation is enabled, the framework produces a final MP4 file corresponding to the performed simulation: this output is obtained by combining the sequence of rendered frames (generated at millisecond resolution) with a wav audio track containing the notes emitted by the swarm robotic. For experimental purposes, the user may also run multiple simulations in order to collect data for further analysis.

Swarm robotics are commonly described in the literature as Multi-Agent Systems (MAS). According to the definition proposed by Haopeng Guo, Tao Wu, and Xiang Xu, these systems consist of distributed intelligent agents that coordinate their capabilities through cooperative interactions, enabling the achievement of complex tasks that would be difficult or impossible for individual agents operating independently [2].

Within a MAS, communication among agents is the primary mechanism for achieving collective objectives. In the proposed framework, information exchange occurs through auditory events: whenever a robot plays a note, all other robots perceive its musical features and process them

in parallel through their four internal modules. While communication in the original experiment was *local*, occurring only within a proximity threshold via infrared signaling, the proposed framework adopts *global* communication. Each note played is perceived by all robots on the stage, resembling a jazz performance in which coordination emerges through real-time improvisation rather than a predefined score [1].
Within the software architecture, the component responsible for simulating the robots' auditory perception is the `Supervisor.py` class, which also performs core system-level operations such as initializing and positioning the virtual robots on the stage.
Once contextualized components in the framework, 4 modules implemented in the robots can be examined.

2.1. Phase Module

This module addresses the question **When should I play my note?** The Phase module is based on the internal temporal structure of each robot: every agent possesses an internal phase that functions as an intrinsic clock. From a theoretical perspective, a full phase cycle corresponds to a musical measure, which is divided into equal subdivisions referred to as movements, each associated with a sequential index known as the **beat**. Each beat is assigned an initial phase value, randomly initialized by the Supervisor at the beginning of the simulation.
The objective of the Phase module is to synchronize these phase values across robots, thereby achieving rhythmic coordination at the swarm level.

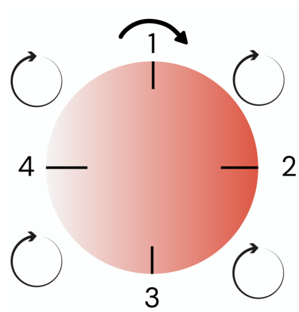

Figure 1: Example of internal phase functioning, where numbers correspond to the related beat number and the rounded arrows to the internal phases of the beat.

The image provides a visual representation of the phase mechanism operating within each agent of the swarm: in the example of a 4/4 time signature, each beat possesses its own phase and, once a full cycle is completed, the system progresses to the next beat in sequence. Once the final beat is completed, the internal clock resets and starts again from beat 1.
To coordinate phase, framework implements `Kuramoto model`, originally introduced by Yoshiki Kuramoto [3]. The algorithm provides a mathematical foundation for studying synchronization in large populations of oscillators, that update their phase *internally* each iteration (in the framework each millisecond of the simulation) with the equation:

$$\theta_i(t+1) = \theta_i(t) + \frac{2\pi}{T} \bmod 2\pi \tag{1}$$

and *externally* broadcasting their phase value. The algorithm can be summarized with the formula.

$$\theta_i(t) = (\theta_i(t) + K \sin[\theta_j(t) - \theta_i(t)]) \bmod 2\pi \tag{2}$$

where K = 1 is the coupling strength, $\theta_i(t)$ and $\theta_j(t)$ are the phase counters of robots i (receiver) and j (sender), respectively.

2.2. Beat Module

This module answers to the doubt **"At what point of the measure are we?** Each robot is associated with an internal beat counter linked to its beat phase, which either increments or resets to one upon the completion of each beat or measure cycle. A **delay** parameter is assigned to each robot, corresponding to an integer value ranging from 1 to N (the total number of beats), which determines the specific beat on which the robot should perform its note. This value is randomly initialized for each agent to ensure an even distribution of beats across the swarm.
When an agent performs, the exploited characteristic to permit beat synchronization is the **dynamic** with which the note has been played, that corresponds on how much loudness the musician applies to the note.
In the framework, robots can play note with pianissimo (*pp*), mezzoforte (*mf*) and fortissimo (*ff*) dynamics; *ff* dynamic is associated with the first beat, as to simulate the accent of the

first movement in western music. To synchronize their beat counter robots apply `Firefly synchronization`, a method inspired by firefly behavior to coordinate light that they emit their abdomen [5]. Listening on where is the first beat, robots move by 1 their counter depending the distance with their actual position. The beat synchronization enforce rhythmic swarm coordination.

