

A Novel Quantum-Computing-Based Quaternions Model for a Robotic Arm Position

Nadjet Zioui

Université du Québec à Trois-Rivières & Institut de Recherche sur l'Hydrogène, 3351 Boulevard des Forges, Trois-Rivières, QC G8Z 4M3, Canada, nadjet.zioui@uqtr.ca

Yousra Mahmoudi

Laboratoire RECITS, Université de la Science et de la Technologie Houari Boumediene, BP 32 Bab Ezzouar, 16111, Algiers, Algeria & Département Hydraulique Urbaine, École Nationale Supérieure d'Hydraulique Arbaoui Abdallah, 29 route de Soumaâ, Blida, Algeria & Université du Québec à Trois-Rivières, 3351 Boulevard des Forges, Trois-Rivières, QC G8Z 4M3, Canada, yousra.mahmoudi@uqtr.ca

Aïcha Mahmoudi

Projet Véo, Université de Sherbrooke, 2500 Boulevard de l'Université, Sherbrooke, QC J1K 2R1, Canada, aïcha.mahmoudi@usherbrooke.ca

Mohamed Tadjine

Laboratoire de Commande des Processus, École Nationale Polytechnique, 10 Rue des Frères OUDEK, El Harrach 16200, Algiers, Algeria, mohamed.tadjine@g.enp.edu.dz

Said Bentouba

Faculty of Engineering and Science, Western Norway University of Applied Sciences Bergen, 5063, Norway, said.bentouba@hvl.no

Date of Submission: 03rd June 2021 Revised: 06th August 2021 Accepted: 12th August 2021

How to Cite: Zioui, N., Mahmoudi, Y., Mahmoudi, A., Tadjine, M., Bentouba, S. (2021). A novel quantum-computing-based quaternions model for a robotic arm position. *International Journal of Computational Intelligence in Control* 13(2), pp.71-77.

Abstract - This paper presents a novel model for describing the position of robotic arms using quantum-computing tools.

Because of the successful use and implementation of the quaternions in the rigid bodies and robotic arms' kinematics modelling, this research examines the equivalence between the quaternions and the Pauli matrices in the context of modelling the robotic arm's position. Consequently, the paper proposes a model based on the established equivalence between the quaternions and the Pauli gates through the formulation of the quaternions in the 2-D matrix representation. This equivalence offers the possibility of defining quantum circuits and algorithms for modelling robotic arm movement.

As an illustrative example, the paper presents a SCARA robotic arm model using this new technique. The simulation results underline that the classic methods and the quantum-based model appear to be equivalent in vector spaces and emphasize the effectiveness of the proposed method.

As a main result, the quantum model of the position of the robotic arm can be implemented using only one qubit, which brings compactness to the proposed model. Unlike the classic computers that compute the position models using successive rotations and translations, involving many multiplications and additions operations, including many bytes, depending on the machine's resolution, this model is very important as it contributes to the reduction of the computing resources for the future quantum computers.

Index Terms - Quantum computing, Quaternion, Pauli matrices, Robotic arm, Position's model.

INTRODUCTION

Quantum computing is a rapidly expanding field of research that is expected to revolutionize computers and software in the near future. New concepts are being developed all around the world to support the engineering of practical quantum computers. Most of the progress achieved

so far has focused on control, electrical and electronics engineering related concepts, namely the quantum Laplace transform [1, 2] and Fourier transform [3], [4], solving systems of linear differential equations [5-12] and optimization based on the Grover algorithm [13, 14]. The field remains replete with opportunities for innovation as well as improvement and optimization of the tools developed to date [15, 16].

The quantum technological revolution is of special interest in mechatronics-related fields and robotics. Robotic arm kinematics models express the spatial position and orientation of an end-effector relative to the base frame using the angular status of joints, and vice versa. In forward models, Denavit-Hartenberg formalism is used to establish joint frames and determine the homogenous matrices that allow pas-sage from one joint variable frame to the next until the end-effector frame is reached [17-22]. The product of these matrices yields the Cartesian x-y-z positional function of the joint variables. Another recent approach to robotic arm position and orientation modelling is to use quaternions, single and dual [23-28]. To the best of our knowledge, no quantum-computing-based model of industrial robotic arms has been published, nor has the quaternions based Pauli matrices model for arm modelling purposes.

As a first step in this direction, we examine a quantum forward kinematics model of a robotic arm, implemented successfully based on results that match those of classical approaches, namely quaternions and Denavit-Hartenberg formalism. We feel that this represents a major step forward in robotics and in solid body quantum modelling and simulation in general.

MATERIALS AND METHODS

1. Articulated robotic arm modelling

In the sections below, we present the quantum-tool-based quaternions approach to robotic arm kinematics modelling and introduce the equivalence between quaternions and quantum computing formalism based on Pauli matrices.

