

Density matrix-based dynamics for quantum robotic swarms

Maria Mannone^{a,b,c}^{*}, Mahathi Anand^d, Peppino Fazio^{e,f}, Abdalla Swikir^g

^a ICAR, National Research Council (CNR), Palermo, Italy

^b Institute of Physics and Astronomy, University of Potsdam, Germany

^c Potsdam Institute for Climate Impact Research (PIK), Member of the Leibniz Association, Germany

^d Learning Systems and Robotics Lab, Technical University of Munich (TUM), Germany

^e Department of Molecular Sciences and Nanosystems (DSMN), Ca' Foscari University of Venice, Italy

^f VSB, Technical University of Ostrava, Czechia

^g Mohamed bin Zayed University of Artificial Intelligence (MBZUAI), Abu Dhabi, United Arab Emirates

ARTICLE INFO

MSC:

03.65.Ca

87.85.St

Keywords:

Quantum mechanics

Swarm intelligence

Multi-robot coordination

Networks

ABSTRACT

In a robotic swarm, parameters such as position and proximity to the target can be described in terms of probability amplitudes. This idea led to recent studies on a quantum approach to the definition of the swarm, including a block-matrix representation. However, the size of such matrix-based representation increases drastically with the swarm size, making them impractical for large swarms. Hence, in this work, we propose a new approach for modeling robotic swarms and robotic networks by considering them as mixed quantum states that can be represented mathematically via density matrices. The size of such an approach only depends on the available degrees of freedom of the robot, and not its swarm size and thus scales well to large swarms. Moreover, it also enables the extraction of local information of the robots from the global swarm information contained in the density matrices, facilitating decentralized behavior that aligns with the collective swarm behavior. Our approach is validated on several simulations including large-scale swarms of up to 1000 robots. Finally, we provide some directions for future research that could potentially widen the impact of our approach.

1. Introduction

At the frontier of classical and quantum physics: defining a swarm of robots small enough to operate under quantum effects is a challenging task. Here, the considered problem is to obtain a suitable description of the swarm as a whole system composed of individual robots. As such, the domain of swarm robotics has piqued considerable interest over the last decade. Much of this attention is attributed to the deployment of multi-agent robot teams to accomplish a wide variety of challenging tasks such as target monitoring, surveillance, environmental sensing, disaster recovery and exploration [1]. On the other hand, the development of several bio-inspired algorithms [2–5] including deployment of robot swarms inspired by a flock of birds [6], insects [7,8], fish [9], and even the microscopic world [10] has led to the progressive miniaturization of robotic swarms, making it possible to advance robotic solutions in a wide range of applications including medical technology [11,12] and telecommunications [13,14]. However, the smaller the robotic swarm, the harder it is to control. In particular, as systems approach the nanoscale, quantum effects start to dominate, interactions between agents and their environment are influenced by quantum phenomena, and systems can no longer be explained by classical physics. As a result,

controlling such robotic swarms is a challenging problem. Moreover, standard control algorithms require precise physical modeling and can no longer be applied to robotic swarms that operate under quantum effects. Therefore, it is critical to account for quantum-mechanical effects when modeling and reasoning about nano robotic swarms [15].

The field of quantum computing uses quantum bits (*qubit*) to extend the values of the standard 0, 1 to all the real values comprised between them (the interval [0, 1]) [16]. The paradigm offers a significant computational advantage to the traditional classical computing by taking advantage of quantum-mechanical principles including superposition and entanglement, and has been utilized for modeling and simulating forward kinematics of industrial robot manipulators [17, 18], human robot interaction [19,20], and robot perception [21,22]. Moreover, quantum computing has been central in the advancement of machine learning techniques [23–25], motion planning [26–28], and optimization [18,29], and has also been applied in the context of swarm robotics [30–32]. However there are significant challenges in utilizing quantum algorithms in the context of swarm robotics. For example, problems concerning stability [33] and robustness [34] of the algorithms have not been well-investigated, though it has been

* Corresponding author at: ICAR, National Research Council (CNR), Palermo, Italy.

E-mail addresses: maria.mannone@icar.cnr.it, maria.mannone@uni-potsdam.de (M. Mannone).

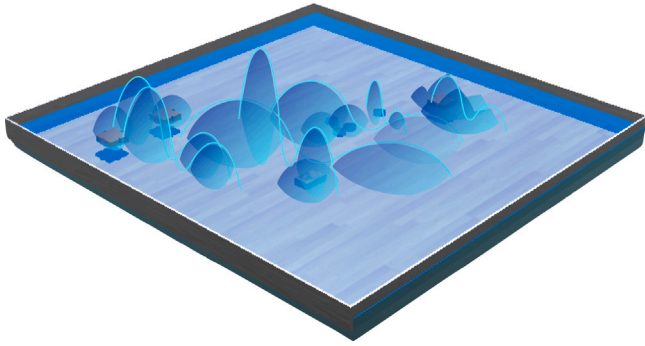


Fig. 1. Pictorial representation of the transition from a classic robotic swarm to a quantum-robotic swarm: in a given arena, each robot is considered in terms of a probability amplitude to be found in a certain space position.

suggested that algorithms such as quantum particle swarm optimization can lead to improvements in path-planning convergence [29]. This is due to the lack of appropriate modeling formalisms for quantum swarms. Therefore, in this article, we prepare the necessary ground for advanced control problems for quantum swarms by suggesting appropriate techniques to model them mathematically.

Concerning robotic swarm modeling, in [35], a description of a robotic swarm is proposed in terms of a nested matrix whose diagonal submatrices contain the parameters of a single robot, while the off-diagonal elements contain the information exchange between the robots. Thus, the size of the matrix depends upon the degrees of freedom of each robot, the exchanged information, and the number of robots. In particular, the size of the matrix increases with the number of robots, which constitutes a computational problem for large swarms. Though the article does not specifically concern quantum robots, the authors leverage quantum computing to quantize some information (e.g., position as well as proximity to the target) to model local interactions between the robots. Moreover, [35] only considers only the local-to-global approach, i.e., the global swarm behavior emerged from the local pairwise interactions, but not the vice-versa.

As a solution to the problems emerging in [35], in this article we propose a different approach for modeling quantum robotic swarms. Particularly, we use the notion of *density matrices* [36], first introduced for the treatment of epistemic uncertainty on systems and their noisiness, for describing robotic swarms. Particularly, we model a quantum robotic swarm as a mixed state given by a superposition of pure states that represent individual robots, mathematically indicating them with a density matrix. By shaping the density matrix for a robotic swarm, one can develop a suitable overall dynamics for the whole swarm as an operator acting on the corresponding density matrix. It is important to note that in our approach, the size of the density matrices is rendered independent from the swarm size. In fact, it only depends on the number of degrees of freedom of the system, at the condition of treating robots as particles within a “cloud”: the swarm is thus investigated as a whole, rather than as a collection of distinguishable objects (see Fig. 1). Moreover, information of local robots can also be extracted from the density matrices through partial trace operation, thus also facilitating a global-to-local approach in the modeling of quantum swarms.

Another crucial difference of our approach with [35] is the neglect of interaction terms. In particular, while [35] considers the pair-wise interaction terms between the robots, which contribute to the size of matrices obtained for the swarm, in our paper, we can neglect these interactions. This is rooted in the fact that the density matrices already consider the effect of the interaction terms through the mixing of pure quantum states. Therefore, our approach offers a scalable yet efficient solution for modeling the information contained in the swarm without introducing any unnecessary redundancies.

The article is organized as follows. In Section 2 we present the necessary preliminaries from quantum computing, such as density matrices and their operations, that are useful for our results. After having introduced the adopted quantum notation for single robots (Section 3), in Section 4 we propose our modeling framework and present its advantages. In Section 5 we demonstrate the application of our results through simple examples, also showing their implementation in Python with the library Qiskit. In Section 5.5 we discuss in detail the problem of time evolution, proposing a pseudocode of local-global swarm behavior, with a special reference to the network modeling in telecommunications, ending the section with a toy example of implementation. Finally, in Section 6 we summarize our results and present some extensions and applications of our approach to be considered in the future.

2. Notations and preliminaries

2.1. Notations

The set of real numbers are denoted by \mathbb{R} , and \mathbb{R}^n denotes the n -dimensional real space. The set of complex numbers is denoted by \mathbb{C} , and a complex matrix $A \in \mathbb{C}^{m \times n}$ is of size $m \times n$. We use the standard bra-ket notation to describe a quantum state: $|\psi\rangle$ is the ket vector, a quantum pure state in a (complex) inner product vector space V corresponding to an eigenstate of an operator, and $\langle\psi|$ is the bra vector, the complex conjugate of $|\psi\rangle$. Moreover, the notation $\langle\psi|\psi\rangle$ indicates the inner product, i.e., the product of bra and ket vectors, while $|\psi\rangle\langle\psi|$ denotes the outer product.

For a matrix A , $Tr(A)$ is the trace operation, the sum of diagonal elements of a matrix. The operator Tr indicates the Trace operation, defined as the sum of the diagonal elements of the matrix, that is $\sum_i T_{i,i}$, which is indicated as $Tr(T) := \sum_i \langle i|T|i\rangle$ in quantum mechanics. A_t is a generic time-dependent operator; its complex conjugate is A_t^\dagger , such that $(A_t^\dagger)^\dagger = A_t$. If A is Hermitian (self-adjoint), thus it is equal to its complex conjugate: $A_t^\dagger = A_t$. Finally, we use \otimes to describe the tensor product.

2.2. Preliminaries

In quantum mechanics, the behavior of a quantum state is described by the Schrödinger equation [37]:

$$i\hbar \frac{d}{dt} |\psi(t)\rangle = \mathbf{H} |\psi(t)\rangle, \quad (1)$$

where \mathbf{H} is the Hamiltonian operator, i is the imaginary unit, \hbar is the reduced Planck constant that will be set equal to 1 here, as often done in quantum physics, and $|\psi(t)\rangle$ represents the wavefunction at time t as a mathematical description of the *pure quantum state* in terms of its probability amplitude. In fact, a pure quantum state describes the position of a robot along its dimensional axes. For example, in case of one-dimensional movement, the x - position of a robot is given by

$$|\psi(t)\rangle = \alpha^x |0\rangle + \beta^x |1\rangle, \quad (2)$$

where α^x and β^x are probability amplitudes along $|0\rangle$ (left) and $|1\rangle$ (right), respectively with $\alpha^{x^2} + \beta^{x^2} = 1$. Similarly, in the 2-dimensional (x - y) plane, one obtains the robot position as a pure state that is a superposition of several qubits describing the position along the x and y directions, i.e., as a superposition of the basis states $|00\rangle$, $|01\rangle$, $|10\rangle$, and $|11\rangle$, similar to that defined in (2). When dealing with an ensemble of quantum states, such as in a quantum robotics swarm consisting of multiple robots, one considers a mixture of different pure quantum states, known as *mixed states*. In this case, one uses the more general master equation [38,39]

$$\frac{d}{dt} \rho(t) = \mathcal{L}(\rho(t)), \quad (3)$$

where $\rho(t)$ is the density operator, generalizing the mathematical description of the mixed quantum states, and \mathcal{L} is the Lindblad operator given by

$$\mathcal{L}(\rho(t)) = -i[H, \rho(t)] + \sum_{\alpha} \gamma_{\alpha}(t) \left(L^{\dagger} \rho(t) L - \frac{1}{2} L^{\dagger} L \rho(t) - \frac{1}{2} \rho(t) L^{\dagger} L \right), \quad (4)$$

where the operator L represents the interaction of the system with the environment. Eq. (4) corresponds to a Markovian dynamics when $\gamma_{\alpha} \geq 0$, for all α .

