IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 17 September 2025, accepted 30 October 2025, date of publication 3 November 2025,
date of current version 7 November 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3628560

RESEARCH ARTICLE

A Grover-Operator-Based Quantum Variant of Particle Swarm Optimization for Robot Manipulators' Inverse Kinematics: Theory and Simulation

DAVID O. SANTOS[1], FRANCISCO M. DE ASSIS[1], ELYSON A. N. CARVALHO[2], AND LUCAS MOLINA[2]

[1]Department of Electrical Engineering, Federal University of Campina Grande, Campina Grande 58429-900, Brazil
[2]Department of Electrical Engineering, Federal University of Sergipe, São Cristóvão 49100-000, Brazil

Corresponding author: David O. Santos (david.oliveira.santos@ee.ufcg.edu.br)

The Article Processing Charge (APC) for the publication of this research was funded by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) - ROR identifier: 00x0ma614).

ABSTRACT Particle Swarm Optimization (PSO) is a widely used solver for the Inverse Kinematics (IK) problem, offering competitive convergence rates compared to methods like Damped Least Squares and Forward and Backward Reaching Inverse Kinematics. However, PSO's high computational cost per iteration remains a challenge. While many PSO variants focus on reducing the number of iterations, this work introduces a novel approach to decrease the computational cost per iteration by minimizing direct kinematics calculations. Leveraging quantum computing algorithms, specifically Quantum Amplitude Amplification (QAA), we developed Quantum-Grover PSO (QG-PSO), which identifies a suboptimal particle to guide the swarm. This approach capitalizes on QAA's quadratic speedup for unstructured search problems. Simulated experiments demonstrate that QG-PSO significantly reduces computational costs compared to classical PSO when solving the IK problem for robotic manipulators with 7 and 15 joints, highlighting the potential of quantum computing in robotics. In addition to position correction, we considered orientation correction while respecting joint limits and avoiding obstacles.

INDEX TERMS Grover operator, inverse kinematics, particle swarm optimization, quantum amplification amplitude, quantum algorithm, robot manipulator.

I. INTRODUCTION

Manipulator robots are increasingly incorporated into surgical procedures, such as hip replacement, spinal fusion, biopsy collection, and minimally invasive surgeries, highlighting their importance as a technology for human care [1]. A significant challenge in controlling the robot for these and other tasks lies in determining the joint angles that allow the end-effector to achieve a desired position, a problem known as the Inverse Kinematics (IK) problem [2].

Although the IK problem has been studied for many years, it remains an active area of research, continually receiving significant new algorithms, particularly numerical methods [3], [4], [5], [6] and learning-based approaches [7], due to the broad applicability of the IK problem.

Particle Swarm Optimization (PSO) is a numerical IK solver based on a random search in the space defined by the variables of the robot manipulator's joints [8]. It consists of initializing a population of random solutions, called particles, which are then iteratively adjusted in the direction of the particle with best evaluation according to a fitness function.

In [9], PSO was compared to other important numerical IK solvers, including Damped Least Squares (DLS) [10], Forward And Backward Reaching Inverse Kinematics (FABRIK) [11], and Genetic Algorithm (GA) [12]. While PSO achieved competitive performance in terms of final error,

The associate editor coordinating the review of this manuscript and approving it for publication was Zheng H. Zhu.

it performed poorly in terms of the number of iterations and the time required to reach a solution, especially as the number of joints increased. This poor performance limits the utilization of PSO in tasks that require a real-time solution like surgical procedures. In response to this and other challenges, several PSO variants have been introduced. As discussed in [8], most of the PSO variants incorporate strategies such as multiple populations, adaptive parameters, and particle resampling.

A PSO variant with multiple populations was proposed in [13], in which the interaction between groups is based on replacing the worst particles of one group with the best particles from another group, resulting in greater diversity. Another variant with parallel swarms was proposed in [14], where the interaction between groups consists of moving the worst-evaluated particles of one group towards the best-evaluated particles of another group.

The strategy of adaptive behavior was incorporated in the PSO variant proposed in [15], in which PSO is divided into two phases, each using a different probability distribution for the particle position update. In the variant proposed in [16], adaptive behavior is constructed by changing the inertia parameter in each iteration.

In [17], a PSO variant based on particle resampling was proposed, where resampling occurs after a fixed number of iterations. Particle resampling is also seen in the variants proposed in [8] and [18], where resampling occurs after each iteration using a Gaussian distribution. The main difference between them is that in [18], both local and global information are used, while in [8], only global information is utilized.

The Quantum-Behaved PSO (QBPSO), explored in [19], is an important PSO variant that is not based on the strategies of multiple populations, adaptive parameters, or particle resampling. Instead, it replaces the position update equations, inspired by the movement of schools of fish and flocks of birds, with equations inspired by the movement of quantum particles. Another PSO variant was proposed in [20], called Game-based Social Learning PSO (GSLPSO), in which the new equation used to update the particles' positions is based on game theory.

All the cited variants have, as one of their motivations, the goal of reducing the number of iterations required for the method to solve the IK problem. In these works, it is assumed, even implicitly, that the cost per iteration is similar to that of the original PSO. Therefore, a smaller number of iterations would imply a shorter time to solve the IK problem. In this paper, we propose a new perspective: instead of reducing the number of iterations, the goal is to significantly reduce the cost per iteration. To achieve this, we explored two key principles: using a suboptimal particle to guide the others and incorporating quantum algorithms into the PSO. Quantum algorithms are based on Quantum Computation theory, with their performance gains being realized only when implemented on quantum hardware, in contrast to QBPSO, which is a classical method that merely draws inspiration from quantum phenomena without requiring quantum hardware.

A. QUANTUM ALGORITHMS

Quantum Computation is the study of information processing tasks performed using quantum mechanical systems. Information is represented by physical quantities that exhibit quantum characteristics such as superposition and entanglement, allowing multiple pieces of information to be processed in parallel. Algorithms designed to leverage these principles are known as quantum algorithms, and they have shown potential to be significantly less costly compared to classical methods for many complex problems [21]. However, significant technological limitations remain regarding the number of quantum bits (qubits) that can be integrated into a single chip or device. How discussed in [22], this number is far below what is required to execute complex quantum algorithms.

Despite the technological limitations, the theory of quantum computation is well-established, and many quantum algorithms have been proposed. Two emblematic examples are Shor's and Grover's algorithms [21]. Shor's algorithm was developed to solve the prime factorization problem, providing a theoretical demonstration of exponential speedup [23]. On the other hand, Grover's algorithm addresses the search problem in unstructured databases, achieving quadratic speedup [24].

Another emblematic algorithm is Quantum Amplitude Amplification (QAA), which generalizes Grover's algorithm for situations where there is more than one desirable element in the database [25], [26]. QAA is a versatile algorithm that has been integrated with classical algorithms to create new quantum algorithms. For example, in [27], a quantum reinforcement learning algorithm was proposed, which uses QAA as a subroutine.

Quantum Genetic Sampling is another example of a quantum algorithm that integrates QAA with a classical algorithm [28]. Specifically, QAA is employed to create a new selection mechanism for the crossover step. The goal of this new formulation is to improve the balance between the diversity of solutions and satisfactory convergence capabilities.

In the works [29], [30], quantum variants of the Rapidly Exploring Random Tree (RRT) method were proposed, utilizing QAA as a subroutine. RRT is a popular path-planning method for mobile robots. In simulated experiments, these quantum variants successfully achieved a speedup in path construction.

The Grover algorithm was recently explored in the context of manipulator robots in [31], as part of a control strategy in which the task of computing the control law was reformulated as a search problem. The approach demonstrated promising theoretical performance in simulation results. Thus, Grover's algorithm or QAA may also enhance the performance of metaheuristic methods such as PSO, since certain steps in

these algorithms can be reformulated as search problems, allowing QAA to contribute to cost reduction.

B. CONTRIBUTIONS

In this paper, we introduce a new perspective for meta-heuristic methods by using a quantum formulation to reduce the cost per iteration, rather than proposing variations to reduce the number of iterations, which is a common focus in the literature, especially in PSO variants. Although our formulation can be applied to other meta-heuristic algorithms, we specifically investigate this new perspective within the original PSO algorithm. To achieve this, we propose a new method called Quantum-Grover PSO (QG-PSO), which integrates PSO with the QAA. The QAA is used to identify a suboptimal particle in each iteration, which then guides the other particles in the swarm. Although the widespread availability of quantum hardware is on the horizon, this work represents a significant step toward the application of quantum computing for practical optimization problems in robotics, as discussed in [32], and it demonstrates the potential of quantum algorithms to enhance existing methodologies.

