A bio-inspired quaternion local phase CNN layer with contrast invariance and linear sensitivity to rotation angles

E. Ulises Moya-Sánchez\textsuperscript{a,b}, Sebastián Xambó-Descamps\textsuperscript{b}, Abraham Sánchez Pérez\textsuperscript{c}, Sebastián Salazar-Colores\textsuperscript{d}, Jorge Martínez-Ortega\textsuperscript{e}, Ulises Cortés\textsuperscript{b}

\textsuperscript{a} Gobierno del Estado de Jalisco y, Universidad Autónoma de Guadalajara, Independencia 55, Guadalajara 44100, Jalisco, México
\textsuperscript{b} Universitat Politècnica de Catalunya and Barcelona Supercomputing Centre, Jordi Girona 1–3, Omega, 08034, Barcelona, Spain
\textsuperscript{c} Gobierno del Estado de Jalisco, Independencia 55, Guadalajara, 44100 Jalisco, México
\textsuperscript{d} Facultad de Informática, Universidad Autónoma de Querétaro, Querétaro, México

\section{Introduction}

Previous works indicate that pattern recognition in a deep learning classification process does not guarantee the required equivariance and invariance properties [13,14,26,27].

Convolutional Neural Networks (CNN), which were inspired by neuroscience principles, capture three essential properties of the primary visual cortex V1: extraction of features from two-dimensional data; spatial localization of the receptive field; and shift equivariance in the position of the feature [18].

It is to be stressed, however, that visual neurons of the mammalian visual system are also resilient in front of equivariant transformations such as local invariance response to some changes in lighting and rotations [5,18,19].

To emulate as much as possible the early visual system and add more equivariant capacities to the CNN, we have been inspired by some physiological experimental results like those reported in [22], including the stronger response to oriented lines and edges (local even and odd signals, respectively) [5,19].

To move forward toward such functionality, namely to be able to extract contrast invariant local features and to predict rotation angles of images, one idea is to embrace suitable hypercomplex geometric methods, in the sense of [16,20,24,35].

More specifically, in this work we propose what we call a Quaternion Local Phase CNN Layer (Q9 in this paper) and explore the performance boost it provides when it is stacked in front of a very simple CNN. What we find is that the compound net features contrast invariance and the capability of ascertaining rotation angles.

The remainder of the paper is structured as follows. In Section 2 we report on recent related work and in Section 3 we recall background notions and notations needed later on. Of particular relevance is Section 3.4, in which we specify our approach to the application of quaternionic local phases to image processing. Our main theoretical contribution, namely the quaternionic local phase layer Q9, is the subject of Section 4. The behaviour of Q9 is also illustrated with images. The experimental setup, including the data used, is described in Section 4 and a summary of the results, conclusions and future outlook can be found in Sections 6 and 7. The material on quaternionic phases used in the paper, particularly in Section 3 and 4, is recalled in the Appendix A.

\section{Related work}

The use of hypercomplex NN has been mostly in shallow cases: [4,8,9,32].
On the other hand, [17,36] formulate and implement quaternion convolution, batch normalization, weight initialization, and backpropagation for a deep quaternion CNN. The main difference between these references and our work is that we use the local quaternion phases. In addition, we propose only one CNN layer, not a full quaternion CNN. In fact, our experimental setup and CNN architectures have different objectives. Their CNNs tests aim at classification tasks and do not consider contrast invariance or linear response to rotations. Our CNNs architecture is designed to favor a systematic comparison with the performance of a traditional convolutional layer. To note that we care for maintaining the accuracy across all contrast degradation levels and not merely for getting the best accuracy for one degradation.

3. Background

3.1. Equivariance and invariance

The term equivariance tends to be used to refer to the predictable way in which features of a signal change under certain transformations: [14]. More formally, a function \( f: X \rightarrow X \) is equivariant with respect to a group of transformations \( G \) of \( X \) if

\[
f(g(x)) = g(f(x)),
\]

for all \( x \in X \) and \( g \in G \).

For instance, one of the most important equivariant properties of the mammalian visual system (measured by [22]) is its equivariance under rotations. Subsequently, many authors have been interested in extending this property to NNs and CNNs: [7,13,14,21].

On the other hand, by invariance of a feature map \( f: X \rightarrow Y \) we understand (see [16]) that

\[
f(g(x)) = f(x)
\]

for all \( x \in X \) and \( g \in G \).

