Research Article

BIOMEDICAL Journal of Scientific & Technical Research

ISSN: 2574 -1241

DOI: 10.26717/BJSTR.2025.61.009575

Conformal Geometric Algebra for Medical Robotics and Quaternion Wavelet Neural Network for Adaptive PID Control of Haptic Devices

Eduardo Bayro- Corrochano[1]*, Gerardo Ortega- Flores[2] and Gerardo Martínez-Terran[2]

[1]*Institute of Automation and Robotics of the Poznan University of Technology, Poland*

[2]*CINVESTAV, Department of Computer Science and Electrical Engineering, Campus Guadalajara, México*

***Corresponding author:** Eduardo Bayro- Corrochano, Institute of Automation and Robotics of the Poznan University of Technology, Poland

ARTICLE INFO

Received: March 17, 2025

Published: April 08, 2025

Citation: Eduardo Bayro- Corrochano, Gerardo Ortega- Flores and Gerardo Martínez-Terran. Conformal Geometric Algebra for Medical Robotics and Quaternion Wavelet Neural Network for Adaptive PID Control of Haptic Devices. Biomed J Sci & Tech Res 61(2)-2025. BJSTR. MS.ID.009575.

ABSTRACT

This work aims to apply geometrical methods in generating an interpolated trajectory using polynomial functions to simulate medical procedures. Translation and rotation are represented using dual quaternions, a mathematical tool that provides a unified and compact representation of complex problems. The approach involves converting dual-quaternions into motors—operators that facilitate the rotation and translation of virtual points to align with real points. These motors are processed using an interpolation method based on geometric algebra, then transformed into coordinates and orientations. Finally, an industrial robot executes the interpolated trajectory in real time, enabling its interpretation as a suture maneuver, implant placement, ultrasound procedure, or incision. This work highlights the efficiency of geometric algebra in modeling geometric entities, offering a less complex alternative to matrix algebra while providing greater flexibility in representing various problems through geometric elements. Additionally, we introduce an adaptive PID control law for motion paths in an open-chain haptic system. The plant's identification is achieved using a Quaternion Wavelet Neural Network, which optimally tunes the PID values, ensuring precise movement within the workspace.

Keywords: Conformal Geometric Algebra; Dual Quaternions; Motors; Conformal Transformation; Direct Kinematics; Invers Kinematics; Interpolation; Robotic Surgery

Introduction

Surgery is a medical specialty focused on healing injuries or diseases through operative procedures. It involves manual or instrument-assisted techniques, where precise suturing is essential for success. Robotics, a branch of engineering, supports the design, construction, operation, and application of systems capable of performing various tasks. The primary goal of robotic systems is to replace human intervention—whether for economic, industrial, or safety-related purposes. Robotic manipulator arms have proven especially useful, offering high precision in executing commands from operators. Their applications span from industrial settings to medical environments, including training scenarios. On another front, geometric methods offer a formal approach to analyzing geometrically structured objects. Geometric algebra, which stems from the works of Grassmann [1] and Clifford [2], was further developed by physicist David Hestenes [3]. Notable contributors to this field include Bayro-Corrochano [4], Perwass [5], Hildenbrand [6], and Dorst [7]. Geometric algebra is a mathematical framework that blends abstract algebra with geometry. It examines various algebras that represent geometric manifestations of polynomial equation solutions, making it a powerful and flexible tool [8]. One key aspect of geometric algebra is interpolation, which offers competitive advantages over other algorithms. It tends to eliminate redundant coefficients, maintains geometric intuition, and is both compact and efficient. As such, geometric interpolation holds significant potential in developing algorithms for efficient medical interventions.

For example, geometric interpolation allows robots to develop jobs of high accuracy. In this paper, the way of interpolation applicated will be an algorithm of interpolation in the quadratic of Study with geometric algebra because, on some occasions, the desired way has discontinuities or breaks. The objective of this assignment will be to use geometrical methods for its application in the creation of interpolate trajectories, with polynomial functions, that simulate some types of medical interventions. The operator (surgeon) wills supply implicitly the coordinates and the orientations of the desired points and will be transformed into a combination of theory using dual numbers and quaternions called dual quaternions with the aim of generating achievable surgery paths for a "MOTOMAN" industrial robot. Haptic devices for surgery tasks have non-linear dynamic properties where diverse contributions consider linear approximations for to control the system in an optimal region of space, however, it does not guarantee convergence in other neighborhoods [9-12]. Robotics and haptic systems have allowed new and diverse applications in the field of medicine, such as assisted surgery and teleoperation which have increasingly stringent requirements for accuracy, convergence, and low computational consumption. In this paper, we present alsoan adaptive PID control law (Proportional Integral Derivative controller, PID), of indirect architecture is presented for movement paths in a haptic system of open chain, where the identification of the plant is through a quaternionic wavelet neural network (Quaternion Wavelet Neural Network, QWNN) for tune the PID values, this allows the optimal movement into the regions of the workspace.

The organization of this paper is as follows: Sect. 2 outlines geometric algebra, motor algebra and quaternion and dual quaternion algebra. In Sect. 3, we explain the Kinematics of a serial robot arm. Sect. 4 explains the algorithm for interpolation. In Sect. 5, we explain the algorithms for trajectory planning. Sect. 6 explains the design of the Quaternion Wavelet Neural Network. Sect. 7 describes the techniques for robot maneuvers in liver surgery. Sect. 8 presents Wavelet Neural Network for Adaptive PID Control in Haptics. And, finally, Sect.9 is devoted to the conclusions.

Geometric Algebra

Geometric algebra is defined as a G_n algebra, which is generated by a vector V^n with n dimensions and a set of vectors withorthonormal base $\{e_1, e_2, .., e_n\}$. If we propose two base vectors whatever $\{e_j, e_k\}$, a algebra is gotten by a product and denoted by:

$$e_j e_k = e_{jk} \quad \text{.....(1)}$$

Where the product by two base vectors is orthonormal and non-commutative:

$$e_j e_k = -e_k e_j \quad \text{.....(2)}$$

Base vectors must have the next property:

$$e_i e_i = e_i^2 = \begin{cases} +1 \, for \; i = 1, \ldots, p \\ -1 \, for \; i = p+1, \ldots, p+q \\ 0 \, for \; i = p+q+a, \ldots, n \end{cases} \quad \text{.....(3)}$$

This product is known as geometric product of G_n with dimension $n = p+q+r$, where p, q and rare non- negative entire. The number of elements is equal to:

$$\sum_{k=0}^{n} \binom{n}{k} = 2^n \quad \text{.....(4)}$$

Which are homogeneous subspaces with 0 grade (scalar), 1 grade (vector), 2 grade (bivector), 3 grade (trivector), ... and n grade.

For example, if we have a geometric algebra of with base multivectors:

$$\{1, e_1, e_2, e_3, e_{12}, e_{31}, e_{23}, I_3 = e_{123}\} \quad \text{.....(5)}$$

Now, we can define the inside product as:

$$a.b = \frac{1}{2}(ab + ba) \quad \text{.....(6)}$$

And the outside or *wedge* product:

$$a \wedge b = \frac{1}{2}(ab - ba) \quad \text{.....(7)}$$

In this form, geometric product can be expressed now as:

$$ab = a.b + a \wedge b \quad \text{.....(8)}$$

Where $a.b = b.a$ and $a \wedge b = -b \wedge a$.

Finally, we can view in the previous equations that the point product makes a scalar while the *wedge* product makes a bivector [4].

Conformal Geometric Algebra

Conformal geometric algebra is an interesting case because we can find that its unit element is the sphere, and it allows us to represent other geometrics in terms of primitive entities. This is possible by representing the Euclidean spaceR^nin$R^{n+1,1}$ (R^{pq} is a vectorial space of n=p+q dimention on the field with signature(p,q) [13].

There is a set of base vectors with the next properties:

$$e_i^2 = 1, i = 1, \ldots, n \quad \text{.....(9)}$$

$$e_{\pm} = \pm 1 \quad \text{.....(10)}$$

$$e_i.e_+ = e_i.e_- = e_+.e_- = 0, i = 1, \ldots, n \quad \text{.....(11)}$$

The null bases $\{e_0, e_\infty\}$ are defined as:

$$e_0 = \frac{(e_- - e_+)}{2} \quad \text{.....(12)}$$

$$e_{\infty} = e_{-} + e_{+} \text{.....(13)}$$

Where:

$$e_0^2 = e_{\infty}^2 = 0 \text{.....(14)}$$

$$e_{\infty} . e_0 = -1 \text{.....(15)}$$

While pseudoscalar

$(set\ of\ vectors\ that\ square\ up\ in\ a\ positive\ number,\ negative\ number\ and\ zero\ I_n = e_{12\ldots n}) E \in R^{1,1}$

is defined as:

$$E = e_{\infty} \wedge e_0 = e_{+} \wedge e_{-} = e_{+} e_{-} \quad (16)$$

Where:

$$E^2 = 1 \quad (17)$$

$$\tilde{E} = -E \quad (18)$$

$$E_{e\pm} = e_{+}^{-} = -e_{\pm} \quad (19)$$

$$E_{e_{\infty}} = -e_{\infty} E = -e_{\infty} \quad (20)$$

$$E_{e_0} = -e_0 E = e_0 \quad (21)$$

$$1 - E = -e_{\infty} e_0 \quad (22)$$

$$1 + E = -e_0 e_{\infty} \quad (23)$$

Conformal Transformation

When we work with geometric algebra, there are operations of transformation that use invertible multivectors through geometric products. There are 2 types of geometric products of invertible multivectors called versors too:

Uneven Versor:

$$-VO\tilde{V} \text{.....(24)}$$

Even Versor:

$$VO\tilde{V} \text{.....(25)}$$

Where V is the multivector of transformation and O is the geometric parameter for transformation [8].