The main contributions of this work are as follows:

- Proposing a suboptimal search as a new strategy to reducing the cost of PSO.
- Introducing the use of the QAA algorithm as a novel approach to accelerate PSO.
- Formulating a new variant of PSO, called Quantum-Grover PSO (QG-PSO), which incorporates both strategies to enhance PSO's performance.
- Evaluating the performance of QG-PSO in solving the IK problem for manipulator robots, considering environments with and without obstacles. Comparing the performance of QG-PSO against classical PSO and MFRPSO in the IK problem, assuming the QG-PSO is implemented on quantum hardware.

The paper is organized as follows: Section II describes the formulation of the IK problem and the formulation of PSO; Section III provides an introduction to the Quantum Computation theory; QAA and QG-PSO methods are explained in detail in Section IV; An illustrative example of the concepts discussed in our work is presented in Section V; Section VI is dedicated to the analysis of experiments comparing the performance of Classical and Quantum-Grover PSO algorithms in solving the IK problem; in Section VII, the limitations of the work are presented; Section VIII presents the conclusions.

II. INVERSE KINEMATICS AND PSO

A. FORMULATION OF THE INVERSE KINEMATICS PROBLEM

Let $\xi_d = [\boldsymbol{p}_d, \boldsymbol{o}_d]^T \in \mathcal{R}^6$ be the pose of the robot, where $\boldsymbol{p}_d = [x_d, y_d, z_d]$ is the vector representing the desired position and $\boldsymbol{o}_d = [\phi_d, \theta_d, \psi_d]$ is the vector representing the desired orientation using the Euler convention [2]. The IK problem is a mathematical problem that aims to solve the equation

$$f(\boldsymbol{q}) = \xi_d, \tag{1}$$

where $\boldsymbol{q} \in \mathcal{R}^n$ is the robot's configuration, which is a vector composed of the articulation variables of the robot arm, n is the number of robot arm's articulations, and f is the forward kinematic function that maps a vector $\boldsymbol{q}$ to a vector $\xi = [\boldsymbol{p}, \boldsymbol{o}]^T$. The $f(\boldsymbol{q})$ can be systematically computed using homogeneous transformations and the Denavit-Hartenberg convention [2].

For meta-heuristic methods, the IK problem is reformulated as an optimization problem. To achieve this, a cost function $J(\boldsymbol{q})$ can be defined as

$$J(\boldsymbol{q}) = w_p \|\boldsymbol{p}_d - \boldsymbol{p}(\boldsymbol{q})\| + w_o \|\boldsymbol{o}_d - \boldsymbol{o}(\boldsymbol{q})\|, \tag{2}$$

where $\boldsymbol{p}(\boldsymbol{q})$ and $\boldsymbol{o}(\boldsymbol{q})$ are obtained through $f(\boldsymbol{q})$, $\|\cdot\|$ denotes the Euclidean norm, w_p and w_o are parameters that weight the position and orientation errors, respectively. They are set to 1 / m and 0.1 / rad in this work, with positions measured in meters and angles measured in radians. The IK problem is considered solved when $J(\boldsymbol{q}) < \epsilon$, where ϵ is the desired threshold.

B. PSO

The PSO initializes N vectors $\boldsymbol{q}_i \in \mathcal{R}^n$, denoted as position of i-th particle with $i = 1, \ldots, N$. The sampling is done with a uniform distribution $\mathcal{U}(\boldsymbol{q}_{\min}, \boldsymbol{q}_{\max})$, where $\boldsymbol{q}_{\min}$ and $\boldsymbol{q}_{\max}$ are vectors representing the limits of the articulation variables. After initialization, the PSO consists in updating the values of x_i through the following equations:

$$\begin{aligned}\boldsymbol{v}_i(k) = w\boldsymbol{v}_i(k-1) &+ c_l \text{rand}()(\boldsymbol{q}_{\text{lbest},i} - \boldsymbol{q}_i(k)) \\ &+ c_g \text{rand}()(\boldsymbol{q}_{\text{gbest}} - \boldsymbol{q}_i(k)),\end{aligned} \tag{3}$$

$$\boldsymbol{q}_i(k+1) = \boldsymbol{q}_i(k) + \boldsymbol{v}_i(k), \tag{4}$$

where $v_i(k)$ is the velocity vector of i-th particle in iteration k, $q_{\text{lbest},i}$ is the best position of the particle i up to iteration k and $\boldsymbol{q}_{\text{gbest}}$ is the global best swarm position. The parameters c_l e c_g weigh the local and global behavior, respectively. The term $rand()$ is random variable $\sim \mathcal{U}(0, 1)$. Additionally, w is the inertial weight.

In each iteration, the PSO calculates the forward kinematics of each particle $\boldsymbol{q}_i$ to find $\boldsymbol{q}_{\text{gbest}}$. Thus, the cost per iteration of the algorithm in terms of forward kinematics calculations is N. The aim of this paper is to decrease this value. To address this, a quantum variation of PSO is proposed.

III. AN INTRODUCTION TO QUANTUM COMPUTATION

Classical computation consists of manipulating information represented in terms of bits, which are implemented through physical phenomena that have only two mutually exclusive states. In contrast, quantum computation manipulates information represented in terms of quantum bits or qubits, which are implemented through quantum phenomena that contain quantum attributes such as superposition and entanglement [21].

A. DEFINITION OF A QUBIT AND DIRAC NOTATION

A quantum state is normally denoted in the Dirac notation, $|\cdot\rangle$, where the states of the computational base of a single qubit are denoted as $|0\rangle$ and $|1\rangle$. Quantum Superposition consists of a qubit $|\psi\rangle$ assuming both states simultaneously [21]. In this case, the qubit $|\psi\rangle$ can be expressed as a linear combination of the states $|0\rangle$ and $|1\rangle$,

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \tag{5}$$

where α and β are complex scalars and represented probability amplitudes. Although a qubit can assume both states simultaneously during the processing, when measurement occurs, the qubit randomly collapse only one of the states [21]. The probability of measuring the state $|0\rangle$ is $|\alpha|^2$ and the probability of measuring the state $|1\rangle$ is $|\beta|^2$. Thus, $|\alpha|^2 + |\beta|^2 = 1$.

Defining $|0\rangle = \begin{bmatrix}1\\0\end{bmatrix}$ and $|1\rangle = \begin{bmatrix}0\\1\end{bmatrix}$, the state $|\psi\rangle$ in (5) can be rewritten in a vector notation as

$$|\psi\rangle = \begin{bmatrix}\alpha\\\beta\end{bmatrix}. \tag{6}$$

In this notation, quantum gates can be represented in matrix format. For example, the quantum gate NOT (X), defined as a quantum operation that $|0\rangle \xrightarrow{X} |1\rangle$ and $|1\rangle \xrightarrow{X} |0\rangle$, is expressed as

$$\boldsymbol{X} = \begin{bmatrix}0 & 1\\1 & 0\end{bmatrix}, \tag{7}$$

and the result of its application is

$$\boldsymbol{X}\begin{bmatrix}\alpha\\\beta\end{bmatrix} = \begin{bmatrix}\beta\\\alpha\end{bmatrix}. \tag{8}$$

B. QUANTUM SYSTEMS WITH MORE THAN ONE QUBIT

The complete state of a quantum system with more than one qubit can be expressed using the tensor product ($\otimes$) [21]. The operation $A_{M\times N} \otimes B_{O\times P}$ is defined as

$$\boldsymbol{A}_{M\times N} \otimes \boldsymbol{B}_{O\times P} = \begin{bmatrix} a_{11}\boldsymbol{B} & \cdots & a_{1N}\boldsymbol{B}\\ \vdots & \ddots & \vdots\\ a_{M1}\boldsymbol{B} & \cdots & a_{MN}\boldsymbol{B}\end{bmatrix}_{MO\times NP}, \tag{9}$$

where a_{ij} represents the element (i, j) of the matrix A. If there are n systems enumerated 1 to n, and in each system i is represented as $|\psi_i\rangle$ with $i = 1, \ldots, n$, then the complete state is $|\psi_1\rangle \otimes |\psi_2\rangle \otimes \ldots \otimes |\psi_n\rangle$. In cases where $|\psi_i\rangle$ and $|\psi_j\rangle$ represents binary words, $|\psi_i\rangle \otimes |\psi_j\rangle = |\psi_i\psi_j\rangle$ [21]. For example, $|0\rangle \otimes |1\rangle = |01\rangle$ and $|1\rangle \otimes |0\rangle = |10\rangle$. Following this notation, the following results can be achieved:

$$|00\rangle = \begin{bmatrix}1\\0\\0\\0\end{bmatrix}, |01\rangle = \begin{bmatrix}0\\1\\0\\0\end{bmatrix}, |10\rangle = \begin{bmatrix}0\\0\\1\\0\end{bmatrix}, |11\rangle = \begin{bmatrix}0\\0\\0\\1\end{bmatrix}. \tag{10}$$

Quantum entanglement refers to a quantum state $|\psi\rangle$ involving multiple qubits that cannot be factored into the direct product of individual qubit states [21]. For example:

$$|\psi\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}} \tag{11}$$

cannot be expressed as $|\psi_1\rangle \otimes |\psi_2\rangle$. The state $|\psi\rangle$ is a superposition of the states $|00\rangle$ and $|11\rangle$. In this case, the qubits involved are entangled, and after measurement, the outcome will be either $|00\rangle$ or $|11\rangle$ with equal probability. It is impossible to obtain the states $|01\rangle$ or $|10\rangle$. Furthermore, by measuring one of the qubits, it is possible to determine unequivocally what the result of the other will be. If one qubit collapses to $|0\rangle$, the other will also collapse to $|0\rangle$. Similarly, if one qubit collapses to $|1\rangle$, the other will collapse to $|1\rangle$.

