Robust Constant Curvature Curve Communications with Complex and Quaternion Neural Networks

Anders Buvarp ¹ and Lamine Mili ²

¹Virginia Tech
²Affiliation not available

October 31, 2023

Abstract

The concept of Digital Twin has recently emerged, which requires the transmission of a massive amount of sensor data with low latency and high reliability. Analog error correction is an attractive method for low-latency communications; hence, in this paper, we propose the use of complex-valued neural networks and Quaternionic Neural Networks (QNNs) to decode analog codes. Furthermore, we propose mapping our codes to the baseband of the frequency domain to enable easy time and frequency synchronization as well as to mitigate frequency-selective fading using robust estimation theory. This is accomplished by applying inverse Discrete Fourier Transform (DFT) modulation, which achieves a significant reduction in hardware complexity, power, and cost as compared to our previously proposed analog coding scheme. Additionally, we introduce a scaled version of our previous analog codes that enables statistical signal processing, something we have not been able to achieve until now. This achieves significant noise immunity with drastic performance improvements at low Signal-to-Noise Ratios (SNR) and a small loss at high SNR.
Robust Constant Curvature Curve Communications with Complex and Quaternion Neural Networks

Anders M. Buvarp Graduate Student Member, IEEE, and Lamine Mili Life Fellow, IEEE

Abstract—The concept of Digital Twin has recently emerged, which requires the transmission of a massive amount of sensor data with low latency and high reliability. Analog error correction is an attractive method for low-latency communications; hence, in this paper, we propose the use of complex-valued neural networks and Quaternionic Neural Networks (QNNs) to decode analog codes. Furthermore, we propose mapping our codes to the baseband of the frequency domain to enable easy time and frequency synchronization as well as to mitigate frequency-selective fading using robust estimation theory. This is accomplished by applying inverse Discrete Fourier Transform (DFT) modulation, which achieves a significant reduction in hardware complexity, power, and cost as compared to our previously proposed analog coding scheme. Additionally, we introduce a scaled version of our previous analog codes that enables statistical signal processing, something we have not been able to achieve until now. This achieves significant noise immunity with drastic performance improvements at low Signal-to-Noise Ratios (SNR) and a small loss at high SNR.


I. INTRODUCTION

The concept of a Digital Twin has been proposed for sixth generation mobile communications systems (6G), with a virtual representation of the actual physical communications system including machine learning, optimization algorithms, and other component [1]. Internet of Everything (IoE) and EXtended Reality (XR) applications include human-robot collaboration, holoportation, telepresence, remote sensing, etc. A real-life environment could be scanned using a massive number of sensors where the output is transmitted to a remote computing environment and virtually reproduced in XR. Conversely, the real environment could be equipped with actuators and the operations performed in the XR could be carried out in the real-life environment. To minimize errors, such activities would require highly robust systems with low latency, hence the key implementation challenges of Digital Twin. XR and IoE architectures are the minimization of latency, the assurance of reliability and system robustness.

Digital communication systems are based on the separation principle, where the source and channel coding steps are split and the code block lengths are large [2]. This introduces significant complexity and hence latency in transmission. Although perhaps quaint, old-fashioned and certainly contrary to current state-of-the-art technology, analog coding methods have been shown to exhibit optimal and minimal latency [3, 4]. With significantly less latency, Joint Source Channel Codes (JSCC) maps real-valued samples straight to the channel input, avoiding buffer delays, encoding overhead, etc., and therefore exhibit extremely low latency [5–8].

Our previous contributions in the area of JSCC include the Constant Curvature Curve Tubing Codes (C3T) [3]. Limited by a power constraint, these knot theory-based [9] codes densely pack tubes inside a hypersphere in an arbitrary n-dimensional space, including odd dimension. These C3T have the ability to meet the latency requirements of a twins-based architecture and other proposed applications for 6G. The benefits of C3T codes are low complexity and that performance degrades gracefully and therefore are not subject to the threshold effect [4] prevalent in digital codes. Furthermore, with uniformly distributed sources, C3T has superior performance at low Signal-to-Noise Ratio (SNR) levels and exceeds the performance of the maximum likelihood decoder by almost 8 dB. We implemented various neural network-based decoders in arbitrary dimensions and evaluated the performance at very low SNR levels. High-dimensional C3T codes are necessary to obtain adequate performance at very low SNR.

Furthermore, we analyzed and compared our short block length C3T code with long block length digital codes and found that given a Signal-to-Distorsion Ratio (SDR) of 20 dB and a SNR of 3.0 dB, a digital code would require a block length close to 700 at code rate 0.72 to achieve an error rate of $10^{-6}$. For Ultra-Reliable Low-Latency Communications (URLLC) applications, the requirement to wait nearly 700 samples before transmission would be prohibitive. Our minimum mean square error C3T decoder yields an accuracy of 99.96%.