- **Quaternions:** Discovered by Sir William Hamilton in 1843 [28], quaternions are hyper-complex numbers that have a real part and three imaginary parts. A typical quaternion can be written in a unique form where r is the real part and x , y and z the imaginary parts.

$$q = r + xi + yj + zk \tag{1}$$

i, j and k are pure imaginary numbers that satisfy the condition

$$i^2 = j^2 = k^2 = -1 \tag{2}$$

Numbers i, j and k also satisfy the following conditions:

$$\begin{cases} ij = k \\ jk = i \\ ki = j \\ ji = -k \\ kj = -i \\ ik = -j \end{cases} \tag{3}$$

A three-dimensional vector $v(x, y, z)$ can be expressed as a pure imaginary quaternion:

$$q = xi + yj + zk \tag{4}$$

The conjugate of which can be defined as follows:

$$q^* = r - xi - yj - zk \tag{5}$$

A unit quaternion has a unit norm $\|q\| = 1$, given the quaternion's norm:

$$\|q\| = \sqrt{r^2 + x^2 + y^2 + z^2} \tag{6}$$

Given this definition of quaternion norm, the inverse of a quaternion is computed as:

$$q^{-1} = \frac{q^*}{\|q\|} \tag{7}$$

A quaternion in a (4x4) dimension space can be represented as follows:

$$q = \begin{pmatrix} r & -x & -y & -z \\ x & r & -z & -y \\ y & z & r & -x \\ z & -yx & r & \end{pmatrix} \tag{8}$$

A quaternion can also be represented in (2x2) dimension space as follows:

$$q = \begin{pmatrix} r + xi & -y - zi \\ y - zi & r - xi \end{pmatrix} \tag{9}$$

Which can be linked directly to the quantum domain using the new entities, I_x, I_y and I_z

$$q = rI + xI_x + yI_y + zI_z \tag{10}$$

Where I is the (2 x 2) identity matrix $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ and I_x, I_y and I_z are defined as follows:

$$I_x = \begin{pmatrix} -i & 0 \\ 0 & i \end{pmatrix}, I_y = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}, I_z = \begin{pmatrix} 0 & -i \\ -i & 0 \end{pmatrix}$$

The product of two quaternions $q_1 = r_1 + x_1i + y_1j + z_1k$ and $q_2 = r_2 + x_2i + y_2j + z_2k$ can be expressed as follows:

$$q_1q_2 = r_{12} + x_{12}i + y_{12}j + z_{12}k \tag{11}$$

The product of two quaternions is another quaternion, with the real part r_{12} and the imaginary parts x_{12}, y_{12} and z_{12} defined as follows:

$$\begin{aligned} r_{12} &= r_1r_2 - x_1x_2 - y_1y_2 - z_1z_2 \\ x_{12} &= x_1r_2 + r_1x_2 - z_1y_2 - y_1z_2 \\ y_{12} &= y_1r_2 + z_1x_2 + r_1y_2 - x_1z_2 \\ z_{12} &= z_1r_2 - y_1x_2 + x_1y_2 + r_1z_2 \end{aligned}$$

For computing purposes, it is more convenient to represent the quaternion product as matrix product. Hence, a quaternion q_1 can be represented in (4x4) dimension space by the matrix M_1 . Therefore, the quaternions product can be expressed as follows

$$q_1q_2 = M_1Q_1 \tag{12}$$

Where

$$M_1 = \begin{pmatrix} r_1 & -x_1 - y_1 & -z_1 \\ x_1 & r_1 & -y_1 \\ y_1 & z_1 & r_1 & -x_1 \\ z_1 & -y_1 x_1 & r_1 \end{pmatrix} \text{ and } Q_1 = \begin{pmatrix} r_2 \\ x_2 \\ y_2 \\ z_2 \end{pmatrix}$$

- **Expressing positions and rotations using quaternions:** An angle θ rotation around a unit vector $u = u_x i + u_y j + u_z k \sqrt{u_x^2 + u_y^2 + u_z^2} = 1$ is given by the quaternion defined below:

$$q_R = \cos\left(\frac{\theta}{2}\right) + \sin\left(\frac{\theta}{2}\right)u = \cos\left(\frac{\theta}{2}\right) + \sin\left(\frac{\theta}{2}\right)(u_x i + u_y j + u_z k) \quad (13)$$

An angle θ rotation of a vector $v = xi + yj + zk$ around u is given by the sandwich product $q_R v q_R^*$ where the conjugate of q_R

$$q_R^* = \cos\left(\frac{\theta}{2}\right) - \sin\left(\frac{\theta}{2}\right)u = \cos\left(\frac{\theta}{2}\right) + \sin\left(-\frac{\theta}{2}\right)u \quad (14)$$

Since q_R is a unit quaternion, its conjugate is also its inverse.

Translation from an initial position q_i to a final position q_f can be represented as quaternion $q_T = x_T i + y_T j + z_T k$, meaning that:

$$q_f = q_i + q_T \quad (15)$$

Therefore, if a vector v_i rotates and then translates, the transformation that yields the final value v_f can be expressed as follows:

$$v_f = q_T + q_R v_i q_R^* \quad (16)$$

II. Quantum computing's basics

Quantum computing rules and tools depend mainly on quantum mechanics and concepts reasoned by Dirac [31]. The definitions in the sections that follow will establish the equivalence between quantum computing principles and quaternion representations [29-31].