Definition. A density operator [39,40], described by a density matrix, is a Hermitian operator ($\rho(t) = \rho(t)^{\dagger}$), positive ($\rho(t) > 0$) and with unit trace ($Tr[\rho(t)] = 1$).

We now analyze some properties of the density operator and the corresponding matrix. For a pure state, the density matrix is defined as a projector

$$\rho(t) = |\psi(t)\rangle\langle\psi(t)|, \quad (5)$$

while for a mixed state we have

$$\rho(t) = \sum_i p_i |\psi_i(t)\rangle\langle\psi_i(t)|, \quad (6)$$

where p_i is the i th weight, $p_i \in \mathbb{R}$, and $\sum_i p_i = 1$. The index i denotes the i th robot. It can be easily observed from the above that a pure state cannot be represented as the mixture of other states. Moreover, while a pure state is described by the wavefunction $|\psi(t)\rangle$, a mixed state is given by a superposition $\sum_i p_i |\psi_i(t)\rangle$. Note that for a pure state, we have $\rho^2(t) = \rho(t)$ and $Tr[\rho^2(t)] = 1$.

As a general note, to each quantum state, one can determine a corresponding density matrix, but given a density matrix, there can be more than a single quantum state that corresponds to it, up to a phase difference [39]. In fact, the density matrix is only a statistical object which captures the distribution of qubit information. The process of mapping a density matrix to a possible quantum state is called *quantum state tomography*, but it is a complex process that presents no particular advantage to us and therefore, irrelevant to this work [39]. We present now the following remark that will be useful for our research.

Remark. Two quantum states with similar density matrices are themselves similar, even though multiple states can correspond to a single density matrix. This similarity, or *closeness* between the density matrices is quantified using a metric known as fidelity, with values comprised between 0 (low fidelity) and 1 (high fidelity) [39]. A commonly used metric for this is the trace distance, which provides information on how distinguishable two quantum states are. The trace distance between two density matrices $\rho_1(t)$ and $\rho_2(t)$ is defined as:

$$D[\rho_1(t), \rho_2(t)] = \frac{1}{2} Tr|\rho_1(t) - \rho_2(t)|, \quad (7)$$

with

$$|\rho_1(t) - \rho_2(t)| := \sqrt{(\rho_1(t) - \rho_2(t))(\rho_1(t) - \rho_2(t))}. \quad (8)$$

For a system that consists of two parts (e.g., two swarm particles, a particle and an agglomerate of particles, or two agglomerates of particles) A and B, each of them described by density matrices $\rho_A(t)$ and $\rho_B(t)$ respectively, one can describe the density matrix of the system by

$$\rho_{AB}(t) = \rho_A(t) \otimes \rho_B(t). \quad (9)$$

To recover the information on state A from the matrix $\rho_{AB}(t)$, we need to compute its partial trace with respect to the state B, which is defined as

$$\begin{aligned} Tr_B[\rho_{AB}(t)] &= \sum_B [\rho_{AB}(t)] \\ &= \sum_n \rho_A(t) \otimes \langle n | \rho_B(t) | j \rangle \\ &= \rho_A(t), \end{aligned} \quad (10)$$

where n indicates the qubit which can be 0 or 1, and thus the sum involves the elements $\langle 0 | \dots | 0 \rangle$ and $\langle 1 | \dots | 1 \rangle$, that is, the measurement of the density operator along states 0 and 1, respectively. The operation

of partial trace is helpful, for example, to detach the information of a system from its environment. When we have a system composed by multiple elements, we can consider the sub-system of interest as *the* system, and the other elements as part of the environment (as stated in the previous section).

In Section 4, we use the idea of tensor product to build up a density matrix for the robotic swarm starting from individual density matrices, and the partial trace to achieve the inverse result, i.e., recovering information on single robots from the overall matrix.

3. Quantum robotic swarm modeling

Before moving to the density matrix-based description of a quantum robotic swarm, we establish a correspondence between the parameters of the swarm by introducing the adopted bra-ket notation for a single robot. To do so, similarly to [35], we consider the position of each robot in the 2-dimensional plane as described by the superimposition of qubits along the x and y directions (as described in Section 2.2), and its success in target finding as probability amplitudes. Specifically, these amplitudes are used to construct the density matrices of the single robots. Then, the density matrix of the swarm can be obtained via the tensor product of the individual density matrices.

Let us consider an \mathcal{R}^2 mobility plane (x and y axes), with a certain degree of success in reaching the target. Let $|\psi_{R_i}(t)\rangle$ be the state of robot R_i at time t , defined as

$$\begin{aligned} |\psi_{R_i}(t)\rangle &= \alpha_i^1(t)|111\rangle + \alpha_i^2(t)|110\rangle + \alpha_i^3(t)|100\rangle + \\ &\quad \alpha_i^4(t)|000\rangle + \alpha_i^5(t)|001\rangle + \alpha_i^6(t)|010\rangle + \\ &\quad \alpha_i^7(t)|101\rangle + \alpha_i^8(t)|011\rangle. \end{aligned} \quad (11)$$

$|\psi_{R_i}(t)\rangle$ is the superimposition state vector of 3 qubits, of the form $|a, b, c\rangle$, with qubits a, b, c , where the first qubit indicates the position projection along a segment $[0, 1]$ on the x -axis (0 and 1 are the two extremes), the second qubit denotes the projection of the position along a segment $[0, 1]$ on the y -axis (similarly defined), and the third qubit indicates the success of finding the target, i.e., $|0\rangle$ indicates failure and $|1\rangle$ indicates success; the coefficients $\alpha_i^j(t)$, with i indicating the robot, and j indicating the eigenvector, denote the probability amplitudes for the respective eigenstates/eigenvectors, satisfying $\sum_{j=1}^8 |\alpha_i^j(t)|^2 = 1$.

4. Density matrix-based modeling of quantum robotic swarm

Considering the quantum state of a single robot, expressed in Eq. (11), we can now propose the density-matrix depiction of the swarm of robots. Then, for a robot R_i , the density operator is thus

$$\rho_{R_i}(t) = |\psi_{R_i}(t)\rangle\langle\psi_{R_i}(t)|, \quad (12)$$

and the state of R_i can be written as a column matrix

$$|\psi_{R_i}(t)\rangle = \begin{pmatrix} \alpha_i^1(t) \\ \alpha_i^2(t) \\ \vdots \\ \alpha_i^8(t) \end{pmatrix}. \quad (13)$$

The density matrix is thus obtained via the product between the column matrix and the row matrix of complex conjugates coefficients, as follows:

$$\rho_{R_i}(t) = \begin{pmatrix} \alpha_i^1(t) \\ \alpha_i^2(t) \\ \vdots \\ \alpha_i^8(t) \end{pmatrix} \left(\alpha_i^{1,*}(t), \alpha_i^{2,*}(t), \dots, \alpha_i^{8,*}(t) \right). \quad (14)$$

Remark. In this article, we only consider real values for the probability amplitudes, so the complex conjugates are equal to the original values. The time evolution of a single robot can be obtained via a suitable time-evolution operator acting on the corresponding density operator.

Such an operator can include a dependence on the state of the other robots, acting according to the scope of the robotic mission. In [35], the swarm behavior emerged from pairwise robotic interactions modeled upon a quantum circuit. Here, we consider a more quantum-mechanics style approach. Also, the matrix size does not increase when the swarm becomes bigger (with more units).

Then, the density matrix of a swarm of N robots is computed as

$$\rho_{swarm}(t) = \rho_{R_1}(t) \otimes \cdots \otimes \rho_{R_N}(t), \quad (15)$$

and the information on the i th robot can be retrieved performing the partial trace on all the other robots using

$$\rho_{R_i}(t) = Tr_{R_{j \neq i}(t)}(\rho_{swarm}(t)), \quad (16)$$

where $Tr_{R_{j \neq i}(t)}$ denotes the sum of the diagonal elements of $\rho(t)$ with the exclusion of the ones indexed as i , that is, corresponding to the i th robot. The formulation of a quantum swarm in terms of tensor products of density matrices unfortunately has a major drawback: it results in large matrix sizes for large swarm sizes. An alternate approach to alleviate this problem is to consider the swarm as a mixed state given by the superposition of pure states that represent single robots. This reduces the size of the matrix associated with the robot swarm, making the description more concise, compared to [35]. Indeed, one can also consider single robots as mixed states, in which case, one needs to resort once again to consider the tensor product of density matrices for a clear representation of density matrix of the swarm.

To illustrate this, consider an N -robot swarm, where the density matrix of the i th robot is given by $\rho_i(t) = |\psi_{R_i}(t)\rangle\langle\psi_{R_i}(t)|$. Considering the swarm as a *mixed state*, its density matrix can be simply computed as

$$\rho_{swarm}(t) = \sum_{i=1}^N w_i \rho_i(t), \quad (17)$$

where w_i , with $i \in \{1, \dots, N\}$ robots, are constant weights such that $\sum_i w_i = 1$. If the contribution of all the robots is equal, Eq. (17) becomes

$$\begin{aligned} \rho_{swarm}(t) &= \frac{1}{N} \sum_{i=1}^N \rho_i(t) \\ &= \frac{1}{N} \sum_{i=1}^N |\psi_{R_i}(t)\rangle\langle\psi_{R_i}(t)| \end{aligned} \quad (18)$$

Eq. (17) involves the sum of matrices, not their tensor product, and thus the dimension of the density matrix of the swarm, while seen as a mixed state, remains the same, as the dimensions of the single density matrices for the robots.