C. QUANTUM OPERATORS

The logical gates AND, OR, and NOT can implement any arbitrary classical combinational logic circuit, and for that reason, the set composed of these gates is said to form a universal set. In quantum computation, one of the universal sets is $\{\boldsymbol{H}, \boldsymbol{T}, \boldsymbol{CNOT}\}$ [21]. Since these gates are linear operators, they can be expressed in matrix form:

$$\boldsymbol{H} = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\1 & -1\end{bmatrix}, \tag{12}$$

$$\boldsymbol{T} = \begin{bmatrix}1 & 0\\0 & e^{i\pi/4}\end{bmatrix}, \tag{13}$$

$$\boldsymbol{CNOT} = \begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&0&1\\0&0&1&0\end{bmatrix}. \tag{14}$$

All three operators are unitary, meaning they have inverses and preserve the norm. Any quantum operator must proposed have these properties to satisfy the postulates of quantum mechanics.

The gate $\boldsymbol{H}$, known as the Hadamard Gate or Walsh-Hadamard transformation, is widely used in quantum algorithms [21]. It is an operator that maps a quantum state from the computational basis to a superposition of all states with a uniform probability distribution. For example:

$$\boldsymbol{H}|0\rangle = \frac{1}{\sqrt{2}}\begin{bmatrix}1\\1\end{bmatrix}, \boldsymbol{H}|1\rangle = \frac{1}{\sqrt{2}}\begin{bmatrix}1\\-1\end{bmatrix} \tag{15}$$

In both cases, the outcome is a new qubit with a uniform probability distribution over the states of the computational basis ($|\alpha| = |\beta| = 0.5$). For systems with n qubits, this outcome can be achieve by applying the gate $\boldsymbol{H}$ to each qubit in parallel. These operations can be represented by a unique operator: $H^{\otimes n} = \boldsymbol{H} \otimes \boldsymbol{H} \otimes \ldots \otimes \boldsymbol{H}$ ($\boldsymbol{H}$ repeated n times with $n-1$ repetitions of $\otimes$).

Another operator used in quantum algorithms is the Phase Shift Operator (P), defined as:

$$|\psi\rangle \xrightarrow{\boldsymbol{P}} \begin{cases} |\psi\rangle, & \text{if } |\psi\rangle = |0\rangle^{\otimes n}\\ -|\psi\rangle, & \text{if } \psi\rangle \neq |0\rangle^{\otimes n}\end{cases}, \tag{16}$$

where n is the number of qubits used to represent $|\psi\rangle$. The operator P shifts the sign of all elements of the vector that represents the quantum state, except the first one. Thus, it acts as a reflection operator with respect to the vector $|0\rangle^n$.

A reflection operator $\boldsymbol{R}_f(\boldsymbol{v})$ with respect to a unit column vector $\boldsymbol{v}$ is defined as

$$\boldsymbol{R}_f(\boldsymbol{v}) = 2\boldsymbol{v}\boldsymbol{v}^T - I, \tag{17}$$

where $\boldsymbol{I}$ is the operator identity. Equivalently, in Dirac notation,

$$\boldsymbol{R}_f(|v\rangle) = 2|\boldsymbol{v}\rangle\langle\boldsymbol{v}| - I, \tag{18}$$

where $\langle\boldsymbol{v}|$ denote the transpose conjugate of $|\boldsymbol{v}\rangle$. Thus, the operator $\boldsymbol{P}$ can expressed as

$$\boldsymbol{P} = 2|0\rangle^{\otimes n}\langle 0|^{\otimes n} - \boldsymbol{I}. \tag{19}$$

IV. QUANTUM ALGORITHMS

In this section, the quantum algorithms QAA and QG-PSO are introduced in subsections A and B, respectively.

A. QUANTUM AMPLIFICATION AMPLITUDE

In classical computation, a vector $\boldsymbol{q}$ is represented by a classical binary word. In quantum computation, this same vector can represented in a binary word, in which each bit is a quantum bit. This form, a quantum state $|\boldsymbol{q}\rangle$ can be prepared in superposition of binary words that represents the vectors $\boldsymbol{q}_i$ with $i = 1, \ldots, N$,

$$|\boldsymbol{q}\rangle = \sum_{i=1}^{N} a_i|\boldsymbol{q}_i\rangle, \tag{20}$$

where each state $|\boldsymbol{q}_i\rangle$ is represented computationally by a binary word that corresponds to the value of integer $i-1$ and $\sum_{i=1}^{N}|a_i|^2 = 1$.

The forward kinematics is computed on a classical computer through a set of logical gates. A quantum version of this function can be built by replacing each logical gate with a similar quantum gate [21]. In this way, the quantum state $|f(\boldsymbol{q})\rangle$ can be computed, which is a superposition of each $|f(\boldsymbol{q}_i)\rangle$. However, these results cannot be directly observed because when a measurement occurs, the state will collapse to a single $|f(\boldsymbol{q}_i)\rangle$ with probability $|a_i|^2$.

Let $\hat{\mathcal{Q}} = \{\boldsymbol{q}_1, \boldsymbol{q}_2, \ldots, \boldsymbol{q}_N\}$ be a set of configurations of a robot manipulator. Assume that all configurations are represented by binary words of $n = \log_2 N$ bits, with $\boldsymbol{q}_i$ corresponding to the i-th value from 0 to $N-1$, respectively, among which M configurations are desirable. Additionally, let $\boldsymbol{f}_o(\boldsymbol{q})$ be a Boolean function that maps desirable configurations to 1 and undesirable configurations to 0. Let $\mathcal{Q} = \{|\boldsymbol{q}_1\rangle, |\boldsymbol{q}_2\rangle, \ldots, |\boldsymbol{q}_N\rangle\}$ represent the same set, but with each configuration represented by binary words using quantum bits. In this context, we can define three states:

$$|\boldsymbol{q}_d\rangle = \frac{1}{\sqrt{M}}\sum_{\boldsymbol{q}\in\mathcal{Q}_d}|\boldsymbol{q}\rangle, \tag{21}$$

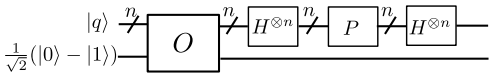


FIGURE 1. An implementation of the Grover operator, considering that the state $|\boldsymbol{q}\rangle$ is represented by n qubits.

$$|\boldsymbol{q}_{\bar{d}}\rangle = \frac{1}{\sqrt{N-M}}\sum_{\boldsymbol{q}\in\mathcal{Q}_{\bar{d}}}|\boldsymbol{q}\rangle, \tag{22}$$

$$|u\rangle = \sqrt{\frac{N-M}{N}}|\boldsymbol{q}_{\bar{d}}\rangle + \sqrt{\frac{M}{N}}|\boldsymbol{q}_d\rangle, \tag{23}$$

where $\mathcal{Q}d = \{|\boldsymbol{q}\rangle \in \mathcal{Q}/f_o(\boldsymbol{q}) = 1\}$ and $\mathcal{Q}\bar{d} = \{|\boldsymbol{q}\rangle \in \mathcal{Q}/f_o(\boldsymbol{q}) = 0\}$. The states $|u\rangle$, $|\boldsymbol{q}_d\rangle$, and $|\boldsymbol{q}_{\bar{d}}\rangle$ represent uniform distributions over the sets $\mathcal{Q}$, $\mathcal{Q}d$, and $\mathcal{Q}\bar{d}$, respectively.

The QAA algorithm involves manipulating the quantum state $|\boldsymbol{u}\rangle$ to obtain the state $|\boldsymbol{q}_d\rangle$. To achieve this, Grover's operator, denoted as $\boldsymbol{G}$, is employed, which is implemented by the quantum circuit shown in Fig. 1. The circuit consists of the sequential application of four operators in the following order: Oracle ($\boldsymbol{O}$), Hadamard ($\boldsymbol{H}^{\otimes n}$), Phase Shift Operator ($\boldsymbol{P}$), and Hadamard again.