- **The qubit:** The qubit is the basic element of information in quantum computing. Unlike the classic bit, which exists in state 0 or 1 with 100% probability, a qubit's state, denoted by $|\varphi\rangle$ ("ket phi"), has the following expression:

$$|\varphi\rangle = \alpha|0\rangle + \beta|1\rangle \quad (17)$$

Where α and β are complex numbers such that $|\alpha|^2 + |\beta|^2 = 1$. Some authors regard $|\alpha|^2$ as the probability that the qubit state is $|0\rangle$ and $|\beta|^2$ as the probability that its state is $|1\rangle$.

Superposition is a linear combination of two qubit states. The relationship (17) is obtained from the principle of superposition. It may be thought of as the possibilities for the qubit state to be $|0\rangle$ or $|1\rangle$ with their respective probabilities of $|\alpha|^2$ and $|\beta|^2$.

The qubit state can also be expressed as a vector, as illustrated in Figure 1, also known as the Hilbert space. In the Dirac two-dimensional vector space, the qubit state is expressed as the following vector:

$$|\varphi\rangle = \begin{pmatrix} \varphi_1 \\ \varphi_2 \end{pmatrix} = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \quad (18)$$

Where $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$ denotes the qubit's state $|0\rangle$ and $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$ denotes state $|1\rangle$.

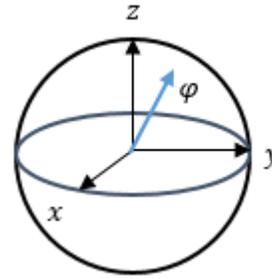


FIGURE 1
SPHERICAL REPRESENTATION OF A QUBIT φ (KNOWN AS THE BLOCH SPHERE).

- **The joint state of a system of qubits:** The joint state of two qubit states $|\varphi\rangle = \alpha|0\rangle + \beta|1\rangle$ and $|\psi\rangle = \alpha'|0\rangle + \beta'|1\rangle$ may be thought of as the mean of the tensor product operation on the two states, which can be computed as follows:

$$|\varphi\psi\rangle = |\varphi\rangle \otimes |\psi\rangle = \alpha\alpha'|00\rangle + \alpha\beta'|01\rangle + \beta\alpha'|10\rangle + \beta\beta'|11\rangle \quad (19)$$

The joint state of three qubits $|\varphi_1\rangle = \alpha_1|0\rangle + \beta_1|1\rangle$, $|\varphi_2\rangle = \alpha_2|0\rangle + \beta_2|1\rangle$ and $|\varphi_3\rangle = \alpha_3|0\rangle + \beta_3|1\rangle$ likewise can be computed as follows:

$$|\varphi_1\varphi_2\varphi_3\rangle = |\varphi_1\rangle \otimes |\varphi_2\rangle \otimes |\varphi_3\rangle = \alpha_1\alpha_2\alpha_3|000\rangle + \alpha_1\alpha_2\beta_3|001\rangle + \alpha_1\beta_2\alpha_3|010\rangle + \alpha_1\beta_2\beta_3|011\rangle + \beta_1\alpha_2\alpha_3|100\rangle + \beta_1\alpha_2\beta_3|101\rangle + \beta_1\beta_2\alpha_3|110\rangle + \beta_1\beta_2\beta_3|111\rangle \quad (20)$$

Once normalized, the result of the operation should be a valid qubit state. A qubit state thus can be expressed as a linear combination of basic states.

- **Measuring a qubit:** The inner product of two qubit states $|\varphi\rangle = \alpha|0\rangle + \beta|1\rangle$ and $|\psi\rangle = \alpha'|0\rangle + \beta'|1\rangle$ is a complex number given by the following operation:

$$\langle\varphi|\psi\rangle = \alpha'^*\alpha + \beta'^*\beta \quad (21)$$

Where α'^* and β'^* are the complex conjugates of α' and β' . The inner product of a qubit state and itself $\langle\varphi|\varphi\rangle$ yields the number $Re(\alpha)^2 + Re(\beta)^2$. The inner product of a state $|0\rangle$ or $|1\rangle$ with a qubit state $|\varphi\rangle$ will give the corresponding coefficient α or β .

The outer product of two qubit states yields a matrix given by the inner products:

$$|\varphi\rangle\langle\psi| = \begin{pmatrix} \alpha'\alpha^* & \alpha'\beta^* \\ \beta'\alpha^* & \beta'\beta^* \end{pmatrix} \quad (22)$$

It is possible to measure a qubit by reading the information stored within it or querying it to see if its status is 0 or 1. Many researchers view the result in terms of probability, meaning that measuring the qubit defined in the equation (17) will give the state $|0\rangle$ with the probability $|\alpha|^2$ or the state $|1\rangle$ with the probability $|\beta|^2$. One can obtain these measurements in the projection forms $|\langle 0|\varphi\rangle|^2$ and $|\langle 1|\varphi\rangle|^2$, respectively.