Moreover, while the swarm modeling approach presented in [35] utilizes interaction terms in the swarm submatrices to describe the interactions between robots, in this work, we neglect the interaction terms by considering that each robot sends all information to all the other robots, and each robot receives information from all the other robots. Thus, copying this information in the interaction sub-matrices would just be redundant. Note that the effect of “interaction” are automatically included in the density-matrix depiction of the swarm, seen as a whole system in a mixed state, resulting from the mixing of single states of the single robots. However, the utilization of explicit interaction terms largely depends on the size of the considered devices: while bigger robots may use classical wireless communications, for micro/nano scale devices, newer opportunities arise [41], since they are nearer to quantum systems [42].

In the following section, we illustrate our proposed method with several examples.

5. Examples

5.1. Pure states

As a first example, we consider a toy-swarm of two robots, R_1 and R_2 , moving along a line of length 1, i.e., from $x = 0$ to $x = 1$. Suppose that the target is located in $x = 1$. From Eq. (11), we can write

$$|\psi_{R_i}(t)\rangle = \alpha_i^1(t)|11\rangle + \alpha_i^2(t)|10\rangle + \alpha_i^3(t)|01\rangle + \alpha_i^4(t)|00\rangle, \quad (19)$$

where the first qubit indicates the position along the x axis, and the second qubit the degree of success in target finding, and $\sum_j |\alpha_i^j(t)|^2 = 1$. With a slight abuse of notation, we represent the position of the robot instead with the peak amplitude of the wavefunction of the position [35]. Moreover, we remove the time argument from the state for notational simplicity by assuming that the positions are considered at a pre-defined time instant. Now suppose that R_1 is at $x = 0.5$ (the peak of the wavefunction is centered in 0.5), and it has 0 success (the peak of the “success” wavefunction is in 0). On the other hand, R_2 at $x = 1$ and with a successful result in target finding (1); it is thus in an eigenstate. Substituting the non-null coefficients $\alpha_1^2 = \alpha_1^4 = \sqrt{1/2}$ and $\alpha_2^1 = 1$ for the two robots, respectively, we obtain

$$|\psi_1\rangle = \frac{1}{\sqrt{2}}(|10\rangle + |00\rangle), \quad (20)$$

$$|\psi_2\rangle = |11\rangle, \quad (21)$$

Correspondingly, the density matrix of R_1 is given by

$$\rho_1 = |\psi_1\rangle\langle\psi_1| = \frac{1}{2}(|10\rangle\langle 10| + |10\rangle\langle 00| + |00\rangle\langle 10| + |00\rangle\langle 00|), \quad (22)$$

and that of R_2 is given by

$$\rho_2 = |\psi_2\rangle\langle\psi_2| = |11\rangle\langle 11|. \quad (23)$$

We recall the definition of the density matrix ρ_j , computed from the ket $|\psi_j\rangle$ and the bra $\langle\psi_j|$. A ket lives in a Hilbert space \mathcal{H} , the bra in its dual space (that, for finite-dimensional Hilbert spaces, is its conjugate transpose). The density operator acts on the Hilbert space, and the density matrix is its representation in terms of matrix. As a general note, in this article, we adopt the term “matrix” to denote both the operator and its matrix representation. The density operator lives in the space of bounded linear operators acting on the Hilbert space: $\rho_j \in \mathcal{B}(\mathcal{H})$. To obtain the matrix representation, we have to consider, as rows, the kets $|11\rangle, |10\rangle, |01\rangle, |00\rangle$, and as columns, the bras $\langle 11|, \langle 10|, \langle 01|, \langle 00|$. In this way, for the density matrices of Eqs. (22) and (23), we can obtain the following matrix forms:

$$\rho_1 = \frac{1}{2} \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}, \quad \rho_2 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

The density matrix of the toy swarm is obtained as their tensor product. If we have the state as the initial information, and we want to recover the probability of the robot to be at $x = 1$, we compute:

$$|\langle 1_x | \psi_1 \rangle|^2 = 0.5, \quad |\langle 1_x | \psi_2 \rangle|^2 = 1. \quad (24)$$

The probability amplitude that the swarm barycenter is in $x = 1$ is the mean of the two terms in Eq. (24), i.e., 0.75. An ideal swarm with two robots, both centered on the target in 1 and successful, of course has a barycenter with probability 1 to be found on 1. A measure of distance between the swarm barycenter and the ideal-swarm barycenter is the difference $|0.75 - 1|$. The information on positions of single robots can also be retrieved from the density matrix; this is useful in cases where we only have access to the density matrix. The partial trace of the individual density matrices with respect to the second qubit (taking away the information on “success”, i.e., s , to not be confused with

the initial for *swarm*) provides information on the individual robot's position. Concerning R_1 , after a few passages we obtain:

$$\begin{aligned} Tr_s \rho_1 &= \langle 1_s | \rho_1 | 1_s \rangle + \langle 0_s | \rho_1 | 0_s \rangle \\ &= \frac{1}{2} (|1\rangle\langle 0| + |0\rangle\langle 0|), \end{aligned} \quad (25)$$

while for R_2 we simply have

$$Tr_s \rho_2 = \langle 1_s | \rho_2 | 1_s \rangle + \langle 0_s | \rho_2 | 0_s \rangle = |1\rangle\langle 1|. \quad (26)$$

We notice that R_1 has 1/2 probability of being in 1, while R_2 has probability 1. Thus, the barycenter of the swarm is their mean, 0.75. In this way, we prove that the same information can be recovered from density matrices. Tracing the overall density matrix $\rho_{swarm} = \rho_{12} = \rho_1 \otimes \rho_2$ with respect to the second qubit, we can also recover the position of the swarm barycenter. The same applies to the ideal swarm centered on the target. The comparison between the values for the given swarm and the ideal one, gives a measure of the precision with which the target has been reached. We can define an opportune distance between matrices as the trace of the difference between the reduced matrices, that is $Tr(|\langle 1_s | \rho_{12} | 1_s \rangle| - |\langle 1_s | \rho_{12(T)} | 1_s \rangle|)$, where we focus on the probability of finding the barycenter close to $x = 1$, and where ρ_{12} is the density matrix of the ideal swarm. It is clear that we can think of a whole class of "successful" swarms, whose robots are inside a small neighbor of the target. In addition, using our modeling framework, one may also consider stability problems. Specifically, one may define a Lyapunov as the Hamiltonian of the barycenter of the swarm, which is a function of the robot's position displacements (potential term) and momentum (speed term). Stability will be investigated in future research.

5.2. Mixed states

Consider again the 2-robot swarm described above. As discussed previously, one can significantly benefit in formalizing the robot swarm as a mixed state consisting of robots that are described as pure states. In this case, the density matrix of the swarm would be defined as the weighted sum of the states, rather than as their tensor product.

$$\rho_{swarm} = \frac{1}{2} |\psi_1\rangle\langle\psi_1| + \frac{1}{2} |\psi_2\rangle\langle\psi_2| \quad (27)$$

The information on a single robot can still be recovered via the partial trace, tracing the ρ_{swarm} with respect to the other robot(s). In matrix form, Eq. (27) would give

$$\rho_{swarm} = \begin{pmatrix} 1/2 & 0 & 0 & 0 \\ 0 & 1/4 & 0 & 1/4 \\ 0 & 0 & 0 & 0 \\ 0 & 1/4 & 0 & 1/4 \end{pmatrix}. \quad (28)$$

We verify that $Tr(\rho_{swarm}) = 1$; to check that ρ_{swarm} is a mixed state, we compute the trace of the squared matrix, verifying that it is smaller than 1

$$Tr(\rho_{swarm}^2) = Tr \begin{pmatrix} 1/4 & 0 & 0 & 0 \\ 0 & 1/8 & 0 & 1/8 \\ 0 & 0 & 0 & 0 \\ 0 & 1/8 & 0 & 1/8 \end{pmatrix} = \frac{1}{2}.$$

For the swarm we compute the overall energy E as $E = \sum_{i=1}^N E_i$, with $N = 2$ in our example, and E_i the energy of each single robot. Then, we compute $\Delta E = E(t) - E(t')$, where the time dependence is related with the instantaneous peak position, and we can finally choose the Hamiltonian as

$$H = \frac{\Delta E}{2} \sigma_z = \frac{\Delta E}{2} (|0\rangle\langle 0| - |1\rangle\langle 1|). \quad (29)$$

To reduce the dimensions of the problem, we can perform the partial trace operation on ρ_{swarm} with respect to the *success* information, that we can denote with s as in the previous section. We also perform this operation beforehand on the density matrices of single robots, as

showed in Eqs. (25), (26). The reduced density matrix of the swarm as a mixed state is obtained by tracing on the success s

$$\begin{aligned} \rho_{swarm}^r &= \frac{1}{2} Tr_s(\rho_1) + \frac{1}{2} Tr_s(\rho_2) \\ &= \frac{1}{4} (|1\rangle\langle 0| + |0\rangle\langle 0| + 2|1\rangle\langle 1|). \end{aligned} \quad (30)$$

If we consider the ideal situation where the interaction of the swarm with the environment is absent or negligible, the L operators of the master equation in Lindblad form are negligible as well, and the master equation is reduced to

$$\mathcal{L}(\rho_{swarm}^r) = -i[H, \rho_{swarm}^r] = -i \frac{\Delta E}{2} (|0\rangle\langle 0| - |1\rangle\langle 1|). \quad (31)$$

To consider the interaction with the environment, we have to include non-null L operators. For the sake of simplicity, we can choose $L = |0\rangle\langle 0|$. Thus, the remaining terms of the master equation are computed as

$$\begin{aligned} L^T \rho_{swarm}^r L^* &= \frac{1}{4} |0\rangle\langle 0| \\ -\frac{1}{2} L^\dagger L \rho_{swarm}^r &= -\frac{1}{8} |0\rangle\langle 0| \\ -\frac{1}{2} \rho_{swarm}^r L^\dagger L &= -\frac{1}{8} (|1\rangle\langle 0| + |0\rangle\langle 0|). \end{aligned} \quad (32)$$

Summing up the terms of Eqs. (31), (32), we obtain

$$\mathcal{L}(\rho_{swarm}^r) = -i \frac{\Delta E}{2} (|0\rangle\langle 0| - |1\rangle\langle 1|) - \frac{1}{8} |1\rangle\langle 0|. \quad (33)$$

5.3. Other computational examples

Next, consider another toy swarm, constituted by two robotic fish R_1 , R_2 , moving along the segment $[0, 1]$ of the x axis, as shown in Fig. 2. For the sake of simplicity, we neglect here information about success in target reaching, focusing only on information about position. Case A presents two robots maintaining a constant distance between them, while case B presents two robots getting distant from each other.

