

Article

Categories, Quantum Computing, and Swarm Robotics: A Case Study

Maria Mannone ^{1,2,3,*} , Valeria Seidita ¹  and Antonio Chella ^{1,4} 

- ¹ Department of Engineering, University of Palermo, 90128 Palermo, Italy; valeria.seidita@unipa.it (V.S.); antonio.chella@unipa.it (A.C.)
- ² Department of Mathematics and Computer Sciences, University of Palermo, 90123 Palermo, Italy
- ³ European Centre for Living Technology (ECLT), Dipartimento di Scienze Ambientali, Informatica e Statistica (DAIS), Ca' Foscari University of Venice, 30172 Venice, Italy
- ⁴ ICAR-CNR National Research Council, 90146 Palermo, Italy
- * Correspondence: mariacaterina.mannone@unipa.it or maria.mannone@unive.it

Abstract: The swarms of robots are examples of artificial collective intelligence, with simple individual autonomous behavior and emerging swarm effect to accomplish even complex tasks. Modeling approaches for robotic swarm development is one of the main challenges in this field of research. Here, we present a robot-instantiated theoretical framework and a quantitative worked-out example. Aiming to build up a general model, we first sketch a diagrammatic classification of swarms relating ideal swarms to existing implementations, inspired by category theory. Then, we propose a matrix representation to relate local and global behaviors in a swarm, with diagonal sub-matrices describing individual features and off-diagonal sub-matrices as pairwise interaction terms. Thus, we attempt to shape the structure of such an interaction term, using language and tools of quantum computing for a quantitative simulation of a toy model. We choose quantum computing because of its computational efficiency. This case study can shed light on potentialities of quantum computing in the realm of swarm robotics, leaving room for progressive enrichment and refinement.



Citation: Mannone, M.; Valeria S.; Antonio, C. Categories, Quantum Computing, and Swarm Robotics: A Case Study. *Mathematics* **2022**, *10*, 372. <https://doi.org/10.3390/math10030372>

Academic Editor: Daniel-Ioan Curiaic

Received: 10 December 2021

Accepted: 24 January 2022

Published: 25 January 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: swarm robotics; quantum computing; 4-qubit system; matrix representation; colimit

1. Introduction

Acrobatics evolution of flocks of birds in our skies and colorful movements of schools of fish in our oceans inspire poets and artists, as well as computer scientists and engineers. Complex behaviors exhibited by swarms of animals [1,2], in fact, often inspire the development of swarms of robots, passing through toy models which are progressively enriched and refined [3].

In a swarm of robots [3–5], every single robot is performing simple tasks and showing a simple behavior, but the interaction and information exchange amongst robots allows the accomplishment of more complex tasks, impossible to be achieved by a single unit. Typically, the collaboration in a swarm is decentralized: there is no such a thing as a “robot leader;” instead, every single robot is acting autonomously, reacting to the information received by its neighbors, and transmitting information about its own activity. Swarms of robots show the emergence of global behavior and a form of collective intelligence—a classic example is given by ants’ behavior. Thus, we can talk of *swarm intelligence* [6]. Examples of swarm intelligence are stochastic diffusion search [7], ant colony optimization [8], and artificial swarm intelligence [9].

A swarm is an example of a complex system, which can be instantiated as a robot swarm. Applications of swarm robotics include field tasks, underwater tasks, and air tasks. Terrestrial, underwater, and flying autonomous simple vehicles have thus to be developed and made communicating between them. There are also miniature robots going inside the human body for healthcare purposes [10].

From an engineering point of view, the connection between local behavior (single or subgroups of robots in a swarm) and the global behavior of a swarm is not trivial. Some existing research exploits particle analogies from physics, with a microscopic behavior described by Langevin equation, and a macroscopic behavior formalized through the Fokker–Planck Equation [3]. A great inspiration source for mathematical modeling is also given by quantum mechanics and related quantum computing.

Recently, the widespread interest in quantum computing [11], motivated by its computational power [12] and formal elegance is influencing several fields of research, e.g., quantum computing has recently been applied to machine learning for artificial intelligence [13]. In a recent study [14], fuzzy logic is applied to a swarm of robots. Quantum logic can be considered as a particular case of fuzzy logic [15].

Some first applications of quantum computing to swarm robotics exploit evolutionary methods inspired by particle modeling dynamics and involving Monte-Carlo simulations [16], and the strong hypothesis of entanglement between swarm robots [17].

Quantum applications to robots are a flourishing research field. With the exceptions of some pioneering works, quantum applications to swarm robotics appear instead as a largely unexplored domain. Another gap is the lack of simple connections of robotics, and in particular of swarm robotics, with quantum computing at a fundamental level. A closely-related research [18], in fact, focuses on a single agent rather than on a swarm of robots.

The use of quantum computing is inspired by core ideas and discoveries in basic quantum mechanics, with energy quantization, state superposition, destructive measurement. Physics Nobel prize Richard Feynman had already hypothesized that the application of quantum principles to computer science could have enabled the construction of more powerful computers [11]. In quantum computing, the information units 0, 1 are identified with the ground and excited states of a (hydrogen) atom, respectively. Quantum computing makes use of reversible logic gates, in analogy with invertible quantum operators. The use of quantum computing is mainly motivated by computational increased efficiency. This is a flourishing research area, and thus, a connection with quantum computing can strengthen other fields of research.