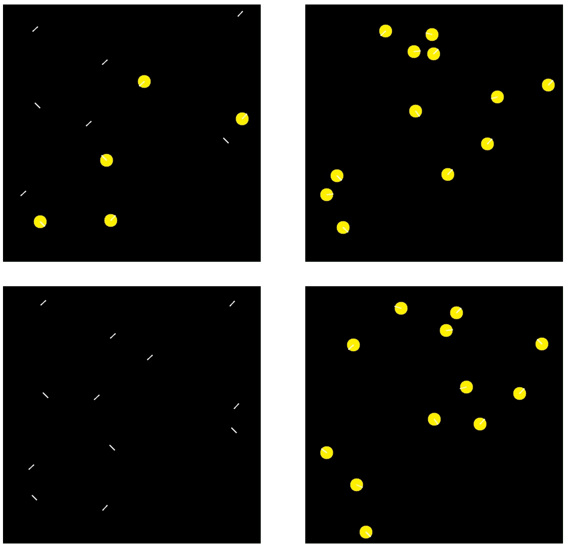

Figure 2: Visual effect of firefly module: yellow corresponds to beat counter 1 of the robot. After some iterations, they coordinate their beat counter.

2.3. Harmony module

The module responsible for establishing harmonic coherence within the swarm is the harmonic module, whose purpose is to answer the question **"Which note should I play?"**. Each robot aims to select and perform pitches that belong to at least a common musical scale, thereby fostering a form of harmonic consensus that emerges through interaction.
From a broader perspective, the collective selection of harmonically compatible notes can be compared to social conventions that emerge without formal rules or explicit enforcement. These phenomena rely on *social consensus*: they are not dictated by specific commands, but arise from shared patterns of behavior. To model this type of consensus formation, Vito Trianni, Daniele De Simone, Andreagiovanni Reina and Andrea Baronchelli developed the `naming game` for multi-robot networks, which provides conceptual inspiration for the harmony module [9].

In the proposed framework, each robot broadcasts a note whenever it plays and stores the received notes in a fixed-length buffer of size $L = 4$. Agents record the *pitch* of each received note and, once the buffer is full, the oldest entry is removed according to a FIFO (first-in, first-out) policy. Each robot continuously compares its currently played pitch with those stored in the buffer. If all buffered notes are compatible with at least one common musical scale, the swarm is considered to have reached harmonic convergence; otherwise, the agent adjusts its pitch moving from a note near to the previous one.

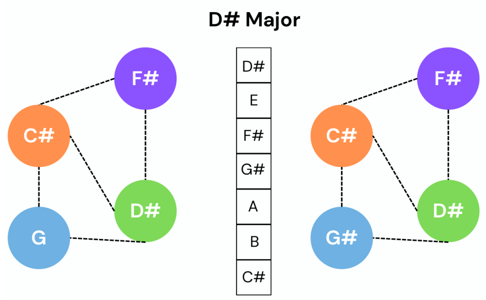

Figure 3: harmony example on a common scale

The change of notes occurs with a probability $r_c < 1$, introduced to avoid continuous note replacement that would hinder the swarm's ability to reach harmonicity (in the present framework, $r_c = 0.7$). Although agents have knowledge of multiple musical scales, only one scale pattern is instantiated per simulation.

2.4. Timbre Module

The module discusses how each agent selects its instrument, in order to contribute the *timbre synchronization* and to answer the question **Which instrument do I have to choose?** Robots have to reach a certain instrument distribution based on the relation between how many robots are in the swarm and the number of the beats implemented: the result of the division will give to robots the ensemble configuration to reach (f.e. 4 stays for quartet made by trumpet, horn, trombone and bassoon). The module takes inspiration from the `response threshold reinforcement and division of labor in insect societies`, developed by Guy Theraulaz, Eric Bonabeau, and Jean-Louis Deneubourg [7]. The model is based on 3 key concepts:

- **task**: an action that can be performed among range of possibilities. In the framework, a task corresponds to an instrument, and the robot can choose among 13 different ones.
- **response threshold**: an internal value linked with a specific task that indicates how much the individual is familiar with that specific labor. From a musical perspective, threshold related to an instrument corresponds to the question *"How good am I in playing that instrument?"* Long-term experience with an instrument lowers the threshold for its use, whereas lack of familiarity increases the difficulty of performing with it.
- **stimuli**: an internal value of the agent that stays for the perception of what the population is needing in that moment. Stimulus can be traduced as *What the swarm needs that I play ?*