Let us focus on case A, an example of swarm compactness over time. Fig. 2 exemplifies the passage to a quantum representation, where the positions are substituted by the peaks of the wavefunctions. As an example, we choose to write the density operator of the swarm as a mixed state:

$$\rho^A(t_0) = \frac{1}{2} |0\rangle\langle 0| + \frac{1}{2} (\sqrt{0.8}|0\rangle + \sqrt{0.2}|1\rangle)(\sqrt{0.8}\langle 0| + \sqrt{0.2}\langle 1|). \quad (34)$$

The swarm at the following time point is represented by the density operator:

$$\rho^A(t_1) = \frac{1}{2} |1\rangle\langle 1| + \frac{1}{2} (\sqrt{0.8}|1\rangle + \sqrt{0.2}|0\rangle)(\sqrt{0.8}\langle 1| + \sqrt{0.2}\langle 0|). \quad (35)$$

The respective matrices, for the swarm at t_0 and at t_1 , are:

$$\rho^A(t_0) = \begin{pmatrix} 0.9 & 0.2 \\ 0.2 & 0.1 \end{pmatrix}, \quad \rho^A(t_1) = \begin{pmatrix} 0.1 & 0.2 \\ 0.2 & 0.9 \end{pmatrix}. \quad (36)$$

To obtain these results, we have to multiply bra-kets, and sum the terms:

$$\begin{aligned} \rho^A(t_0) &= \frac{1}{2} |0\rangle\langle 0| + \frac{1}{2} (\sqrt{0.8}|0\rangle + \sqrt{0.2}|1\rangle)(\sqrt{0.8}\langle 0| + \sqrt{0.2}\langle 1|) \\ &= \frac{1}{2} |0\rangle\langle 0| + \left(\frac{0.8}{2} |0\rangle\langle 0| + \frac{\sqrt{1.6}}{2} |0\rangle\langle 1| + \frac{\sqrt{1.6}}{2} |1\rangle\langle 0| + \frac{0.2}{2} |1\rangle\langle 1| \right) \\ &\quad 0.9|0\rangle\langle 0| + 0.2|0\rangle\langle 1| + 0.2|1\rangle\langle 0| + 0.1|1\rangle\langle 1|, \end{aligned} \quad (37)$$

and then, organizing the coefficients 0.9, 0.2, 0.2, 0.1 inside the matrix with columns $|0\rangle$, $|1\rangle$ and rows $\langle 0|$, $\langle 1|$, we get the result.

Supposing that the variation of energy is positive, we obtain:

$$Tr[\mathcal{L}(\rho_{t_1}^A) \Delta \rho^A] = -0.16i \Delta E < 0. \quad (38)$$

Recalling the definition of ρ -based stability in [43], in the case of positive ΔE , the condition of Eq. (38) denotes stability for the case A. In

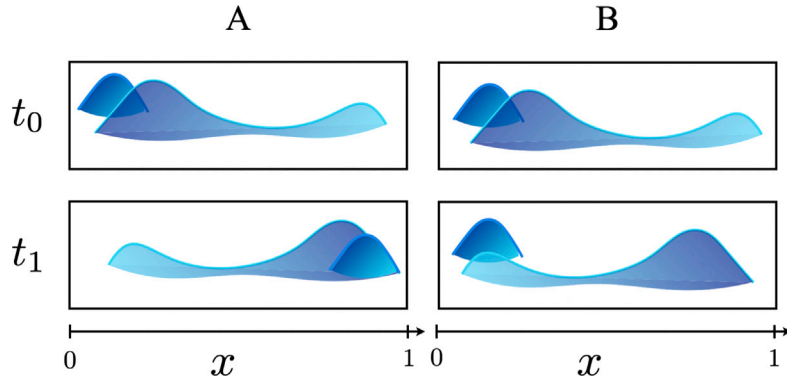


Fig. 2. Quantum-like representation of a toy 2-robot swarm, where the position of the robot is considered as the weighted peak of the corresponding wavefunction.

this article, we chose to just present a general approach to a quantum swarm, but this formalism can be helpful for future technological developments of the research.

We conclude the section of Examples with a case of target-reaching application. Let us start from the previously-found swarm density matrix, that we will simply indicate with ρ_{swarm}

$$\rho_{swarm} = \begin{pmatrix} 0.9 & 0.2 \\ 0.2 & 0.1 \end{pmatrix}, \quad (39)$$

and let us consider a target in $|0\rangle\langle 0|$, as an ideal swarm centered on the target, described by a state ρ_T having density matrix

$$\rho_T = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}. \quad (40)$$

The distance between the swarm and the target can be computed with the trace distance definition:

$$d(\rho_{swarm}, \rho_T) = \frac{1}{2} \text{Tr}[\sqrt{(\rho_{swarm} - \rho_T)(\rho_{swarm} + \rho_T)}] = 0.05. \quad (41)$$

The adoption of density matrices for swarms of quantum robots can pave the way toward applications in control theory, adapting to our cases criteria of stability for open quantum systems, that are based on density matrices and suitable choices of Lyapunov functions [43].

5.4. Movements on a 2D surface, first examples with Qiskit, and larger swarms

First, we propose a working example to visualize a toy swarm of two robots as surfaces representing the wavefunctions, neglecting the target, and instead focusing only on the swarm. The process describes a global-local approach, wherein density matrices of the swarm and a time evolution operator are computed to determine the evolution of the swarm. Then, local distribution of the individual robots are updated accordingly. Such a process, as opposed to the local-to-global approach proposed in [35], allows to streamline local information of the robots from global information on the swarm. Suppose that the initial state of particle 1 at time t_0 is

$$|\psi_1(t_0)\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle), \quad (42)$$

and the initial state of particle 2 is

$$|\psi_2(t_0)\rangle = |00\rangle \quad (43)$$

Thus, the density matrix for the first particle-robot is

$$\rho_1(t_0) = \frac{1}{2}(|00\rangle\langle 00| + |11\rangle\langle 11|) = \frac{1}{2} \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}, \quad (44)$$

and the density matrix for the second particle-robot is

$$\rho_2(t_0) = |00\rangle\langle 00| = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}. \quad (45)$$

From this, we obtain the density matrix of the swarm as a mixed density matrix. To highlight this, here we use the pedex *mix*:

$$\rho_{mix}(t_0) = \frac{\rho_1(t_0) + \rho_2(t_0)}{2} = \begin{pmatrix} 0.75 & 0. & 0. & 0.25 \\ 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. \\ 0.25 & 0. & 0. & 0.25 \end{pmatrix} \quad (46)$$

We choose the final state of particle 1 at t_1 as

$$|\psi_1(t_1)\rangle = \frac{\sqrt{0.2}}{\sqrt{0.8^2 + 0.2^2}}|00\rangle + \frac{\sqrt{0.8}}{\sqrt{0.8^2 + 0.2^2}}|11\rangle, \quad (47)$$

and that of particle 2 as

$$|\psi_2(t_1)\rangle = \frac{\sqrt{0.1}}{\sqrt{0.9^2 + 0.1^2}}|00\rangle + \frac{\sqrt{0.9}}{\sqrt{0.9^2 + 0.1^2}}|11\rangle. \quad (48)$$

Finally, the density matrix of the mixed state is given by

$$\rho_{mix}(t_1) = \begin{pmatrix} 0.15 & 0. & 0. & 0.35 \\ 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. \\ 0.35 & 0. & 0. & 0.85 \end{pmatrix}. \quad (49)$$

Fig. 3 shows the robots corresponding to the two “particles” as wavefunctions, whose height along the z -axis denotes the probability amplitude. Such a representation is an additional overcoming of the visualizations of [35], where a classic position was considered in correspondence to the peak of the wavefunctions.

The computation of density matrices can be automatized by using Qiskit with the help of loading functions *Statevector* and *DensityMatrix*. The matrices and the visualizations in Fig. 3 are obtained via the implementation in Python (code provided in Code Availability) of Algorithm 1. For our implementation, we chose a Gaussian appearance for the wavefunctions. The code yields Fig. 3, and the robot states as well as density matrices for each robot as well as for the whole swarm at t_0 obtained as:

$$|\psi_1(t_0)\rangle = [0.707, 0, 0, 0.707], \quad |\psi_2(t_0)\rangle = [1, 0, 0, 0] \quad (50)$$

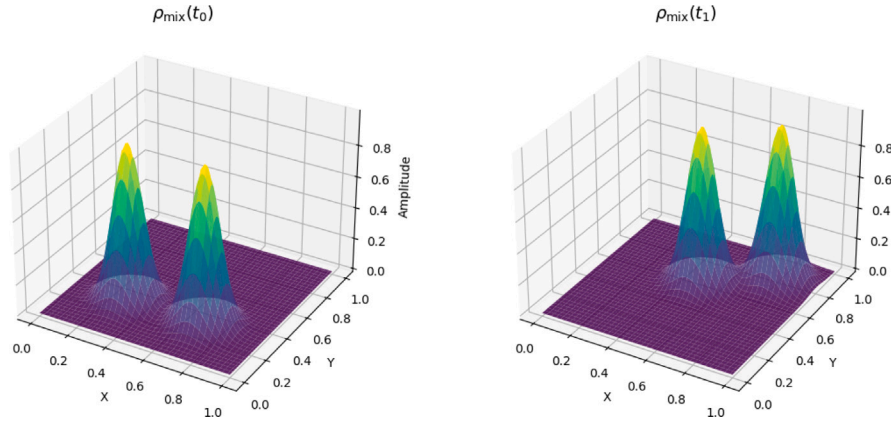


Fig. 3. Wavefunction visualization of the probability amplitudes to find the two robots-particles described in Eqs. (46) and (49), respectively.