In our contribution, we aim to create a general theoretical approach for interacting multi-entities with an emerging swarm behavior. As an instance of complex models with swarm behavior, we focus on robots, defining interaction terms as submatrices of the swarm matrix. As an example of an interaction term, we create a toy model with two robots, modeling robots' behavior through a quantum circuit.

In fact, in this article we aim to address the issue of the connection between macro- and micro-behaviors in a swarm of robots, crucial for swarm modeling, by using nested matrices and the basics of quantum computing. While we choose to focus on robots, our methodology is meant to be more general: in fact, we model relationships between information exchange and behavior of interacting simple entities, showing an emergent behavior.

First, we discuss a framework to describe the progressive “embodiment” of the swarm idea, from an ideal swarm to a particular system.

Second, we propose a swarm description in terms of nested matrices describing single robots' behaviors and their interaction. In fact, we can translate the swarm condition into a matrix structure. While in [17] there is a matrix where all elements (robots) are entangled, here we relax this hypothesis, and we include terms describing autonomous behavior of robots, distinguishing them from the off-diagonal terms for robots' interactions. A swarm can be described by a block matrix, where the sub-matrices along the main diagonal describe the behavior of isolated robots, and the other sub-matrices indicate the interactions between them.

Third, we work out a simple example with two robots, formalized through four quantum qubits (where the qubit is the quantum information unit), discussing the obtained results.

The experiments with the quantum circuit (corresponding to the truth table of Table 1) are set up by typing the provided code (Section 2) on IBM Quantum Composer and running it through a simulator (e.g., QASM simulator) or an IBM-owned quantum computer (e.g., the computer in Bogotá), to which the circuit has been remotely sent. We initialize input qubits through suitable quantum gates (Hadamard, Not, and Ry in our study). Each experiment consists of 1024 shots of the code. The population for each output state is indicated in Table 2.

The article is organized as follows. In Section 1.1, we summarize the key concepts of swarm robotics. In Section 2, we provide details on the mathematical formalization of the core idea, introducing the 4-qubit case study. In Section 3, we present our results. In Section 4, we summarize our study and envisage possible research developments.

1.1. Swarms of Robots

Swarm robotics is an application of swarm intelligence [6,19] to robotics. The core idea of this discipline is the reference to social insects behavior. In fact, robots in a swarm are meant to autonomously coordinate to achieve complex goals. Organisms in a natural swarm use simple rules to govern their behavior. Every single organism in a swarm can perform a few simple actions. Communication and interaction with other organisms in the swarm enable the achievement of complex goals, e.g., heavy prey transporting, foraging, or massive and complex structures building. An intuitive definition of swarm robotics is the following: *swarm robotics is the study of how a large number of relatively simple physically embodied agents can be designed such that a desired collective behavior emerges from the local interactions among agents and between the agents and the environment* [20].

A swarm of robots is constituted by a set of interacting robots, individually performing a simple job, and collectively achieving a complex task [3,4]. A complex task is achieved through task partitioning. Swarms of robots show self-organizational properties. The *swarm behavior* is considered as an emerging property. A swarm of robots is characterized by redundancy (the lack of a unity connection does not affect the whole behavior), flexibility (no specialization is required), and scalability (the control algorithms do not depend upon the size of the swarm). In swarm robotics, we can distinguish between a *micro* level and a *macro* level. The micro-level concerns the local behavior, describing single unities and pairwise interactions. The macro-level concerns global behavior, that is, the behavior of the whole swarm [3].

1.2. Ideal Swarms

In Section 2, we introduce the concept of an *ideal swarm* of robots. The ideal swarm is ontologically different from any feasible swarm: it is an abstraction. It can be visualized as a swarm of robots that:

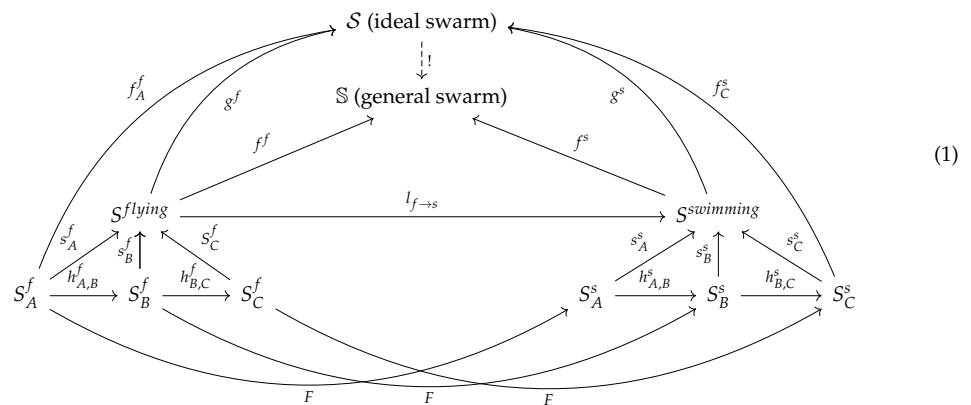
- Can work in whatever scenario (e.g., on the ground, underwater, on-air);
- Can adapt to any possible adverse condition (during a storm, after an earthquake, a tsunami, an avalanche);
- Is able to communicate in any imaginable way (though laser, infrared light, visible light, radar, sonar);
- Is ready for whichever task (e.g., search and rescue, object retrieval, stacking together to perform a more complex task).