The Oracle operator in this context is defined as:

$$|\boldsymbol{q}\rangle \otimes |\boldsymbol{a}\rangle \xrightarrow{\boldsymbol{O}} |\boldsymbol{q}\rangle \otimes |\boldsymbol{a} \oplus f_o(\boldsymbol{q})\rangle, \tag{24}$$

where $\oplus$ represents the XOR (exclusive OR) operation. Note that one application of the operator $\boldsymbol{O}$ implies one application of the function f_o. In the QAA algorithm, the auxiliary qubit $|\boldsymbol{a}\rangle$ is set to the state:

$$|\boldsymbol{a}\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle). \tag{25}$$

For this value of $|\boldsymbol{a}\rangle$, the Oracle operator modifies the state as follows:

$$|\boldsymbol{q}\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \xrightarrow{\boldsymbol{O}} |q\rangle(-1)^{f_o(q)} \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle). \tag{26}$$

For analysis purposes, since the term $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ appears on both sides of the equation, it can be omitted without any loss of generality. Thus, the Oracle operator can be simplified as:

$$|\boldsymbol{q}\rangle \xrightarrow{\boldsymbol{O}} \begin{cases} -|\boldsymbol{q}\rangle, & \text{if } f_o(\boldsymbol{q}) = 1 \\ |\boldsymbol{q}\rangle, & \text{if } f_o(\boldsymbol{q}) = 0 \end{cases}. \tag{27}$$

In this simplified form, the Oracle acts like a reflection operator with the state $\boldsymbol{q}_d$.

Using the simplified form of Oracle Operator, the Grover operator can be expressed as

$$\boldsymbol{G} = \boldsymbol{H}^{\otimes n}\boldsymbol{P}\boldsymbol{H}^{\otimes n}\boldsymbol{O}. \tag{28}$$

Substituting (19) in (28) and using the properties that $\boldsymbol{H}^{\otimes n}\boldsymbol{H}^{\otimes n} = \boldsymbol{I}_{2^n}$ and $\boldsymbol{H}^{\otimes n}|0\rangle = |u\rangle$, we can rewrite the Grover Operator as:

$$\boldsymbol{G} = \boldsymbol{H}^{\otimes n}(2|0\rangle^{\otimes n}\langle 0|^{\otimes n} - \boldsymbol{I}_{2^n})\boldsymbol{H}^{\otimes n}\boldsymbol{O}, \tag{29}$$

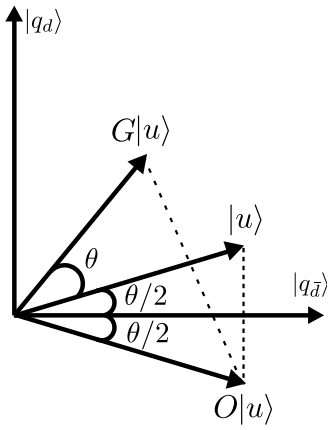


FIGURE 2. Illustration of the application of both operators O and G in the plane generated by the basis $\{|q_{\bar{d}}\rangle, |q_d\rangle\}$, based in [21].

$$G = (2|\boldsymbol{u}\rangle\langle\boldsymbol{u}| - \boldsymbol{I}_{2^n})\boldsymbol{O}. \tag{30}$$

The expression $(2|\boldsymbol{u}\rangle\langle\boldsymbol{u}| - \boldsymbol{I}_{2^n})$ in (30) represents a reflection operator with respect to the state $|u\rangle$. Thus, the Grover operator is a sequential application of two reflection operators: the first with respect to $|\boldsymbol{q}_d\rangle$ and the second with respect to $|\boldsymbol{u}\rangle$.

Let θ be an angular variable such that $\cos(\theta/2) = \sqrt{(N-M)/N}$, then

$$|\boldsymbol{u}\rangle = \cos(\theta/2)|\boldsymbol{q}_{\bar{d}}\rangle + \sin(\theta/2)|\boldsymbol{q}_d\rangle. \tag{31}$$

As illustrated in Fig.2, the two reflections of the Grover operator on the state $|\boldsymbol{u}\rangle$ result in a rotation by an angle θ in the plane generated by the basis $\{|\boldsymbol{q}_{\bar{d}}\rangle, |\boldsymbol{q}_d\rangle\}$.

Therefore,

$$|\boldsymbol{u}\rangle \xrightarrow{G} \cos\left(\frac{3\theta}{2}\right)|\boldsymbol{q}_{\bar{d}}\rangle + \sin\left(\frac{3\theta}{2}\right)|\boldsymbol{q}_d\rangle. \tag{32}$$

Furthermore, after k applications of the Grover operator, the result is

$$|\boldsymbol{u}\rangle \xrightarrow{\boldsymbol{G}^k} \cos\left(\frac{2k+1}{2}\theta\right)|\boldsymbol{q}_{\bar{d}}\rangle + \sin\left(\frac{2k+1}{2}\theta\right)|\boldsymbol{q}_d\rangle. \tag{33}$$

The goal of the QAA algorithm is to make

$$|\boldsymbol{u}\rangle \xrightarrow{\boldsymbol{G}^R} \approx |q_d\rangle \tag{34}$$

a good approximation, where in the original formulation R is calculated as a function of M. However, in the context of this paper, M is unknown. Therefore, the iterative version of QAA discussed in [26] is used, which does not require knowledge of the value of M, and its cost, in terms of the number of times the boolean function f is evaluated, is $O(\sqrt{N/M})$.

The iterative version of the QAA algorithm is described in Algorithm 1. Instead of applying the Grover operator R times and performing only one measurement, the algorithm iteratively applies the Grover operator j times, followed by a measurement. If a desirable state is not obtained, the Grover operator is applied again j times, and the process repeats until either a desirable state is found or the control variable $m \geq \sqrt{N}$. The value of j is updated in each iteration using the probability distribution $\mathcal{U}(1, \textbf{round}(m))$.

Algorithm 1 Quantum Amplification Amplitude (QAA) Based on [26]

Require: N,Q
Ensure: $\hat{\boldsymbol{q}}$,success
$m \leftarrow 1$
$\lambda \leftarrow 6/5$
success $\leftarrow 0$
while not(success) **do**
  $j \leftarrow \mathcal{U}(1, \textbf{round}(m))$ {Sampling}
  Prepare the state $|\boldsymbol{u}\rangle$ as a superposition of all elements of Q with uniform distribution.
  $|\boldsymbol{q}\rangle \leftarrow \boldsymbol{G}^j|\boldsymbol{u}\rangle$
  $\hat{\boldsymbol{q}}$ = **measure**($|\boldsymbol{q}\rangle$) {Transition from quantum to classical hardware.}
  if $f(\hat{\boldsymbol{q}}) \stackrel{?}{=} 1$ **then**
    success = 1
  else
    if $m \geq \sqrt{N}$ **then**
      break
    else
      $m \leftarrow \textbf{min}(\lambda m, \sqrt{N})$
    end if
  end if
end while

B. QUANTUM-GROVER PARTICLE SWARM OPTIMIZATION

Assuming N particles, each PSO iteration requires N calculations of forward kinematics to find $\boldsymbol{q}_{\text{gbest}}$. In this paper, it is proposed to search a suboptimal particle $\boldsymbol{q}_{\text{gsub}}$ instead of the best particle $\boldsymbol{q}_{\text{gbest}}$. The $\boldsymbol{q}_{\text{gsub}}$ define the first particle found in iteration k that is better evaluated than the current $\boldsymbol{q}_{\text{gsub}}$. This must reduce the cost from N to N/M, where M is the number of particles that are better evaluated than the current $\boldsymbol{q}_{\text{gsub}}$. Furthermore, a Quantum-Grover PSO is proposed, which finds $\boldsymbol{q}_{\text{gsub}}$ using the QAA algorithm, aiming to reduce the cost per iteration.

The boolean function $f_o(\boldsymbol{q})$ used to find a new $\boldsymbol{q}_{\text{gsub}}$ is defined as

$$f_o(\boldsymbol{q}) = J(\boldsymbol{q}) < cJ_{\text{best}}, \tag{35}$$

where $0 < c \leq 1$ is an adjust parameter and $J_{\text{best}} = J(\boldsymbol{q}_{\text{gsub}})$ that is calculated in stored and past iterations. When using the QAA algorithm, only one particle is returned. Consequently, the values of $\boldsymbol{q}_{\text{lbest},i}$ are unknown. Therefore, in QG-PSO, c_l is set to zero, and the new equation becomes:

$$\boldsymbol{v}_i(k) = w\boldsymbol{v}_i(k-1) + c_g\text{rand}()(\boldsymbol{q}_{\text{gsub}} - \boldsymbol{q}_i(k)). \tag{36}$$

Note that one application of the operator $\boldsymbol{O}$ means one application of f_o, which in turn triggers an application of J, and that also means one go at the forward kinematics function f. The QG-PSO is described in Algorithm 2.