Quaternion Algebra

Quaternion numbers were discovered by William Hamilton in 1843. These numbers represent four-dimensional space and are composed of a real part and three imaginary parts. The four-dimensional quaternion algebra H [13] is associative, non-commutative

$$H = \{q = q_0 + q_x i + q_y j + q_z k \mid q_o, q_x, q_y, q_z \in R\} \text{,.....(26)}$$

where the orthogonal imaginary numbers *i, j,* and *k* fulfill

$$i^2 = j^2 = -1, k = ij = -ji \rightarrow k^2 = -1 \text{.....(27)}$$

A quaternion is conjugated as follows

The modulus of the quaternion $q = q_0 + q_x i + q_y j + q_z k$ is defined by

$$|q| = \sqrt{q\bar{q}} = \sqrt{{q_0}^2 + {q_z}^2 + {q_y}^2 + {q_z}^2} \quad (28)$$

Given q,p∈H

$$|q| = |\bar{q}|, and\ \tilde{qp} = \tilde{p}\tilde{q},$$
$$|qp| = |pq| = |q||p| \text{,.....(29)}$$
$$|q + p| \leq |q| + |p|.$$

One computes the inverse of a quaternionwith$q_6$different to zero,as follows

$$q^{-1} = \frac{\bar{q}}{|q^2|} \text{.....(30)}$$

A pure quaternion is a quaternion with its scalar element zero $q_0 = 0$

$$q_p = q_x i + q_y j + q_z k,$$

and a unit quaternion is given by

$$q_u = s + xi + yj + zk, for\ ||\ q_u\ ||\ = 1$$

The polar representation of a quaternion $q = q_0 + q_x i + q_y j + q_z k$is

$$q = |\ q\ | e^{i\varphi} e^{k\psi} e^{j\theta} \text{,.....(31)}$$

where the phase ranges are delimited as follows:

$$(\varphi, \theta, \psi) \left[-\pi, \pi \left[\times [-\frac{\pi}{2}, \frac{\pi}{2} [\times [-\frac{\pi}{4}, \frac{\pi}{4}] \right] \right] \in .$$

Dual Quaternions

In 1882, Clifford introduced a combination of the theory of the dual number $(z = r + d\varepsilon,\ where \varepsilon\ is\ the\ dual\ operator, is\ the\ real\ part\ and\ the\ dual\ part)$ with the quaternions q=w+(xi + yj+ zk), where , , y are numerical values and , y are the imaginary components), which Clifford called dual quaternions [2]. Dual quaternions can represent rotation and translation. They are formed by 2 quaternions called real and dual part, with 4 parameters each one:

$$q = q_r + q_d \varepsilon \text{.....(32)}$$

The operations used for this coupling are:

Multiplication by a scalar (where s is scalar):

$$sq = sq_r + sq_d\varepsilon \text{.....(33)}$$

Addition:

$$q_1 + q_2 = q_{r1} + q_{r2} + (q_{d1} + q_{d2})\varepsilon \text{.....(34)}$$

Multiplication:

$$q_1 q_2 = q_{r1}q_{r2} + \left(q_{r1}q_{d2} + q_{d1}q_{r2}\right)\varepsilon \text{.....(36)}$$

Conjugate:

$$q^* = q_r^* + q_d^*\varepsilon \text{.....(37)}$$

Magnitude:

$$\|q\| = qq^* \text{.....(38)}$$

Motors

Clifford introduced motors with the name of biquaternions [4]. They are in the subalgebra a of $G_{3,0,1}$ ($G_{p,q,r}$ is a special geometric al gebra or deformed on R_{pq}) as $G^+_{3,0,1}$ (when a subalgebra is shaped by vectors of base grade of a G_n algebra, it will be denoted G_n^+):

$$1, e_2e_3, e_3e_1, e_1e_2, e_4e_1, e_4e_2, e_4e_3, I \text{.....(39)}$$

For motors to be isomorphic to dual quaternions, it is necessary that they meet the condition:

$$I^2 = 0 \text{.....(40)}$$

Motors can be separated by multiplication because they are made by rotation and translation successive, which they are called rotor and translator:

$$M = T_s R_s = \left(a_c - Ia_s\frac{d}{2}\right) + \left(a_s + Ia_c\frac{d}{2}\right)l \text{.....(41)}$$

Where:

$$a_c = \cos\frac{\theta}{2} \text{.....(42)}$$

$$a_s = \sin\frac{\theta}{2} \text{.....(43)}$$

where d is the distance of displacement and l is the axis of rotation.

In Euler terms:

$$M = \cos\left(\frac{\theta}{2} + I\frac{d}{2}\right) + \sin\left(\frac{\theta}{2} + I\frac{d}{2}\right)l \text{.....(44)}$$

Some of their features are:

$$M\alpha = \alpha M \text{ where } \alpha \in R \text{.....(45)}$$

$$\left|M\right| = M\tilde{M} = 1 \text{.....(46)}$$

$$\tilde{M} = \tilde{R}_s\tilde{T}_s \text{.....(47)}$$

See in Figure 1 a depiction of the screw motion on l axis, where the motion is carried out by a motor.

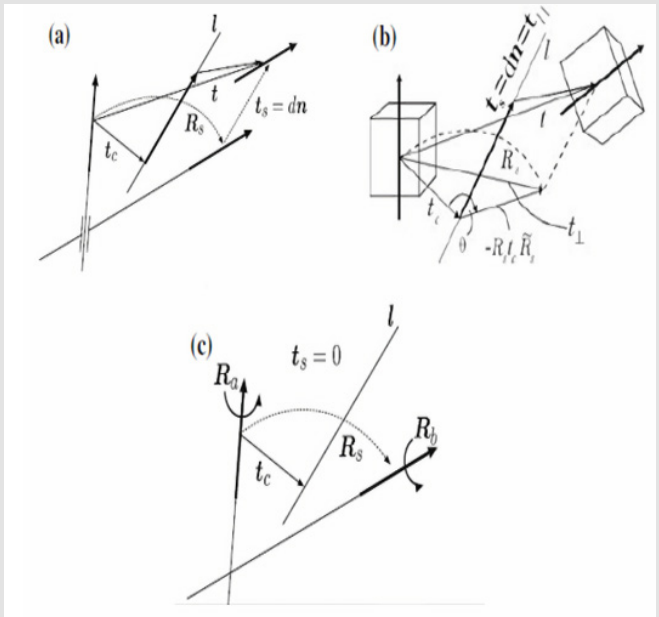

Figure 1: Screw movement on l axis.
a) Motor with 2 axes,
b) Motor applied to an object and
c) Motor without displacement [1].

Kinematics

Direct Kinematic

Direct Kinematic help us to find the solve for problems relation with individual joints for the robot, the position and orientation of end effector. According to this work, the "MOTOMAN" robot of Control laboratory in CINVESTAV is showed in Figure 2. With the previous information, we can compute theDenavit- Hartenberg parameters as shown in Table 1.

Figure 2: "MOTOMAN" robot.

Table 1: Denavit- Hartenberg parameters of "MOTOMAN" robot (*variables).

i	a	α	d	θ
1	a_1	$-\frac{\pi}{2}$	0	θ_1
2	a_2	$-\pi$	0	$\theta_2^* - \frac{\pi}{2}$
3	a_3	$\frac{\pi}{2}$	0	θ_3^*
4	0	$-\frac{\pi}{2}$	d_4	θ_4
5	0	$\frac{\pi}{2}$	0	θ_5
6	0	0	d_6	θ_6

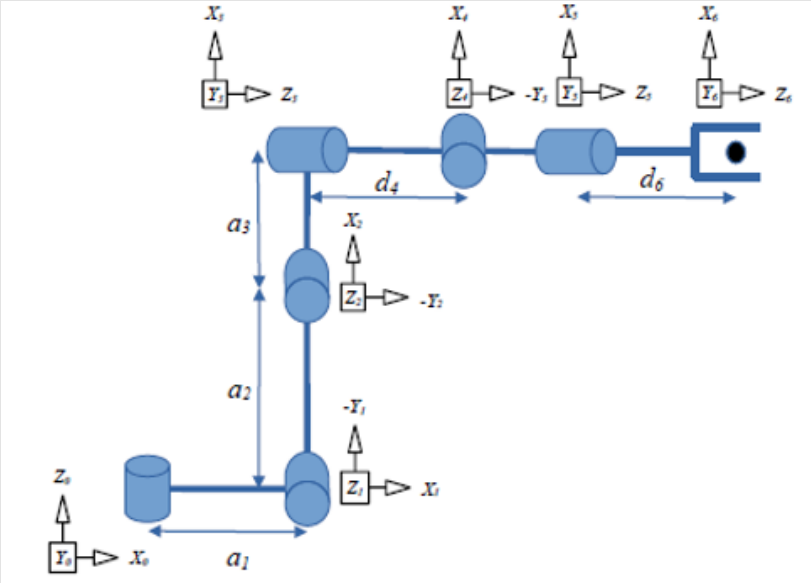

Figure 3: Frame of coordinates of "MOTOMAN" robot.

Using the robot dynamic and its lengths, we will select the next frame of coordinates (Figure 3)

Using Denavit-Hartenberg parameters of Table 1., the Homogeneous Transformation Matrix T_6^0 is

$$H = T_6^0 = A_1 A_2 A_3 A_4 A_5 A_6 \quad (48)$$

$$= \begin{bmatrix} h_{11} & h_{12} & h_{13} & h_{14} \\ h_{21} & h_{22} & h_{23} & h_{24} \\ h_{31} & h_{32} & h_{33} & h_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (49)$$

where:

$$A_i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i \cos\alpha_i & \sin\theta_i \sin\alpha_i & a_i \cos\theta_i \\ \sin\theta_i & \cos\theta_i \cos\alpha_i & -\cos\theta_i \sin\alpha_i & a_i \sin\theta_i \\ 0 & \sin\sin\alpha_i & \cos\cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (50)$$

$$T_6^0 = \begin{bmatrix} R_6^0 & o_6^0 \\ 0 & 1 \end{bmatrix} \quad (51)$$

We can get the orientation and the position of end effector with R_6^0 matrix coordinate and o_6^0 vector coordinate respectively.

Inverse Kinematic

Inverse Kinematic help us to find the joint variables for the position and orientation of the end effector:

$$R_n^0(q_1,\ldots,q_n) = R \quad (52)$$

$$o_n^0(q_1,\ldots,q_n) = o \quad (53)$$

Where R is the desired rotation and is the desired position. We are given O and R and the inverse kinematics problem is to solve for

$$q_1,\ldots,q_n$$

The "MOTOMAN" robot has 6 joints, which the last 3 joints intersect at a point$_c$, so we can make a decouple. With this information and solving some equations, we get:

$$o = \begin{bmatrix} o_x \\ o_y \\ o_z \end{bmatrix} \quad (54)$$

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (55)$$

$$o_c = \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = \begin{bmatrix} o_x - l_6 r_{13} \\ o_y - l_6 r_{23} \\ o_z - l_6 r_{33} \end{bmatrix} \quad (56)$$

Due the movement of the machine and using the plane projection, it allows us to deduce:

$$\theta_1 = Atan(x_{04}, y_{04}) \quad (57)$$

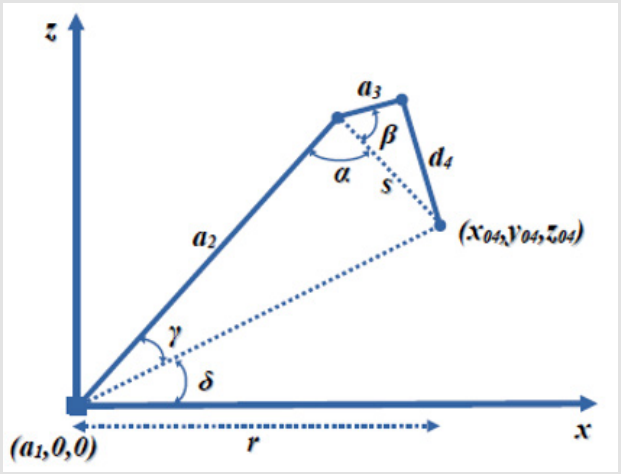

Figure 4: Inverse Kinematic Model.