Algorithm 1 computation of density matrices

Input: N robots with states $\{\psi_i\}_{i=1}^N$, N of robots, optional: target position as $|\psi_T\rangle$, maximum number of time steps n to reach the target

Output: swarm density matrix at each time step t ; visualizations

Procedure: initialize robot states $\psi_i(t_0)$

if target is present **then**

 compute density matrix of the target as $\rho_T = |\psi_T\rangle\langle\psi_T|$

end if

for each time step **do**

 update individual robot states according to swarm interaction rules

 compute density matrices for the individual robots as $\rho_i(t) = |\psi_i(t)\rangle\langle\psi_i(t)|$

 compute swarm density matrix $\rho_{swarm}(t)$ as $\frac{1}{N} \sum_i \rho_i(t)$

if target is present **then**

 compute trace distance between the matrix of the swarm and the matrix of the target as $tr|\rho_{swarm}(t) - \rho_T(t)|$

end if

end for

$$\rho_1(t_0) = \begin{pmatrix} 0.5 & 0 & 0 & 0.5 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0.5 & 0 & 0 & 0.5 \end{pmatrix}, \quad \rho_2(t_0) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad (51)$$

$$\rho_{swarm}(t_0) = \begin{pmatrix} 0.75 & 0 & 0 & 0.25 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0.25 & 0 & 0 & 0.25 \end{pmatrix}$$

while the final states and matrices for time t_1 is obtained as:

$$|\psi_1(t_1)\rangle = [0.447, 0, 0, 0.894] \quad |\psi_2(t_1)\rangle = [0.316, 0, 0, 0.949] \quad (52)$$

$$\rho_1(t_1) = \begin{pmatrix} 0.2 & 0 & 0 & 0.4 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0.4 & 0 & 0 & 0.8 \end{pmatrix}, \quad \rho_2(t_1) = \begin{pmatrix} 0.1 & 0 & 0 & 0.3 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0.3 & 0 & 0 & 0.9 \end{pmatrix}, \quad (53)$$

$$\rho_{swarm}(t_1) = \begin{pmatrix} 0.15 & 0 & 0 & 0.35 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0.35 & 0 & 0 & 0.85 \end{pmatrix}$$

Now, we include visualizations for larger swarms. Here, we include a fixed target, and assume that the robot dynamics are such that they advance toward the target over time. This results in 3D representations

of the swarm, with success of reaching the target included as in (11). Visualizations are fixed for a maximum number of time steps. In Fig. 4, we show an example of a swarm of $N = 10$ robots, whose initial states are defined by Eq. (54):

$$\begin{aligned} |\psi_1\rangle &= \left[\frac{1}{\sqrt{2}}, 0, 0, 1/\sqrt{2} \right], & |\psi_2\rangle &= [1, 0, 0, 0], \\ |\psi_3\rangle &= [0, 0, -1, 0], & |\psi_4\rangle &= [0, 1, 0, 0], \\ |\psi_5\rangle &= \left[0, \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}, 0 \right], & |\psi_6\rangle &= \left[\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, 0, \frac{1}{\sqrt{3}} \right], \\ |\psi_7\rangle &= \left[-\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, 0, \frac{1}{\sqrt{3}} \right], & |\psi_8\rangle &= \left[-\frac{1}{\sqrt{3}}, 0, 0, \frac{1}{\sqrt{3}} \right], \\ |\psi_9\rangle &= \left[-\frac{1}{\sqrt{2}}, 0, 0, -\frac{1}{\sqrt{2}} \right], & |\psi_{10}\rangle &= \left[0, 0, \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}} \right]. \end{aligned} \quad (54)$$

With the help of Algorithm 1, we obtain the density matrix of the swarm at the initial time t_0 as:

$$\rho_{swarm,10}(t_0) = \begin{pmatrix} 0.3 & 0.067 & 0 & 0.067 \\ 0.067 & 0.217 & 0.05 & 0 \\ 0 & 0.05 & 0.2 & 0.05 \\ 0.067 & 0 & 0.05 & 0.25 \end{pmatrix}. \quad (55)$$

Then, we can define a fixed target that does not change over time with density matrix $\rho_T = |11\rangle\langle 11|$, from $\psi_T = [0, 0, 0, 1]$. Thus, the trace distance between the density matrix $\rho_{swarm,10}(t_i)$ of our 10-robot swarm and $\rho_T(t_i)$ is 0.7402. At this point, for the sake of simplicity, we consider a progressive movement of robots toward the target, for a given number of steps. Thus, we can compute the density matrix of the swarm at each time step, and we add the visualization of the classical peaks within an arena, as the 2D representation adopted in [35].

After 10 steps, we get the following visualizations and swarm density matrix as:

$$\rho_{swarm,10}(t_1) = \begin{pmatrix} 0.241 & 0.032 & 0 & 0.074 \\ 0.032 & 0.147 & 0.038 & 0.072 \\ 0 & 0.038 & 0.137 & 0.029 \\ 0.074 & 0.072 & 0.029 & 0.475 \end{pmatrix}. \quad (56)$$

At this step, when the swarm reaches the target, the trace distance between the density matrix of the swarm and the target is 0.5399. Fig. 5 shows the final appearance of the swarm. The initial positions of the wavefunctions peaks can also be randomized. Having applied these functions, we provide here an example of visualization with 30, 100, and 1000 robots, also showing their respective density matrices,

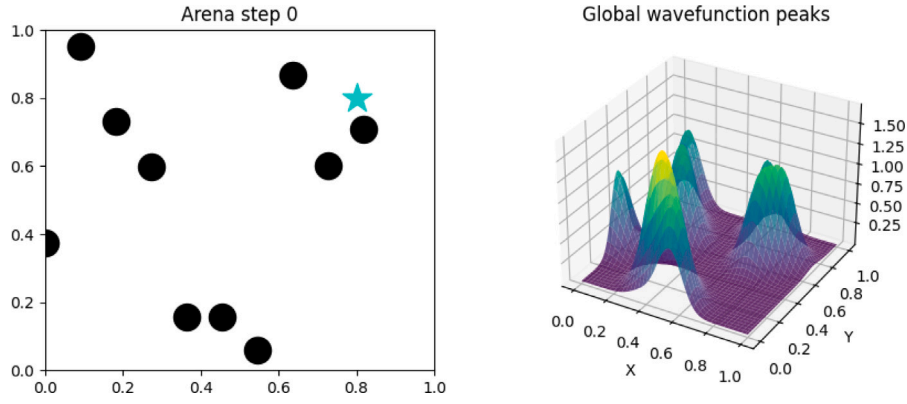


Fig. 4. Visualization of the 10-robot swarm at the beginning of our simulation. The turquoise star indicates the target, adopting the style of 2D representation of [35].

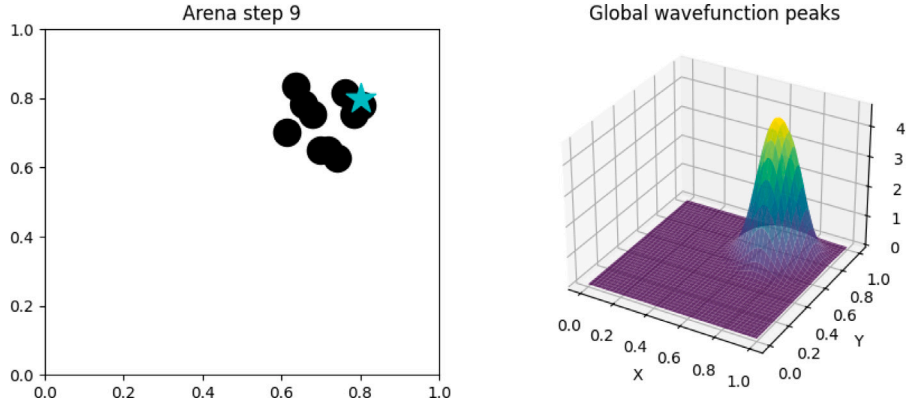


Fig. 5. Visualization of the 10-robot swarm once it reached the target.

see Fig. 6. The matrices are given as:

$$\begin{aligned}
 &\rho_{\text{swarm},30} \\
 &= \begin{pmatrix} 0.186 & 0.133 + 0.093i & 0.143 + 0.034i & 0.156 + 0.087i \\ 0.133 - 0.093i & 0.276 & 0.191 - 0.076i & 0.239 + 0.002i \\ 0.143 - 0.034i & 0.191 + 0.076i & 0.246 & 0.179 + 0.053i \\ 0.156 - 0.087i & 0.239 - 0.002i & 0.179 - 0.053i & 0.291 \end{pmatrix} \\
 &\rho_{\text{swarm},100} \\
 &= \begin{pmatrix} 0.237 & 0.174 + 0.009i & 0.153 - 0.007i & 0.183 \\ 0.174 - 0.009i & 0.258 & 0.179 - 0.022i & 0.207 - 0.007i \\ 0.153 + 0.007i & 0.179 + 0.022i & 0.227 & 0.188 + 0.014i \\ 0.183 & 0.207 + 0.007i & 0.188 - 0.014i & 0.278 \end{pmatrix} \\
 &\rho_{\text{swarm},1000} \\
 &= \begin{pmatrix} 0.251 & 0.183 - 0.009i & 0.180 - 0.001i & 0.182 + 0.002i \\ 0.183 + 0.009i & 0.253 & 0.180 + 0.001i & 0.182 + 0.008i \\ 0.180 + 0.001i & 0.180 - 0.001i & 0.246 & 0.179 + 0.008i \\ 0.182 - 0.002i & 0.182 - 0.008i & 0.179 - 0.008i & 0.251 \end{pmatrix}.
 \end{aligned} \tag{57}$$

One can notice that there are complex terms the off-diagonal terms of the density matrices: these are the coherences, and they indicate *entanglement* in the swarm. We can interpret them as the emerging of quantum effects from our definition of the swarm. However, a detailed investigation of this point is out of the scope of this article.

As it can be noticed from Eq. (57), the size of the density matrices is not changed by the increasing of the size of the swarms, allowing for a more compact and less computationally challenging description of the swarm. In fact, in our representation, the size of the matrix

describing the swarm depends only upon the degrees of freedom of its components, and their proximity to the target.

5.5. Time evolution

We are also interested in finding a method to approximate the time evolution operator, that is, a unitary matrix $O(t)$ such that:

$$O(t_1)\rho_{\text{swarm}}(t_0) = \rho_{\text{swarm}}(t_1). \tag{58}$$

A possible technique is the singular value decomposition (SVD). We first compute the SVD of $\rho_{\text{swarm}}(t_0)$ obtaining two matrices U and V^\dagger and then the SVD of $\rho_{\text{swarm}}(t_1)$ obtaining two other matrices U' and V'^\dagger , such as:

$$\rho_{\text{swarm}}(t_0) = USV^\dagger, \quad \rho_{\text{swarm}}(t_1) = U'S'V'^\dagger, \tag{59}$$

where S (respectively S') is a diagonal matrix with the singular values of $\rho_{\text{swarm}}(t_0)$ (respectively $\rho_{\text{swarm}}(t_1)$). These new matrices can be used to reconstruct the process from $\rho_{\text{swarm}}(t_0)$ to $\rho_{\text{swarm}}(t_1)$. In fact, the needed time-evolution operator is:

$$O = U'(V'^\dagger)^{-1}VU^\dagger. \tag{60}$$

The choice of the dynamics is fundamental to shape the global motion of a swarm. Here, the robots exchange information about individual target proximity, and then the overall swarm reaches a point in space, e.g., toward a search and rescue scenario where the robot swarm needs to identify and reach targets for performing rescue operations. The point in space is represented as an ideal swarm centered on it. In our case, it can be the target density matrix. The approximation of the time operator represented the required overall transformations to let the swarm approximately reach that region. We can also apply to $\rho_{\text{swarm}}(t_0)$ some given specific dynamics, and observe how the swarm is moving