Algorithm 2 Quantum-Grover Particle Swarm Optimization

Require: ξ_d,K,N,ϵ
Ensure: $\boldsymbol{q}_{gsub}$
1: $Q \leftarrow$ Initialize N configurations $\boldsymbol{q}_i$ sampled from $\sim \mathcal{U}(\boldsymbol{q}_{\min}, \boldsymbol{q}_{\max})$, with $i = 1, \ldots, N$.
2: Initialize the velocities of N particles with null vectors.
3: Evaluate all N particles using $J(\boldsymbol{q})$, and through an exhaustive search find $\boldsymbol{q}_{\text{gbest}}$.
4: $\boldsymbol{q}_{\text{gsub}} \leftarrow \boldsymbol{q}_{\text{gbest}}$
5: **for** $k = 1$ to K **do**
6: **for** $i = 1$ to N **do**
7: $\boldsymbol{v}_i(k) = w\boldsymbol{v}_i(k-1) + c_g\text{rand}()(\boldsymbol{q}_{\text{gsub}} - \boldsymbol{q}_i(k))$
8: $\boldsymbol{q}_i(k+1) \leftarrow \boldsymbol{q}_i(k) + \boldsymbol{v}_i(k)$
9: **end for**
10: $\hat{\boldsymbol{q}}$, success $\leftarrow$ QAA(Q, N) (Using a quantum hardware)
11: **if** success $\stackrel{?}{=} 1$ **then**
12: $\boldsymbol{q}_{\text{gsub}} \leftarrow \hat{\boldsymbol{q}}$
13: **end if**
14: **if** $J(\boldsymbol{q}_{\text{gsub}}) < \epsilon$ **then**
15: **break**
16: **end if**
17: **end for**

V. ILLUSTRATIVE EXAMPLE AND DISCUSSION ON GROVER OPERATOR

In this section, we present a toy example of the Grover operator to aid in understanding the concepts underpinning our work

Let $\mathcal{Q} = \{\boldsymbol{q}_1, \boldsymbol{q}_2, \boldsymbol{q}_3, \boldsymbol{q}_4\}$ be a set of four configurations of a robot manipulator, represented by the binary words 00_2, 01_2, 10_2, and 11_2, respectively. Assume that only the configuration q_3 (or 10_2) satisfies the function $f(\boldsymbol{q})$. Thus. $\mathcal{Q}_d = \{\boldsymbol{q}_3\}$ and $\mathcal{Q}_{\bar{d}} = \{\boldsymbol{q}_1, \boldsymbol{q}_2, \boldsymbol{q}_4\}$. Using the definitions in equations (21), (22) and (23), we obtain:

$$|q_d\rangle = \begin{bmatrix}0\\0\\1\\0\end{bmatrix}, |q_{\bar{d}}\rangle = \frac{1}{\sqrt{3}}\begin{bmatrix}1\\1\\0\\1\end{bmatrix}, |u\rangle = \frac{1}{2}\begin{bmatrix}1\\1\\1\\1\end{bmatrix}. \tag{37}$$

According to equations (27), (19) and (12) the operators that compose the Grover operator are expressed as follows:

$$\boldsymbol{O} = \begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&-1&0\\0&0&0&1\end{bmatrix}, \tag{38}$$

$$\boldsymbol{P} = \begin{bmatrix}1&0&0&0\\0&-1&0&0\\0&0&-1&0\\0&0&0&-1\end{bmatrix}, \tag{39}$$

$$\boldsymbol{H}^{\otimes 2} = \frac{1}{2}\begin{bmatrix}1&1&1&1\\1&-1&1&-1\\1&1&-1&-1\\1&-1&-1&1\end{bmatrix}. \tag{40}$$

By substituting the values of the operators into equation (28), we obtain the following expression for the Grover operator:

$$\boldsymbol{G} = \frac{1}{2}\begin{bmatrix}-1&1&-1&1\\1&-1&-1&1\\1&1&1&1\\1&1&-1&-1\end{bmatrix}. \tag{41}$$

Finally, applying the Grover operator to the initial state in the Quantum Amplitude Amplification (the uniform superposition state $|\boldsymbol{u}\rangle$), we obtain:

$$\boldsymbol{G}|u\rangle = \begin{bmatrix}0\\0\\1\\0\end{bmatrix}, \tag{42}$$

which corresponds to the state $|\boldsymbol{q}_d\rangle$.

Thus, in this toy example, with a single application of the Grover operator, the probability of measuring the state q_3 (desirable state) increase from 0.25 to 1 Although this is a toy example, it illustrates the key aspects of the Grover algorithm. In cases with larger sets $\mathcal{Q}$ with more than one desirable state, it is common that more applications of the Grover operator are necessary to achieve a high probability of measuring a desirable state. Furthermore, the algorithm does not guarantee that the probability of measuring a desirable state will be exactly 1 after applying the Grover operator multiple times.

We emphasize that the mathematical expression for the Oracle operator was derived with prior knowledge of the desirable states, which is not available in real-world scenarios. Moreover, any computational advantage can only be realized if the process is implemented on quantum hardware, as the algorithm relies on the oracle circuit implementing the function $f(\boldsymbol{q})$ being applied to all configurations simultaneously in a single execution. In contrast, a classical implementation would need to apply the function $f(\boldsymbol{q})$ to each state individually-potentially multiple times in real-world scenarios-resulting in a much higher computational cost. The analysis of the experimental results in the following section assumes that the Grover operator is implemented on quantum hardware.

VI. EXPERIMENTAL RESULTS

In this section, the experiments performed to evaluate aspects of QG-PSO in the IK problem are presented. Experimental results using classical PSO are also reported to serve as a baseline for comparison. The comparison is further extended to include MFRPSO, which outperformed other algorithms in experiments reported in [20], including PSO, QPSO, and GA algorithms applied to the IK problem of a 6-DoF manipulator. The experiments are conducted in scenarios with and without obstacles, in which the methods are applied to 7-DoF and 15-DoF manipulators. The 7-DoF manipulator is the Pioneer arm described in [11], while the 15-DoF is a theoretical custom version with additional joints. The methods are

TABLE 1. Denavit-Hartenberg table for the 7-DoF manipulator (Pioneer arm) [11]. Length parameters are given in millimeters (mm), and angles are given in degrees.

Junta	a	α	d	θ
1	0	90	125	$-60 \leq 90 + q[1] \leq 240$
2	0	-90	0	$-88 \leq q[2] \leq 88$
3	0	90	150	$-150 \leq q[3] \leq 150$
4	0	-90	0	$-93 \leq q[4] \leq 93$
5	0	90	145	$-150 \leq q[5] \leq 150$
6	75	90	0	$-16 \leq 90 + q[6] \leq 196$
7	75	0	0	$-103 \leq q[7] \leq 103$

TABLE 2. Denavit-Hartenberg table for the 15-DoF manipulator. Length parameters are given in millimeters (mm), and angles are given in degrees.

Joint	a	α	d	θ
1	0	90	125	$-60 \leq 90 + q[1] \leq 240$
$i = 2, 4, \cdots, n-3$	0	-90	0	$-93 \leq q[i] \leq 93$
$i = 3, 5, \cdots, n-2$	0	90	150	$-150 \leq q[i] \leq 150$
14	75	90	0	$-16 \leq 90 + q[14] \leq 196$
15	75	0	0	$-103 \leq q[15] \leq 103$

compared in terms of convergence rate (c_r) and total quantity of direct kinematic calculations (f_t) required for the method to converge to a desirable solution. The convergence rate c_r quantifies the probability of the algorithm converging to the desirable solution, and f_t represents the cost to obtain the solution in terms of direct kinematic calculations. The implementations used in this study are available at [33].

The experimental setup is as follows: in environments without obstacles, each experiment is repeated 1000 times, while in environments with obstacles, each experiment is repeated 100 times due the high computational cost in collision detection and quantum simulation. In both cases, the maximum number of iterations is set 300 and ξ_d is calculated as $\xi_d = f(\boldsymbol{q}_d)$ where $\boldsymbol{q}_d$ is uniformly sampled in the C-space for each trial. The Denavit-Hartenberg parameters of the 7-DoF and 15-DoF manipulators used in the experiments are shown in the tables 1 and 2, respectively. The PSO parameters used in the experiments were: $c_l = 0.1$, $c_g = 2$, $w \sim \mathcal{U}(0.5, 1)$ and $v_{max} = 2$. The QG-PSO parameters were: $c = 1$, $c_g = 2$, $w \sim \mathcal{U}(0.5, 1)$ and $v_{max} = 2$. The MFRSO parameters were: $\tau = 0.5$ and $N = 5$ (number of best particles). These values were inspired by the values used in [8].

This section is organized as follows: in Subsection A, an analysis of the parameter c in the performance of the QG-PSO is conducted. In Subsection B, the PSO,MFRPSO and QG-PSO are compared in the IK problem of 7-DoF and 15-DoF manipulators in environments without obstacles for different values of N. In Subsection C, the PSO,MFRPSO and QG-PSO are compared again in environments with obstacles. In Subsection D, both methods are evaluated for different values of the threshold ϵ.

