Robust Constant Curvature Curve Communications with Complex and Quaternion Neural Networks

Anders M. Buvarp *Member, IEEE*, Lamine Mili *Life Fellow, IEEE*, and Amir I. Zaghloul *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.**

***Keywords*—Robust estimation, M-estimators, complex-valued neural networks, quaternion neural networks, joint source-channel codes, URLLC, digital twin, extended reality, machine learning.**

I. Introduction

The concept of a *Digital Twin* has been proposed for sixth-generation mobile communications systems (6G), where a system of virtual objects reflect physical objects accurately. The system is updated with low-latency real-time data and machine learning is used for making decisions [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. In contrast, the real environment could be equipped with actuators, and the operations performed in the XR could be carried out in the real world 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 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) map real-valued samples directly to channel input, avoiding buffer delays, encoding overhead, etc., and therefore exhibit extremely low latency [5–8].

Deniz Gündüz has very recently applied deep learning neural networks to JSCC in various unique and interesting low-latency applications. In [9], they considered a JSCC space-time design using deep learning for the coding of images over MIMO channels. An autoencoder-based adaptive deep learning JSCC architecture for semantic communication of e.g. text and images was proposed in [10], where the channel noise was injected in the latent space. Distributed and data-driven deep learning JSCC for image transmission in practical schemes was considered in [11].

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 [12] codes densely pack tubes inside a hypersphere in an arbitrary n-dimensional space, including odd dimensions. 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 [10] 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. Uniform sources are considered because they have maximum entropy [13] and hence represent the worst case of all possible practical scenarios. Furthermore, comparable JSCC systems [4–8, 14] also use uniform sources. We implemented various neural network-based decoders in arbitrary dimensions and evaluated the performance at very low levels SNR. 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 of about 700 at code rate 0.72 to achieve an error rate of 10^{-6}. For details regarding the comparison with digital codes, see [3]. SDR is the performance measure used in the

A. M. B., L. M., and A. I. Z. are with Bradley Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA 24061 USA. Email: {abuv,lmili,amirz}@vt.edu.

A. M. B. is a Davenport Leadership Scholar.

A. I. Z. is with the United States DEVCOM Army Research Laboratory, Adelphi, MD 20783 USA, amir.i.zaghloul.civ@army.mil.

JSCC literature [4–8, 14] and is defined as SDR $= \sigma_s/MSE$, where σ_s is the source variance and MSE the mean-square reconstruction error. For Ultra-Reliable Low-Latency Communications (URLLC) applications, the requirement to wait 700 samples before transmission would be prohibitive. Our minimum mean square error C3T decoder yields an accuracy of 99.96% at 10 dB SNR, which was computed as the average absolute error of 200 transmissions each of 400 real numbers evenly distributed in the range $[-\pi,+\pi]$. Therefore, we strongly suggest that a communication system based on C3T as proposed in this paper would be an ideal solution in digital twin, IoE, and XR applications which requires both low latency and high reliability.

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 [15], 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 [16]. This weakness led Peter Huber to develop the theory of robust statistics in 1964, and extended by Frank Hampelin 1968. Recently, very sophisticated estimation methods have been developed for wireless systems that is based on the distribution of the data [17]. Furthermore, robust estimation has been applied to the important topic of dynamic mode decomposition [18]. Until recently however, only a few papers considered channel estimation with robust statistics.

In this work, our main contributions include the following:
1) Reduction of cost, power, and complexity. For the simulations in [3] we considered $n/2$ orthogonal channels. A practical realization of C3T based on that 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. In our first paper on C3T, the decoder processed a single code word making statistical signal processing impossible. In this paper, we propose to scale down the source samples in order to facilitate the extraction of a large quantity of derived samples. Furthermore, this process enables us to apply 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 or more frequency bins becoming outliers. The Huber estimator down-weights these outliers during the equalization process, yielding better quality estimates. Additionally, having a large quantity of samples allows for noise immunity. Finally, the transmitted signal is restored by a process we call torus projection and we thereby coin the term Robust Torus Projection (RTP).
3) Complex-valued and quaternion neural networks. A C3T vector can be viewed as a vector of complex numbers, hence we train complex-valued C3T (CV-C3T) decoders using Multi-Layer Perceptrons (MLP) taking into account the phase between pairs $\cos / \sin$. Furthermore, we consider pairs of complex numbers as quaternions and train Quaternion Neural Networks (QNN) accordingly, achieving a performance improvement of 33 dB SDR (at 0 dB SNR) over our previous results [3].

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 experiments 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, E is the expectation function, $\mathcal{O}$ is the Bachmann-Landau big O notation, C^r is an r times continuously differentiable parametric curve, $\mathbb{T}^{n/2}$ 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 $x \sim \mathcal{N}(\mu,\sigma^2)$ represents the Gaussian distribution with mean μ and variance σ^2. The complex-valued and quaternion Gaussian distributions are presented as $n \sim \mathcal{N}_{\mathcal{C}}(0,\sigma_n^2)$ and $n \sim \mathcal{N}_{\mathcal{H}}(0,\sigma_n^2)$, 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 $\mathbf{x}(\alpha)$ defined as [19]

$$\mathbf{x}(\alpha) = [r_1\cos(\omega_1\alpha), r_1\sin(\omega_1\alpha), r_2\cos(\omega_2\alpha), r_2\sin(\omega_2\alpha), \ldots, r_{n/2}\cos(\omega_{n/2}\alpha), r_{n/2}\sin(\omega_{n/2}\alpha)]^T, \quad (1)$$

where $\alpha \in I$ is a stretched version of the source sample s, I is an interval of the real line, r_i and ω_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 [20]. Geodesics on flat tori must all have constant generalized curvature [19].