In wireless communications, channel estimation is the most essential part of the design because the channel may exhibit frequency-selective fading that severely distorts the received signal. Channel estimators are typically based on the classical parametric estimation theory, which follows Ronald Fisher [10], who assumed that the probability distribution of the data is known and therefore developed the optimal estimation based on this assumption. However, the latter is usually violated in practice. John Tukey realized in 1960, while surveying contaminated distributions, that conventional estimation methods, while optimal under Gaussian assumptions, are highly sensitive to a small fraction of contamination [11]. This weakness led Peter Huber [12] to develop the theory of robust statistics, and later extended by Frank Hampel [13]. Until recently, only a few papers considered channel estimation with robust statistics.

In this work, our main contributions include the following:

Anders M. Buvarp and Lamine Mili are with Bradley Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA 24061 USA. Email: {abuv,lmili}@vt.edu.

Anders M. Buvarp is a Davenport Leadership Scholar.
1) Reduction of cost, power, and complexity. Our previous work on C3T would require very complex and costly RF hardware to implement. To reduce power consumption, energy, and cost, we now define the C3T codes as complex numbers in the baseband of the frequency domain. This allows for simplified time/frequency synchronization and the introduction of robust channel equalization processing in case of frequency-selective fading. Furthermore, it allows for the mapping of multiple C3T code words (i.e., several sensor readings) to the frequency domain, thus reducing the DFT processing overhead per code word. There are potentially other benefits that have not yet been considered.

2) Equalization via robust torus projection. We propose a robust method based on the Huber M-estimator for equalization in the case of small-scale fading due to multipath delay spread resulting in frequency-selective fading. Since the C3T code is defined in the frequency domain, the fading results in one of more frequency bins becoming outliers. The Huber estimator downweights these outliers during the equalization process, yielding better quality estimates. The transmitted signal is restored by a process we call Torus Projection (TP).

3) Complex-valued and quaternion neural networks. We train complex-valued C3T (CV-C3T) decoders using Multi-Layer Perceptrons (MLP) taking into account the phase between pairs cos/sin. We consider pairs of complex numbers as quaternions and train Quaternion Neural Networks (QNN) accordingly.

The remainder of this paper is organized as follows. In the next section, we provide the background for constant curvature codes, robust estimation, and quaternions. The system model is presented in Section III. Our robust estimator for decoding C3T is introduced in Section IV. Our complex-valued decoder in Section V-A and the quaternion decoder is described in Section V-B. We validate the performance of our decoders through numerical examples in Section VI. Section VII contains our conclusions and proposed future research.

We use the following notations: $\mathbb{R}$ is the set of real-valued numbers, $\mathbb{C}$ is the set of complex-valued numbers, $\mathbb{H}$ is the set of quaternion numbers, $\mathbb{E}^n$ is the n-dimensional Euclidean space, $\mathcal{E}$ is the expectation function, $\mathcal{O}$ is the Bachmann-Landau big O notation, $C^r$ is an $r$ times continuously differentiable parametric curve, $T^{n/2}$ is a n/2-dimensional flat torus, and $|\cdot|$ is the absolute value. The notation $x \sim \mathcal{U}(a, b)$ means a random variable drawn from the uniform distribution over the interval $[a, b]$ and $x \sim N(\mu, \sigma^2)$ represents the Gaussian distribution with mean $\mu$ and variance $\sigma^2$. The complex-valued and quaternion Gaussian distributions are presented as $n \sim \mathcal{N}_C(0, \sigma^2_C)$ and $n \sim \mathcal{N}_H(0, \sigma^2_H)$, respectively. $\Re$ and $\Im$ represent the real and imaginary parts of a complex-valued or a quaternion number, respectively.

II. BACKGROUND

A constant curvature curve of class $C^r$ is represented by the n-dimensional vector $x(\alpha)$ defined as [14],

$$x(\alpha) = [r_1 \cos(\omega_1 \alpha), \sin(\omega_1 \alpha), r_2 \cos(\omega_2 \alpha), r_2 \sin(\omega_2 \alpha), \ldots, r_{n/2} \cos(\omega_{n/2} \alpha), r_{n/2} \sin(\omega_{n/2} \alpha)]^T,$$

where $\alpha \in I$ is a stretched version of the source sample $s$. $I$ is an interval of the real line, $r_i$ and $\omega_i$, are the radii and frequency parameters. In short, the source sample is mapped to a manifold in $n$-dimensional Euclidean space $\mathbb{E}^n$. Constant curvature curves in even dimensions are a class of curves that are geodesics on flat tori $T^{n/2}$, which are closed Riemannian manifolds that have zero Gaussian curvature everywhere. The flat tori are embedded in a hypersphere of dimension $n-1$, thus constraining the transmission power. The sphere is embedded in an $n$-dimensional Euclidean space. We can translate, rotate, or scale the image of the curve, thus modifying the parameters of the map accordingly, without changing the generalized curvatures [15]. Geodesics on flat tori must all have constant generalized curvature [14].