It is impossible to envisage such a swarm or conceive any similar implementation.

In a nutshell, swarm conditions are relationships between individual behaviors of single robots and the overall actions of the swarm. Complex swarm behavior is supposed to emerge from simple individual actions and, as in our model, simple pairwise interactions. Differences in modeling between swimming [21], flying [22], and walking swarms [23] intuitively reside on locomotion features and target scenarios to which robots should be adapted.

2. Theoretical Framework and Methods

In this article, swarm conditions become conditions on matrices. Let us consider an ideal swarm (Section 1.2); we can call it \mathcal{S} . It can be obtained as an idealization from a general yet feasible swarm structure; let us call it \mathbb{S} . Such a general swarm can be specialized into main typologies of swarms, such as *swimming swarms*, *flying swarms*, and *walking swarms*. Each one of these three classes can be distinguished into more and more specific swarms, toward existing swarms, made of simple and cheap robots, which communicate between them and collaborate to perform tasks.



Moving downward in the diagram of Equation (1) we get a progressive embodiment of the general idea, toward specific realizations of swarms of robots. Vice versa, the upward paths leads to a progressive abstraction. This diagram can be compared with the species classification in biology, which has been formalized through the concept of *colimit* in Category Theory [24]. In [25], emergence of species similarities are shown through convergence of arrows. Here, emergence of similarities in swarm behaviors can also be thought of as arrows convergence. In fact, diagram (1) is shaped as a colimit construction. We can have emergent behavior similarities while comparing swarms between them, but also regarding the very definition of a swarm itself: the swarm behavior is more than the sum of single robots' movements.

In diagram (1), horizontal comparisons are comparisons across swarms with the same degree of abstraction. The comparison between two swarms S_A^f, S_B^f belonging to the category *flying swarms* is performed through suitable morphisms $h_{A,B}^f : S_A^f \rightarrow S_B^f$. The comparison between characteristics of the category of *flying robots*, S^{flying} , with the characteristics of the category of *swimming robots*, $S^{swimming}$, is represented by the arrow $l_{f \rightarrow s} : S^{flying} \rightarrow S^{swimming}$. Letter F between specific swarms of a category *flying* and swarms of a category *swimming* indicates a functor (a generalization of functions, mapping points of a category into points another category, and morphisms of a category into morphisms of another category). While it is not indicated here for reasons of graphical clarity, F also maps $h_{A,B}^f \rightarrow h_{A,B}^s, h_{B,C}^f \rightarrow h_{B,C}^s$, and $h_{A,C}^f \rightarrow h_{A,C}^s$.

Given an ideal swarm \mathcal{S} , here treated as a categorical colimit, there must be some characteristics to be taken into account to create a general yet feasible swarm \mathbb{S} . This is the meaning of the dotted arrow. According to the definition of a colimit in Category Theory, there is a unique morphism $! : \mathcal{S} \rightarrow \mathbb{S}$ such that $f^x = ! \cdot g^x$, where x is the label of a swarm category and \cdot indicates the composition; e.g., $f^f = ! \cdot g^f$. Specific swarms can be related to the abstract swarms through arrow compositions; e.g., $g^f \cdot s_A^f : S_A^f \rightarrow \mathcal{S}$ and $f^f \cdot s_A^f : S_A^f \rightarrow \mathbb{S}$.

Each swarm in diagram (1) can in principle be described by a matrix. In the case of existing physical swarms (S_A^f, \dots, S_C^f), all elements of matrices can be computed. Let us now focus on a generic swarm constituted by n robots. The block matrix describing it is sketched in Equation (2), where the sub-matrices along the principal diagonal describe the behavior of each single robot R_1, \dots, R_n , and the other sub-matrices are the pairwise

interaction terms. If these interaction terms are null matrices, there are no swarm effects. Thus, off-diagonal matrices can remind us of the *coherences* in quantum mechanics, i.e., the off-diagonal terms representing entanglement.

$$S_n = \begin{pmatrix} R_1 & R_1 * R_2 & \dots & R_1 * R_{n-1} & R_1 * R_n \\ R_2 * R_1 & R_2 & \dots & R_2 * R_{n-1} & R_2 * R_n \\ & & \vdots & & \\ & & & \vdots & \\ R_{n-1} * R_1 & R_{n-1} * R_2 & \dots & R_{n-1} & R_{n-1} \\ R_n * R_1 & R_n * R_2 & \dots & R_n * R_{n-1} & R_n \end{pmatrix} \tag{2}$$

If there are only $n = 2$ robots, the matrix is simplified as shown in Equation (3):

$$S_2 = \begin{pmatrix} R_1 & R_1 * R_2 \\ R_2 * R_1 & R_2 \end{pmatrix} \tag{3}$$

The matrix S_2 is a block matrix which can be obtained as in Equation (4):

$$S_2 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \otimes R_1 + \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \otimes (R_1 * R_2) + \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} \otimes (R_2 * R_1) + \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \otimes R_2, \tag{4}$$

where \otimes is the Kronecker product (tensor product) between matrices. The matrices R_1, R_2 describe the behavior of the two isolated robots. The interaction terms $R_1 * R_2$ and $R_2 * R_1$ are not equal, because they depend on:

- Signals transmitted from R_1 to R_2 and reactions of R_2 (term $R_1 * R_2$),
- Signals transmitted from R_2 to R_1 and reactions of R_1 (term $R_2 * R_1$), respectively.