3.2. Bio-inspired CNN

For convenience, in this section we borrow from [31, Section II.B; see also the references therein] the main properties of the V1 simple cells:

1. The V1 cells form the first layer of the hierarchical cortical processing.
2. They are insensitive to the color of the light falling on their receptive fields.
3. These neurons respond vigorously only to edges (odd-signal) and lines (even-signal) at a particular spatial direction through the orientation columns.

In this work we use four main bio-inspired tools: the local phase, (Quaternion) Gabor functions, the HSV color space and the artificial neural networks. Altogether, the proposed layer emulates some properties of V1 cells, as detecting edges (odd-signal) and lines (even-signal) at a particular direction. In addition, our approach exhibits translation equivariance and contrast invariance in object recognition.

3.3. Notations

The notations, structure and terminology we are going to use are as in [31, Sections II.C and IV.C]. We define 1D (resp. 2D) multivectorial signals as \( \mathcal{C} \) maps \( U \rightarrow \mathcal{g} \) from an interval \( U \subset \mathbb{R} \) (a region \( U \subset \mathbb{R}^2 \)) into a geometric algebra \( \mathcal{g} \) (see [35]). For \( \mathcal{g} = \mathbb{R} \) (\( \mathcal{g} = \mathbb{C}, \mathcal{g} = \mathbb{H} \)) we say that the signal is scalar (complex, quaternionic). For technical reasons, we also assume that signals are in \( L^2 \) (that is, the modulus is square-integrable).

<table>
<thead>
<tr>
<th>Table 1</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Synopsis of notations and conventions.</strong></td>
</tr>
<tr>
<td><strong>Symbol</strong></td>
</tr>
<tr>
<td>( x^l )</td>
</tr>
<tr>
<td>( x = x^l )</td>
</tr>
<tr>
<td>( w = w^l )</td>
</tr>
<tr>
<td>( b = b^l )</td>
</tr>
<tr>
<td>( \sigma )</td>
</tr>
<tr>
<td>( C )</td>
</tr>
<tr>
<td>( F )</td>
</tr>
<tr>
<td>( Q )</td>
</tr>
<tr>
<td>( Q9 )</td>
</tr>
</tbody>
</table>

Table 1 presents a synopsis of the notation, structure and meaning of the most important variables and functions occurring in a deep CNN. See also [2,18,33].

3.4. Quaternion local phase

A quaternion local phase array \( \mathbf{x}_Q(x, y) \in \mathbb{H} \) is associated to a 2D signal \( \mathbf{x} = x(x, y) \in \mathbb{R} \) (where \( x, y \in U \), \( U \) a region of \( \mathbb{R}^2 \)). The computation of \( \mathbf{x}_Q \) is inspired on the version of the Quaternion Fourier Transform (QFT) proposed in [11]. To explain how it works, it is convenient to introduce a few notations.

A Gaussian filter \( g(u_1, u_2) \) in the frequency domain, rotated by an angle \( \alpha \) and with standard deviations \( \sigma_1 \) and \( \sigma_2 \) in the \( u_1 \) and \( u_2 \) directions, respectively, is defined by the formula

\[
g(u_1, u_2) = \exp \left( -\frac{u_1^2}{2\sigma_1^2} - \frac{u_2^2}{2\sigma_2^2} \right)
\]

where \( u_1' = u_1 \cos(\alpha) + u_2 \sin(\alpha) \) and \( u_2' = -u_1 \sin(\alpha) + u_2 \cos(\alpha) \).

Given a function \( f(u_1, u_2) \), which we regard as a filter in the frequency domain, we will also use the expression

\[
\mathbf{x} \ast f(u_1, u_2) = \mathcal{F}^{-1}(\mathcal{F}(\mathbf{x}) \ast f(u_1, u_2)),
\]

where \( \mathcal{F}^{-1} \) denotes the Fourier transform from the space variables \((x, y)\) to the frequency variables \((u_1, u_2)\).