Using the plane projection and some geometrics methods, the solution is gotten with the next form:

Where the conclusion is (Figure 4)

$$\theta_3 = \alpha - \beta + \frac{\pi}{2} \quad \text{.....(58)}$$

Where:

$$\alpha = Atan\left(\cos(\alpha), -\sqrt{1-\cos^2(\alpha)}\right) \quad \text{.....(59)}$$

$$\cos(\alpha) = \frac{r^2 + z_{04}^2 - s^2 - a_2^2}{2sa_2} \quad \text{.....(60)}$$

$$r = \sqrt{x_{04}^2 + y_{04}^2} - a_1 \quad \text{.....(61)}$$

$$\beta = \text{asin}\left(\frac{a_3}{s}\right)$$

Continuing with the operations, the next equation is obtained:

$$\theta_2 = \gamma - \delta + \frac{\pi}{2} \quad \text{.....(62)}$$

$$\gamma = -\text{atan}\left(\frac{\sin(\alpha)s}{a_2 + (s\cos(\alpha))}\right) \quad \text{.....(63)}$$

$$\delta = \text{atan}(z_{04}, r) \quad \text{.....(64)}$$

Due the movement of the robot and its initial position, we can get:

$$\theta_2 = -(\theta_2 - 90^0) \quad \text{.....(65)}$$

$$\theta_3 = \theta_2 + 90^0 \quad \text{.....(66)}$$

For the next three movements, we can use the next knowledge:

$$A_1 A_2 A_3 = \begin{bmatrix} R_3^0 & o_3^0 \\ 0 & 1 \end{bmatrix} \quad \text{.....(67)}$$

$$A_4 A_5 A_6 = \begin{bmatrix} R_6^3 & o_6^3 \\ 0 & 1 \end{bmatrix} \quad \text{.....(68)}$$

$$R = R_3^0 R_6^3 \Rightarrow R_6^3 = \left(R_3^0\right)^{-1} R = \left(R_3^0\right)^T R \quad \text{.....(69)}$$

As R_6^3 is a function of θ_4, θ_5 *and* θ_6, we can get:

$$\theta_4 = \text{atan}(-R_{6M23}^3, R_{6M13}^3) \quad \text{.....(70)}$$

$$\theta_5 = \text{atan}(\sqrt{1-\left(R_{6M33}^3\right)^2}, R_{6M33}^3) \quad \text{.....(71)}$$

$$\theta_6 = \text{atan}(-R_{6M32}^3, -R_{6M31}^3) \quad \text{.....(72)}$$

In the same form, due the movement of the robot, we get:

$$\theta_6 = -\theta_6 \quad \text{.....(73)}$$

Interpolation

In this work, algorithm of interpolation in quadratic of *study* with geometric algebra is used as method of interpolation, because, on some occasion, the decided trajectories have discontinuities or breaking. The interpolation algorithm which is based in mapping of movements in *study* quadratic says: Through a mapping, can be represented whatever rigid transformation as homogeneous vector in a projective space:

$$\alpha \to x = \{x_0, \ldots, x_7\}^T \quad \ldots\ldots\ldots\ldots\ldots\ldots(74)$$

X should be the following:

$$x_0x_4 + x_1x_5 + x_2x_6 + x_3x_7 = 0 \ldots\ldots\ldots\ldots\ldots\ldots(75)$$

Now, if there are 3 homogeneous vectors x_0, x_1 and x_2, then we can represent a curve of interpolation, which is shaped by a group of vectors and corresponds to a tracking:

$$C = \frac{f_0(t)X_1^TQX_2X_0 + f_1(t)X_0^TQX_2X_1 + f_2(t)X_0^TQX_1X_2}{2} \ldots\ldots\ldots\ldots\ldots\ldots(76)$$

Where $f_0(t), f_1(t)$ *and* $f_2(t)$ are the polynomials of interpolation and are given by:

$$f_0(t) = (t_0 - t_1)(t_0 - t_2)(t - t_1)(t - t_2) \ldots\ldots\ldots\ldots\ldots\ldots(77)$$

$$f_1(t) = (t_1 - t_0)(t_1 - t_2)(t - t_0)(t - t_2) \ldots\ldots\ldots\ldots\ldots\ldots(78)$$

$$f_2(t) = (t_2 - t_0)(t_2 - t_1)(t - t_0)(t - t_1) \ldots\ldots\ldots\ldots\ldots\ldots(79)$$

Where t is the value of interpolation that goes to 0 and 1 while t_0, t_1 and t_2 are points of the curve where homogeneous vectors coincide with the interpolation vectors. Q is a bilinear matrix of *study* quadratic:

$$Q = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \ldots\ldots\ldots\ldots\ldots\ldots(80)$$

As in the previous case, homogeneous pointscan be gotten by mapping of quaternions or motors, too: (Table 2)

Table 2: Table of equivalences [9].

Vector Homogeneo	Cuaternion Dual	Motor	Motor
X_0	α_0	α_0	α_0
X_1	α_1^i	$\alpha_1 e_{12}$	$\alpha_1 e_{12}$
X_2	α_1^i	$\alpha_2 e_{31}$	$\alpha_2 e_{31}$
X_3	α_3^k	$\alpha_3 e_{23}$	$\alpha_3 e_{23}$
X_4	$\alpha_4^\in$	$\alpha_{4\ e1234}$	$\alpha_4 e_{123\infty}$
X_5	$\alpha_4^\in$	$\alpha_5 e_{43}$	$\alpha_5 e_{\infty 3}$
X_6	$\alpha_4^\in$	$\alpha_6 e_{42}$	$\alpha_6 e_{\infty 2}$
X_7	$\alpha_7^{\in k}$	$\alpha_7 e_{41}$	$\alpha_7 e_{\infty 1}$

For this work, we use mapping by motors, because, in previous works, this method has been used and given best results with respect to time of implementation [8]. Now, we know that:

$$X_a^TQX_b = \sum_{i=0}^{7} X_{a_i} X_{b_{7-i}} \ldots\ldots\ldots\ldots\ldots\ldots(81)$$

In case of motors, if the previous equation is substituted and if we see the *blade* appropriate:

$$M_aM_b = M_{a_0}M_{b_7} - \sum_{i=1}^{6} M_{a_i}M_{b_6} + M_{a_7}M_{b_0} \ldots\ldots\ldots\ldots\ldots\ldots(82)$$

Using the throwback operations:

$$M_a \tilde{M}_b = \left(\sum_{i=0}^{7} M_{a_i} M_{b_{7-i}} \right) e_{123\infty} \ldots\ldots\ldots\ldots\ldots\ldots(83)$$

Extracting the *blade* coefficient and using partial derivatives, we can get:

$$M_t' = \frac{f_0(t)K_0M_0 + f_1(t)K_1M_1 + f_2(t)K_2M_2}{2} \ldots\ldots\ldots\ldots\ldots\ldots(84)$$

Where:

$$K_0 = M_1 \tilde{M}_2 \ldots\ldots\ldots\ldots\ldots\ldots(85)$$

$$K_1 = M_0 \tilde{M}_2 \ldots\ldots\ldots\ldots\ldots\ldots(86)$$

$$K_2 = M_0 \tilde{M}_1 \ldots\ldots\ldots\ldots\ldots\ldots(87)$$

(Algorithm 1).

Función `Interpolación`(X,t):
Entrada: M : Conjunto de N motores base necesarios para la interpolación.
Entrada: t : Variable de interpolación en el rango de 0 a 1.
Salida: M_p : Motor que representa la nueva posición del punto inicial.

inicio
  si $|M|$ *es impar* **entonces**
    si $|M| = 1$ **entonces**
      $Mp = M$
    en otro caso
      $Mc \longleftarrow \varnothing$
      $f_0 \longleftarrow (t_0 - t_1)(t_0 - t_2)(t - t_1)(t - t_2)$
      $f_1 \longleftarrow (t_1 - t_0)(t_1 - t_2)(t - t_0)(t - t_2)$
      $f_2 \longleftarrow (t_2 - t_0)(t_2 - t_1)(t - t_0)(t - t_1)$
      para $i \longleftarrow 2$ **a** $|X| - 1$ **hacer**
        $K_0 \longleftarrow$ `Coeficiente` $\left(\left\langle M_i \widetilde{M_N} \right\rangle_4\right)$
        $K_1 \longleftarrow$ `Coeficiente` $\left(\left\langle M_1 \widetilde{M_N} \right\rangle_4\right)$
        $K_2 \longleftarrow$ `Coeficiente` $\left(\left\langle M_1 \widetilde{M_i} \right\rangle_4\right)$
        $Mc_{i-1} \longleftarrow \dfrac{f_0 K_0 M_1 + f_1 K_1 M_i + f_2 K_2 M_N}{2}$
      fin
    fin
    $Mp \longleftarrow$ `Interpolación`(Mc,t)
  fin
  devolver Mp
fin

Algorithm 1: Motor interpolation is based in geometric algebra [9].

Trajectory Planning

Motion planning is developed by algorithms oriented to medical robotics and the use of various conformal algebra techniques such as those presented above, for application purposes.

Organ Record

For surgery, it is necessary to register the pose of the real organ with its representation in the virtual world and it consists of obtaining a series of points, corresponding to important positions. Once the registration is done, the surgeon cancollate key cuts and surgical routes. To perform this operation, it is necessary to determine the motor , we can correct the position of the virtual organ according to the real one. This operation is expressed as: according to the real one. This operation is expressed as:

$$X_i^{'} = MX_i \tilde{M} \quad \text{.....(88)}$$

The registration of an organ allows us to always know the position and orientation of important points for the surgeon (Figure 5).

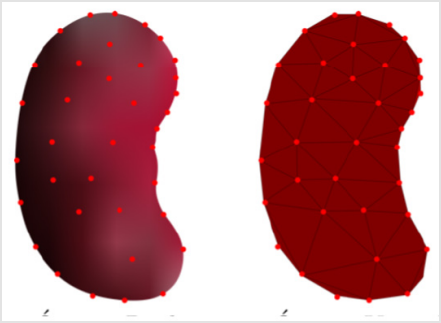

Figure 5: Organ record [9].

Surgical Cut

For surgical cutting it is important to know the position and orientation of the instrument with respect to the surface of the organ. To calculate the incidence between the scalpel and the surface of the organ, the plane of the surface represented three points $\phi = p_i \wedge p_j \wedge p_k$ is taken and for the line of incidence of the scalpel two points $L = p_1 \wedge p_2$ that describe its contour as shown in figure. This relation can be expressed by equation 89:

$$\sigma_{cq} = \pi L + L \pi \text{.....(89)}$$

If $\sigma_{cq} = 0$, it means that the scalpel is on the surface of the organ and if $\sigma_{cq} < 0$, the scalpel has penetrated the organ.