How are the interaction terms built? For $R_1 * R_2$, we may consider the signal robot 1 is transmitting to robot 2, and the reaction of the last one. A possible communication is “where I am, where I go.” However, even in a toy model for a swarm it is important to define a task. A task could be, for example, the achievement of a goal, as a target to reach, or as a reward to obtain. (In fact, target search is a classic task for a swarm of robots [26]). Thus, in our simple description the signal from a robot provides information on its *position* as a portion of a linear space and its *target* achievement—and thus, the content of the message would be “where I am, what I found.” Hereinafter, to take into account an inner uncertainty related to the decision-making autonomy of robots, we formulate a probabilistic approach based on quantum state superposition. Thus, the possible communication scheme from R_1 to R_2 could be a matrix as:

$$\begin{pmatrix} \text{place down} & \text{reward no} \\ \text{place up} & \text{reward yes} \end{pmatrix},$$

where each entry is the probability amplitude to have that signal. To enrich this idea, we can add the reaction by R_2 :

$$(R_1 * R_2)(t) = \begin{pmatrix} R_1 \text{ place down}(t_0) & R_1 \text{ reward no}(t_0) \\ R_1 \text{ place up}(t_0) & R_1 \text{ reward yes}(t_0) \\ - & R_2 \text{ place down}(t_1) \\ - & R_2 \text{ place up}(t_1) \end{pmatrix}, \tag{5}$$

where, to a signal from R_1 at time t_0 , corresponds a position displacement by R_2 at time t_1 . The empty entries correspond to the reward from R_2 , that will be made known by the robot only at t_2 . The term $R_2 * R_1$ will thus have the same structure. To date, we did not choose a specific framework, because we are interested in creating a general approach, which can be instantiated with simple, interacting robots. In the proposed toy model, we assume to have two simple robots moving back and forth along a line.

Let us now work out a quantitative example. To this aim, we can treat the behavior of each robot as a 2-qubit quantum state, where, for the sake of simplicity, each qubit indicates the superposition of possible states of one-dimensional position (up = 1, down = 0) and of “rewards” as a reached target (yes = 1, no = 0). In total, we have a system with four qubits, that we used for a simulation with IBM simulators and real quantum computers. (<https://www.ibm.com/quantum-computing/> accessed on 10 December 2021). In Equation (6), we indicate probability amplitudes to obtain a specific measure outcome as $\alpha_i, \beta_i, \gamma_i, \delta_i, i = 1, 2$.

$$\begin{aligned} R_1 &\rightarrow |q_0(t)\rangle = \alpha_1(t)|0\rangle + \beta_1(t)|1\rangle, |q_1(t)\rangle = \gamma_1(t)|0\rangle + \delta_1(t)|1\rangle \\ R_2 &\rightarrow |q_2(t)\rangle = \alpha_2(t)|0\rangle + \beta_2(t)|1\rangle, |q_3(t)\rangle = \gamma_2(t)|0\rangle + \delta_2(t)|1\rangle \end{aligned} \tag{6}$$

The matrix of Equation (5) will then be:

$$(R_1 * R_2)(t) = \begin{pmatrix} \alpha_1(t_0) & \gamma_1(t_0) \\ \beta_1(t_0) & \delta_1(t_0) \\ - & \alpha_2(t_1) \\ - & \beta_2(t_1) \end{pmatrix}. \tag{7}$$

Let us analyze a possible realization of the scheme of communication/reaction building a reversible logic gate. The logic behind the truth tables in Table 1 is the following. At time t_0 , if R_1 is in position up ($q_0 = 1$) and it gets the reward ($q_1 = 1$), then R_2 reaches it at t_1 , independently from its former position. However, if R_1 does not get the reward ($q_1 = 0$), then R_2 explores the space down ($q_2 = 0$). Same reasoning if the starting position is the inverse. Then, the roles of R_1 and R_2 are exchanged at t_2 and t_3 . (Time in quantum gates is simultaneous, while we know that a robot, received the communication from the other one, implements the decision to change the position or keep it in the following time instant). The permutation matrix associated with Table 1 is presented in Equation (8), with quantum states ordered as 00, 01, 10, 11.

Table 1. Truth tables (reversible equivalents of XNOR gates), representing the interaction between robot 1, R_1 (q_0 : position, q_1 : reward) and robot 2, R_2 (q_2 : position, q_3 : reward). At the beginning (left table), R_1 communicates its position (down/up) and reward (no/yes); then it waits, and, according to its information, R_2 reaches R_1 or not (when R_2 reaches R_1 , q_2 becomes equal to q_0). Then (right table), we have the symmetric situation, where R_2 communicates position and reward, and R_1 decides to reach it nor not (if R_1 reaches R_2 , q_0 becomes equal to q_2). Equation (8) shows the corresponding permutation matrix.