Now letting \( \omega_1 \) and \( \omega_2 \) denote positive constant frequencies, \( \mathbf{x}_Q \) is defined as follows:

\[
\mathbf{x}_Q = \mathbf{x}_{\text{QCC}} + i\mathbf{x}_{\text{QGC}} + j\mathbf{x}_{\text{QCS}} + k\mathbf{x}_{\text{QHS}}
\]

where

\[
\mathbf{x}_{\text{QCC}} = \mathbf{x} \ast g(u_1, u_2) \cos(\sigma_1 \omega_1 u_1) \cos(\sigma_2 \omega_2 u_2)
\]

\[
\mathbf{x}_{\text{QGC}} = \mathbf{x} \ast g(u_1, u_2) \sin(\sigma_1 \omega_1 u_1) \cos(\sigma_2 \omega_2 u_2)
\]

\[
\mathbf{x}_{\text{QCS}} = \mathbf{x} \ast g(u_1, u_2) \cos(\sigma_1 \omega_1 u_1) \sin(\sigma_2 \omega_2 u_2)
\]

\[
\mathbf{x}_{\text{QHS}} = \mathbf{x} \ast g(u_1, u_2) \sin(\sigma_1 \omega_1 u_1) \sin(\sigma_2 \omega_2 u_2)
\]

As a result, we have rotated Quaternion Gabor filters (see [11]) which also depend on the parameters \( \alpha, \sigma_1, \sigma_2, \omega_1, \omega_2 \). The Gabor filters are well known bio-inspired feature extractors: [11,28–30,32]. See Fig. 1 for an illustration.

Now based on the Eqs. (A.7)–(A.10) of Appendix A, we can rewrite the equation 5 as follows:

\[
\mathbf{x}_Q = \|\mathbf{x}_Q\| e^{i\phi_0} e^{k\gamma u_1} e^{\phi_2 u_2},
\]

1 See [3] for a Clifford Fourier Transform generalization.
where \( \phi_x, \psi_x, \theta_x \) are the phases of the unit quaternion \( x_0/x_0 \).

It is important to remark that the frequency domain convolution is scarcely used on CNN. There are two main reasons for this: i) for \( N \) training weights, the complexity is of order \( N^2 \); ii) the Fourier transform is a global transformation, and it is not possible to correctly localize the features [6]. Previous works such as [34] try to use the cosine transform in order to avoid the local problem and [15] propose a Spectrum pooling and return to the spatial domain to avoid both drawbacks for 3D data.

4. Quaternion local phase layer \( \text{Q9} \)

The proposed convolution layer, \( \text{Q9} \), a quaternion local phase layer, is described as follows:

1. Create a Hue Saturation Value (HSV) array using the quaternion phases, magnitude and constant \( 1 \):
   \[
   \phi_{HSV} = (\phi_x, |x_0|, 1). \tag{7}
   \]
   \[
   \theta_{HSV} = (\theta_x, |x_0|, 1). \tag{8}
   \]
   \[
   \psi_{HSV} = (\psi_x, |x_0|, 1). \tag{9}
   \]
   where \( 1 \) is an \( [m, n] \) array with all entries equal to 1.
2. Transform the HSV images into RGB images:
   \[
   \phi_{HSV} \rightarrow \phi_{RGB}. \tag{10}
   \]
   \[
   \theta_{HSV} \rightarrow \theta_{RGB}. \tag{11}
   \]
   \[
   \psi_{HSV} \rightarrow \psi_{RGB}. \tag{12}
   \]
   according to the standard conventions (cf. [1], p. 304). See Fig. 2 for an illustration. Remark that \( \phi_{RGB} \) enhances vertical lines (yellow), that \( \theta_{RGB} \) enhances horizontal lines (yellow), and that \( \psi_{RGB} \) features a dark blue all over the image. Fig. 3 illustrates the nine feature maps of \( \phi_{RGB}, \theta_{RGB} \) and \( \psi_{RGB} \).

In our experimental setup, we notice that the phase \( \psi_{RGB} \) is sensitive to rotation. See Fig. 4 for an illustration of the three channels of \( \psi_{RGB} \) and their response to rotations.

In the Table 2 we can compare a simple convolution layer with the proposed layer \( \text{Q9} \).

5. Data and experimental setup

We have used the MNIST dataset, see [25] and CIFAR10 [23], in two main experimental setups: contrast invariance classification and regression for equivariance response measurement.

---

**Fig. 1.** Quaternion filters from Eq. (5). From left to right \( g(u_1, u_2)\cos(\sigma_1 x u_1)\cos(\sigma_2 y u_2), g(u_1, u_2)\sin(\sigma_1 x u_1)\cos(\sigma_2 y u_2), g(u_1, u_2)\cos(\sigma_1 x u_1)\sin(\sigma_2 y u_2), g(u_1, u_2)\sin(\sigma_1 x u_1)\sin(\sigma_2 y u_2) \).