Ultrasound Routine

Ultrasound allows us to locate contours, regions of interest of the organ and determine the position to adequately visualize the task to be performed on the organ. By means of a function we can determine if the ultrasound sensor is correctly positioned on the contact surface, this is possible thanks to equation 90.

$$\pi_1 \pi_2 + \pi_2 \pi_1 \text{.....(90)}$$

A plane is calculated with three points $X_{us_{1,2,3}}$ on the face of the probe

$$\pi_1 = X_{us1} \wedge X_{us2} \wedge X_{us3} \text{.....(91)}$$

and a plane on the surface of the organ, which is also calculated with three points

$$\pi_2 = X_{kn1} \wedge X_{kn2} \wedge X_{kn3} \text{.....(92)}$$

Figure 6 illustratively shows the incidence planes for the test lead and the contact surface (Figure 7).

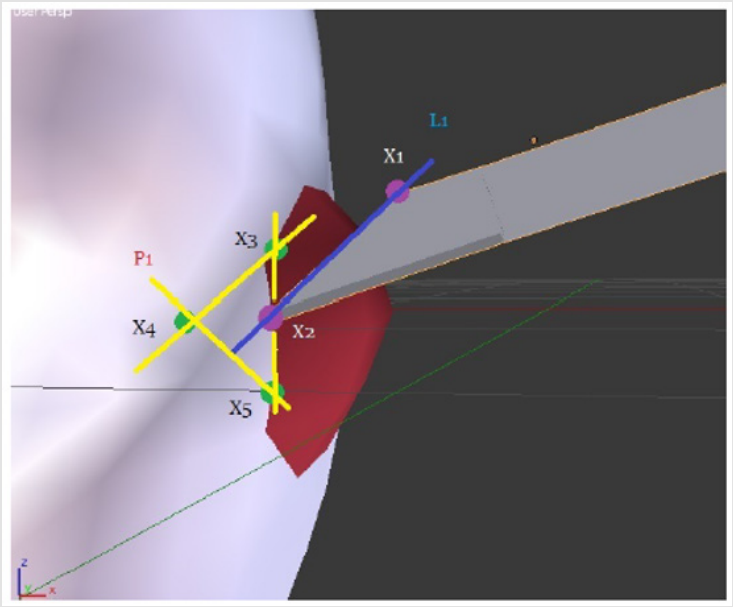

Figure 6: Plane and scalpel for surgical cutting.

 DOI: 10.26717/BJSTR.2025.61.009575

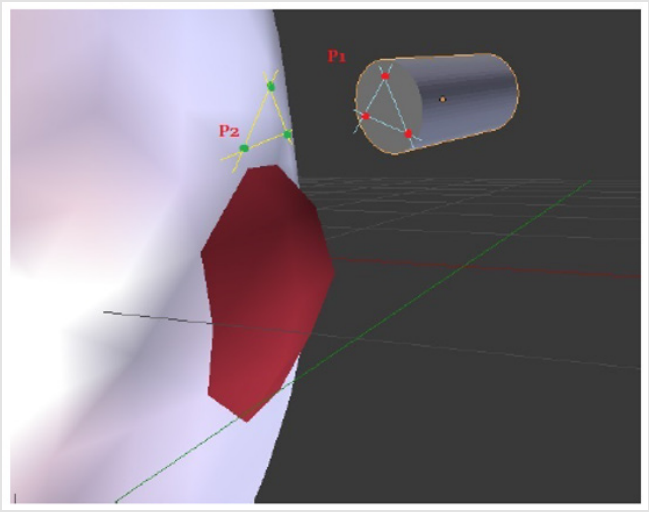

Figure 7: Representation of incidence between planes.

Suture

The last step after surgery is to suture the wound. For that some points are assigned around the wound, these points are marked and transferred to the virtual organ, Figure 8 shows some selected points and Figure 9 exemplifies the suture.

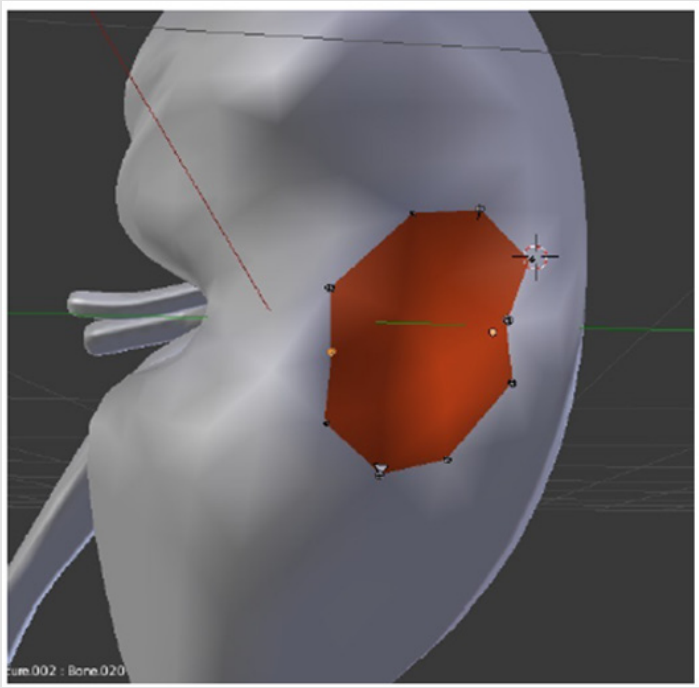

Figure 8: Selection of suture points.

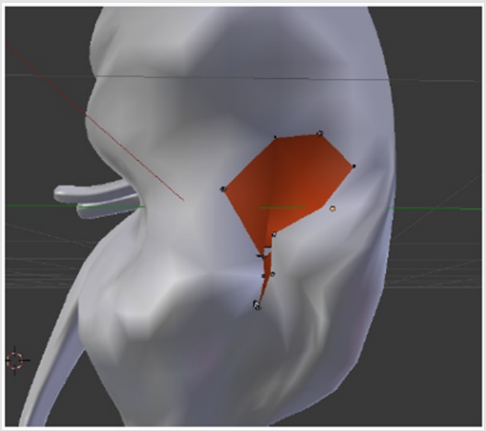

Figure 9: Suture closure.

First, we have to calculate T to move from one point to another:

$$T = 1 + e_{\infty}\left(\frac{X_i - X_j}{2}\right) \ldots\ldots\ldots\ldots\ldots\ldots (93)$$

Wherex_i and x_j are a pair of points to join, then we translate the needle to the opposite point, that is, a simple translation from one point to another opposite point.

$$Z_i = TZ_i \tilde{T} \ldots\ldots\ldots (94)$$

Design of the Quaternion Wavelet

Neural Network

The QWNN is first designed in a similar structure to the RBF described in [14] with an infinite impulse response (IIR) filter on the output, where the activation functions are $\psi_{Ql}[kT] \in H$ in pure quaternion form that incorporate daughter wavelets $\psi_{Qli,j,k}[kT]$, resulting from mother wavelet functions $\Psi_{Qli,j,k}[kT]$of type RASP1.

Recurrent QWNN

Thestructure of the recurrent QWNN is simpler because the coefficients $c_{Qp,n}[kT]$and $d_{Qp,n}[kT]$ are converted into weights, with recurrent delays as shown in Figure 10. Also, Sanchez Camperos in [14,15], names a similar recursive topology as the"state space model". The parameters $w_{Qp,l}[kT], b_{Qg,l}[kT]$and $a_{Ql}[kT] \in H$, are described in their vector form as:

$$W_Q[kT] = [w_{Qp,1}[kT] w_{Qp,2}[kT] \cdots w_{Qp,L}[kT]]^T,$$
$$B_Q[kT] = [b_{Qg,1}[kT] b_{Qg,2}[kT] \cdots b_{Qg,L}[kT]]^T,$$
$$A_Q[kT] = [a_{Q1}[kT] a_{Q2}[kT] \cdots a_{QL}[kT]]^T.$$

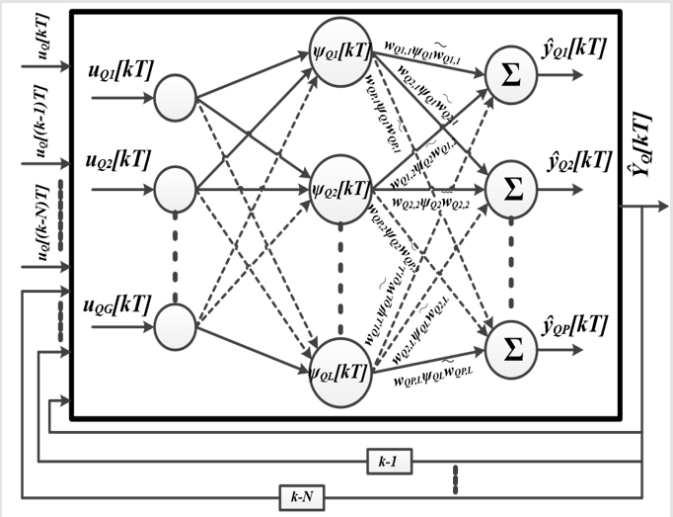

Figure 10: Recurrent QWNN.