By selecting $\omega_i = i \omega_1$, $i = 2, 3, \ldots, n/2$, we obtain curves that repeat themselves and hence form an $n$-dimensional knot, and thus achieving curvatures that are all greater than zero [14]. To prevent the end points of the curve to join we scale $\alpha$ by a factor 0.9. This scale factor was obtained experimentally to maximize the decoder performance. Too much scaling reduces the performance at high SNR. The smallest distance between folds of the curve is twice the global radius [9] of the curve given by

$$\rho^2(\Delta) = \frac{\sum_{i=1}^{n/2} r_i^2 (1 - \cos(\omega_i \Delta))^2}{\sum_{i=1}^{n/2} r_i^2},$$

where $\Delta$ is the distance between two points on the curve. This allows for a small ball radius [16, Definition 2] at any point of the curve to avoid self-intersections or tight bending that leads to the threshold effect [4].

We considered the curve length, $L_r = 2\pi \sqrt{\sum_{i=1}^{n/2} r_i^2}$, the tube cross-sectional volume, $V_{cs} = r^{n-1} \rho^{n-2}$, the bounding n-sphere volume, $V_b = C_n (1 + r \rho)^n$, and the constant, $C_n$, we obtained the associated set of radii, $r$, that maximize the tube density given by

$$D = L_r V_{cs} / V_b.$$  

Table I shows the resulting C3T radii, the global radius (i.e. the tube radius), and the maximized tube density.

<table>
<thead>
<tr>
<th>$n$</th>
<th>$r^*$</th>
<th>$\rho$</th>
<th>$D_{max}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>0.38618, 0.40925, 0.40228, 0.39142, 0.36800, 0.32634, 0.28696, 0.20993</td>
<td>0.72546</td>
<td>5.8045e-5</td>
</tr>
<tr>
<td>20</td>
<td>0.35251, 0.36758, 0.37378, 0.3705, 0.36659, 0.34508, 0.30498, 0.24627, 0.19969, 0.12894</td>
<td>0.72424</td>
<td>2.2944e-6</td>
</tr>
<tr>
<td>40</td>
<td>0.23259, 0.21674, 0.25316, 0.23812, 0.26905, 0.25430, 0.27287, 0.26388, 0.26076, 0.25428</td>
<td>0.71011</td>
<td>1.2154e-13</td>
</tr>
<tr>
<td>100</td>
<td>0.12382, 0.10848, 0.15717, 0.17994, 0.06367, 0.01646, 0.17460, 0.13555, 0.12845, 0.11500</td>
<td>0.67371</td>
<td>3.0959e-37</td>
</tr>
</tbody>
</table>

$a$ We presented the radii for $n \in \{4, 8, 12\}$ in [3].
A. Introduction to Robust Statistics

Robust estimation theory was developed to overcome the weakness of the classical Fisherian-parametric estimation theory, which considers the ideal assumptions made to be exact. However, in practice, these assumptions are violated. One common case is the presence of a fraction of data, termed outliers, that do not follow the assumed probability distribution; the actual distribution is said to be contaminated. An example of contamination in wireless systems is the impulsive noise [17]. Popularized by Huber [12], the contamination distribution model is expressed as

$$ G = (1 - \epsilon)F + \epsilon H, $$

(4)

where $H$ is the distribution of the contamination, $F$ is the assumed probability distribution, and $\epsilon \leq 1/2$ is the fraction of contamination. Robust estimators are those that exhibit a bounded bias and variance in the presence of contamination [12]. The largest fraction of contamination an estimator can deal with is its breakdown point [13]. Because the mean as the first moment is attracted by the tail of an asymmetric probability distribution and does not always exist when the tails become very thick (e.g., the Cauchy distribution), it is typically replaced by the median, which is the center of probability of the distribution. Therefore, the sample median is used instead of the mean as an estimator of location. Similarly, the standard deviation is replaced by a robust scale, for example, by the Median-Absolute-Deviation (MAD) from the median, which is defined as

$$ s = 1.4826 \times \text{median} |z_i - \text{median}(z_j)|. $$

(5)

Now, it is also desirable that a robust estimator be unbiased and has a high asymptotic efficiency at the assumed probability distribution, $F$, typically the Gaussian distribution. Recall that the asymptotic efficiency of an estimator is defined as the ratio between the smallest possible variance equal to the reciprocal of the Fisher information, termed the Cramér–Rao Lower Bound (CRLB), and the asymptotic variance of the normalized estimator.