A. ANALYSIS OF THE PARAMETER C

In order to study the impact of the parameter c on the QG-PSO performance, experiments were conducted where the QG-PSO was applied to the IK problem of the 7-DoF manipulator in an environment without obstacles with $c = 1$, 0.99, and 0.95. The results of these experiments are shown in the Table 3. From results, it can be concluded that decreasing the value of the c negatively impact the QG-PSO performance in terms of c_r. Since the best performance was found for $c = 1$, this value is used in the other experiments reported in the subsequent subsections.

TABLE 3. Evaluation of the impact of the parameter c on the QG-PSO performance.

Methods	N	c	$c_r(\%)$	f_t
QG-PSO	256	1	76.0	1726
QG-PSO	256	0.99	71.8	1591
QG-PSO	256	0.95	60.7	1606

TABLE 4. Comparison between QG-PSO, PSO and MFRPSO for different values of the number of particles (N) using the 7-DoF robot.

Methods	N	$c_r(\%)$	a_i	f_a	f_t
PSO	128	74.7	43	128	5462
MFRPSO	128	88.8	37	128	4732
QG-PSO	128	67.0	58	26	1533
PSO	256	81.9	34	256	8750
MFPSO	256	92.4	25	256	6501
QG-PSO	256	76.0	49	35	1726
PSO	512	88.2	29	512	15047
MFPSO	512	92.8	18	512	9194
QG-PSO	512	82.0	41	46	1879

TABLE 5. Comparison between QG-PSO, PSO and MFRPSO for different values of the number of particles (N) using the 15-DoF robot.

Métodos	N	$c_r(\%)$	a_i	f_a	f_t
PSO	128	96.0	46	128	5932
FRPSO	128	99.3	63	128	8116
QG-PSO	128	95.8	67	31	2079
PSO	256	98.5	36	256	9296
FRPSO	256	99.9	35	256	8844
QG-PSO	256	97.6	54	41	2227
PSO	512	99.6	32	512	16300
FRPSO	512	99.8	19	512	9942
QG-PSO	512	98.6	45	55	2473

B. EXPERIMENTS IN ENVIRONMENTS FREE OF OBSTACLES

The results of the experiments in environments without obstacles for different values of N are shown in Table 4 and Table 5 for 7-DoF and 15-DoF manipulators, respectively. In addition to the metrics c_r and f_t, the metrics a_i and f_a are also shown, where a_i is the average number of iterations and f_a is the quantity of direct kinematic calculations per iteration, obtained though ration f_t/a_i.

In the case of the experiments with the 7-DoF manipulator (Table 4), the PSO obtained higher values of c_r than the QG-PSO for each value of N tested. On the other hand, the QG-PSO obtained values of f_t approximately 4 to 8 times smaller than those of the PSO. In the results for the 15-DoF manipulator (Table 5), the QG-PSO again obtained smaller values of c_r than the PSO, but the difference between the values was smaller than in the case of the 7-DoF manipulator.

For the 15-DoF, the difference was smaller or equal to 1%, whereas for the 7-DoF, the worst case was a difference of 7.7% ($N = 128$). Furthermore, the QG-PSO was again less costly than the PSO in terms of f_t, obtaining values 3 to 7 times smaller. Thus, for both manipulators, although the QG-PSO was less efficient in terms of c_r, it was less costly in terms of f_t. Furthermore, the results may indicate that, for high-DoF manipulators, the QG-PSO is even more competitive relative to PSO, as the performance difference was smaller for the 15-DoF manipulator in terms of c_r.

Although the QG-PSO always had smaller values of c_r for the same value of N, it was able to outperform PSO by increasing its value of N while keeping the value of N fixed for the PSO on both manipulators. In the 7-DoF manipulator, for example, for $N = 128$, the PSO obtained $t_c = 74.7\%$ and $f_t = 5462$, whereas for $N = 256$, the QG-PSO obtained $t_c = 76.0\%$ and $f_t = 1726$. Thus, with a higher value of N, the QG-PSO simultaneously achieved a higher value of t_c and a smaller value of f_t than those of the PSO. This was possible because the cost of QG-PSO per iteration, f_a, is not equal to N as in PSO and MFRPSO. Instead, it is smaller than N due to the use of the QAA algorithm to reach q_{gsub}, which reduces the cost from N to a value proportional to N/M [21].

In both manipulator cases, the QG-PSO was not able to outperform MFRPSO in terms of c_r, even by increasing only its value of N. However, the QG-PSO was approximately 3 to 5 times less costly in terms of f_t.

C. EXPERIMENTS IN ENVIRONMENTS WITH OBSTACLES

In this subsection, the performances of the QG-PSO,PSO and MFRPSO are compared in an environment with obstacles in the IK problem of the 7-DoF and 15-DoF manipulators. The obstacles consist of 12 spheres around the robot with a radius of 25 mm. The coordinates of the robot's base are defined as (0,0,0), and the coordinates of the spheres are defined as: $(\pm 100, \pm 100, 100)$mm, $(\pm 100, 0, 200)$, $(0, \pm 100, 200)$, and $(\pm 100, \pm 100, 300)$mm, for the 7-DoF manipulator, while $(\pm 100, \pm 100, 200)$mm, $(\pm 100, 0, 400)$, $(0, \pm 100, 400)$, and $(\pm 100, \pm 100, 600)$mm, for the 15-DoF manipulator. In the collision detection, the links of the robots are modeled as cylinders with a radius of 25mm. This experimental setup is illustrated in Fig. 3 for the 7-DoF manipulator.

The results of the experiments with the 7-DoF and 15-DoF manipulators are shown in Table 6 and Table 7, respectively. The conclusions are similar to the case without obstacles, but with less statistical strength, since the experiments were repeated 10 times fewer. QG-PSO was always less costly than PSO and MFRPSO in terms of f_t, while achieving values smaller than or equal to those of PSO, and always smaller than those of MFRPSO. QG-PSO obtained f_t values approximately 3 to 8 times smaller than PSO, and approximately 3 to 5 times smaller than MFRPSO.

Similar to the experiments without obstacles, the performance of QG-PSO was more competitive compared to PSO and MFRPSO in terms of c_r with the 15-DoF manipulator. Furthermore, in both manipulator cases, QG-PSO was able to outperform PSO in terms of c_r by increasing its swarm size while keeping the same size for PSO, and simultaneously achieved a lower cost in terms of f_t. However, for the values of N tested, QG-PSO was not able to outperform MFRPSO in terms of c_r, even when increasing N. Thus, although the new formulation achieved a significant cost reduction, allowing an increase in N without a high impact on its cost, QG-PSO had inferior performance compared to MFRPSO in terms of c_r, while still being less costly.

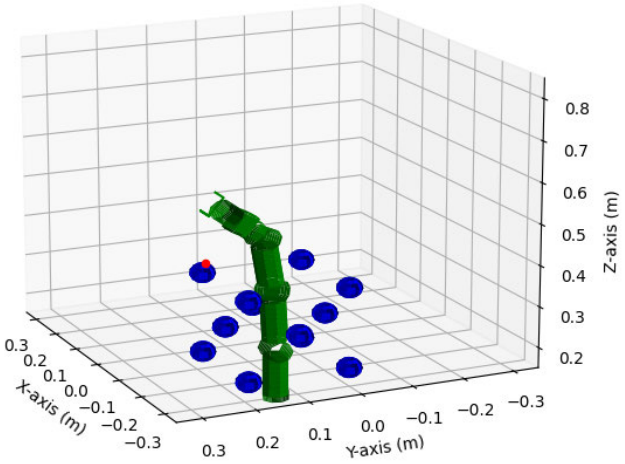


FIGURE 3. The 7-DOF robot in an environment with obstacles. The robot is in green, the obstacles (spheres) are in blue, and red is a possible desirable position.

TABLE 6. Comparison between QG-PSO, PSO and MFRPSO for different values of the number of particles(N) using the 7-DoF manipulator in the environment with obstacles.

Métodos	N	$c_r(\%)$	a_i	f_a	f_t
PSO	128	56.0	34	128	4366
MFRPSO	128	79.0	29	128	3764
QG-PSO	128	62.0	54	28	1488
PSO	256	84.0	29	256	14976
MFRPSO	256	89.0	20	256	10447
QG-PSO	256	73.0	44	47	2077
PSO	512	83.0	29	512	14626
MFRPSO	512	90.0	20	512	10138
QG-PSO	512	68.0	41	47	1936

TABLE 7. Comparison between QG-PSO, PSO and MFPRSO for different values of the number of particles(N) using the 15-DoF manipulator in the environment with obstacles.