Gate $t_0 \rightarrow t_1$				Gate $t_1 \rightarrow t_2$			
input		output		input		output	
q_0	q_1	q_0	q_2	q_2	q_3	q_2	q_0
0	0	0	1	0	0	0	1
0	1	0	0	0	1	0	0
1	0	1	0	1	0	1	0
1	1	1	1	1	1	1	1

$$\begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{8}$$

The matrix in Equation (9) shows the Toffoli gate, exploited to implement our proposed gate. The Toffoli gate is a reversible quantum gate, which takes two inputs and gives one output. It is often indicated as ccx, that is, a not (x) with two conditions. Figure 1 shows the considered quantum circuit.

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{9}$$

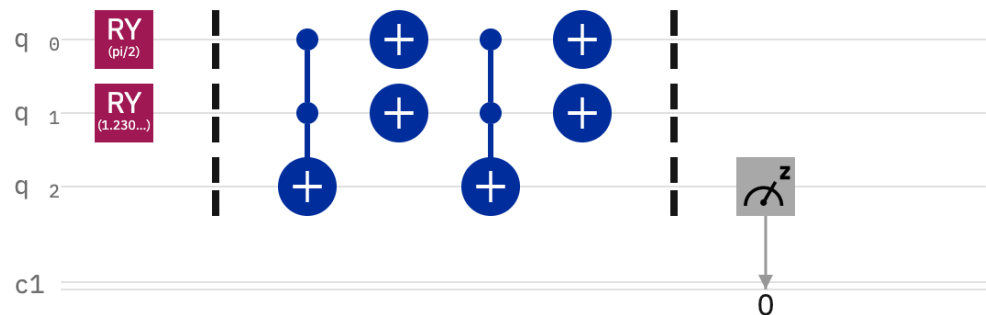


Figure 1. The considered quantum circuit for $|q_0\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$, $|q_1\rangle = \sqrt{\frac{2}{3}}|0\rangle + \frac{1}{\sqrt{3}}|1\rangle$. Changing the initializations of q_0 and q_1 through suitable R_y gates, we can span the different cases.

In Section 3, we show our results, obtained through the interface IBM Quantum Composer (<https://quantum-computing.ibm.com/composer/> accessed on 10 December 2021), an open computing environment, with simulators and quantum computers which can be accessed remotely. Real IBM quantum computers allow up to five qubits.

Because it was not possible to change probabilities directly in Quantum Composer, we created the Qiskit (<https://qiskit.org/> accessed on 10 December 2021) code and then exported the code lines in QASM, the coding environment required for Quantum Composer. In this way, each specific probability amplitude was obtained as a special quantum gate R_y ; e.g., `initial_state = [0.5, 0.5]` (that is, $\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$) was written as $R_y(\frac{\pi}{2})$. This particular state can also be obtained with a Hadamard gate acting on $|0\rangle$. In the following, we present our code. The state $|1\rangle$ is obtained through a Not gate as $x|0\rangle$, because states in Quantum Composer are initialized to 0 by default. In the code, if both $|q_0\rangle$ and $|q_1\rangle$ are $|0\rangle$ (indicating position 0 and failure to reach the target, respectively), then $|q_2\rangle$ is flipped to $|1\rangle$, that is, R_2 goes to position 1. If both $|q_0\rangle$ and $|q_1\rangle$ are $|1\rangle$ (indicating position 1 and success in reaching the target, respectively), then $|q_2\rangle$ is flipped to $|1\rangle$ as well. This second effect is obtained by adding Not gates (x) before qubits $|q_0\rangle$ and $|q_1\rangle$. If $|q_0\rangle$ and $|q_1\rangle$ are different between them, then $|q_2\rangle$ remains in the default state $|0\rangle$.

```
OPENQASM 2.0;
include 'qelib1.inc';
qreg q[3];
creg c[1]; //classic qubit for the measure
//states are initialized as 0 by default.
ry(pi/2) q[0]; // to obtain 0.5, 0.5 as amplitudes
//x q[1]; // to obtain input amplitude 0.0, 1.0, that is, eigenstate |1>
ry(1.2309594) q[1]; // for input amplitudes 0.7, 0.3
// 1.9106332 for input amplitudes 0.3, 0.7
barrier q[0], q[1], q[2];
ccx q[0], q[1], q[2];
x q[0];
x q[1];
ccx q[0], q[1], q[2];
x q[0];
x q[1];
barrier q[0], q[1], q[2];
measure q[2] -> c[0];
```