The Least Mean Squares (LMS) method is used to minimize the cost functions, with their n^{-th}output from the network, as follows:

$$E_Q[kT] = \left[E_{Q1}[kT]E_{Q2}[kT]\cdots E_{QP}[kT]\right]^T$$

$$E_{Qp}[kT]=\frac{1}{2}\sum_k e_{Qp}^2[kT]$$

where

$e_{Qp}[kT] = y_{Qp}[kT] - \hat{y}_{Qp}[kT] \in H$ is the estimation error. E_Q[kT] is minimized using the gradient descent method, for each parameter by the chain rule:

$$\frac{\partial E_Q[kT]}{\partial W_Q[kT]}=\begin{bmatrix}\frac{\partial E_{Qp}[kT]}{\partial y_{Qp}[kT]}\frac{\partial W_{Qp,1}[kT]\varphi_1[kT]\tilde{w}_{Qp,1}[kT]}{\partial W_{Qp,1}[kT]}\\ \frac{\partial E_{Qp}[kT]}{\partial y_{Qp}[kT]}\frac{\partial W_{Qp,2}[kT]\varphi_2[kT]\tilde{w}_{Qp,2}[kT]}{\partial W_{Qp,2}[kT]}\\ \frac{\partial E_{Qp}[kT]}{\partial y_{Qp}[kT]}\frac{\partial W_{Qp,L}[kT]\varphi_l[kT]\tilde{w}_{Qp,L}[kT]}{\partial W_{Qp,L}[kT]}\end{bmatrix}$$

$$\frac{\partial E_Q[kT]}{\partial B_Q[kT]}=\begin{bmatrix}\frac{\partial E_{Qp}[kT]}{\partial y_{Qp}[kT]}\frac{\partial W_{Qp,1}[kT]\frac{\partial \varphi_1[kT]}{\partial b_{Qg,1}[kT]}\tilde{w}_{Qp,1}[kT]}{\partial W_{Qp,1}[kT]}\\ \frac{\partial E_{Qp}[kT]}{\partial y_{Qp}[kT]}\frac{\partial W_{Qp,2}[kT]\frac{\partial \varphi_2[kT]}{\partial b_{Qg,2}[kT]}\tilde{w}_{Qp,2}[kT]}{\partial W_{Qp,2}[kT]}\\ \frac{\partial E_{Qp}[kT]}{\partial y_{Qp}[kT]}\frac{\partial W_{Qp,L}[kT]\frac{\partial \varphi_L[kT]}{\partial b_{QG,L}[kT]}\tilde{w}_{Qp,L}[kT]}{\partial W_{Qp,L}[kT]}\end{bmatrix}$$

$$\frac{\partial E_Q[kT]}{\partial A_Q[kT]}=\begin{bmatrix}\frac{\partial E_{Qp}[kT]}{\partial y_{Qp}[kT]}\frac{\partial W_{Qp,1}[kT]\frac{\partial \varphi_1[kT]}{\partial a_{Qg,1}[kT]}\tilde{w}_{Qp,1}[kT]}{\partial W_{Qp,1}[kT]}\\ \frac{\partial E_{Qp}[kT]}{\partial y_{Qp}[kT]}\frac{\partial W_{Qp,2}[kT]\frac{\partial \varphi_2[kT]}{\partial a_{Qg,2}[kT]}\tilde{w}_{Qp,2}[kT]}{\partial W_{Qp,2}[kT]}\\ \frac{\partial E_{Qp}[kT]}{\partial y_{Qp}[kT]}\frac{\partial W_{Qp,L}[kT]\frac{\partial \varphi_L[kT]}{\partial a_{QG,L}[kT]}\tilde{w}_{Qp,L}[kT]}{\partial W_{Qp,L}[kT]}\end{bmatrix}$$

The partial derivatives corresponding to the descending gradient of A_Q, $B_Q$and $W_Q$are:

$$\frac{\partial E_{Qp}[kT]}{\partial y_{Qp}[kT]}=e_{Qp}[kT],$$

$$\frac{\partial \psi_1[kT]}{\partial a_{Q1}[kT]}=\left[0,\frac{\partial \psi_{Q_i}[kT]}{\partial a_{Q1i}[kT]}j,\ \frac{\partial \psi_{Q1j}[kT]}{\partial a_{Q1j}[kT]}j,\ \frac{\partial \psi_{Q1k}[kT]}{\partial a_{Q1k}[kT]}k\right],$$

$$\frac{\partial \psi_{Q1(i,j,k)}[kT]}{\partial a_{Q1(i,j,k)}[kT]}=-\frac{\gamma}{a_{Q1(i,j,k)}[kT]^2\left(\frac{\gamma}{a_{Q1(i,j,k)}[kT]}+1\right)^2}+\frac{\gamma}{a_{Q1(i,j,k)}[kT]^3\left(\frac{\gamma}{a_{Q1(i,j,k)}[kT]}+1\right)^3}$$

$$\frac{\partial \psi_1[kT]}{\partial b_{Q,g1}[kT]}\left[0,\frac{\partial \psi_{Q1i}[kT]}{\partial b_{Q,g1}[kT]}i,\frac{\partial \psi_{Q1j}[kT]}{\partial b_{Qg,1j}[kT]}j,\frac{\partial \psi_{Q1j}[kT]}{\partial b_{Qg,1j}[kT]}k\right],$$

$$\frac{\partial \psi_{Q1(i,j,k)}[kT]}{\partial b_{Qg,1(i,j,k)}[kT]}=\frac{-u_{Qg_{(i,j,k)}[kT]}+b_{Qg,1(i,j,k)}[kT]}{\gamma a_{Q1(i,j,k)}[kT]\left(\frac{\gamma}{a_{Q1(i,j,k)}[kT]}+1\right)^2}-\frac{-2u_{Qg_{(i,j,k)}[kT]}+2b_{Qg,1(i,j,k)}[kT]}{a_{Q1(i,j,k)}[kT]^2\left(\frac{\gamma}{a_{Q1(i,j,k)}[kT]}+1\right)^3},$$

$$\gamma=\sqrt{\left(\sum_{g=1}^{G}\left((\delta Qg,1)[kT]\right)^2\right)}$$

The minimization of E_Q[kT] in terms of weights W_Q[kT] is presented as a function of the Euler angles (ϕ, θand ψ), which has direct equivalence with the rotation matrix of the form *ZYZ* [10], or in terms of quaternions by *k*, *j* and *k*. Where we can decompose the rotation of a quaternion into three unit rotations $w_{Q\Phi}$, $w_{Q\theta}$and $w_{Q\Psi}$ as follows

$$_{Q\phi\theta\psi}=\frac{\partial w_Q[kT]\lambda\tilde{w}_Q[kT]}{\partial w_{Q\phi\theta\psi}[kT]}$$

$$\left[\frac{\partial w_{Q\phi}[kT]w_{Q\theta}[kT]w_{Q\varphi}[kT]\lambda\tilde{w}_{Q\phi}[kT]\tilde{w}_{Q\theta}[kT]\tilde{w}_{Q\theta}[kT]}{\partial w_{Q\phi\theta\psi}[kT]}\right]$$

where $J_{Q\phi\theta\psi}$represents the Jacobian of the weights in the next equation

$$j_{Q\phi\theta\psi}=\left[J_{Q\phi}\quad J_{Q\theta}\quad J_{Q\psi}\right]$$

$$=\left[\frac{\partial w_Q[kT]\lambda\tilde{w}_Q[kT]}{\partial w_Q\phi[kT]}\frac{\partial w_Q[kT]\lambda\tilde{w}_Q}{\partial w_{Q\theta}[kT]}\frac{\partial w_Q[kT]\lambda\tilde{w}_Q}{\partial w_{Q\psi}[kT]}.\right]$$

Considering each quaternion in rotations ϕ, θand ψwe can represent it in Euler form by:

$$w_{Q\phi}[kT]=\cos\left(\frac{\phi}{2}\right)+\sin\left(\frac{\phi}{2}\right)k=e^{\left(\frac{\phi}{2}k\right)},$$

$$w_{Q\theta}[kT]=\cos\left(\frac{\theta}{2}\right)+\sin\left(\frac{\theta}{2}\right)j=e^{\left(\frac{\theta}{2}j\right)},$$

$$w_{Q\psi}[kT]=\cos\left(\frac{\psi}{2}\right)+\sin\left(\frac{\psi}{2}\right)j=e^{\left(\frac{\psi}{2}j\right)},$$

Each component of the Jacobian is defined as as follows

$$j_{Q\phi}=\frac{\partial w_Q[kT]\lambda\tilde{w}_Q[kT]}{\partial w_Q\phi[kT]}$$

$$=\frac{\partial e\left(\frac{\Phi}{2}k\right)w_{Q\theta}[kT]w_{Q\psi}[kT]\tilde{w}_Q\psi[kT]\tilde{w}_{Q_\theta}[kT]e\left(-\frac{\Phi}{2}k\right)}{\partial w_{Q\phi}[kT]}$$

Vectors A_Q, B_Q and W_Q are updated, in proportion to the learning coefficients μ_a μ_b and μ_w as follows: the gradients ΔA_Q, ΔB_Q and ΔW_Q are vectors for the update, where each element has three quaternions and the real part is the function of the gradient on i, j and k respectively for each element of the vector A and B.

$$\Delta A_Q(r)(kT)=\frac{\partial E_Q[kT]}{\partial A_Q[kT]}=\begin{bmatrix}e_{Qp}[kT]J_{aQ1\phi\theta\psi}\\e_{Qp}[kT]J_{aQ2\phi\theta\psi}\\.\\.\\.\\e_{Qp}[kT]J_{aQ1\phi\theta\psi}\end{bmatrix}$$

$$\Delta B_Q(r)(kT)=\frac{\partial E_Q[kT]}{\partial B_Q[kT]}=\begin{bmatrix}e_{Qp}[kT]J_{bQg,1\phi\theta\psi}\\e_{Qp}[kT]J_{bQg,2\phi\theta\psi}\\.\\.\\.\\e_{Qp}[kT]J_{bQg,1\phi\theta\psi}\end{bmatrix}$$

Therefore, the updating of $A_Q$and $B_Q$as a function of pure quaternions is:

$$Bo\left[(k+1)T\right]=B_Q\left[kT\right]+\mu_B\Delta B_Q\left(r\right)\left[kT\right],$$

$$A_Q\left[(k+1)T\right]=A_Q\left[kT\right]+\mu_A\Delta A_Q\left(r\right)\left[kT\right].$$

In the weights, W, the real part of each element of the vector constitutes the gradient at ϕ, θ, and ψ

$$\Delta W_Q(r)(kT)=\frac{\partial E_Q[kT]}{\partial W_Q[kT]}=\begin{bmatrix}e_{Qp}[kT]J_{wQp,1\phi\theta\psi}\\e_{Qp}[kT]J_{wQp,2\phi\theta\psi}\\.\\.\\.\\e_{Qp}[kT]J_{wQp,1\phi\theta\psi}\end{bmatrix}$$

The weight can be updated in two ways, in the first method we can update the angles (ϕ, θand ψ) with the real part of the gradient and perform the quaternionic operations to obtain W_Q[(k + 1)T] in the following way:

$$\begin{bmatrix}\Phi\left[(k+1)T\right]\\\theta\left[(k+1)T\right]\\\psi\left[(k+1)T\right]\end{bmatrix}=\begin{bmatrix}\Phi\left[kT\right]\\\theta\left[kT\right]\\\psi\left[kT\right]\end{bmatrix}+\mu w\Delta W_Q\left(r\right)\left[kT\right],$$

subsequently, the quaternion is updated in all three axes,

$$w_Q\left[(k+1)T\right]=w_{Q\Phi}\left[(k+1)T\right]w_{Q\theta}\left[(k+1)T\right]w_{Q\Psi}\left[(k+1)T\right]$$

The second weight update method is the quaternionic product between the weights and the real part of the gradient ($\mu_W\Delta W_Q$(r)[kT]), as shown:

$$\Delta w_Q\left[kT\right]=\Delta w_{Q\Phi}\left[kT\right]\Delta w_{Q\theta}\left[kT\right]\Delta w_{Q\Psi}\left[kT\right],$$

$$w_Q\left[(k+1)T\right]=w_Q\left[kT\right]\Delta w_Q\left(r\right)\left[kT\right]\cdot$$

Both update methods are equivalent, however, the implementation of the first one has an advantage because it normalizes the quaternion in each iteration and the second one would require an additional normalization stage.