**Fig. 2.** From left to right: Original image, \( \phi_{RGB}, \theta_{RGB}, \psi_{RGB} \).

**Fig. 3.** In the first column \( \phi_{RGB}, \theta_{RGB} \) and \( \psi_{RGB} \). Each row presents the three components of each RGB phase representation.

**Fig. 4.** In the first column \( \psi_{RGB} \) with 0, 15, 30 and 45 degrees. Each row depicts the components of each \( \psi_{RGB} \).

---

**Table 2**

Comparison of the main characteristics of a standard convolutional layer \( C \) and our layer \( \text{Q9} \). Although \( \text{Q9} \) is not itself trainable, the learning is carried out, in any particular application, by the net to which it is coupled.

<table>
<thead>
<tr>
<th>Convolution Layer Comparison</th>
<th>Characteristics</th>
<th>( C )</th>
<th>( \text{Q9} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hyperparameters</td>
<td>3</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Kernel shape</td>
<td>( a, , \sigma_1, , \sigma_2, , \omega_1, , \omega_2 )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Convolution domain</td>
<td>space</td>
<td>frequency</td>
<td></td>
</tr>
<tr>
<td>Output domain</td>
<td>space</td>
<td>space</td>
<td></td>
</tr>
<tr>
<td>Padding</td>
<td>zero</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>Nonlinear function</td>
<td>ReLU</td>
<td>arctan, arcsin</td>
<td></td>
</tr>
<tr>
<td>Layer position</td>
<td>Any</td>
<td>First hidden</td>
<td></td>
</tr>
<tr>
<td>Learning</td>
<td>Yes</td>
<td>No</td>
<td></td>
</tr>
</tbody>
</table>

---

5.1. Contrast invariance classification

Table 3 shows the main characteristics of the degradation labels and the contrast values. The contrast reduction process was done by first normalizing the pixel values from 0 to 1, and then rescaling the pixel values to an interval \([d, 1] \subset [0, 1]\), which amounts to a contrast of \((1 - d)\). For instance, in the case \( d = 1 \) the pixel values are rescaled to the interval \([0.3, 1]\), which amounts to a 70% contrast.
Table 3
Characteristics of datasets (top). Degradation labels and the corresponding contrast values used in our experiments (bottom).

<table>
<thead>
<tr>
<th>Characteristics</th>
<th>MNIST Values</th>
<th>CIFAR10 Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training set</td>
<td>50,000</td>
<td>40,000</td>
</tr>
<tr>
<td>Validation set</td>
<td>10,000</td>
<td>10,000</td>
</tr>
<tr>
<td>Test set</td>
<td>10,000</td>
<td>10,000</td>
</tr>
<tr>
<td>Total of images</td>
<td>70,000</td>
<td>60,000</td>
</tr>
<tr>
<td>Image shape</td>
<td>[28,28,1]</td>
<td>[32,32,3]</td>
</tr>
<tr>
<td>Degradation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$d_0$</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>$d_1$</td>
<td>70</td>
<td></td>
</tr>
<tr>
<td>$d_2$</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>$d_3$</td>
<td>10</td>
<td></td>
</tr>
</tbody>
</table>

Table 4
CNN Architectures for MNIST and CIFAR10 classification task. The C layer and all the F layers have a ReLU as activation function.

<table>
<thead>
<tr>
<th>Layer</th>
<th>Characteristics</th>
<th>Shape</th>
<th>Parameters</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0) Input Layer</td>
<td></td>
<td>Input Image</td>
<td>[28,28,1]</td>
<td>[28,28,9]</td>
</tr>
<tr>
<td>(1) First layer</td>
<td>C</td>
<td>Q9</td>
<td>[3, 3, 9]</td>
<td>[28,28,4]</td>
</tr>
<tr>
<td>(2) FL</td>
<td></td>
<td>[32,32,9]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(3) F 1</td>
<td></td>
<td>[256]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(4) F 2</td>
<td></td>
<td>[128]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(5) SMAX</td>
<td></td>
<td>[10]</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Fig. 5. Example of degradation levels. From left to right: 100% contrast ($d_0$); 70% ($d_1$); 30% ($d_2$); and 10% ($d_3$).