Table I. C3T RADII, GLOBAL RADIUS AND THE MAX TUBE DENSITY.

n	$\mathbf{r}$[a]	$\rho_{\mathbf{r}}$	D_{max}
16	[0.38618, 0.40925, 0.40228, 0.39142, 0.36800, 0.32634, 0.28696, 0.20993]	0.72546	5.8045e-5
20	[0.35251, 0.36758, 0.37378, 0.3705, 0.36659, 0.34508, 0.30498, 0.24627, 0.19969, 0.12894]	0.72424	2.2944e-6
40	[0.23259, 0.21674, 0.25316, 0.23812, 0.26905, 0.25430, 0.27287, 0.26388, 0.26076, 0.25428 0.25006, 0.23870, 0.23285, 0.22071, 0.19166 0.18310, 0.17029, 0.15342, 0.09811, 0.09737]	0.71011	1.2154e-13
100	[0.12382, 0.10848, 0.15717, 0.17994, 0.06367, 0.01646, 0.17460, 0.13555, 0.12845, 0.11500 0.11685, 0.19140, 0.20346, 0.12070, 0.20685 0.05685, 0.20473, 0.03710, 0.10416, 0.08591 0.22643, 0.21106, 0.19916, 0.16115, 0.03119 0.18562, 0.12057, 0.05375, 0.17488, 0.17700 0.15180, 0.20158, 0.13782, 0.03849, 0.14233 0.15443, 0.12247, 0.068440, 0.15670, 0.16761 0.17627, 0.17180, 0.06183, 0.10258, 0.08054 0.13447, 0.10681, 0.12692, 0.11187, 0.05277]	0.67371	3.0959e-37

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

By selecting $\omega_i = i\omega_1$, $i = 2,3,\cdots,n/2$, we obtain curves that repeat themselves and hence form an n-dimensional *knot*, and thus achieving curvatures that are all greater than zero [19]. To prevent the end points of the curve from joining, we scale α by a factor 0.9. This scale factor was obtained experimentally to maximize the decoder performance. Too much scaling reduces performance at high SNR. The smallest distance between the folds of the curve is twice the global radius [12] of the curvature given by

$$\rho_{\mathbf{r}}^2(\Delta) = \frac{[\sum_{i=1}^{n/2} r_i^2(1-\cos(i\Delta))]^2}{2\sum_{i=1}^{n/2} r_i^2(1-\cos(i\Delta)) - \frac{[\sum_{i=1}^{n/2} i r_i^2 \sin(i\Delta)]^2}{\sum_{i=1}^{n/2} i^2 r_i^2}} , \quad (2)$$

where Δ is the distance between two points on the curve. This allows a small radius of the ball [14, Definition 2] at any point of the curve to avoid self-intersections or tight bending that leads to the threshold effect [4]. Considering the curve length, $L_{\mathbf{r}} = 2\pi\sqrt{\sum_{i=1}^{n/2} i^2 r_i^2}$, the tube cross-sectional volume, $V_{cs} = \mathcal{C}_{n-1}\rho_{\mathbf{r}}^{n-1}$, the bounding n-sphere volume, $V_b = \mathcal{C}_n(1+\rho_{\mathbf{r}})^n$, and the constant, $\mathcal{C}_n$, we obtained the associated set of radii, $\mathbf{r}$, for each dimension n, that maximizes the tube density given by

$$\mathrm{D} = L_{\mathbf{r}} \frac{V_{cs}}{V_b}. \quad (3)$$

Table I shows the resulting C3T radii, the global radius (i.e. the tube radius), and the maximized tube density. The optimization objective is to minimize the tube radius in order to ensure that the the volume of the tube surrounding the curve is maximized within the bounding unit sphere. Generally, the C3T performance increases with the dimensionality [3], which in part is because of additional redundancy.

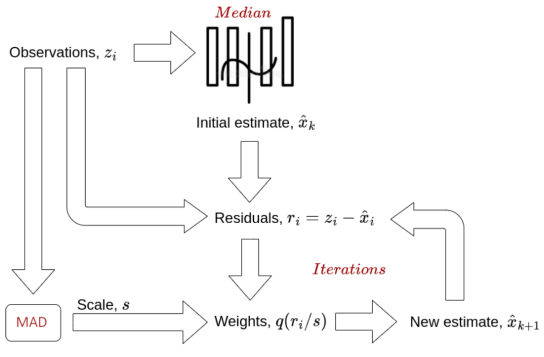

Figure 1. The iterative process of robust estimation.

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* [21]. Popularized by Huber [22], the contamination distribution model is expressed as

$$G = (1-\epsilon)F + \epsilon H , \quad (4)$$

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 [22]. The largest fraction of contamination an estimator can deal with is its *breakdown point* [23]. 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 sample mean as an estimator of location. Similarly, the standard deviation is replaced by a robust scale, s, defined as the Median-Absolute-Deviation (MAD) from the median

$$s = 1.4826 \times median\ |z_i - median(z_j)| , \quad (5)$$

where z_i are the observations.