• **The most common gates and operators:**

The X gate:The X gate is the quantum version of the NOT gate. The operator X defined below is its matrix representation.

$$X \equiv \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} i.e. X = |0\rangle\langle 1| + |1\rangle\langle 0| \quad (23)$$

The Y gate: The Y gate matrix representation and operator are defined as follows:

$$Y \equiv \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \quad (24)$$

$$Y = -i|0\rangle\langle 1| + i|1\rangle\langle 0| \quad (25)$$

i is the imaginary number $\sqrt{-1}$.

The Z gate: The Z gate matrix representation and operator are defined as follows:

$$Z = |0\rangle\langle 0| - |1\rangle\langle 1| \quad (26)$$

$$Z \equiv \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \quad (27)$$

The rotation gates or basic spins: Three rotation gates are used in quantum computing: $R_x(\theta)$, $R_y(\theta)$ and $R_z(\theta)$. Also called quantum spins, they represent the simple rotations around the *x*, *y* and *z* axes, defined in matrix form as follows:

$$R_x(\theta) = \begin{pmatrix} C\frac{\theta}{2} & -iS\frac{\theta}{2} \\ -iS\frac{\theta}{2} & C\frac{\theta}{2} \end{pmatrix} \quad (28)$$

$$R_y(\theta) = \begin{pmatrix} C\frac{\theta}{2} & -S\frac{\theta}{2} \\ S\frac{\theta}{2} & C\frac{\theta}{2} \end{pmatrix} \quad (29)$$

$$R_z(\theta) = \begin{pmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{pmatrix} \quad (30)$$

Quantum circuits:Circuits often represent the quantum algorithms. Horizontal lines symbolise the qubits and rectangular symbols denote the gates that act on the line from left to right. The initial state of the qubit appears at the left. Figure 2 shows a circuit representation of a Y gate.

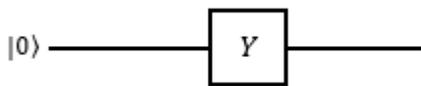


FIGURE 2
EXAMPLE OF A QUANTUM CIRCUIT.

Pauli Hermitian matrices: Pauli matrices are useful for defining quantum operators' frames as follows:

$$\sigma_0 = I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \sigma_1 = \sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \sigma_2 = \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \sigma_3 = \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \quad (31)$$

Every operator in quantum computing can be defined as a linear combination using these Hamiltonian matrices. For example, operators $R_x(\theta)$, $R_y(\theta)$ and $R_z(\theta)$ can be written using σ_0 , σ_x , σ_y and σ_z as follows:

$$R_x(\theta) = C\left(\frac{\theta}{2}\right)\sigma_0 - iS\left(\frac{\theta}{2}\right)\sigma_x \quad (32)$$

$$R_y(\theta) = C\left(\frac{\theta}{2}\right)\sigma_0 - iS\left(\frac{\theta}{2}\right)\sigma_y \quad (33)$$

$$R_z(\theta) = C\left(\frac{\theta}{2}\right)\sigma_0 - iS\left(\frac{\theta}{2}\right)\sigma_z \quad (34)$$

On close examination, Pauli matrices σ_x , σ_y and σ_z are none other than the matrix representations of the X, Y and Z gates defined above in equations (23), (25) and (27). The Pauli matrices possess the following properties:

$$\begin{cases} \sigma_j^2 = \sigma_0, j = 1,2,3 \\ \sigma_x\sigma_y = -\sigma_y\sigma_x = i\sigma_z \\ \sigma_y\sigma_z = -\sigma_z\sigma_y = i\sigma_x \\ \sigma_z\sigma_x = -\sigma_x\sigma_z = i\sigma_y \end{cases} \quad (35)$$

The first property supports the fundamental principle of quantum gate reversibility. In fact, quantum operators have strong reversibility properties, meaning that when applied twice and successively, a quantum operator will restore the qubit to its initial state, which is not always possible to obtain using classical logic gates.

METHODOLOGY

I. Expressing quaternions using Pauli matrices

The matrix representation of a quaternion as defined above (10) can be formulated using Pauli matrices, considering the relations defined below:

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = \sigma_0 \quad (36)$$

$$I_x = \begin{pmatrix} -i & 0 \\ 0 & i \end{pmatrix} = -i\sigma_z \Rightarrow \sigma_z = iI_x \quad (37)$$

$$I_y = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} = -i\sigma_y \Rightarrow \sigma_y = iI_y \quad (38)$$

$$I_z = \begin{pmatrix} 0 & -i \\ -i & 0 \end{pmatrix} = -i\sigma_x \Rightarrow \sigma_x = iI_z \quad (39)$$

A quaternion therefore can be expressed in 2D Hilbert space using Pauli matrices and thus computed using the Pauli gates X, Y and Z, as follows:

$$q = r\sigma_0 - i(x\sigma_x + y\sigma_y + z\sigma_z) = rI + xI_x + yI_y + zI_z \quad (40)$$