An M-estimator is a generalization of the Gaussian maximum likelihood estimator used in wireless systems, with an objective function given by

$$ J(\theta) = \min_\theta \sum_{i=1}^m \rho \left( \frac{r_i}{s} \right), $$

(6)

and residuals $r_i = z_i - \hat{x}_i$, observations $z_i$ and our estimate $\hat{x}_i$. The Huber M-estimator is a combination of least squares and least absolute value estimator. In the location case and for the threshold $b$ equal to 1.5, this estimator has a high statistical efficiency in the Gaussian distribution and has a bounded bias under contamination, that is, it is robust to outliers with an asymptotic breakdown point of 50%. The outliers are down-weighted [18] using the iterative process shown in Fig. 1 and expressed as

$$ \hat{x}_{k+1} = \frac{\sum_{i=1}^m z_i q \left( \frac{r_i}{s} \right)}{\sum_{i=1}^m q \left( \frac{r_i}{s} \right)}. $$

(7)

B. Quaternions

Although quaternion mathematics has mainly been used in physics, computer graphics, and mechanics, it has also been used in wireless communications. The quaternion-valued space $\mathbb{H}$ extends the complex numbers $\mathbb{C}$ and a quaternion is defined as the quotient of two vectors in $\mathbb{R}^3$ [19]. Consider $a, b, c, d \in \mathbb{R}$, then a quaternion $q \in \mathbb{H}$ and its conjugate $q^* \in \mathbb{H}$ are defined as

$$ q = a + bi + cj + dk $$

(8)

$$ q^* = a - bi - cj - dk, $$

(9)

where $\mathbb{R}(q) = a$, and $\Im(q) = bi + cj + dk$, with the quaternion units $\{1, i, j, k\}$ abiding $i = jk = -kj$, hence quaternions are not commutative and thus do not satisfy the field axioms. Because $\mathbb{H}$ is noncommutative, it is not a field algebra like $\mathbb{R}$ and $\mathbb{C}$ and instead $\mathbb{H}$ is a division and non-zero ring that forms a four-dimensional vector space and an associative normed division algebra over the real numbers with $\{1, i, j, k\}$ as basis. The Hamilton product $(p \cdot q)$ is determined by the products of the basis elements and the distributive law and is defined as

$$ p \cdot q = (a_p a_q - b_p b_q - c_p c_q - d_p d_q) + (a_p b_q + b_p c_q + c_p a_q - d_p d_q) i + $$

$$ + (a_p c_q + b_p b_q + c_p a_q + d_p d_q) j + (a_p d_q - b_p c_q - c_p b_q + d_p a_q) k, $$

(10)

where the conjugate of the product can be decomposed as $(p \cdot q)^* = q^* \cdot p^*$.

Consider two mutually orthogonal complex values $x_0 \in \mathbb{C}$ and $x_1 \in \mathbb{C}$, using the Cayley-Dickson notation, $q = x_0 + x_1$. For a function $f(q) \in \mathbb{H}$, the gradient of $f$ with respect to $q$ is defined as

$$ \frac{\partial f}{\partial q} = \frac{1}{4} \left( \frac{\partial f}{\partial q_0} - \frac{\partial f}{\partial q_1} i - \frac{\partial f}{\partial q_2} j - \frac{\partial f}{\partial q_3} k \right), $$

$$ \frac{\partial f}{\partial q^*} = \frac{1}{4} \left( \frac{\partial f}{\partial q_0} + \frac{\partial f}{\partial q_1} i + \frac{\partial f}{\partial q_2} j + \frac{\partial f}{\partial q_3} k \right). $$

(11)
III. DFT-BASED C3T SYSTEM MODEL

Fig. 3 illustrates the three basic components of our C3T communications system; transmitter, channel and receiver. The transmitter consists of a stretching function \( \alpha(s) \) applied to a source sample \( s \in \mathbb{R} \), where \( s \sim U(a, b) \), which is followed by a folding function

\[ f : \alpha(s) \rightarrow x \quad (12) \]

This is a unitarian code block-length because only one source sample is required. An example of the encoded \( x(\alpha) \) for \( n = 20 \) and \( \alpha = \pi/3 \) can be seen in Fig. 2b. The \( n \)-dimensional vector, \( x(\alpha) \) is mapped to \( n/2 \) In-phase/Quadrature-phase (I/Q) values in the frequency domain, where each I/Q pair represents a narrowband subcarrier in the baseband and avoiding the DC bin. The resulting vector is converted to the time domain using the inverse DFT as shown in Fig. 3.