Table 5
Experimental scheme for the classification task.

<table>
<thead>
<tr>
<th>Experimental set up</th>
<th>Trained and Contrast</th>
<th>Tested</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$d_0$</td>
<td>$d_0$, $d_1$, $d_2$, $d_3$</td>
</tr>
<tr>
<td></td>
<td>$d_1$</td>
<td>$d_0$, $d_1$, $d_2$, $d_3$</td>
</tr>
<tr>
<td></td>
<td>$d_2$</td>
<td>$d_0$, $d_1$, $d_2$, $d_3$</td>
</tr>
<tr>
<td></td>
<td>$d_3$</td>
<td>$d_0$, $d_1$, $d_2$, $d_3$</td>
</tr>
</tbody>
</table>

Table 6
Experimental scheme for rotation for each image.

<table>
<thead>
<tr>
<th>Experimental rotation set up</th>
<th>Set</th>
<th>Angles (Degree)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Train</td>
<td>[0, 46] steps=3</td>
</tr>
<tr>
<td></td>
<td>Validation</td>
<td>[1, 46] steps=3</td>
</tr>
<tr>
<td></td>
<td>Test</td>
<td>[2, 46] steps=3</td>
</tr>
</tbody>
</table>

Table 7
CNN Architectures for a regression task to get the rotation angle. The C and all the F layers have a sigmoid as activation function.

<table>
<thead>
<tr>
<th>Layer</th>
<th>Characteristics</th>
<th>Shape</th>
<th>Parameters</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0) Input Layer</td>
<td></td>
<td>Image</td>
<td>[28,28,1]</td>
<td>[28,28,9]</td>
</tr>
<tr>
<td>(1) First layer</td>
<td>C</td>
<td>Q9</td>
<td>[3, 3, 9]</td>
<td>[28,28,4]</td>
</tr>
<tr>
<td>(2) FL</td>
<td></td>
<td>[32,32,9]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(3) F 1</td>
<td></td>
<td>[1000]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(4) F 2</td>
<td></td>
<td>[64]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(5) F 3</td>
<td></td>
<td>[32]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(6) F 4</td>
<td></td>
<td>[8]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(7) SG</td>
<td></td>
<td>[1]</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

5.2. Regression for rotational response

We created 100 data-sets (one for each image), by rotating according to Table 6. We trained, validated and tested with the first 100 (nonzero digits) from the MNIST and the first 100 images from CIFAR10. Fig. 6 presents an example of rotation data.

We have used the CNN architecture presented in Table 7. We trained both CNNs for regression with the same hyperparameters: learning rate, 0.001; number of epochs, 1000; loss function, mean

Fig. 6. Rotation examples for regression experimental setup from MNIST and CIFAR10. The value of the angle in degrees.
squared error; optimizer, rmsprop; activation function, sigmoid; and for Q9 we chose \( \alpha = 0, \sigma_1 = 4, \sigma_2 = 4, \omega_1 = 0.5, \omega_2 = 0.5 \).

### 6. Results and analysis

#### 6.1. Contrast invariance (classification)

Tables 8 and 9 show the best accuracy reached in the classification tasks by the C and Q9 nets for MNIST and CIFAR10, respectively. Each row shows the performance of both CNNs with different testing samples for a given training sample. As mentioned previously, samples are grouped according to the different contrast levels. The values in bold indicate the best result. As seen in Table 8, the superiority of Q9 over C is conspicuous on MNIST images for its high and virtually uniform accuracy. The gains for CIFAR10 images are also quite uniform and, except for the diagonal slots \( d_0-d_0, d_1-d_1 \) and \( d_2-d_2 \), very appreciable (Table 9). In sum, Q9 exhibits invariance under contrast degradation while C does not.

#### 6.2. Rotational response (regression)

The first comparison we can try on the performance of C and Q9 is the behaviour of the loss function. Fig. 7 is an error plot to show the evolution of the mean value and the variance (as vertical line of each point) of the loss function, respectively, for both CNNs across epochs. These figures show that the loss function converges faster, and that it is more stable, for Q9 than for C. Faster convergence and higher stability are desirable properties that have a bearing, in particular, on a more efficient use of computing resources.