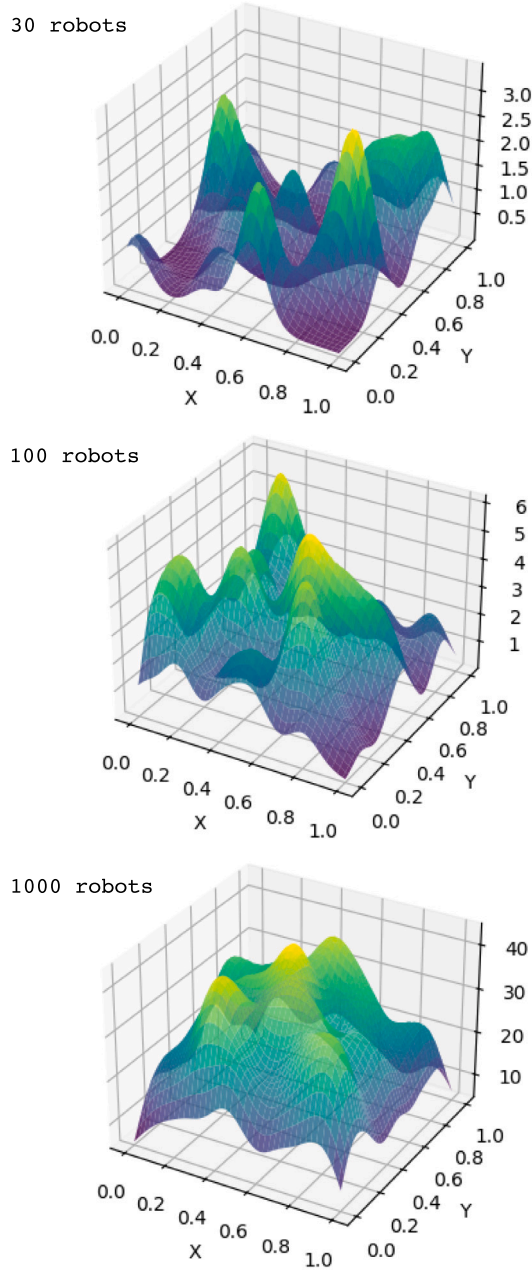


Fig. 6. Visualizations and density matrices of swarms of different sizes ($N = 30, 100, 1000$ robots).

accordingly. A classic example is $U = e^{-iHt}$. The systematic definition of density matrices for a swarm, and a set of subsequent computations of the dynamics between pairs of time points, can allow one to approximate the functions inside the dynamics. Such an approach can foster precise applications in the domain of global-to-local control of the swarms.

In particular, we can define the algorithm illustrated in Fig. 7 and in Algorithm 2. Our formalism can thus give some theoretical prediction on target reaching that can be experimentally verified. Let us describe the idea of Fig. 7. We start with the local dimension, defining an initial distribution of robots in terms of $|\psi_i\rangle$. Then we compute their density matrix distribution, thus going into the global dimension. We also need a target position defined as an estimated probability density, denoted ρ_T^e . This information will be progressively refined. At this point we can approximate a dynamics U such as $U\rho_{swarm} \sim \rho_T^e$. We can update the swarm distribution in terms of density matrix. And we also

have to update the distribution of robots at the local dimension. Since more individual configuration correspond to the same density matrix distribution, the robots can be shifted with the smallest displacements such that their ψ_i approximate the new ρ_{swarm} . The robots activate their sensors, and detect their individual proximity to the target. This information allow us to update the target distribution ρ_T^e , which allows for a new dynamics computation. The cycle continues until the swarm reaches the target with a great precision. This method, inspired by optimization techniques, can help yield prediction on the number of passages for target reaching. In the algorithm, we adopt the notation ρ_i, ρ_f to denote the density matrix at the beginning of arena exploration and at the end of it, respectively. We do not use here t_0 and t_1 because we prefer to not make hypothesis here on the number of time points required to reach the target.

Algorithm 2 Distributed swarm algorithm.

Require: N robots with states $\{|\psi_i\rangle\}_{i=1}^N$, expected target density ρ_T , arena target position T

- 1: **Initialize:** robots' positions, individual quantum states
- 2: **repeat**
- 3: **(A) Local \rightarrow Global:** each robot i sends its local classical position p_i and its quantum state $\rho_i = |\psi_i\rangle\langle\psi_i|$ to the global node
- 4: **(B) Global aggregation:** compute swarm density matrix ρ_{swarm}
- 5: **(B) Global dynamics estimation:** use ρ_{swarm} and the estimated target cloud to compute an approximate dynamics U driving $\rho_{swarm} \rightarrow \rho_T$
- 6: **(B) Send dynamics to individuals:** determine per-robot quantum updates and transmit them
- 7: **(C) Classical motion plan (global \rightarrow local):** computation of the global desired swarm spatial distribution approximating the estimated target location; instructions sent to the individuals
- 8: **for** each robot i (local) **do**
- 9: Robot i activates sensors to estimate target-proximity
- 10: Robot i : classical motion, quantum-update received:
- $\psi_i \leftarrow \text{normalize}(\psi_i + \gamma_i(\psi_T - \psi_i))$
- γ_i is computed globally
- 11: **end for**
- 12: **(A) Repeat:**
- 13: **until** trace_distance(ρ_{swarm}, ρ_T) below tolerance **or** max iterations reached

5.6. Distributed quantum swarm

We end this Section with an example of possible implementation of Algorithm 2, without an explicit approximation of the time-evolution operator, but including the effects of the dynamics on the robotic movements. We consider the same swarm and scenario as seen in Section 5.4. We implement here the A/B/C loop, where A indicates the dimension of local robots, B the overall swarm, and C the target, and each of them is described by density matrices (or approximate density matrices, as for the target). For the sake of simplicity, we consider here an omniscient scenario, as in [35], and an exchange between local and global dimensions of the swarm [44]. The implementation contains the following steps:

- robots send positions and local ρ to the global node;
- global dimension: computation of aggregates ρ_{swarm} (with per-robot mixing factors) and approximation of the dynamics;
- global dimension: sends these mixing factors and a desired spatial distribution (target cloud) to the individual robots;
- robots move according to their local sensors to match the desired distribution and apply quantum updates;
- Finally, as output, each robot returns the states of the final robots and the final ρ_{swarm} .

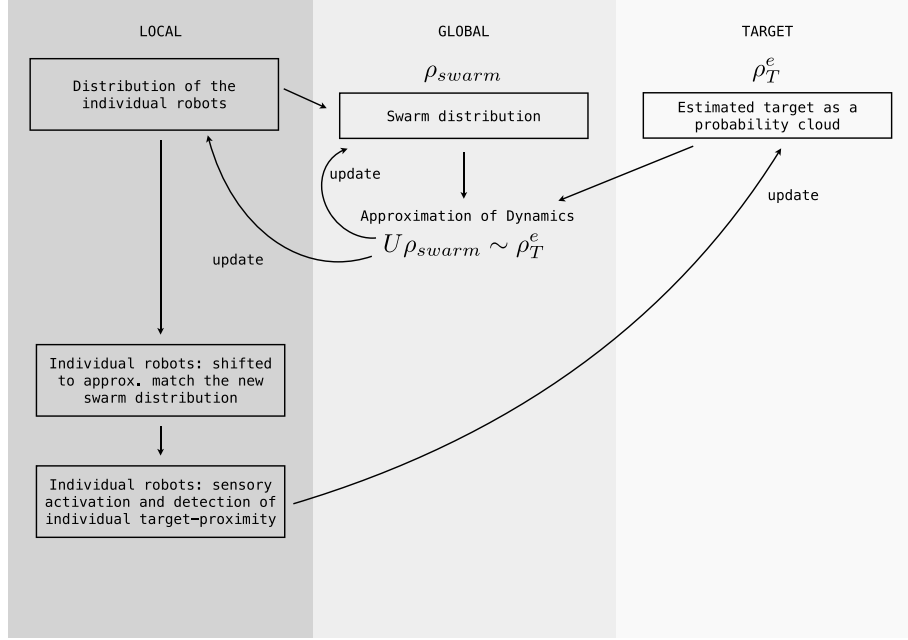


Fig. 7. An algorithm for swarm-robotic target reaching based on the defined matrices, and including an interaction between local and global.

Here, the ‘approximate dynamics’ U is implemented as a small convex mixing of each local ψ toward ψ_T of the target, with individual mixing coefficient obtained from fidelity $F(\rho, \sigma) = \left(\text{tr} \sqrt{\sqrt{\rho} \sigma \sqrt{\rho}} \right)^2$, where ρ, σ are the density matrices of two quantum states, but it can also be done with trace distance. We implement the discussed time evolution in three different ways:

- via a *Classical–Quantum Hybrid Iteration*, constituted by a dissipative quantum relaxation [45] and a consensus-type movement rule [46] for the robots. Each robot of the swarm has a local, pure quantum state ψ_i and moves classically toward the target through the space. The global ρ_{swarm} is updated iteratively, according to local information and a small, quantum-inspired feedback term (a Lindblad-like dissipative operator [45]), according to the update rule:

$$\rho_{\text{swarm}}(t+1) = \rho_{\text{swarm}}(t) + \alpha(\rho_T - \rho_{\text{swarm}}(t)) + \mathcal{L}_{\text{noise}}[\rho_{\text{swarm}}(t)], \quad (61)$$

$$(x_i, y_i)_{t+1} = (x_i, y_i)_t + \beta \sum_j w_{ij} ((x_j, y_j)_t - (x_i, y_i)_t); \quad (62)$$

- via a *Completely Positive Trace-Preserving Map (CPTP) Update* [39, 47], a quantum map that replaces the additive update, according to the update rule:

$$\rho_{\text{swarm}}(t+1) = (1 - \eta) \rho_{\text{swarm}}(t) + \eta \rho_T, \quad (63)$$

$$(x_i, y_i)_{t+1} = (1 - \beta) (x_i, y_i)_t + \beta (x_T, y_T), \quad (64)$$

- finally, with the *Pure CPTP Global Update* [47–49], with a direct projection toward the target state. For a specific choice of parameters, one obtains an instantaneous projection (and a trivial convergence). This is a special case of the CPTP update in Eqs. (63) and (64), obtained for $\eta = \beta = 1$:

$$\rho_{\text{swarm}}(t+1) = \rho_T, \quad (x_i, y_i)_{t+1} = (x_T, y_T).$$

Finally, we define control parameters such as the number of iterations and the tolerance.

Table 1

Comparison of the convergence speed as obtained distance between ρ_{swarm} and ρ_T for some variants of Algorithm 2, for the initial distance $\rho_{\text{swarm}}, \rho_T = 0.740199$.