3. Results

The dataset used in the experiment is summarized in Table 2. To replicate the experiment, it is sufficient to create an account on IBM Quantum (<https://www.ibm.com/quantum-computing/> accessed on 10 December 2021), open a Quantum Composer new file, type the given code, and customize the initial states according to the $|q_0\rangle, |q_1\rangle$ entries in Table 1. In IBM Quantum Composer, states are initialized by default as $|0\rangle$. To have $|1\rangle$, it is sufficient to use a NOT gate: $x|0\rangle = |1\rangle$. To have the $1/\sqrt{2}$ superposition of 0 and 1, one can use the Hadamard gate H or the $R_y(\pi/2)$ gate. To obtain the other combinations, it is necessary to opportunely change the angle of the $R_y(\theta)$ gate. Then, one can hit the “Setup and Run” command, selecting one of the quantum computers available or one of the simulators, and waiting in a queue for the required time. The obtained values can slightly fluctuate because of quantum error, and because of the very nature of quantum world: probabilistic. No other parameters are required in this experiment. We tested the following combinations of initial states:

$$\begin{aligned}
 |q_0\rangle &= |1\rangle, & |q_1\rangle &= |1\rangle; \\
 |q_0\rangle &= |1\rangle, & |q_1\rangle &= |0\rangle; \\
 |q_0\rangle &= \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle, & |q_1\rangle &= |0\rangle; \\
 |q_0\rangle &= \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle, & |q_1\rangle &= |1\rangle; \\
 |q_0\rangle &= |0\rangle, & |q_1\rangle &= \sqrt{\frac{2}{3}}|0\rangle + \frac{1}{\sqrt{3}}|1\rangle; \\
 |q_0\rangle &= |1\rangle, & |q_1\rangle &= \sqrt{\frac{2}{3}}|0\rangle + \sqrt{\frac{2}{3}}|1\rangle; \\
 |q_0\rangle &= |0\rangle, & |q_1\rangle &= \frac{1}{\sqrt{3}}|0\rangle + \sqrt{\frac{2}{3}}|1\rangle; \\
 |q_0\rangle &= |1\rangle, & |q_1\rangle &= \frac{1}{\sqrt{3}}|0\rangle + \sqrt{\frac{2}{3}}|1\rangle; \\
 |q_0\rangle &= \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle, & |q_1\rangle &= \sqrt{\frac{2}{3}}|0\rangle + \frac{1}{\sqrt{3}}|1\rangle; \\
 |q_0\rangle &= \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle, & |q_1\rangle &= \frac{1}{\sqrt{3}}|0\rangle + \sqrt{\frac{2}{3}}|1\rangle.
 \end{aligned} \tag{10}$$

In Table 2, we compare the outcomes obtained with IBM simulators and IBM quantum computers. With the quantum computer, the results are mostly as expected, but the noise is high; the simulators give a result that perfectly matches theoretical expectations.

A case of particular interest are the eigenstates, to measure the amount of possible quantum noise, and the case with random position of R_1 ($|q_0\rangle$ with 50% of chances to be in 0 and to be in 1), and maximal certitude on the reward information ($|q_1\rangle$ with 100% of chances to be in 1 and 0% to be in 0, or vice versa).

Results with real quantum computers can differ according to the chosen machine, and they present a higher noise. The noise is particularly evident when we tested q_0, q_1 as eigenstates. The simulator gave us all the outcomes of q_2 as 0 or as 1, as expected from Table 1, while the quantum computer also presented a non-zero population of the unexpected outcome. Each run of the code gave 1024 shots of measurements. Fluctuations at each run of the code, not displayed here, are present for both typologies of computing devices.

It is not possible to carry out a comparison with other methods, because the proposed methodology is new. To the best of our knowledge, the only case of a close-related research is the application proposed by Koukam et al. [18], where an IBM Quantum computer has been used for a reactive agent. In their study, a single robot (and not a swarm) modifies its behavior according to the external information recovered through suitable sensors.

Table 2. Comparisons of measurement outcomes for 1024 shots of the code. S indicates the simulator; C stands for a real quantum computer located in Bogotá, Colombia.

$ q_0\rangle$	$ q_1\rangle$	$ q_2\rangle$ (Expected)	Counts for 0, 1	Device
$ 1\rangle$	$ 1\rangle$	1	0; 1024 455; 569	S C
$ 1\rangle$	$ 0\rangle$	0	1024; 0 527; 497	S C
$\frac{1}{\sqrt{2}} 0\rangle + \frac{1}{\sqrt{2}} 1\rangle$	$ 1\rangle$	uncertain	494; 530 502; 522	S C
$\frac{1}{\sqrt{2}} 0\rangle + \frac{1}{\sqrt{2}} 1\rangle$	$ 0\rangle$	uncertain	527; 497 431; 593	S C
$ 0\rangle$	$\sqrt{\frac{2}{3}} 0\rangle + \frac{1}{\sqrt{3}} 1\rangle$	0	686; 338 531; 493	S C
$ 1\rangle$	$\sqrt{\frac{2}{3}} 0\rangle + \frac{1}{\sqrt{3}} 1\rangle$	0	702; 322 570; 414	S C
$ 0\rangle$	$\frac{1}{\sqrt{3}} 0\rangle + \sqrt{\frac{2}{3}} 1\rangle$	1	315; 709 452; 572	S C
$ 1\rangle$	$\frac{1}{\sqrt{3}} 0\rangle + \sqrt{\frac{2}{3}} 1\rangle$	1	335; 669 376; 648	S C
$\frac{1}{\sqrt{2}} 0\rangle + \frac{1}{\sqrt{2}} 1\rangle$	$\sqrt{\frac{2}{3}} 0\rangle + \frac{1}{\sqrt{3}} 1\rangle$	uncertain	518; 506 351; 673	S C
$\frac{1}{\sqrt{2}} 0\rangle + \frac{1}{\sqrt{2}} 1\rangle$	$\frac{1}{\sqrt{3}} 0\rangle + \sqrt{\frac{2}{3}} 1\rangle$	uncertain	508; 516 433; 591	S C