Now, it is also desirable that a robust estimator is 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

$$J(\theta) = \min_{\hat{\theta}} \sum_{i=1}^{m} \rho\left(\frac{r_i}{s}\right) , \quad (6)$$

and residuals $r_i = z_i - \hat{x}_i$, observations z_i, our estimate $\hat{x}_i$, and some arbitrary function ρ. The Huber M-estimator is a combination of the *least squares* and the *least absolute value* estimator. This estimator of location has a high statistical efficiency at the Gaussian distribution and has a bounded bias under contamination, that is, it is robust to outliers with an

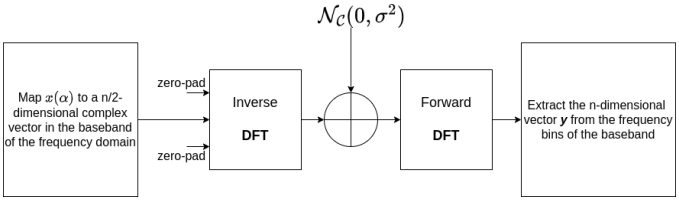

Figure 2. Mapping the encoded n-dimensional C3T vector to sub-carriers in the frequency domain followed by zero-padding and an Inverse FFT to bring the signal to the time domain. At the receiver, the n-dimensional input to the C3T decoder is extracted in the frequency domain after the Forward FFT.

asymptotic breakdown point of 50%. The outliers are down-weighted [24] using the iterative process shown in Fig. 1 and expressed as

$$\hat{\theta}^{k+1} = \frac{\sum_{i=1}^{m} z_i q\left(\frac{r_i}{s}\right)}{\sum_{i=1}^{m} q\left(\frac{r_i}{s}\right)}\,, \tag{7}$$

where q is the weight function and defined as the second-derivative of the ρ-function.

III. DFT-Based C3T System Model

Fig. 2 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 $s \sim \mathcal{U}(a,b)$, which is followed by a *folding* function

$$\mathbf{f} : \alpha(s) \to \mathbf{x}\,. \tag{8}$$

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. 3 including the associated received vector. 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 avoids the DC bin. The resulting vector is converted to the time domain using the inverse DFT as shown in Fig. 2.

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

$$y[k] = x[k] + w[k]\,, \tag{9}$$

where $x[k] = IDFT\{\mathbf{x}(\alpha)\}$, $w(k) \sim \mathcal{N}_{\mathcal{C}}(0,\sigma_w^2)$ 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 in the time domain. Since we defined the signal, $\mathbf{x}(\alpha)$, 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,

$$\begin{aligned} P &= var(x) \\ &= \mathrm{E}[x^2] - \mathrm{E}[x]^2\,. \end{aligned} \tag{10}$$

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

$$P_{tx} = \frac{2}{n}\sum_{i=1}^{n/2} r_i^2\,, \tag{11}$$

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

$$\sigma_w^2 = P_{tx} 10^{-\frac{SNR_{dB}}{10}}\,. \tag{12}$$

In our complex-valued simulation models, the noise power, σ_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 $w \sim \mathcal{N}_{\mathcal{H}}(0,\sigma_w^2)$ and the noise power in Eq. (12) is distributed uniformly over all four parts of the quaternion samples.

Consider impulse noise [21] with some noise power $\sigma_{imp} \gg \sigma_w$, and following Huber's contamination model, the impulse noise model is as follows:

$$w \sim (1-\epsilon)\mathcal{N}_{\mathcal{C}}(0,\sigma_w^2) \;+\; \epsilon\mathcal{N}_{\mathcal{C}}(0,\sigma_{imp}^2)\,, \tag{13}$$

where ϵ 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. 3. In the frequency domain, $n/2$ complex values are extracted to form a vector $\mathbf{y}$, see Fig. 3, to be used for decoding. The decoder is a surjective map composed of an *unfolding* function g that maps the vector $\mathbf{y}$ into an estimate $\hat{\alpha}$, that is,

$$\mathbf{g} : \mathbf{y} \to \hat{\alpha}\,. \tag{14}$$

A *shrinking* function $\alpha^{-1}(\cdot)$ then maps $\hat{\alpha}$ to the source estimate, $\hat{s}$. 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 sub-carrier 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 $\arctan$ 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 curves 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.