The entities I_x , I_y and I_z represent a new basis possessing the following properties:

$$\begin{cases} I_x^2 = I_y^2 = I_z^2 = -I \\ I_xI_y = -I_yI_x = -I_z \\ I_yI_z = -I_zI_y = -I_x \\ I_zI_x = -I_xI_z = -I_y \end{cases} \quad (41)$$

Indeed, considering the expression of a quaternion in (2x2) space (equation 10) and the properties set forth in definition (3) as well as the Pauli matrix properties (35), the following relations readily become apparent:

$$\begin{cases} I_x^2 = I_y^2 = I_z^2 = (i\sigma_x)^2 = (i\sigma_y)^2 = (i\sigma_z)^2 = -I \\ I_xI_y = i\sigma_z \cdot i\sigma_y = -\sigma_z \cdot \sigma_y = \sigma_y \cdot \sigma_z = i\sigma_x = -I_yI_x = -I_z \\ I_yI_z = i\sigma_x \cdot i\sigma_z = -\sigma_y \cdot \sigma_x = \sigma_x \cdot \sigma_y = i\sigma_z = -I_zI_y = -I_x \\ I_zI_x = i\sigma_x \cdot i\sigma_z = -\sigma_x \cdot \sigma_z = \sigma_z \cdot \sigma_x = i\sigma_y = -I_xI_z = -I_y \end{cases} \quad (42)$$

Therefore, the quantum-computing-based quaternions model for robotic arm position can be established by converting the equivalent Pauli gate model to quaternions.

I. Main result

Determining the spatial orientation and end effector position of an articulated robotic arm relative to the base frame requires accurate and precise quantitation of successive rotations and translations, which has been achieved previously using quaternions.

Based on the definitions of rotation and translation quaternions, simple rotations, and translations around and along the axes x, y and z are expressed using the following quaternions and hence quantum operators:

$$\begin{cases} q_{Rx} = \cos\left(\frac{\psi}{2}\right) + \sin\left(\frac{\psi}{2}\right)i \equiv \cos\left(\frac{\psi}{2}\right)I + \sin\left(\frac{\psi}{2}\right)I_x \equiv \cos\left(\frac{\psi}{2}\right)\sigma_0 - i\sin\left(\frac{\psi}{2}\right)\sigma_z \\ q_{Ry} = \cos\left(\frac{\theta}{2}\right) + \sin\left(\frac{\theta}{2}\right>j \equiv \cos\left(\frac{\theta}{2}\right)I + \sin\left(\frac{\theta}{2}\right)I_y \equiv \cos\left(\frac{\theta}{2}\right)\sigma_0 - i\sin\left(\frac{\theta}{2}\right)\sigma_y \\ q_{Rz} = \cos\left(\frac{\phi}{2}\right) + \sin\left(\frac{\phi}{2}\right>k \equiv \cos\left(\frac{\phi}{2}\right)I + \sin\left(\frac{\phi}{2}\right)I_z \equiv \cos\left(\frac{\phi}{2}\right)\sigma_0 - i\sin\left(\frac{\phi}{2}\right)\sigma_x \\ q_{Tx} = x_T i \equiv x_T I_x \equiv -ix_T \sigma_z \\ q_{Ty} = y_T j \equiv y_T I_y \equiv -iy_T \sigma_y \\ q_{Tz} = z_T k \equiv z_T I_z \equiv -iz_T \sigma_x \end{cases} \quad (43)$$

Where ψ , θ and ϕ are respectively the simple rotations around the x, y and z 3D base frame axes, also called the yaw, pitch, and roll angles [17], q_{Rx} , q_{Ry} and q_{Rz} are the rotation quaternions, and q_{Tx} , q_{Ty} and q_{Tz} are the translation quaternions along these axes. The quantum model of a robotic arm can be determined using this Pauli-gate-based expression of quaternions. By virtue of relations (35), (36) and (37), the simple rotation and translation quaternions derive quantum computing equivalents from quantum spins as follows:

$$\begin{cases} q_{Rx} \equiv R_z(\psi) \\ q_{Ry} \equiv R_y(\theta) \\ q_{Rz} \equiv R_x(\phi) \\ q_{Tx} \equiv x_T R_z(\pi) \\ q_{Ty} \equiv y_T R_y(\pi) \\ q_{Tz} \equiv z_T R_x(\pi) \end{cases} \quad (44)$$

From (43) and (44), it can be inferred that Pauli's and the quaternion's representations have only the y axis in common as there is a linear relationship between the unit vectors I_y and σ_y ($I_y = -i\sigma_y$). One could expect this equivalence since in the quantum mechanics, a quantum representation of a function remains unaltered when multiplied by a constant [29]. On the other hand, the rotation around the axes x or z in Pauli algebra seems to correspond to a rotation around the other axis in the quaternion's 2D representation.

The above set of equations (44) is fundamental in the robotics field and important for quantum modelling of industrial robotic arms since it models orientation directly while also defining position in the last three lines. To complete the position model, a unit version must be considered, since in quantum computing, a qubit must be normalized during processing.