	5 iterations	20 iterations	30 iterations
Quantum–classical hybrid	0.54	0.24	0.18
CPTP only	0.47	0.40	0.40
Pure CPTP Global Update	0.0072	0.0000031	0.00000002

Let us start with the first method. And the robotic sensors are here simplified as a measure of individual proximity to the target. The mixing coefficient is larger when fidelity is low or the global distance is high. For the overall spatial distribution, we consider a Gaussian cloud centered at target, with a σ depending on the global distance. At this point, the updates are sent to the individuals. They concern the mixing factors and the cloud parameters. The individual robots receive them and then move on the arena accordingly. The overall procedure is repeated until the target is reached, or when a maximum number of iteration is attained.

For our implementation, we consider the previously-defined ten robots. The initial trace distance of 0.74 decreases to 0.24 after 20 iterations, 0.18 after 30 iterations, and, after only 5 iterations, of 0.54. Apparently, this method may appear slow. However, one should take into account that no quantum-implemented logic gate has been used, avoiding the time of its implementation on a quantum circuit, as it was done in [35], and there was no need for a signal transmission between the individual robots and a computational central.

We can then implement the two other techniques. As aforementioned, the convergence of the second one, the pure CPTP, is trivial: it only corresponds to equating the density matrix of the swarm to the one of the target. For smaller values of α , it is visible a progressive approach of the swarm. With the initial distance of 0.74019942, after one iteration we get 0.44, and after two, we get 0.05. We show the results of the comparison between the three techniques in Table 1.

Quantum Swarm in the Real World. In a concrete scenario, nodes move inside a defined 2D Geographical Area (GA), whose dimensions are X and Y . At the application layer, we can imagine that robots can exchange few coordination packets:

- Status and health messages: generally used to maintain awareness of each robot condition (status update, error alert, low battery, etc.);
- Localization and mapping messages: generally used for exploration (position update, map fragment, landmark detection, etc.);
- Task coordination messages: used to distribute and synchronize tasks among robots (task announce, task claim, task status, task complete, etc.);
- Motion management and collision avoidance: for ensuring safe movements and swarm keeping (move willing, obstacle detection, formation update, etc.);
- General information sharing: used for distributed consensus or decision-making;
- roles management: used to assign specialized roles to different nodes (resource available, role assignment, load transfer);
- Global coordination: generally used for mission flow control (mission start, mission pause, mission complete, global alert).

The description above is referred to just one of the possible scenarios. Our proposed approach is completely general, so the set of provided messages can be stretched or extended.

We believe that the physical layer can be contextualized in function of the considered environment. In particular, RF waves can be used in free space, acoustic waves can be considered for underwater missions, etc. In particular, a swarm primarily utilizes wireless communication for inter-robot data transmission. Then, depending on mission requirements and deployment environments, one or more of the following technologies may be employed:

- Wi-Fi (IEEE 802.11 protocols family): it provides high data rates suitable for rich information exchange such as maps, images, or sensor data. Common in indoor or short-range outdoor environments;
- ZigBee (IEEE 802.15.4 protocols family): generally used for low-power, low-bandwidth communication where energy efficiency and robustness are prioritized over data throughput. It is suitable for large-scale swarms with lightweight message exchanges;
- Bluetooth: it is more appropriate for small teams or close-proximity coordination with minimal power consumption;
- LoRa or Sub-GHz Links (optional in long-range missions): these technologies are used when the swarm must cover large outdoor areas with sparse connectivity and low data rate requirements.

6. Conclusions and discussions

In this article, we proposed a new modeling methodology for a quantum robotic swarm using density matrices to describe swarms of micro and nano robots under quantum effects in a compact yet precise manner. We also proposed some computational examples illustrating applications of our modeling formalism. The proposed framework is illustrated in both small and large robotic swarms, ranging from swarms of 2 robots up to 1000 robots. As the number of the robots increase, the shape of the cloud at each time instant is different. However, we show that the size of the obtained density matrices remain constant, dependent only on the degrees of freedom of the robots. This presents a significant computational advantage over previous approaches such as in [35]. In addition, we have proposed algorithms and implementations for target-reaching, and analyzed the obtained results. For toy robotic swarms with 2 or 3 robots, the computations are straightforward and thus performed manually. For more complex computations, such as larger swarms and density matrix computations, the open-source software Qiskit¹ is used. Finally, a brief discussion on telecommunication tools and features, as a further step toward a more realistic and real-life implementation of our model is presented.

In the future, one may consider the application of our approach to describe the motion of single robots as well as the center of mass of the quantum swarm. Experiments on physical robots may also help investigate the limits of our approach, especially concerning the necessary calibration of parameters representing divergence and convergence, which also depend upon the size of the swarm, the amplitude of movements of their components. Future research can focus on the matrix definition of multiple swarms and hierarchical approaches. In fact, we can define the density matrix of a system with multiple swarms as the tensor product of the density matrices of each swarm, or as the density matrix of the whole swarm seen as a mixed state of individual swarms. The information on the single swarm can vice versa be recovered performing the operation of partial trace with respect to all the other swarms, considered as “environment”, to the overall density matrix.

Further studies can exploit the density-matrix based formalism to develop control approaches for quantum robots [43]. In fact, our formalism with density matrices for robotics swarms can pave the way toward future applications of stability criteria to the robotics swarms, adapting control criteria defined for open quantum systems to the robots [43]. Moreover, one can also address the matrix depiction of multi-layer systems, recently used to characterize hierarchical systems of swarms communicating with mobile computing centrals [50], whose information-sharing systems are shaped according to the protocols of telecommunications [51].

CRedit authorship contribution statement

Maria Mannoni: Writing – review & editing, Writing – original draft, Visualization, Software, Methodology, Investigation, Formal analysis, Conceptualization. **Mahathi Anand:** Writing – review & editing, Validation. **Peppino Fazio:** Writing – review & editing, Writing – original draft, Validation, Supervision, Conceptualization. **Abdalla Swikir:** Writing – review & editing, Validation, Supervision, Project administration, Conceptualization.

Code availability

The codes in Python to implement the proposed examples can be accessed at https://github.com/medusamedusa/density_matrix_for_swarms.

Funding

The work by M. A. is funded by the German Research Foundation (DFG, Deutsche Forschungsgemeinschaft) as part of Germany’s Excellence Strategy – EXC 2050/1 – Project ID 390696704 – Cluster of Excellence “Centre for Tactile Internet with Human-in-the-Loop” (CeTI) of Technische Universität Dresden.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The link to the GitHub folder is included at the end of the article.

¹ <https://www.ibm.com/quantum/qiskit>.