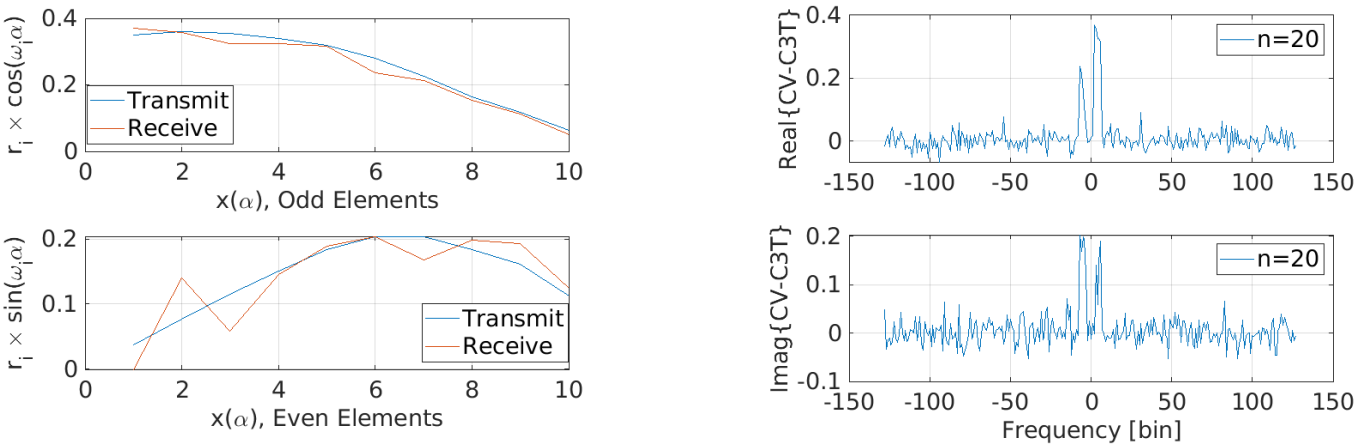

Figure 3. (left) The odd and even elements of the transmitted vector $x(\alpha)$ as in Eq. (1) and the corresponding elements of the received vector $\mathbf{y}$ at 5 dB SNR. (right) The frequency domain representation of a received DFT-C3T signal with 5 dB SNR and for $n = 20$. The signal consists of ten values each in the real and imaginary parts distributed as five in the positive frequency domain and five in the negative frequency domain.

First, the ℓ_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 ℓ_2-norm and the corresponding $n/2$ angles derived using arccos and arcsin followed by pairwise averaging, which reduces uncertainty. The torus projection is obtained by constructing an n-dimensional vector using Eq. (1) with the $n/2$ known transmit radii r_i, the known transmit frequencies ω_i, and the derived angles $\alpha_{\mathbf{y}}$. The newly constructed vector is then decoded using a neural network.

In the aforementioned torus projection, the angles from each pair of arccos and arcsin are averaged, giving some degree of noise immunity. Combining the angles from all pairs of the received vector $\mathbf{y}$, would give significantly more immunity. Therefore, we now propose to scale down the angle α 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 α' as

$$\alpha' = \frac{\alpha}{\omega_{n/2}} \,. \tag{15}$$

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 decoder performance can be seen in Fig. 12b.

In the receiver, we carry out the same $arccos$ and $arcsin$ operations as before, taking into account the four quadrants. Since we scaled down α in the transmitter mapping, we now have to scale up the recovered angles accordingly in order to obtain the n estimates of α.

Let us assume that the transmitted signal was subjected to frequency-selective fading. Because $\mathbf{x}(\alpha)$ is defined in the frequency domain, fading would result in one or more of the recovered angles α 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.

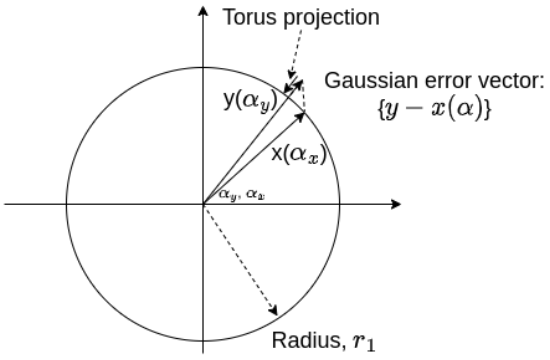

Figure 4. Visualization of the concept of *Torus Projection*, where the circle represents a 2-dimensional torus. The n-dimensional receive vector, $y(\alpha)$ lies on some point *not* on the torus. The projection of the torus moves the point to the torus using the known radii, r_i, and the angles associated 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 $\mathbf{x}(\alpha_{\mathbf{x}})$ and the received vector $\mathbf{y}(\alpha_{\mathbf{y}})$.

V. MLP-BASED DECODERS FOR C3T

We consider supervised neural network-based decoders with data types, real, complex and quaternion, where the MLP

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

C3T Dimension n	Utilization [%]	DFT size	Polarization States
8	25	16	2
16	12.5	32	4
20	10	32	5
40	5	64	10
100	2	128	25

Table III. HYPERPARAMETERS FOR C3T-BASED NEURAL NETWORKS.

Hyperparamter	Values
Number of hidden layers	5
Dimensions of hidden layers	Multiples of 256
Number of epochs	200
Batch size[a]	512
Learning rates[b]	$0.075, 0.01, 0.001, 0.0005, 10^{-4}, 5\cdot 10^{-5}, 10^{-5}$
Loss function	Mean Square Error
Optimizer	Adam
Optimizer decay	10^{-6}

[a] The batch size for the QNNs was 256.
[b] The 0.075 learning rate, proposed in [25] for $\mathbb{C}Arcsinh$, gives poor performance for CV-C3T.

solves a regression problem and the associated hyperparameters are summarized in Table III. We used five hidden layers, where the layer dimensions gradually increases until the middle layer, from where the number of neurons gradually decreases for each additional layer as indicated in Table IV.