ILLUSTRATION, VALIDATION, AND DISCUSSION

The results obtained with the robotic arm position and orientation model are presented below. The concepts are validated using a rigid SCARA robotic arm. Frames L_k , $k = 0,1,2,3$ were established using Denavit-Hartenberg formalism as illustrated below.

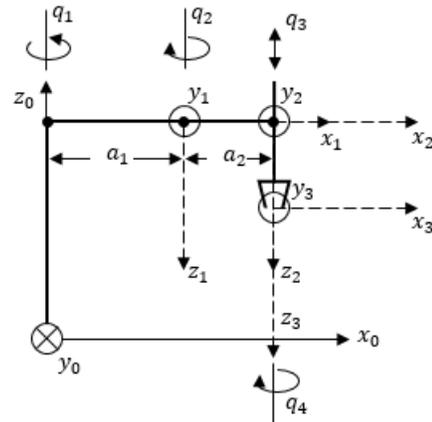


FIGURE 3
SCARA ROBOTIC ARM MOVEMENT RANGE REPRESENTED IN DENAVIT-HARTENBERG FORMALISM [32, 33, 34].

It can be inferred from Figure 3 that expressing the end effector position relative to the base frame requires the following basic transformations:

- i) From L_0 to L_1 , which can be performed as follows:
 - Translation d_1 along z_0 (using the transformation $-i d_1 \sigma_x$), then
 - Rotation θ_1 around z_0 (using the transformation $C\left(\frac{\theta_1}{2}\right)\sigma_0 - i S\left(\frac{\theta_1}{2}\right)\sigma_x$), then
 - Translation a_1 along x_0 (using the transformation $-i a_1 \sigma_z$).
- ii) From L_1 to L_2 , which can be performed as follows:
 - Rotation π around x_1 (using the transformation $-i \sigma_z$), then
 - Rotation θ_2 around z_1 (using the transformation $C\left(\frac{\theta_2}{2}\right)\sigma_0 - i S\left(\frac{\theta_2}{2}\right)\sigma_x$), then
 - Translation a_2 along x_1 (using the transformation $-i a_2 \sigma_z$).
- iii) From L_2 to L_3 , which can be performed as follows:
 - Translation d_3 along z_2 (using the transformation $-i d_3 \sigma_x$).

These successive elementary transformations (rotations and translations) are all performed using additions for translations and sandwich products for rotations. The final transformation that expresses the end effector position relative to base frame L_0 is obtained by applying the above simple transformations in the reverse direction, meaning that we start with the end effector point and work back towards the base frame.

For example, starting from $i d_3 \sigma_x$ of the end effector position, translation $a_2 i \sigma_z$ is then used to obtain $i a_2 \sigma_z + i d_3 \sigma_x$ to which rotation $-\theta_2$ is applied $(C(\frac{\theta_2}{2}) + i S(\frac{\theta_2}{2}) \sigma_x)(i a_2 \sigma_z + i d_3 \sigma_x)(C(\frac{\theta_2}{2}) - i S(\frac{\theta_2}{2}) \sigma_x)$ then rotation $i \sigma_z$ to this result, followed by translation $i a_1 \sigma_x$ then rotation $C(\frac{\theta_1}{2}) + i S(\frac{\theta_1}{2}) \sigma_x$ and ending with translation $i d_1 \sigma_x$. The computation of these successive transformations results in quaternion q_{03} or its equivalent Pauli matrix form shown below:

$$q_{03} = [a_1 \cos(\theta_1) + a_2 \cos(\theta_1 - \theta_2)]i + [a_1 \sin(\theta_1) + a_2 \sin(\theta_1 - \theta_2)]j + (d_1 - d_3)k \tag{45}$$

$$Q_{03} = -i[a_1 \cos(\theta_1) + a_2 \cos(\theta_1 - \theta_2)]\sigma_z - i[a_1 \sin(\theta_1) + a_2 \sin(\theta_1 - \theta_2)]\sigma_y - i(d_1 - d_3)\sigma_x = -i P_x \sigma_z - i P_y \sigma_y - i P_z \sigma_x = \begin{pmatrix} P_x i & -P_y - P_z i \\ P_y - P_z i & P_x i \end{pmatrix} \tag{46}$$

Where $P = (P_x, P_y, P_z)^T$ is the end-effector position vector. Positional information is extracted from the 2x2 matrix (46) as shown below:

$$\begin{cases} P_x = -\frac{i}{2}(Q_{03}(1,1) - Q_{03}(2,2)) \\ P_y = -\frac{1}{2}(Q_{03}(1,2) - Q_{03}(2,1)) \\ P_z = \frac{i}{2}(Q_{03}(1,2) + Q_{03}(2,1)) \end{cases} \tag{47}$$

This result can be compared with that obtained from the 3x3 matrix using the Denavit-Hartenberg formulation for SCARA robotic arm example [17], the position vector being the last column of the matrix shown below, which also contains the orientation 3x3 matrix embedded in the upper left portion.

$$T_0^3 = \begin{pmatrix} C_{1-2} & S_{1-2} & 0 & a_1 C_1 + a_2 C_{1-2} \\ S_{1-2} & -C_{1-2} & 0 & a_1 S_1 + a_2 S_{1-2} \\ 0 & 0 & -1 & d_1 - d_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{48}$$

C_1, S_1, C_{1-2} and S_{1-2} refer respectively to $\cos(\theta_1), \sin(\theta_1), \cos(\theta_1 - \theta_2)$ and $\sin(\theta_1 - \theta_2)$.