Figs. 8 and 9 show the rotation predictions over the 100 images of each CNN at epoch 1000. In addition a mean value of the prediction and its variance are represented by an error curve. We see that the CNN with the normal convolution layer C has more outliers in the ranges (0–20) and (25–45) degrees than Q9. In other words, the dispersion of the predictions is substantially lower for Q9 than for C. This result motivates us to explore in a future work whether Q9, or some variation of it, achieves an equivariant response with respect to plane rotations.
7. Conclusions and outlook

The main objective of the article has been to propose a new bio-inspired quaternionic layer, $Q_9$, based on the kernel of the quaternionic Fourier transform proposed by Thomas Bülow [11] and to compare its performance to that of a regular convolutional layer in two types of computational experiments, one focused on classification tasks and another on rotation prediction by regression. Our layer has 9 channels and insures an invariant response to high contrast changes with almost constant performance in classification tasks even when the CNN is trained with quite different contrasts. The $Q_9$ layer also features a faster learning of image rotation angles than a regular convolution layer. We believe that the proposed layer could be useful to recognize or classify images in outdoor scenarios with haze with no data augmentation. One current drawback is that the C layer has a better performance for CIFAR10 images in the case when the test set has the same degradation level (with the exception of $d_3$) than the training set.

Let us mention two lines of future inquiry: To seek how to insure that the hyperparameters of the layer are learned automatically, and to explore how to obtain an equivariant response to image rotations.

Declaration of Competing Interest

I recognize and disclosing that we don’t have any financial and other conflicts of interest that might bias this work. We acknowledge all financial support for the work. As a result, I declare that we don’t have any conflict of interest related to this work.

Acknowledgment

The authors would like to thank to CONACYT and the Barcelona supercomputing Center. Sebastián Salazar-Colores (CVU 477758) would like to thank CONACYT (Consejo Nacional de Ciencia y Tecnología) for the financial support of his PhD studies under Scholarship 285651. Eduardo Ulises Moya-Sánchez, Jorge Martínez-Ortega and Ulises Cortés are members of the Sistema Nacional de Investiga- gadores (SNI) de México.

Appendix A. Quaternions

The quaternion algebra $H$ is a four dimensional real vector space with basis $i$, $j$, $k$,

$$H = R \oplus Ri \oplus Rj \oplus RK$$

(A.1)

endowed with the bilinear product (multiplication) defined by Hamilton’s relations, namely

$$i^2 = j^2 = k^2 = ij = -ji = k$$

(A.2)

As it is easily seen, these relations imply that

$$ij = -ji = k, \quad jk = -kj = i, \quad ki = -ik = j.$$  

(A.3)

The elements of $H$ are called quaternions, and $i$, $j$, $k$, quaternionic units. By definition, a quaternion $q$ can be written in a unique way in the form

$$q = a + bi + cj + dk, \quad a, b, c, d \in R.$$  

(A.4)

Its conjugate, $\bar{q}$, is defined as

$$\bar{q} = a - (bi + cj + dk).$$

(A.5)

and its modulus, $|q|$, by $|q| = \sqrt{\bar{q}q}$. 

A polar representation of $q$ is defined by

$$q = |q|e^{i\phi}e^{j\psi}e^{k\theta}.$$  

(A.6)

where

$$(\phi, \theta, \psi) \in [-\pi, \pi) \times [-\pi/2, \pi/2] \times [-\pi/4, \pi/4]$$

are the phases of $q$ as defined in [10] and [11]. For a unit q, the phase $\psi$ is found to be

$$\psi = -\arcsin(2(bc - ad)/2).$$  

(A.7)

If $\psi = \pm \frac{\pi}{4}$, set $\theta = 0$ and

$$\phi' = \frac{1}{2} \arctan \left( \frac{-2(cd + ab)}{a^2 + b^2 - c^2 - d^2} \right).$$  

(A.8)

Else,

$$\phi' = \frac{1}{2} \arctan \left( \frac{2(cd + ab)}{a^2 + b^2 + c^2 - d^2} \right).$$

(A.9)

$$\theta = \frac{1 + \pi}{2} \arctan \left( \frac{-2(ab + cd)}{a^2 + b^2 - c^2 - d^2} \right).$$

(A.10)

With this, $e^\phi e^{j\psi}e^{k\theta} = \pm q$ and if it is $-q$, set $\phi = \phi' + \pi \mod 2\pi$.

Here, arc$tan$ 2 is the four quadrant $\arctan$ (see [11]).

References