Our time-domain complex-valued Additive White Gaussian noise (AWGN) channel model can be expressed as

\[ y[k] = x[k] + w[k] , \quad (13) \]

where \( x[k] = IDFT\{x(\alpha)\} \), \( w(k) \sim \mathcal{N}_C(0, \sigma_w^2) \) and \( y[k] \) the received time-domain signal, CV-C3T. According to Parseval’s theorem, the power of a signal can be computed either in the frequency domain or the time domain. Since we defined the signal, \( x(\alpha) \), in the frequency domain, we calculate the power there. The power of a signal, \( x \), can be computed as the variance of \( x \), that is,

\[ P = \text{var}(x) = E[x^2] - E[x]^2 \quad (14) \]

All the elements of the vector \( x(\alpha) \) consists of \( \cos / \sin \)-pairs, hence \( E[x(\alpha)] = 0 \) and the average transmit power per \( \cos / \sin \)-pair can therefore be computed as

\[ P_{tx} = 2 \frac{n}{n/2} \sum_{i=1}^{n/2} r_i^2 , \quad (15) \]

where \( r_i \) are the radii of the \( \cos / \sin \) pairs of the curve \( x(\alpha) \). For a desired \( \text{SNR}_{dB} \) level of the AWGN channel and with transmit power, \( P_{tx} \), the noise power is given by

\[ \sigma_w^2 = P_{tx} 10^{-\frac{\text{SNR}_{dB}}{10} \frac{w}{w}} . \quad (16) \]

In our complex-valued simulation models, the noise power, \( \sigma_w^2 \), is distributed evenly over each I/Q pair. In this work we also perform simulations with quaternions where the quaternion noise is expressed as \( w \sim \mathcal{N}_H(0, \sigma_w^2) \) and the noise power in Eq. (16) is distributed uniformly over all four parts of the quaternion samples.

Consider impulse noise [17] with some noise power \( \sigma_{imp} \gg \sigma_w \), and following Huber’s contamination model, the impulse noise model is as follows:

\[ w \sim (1 - \epsilon)\mathcal{N}_C(0, \sigma_w^2) + \epsilon\mathcal{N}_C(0, \sigma_{imp}^2), \quad (17) \]
Figure 4. Visualization of the concept of Torus Projection, where the circle represents a 2-dimensional torus. The n-dimensional receive vector, \( y(\alpha) \) lies on some point not on the torus. The projection of the torus moves the point to the torus using the known radii, \( r_1 \), and the angles associated with \( y(\alpha) \), that is, \( \alpha_y = \arctan(\Im(y_i))/\Re(y_i) \). The figure also shows the Gaussian error vector between the transmitted vector \( x(\alpha_k) \) and the received vector \( y(\alpha_y) \).

where \( \epsilon \) is the rate of contamination.

At the receiver, a forward DFT brings the time-domain signal, \( y[k] \), to the frequency domain as shown in the example depicted in Fig. 2a. In the frequency domain, \( n/2 \) complex values are extracted to form a vector \( y \), see Fig. 2b, to be used for decoding. The decoder is a surjective map composed of an unfolding function \( g \) that maps the vector \( y \) into an estimate \( \hat{\alpha} \), that is,

\[
g : y \rightarrow \hat{\alpha}.
\]

A shrinking function \( \alpha^{-1}(\cdot) \) then maps \( \hat{\alpha} \) to the source estimate, \( \hat{\hat{\alpha}} \). We use three data types of neural networks to decode \( \hat{s} \); real, complex and quaternions. We also introduce a new type of decoder which we call robust torus projection, see Section IV.

We assume that the local oscillators of the RF hardware at both the transmitter and the receiver is of sufficient quality such that the maximum frequency offset is less than the subcarrier spacing and that they exhibit no imperfections. We have developed two possible ways of extracting the frequency offset. The first involves pre-pending the DFT-C3T symbol by a preamble at the transmitter. The head of the symbol is copied and pre-pended to the tail the symbol in the time domain. At the receiver we used the delayed auto-correlation for energy detection and to find the time/frequency offsets. Numerical experiments showed that this method is highly reliable. The second method is unreliable and depends on multiple arc tan operations on the received complex-valued data followed by finding the slope of the unwrapped angles.

**IV. ROBUST TORUS PROJECTION**

The fact that constant-curvature curve are geodesics on flat tori allows for improved decoder performance since we can project observations onto the torus prior to decoding. An example of this projection for \( n = 2 \) is shown in Fig. 4. First, the \( \ell_2 \)-norm is calculated on the individual pairs received from \( \{ \Im(y_i), \Re(y_i) \} \). Next, the \( \{ \Im(y_i), \Re(y_i) \} \) pairs are scaled by the \( \ell_2 \)-norm and the corresponding \( n/2 \) angles derived using arc cos and arc sin followed by pairwise averaging, which reduces the uncertainty. The torus projection is obtained by constructing an \( n \)-dimensional vector using Eq. (1) with the \( n/2 \) known transmit radii \( r_1 \), the known transmit frequencies \( \omega_i \), and the derived angles \( \alpha_y \). The newly constructed vector is then decoded using a neural network.