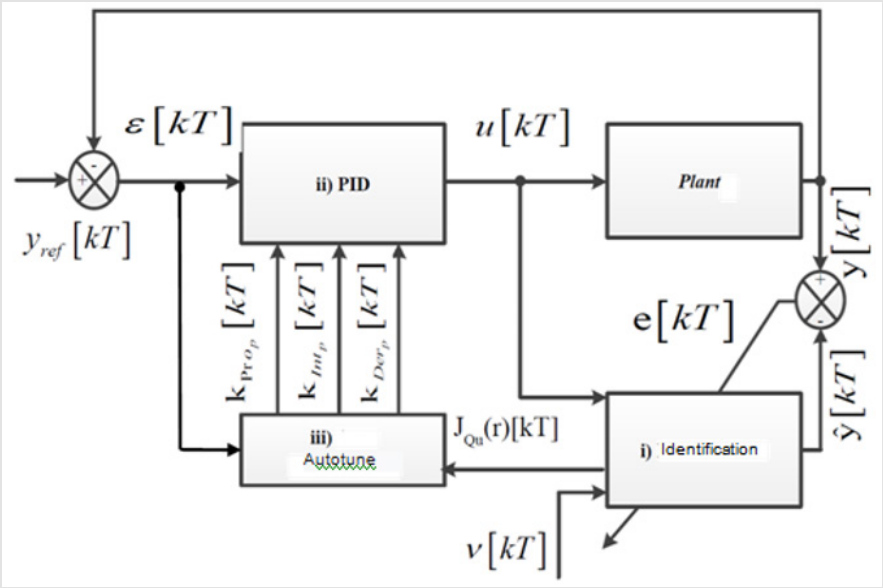

Figure 11: Adaptive PID Controller.

Adaptive PID Controller

The indirect adaptive PID controller (Figure 11), has three stages:

1. System identification (QWNN), its inputs are the estimation error e[kT] and the control signal u[kT], the estimated plant $y^{\wedge}$[kT] and the Jacobian J_{Qu}[kT] are the outputs.

2. The discrete PID controller has as input ε[kT] which is error and output signal u[kT].

3. Auto-tuning in this stage the gains k_{Pro}[kT], k_{Int}[*kT*] and k_{Der}[*kT*], are updated. The discrete PID control [4], which leads to the reference y_{refp}[kT], is presented by the Equation 40; with ε_p[kT] = y_{ref-p}[kT] – y_p[kT] as the error. The gains k_{Prop}[kT], k_{Intp}[kT] and k_{Derp}[kT], correspond to the control law and are modified by the QWNN,

$$u_P[(k+1)T] = u_P[kT] + kp_{rop}[kT]\left[\varepsilon_p[kT] - \varepsilon_p[(k-1)T]\right] + k_{Derp}[kT]\left[\varepsilon_p[kT] - 2\varepsilon_p[(k-1)T] + \varepsilon_p[(k-2)T]\right] + k_{Intp}[kT]\varepsilon_p[kT]$$

The updating of parameters does not require a device model, because the QWNN approximates the plant and allows updating the coefficients k_{Prop}[kT], k_{Intp}[kT]and k_{Derp}[kT], as shown:

$$k_{\Pr op}[kT] = k_{\Pr op}[(k-1)T] + \Delta K_{p,}$$
$$k_{Intp}[kT] = k_{Intp}[(k-1)T] + \Delta K_{i,}$$
$$k_{Derp}[kT] = k_{Derp}[(k-1)T] + \Delta K_{d}\cdot$$

The cost function is calculated as for $E[kT] = \frac{1}{2}\varepsilon_p^2[kT]$

$$\Delta Kp = \mu_P \frac{\partial E_Q[kT]}{\partial Kp[kT]} = \mu_P \frac{\partial E[kT]}{\partial y[kT]} \frac{\partial y[kT]}{\partial u[kT]} \frac{\partial u[kT]}{\partial Kp[kT]}$$
$$= \mu P\varepsilon[kT] \frac{\partial y[kT]}{\partial u[kT]} \left[\varepsilon_p[kT] - \varepsilon_p[(k-1)T]\right],$$

$$\Delta K_i = \mu_I \frac{\partial E[kT]}{\partial y[kT]} = \mu_I \frac{\partial E[kT]}{\partial y[kT]} \frac{\partial y[kT]}{\partial u[kT]} \frac{\partial u[kT]}{\partial K_i[kT]}$$
$$= \mu_I \varepsilon_P[kT] \frac{\partial y[kT]}{\partial u[kT]} \varepsilon_P[kT],$$

$$\Delta K_d = \mu_D \frac{\partial E[kT]}{\partial K_d[kT]} = \mu_D \frac{\partial E[kT]}{\partial y[kT]} \frac{\partial y[kT]}{\partial u[kT]} \frac{\partial u[kT]}{\partial K_d[kT]}$$
$$= \mu_D \varepsilon_P[kT] \frac{\partial y[kT]}{\partial u[kT]} [\varepsilon_P[kT] - 2\varepsilon_p[(k-1)T] + \varepsilon_p[(k-2)T]]$$

$$\frac{\partial y[kT]}{\partial u[kT]} \approx J_{Q\mu}(r)[kT] = \frac{\partial y_{Qp}[kT]}{\partial y_{Qp}[kT]}$$

$$= \begin{bmatrix} \dfrac{\partial \omega_{Qp,1}[kT]}{\partial w_{Qp,1}[kT]} \dfrac{\partial \psi_1[kT]}{\partial u_{Qg,1}[kT]} \tilde{w}_{Qp,1}[kT] \\ \dfrac{\partial \omega_{Qp,2}[kT]}{\partial w_{Qp,2}[kT]} \dfrac{\partial \psi_2[kT]}{\partial u_{Qg,2}[kT]} . \tilde{w}_{Qp,2}[kT] \\ . \\ . \\ . \\ \dfrac{\partial \omega_{Qp,L}[kT]}{\partial w_{Qp,L}[kT]} \dfrac{\partial \psi_L[kT]}{\partial u_{Qg,L}[kT]} . \tilde{w}_{Qp,L}[kT] \end{bmatrix}$$

Where

$$\frac{\partial \varphi_{Ql(i,j,k)}[kT]}{\partial u_{Qg,1(i,j,k)}[kT]} = \frac{uQ_{g(i,j,k)}[kT] - bQ_{g,l(i,j,k)}[kT]}{\gamma a_{Ql(i,j,k)}[kT]\left(\frac{\gamma}{a_{Ql(i,j,k)}[kT]^2}+1\right)^2} - \frac{-2uQ_{g(i,j,k)}[kT] + 2bQ_{g,l(i,j,k)}[kT]}{a_{Ql(i,j,k)}[kT]^2\left(\frac{\gamma}{a_{Ql(i,j,k)}[kT]^2}+1\right)^3}$$

The constants μ_p, μ_I, and μ_D are the learning rates of the discrete PID controller gains.

Robot Maneuvers in Liver Surgery

For the investigation to be a success, we apply the previously explained concepts in real time. The objective is to make some kinds of trajectories in Cartesian space through the MOTOMAN robot (implant of an object, cutting, ultrasound and suture evaluation maneuvers). The user gave the desired points with their respective coordinates and orientations, which would be interpolated in geometric algebra. The next information are the results of these experiments. Figure 12 shows two forms of implantation in the liver. The first one (a and b) is a kind of sewing called US trajectory and the second one (c and d) is MSLERP interpolation. Both pairs of trajectories are displayed in Cartesian plane (b and d) and over the liver (a and c).In Figure 13, the US Trajectory interpolation exercise takes 12 maneuver points with 50 points between samples, having a smoother movement. In this way, we can determine a trajectory with good resolution and coordinated movements in a single mathematical operator (motors) (Table 3). To evaluate the Inverse Kinematics, Table 3 shows us the percentage of errors of position and rotation of the final effector with respect to the desired position, these being very low, so it can be said that the precision is acceptable for a surgical intervention (Figure 14). In Figure 14, the MSLERP interpolation is implemented in a 2D view, with the blue polygon being the implant. The trajectories are curved and smooth, allowing a better grip between the object and the organ (Table 4). As in the previous case, Table 4 shows us the percentage of error in position and rotation of the end effector and in the same way, the results show that inverse kinematics works for any type of trajectory, having favorable results. Figure 15 presents the cutting path. It is observed that the points were interpolated to obtain a smoother trajectory. We can visualize the points generated by the planes on the surface of the test model, to establish the cut penetration function. The interpolation of points for the cut is carried out with 50 points per section, which allows us to obtain 250 points for the entire execution task. The ultrasound trajectory is presented in Figure 16. We can visualize that the trajectory of the points is taken based on the previous maneuver. The trajectory in the ultrasound movement is established by taking the previous planes and we generate pairs of planes like those established in the cutting sequence (Figure 17). Finally, Figure 18 presents the suture trajectory, where there are 5 trajectories (red, green, dark blue, sky blue and purple) with three reference points each in 3D Cartesian plane with millimeters scale. For real time, Figure 18 shows the main steps where the MOTOMAN realized the previous trajectories. The step was in milliseconds scale avoiding overlaps and delays in each interpolation. An object whose design matched the size and shape of a kidney was used for movements that were the closest thing to the reality of a surgical procedure.

Table 3: Error of US trajectory.

	Percentage Error					
	x	y	z	R_x	R_y	R_z
1	0	0	0.0005	0.001	0.0018	0.0049
2	0	0.0013	0.0022	0.0008	0.0061	0.0043
3	0	0.0021	0.0005	0.0007	0.0105	0.0014
4	0.0006	0.0025	0.0005	0.0008	0.0035	0.0021
5	0.0003	0.0017	0.0016	0.0004	0.0079	0.0002
6	0.0001	0.0032	0.0005	0.0005	0.0079	0.0026
7	0.0005	0.0024	0.0011	0.0011	0.0061	0.0029
8	0.0005	0.0012	0	0.0007	0.0079	0.0005
9	0.0001	0.0008	0.0011	0.0009	0.0079	0.0015
10	0	0.0004	0.0011	0.0006	0.0044	0.0015
11	0.0005	0.003	0.0006	0.0007	0.007	0.0028
12	0.0003	0.0019	0.0011	0.0007	0.0088	0.0052

DOI: 10.26717/BJSTR.2025.61.009575

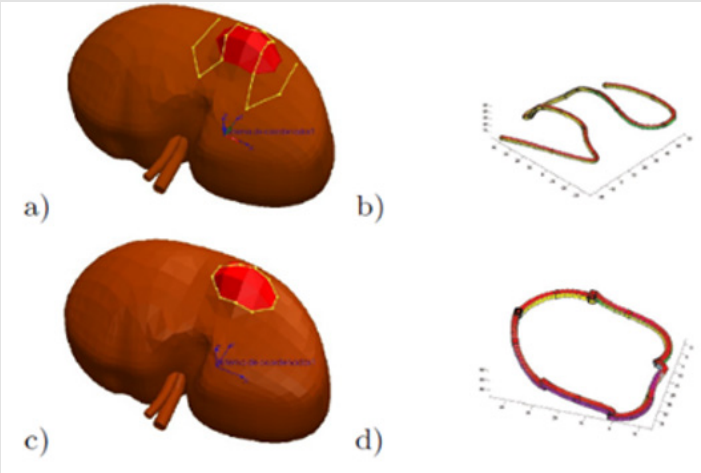

Figure 12: Trajectories in a kindle human model [2].