4. Discussion and Conclusions

In this article, we started from theoretical considerations on swarms of robots, from ideal, abstract swarms down to typologies of swarms with specific existing realizations. We proposed a matrix description of a generic swarm, trying to connect the action of each single robot with the overall behavior. The proposed swarm matrix is a block matrix with sub-matrices indicating the motion of single robots, and off-diagonal blocks representing pairwise interaction terms, with signals from the i -th robot and behavioral response for the j -th robot. Aiming to shape an example of these interaction terms, we exploit the basics of quantum computing to work out a toy model with four qubits, implementing it through IBM computational resources and analyzing the obtained results.

The presented results provide information regarding output states' likelihood. This information is related to the decision-making outcomes in a toy 2-robot swarm. The input represents incoming information from the first robot, and the output indicates the behavioral response from the second robot. The proposed quantum circuit acts as an overall gate, corresponding to an interaction term at time t in the proposed matrix representation. The interaction terms model local behaviors related to global behaviors through the proposed matrix.

Let us briefly analyze the pros and cons of the proposed strategy. The cons are mainly related to technological limitations: IBM Quantum computers are freely available only up to five qubits, while IBM Quantum simulators are available up to 32, 63, 100 and, very recently (after the submission of this article), 5000 qubits. The pros include computational power of quantum computing, proved for classic problems such as integer factorization, and generalization power of the proposed theoretical framework. In fact, we can consider robot interactions as instances of social behaviors, be they of people, animals,

or neuronal entities. Thus, the proposed methodology is not limited to robots: it can instead be applied to interactions amongst multiple entities, constituting complex systems with an emergent behavior.

Future developments of the proposed research will involve the construction of more complex models, with a higher number of qubits, describing more degrees of freedom of robots. From the conceptual level, additional degrees of freedom can help us model more and more complex group behaviors, also imitating aspects of natural swarm scenarios [1,2]. Another point to be addressed in future research is map building as an effect of memory where robots explore their environment looking for the target. Further research can draw upon existing studies on agent-based decision-making [27], localization and map building developed for single [28] and multiple robots [29,30], to build a theoretical unification and envisage a *quantum perspective* on swarms, also exploiting quantum computing computational resources.

Author Contributions: Conceptualization, M.M., V.S., and A.C.; methodology, M.M. and V.S.; software, M.M.; validation, M.M.; formal analysis, M.M.; investigation, M.M., V.S. and A.C.; resources, A.C.; data curation, M.M., V.S. and A.C.; writing—original draft preparation, M.M.; writing—review and editing, M.M., V.S. and A.C.; visualization, M.M.; supervision, V.S. and A.C.; project administration, A.C. and V.S.; funding acquisition, A.C. and V.S. All authors have read and agreed to the published version of the manuscript.

Funding: The research leading to these results takes place within the framework of the project “ARES, Autonomous Robotics for the Extended Ship,” funded by the Italian Ministry of University and Research under grant agreement ARS01_00682.

Data Availability Statement: The data presented in this study are openly available in GitHub: <https://github.com/medusamedusa/quantum-world> (accessed on 10 December 2021).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Delcourt, J.; Bode, N.W.; Denoël, M. Collective Vortex Behaviors: Diversity, Proximate, and Ultimate Causes of Circular Animal Group Movements. *Q. Rev. Biol.* **2016**, *91*, 1–24. [CrossRef] [PubMed]
2. Hemelrijk, C.H.; Hildenbrandt, H. Schools of fish and flocks of birds: Their shape and internal structure by self-organization. *Interface Focus* **2012**, *2*, 726–737. [CrossRef] [PubMed]
3. Hamann, H. *Swarm Robotics: A Formal Approach*; Springer: Cham, Switzerland, 2018.
4. Dorigo, M.; Theraulaz, G.; Trianni, V. Swarm Robotics: Past, Present, and Future [Point of View]. *Proc. IEEE* **2021**, *109*, 1152–1165. Available online: <https://hal.archives-ouvertes.fr/hal-03362874/document> (accessed on 10 December 2021). [CrossRef]
5. Schranz, M.; Umlauf, M.; Sende, M.; Elmenreich, W. Swarm Robotic Behaviors and Current Applications. *Front. Robot. AI* **2020**, *7*, 36. [CrossRef] [PubMed]
6. Eberhart, R.; Shi, Y.; Kennedy, J. *Swarm Intelligence*; The Morgan Kaufmann Series in Artificial Intelligence; Morgan Kaufman: Burlington, MA, USA, 2001.
7. Bishop, J.M. Stochastic Searching Networks. In Proceedings of the 1st IEEE Conference Artificial Neural Networks, London, UK, 16–18 October 1989. Available online: <http://www.reading.ac.uk/web/files/sse/sds-ssn.pdf> (accessed on 10 December 2021).
8. Dorigo, M.; Birattari, M.; Stutzle, T. Ant colony optimization. *IEEE Comput. Intell. Mag.* **2001**, *4*, 28–39.
9. Rosenberg, J.L. Human Swarms, a real-time method for collective intelligence. In Proceedings of the European Conference on Artificial Life, York, UK, 20–24 July 2015; pp. 658–659.
10. Dong, X.; Sitti, M. Controlling two-dimensional collective formation and cooperative behavior of magnetic microrobot swarms. *Int. J. Robot. Res.* **2020**, *39*, 5. Available online: <https://journals.sagepub.com/doi/full/10.1177/0278364920903107> (accessed on 10 December 2021). [CrossRef]
11. Preskill, J. Quantum computing 40 years later. In *Feynman Lectures on Computation*, 2nd ed.; Hey, A.J.G., Ed.; Taylor & Francis Group: Abingdon, UK, 2021.
12. Stolze, J.; Suter, D. *Quantum Computing: A Short Course from Theory to Experiment*; Wiley: Weinheim, Germany, 2004.
13. Wichert, A. *Principles of Quantum Artificial Intelligence*; World Scientific: Singapore, 2020.
14. Sabra, A.; Fung, W.-K. A Fuzzy Cooperative Localisation Framework for Underwater Robotic Swarms. *Sensors* **2020**, *20*, 5496. [CrossRef]
15. Nădăban, S. From Classical Logic to Fuzzy Logic and Quantum Logic: A General View. *Int. J. Comput. Commun. Control.* **2021**, *16*, 1841–9836. [CrossRef]