In the aforementioned torus projection, the angles from each pair of arc cos and arc sin are averaged, giving some degree of noise immunity. Combining the angles from all pairs of the received vector \( y \), would give significantly more immunity. Therefore, we now propose to scale down the angle \( \alpha \) at the transmitter by \( \omega_n/2 \) such that all \( \omega_i \alpha \) arguments in Eq. (1) falls within \([-0.9\pi, +0.9\pi]\), that is, we map \( \alpha' \) as

\[
\alpha' = \frac{\alpha}{\omega_n/2}.
\]

Geometrically, the new mapping would be the same as in [3], but we would effectively only use a \((2/n)\)-fraction of the curve. For example, for \( n = 4 \), we use a 50% segment of the curve and for \( n = 100 \), we would only use a 2% segment of the curve. Table II shows the utilization of various dimensions \( n \) using the proposed mapping. The effect of this scaling on the decoder performance can be seen in Fig. 11b.

In the receiver, we carry out the same arc cos and arc sin operations as before taking into account the four quadrants. Since we scaled down \( \alpha \) in the transmitter mapping, we now have to scale up the recovered angles accordingly in order to obtain the \( n \) estimates of \( \alpha \).

Let us assume that the transmitted signal was subjected to frequency-selective fading. Because \( x(\alpha) \) is defined in the frequency domain, fading would result in one or more of the recovered angles \( \alpha \) becoming outliers from the distribution. In this paper, we propose to make the recovery of the angles robust by applying the Huber M-estimator to the upscaled angles. The results are presented in Section VI-D.

**Table II. C3T CURVE UTILIZATION FOR THE SCALING METHOD, DFT SIZE AND THE NUMBER OF Poincaré POLARIZATION STATES.**

<table>
<thead>
<tr>
<th>C3T Dimension ( n )</th>
<th>Utilization [%]</th>
<th>DFT size</th>
<th>Polarization States</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>25</td>
<td>16</td>
<td>2</td>
</tr>
<tr>
<td>16</td>
<td>12.5</td>
<td>32</td>
<td>4</td>
</tr>
<tr>
<td>20</td>
<td>10</td>
<td>32</td>
<td>5</td>
</tr>
<tr>
<td>40</td>
<td>5</td>
<td>64</td>
<td>10</td>
</tr>
<tr>
<td>100</td>
<td>2</td>
<td>128</td>
<td>25</td>
</tr>
</tbody>
</table>

**Table III. HYPERPARAMETERS FOR C3T-BASED NEURAL NETWORKS.**

<table>
<thead>
<tr>
<th>Hyperparameter</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of hidden layers</td>
<td>5</td>
</tr>
<tr>
<td>Dimensions of hidden layers</td>
<td>Multiples of 256</td>
</tr>
<tr>
<td>Number of epochs</td>
<td>200</td>
</tr>
<tr>
<td>Batch size(^a)</td>
<td>512</td>
</tr>
<tr>
<td>Learning rates(^b)</td>
<td>0.0075, 0.001, 0.0001, 0.00005, 10(^{-4}), 5 \times 10(^{-5}), 10(^{-5})</td>
</tr>
<tr>
<td>Loss function Optimizer</td>
<td>Mean Square Error</td>
</tr>
<tr>
<td>Optimizer decay</td>
<td>Adam</td>
</tr>
</tbody>
</table>

\(^a\) The batch size for the QNNs was 256.

\(^b\) The 0.075 learning rate, proposed in [20] for CArcsinh, gives poor performance for CV-C3T.
Torus projection is explained in Fig. 4. Angles are processed by an M-estimator before projected.

The concatenation of raw and the torus projection samples of angles are processed by an M-estimator before projected.

The function $f(z)$ can then be written as

$$f(z) = u(x, y) + iv(x, y), \quad \frac{\partial f}{\partial x} = \frac{\partial u}{\partial x} - i\frac{\partial v}{\partial y}, \quad \frac{\partial f}{\partial y} = \frac{\partial u}{\partial y} + i\frac{\partial v}{\partial x},$$

and thus for a point $z_0 \in \mathbb{C}$ in Eq. (21) and using the Cauchy-Riemann equations, the Wirtinger derivative can be derived as follows:

$$f'(z_0) = \frac{\partial f}{\partial z}(z_0) = \frac{1}{2} \left( \frac{\partial f}{\partial x}(z_0) - i\frac{\partial f}{\partial y}(z_0) \right),$$

which is the gradient used in our complex-valued neural network decoder, CV-C3T.

We considered four features for complex-valued activation:

1) CRelu: It applies the ReLU activation to the real and imaginary parts separately. CRelu satisfies the Cauchy-Riemann equations when both the real and imaginary parts are strictly positive or strictly negative [24]. However, we observe that CRelu has poor performance for CV-C3T decoding, so we abandoned it.