The mean square error (MSE) is used for the loss function to determine the gradient for back-propagation. Table V lists the types of features on the input of the neural networks used in this work. The starting learning rate was obtained through experimentation depending on the code dimension n and the feature type. We also employed an adaptive algorithm that reduced the learning rate by 15% if the average batch loss relative the loss in the previous batch was greater than 95% after at least 20 batches. We use the hyperbolic tangent, *tanh*, activation function on all real-valued neurons, which performs better than the sigmoid function. The time complexity for training the MLP for n dimensions, t epochs using p examples is shown in Table IV, where k is the dimension of the first hidden layer. The inference time is obtained by setting $t = 1$ and $p = 1$. The CV-C3T complexity is about twice that of C3T and the QNN complexity about sixteen times greater. Table IV also includes the inference complexity for the JSCC most similar to C3T [26]. The inference complexity for C3T is of $\mathcal{O}(n)$, while the Campello et al. has a $\mathcal{O}(n^3)$ complexity.

A. Complex-Valued Decoder

Artificial neural networks are typically employed for processing real-valued data, such as digitized images, music, etc. Complex values are of interest when dealing with wave propagation in RF systems. Our C3T codes embeds the information to be conveyed from the transmitter to the receiver in the phases of the $\cos\,/\,\sin$ vector hence it makes sense to use the complex data type in the network because phase information

Table IV. DIMENSIONS OF FIVE CONSECUTIVE MLP NETWORK LAYERS WITH REAL, COMPLEX AND QUATERNION DATA TYPES RESPECTIVELY.

Input	L1	L2	L3	L4	L5	Output	Complexity[a]
n	256	512	1024	512	256	1	$\mathcal{O}(ptk[n+20k])$[b]
$n/2$	128	256	512	256	128	1	$\mathcal{O}(ptk[2n+84k])$
$n/4$	64	128	256	128	64	1	$\mathcal{O}(ptk[16n+1344k])$
n						1	$\mathcal{O}(n^2[n+2]/32)$[c]

[a] Note that p is the number of examples, t is the number of epochs, and k is the outer hidden layer dimension.
[b] This is the complexity of C3T in [3].
[c] This is the complexity of inference for the flat tori-based JSCC code in [26], which is most similar to our C3T for $n = 8$. We estimated this complexity in [3].

Table V. NEURAL NETWORK INPUT FEATURE TYPES.

Type	Description
Raw	The raw n samples at the output of the channel
Torus Projection	Torus projection is explained in Fig. 4
Raw + TP	The concatenation of raw and the torus projection samples
Robust Torus Projection	Angles are processed by an M-estimator before projected

is preserved with complex numbers. Neural networks to the complex domain was first proposed by Thomas Clarke [27] followed by a large number of work by Akira Hirose and books [25, 28–31].

Consider a complex-analytic function $f(z)$ that is holomorphic on an open set $\mathcal{U}$ containing $z = x + iy$ in the complex plane, where i is the complex unit, and x,$y \in \mathbb{R}$. Furthermore, consider the real-valued harmonic functions $u(x, y)$ and $v(x, y)$, which are related to each other according to the Cauchy-Riemann equations defined as

$$\frac{\partial u}{\partial x} = \frac{\partial v}{\partial y}, \qquad \frac{\partial v}{\partial x} = -\frac{\partial u}{\partial y}\,. \tag{16}$$

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

$$f(z) = u(x, y) + iv(x, y)\,, \tag{17}$$

and thus for a point $z_0 \in \mathbb{C}$ in Eq. (17) and using the Cauchy-Riemann equations, the *Wirtinger derivative* can be derived as follows:

$$f'(z_0) = \frac{\partial f}{\partial z}(z_0) = \frac{1}{2}\left(\frac{\partial f}{\partial x}(z_0) - i\frac{\partial f}{\partial y}(z_0)\right)\,, \tag{18}$$

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

We considered four functions for complex-valued activation:
1) $\mathbb{C}$ReLU: It applies the ReLU activation to the real and imaginary parts separately. $\mathbb{C}$ReLU satisfies the Cauchy-Riemann equations when both the real and imaginary parts are strictly positive or strictly negative [30]. However, we observe that $\mathbb{C}$ReLU has poor performance for CV-C3T decoding, so we abandoned it.
2) zReLU: It is a complex activation based on ReLU called

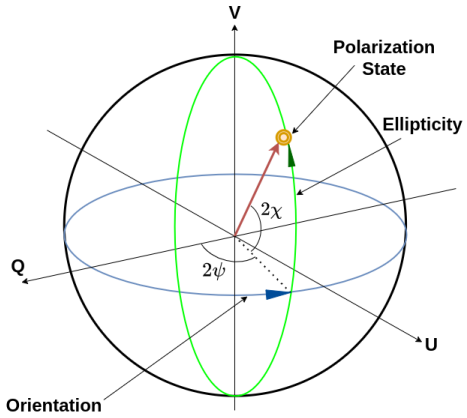

Figure 5. Poincaré sphere defining the polarization state as a function of the orientation ψ and the ellipticity χ.

the Guberman activation [31], 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) $\mathbb{C}$Arcsinh: It is the complex-valued hyperbolic arcsin.

B. Quaternion-Based Decoder

Quaternions were first introduced by Hamilton [32]. In case of complex-valued data, traditional neural networks consider the real and imaginary parts separately, hence phase information is lost. Taking this one step further, a quaternion could be considered containing two phases, see χ and ψ in Fig. 5, hence by utilizing QNNs we *double* the phase information. With the quaternionic Cayley-Dickson notation, the n-dimensional constant curvature curve, can be viewed as one or more *polarization states*, see Fig. 5 and Table II. The polarization slant angle ψ is the magnitude ratio and the ellipticity χ is the phase difference of the two complex numbers. Our constant curvature curve encoder is defined so that the phase for subsequent $\cos/\sin$ pairs is an integer multiple of the first pair, and hence χ 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, there are 25 identical ellipses that are 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 remains constant.