For the simulation, a Matlab program was developed. Table I provides geometrical information defining the simulated robot.

TABLE I
SCARA ROBOT ARM PARAMETERS ACCORDING TO THE D-H FORMULATION

Link	$a_k(m)$	$\alpha_k(rad)$	$d_k(m)$	$\theta_k(rad)$
1	0.3	π	0.5	θ_1
2	0.2	0	0.0	θ_2
3	0.0	0	d_3	--

Another program was implemented in Denavit-Hartenberg formalism to compute the forward kinematic model for comparison purposes.

For the simulation, a trajectory generated using the fifth-order polynomial was used [18], considering the initial values $\theta_{1init} = \frac{\pi}{4} rad, \theta_{2init} = \frac{\pi}{6} rad$ and $d_{3init} = 0.2 m$ and the final values $\theta_{1final} = \frac{\pi}{3} rad, \theta_{2final} = \frac{\pi}{2} rad$ and

$d_{3final} = 0.3 m$. The simulation results are shown in Figure 4. The results from the Denavit-Hartenberg formalism are presented in blue whereas the quantum algorithm's results are in red.

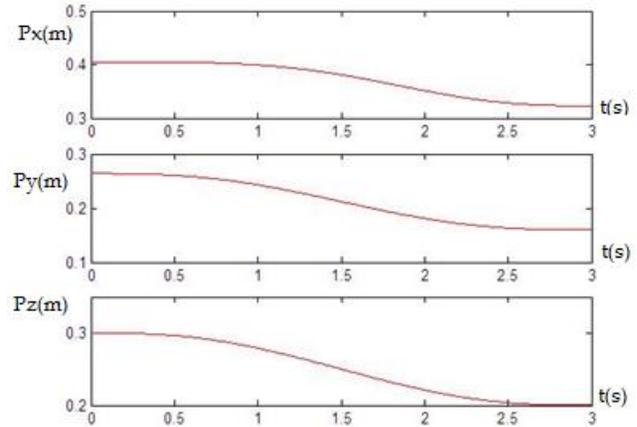


FIGURE 4
SIMULATION RESULTS FOR ROBOT ARM POSITION (FORWARD KINEMATICS) USING THE QUANTUM MODEL WITH DENAVIT-HARTENBERG FORMALISM

These graphs validate the quaternion-based quantum model developed in this study using several series of contiguous values for translation and rotation. The model is equivalent to the Denavit-Hartenberg formulation in terms of behaviour, its main advantage being that it is adapted not only to classical processors but also to quantum computers.

CONCLUSION

We describe, implement and validate a novel model of robotic arm position based on the emerging concept of quantum computing. Equivalence between quaternions and quantum Pauli gates has been established, allowing the transformation between two representations, and leading to a new quantum model for robotic arm position. A program was implemented to compare the new model to classical modelling approaches in simulations. The classical model's results and the quantum model results were comparable and illustrative of robotic arm movement. This study shows that mechatronics systems modelling and powerful software simulations using quantum computers may be expected to become routine in the future.

REFERENCES

- [1] Pimentel, D. R. M. and Castro, A. S., 2013, "A Laplace transform approach to the quantum harmonic oscillator," European Journal of Physics, Volume 34, pp. 199-204.
- [2] Alp, N. and Sarikaya, M. Z., 2020, "q-Laplace transform on quantum integral," Kragujevac Journal of Mathematics, p. 153-164.