2) zReLU: It is a complex activation function based on ReLU called the Guberman activation [25], which satisfies the Cauchy-Riemann equations everywhere except in the positive real and imaginary axes. CV-C3T decoding has poor performance with zReLU, so we abandoned it.

3) Split-Tanh: It applies the hyperbolic tangent to the real and imaginary parts separately and not as a single complex number, hence it cannot fully represent the true gradient.

4) Arcsinh: It is the complex-valued hyperbolic arcsin is shown in Fig. 5.

B. Quaternion-Based Decoder

With the Cayley-Dickson notation, the n-dimensional constant curvature curve, can be viewed as one or more polarization states, see Fig. 6 and Table II. The polarization slant angle $\psi$ is the magnitude ratio and the ellipticity $\chi$ is the phase difference of the two complex numbers. Our constant curvature curve encoder is defined such that the phase for subsequent $\cos/\sin$ pairs is an integer multiple of the first pair, hence $\chi$ is a constant. Looking at groups of four elements of the constant curvature code, considering each group a polarization state, this means that for each group, you have the same ellipse but with different slant angles. For example, in the 100-dimensional case, you have 25 identical ellipses being transmitted over the channel, with some variation in the slant angle for each. This is almost like a repetition code. The slant angles corresponding to the n-dimensional constant curvature code depends on the radii only and hence constant.

We considered six activation functions in the QNNs:

1) dReLU: It applies ReLU activation to the four quaternion parts separately.

2) dModReLU: It is the quaternion extension of the complex-valued modReLU [26], and defined as:

$$\mathbb{H}_\text{ModReLU} = \begin{cases} (|z| + b)^2, & |z| + b \geq 0, \\ 0, & |z| + b < 0. \end{cases}$$

3) dSplit-Arcsinh: It applies the hyperbolic arcsin to the four quaternion parts separately and cannot fully represent the true gradient.
4) **Split-Arcsinh**: The complex-valued hyperbolic arcsin is shown in Fig. 5 and worked well for the CV-C3T decoder. Therefore, we applied CArcsin activation to the Cayley-Dickson pairs of each quaternion.

5) **Split-Tanh**: It applies the hyperbolic tangent to the four quaternion parts separately.

6) **Split-HardTanh**: It applies HardTanh activation to the four quaternion parts separately, where HardTanh is given by

\[
\text{HardTanh} = \begin{cases} 
    b_{\text{max}}, & x > b_{\text{max}}, \\
    b_{\text{min}}, & x < b_{\text{min}}, \\
    x, & \text{otherwise},
\end{cases}
\]

where \(b_{\text{min}}\) and \(b_{\text{max}}\) are thresholds.

VI. **RESULTS FROM NUMERICAL EXPERIMENTS**

Results for real-valued, complex-valued, and quaternion neural networks are presented in this section. For our simulations, we used stochastic gradient descent with the Adam optimizer with decay \(10^{-6}\), a batch size of 512 and trained with SNR levels 10, 5, 0, −5 dB for 200 epochs. We experimented with keeping the dimensionality of the five hidden layers fixed at 256 neurons. We also explored having the dimensionality of the hidden layers increase successively towards the middle layer as shown in Table IV. For each flavor of neural network, we also experimented with the learning rate (LR), that is, the rate at which the network weights are updated for each epoch. The best-performing LR can vary depending on the data type, the activation function and the C3T dimension.

A. **Real-valued Decoder Experiments**

The performance of our real-valued neural network C3T decoding was extensively covered in our previous paper [3]. Here, we add some additional results not previously presented, for instance the effect of the utilization of batch normalization [27] and the effect of keeping the dimensions of all five hidden layers fixed at 256 neurons. Fig. 7 shows that for real-valued C3T and \(n = 8\), better results are achieved without batch normalization produced at both low and high SNR levels compared to the case with batch normalization, especially at high SNR levels. Furthermore, keeping the dimensions of the hidden layers fixed seems detrimental to the performance, especially at low levels of SNR. Some additional real-valued MLP results are presented in Fig. 11b showing very high performance gains at low SNR for high-dimensional C3T codes and associated loss above −2 dB SNR.

B. **Complex-valued Decoder Experiments**

CV-C3T decoders were built with two software libraries CplxModule [28] and complexPyTorch [29]. In CplxModule, the core implementation of complex-valued arithmetic and layers is based on careful tracking of transformations of real and imaginary parts of complex-valued tensors. The batch normalization and weight initialization layers are based on recommendations in [24], which also is the basis for the complexPyTorch library.
C. Quaternion Decoder Experiments

We used the PyTorch software library PyTorch-Quaternion-Neural-Networks [30] to build our QNN networks. We built QNNs with real-valued input dimensions \( n \in \{4, 8, 12, 16, 20, 40, 100\} \). The three types of QNN input features are listed in Table V and the associated activation functions are described in Section V-B. To compare the performance of our QNNs, we also implemented the corresponding reference CV-C3T and C3T networks that match those input dimensions and feature types together with the best performing activation functions for each data-type reference. The dimensions of the hidden layers for each type of data are shown in Table IV.