We considered six activation functions in the QNNs:
1) $\mathbb{H}$ReLU: It applies ReLU activation to the four quaternion parts separately.
2) $\mathbb{H}$ModReLU: It is the quaternion extension of the complex-valued *modReLU* [33], and defined as:

$$\mathbb{H}ModReLU = \begin{cases} (|z|+b)\dfrac{z}{|z|}, & |z|+b \geq 0, \\ 0, & |z|+b < 0\ . \end{cases} \tag{19}$$

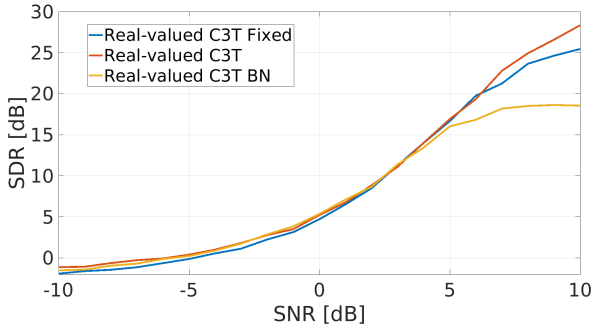

Figure 6. Comparison of real-valued C3T decoding with regards to the dimensions of the hidden layers and batch normalization. *Fixed* indicates that all hidden layers have the same dimensions. In this simulation, we used the raw features of the 8-dimensional channel output with a learning rate of 10^{-5}.

3) Split-Arcsinh: It applies the hyperbolic arcsin to the four quaternion parts separately and cannot fully represent the true gradient.
4) Spit-$\mathbb{C}$Arcsinh: The complex-valued hyperbolic arcsin, which worked well for the CV-C3T decoder. Therefore, we applied $\mathbb{C}$Arcsin 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

$$HardTanh = \begin{cases} b_{max}, & x > b_{max}, \\ b_{min}, & x < b_{min}, \\ x, & otherwise\ , \end{cases} \tag{20}$$

where b_{min} and b_{max} are thresholds.

VI. Numerical Simulation Results

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. The training and evaluation data was generated in Python by drawing uniformly distributed source samples α in the range $[-0.9, 0.9]$. We used 15,000 and 10,000 examples per SNR level for training and evaluation respectively. 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 [34] and the effect of keeping the dimensions of all five hidden layers fixed at 256 neurons. Fig. 6 shows that for real-valued C3T and $n = 8$, better results are achieved without batch normalization produced at 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 performance, especially at low levels of SNR.

B. Complex-valued Decoder Experiments

CV-C3T decoders were built with two software libraries *CplxModule* [36] and *complexPyTorch* [37]. In *CplxModule*, the core implementation of complex-valued arithmetic and layers is based on careful tracking of transformations of real

This article has been accepted for publication in IEEE Transactions on Communications. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2024.3418915

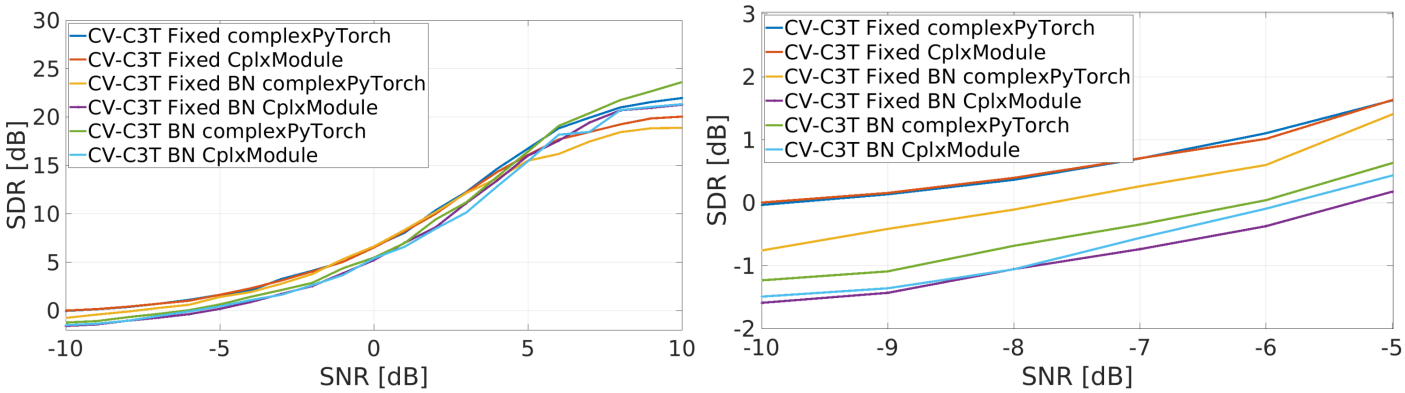

Figure 7. Comparison between two complex-valued PyTorch libraries with regards to the dimensions of the hidden layers and batch normalization. The complex-valued arcsinh was used for activation at each neuron. We used a batch size of 512 and trained with SNR levels $10,5,0,-5$ dB for 200 epochs. In this experiment we used the raw features of the 8-dimensional channel output with a learning rate of 0.001.