- [3] Jorrand, P., 2006, "Transformée de Fourier quantique – Algorithmes. Notes de cours," Module Informatique Quantique, UJF Grenoble.
- [4] "Quantum Fourier Transform," Qiskit, [Online]. Available: <https://qiskit.org/textbook/ch-algorithms/quantum-fourier-transform.html>. [Accessed 10 05 2021].
- [5] BERRY, D. W., 2014, "High-order quantum algorithm for solving linear differential equations," *Journal of Physics A: Mathematical and Theoretical*, Volume 47, Number 10.
- [6] Cao, Y., Daskin, A., Frankel, S. and Kais, S., 2012, "Quantum circuit design for solving linear systems of equations," *Molecular Physics*, Volume 110, pp. 1675-1680.
- [7] Cai, X.-D., Weedbrook, C., Su, Z.-E., Chen, M.-C., Gu, M., Zhu, M.-J., Li, L., Liu, N.-L., Lu, C.-Y. and Pan, J.-W., 2013, "Experimental quantum computing to solve systems of linear equations," *Physical review letters*.
- [8] Nagata, K. and Nakamura, T., 2018, "Quantum algorithm for the root-finding problem," Chapman University, Springer.
- [9] Leyton, S. K. and Osborne, T. J., 2008, "A quantum algorithm to solve nonlinear differential equations," arXiv:0812.4423v1.
- [10] Xin, T., Wei, S., Cui, J., Xiao, J., Arrazola, I., Lamata, L., Kong, X., Lu, D., Solano, E. and Long, G., 2018, "A Quantum Algorithm for Solving Linear Differential Equations: Theory and Experiment," *Phys. Rev. A* 101, 032307.
- [11] Terno, D. R., 1999, "Nonlinear operations in quantum-information theory," *physical review a* volume 59, number 5, pp. 3320-3324.
- [12] Daoud, E. A., 2007, "Quantum Computing for Solving a System of Nonlinear Equations over GF(q)," *The International Arab Journal of Information Technology*, Vol. 4, N. 3, pp. 201-205.
- [13] Grover, L. K., 1996, "A fast quantum mechanical algorithm for database search," In *STOC '96: Proceedings of the twenty-eighth annual ACM symposium on Theory of Computing*.
- [14] Singhal, A. and Chatterjee, A., 2018, "Grover's Algorithm".
- [15] Wossnig, L., Zhao, Z. and Prakash, A., 2017, "A quantum linear system algorithm for dense matrices," *Phys. Rev. Lett.* 120.
- [16] Kyriienko, O., 2020, "Quantum inverse iteration algorithm for programmable quantum simulators," *npj Quantum Information*.
- [17] Schilling, R. J., 1996, "Fundamentals of robotics: analysis and control," Simon & Schuster Trade.
- [18] Dombre, E. and Khalil, W., 1999, "Modélisation, identification et commande des robots," *Hermès science*.
- [19] Siciliano, B., Sciavicco, L., Villani, L. and Oriolo, G., 2010, "Robotics: modelling, planning and control," Springer Science & Business Media.
- [20] Sciavicco, L. and Siciliano, B., 2012, "Modelling and control of robot manipulators," Springer Science & Business Media.
- [21] Förstner, W. and Wrobel, B., 2016, "Photogrammetric computer vision: statistics, geometry, orientation and reconstruction," Springer.
- [22] Mittal, R. and Nagrath, I., 2003, "Robotics and control," Tata McGraw-Hill.
- [23] Blösch, M., 2015, "An Introduction to 3D Orientations and Quaternions," In ETH Zurich, Zurich.
- [24] Adorno, B. V., 2017, "Robot Kinematic Modeling and Control Based on Dual Quaternion Algebra - Part I: Fundamentals," Department of Electrical Engineering, Federal University of Minas Gerais, Brazil.
- [25] Al Attar, A. and Kormushev, P., 2020, "Kinematic-Model-Free Orientation Control for Robot Manipulation Using Locally Weighted Dual Quaternions" *Robotics*.
- [26] Barbic, J., 2020, "Quaternions and Rotations," In *CSCI 420 Computer Graphics*.
- [27] Chen, Zielinska, L. T., Wang, J. and Ge, W., 2020, "Solution of an inverse kinematics problem using dual quaternions," *International Journal of Applied Mathematics and Computer Science*.
- [28] Valverde, A. and Tsiotras, P., 2002, "Spacecraft Robot Kinematics Using Dual Quaternions," *Robotics* 2018, 7, 64, 218.
- [29] Griffiths, R. B. (2002). *Consistent Quantum Theory*. Cambridge University Press.
- [30] McMahon, D., 2007, "Quantum Computing Explained," Wiley - IEEE.
- [31] Abhijith, J., Adedoyin, A., Ambrosiano, J., Anisimov, P., Bärttschi, A., Casper, W., Chennupati, G., Coffrin, C., Djidjev, H., Gunter, D., Karra, S., Lemons, N., Lin, S., Malyzhenkov, A., Mascarenas, D., Mniszewski, S., Nadiga, B., O'Malley, D., Oyen, D., Pakin, S., Prasad, L., Roberts, R., Romero, P., Santhi, N., Sinitsyn, N., Swart, P.J., Wendelberger, J.G., Yoon, B., Zamora, R., Zhu, W., Eidenbenz, S., Coles, P.J., Vuffray, M. and Likhov A.Y., 2018, "Quantum Algorithm Implementations for Beginners," ArXiv.
- [32] Zioui, N. and Chikh, L., 2006, "Design, realization and control of a SCARA robot," National Polytechnic School (ENP), Algiers, Algeria.
- [33] Zioui, N., Mahmoudi, A., Mahmoudi, Y. and Rezgui, A., 2021, "A comparative study of performances between the sliding modes and the thrust control strategies for an articulated robotic arm position control," *International Journal of Mechatronics and Mechanical Engineering*, vol. 21, no. 1.
- [34] Zioui, N., Mahmoudi, Y., Mahmoudi, A., Tadjine, M. and Bentouba, S., 2021, "A New Quantum-computing-based Algorithm for Robotic Arms and Rigid Bodies' Orientation," *Journal of Applied and Computational Mechanics*, vol. 07, no. 3.