References

- [1] E. Debie, K. Kasmarik, M. Garratt, Swarm robotics: A survey from a multi-tasking perspective, *ACM Comput. Surv.* 56 (2) (2023) <http://dx.doi.org/10.1145/3611652>.
- [2] A.K. Kar, Bio inspired computing – A review of algorithms and scope of applications, *Expert Syst. Appl.* 59 (2016) 20–32, <http://dx.doi.org/10.1016/j.eswa.2016.04.018>.
- [3] B. Sun, D. Zhu, C. Tian, C. Luo, Complete coverage autonomous underwater vehicles path planning based on gladius bio-inspired neural network algorithm for discrete and centralized programming, *IEEE Trans. Cogn. Dev. Syst.* 11 (1) (2019) 73–84, <http://dx.doi.org/10.1109/TCDS.2018.2810235>.
- [4] P. Ngamkajornwiwat, J. Homchanthanakul, P. Teerakittikul, P. Manoonpong, Bio-inspired adaptive locomotion control system for online adaptation of a walking robot on complex terrains, *IEEE Access* 11 (2020) 91587–91602, <http://dx.doi.org/10.1109/ACCESS.2020.2992794>.
- [5] M. Luo, X. Hou, S.X. Yang, A multi-scale map method based on bioinspired neural network algorithm for robot path planning, *IEEE Access* 7 (2019) 11, <http://dx.doi.org/10.1109/ACCESS.2019.2943009>.
- [6] R. Oung, R. D'Andrea, The distributed flight array, *Mechatronics* 21 (2011) 908–917.
- [7] M. Schranz, M. Umlauf, M. Sende, W. Elmenreich, Swarm robotic behaviors and current applications, *Front. Robot. AI* 7 (2020).
- [8] H. Hamann, *Swarm Robotics: A Formal Approach*, Springer International Publishing, 2018, <http://dx.doi.org/10.1007/978-3-319-74528-2>.
- [9] M. Duarte, V. Costa, J. Gomes, T. Rodrigues, F. Silva, S.M. Oliveira, A.L. Christensen, Evolution of collective behaviors for a real swarm of aquatic surface robots, *PLOS ONE* 11 (3) (2016) e0151834, <http://dx.doi.org/10.1371/journal.pone.0151834>.
- [10] X. Dong, M. Sitti, Controlling two-dimensional collective formation and cooperative behavior of magnetic microrobot swarms, *Int. J. Robot. Res.* 39 (2020).
- [11] S. Fusco, F. Ullrich, J. Pokki, G. Chatzipirpiridis, B. Özkale, K.M. Sivaraman, O. Ergeneman, S. Pané, B.J. Nelson, *Microrobots: A new era in ocular drug delivery*, *Expert. Opin. Drug Deliv.* (2014).
- [12] C.K. Schmidt, M. Medina-Sánchez, R.J. Edmondson, O.G. Schmidt, Engineering microrobots for targeted cancer therapies from a medical perspective, *Nat. Commun.* 11 (1) (2020) 5618, <http://dx.doi.org/10.1038/s41467-020-19322-7>.
- [13] L. Ferranti, F. Cuomo, Nano-wireless communications for microrobotics: An algorithm to connect networks of microrobots, *Nano Commun. Netw.* 12 (2017) 53–62, <http://dx.doi.org/10.1016/j.nancom.2017.01.007>.
- [14] L. Ferranti, F. Cuomo, Nano-wireless communication for microrobotics: bridging the gap, in: *Proceedings of the Second Annual International Conference on Nanoscale Computing and Communication*, 2015, pp. 1–6.
- [15] F. Büther, F.-L. Lau, M. Stelzner, S. Ebers, A formal definition for nanorobots and nanonetworks, in: O. Galinina, S. Andreev, S. Balandin, Y. Koucheryavy (Eds.), *Internet of Things, Smart Spaces, and Next Generation Networks and Systems*, Springer International Publishing, Cham, 2017, pp. 214–226, http://dx.doi.org/10.1007/978-3-319-67380-6_20.
- [16] J. Stolze, D.Q.C.A. Suter, *Short Course from Theory to Experiment*, Wiley, 2004.
- [17] M. Fazilat, N. Zioui, J. St-Arnaud, A novel quantum model of forward kinematics based on quaternion/Pauli gate equivalence: Application to a six-jointed industrial robotic arm, *Results Eng.* 14 (2022) 100402, <http://dx.doi.org/10.1016/j.rineng.2022.100402>, URL: <https://www.sciencedirect.com/science/article/pii/S259012302200072X>.
- [18] T. Otani, A. Takamishi, N. Hara, Y. Takita, K. Kimura, Quantum computation for robot posture optimization, *Sci. Rep.* 15 (1) (2025) 28508, <http://dx.doi.org/10.1038/s41598-025-12109-0>, URL: <https://www.nature.com/articles/s41598-025-12109-0>, Publisher: Nature Publishing Group.
- [19] B.N. De Carolis, C. Loglisci, M.G. Miccoli, G. Palestra, S. Violante, Quantum-enhanced social robotics: The QUADRI project, in: *Adjunct Proceedings of the 33rd ACM Conference on User Modeling, Adaptation and Personalization*, in: UMAP Adjunct '25, Association for Computing Machinery, New York, NY, USA, 2025, pp. 151–155, <http://dx.doi.org/10.1145/3708319.3735545>.
- [20] M. Mannone, A. Chella, G. Pilato, V. Seidita, F. Vella, S. Gaglio, Modeling robotic thinking and creativity: A classic-quantum dialogue, *Math.* 12 (5) (2024) 642, <http://dx.doi.org/10.3390/math12050642>.
- [21] D. Lanza, P. Solinas, F. Mastrogianni, A preliminary study for a quantum-like robot perception model, 2020, URL: <https://arxiv.org/abs/2006.02771>. arXiv: 2006.02771.
- [22] D. Lanza, P. Solinas, F. Mastrogianni, Multi-sensory integration in a quantum-like robot perception model, in: *Experimental Robotics*, Springer International Publishing, 2021, pp. 502–509, http://dx.doi.org/10.1007/978-3-030-71151-1_44.
- [23] A. Wichert, *Principles of Quantum Artificial Intelligence*, World Scientific, 2020.
- [24] D. Dong, C. Chen, H. Li, T. Tarn, *IEEE transactions on systems man and cybernetics part B (cybernetics)*, *Quantum Reinf. Learn.* 38 (2008).
- [25] Y. Kwak, W. Yun, S. Jung, J. Kim, J. Kim, Introduction to Quantum Reinforcement Learning: Theory and PennyLane-based Implementation, *International Conference On Information And Communication Technology Convergence (ICTC)*, 2021.
- [26] P. Lathrop, B.L. Boardman, S. Martínez, Quantum search approaches to sampling-based motion planning, *IEEE Access* 11 (2023) 89506–89519, URL: <https://api.semanticscholar.org/CorpusID:258107979>.
- [27] P. Lathrop, B. Boardman, S. Martínez, Parallel quantum rapidly-exploring random trees, *IEEE Access* 12 (2024) 47173–47189, <http://dx.doi.org/10.1109/ACCESS.2024.3383313>, arXiv:2310.15303.
- [28] A. Chella, S. Gaglio, M. Mannone, G. Pilato, V. Seidita, F. Vella, S. Zammuto, Quantum planning for swarm robotics, *Robot. Auton. Syst.* 161 (C) (2023) <http://dx.doi.org/10.1016/j.robot.2023.104362>.
- [29] Q. Qian, J. Wu, Z. Wang, Optimal path planning for two-wheeled self-balancing vehicle pendulum robot based on quantum-behaved particle swarm optimization algorithm, *Pers. Ubiquitous Comput.* 23 (3) (2019) 393–403, <http://dx.doi.org/10.1007/s00779-019-01216-1>.
- [30] A. Koukam, A. Abbas-Turki, V. Hilaire, Y. Ruichek, Towards a quantum modeling approach to reactive agents, in: *2021 IEEE International Conference on Quantum Computing and Engineering, QCE*, 2021.
- [31] K. Zhu, M. Jiang, Quantum artificial fish swarm algorithm, in: *Proceedings of the 8th World Congress on Intelligent Control and Automation*, 2010.
- [32] P. Atchade-Adelomou, P. Alonso-Linaje, J. Albo-Canals, D.q.A. Casado-Fauli, Quantum computing approach in mobile robot order picking and batching problem solver optimization, *Algorithms* 14 (2021).
- [33] S. Dasgupta, Stability of quantum computers, 2024, <http://dx.doi.org/10.48550/arXiv.2404.19082>, URL: <http://arxiv.org/abs/2404.19082>, arXiv:2404.19082 [quant-ph]].
- [34] J. Berberich, D. Fink, C. Holm, Robustness of quantum algorithms against coherent control errors, *Phys. Rev. A* 109 (1) (2024) 012417, <http://dx.doi.org/10.1103/PhysRevA.109.012417>, URL: <https://link.aps.org/doi/10.1103/PhysRevA.109.012417>, Publisher: American Physical Society.
- [35] M. Mannone, V. Seidita, A. Chella, Modeling and designing a robotic swarm: a quantum computing approach, *Swarm Evol. Comput.* 79 (101297) (2023) <http://dx.doi.org/10.1016/j.swevo.2023.101297>.
- [36] U. Fano, Description of states in quantum mechanics by density matrix and operator techniques, *Rev. Modern Phys.* 29 (1) (1957) 74–93, <http://dx.doi.org/10.1103/RevModPhys.29.74>, URL: <https://link.aps.org/doi/10.1103/RevModPhys.29.74>, Publisher: American Physical Society.
- [37] E. Schroedinger, Quantisierung als eigenwertproblem, *Annalen der Physik* 79 (1926).
- [38] S. Nakajima, On quantum theory of transport phenomena: Steady diffusion, *In Prog. Theor. Phys.* 20 (6) (1958) 948–959, <http://dx.doi.org/10.1143/PTP.20.948>.
- [39] M.A. Nielsen, I.L. Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press, 2010.
- [40] R.L. Rose, Quantum information and computation seminar at michigan state university, QuIC seminar 6, quantum states and partial trace, 2023, <https://www.ryanlarose.com/uploads/1/1/5/8/115879647/quic06-states-trace.pdf>.
- [41] A.M. Abbas, *Molecular nano communication networks: Architectures*, in: *Proceedings of the 5th International Conference on Multimedia, Signal Processing and Communication Technologies, IMPACT, Protocols and Technologies*, 2012, pp. 1–5.
- [42] F. Phillipson, Quantum computing in telecommunication—A survey, *MDPI Math.* 11 (2023) 15, <http://dx.doi.org/10.3390/math11153423>.
- [43] M.F. Emzir, M.J. Woolley, I.R. Petersen, Stability analysis of quantum systems: A Lyapunov criterion and an invariance principle, *Automatica* 146 (2022).
- [44] M. Mannone, V. Seidita, A. Chella, Quantum computing for swarm robotics: a local-to-global approach, *Philos. Trans. A* 383 (2289) (2025) <http://dx.doi.org/10.1098/rsta.2024.0139>.
- [45] G. Lindblad, On the generators of quantum dynamical semigroups, *Comm. Math. Phys.* 48 (119) (1976).
- [46] R. Olfati-Saber, J.A. Fax, R.M. Murray, Consensus and cooperation in networked multi-agent systems, *Proc. IEEE* 95 (1) (2007) 215–233, <http://dx.doi.org/10.1109/JPROC.2006.887293>.
- [47] G. Lindblad, Completely positive maps and entropy inequalities, *Comm. Math. Phys.* 40 (147–151) (1974).
- [48] M.B. Ruskai, S. Szarek, E. Werner, An analysis of completely-positive trace-preserving maps on 2x2 matrices, *Linear Algebr. Appl.* 347 (2001) 159, arXiv: quant-ph/0101003.
- [49] E. Carlen, J. Maas, Gradient flow and entropy production for quantum Markov semigroups, *J. Funct. Anal.* 273 (2017) 1810–1869.
- [50] M. Mannone, N. Marwan, V. Seidita, A. Chella, A. Giacometti, P. Fazio, Entangled gondolas: Design of multi-layer networks of quantum-driven robotic swarms, in: M. Villani, S. Cagnoni, R. Serra (Eds.), *Artificial Life and Evolutionary Computation. WIVACE 2023*, in: *Communications in Computer and Information Science*, vol. 1977, Springer, Venice, 2023, pp. 177–189, http://dx.doi.org/10.1007/978-3-031-57430-6_14.
- [51] M. Mannone, V. Seidita, A. Chella, A. Giacometti, P. Fazio, Energy and SNR-aware robotic swarm coordination for aquatic cleaning operations, in: *Proceedings of the IEEE VTC Conference*, Vol. 97, Florence, Italy, 2023, pp. 1–7.



Maria Mannone holds an MSc in Theoretical Physics, and three MAs in Composition, Orchestral Conducting and Piano, respectively. She earned the Master 2 ATIAM (IRCAM - UPMC Paris VI Sorbonne), and the Ph.D. in Composition (University of Minnesota, USA). She is currently a researcher at the National Research Council of Italy (CNR). Inventor of the CubeHarmonic, she is a doctoral candidate in Theoretical Physics at the University of Potsdam (Germany). Her research concerns mathematics and music, quantum computing for swarms of robots, and mathematical modeling of neuropsychiatric disorders.



Peppino Fazio received a graduate degree in Computer Science in 2004 and a Ph.D. in Electronics and Telecommunications Engineering in 2008, both from the University of Calabria, Italy. Today he is an Associate Professor at the Ca Foscari University (DSMN) and senior researcher at the VSB-TUO in Czechia. He has co-authored more than 150 papers for international journals and proceedings. His research interests include QoS architectures, mobile ad-hoc networks, quantum key distribution networks, and vehicular networks. He has been member of various start-up companies and is the recipient of international awards, for example, the Intel Business Challenge Award in 2013.



Mahathi Anand Mahathi Anand is a senior researcher at Learning Systems and Robotics Lab, Technical University of Munich. Previously, she has held positions at the Chair of Robotics and System Intelligence, TU Munich and the Institute for Systems Theory and Automatic Control, University of Stuttgart. Earlier, received her Ph.D. from LMU Munich, where she specialized in control theory. Her research interests lie in the intersection between control theory, robotics, and machine learning, and she seeks to develop robot learning and control methodologies with sound theoretical guarantees.



Abdalla Swikir is an Assistant Professor of Robotics at Mohamed bin Zayed University of Artificial Intelligence (MBZUAI), Abu Dhabi, UAE. He is also an IEEE senior member and the recipient of the 2023 IEEE CSS George S. Axelby Outstanding Paper and the 2023 IEEE Robotics and Automation Letters Best Paper Awards. Before joining MBZUAI, he was a Senior Scientist and Teaching Coordinator at the Munich Institute of Robotics and Machine Intelligence (MIRMI) of TUM. His research interests lie at the intersection of artificial intelligence and robotics, pioneering new strategies that harness AIs potential to enhance robotic control systems.