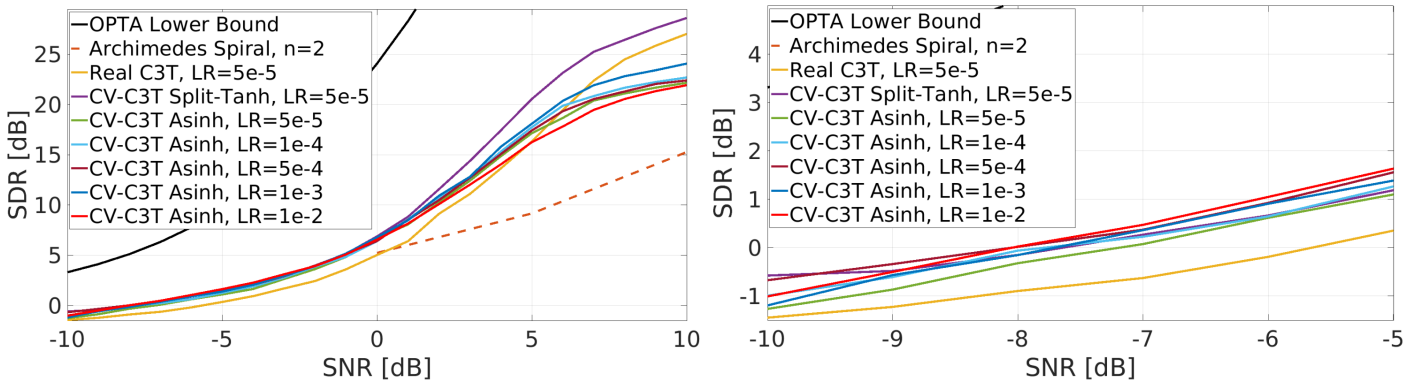

Figure 8. The performance of the complex-valued neural network for various learning rates using the raw features of the 8-dimensional channel output. The lower bound of the optimal performance theoretically achievable (OPTA) [35] is included as reference in addition to the performance at and above 0 dB SNR of the two-dimensional Archimedes' spiral [4].

and imaginary parts of complex-valued tensors. The batch normalization and weight initialization layers are based on recommendations in [30], which also is the basis for the *complexPyTorch* library.

To produce the final one-dimensional real-valued decoder output, that is, the decoded value that estimates the source sample α, we took the real and imaginary parts from the final complex layer and fed them through a linear layer, that is, a transformation from $\mathbb{R}^2$ to $\mathbb{R}$. Therefore, the complex-valued networks have one additional output layer over the real-valued C3T. We tried using magnitude and phase instead of the real and imaginary parts at the complex-valued network output but it yielded poor results. Concatenating real and imaginary parts with magnitude and phase, that is, a transformation from $\mathbb{R}^4$ to $\mathbb{R}$ at the output produces better performance at high SNR levels but *worse* performance at low levels of SNR.

Fig. 7 shows the performance of the *complexPyTorch* and *CplxModule* libraries decoding CV-C3T. Both seem to have similar performance overall although *CplxModule* achieved the best performance at low SNR (for fixed-size hidden di-

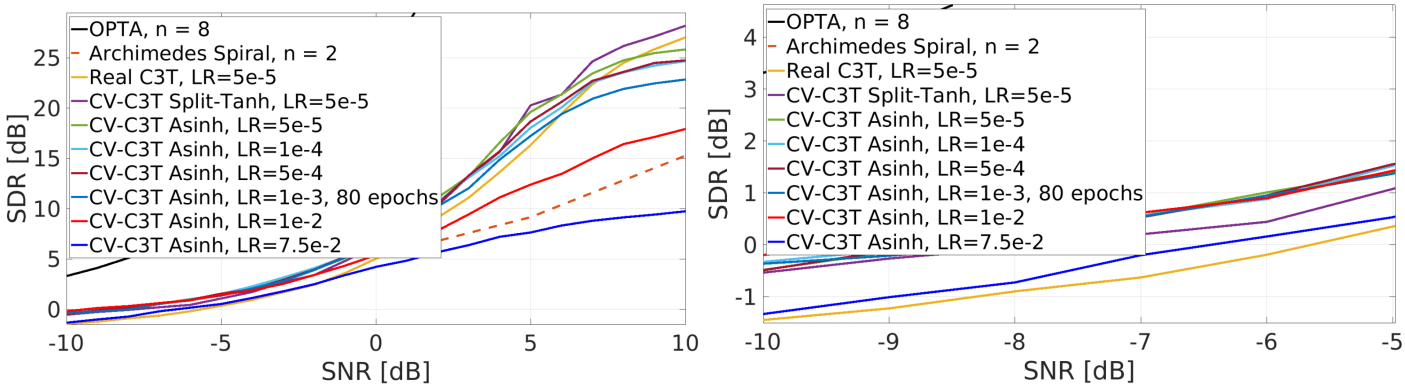

Figure 9. Same as Fig. 8 but with an additional linear output layer.