Assuming that m tasks must be performed, each of them is associated with a specific stimulus. Let there be N workers, indexed by $i = 1, \dots, N$, each of them has a proper a set of response thresholds θ_{ij} $(j = 1, \dots, m)$ corresponding to the stimuli associated with task j. Let s_j denote the intensity of the stimulus related to task j; in the fixed-threshold model, individual i chooses task j with probability

$$T_{\theta_{ij}}(s_j) = \frac{s_j^2}{s_j^2 + \theta_{ij}^2} \tag{3}$$

Threshold updates are based on **learning** coefficient α and **forgetting** coefficient ϕ,while x_{ij} denote the fraction of time that individual i spends performing task j by an amount of time $\alpha\,\Delta t$. The threshold update equation is

$$\theta_{ij} \to \theta_{ij} - x_{ij}\,\alpha\,\Delta t + (1 - x_{ij})\,\phi\,\Delta t. \tag{4}$$

The relation that governs stimulus update per unit time is:

$$\partial_t s_j \; = \; \delta \; - \; \frac{\beta}{N}\left(\sum_{i=1}^{N} x_{ij}\right), \tag{5}$$

where $\partial_t s_j$ is the derivative of the function over time, δ is the rate at which stimulus intensity increases, and β is a scaling factor measuring the efficiency of task performance. To reach the target distribution in the framework, δ is updated to compute the range between current instruments distribution and target one with the formula:

$$\delta_j = T_j - C_j \tag{6}$$

where T_j denotes the target distribution and C_j denotes the current distribution of the instruments.

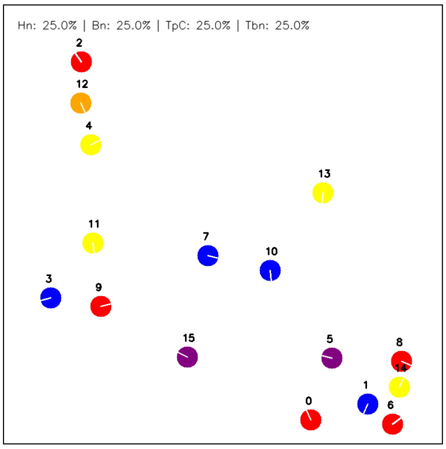

Figure 4: example of frame indicating timbre distribution in a simulation.

2.5. Database

In the proposed project, each robot is able to switch between 13 musical instruments. This capability is enabled by integrating a dedicated database taken from the TinySOL database from IRCAM and modified to integrate into this framework [8]. Used samples instruments are reported in the table below.

Category	**Instruments**
Strings	Violin, Viola, Violoncello, Double Bass
Woodwinds	Flute, Oboe, Clarinet in B^b, Bassoon
Brass	Trumpet in C, Trombone, Horn, Bass Tuba
Other Instruments	Alto Saxophone, Accordion

Table 1: Instrument categories used in the framework.

3. Results

This section presents graphical analyses of the results obtained from the robotic swarm for the four modules mentioned before. For *phase synchronization* has been used the equation

$$\Delta\Theta(t) = \frac{2}{M(M-1)} \sum_{i=1}^{M} \sum_{j=i+1}^{M} \left(|\theta_i(t) - \theta_j(t)| \bmod \pi \right), \tag{7}$$

where M denotes the total number of robots in the system. The normalization factor ensures that $\Delta\Theta(t)$ is bounded in the interval $[0,1]$, where $\Delta\Theta(t) = 0$ indicates perfect phase synchronization among all robots, while $\Delta\Theta(t) = 1$ corresponds to maximum phase asynchronous. The swarm coordinate in the first seconds of the simulation; smaller is the group, faster is the perfect synchronism.

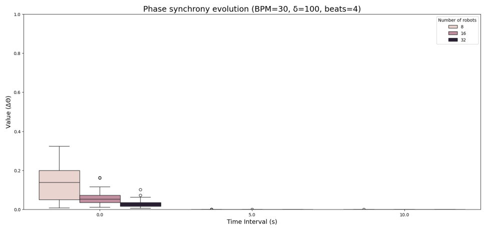

Figure 5: boxplot for simulation with different size of swarm.

For *beat synchronization*, a similar logical formulation was used, yielding results consistent with those obtained for phase coordination. The time required to reach synchronization depends on the BPM, due to the periodicity of the perceived notes.