Métodos	N	$c_r(\%)$	a_i	f_a	f_t
PSO	128	92.0	51	256	6510
MFRPSO	128	100.0	77	256	9874
QG-PSO	128	94.0	71	31	2183
PSO	256	97.0	38	256	9818
MFRPSO	256	100.0	45	256	11622
QG-PSO	256	97.0	56	51	2359
PSO	512	99.0	35	256	17899
MFRPSO	512	100.0	27	256	13983
QG-PSO	512	99.0	47	51	2681

D. ANALYSIS OF THE IMPACT OF THE THRESHOLD ϵ

In this subsection, the QG-PSO, PSO and the MFRPSO are compared in the IK problem of the 7-DoF manipulator considering different values of ϵ in an environment free of

TABLE 8. Comparison between QG-PSO,PSO and MFRPSO for different values of ϵ.

Methods	N	$\epsilon/L(\%)$	$c_r(\%)$	a_i	f_a	f_t
PSO	256	1	81.9	34	256	8750
MFPSO	256	1	92.4	25	256	6501
QG-PSO	256	1	76.0	49	35	1726
PSO	256	0.5	78.6	46	256	11751
MFRPSO	256	0.5	87.1	37	256	9467
QG-PSO	256	0.5	69.2	71	39	2751
PSO	256	0.1	71.5	69	256	17731
MFRPSO	256	0.1	76.4	67	256	17157
QG-PSO	256	0.1	56.6	118	42	4991

obstacles. The values of ϵ tested were ϵ =1%, 0.5%, and 0.1% of L, where L is the total length of the robot. The results of the experiments are shown in Table 8.

For both methods, decreasing the value of ϵ resulted in smaller values of c_r and higher values of f_t. Furthermore, for all values of ϵ tested, QG-PSO obtained smaller values of c_r. On the other hand, the cost of QG-PSO in terms of f_t was approximately 3.6 ($\epsilon = 0.1\%L$) to 5.1 ($\epsilon = 1\%L$) times smaller than that of PSO, and 3.4 ($\epsilon = 0.1\%L$) to 3.8 ($\epsilon = 1\%L$) times smaller than that of MFRPSO. Thus, the performance gain of QG-PSO in terms of f_t is negatively affected by a decrease in the value of ϵ.

VII. LIMITATION

Although the results of QG-PSO performance are promising, this study has limitations. The main limitation, which stems from the current constraints of the quantum computing field, is that QG-PSO was not implemented on a quantum computer. Therefore, its performance was analyzed only from a theoretical perspective through classical simulations, as discussed in Section V. Consequently, it was not possible to directly measure the gain in terms of time required to reach a solution. Instead, the focus was on the number of forward kinematics calculations required to reach a solution. This should translate into a performance gain in terms of time, assuming that the time to compute forward kinematics on quantum hardware is at least approximately equal to that on a classical computer. However, since quantum technology capable of implementing high-level algorithms is not yet available, it was not possible to perform this analysis. The actual computation time would depend on the implementation of the quantum hardware and could vary depending on the physical representation of the qubits.

Thus, because of current technological restrictions, only a theoretical performance gain was observable in this paper, in terms of the reduction of forward kinematics calculations necessary to solve the IK problem.

VIII. CONCLUSION

In this work, a new quantum variant of PSO is explored. PSO is a widely-used optimization method in various engineering problems, particularly in the Inverse Kinematics (IK) problem. The proposed method leverages the quantum algorithm Quantum Amplitude Amplification (QAA) to search for a suboptimal particle rather than the best particle, achieving a lower computational cost. This reduction is made possible through quantum phenomena such as superposition and entanglement.

It is important to emphasize some differences between the Quantum-Grover PSO and the Quantum Behaved PSO explored in [19]. The latter is only a classical method inspired by the movement of quantum particles, but it does not use any quantum phenomena in its computation, such as superposition or entanglement, and thus does not require quantum hardware. In contrast, the Quantum-Grover PSO explored in this work requires quantum hardware to achieve the performance improvements discussed in this paper in a real scenario.

The performance analysis of Quantum-Grover PSO was conducted by comparing it with classical PSO and its variant, MFRPSO, in the IK problem of manipulator robots, considering both position and orientation correction. The performance of the methods was measured in terms of convergence rate (c_r) and the total quantity of direct kinematic calculations (f_t) required for the method to converge to a desirable solution. The convergence rate c_r quantifies the probability of the algorithm converging to the desirable solution, and f_t represents the cost to obtain the solution in terms of direct kinematic calculations. The experiments were performed on two manipulators, one with 7-DoF and the other with 15-DoF, in environments with and without obstacles.

In the experiments without obstacles, it was observed that for both robots, when the methods used the same population size, the Quantum-Grover PSO obtained smaller values of c_r, on the other hand, it obtained values of f_t approximately 4 to 7 times smaller than those of the PSO. Thus, in this situation, the Quantum-Grover PSO was less costly, but converged fewer times to the desirable solutions. Furthermore, it also was observed that, fixing the PSO population size and increasing the Quantum-Grover PSO population size, the Quantum-Grover PSO could obtain higher values of c_r and smaller values of f_t than the Classical PSO. For example, in the case of the 7-DoF manipulator, setting $N = 128$ for classical PSO and $N = 256$ for Quantum-Grover PSO, this obtained $c_r = 76.0\%$ and $f_t = 1726$, while classical PSO obtained $c_r = 74.7\%$ and $f_t = 5462$. In the same way, in the case of the 15-Dof manipulator, fixing $N = 128$ for classical PSO and $N = 256$ for Quantum-Grover PSO, this obtained $c_r = 97.6\%$ and $f_t = 2227$, while classical PSO obtained $c_r = 96.0\%$ and $f_t = 5932$. However, for the values of N tested, even when increasing only the value of N for QG-PSO, it outperformed MFRPSO only in terms of f_t, but was not able to outperform it in terms of c_r.

Still in the experiments without obstacles, an analysis was conducted on the effect of the parameter c on the Quantum-Grover PSO performance. This parameter belongs to the boolean function (f_o) used by the algorithm QAA to

search for the best particle, and its range of operation is (0, 1]. In the experiments, it was observed that decreasing the value of c negatively impacted the Quantum-Grover PSO performance in terms of c_r. For $c = 1$, the method achieved the highest value of c_r.

Another study conducted in the experiments without obstacles focused on the performance of both methods for different values of the threshold of convergence (ϵ). The smaller the values of ϵ, the more difficult it is to solve the IK problem. In the experiments, it was observed that the smaller the value of ϵ the smaller the performance gain of the Quantum-Grover PSO in relation to classical PSO in terms of f_t.

In the experiments with obstacles, similar aspects were observed as in the experiments without obstacles. When fixing the same population size for both methods, the Quantum-Grover PSO was again less costly than the classical PSO in terms of f_t. Additionally, except for $N = 128$, classical PSO obtained a higher value of c_r for the 7-DoF manipulator. However, for the 15-DoF manipulator, the methods obtained equal values of c_r for $N = 256$ and 512, with Quantum-Grover PSO achieving a c_r value 2% higher for $N = 128$. In comparison to MFRPSO, for the values of N tested, QG-PSO outperformed MFRPSO only in terms of f_t, but was not able to outperform it in terms of c_r, even when increasing the value of N.

Therefore, in the experiments with and without obstacles, the Quantum-Grover PSO demonstrated a theoretical less costly performance than the classical PSO. On the other hand, in most experiments, the Quantum-Grover PSO obtained lower success rates. There are at least two causes that jointly contribute to this. The first is that in every iteration the classical PSO chooses the best particle to guide the other particles, while in the Quantum-Grover PSO may choose a suboptimal particle as long as it satisfies the boolean function of the QAA algorithm. The other cause is that the classical PSO updates the particles position using both global and local information, while the Quantum-Grover PSO uses only the global information. These two aspects were imposed in the Quantum-Grover PSO formulation to enable the use of the algorithm QAA. Nevertheless, although the classical PSO outperforms the Quantum-Grover PSO for the same population size in terms of c_r, it was observed in some experiments that increasing the population size for the Quantum-Grover PSO made it simultaneously achieved higher values of c_r and lower values of f_t in relation to classical PSO. It was also observed that, for the 15-DoF manipulator, the performance of QG-PSO was more competitive in comparison with its performance on the 7-DoF manipulator, which may indicate that it is more competitive for high-DoF manipulators.

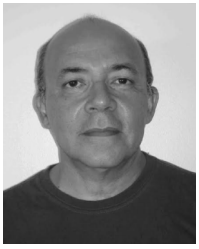
A direction for future works is to explore the QAA algorithm in other PSO variants, especially in those which are less dependent on local information. Furthermore, other Boolean functions that could enhance the performance of Quantum-Grover PSO may be investigated.