Fig. 10 show the performance of the QNNs as a function of SNR. At high SNR, QNN with ReLU activation works very well in general, sometimes even significantly better than the second best activation function. At low SNR, QNN with ReLU performs the worst! ReLU clamps the large received magnitude quaternions (hence, limiting the noise) but lets through smaller ones (not limiting the noise). The magnitude of the transmitted quaternion, \( q_{tx} \), is constant, which is expressed as

\[
|q_{tx}| = \sqrt{r_m^2 + r_p^2}, \quad m = 1, 3, 5, \ldots, p = m + 1, \quad (25)
\]

where \( r \) is the fixed C3T radii of Table I, regardless of what information is transmitted. The magnitude should only vary according to the quaternion-valued Gaussian noise. If that noise sample is great and somehow results in a small quaternion magnitude, the \( \mathbb{H}\text{ModReLU} \) will filter out those examples during training. A large magnitude would be bounded by ReLU. The torus projection also introduces some level of noise immunity. The combination of torus projection and \( \mathbb{H}\text{ModReLU} \) activation seems to work effectively to suppress noise. For the \( \mathbb{H}\text{ModReLU} \) parameter \( b \), we tried \( b = 0.5 \) and \( b = 0.8 \), where the latter yielded better results, hence it was used in the QNN simulations. For Split-HardTanh, we used the thresholds \( b_{\text{min}} = -1 \) and \( b_{\text{max}} = +1 \). The training of the QNN was used a batch size of 256. A geometrical explanation to why the QNN performs excellent at decoding C3T above 0 dB SNR could be it learns the ellipticity.

D. Robust Torus Projection Experiments

We simulated scaling down the stretching function to ensure \( \omega_i\alpha \in [-0.9\pi, +0.9\pi] \) and then extract an estimate of the source using the \( \arccos/\arcsin \) functions. Fig. 11a shows the distribution of 400,000 recovered angles for \( n = 8 \) and \( \alpha = \pi/3 \) with an SNR level of 0 dB. Experiments showed that by scaling down the stretching function by \( n/2 \) we gained 8 dB SDR at \(-10\) dB SNR for \( n = 100 \). However, the scaled-down version experience significant SDR loss above \(-2\) dB SNR, for very high-dimension C3T. For \( n = 8 \), the performance of the scaled version is worse above \(+6\) dB SNR.

We also applied the Huber M-estimator to the extracted angles and observed significantly better RMS errors, see Fig. 11b using parameter \( b = 1 \). The the extracted angles after upsampling can be greater than \( 2\pi \). At low SNR, the performance
Figure 8. Comparison between two complex-valued PyTorch libraries with regards to the dimensions of the hidden layers and batch normalization. The complex-valued arcsinh was used for activation at each neuron. We used a batch size of 512 and trained with SNR levels 10, 5, 0, −5 dB for 200 epochs. In this experiment we used the raw features of the 8-dimensional channel output with a learning rate of 0.001.

Figure 9. The performance of the complex-valued neural network with and without a linear layer at the output and for various learning rates using the raw features of the 8-dimensional channel output.

Figure 10. The performance of QNN with real and complex-valued C3T codes including the OPTA lower bound.

VII. CONCLUSIONS AND FUTURE RESEARCH

We propose a scheme for robust decoding of C3T codes. The median of all received angles shows significant noise immunity with drastic performance improvements at low SNR with some loss at higher SNR. Furthermore, scaling down the source by $n/2$ assures that the transmitted angles can be recovered and processed with a Huber M-estimator. The performance at low SNR is several dB better than before but worse at higher SNR. We built complex-valued and quaternion neural networks for decoding C3T. The quaternion neural network exhibits exceptionally good performance using the quaternion-valued ReLU function. Complex-valued neural networks perform better at decoding C3T compared to real-valued networks.

We have developed a concept of using Reconfigurable Intelligent Surfaces (RIS) for the generation of C3T waveforms and will be presented in a subsequent paper. RIS tiles of various sizes can be used to implement the C3T radii, i.e., the wave magnitudes. Furthermore, the elements of the RIS is programmable to induce various phase shifts, they can
represent the C3T phases. Robust dynamic mode composition [31] can be used to extract the decoder input.

Further research could be carried out investigating, for instance, how to use C3T to propagate sensory inputs to a distributed Kalman Filter used in some kind of localization application.

We would like to investigate the unique algebraic ring structure of the quaternion and consider whether it could be utilized for decoding C3T codes or other associated tasks.

REFERENCES