16. Li, Z.; Liu, W.; Li, E.G.; Li, L. Path Planning Method for AUV Docking Based on Adaptive Quantum-Behaved Particle Swarm Optimization. *IEEE Access Multidiscip.* **2019**, *7*, 78665–78674. [[CrossRef](#)]
17. Ivancevic, V.G. Entangled Intelligence: Quantum Computation for swarm robotics. *Math. Eng. Sci. Aerosp.* **2018**, *7*, 441–451.
18. Koukam, A.; Abbas-Turki, A.; Hilaire, V.; Ruichek, Y. Towards a Quantum Modeling Approach to Reactive Agents. In Proceedings of the 2021 IEEE International Conference on Quantum Computing and Engineering (QCE), Broomfield, CO, USA, 18–22 October 2021; pp. 130–136.
19. Bonabeau, E.; Theraulaz, G.; Dorigo, M. *Swarm Intelligence: From Natural to Artificial Systems*; Oxford University Press: New York, NY, USA, 1999; pp. 3–77.
20. Şahin, E. Swarm robotics: From sources of inspiration to domains of application. In *International Workshop on Swarm Robotics*; Şahin, E., Spears, W.M., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2004; Volume 3342.
21. Berlinger, F.; Gauci, M.; Nagpal, R. Implicit coordination for 3D underwater collective behaviors in a fish-inspired robot swarm. *Sci. Robot.* **2021**, *6*, 50. [[CrossRef](#)] [[PubMed](#)]
22. Soria, E.; Schiano, F.; Floreano, D. Predictive control of aerial swarms in cluttered environments. *Nat. Mach. Intell.* **2021**, *3*, 545–554. [[CrossRef](#)]
23. Ozkan-Aydin, Y.; Goldman, D.I. Self-reconfigurable multilegged robot swarms collectively accomplish challenging terradynamic tasks. *Sci. Robot.* **2021**, *6*, 56. [[CrossRef](#)] [[PubMed](#)]
24. Mac Lane, S. *Categories for the Working Mathematician*; Springer: New York, NY, USA, 1978.
25. Mannone, M. *Mathematics, Nature, Art*; Palermo University Press: Palermo, Italy, 2019.
26. Ismail, Z.H.; Hamami, M.G.M. Systematic Literature Review of Swarm Robotics Strategies Applied to Target Search Problem with Environment Constraints. *Appl. Sci.* **2021**, *11*, 2383. [[CrossRef](#)]
27. Chella, A.; Lanza, F.; Seidita, V. Decision Process in Human-Agent Interaction: Extending Jason Reasoning Cycle. In *Engineering Multi-Agent Systems*; Weyns, D., Mascardi, V., Ricci, A., Eds.; EMAS 2018. Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2018; Volume 11375, pp. 320–339.
28. Dissanayake, M.W.M.; Newman, P.; Clark, S.; Durrant-Whyte, H.F.; Csorba, M. A Solution to Simultaneous Localization and Map Building (SLAM) Problem. *IEEE Trans. Robot. Autom.* **2001**, *17*, 229–241. [[CrossRef](#)]
29. Zhou, Y.; Hu, H.; Liu, Y.; Lin, S.; Ding, Z. A Real-Time and Fully Distributed Approach to Motion Planning for Multirobot Systems. *IEEE Trans. Syst. Man Cybern. Syst.* **2019**, *49*, 2636–2660. [[CrossRef](#)]
30. Umari, H.; Mukhopadhyay, S. Autonomous Robotic Exploration Based on Multiple Rapidly-exploring Randomized Trees. In Proceedings of the IEEE/RJS International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017.