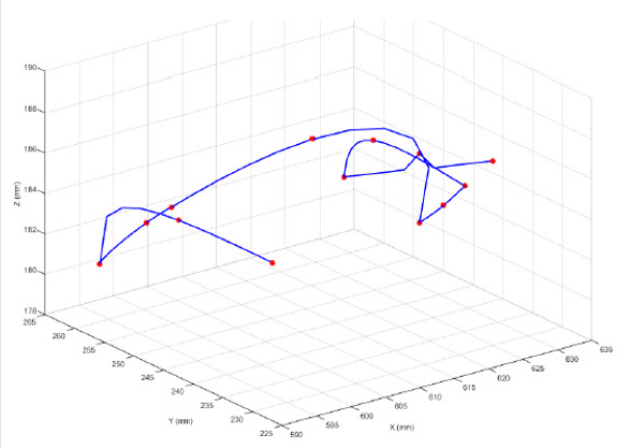

Figure 13: Graphic implant for US Trajectory.

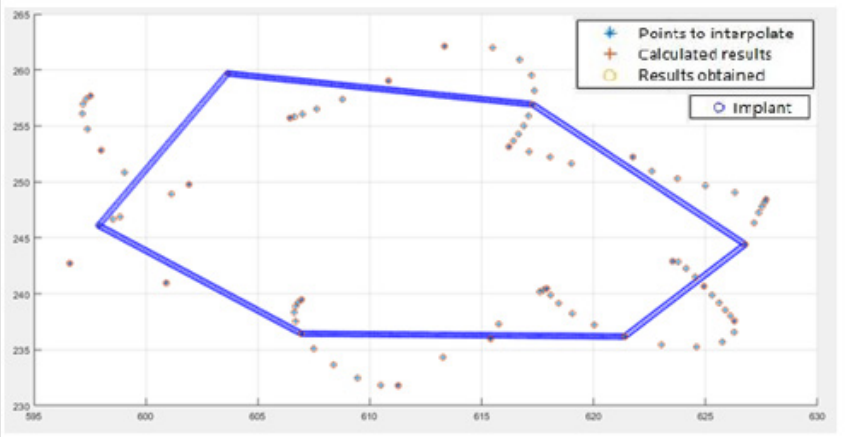

Figure 14: Graphic implant X-Y for MSLERP.

DOI: 10.26717/BJSTR.2025.61.009575

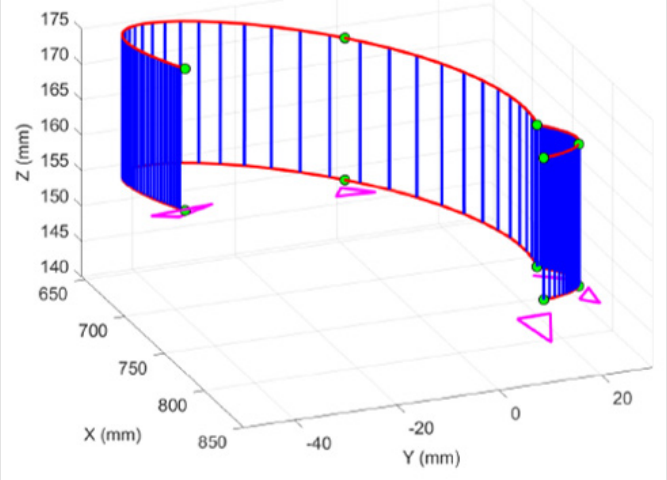

Figure 15: Cutting maneuver.

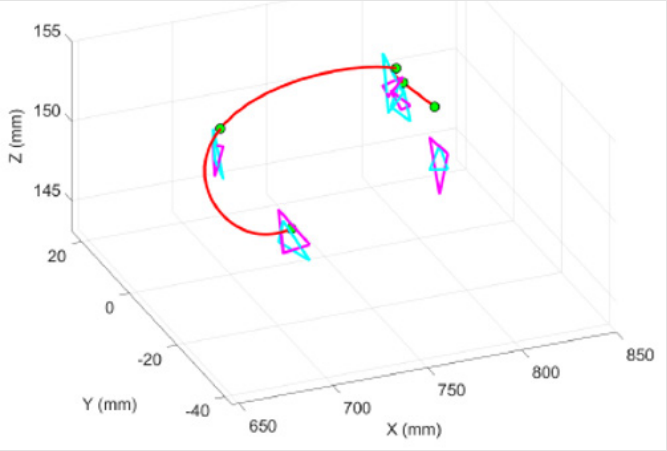

Figure 16: Ultrasound evaluation maneuver.

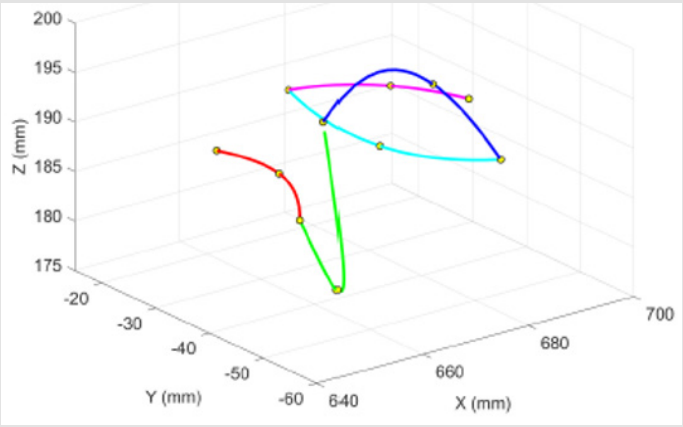

Figure 17: Suture evaluation maneuver.

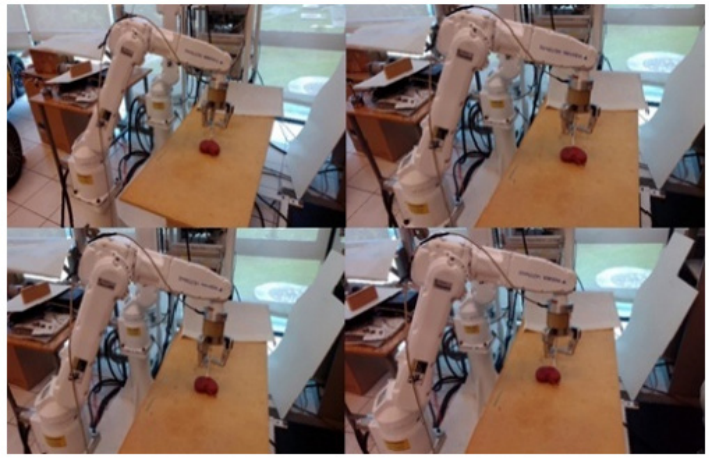

Figure 18: "MOTOMAN" robot in operation.

Table 4: Error of MSLERP.

	Percentage Error					
	x	y	z	R_x	R_y	R_z
1	0.0003	0.0004	0.0005	0.0012	0.0105	0.0043
2	0.0001	0.0008	0.0021	0.0006	0.0097	0.0043
3	0.0001	0.0008	0.0011	0.0006	0.0026	0.0037
4	0.0003	0.0004	0.0011	0.0009	0.0061	0.0041
5	0.0006	0.002	0.0005	0.0011	0.0105	0.0047
6	0.0006	0.0027	0.0005	0.0009	0.0105	0.0057
7	0.0009	0.0015	0.0011	0.0011	0.0088	0.0049
8	0.0006	0.0012	0.0006	0.0008	0.0044	0.0041
9	0.0003	0.0023	0.0022	0.0008	0.0061	0.0052
10	0.0006	0.0019	0.0028	0.0014	0.0096	0.0058
11	0.0003	0.0027	0.0006	0.0008	0.0097	0.0055
12	0.0004	0.0031	0.0011	0.0009	0.0097	0.0054
13	0.0001	0.0016	0.0005	0.0004	0.0088	0.0035
14	0.0004	0.0023	0.0011	0.0007	0.0026	0.0028
15	0.0001	0.0012	0.0016	0.0003	0.007	0.0018
16	0.0003	0.0016	0.0016	0.0005	0.0105	0.0026
17	0.0001	0.0004	0.0005	0.0007	0.0079	0.0017
18	0.0003	0.0004	0.0005	0.0009	0.007	0.0018

Wavelet Neural Network forAdaptive PID Control inHaptics

This section presents the experimental results using the Phantom Omni haptic device (Figure 19), a robotic mechanism with six degrees of freedom, underactuated dynamics, and low tribology. It is designed for interaction with dynamic environments that provide kinesthetic force feedback.In medical robotics, robot manipulators are often controlled via a haptic interface, as these robots are typically heavy and challenging to operate with high precision. Additionally, haptics play a crucial role in telepresence robotics.To evaluate our Quaternion Wavelet Neural Network (QWNN) for Adaptive PID Control in Haptics, we attempt to track a nonlinear trajectory in 3D space. The user's applied pressure affects the impedance of the haptic device, which must be considered in our estimation and control strategy to ensure smooth trajectory tracking. Furthermore, our controlled haptic device can be connected to manipulate a robotic arm, as described in the previous section. The results are presented for the QWNN PID controller and a classical PID controller. The parameters that describe the experiment are found in Table 5.

Table 5: Experimental parameters.

Parameter	Value	Parameter	Value
W	0 a π	μ_{Pro}	0.1
A	0 a 10	μ_{Int}	0.1
B	-10 a 10	μ_{Der}	0.1
$k_{pro_p(t)}$	[5 5 5]	$N^0\ neurons\,(L)$	6
$k_{\text{int}_p(t)}$	[0.05 0.05 0.05]	$Mopther\ \ wavelet\,(\varphi_i(t))$	RASPI
$k_{Der_p(t)}$	[1 1 1]	Epochs	5
μW	0.5	Sampling Period (T)	$\approx 0.001s$
μA	0.75	N^0 Inputs Q recurrency	1
μB	1	N^0 recurrences	3

Figure 19: Phantom Omni haptic device.