REFERENCES

[1] P. Fiorini, K. Y. Goldberg, Y. Liu, and R. H. Taylor, "Concepts and trends in autonomy for robot-assisted surgery," *Proc. IEEE*, vol. 110, no. 7, pp. 993–1011, Jul. 2022.
[2] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Modeling and Control*. Hoboken, NJ, USA: Wiley, 2020.
[3] F. Maric, M. Giamou, A. W. Hall, S. Khoubyarian, I. Petrovic, and J. Kelly, "Riemannian optimization for distance-geometric inverse kinematics," *IEEE Trans. Robot.*, vol. 38, no. 3, pp. 1703–1722, Jun. 2022.
[4] S. Lloyd, R. A. Irani, and M. Ahmadi, "Fast and robust inverse kinematics of serial robots using Halley's method," *IEEE Trans. Robot.*, vol. 38, no. 5, pp. 2768–2780, Oct. 2022.
[5] A. Yonezawa, H. Yonezawa, and I. Kajiwara, "Simple inverse kinematics computation considering joint motion efficiency," *IEEE Trans. Cybern.*, vol. 54, no. 9, pp. 4903–4914, Sep. 2024.
[6] D. O. Santos, R. C. S. Freire, E. A. N. Carvalho, M. C. Santos, L. Molina, J. G. N. Carvalho, and E. O. Freire, "An extension of the MB-RRT method to solve the inverse kinematics problem for real robot manipulators," in *Proc. Brazilian Conf. Robot. (CROS)*, Apr. 2025, pp. 1–6.
[7] O. Limoyo, F. Marić, M. Giamou, P. Alexson, I. Petrović, and J. Kelly, "Generative graphical inverse kinematics," *IEEE Trans. Robot.*, vol. 41, pp. 1002–1018, 2025.
[8] D. O. Santos, L. Molina, J. G. N. Carvalho, E. A. N. Carvalho, and E. O. Freire, "Modifications of fully resampled PSO in the inverse kinematics of robot manipulators," *IEEE Robot. Autom. Lett.*, vol. 9, no. 2, pp. 1923–1928, Feb. 2024.
[9] L. A. Orbegoso Moreno, D. Valverde Ramírez, M. Pasco Sánchez, and L. Ruiz Rodríguez, "Comparative study of iterative methods for inverse kinematics of redundant serial robots with increasing degrees of freedom," in *Proc. IEEE Int. Autumn Meeting Power, Electron. Comput. (ROPEC)*, Oct. 2023, pp. 1–6.
[10] J. Colan, A. Davila, and Y. Hasegawa, "Variable step sizes for iterative jacobian-based inverse kinematics of robotic manipulators," *IEEE Access*, vol. 12, pp. 87909–87922, 2024.
[11] M. C. Santos, L. Molina, E. A. N. Carvalho, E. O. Freire, J. G. N. Carvalho, and P. C. Santos, "FABRIK-R: An extension developed based on FABRIK for robotics manipulators," *IEEE Access*, vol. 9, pp. 53423–53435, 2021.
[12] S. Momani, Z. S. Abo-Hammou, and O. MK. Alsmad, "Solution of inverse kinematics problem using genetic algorithms," *Appl. Math. Inf. Sci.*, vol. 10, no. 1, pp. 225–233, Jan. 2016.
[13] L. Yiyang, J. Xi, B. Hongfei, W. Zhining, and S. Liangliang, "A general robot inverse kinematics solution method based on improved PSO algorithm," *IEEE Access*, vol. 9, pp. 32341–32350, 2021.
[14] F. Liu, H. Huang, B. Li, and F. Xi, "A parallel learning particle swarm optimizer for inverse kinematics of robotic manipulator," *Int. J. Intell. Syst.*, vol. 36, no. 10, pp. 6101–6132, Oct. 2021.
[15] N. Rokbani, M. Slim, and A. M. Alimi, "The beta distributed PSO, ß-PSO, with application to inverse kinematics," in *Proc. Nat. Comput. Colleges Conf. (NCCC)*, Mar. 2021, pp. 1–6.
[16] H. Deng and C. Xie, "An improved particle swarm optimization algorithm for inverse kinematics solution of multi-DOF serial robotic manipulators," *Soft Comput.*, vol. 25, no. 21, pp. 13695–13708, Nov. 2021.
[17] I. Sancaktar, B. Tuna, and M. Ulutas, "Inverse kinematics application on medical robot using adapted PSO method," *Eng. Sci. Technol., Int. J.*, vol. 21, no. 5, pp. 1006–1010, Oct. 2018.
[18] T. J. Collinsm and W.-M. Shen, "Particle swarm optimization for high-DOF inverse kinematics," in *Proc. 3rd Int. Conf. Control, Autom. Robot. (ICCAR)*, Apr. 2017, pp. 1–6.
[19] S. Dereli and R. Köker, "A meta-heuristic proposal for inverse kinematics solution of 7-DOF serial robotic manipulator: Quantum behaved particle swarm algorithm," *Artif. Intell. Rev.*, vol. 53, no. 2, pp. 949–964, Feb. 2020.
[20] K. Hu, Z. Ma, S. Zou, J. Li, and J. Zhang, "Game-based social learning particle swarm optimizer for inverse kinematics of robotic arms," *IEEE Robot. Autom. Lett.*, vol. 10, no. 7, pp. 7078–7085, Jul. 2025.
[21] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*. Cambridge, U.K.: Cambridge Univ. Press, 2010.
[22] R. Van Meter and S. J. Devitt, "The path to scalable distributed quantum computing," *Computer*, vol. 49, no. 9, pp. 31–42, Sep. 2016.
[23] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM Rev.*, vol. 41, no. 2, pp. 303–332, Jan. 1999.
[24] L. K. Grover, "Quantum mechanics helps in searching for a needle in a haystack," *Phys. Rev. Lett.*, vol. 79, no. 2, pp. 325–328, Jul. 1997.

[25] M. Mosca, "Counting by quantum eigenvalue estimation," *Theor. Comput. Sci.*, vol. 264, no. 1, pp. 139–153, Aug. 2001.
[26] M. Boyer, G. Brassard, P. Høyer, and A. Tapp, "Tight bounds on quantum searching," *Fortschritte Der Physik*, vol. 46, nos. 4–5, pp. 493–505, Jun. 1998.
[27] D. Dong, C. Chen, H. Li, and T.-J. Tarn, "Quantum reinforcement learning," *IEEE Trans. Syst., Man, Cybern., B Cybern.*, vol. 38, no. 5, pp. 1207–1220, May 2008.
[28] G. Acampora, R. Schiattarella, and A. Vitiello, "Using quantum amplitude amplification in genetic algorithms," *Expert Syst. Appl.*, vol. 209, Dec. 2022, Art. no. 118203.
[29] P. Lathrop, B. Boardman, and S. Martínez, "Quantum search approaches to sampling-based motion planning," *IEEE Access*, vol. 11, pp. 89506–89519, 2023.
[30] P. Lathrop, B. Boardman, and S. Martínez, "Parallel quantum rapidly-exploring random trees," *IEEE Access*, vol. 12, pp. 47173–47189, 2024.
[31] M. S. Dahassa and N. Zioui, "Optimal control-based Grover's algorithm for a six-jointed articulated robotic arm," *Electronics*, vol. 14, no. 13, p. 2503, Jun. 2025.
[32] C. Petschnigg, M. Brandstötter, H. Pichler, M. Hofbaur, and B. Dieber, "Quantum computation in robotic science and applications," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 803–810.
[33] D. Oliveira. (2025). *Quantum-Grover-PSO*. Accessed: Jul. 14, 2025. [Online]. Available: https://github.com/GPRUFS/Quantum-Grover-PSO

DAVID O. SANTOS received the bachelor's degree in electronic engineering from the Federal University of Sergipe, in 2023, and the master's degree in electrical engineering from the Federal University of Campina Grande, in 2024, where he is currently pursuing the Ph.D. degree. He is also a Volunteer Professor with the Federal University of Sergipe and a Researcher with the Robotics Research Group, Federal University of Sergipe. His research interests include manipulator robotics, mobile robotics, and pattern recognition.



FRANCISCO M. DE ASSIS received the B.Sc. and M.Sc. degrees in electrical engineering from the Military Institute of Engineering, Rio de Janeiro, Brazil, in 1984 and 1992, respectively, and the Ph.D. degree from the Pontifical Catholic University of Rio de Janeiro, Rio de Janeiro, in 1994. He is currently a Professor with the Federal University of Campina Grande, Campina Grande, Brazil. His research interests include coding and information theory.

ELYSON A. N. CARVALHO received the bachelor's degree in electronic engineering from the Federal University of Sergipe, in 2006, and the master's and Ph.D. degrees in electrical engineering from the Federal University of Campina Grande, in 2007 and 2012, respectively. Currently, he is an Associate Professor III with the Federal University of Sergipe and a Permanent Member with the Graduate Program in Electrical Engineering (PROEE).

LUCAS MOLINA received the bachelor's degree in electronic engineering from the Federal University of Sergipe (UFS), in 2007, the master's degree in control, automation, and robotics from the Federal University of Rio de Janeiro, in 2010, and the Ph.D. degree in electrical engineering from the Federal University of Campina Grande, in 2014. He is currently an Adjunct Professor with UFS and a Researcher with the Robotics Research Group of UFS (GPR-UFS).

Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) - ROR identifier: 00x0ma614