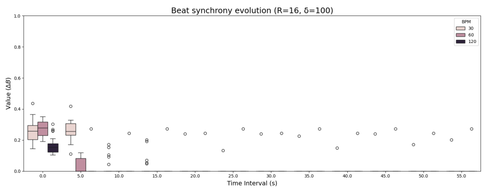

Figure 6: Beat synchronization for different BPM

For *harmonic agreement* has been used

$$H(t) = \frac{1}{M} \max_{r \in \{0,\dots,11\}} N_r(t) \tag{8}$$

where $N_r(t)$ defines the number of robots whose pitches belong to that scale at time t. Harmony synchronization is reached in the first seconds of the simulation.

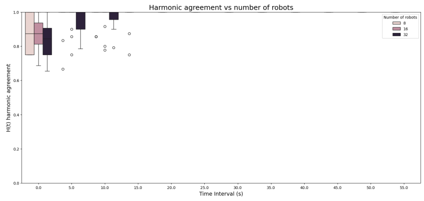

Figure 7: harmony module for different swarm sizes.

For what concerns *timbre distribution*, results are strictly related to *delta* value implemented in the framework, that restricts or not the difference between distributions on each iteration. To a larger *delta* value corresponds a faster correct distribution.

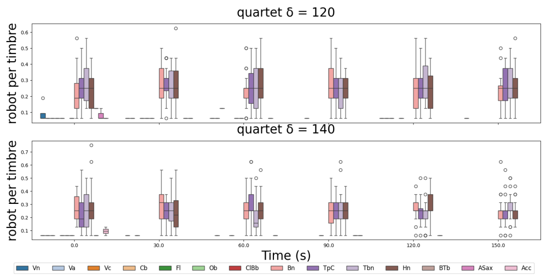

Figure 8: timbre distribution for a quartet comparing *delta* 120 and 140.

4. Conclusions

The framework is designed to be modular and scalable, allowing users to configure parameters and activate specific robot modules via a configuration file, thereby enabling flexible initialization across different experimental setups. Furthermore, project demonstrates high efficiency in achieving rhythmic, beat, and harmonic coordination, which typically emerges within the first seconds of the simulation. Timbre distribution, by contrast, depends more strongly on user-defined parameters and on the chosen delta between the current and target instrumental distributions.

Overall, the framework provides a solid foundation for future developments, including the implementation of a melody module and the introduction of musical rests.

Repository folder is available at the following link: https://github.com/PierluT/Swarm-Robotic-Orchestra. A video of the simulation is available at the following link: https://www.youtube.com/watch?si=JOuei3PDI-awdJ02v=DXyIC0Q369gfeature=youtu.be

References

[1] Margaret S Barrett, Andrea Creech, and Katie Zhukov. Creative collaboration and collaborative creativity: A systematic literature review. *Frontiers in Psychology*, 12:713445, 2021.

[2] Haopeng Guo, Tao Wu, and Xiang Xu. Sample-data output consensus for heterogeneous linear multi-agent systems with time-varying communication delays. In *2025 4th Conference on Fully Actuated System Theory and Applications (FASTA)*, pages 769–774. IEEE, 2025.

[3] Yoshiki Kuramoto. Self-entrainment of a population of coupled non-linear oscillators. In *International symposium on mathematical problems in theoretical physics: January 23–29, 1975, kyoto university, kyoto/Japan*, pages 420–422. Springer, 2005.

[4] Lluc Bono Rosselló, Muhanad Alkilabi, Elio Tuci, Hugues Bersini, and Andreagiovanni Reina. Emergent orchestras: A modular framework for musical robot swarms. In *Artificial Life Conference Proceedings 36*, volume 2024, page 29. MIT Press One Rogers Street, Cambridge, MA 02142-1209, USA journals-info . . . , 2024.

[5] Raphael Sarfati, Kunaal Joshi, Owen Martin, Julie C Hayes, Srividya Iyer-Biswas, and Orit Peleg. Emergent periodicity in the collective synchronous flashing of fireflies. *Elife*, 12:e78908, 2023.

[6] Rodney Taw. Ants' collective intelligence: What could we learn? *Science Insights*, 46(1):1711–1722, 2025.

[7] G Theraulaz, E Bonabeau, and J-N Deneubourg. Response threshold reinforcements and division of labour in insect societies. *Proc. Biol. Sci.*, 265(1393):327–332, 1998.

[8] TinySOL. Instrumental sounds dataset, 2025. Accessed: 2025-04-09.

[9] Vito Trianni, Daniele De Simone, Andreagiovanni Reina, and Andrea Baronchelli. Emergence of consensus in a multi-robot network: from abstract models to empirical validation. *IEEE Robotics and Automation Letters*, 1(1):348–353, 2016.