The experiment was carried out with a sampling period of T = 0.001 (1khz sampling frequency), for a simple trajectory, which allows the movement of the final effector over Euclidean space. Inverse kinematics is applied to find joint control reference. Figure 20 shows the behavior in Euclidean space, it is observed that the QWNN has a better convergence to the desired reference. Figure 20 shows the behavior of Euclidean space for each coordinate axis (x,y,z), It is observed that the three axes have a better position convergence in the QWNN PID, which is more notable in the *y* axis by the gravitational component. Figure 21 shows the trajectory in Euclidean space for each coordinate axis (x,y,z), It is observed that the three axes have a better position convergence in the QWNN PID, which is more notable in the *y* axis by the gravitational component. Figure 22 shows the control signal for each joint, it can be seen that the QWNN PID control signal has areas where it is greater, which determines greatrobustness in tracking the trajectory.

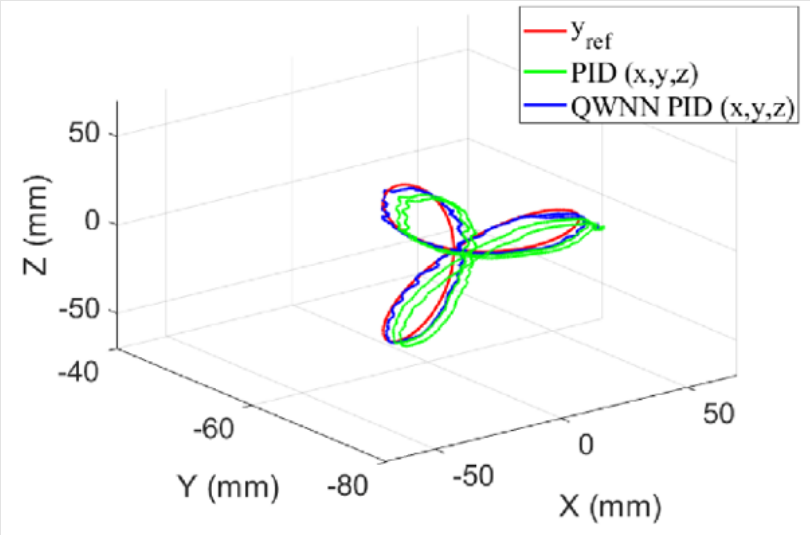

Figure 20: Trajectory in the Euclidean workspace.

DOI: 10.26717/BJSTR.2025.61.009575

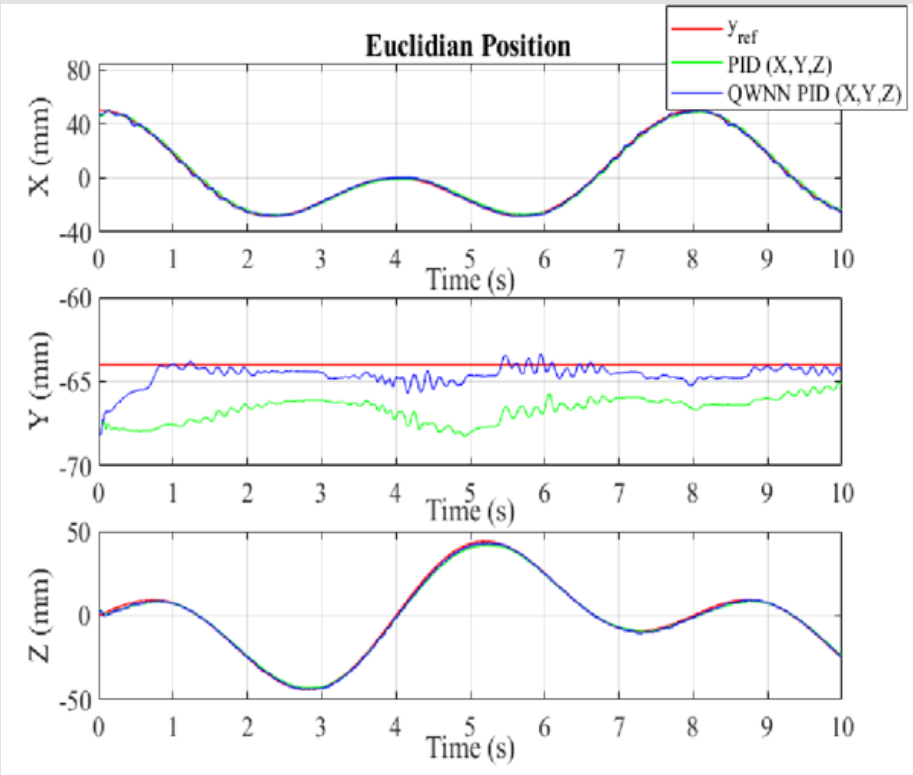

Figure 21: Trajectory in (x,y,z).

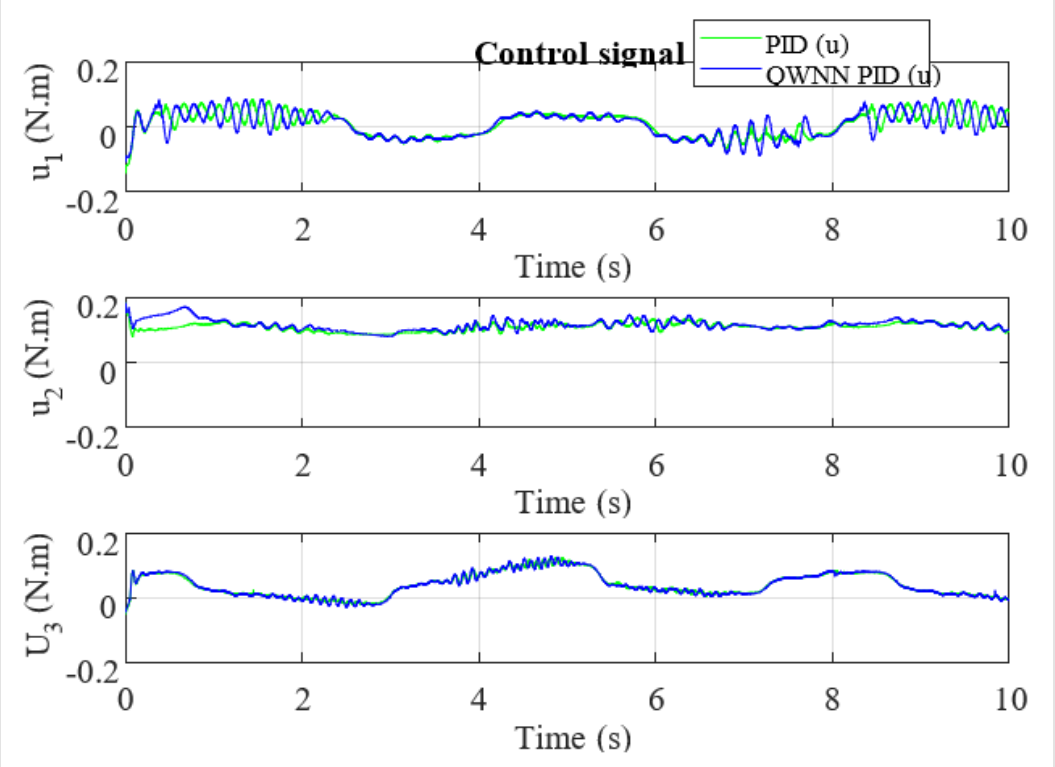

Figure 22: Control signal $u(t) = J^T\tau$.

Conclusions

As demonstrated in this work, as well as in previous theses and articles, geometric algebra enables the efficient modeling of geometric entities. Although it involves a higher level of complexity, it offers the advantage of representing a wide range of problems using geometric elements. When applied to inverse kinematics, it proves to be a valuable tool for determining the necessary rotation values for each robot joint to achieve the desired position. Furthermore, the results highlight how geometric algebra intuitively and effectively models the employed system. The findings also reveal the strong relationship between geometric algebra and the kinematics of robotic models, particularly in calculating the required tracking for a manipulator to perform tasks such as suturing, implant placement, ultrasonic maneuvers, and cutting. These applications can be extended to various types of medical operations. Additionally, this paper presents an adaptive PID control law with an indirect architecture for controlling motion paths in an open-chain haptic system. The plant's identification is achieved using a Quaternion Wavelet Neural Network, which optimally tunes the PID values, ensuring precise movement within the workspace. Finally, the experimental results confirm that the proposed method achieves a high degree of accuracy, with an error percentage close to zero. Thus, we conclude that the proposed approach effectively applies geometric methods to solve various medical robotics tasks, significantly contributing to the potential for autonomous execution in the future.

References

1. Grassmann H (1877) Der ort der Hamilton'schenquaternionen in der ausdehnungslehre. Mathematische Annalen 12(3): 375-386.
2. Clifford WK (1882) Mathhematical Papers. Macmillan and Company.
3. Hestenes D (1966) Space- time algebra 1(6). New York: Gordon and Breach.
4. Bayro-Corrochano E (2018) Geometric Algebra Applications Vol. I: Computer Vision, Graphics and Neurocomputing. Springer.
5. Perwass C, Edelsbrunner H, Kobbelt L, Polthier K (2009) Geometric algebra withapplications in engineering (4). Berlín: Springer.
6. Hildenbrand D (2013) Foundations of geometric algebra computing. Springer, Berlín, Heidelberg, pp. 179-188.
7. Dorst L, Fontijne D y Mann S (2010) Geometric algebra for computer science: an object- oriented approach to geometry. Elsevier.
8. Ureña O (2020) Aceleración de algoritmos de álgebra geométrica implementados en GPU. PhD Thesis, CINVESTAV.
9. KJ Aström, T Hägglund (2009) Advanced PID Control. Madrid: Pearson.
10. MW Spong, S Hutchinson, M Vidyasagar (2020) Robot Modeling and Control, (2nd Edn). John Wiley and Sons.
11. T Yoshikawa (1990) Fundations of Robotics: Analisys and control. Cambridge, Mass: MIT Press.
12. D Powell, MK O'Malley (2012) The task-dependent efficacy of shared control haptic guidance paradigms," IEEE Transactions on Haptics 78: 208-219.
13. Hamilton WR (1866) Elements of Quaternions, Longmans Green, London. Chelsea, New York.
14. EN Sanchez-Camperos, AY Alanıs, AG Loukianov (2008) DiscreteTime High Order Neural Control, Trained with Kalman Filtering. Verlag, Berlin, Heidelberg: Springer.
15. Garza-Burgos A, Bayro-Corrochano E, Del-Valle-Padilla J (2017) Geometric Intuitive Techniques for Human Machine Interaction in Medical Robotics, p. 1-19.

ISSN: 2574-1241

DOI: 10.26717/BJSTR.2025.61.009575

Eduardo Bayro- Corrochano. Biomed J Sci & Tech Res

Submission Link: https://biomedres.us/submit-manuscript.php