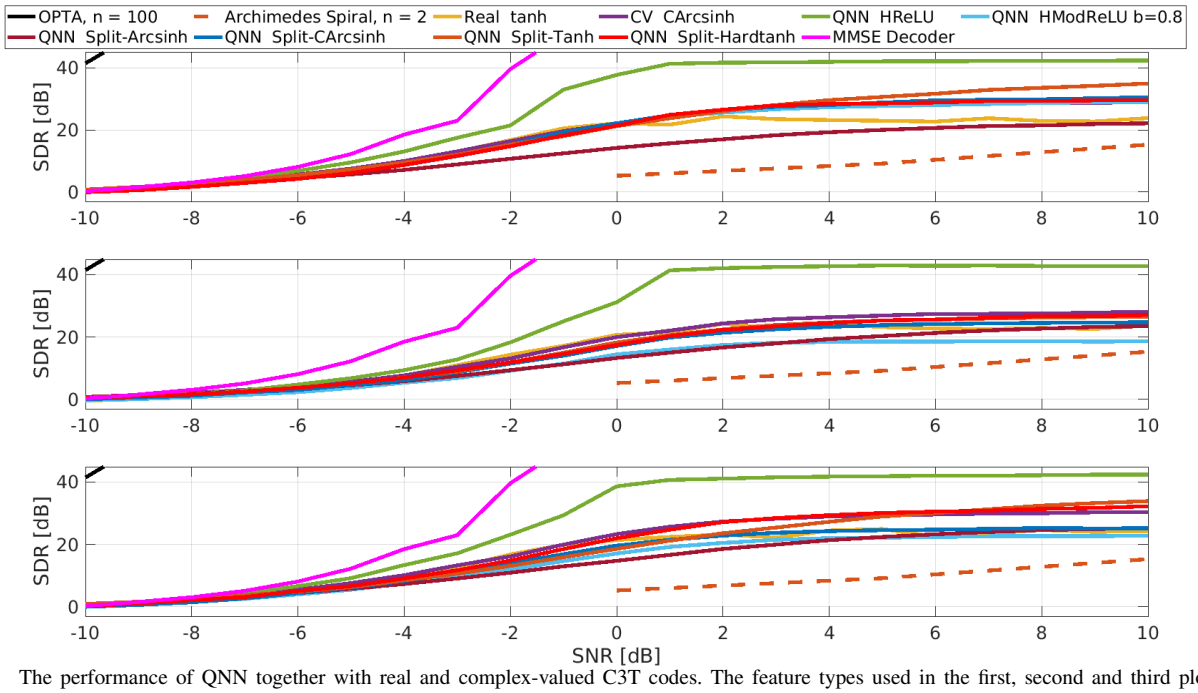

Figure 10. The performance of QNN together with real and complex-valued C3T codes. The feature types used in the first, second and third plot are Raw, Torus Projection and the combination of the two, respectively. The OPTA lower bound, Archimedes' spiral [4] and a MMSE decoder are shown for reference.

mensions). At low SNR levels, the complex-valued networks without batch normalization and with fixed-size hidden layer dimensionality perform consistently the best (observed for multiple experiments). At -10 dB, the *complexPyTorch* C3T performance is very close to 0 dB SDR, which is almost 2 dB better than the corresponding real-valued C3T shown in Fig. 6.

Fig. 8 shows the results for CV-C3T decoding with the various learning rates (LR). We had a single $\arctan$ operation on the output of the complex-valued network to produce one real number representing the decoded source. At high SNR, only the Split-Tanh performs better than the real-valued C3T. $\mathbb{C}$Arcsinh with LR = 5×10^{-5} performs better than real-valued C3T up to approximately 8 dB SNR. At low SNR, all CV-C3T flavors perform better than C3T and $\mathbb{C}$Arcsinh activation performs better than Split-Tanh. Our C3T coder operates at very low SNR (below 0 dB SNR). Comparable systems by e.g. Vaishampayan and Costa [14, 38] operates above $+10$ dB SNR. Furthermore, C3T can have any dimension (e.g. $n = 100$) while Hekland's Archimedes' spiral is two-dimensional [4]. Therefore, it is hard to compare our performance with comparable state-of-the-art. The CV-C3T coder performs better than the Archimedes' spiral as shown in Fig. 8.

A real-valued linear layer was added after $\arctan$ to scale to the real output with the performance shown in Fig. 9. In general, the linear output layer seems to help. Training with learning rates greater than 0.001 generally results in poor performance at high SNR, but, surprisingly, the 0.01 learning rate produces the very best CV-C3T result at low SNR. Note that the fixed-size CV-C3T performs better at low SNR, see Fig. 7.

C. Quaternion Decoder Experiments

We used the PyTorch software library *PyTorch-Quaternion-Neural-Networks* [39] 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 feature 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 shows the performance of the QNNs as a function of SNR. The QNN networks perform exceedingly better than the Archimedes' spiral. In general, QNN performs better than C3T and CV-C3T. However, some activation functions yield poorer results in comparison. At high SNR, the HReLU activation works very well in general, sometimes even significantly better than the second best activation function. At low SNR, Split-Tanh performs the best, see Fig. 11. It should also be noted that at *very low* SNR, the QNN decoder actually performs better than the Bayesian MMSE decoder.

The magnitude of the transmitted quaternion, q_{tx}, is constant, which is expressed as

$$|q_{tx}| = \sqrt{r_m^2 + r_p^2},\ m = 1, 3, 5, \ldots,\ p = m + 1, \qquad (21)$$

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 $\mathbb{H}$ModReLU will filter out those examples during training. The torus projection also introduces some level

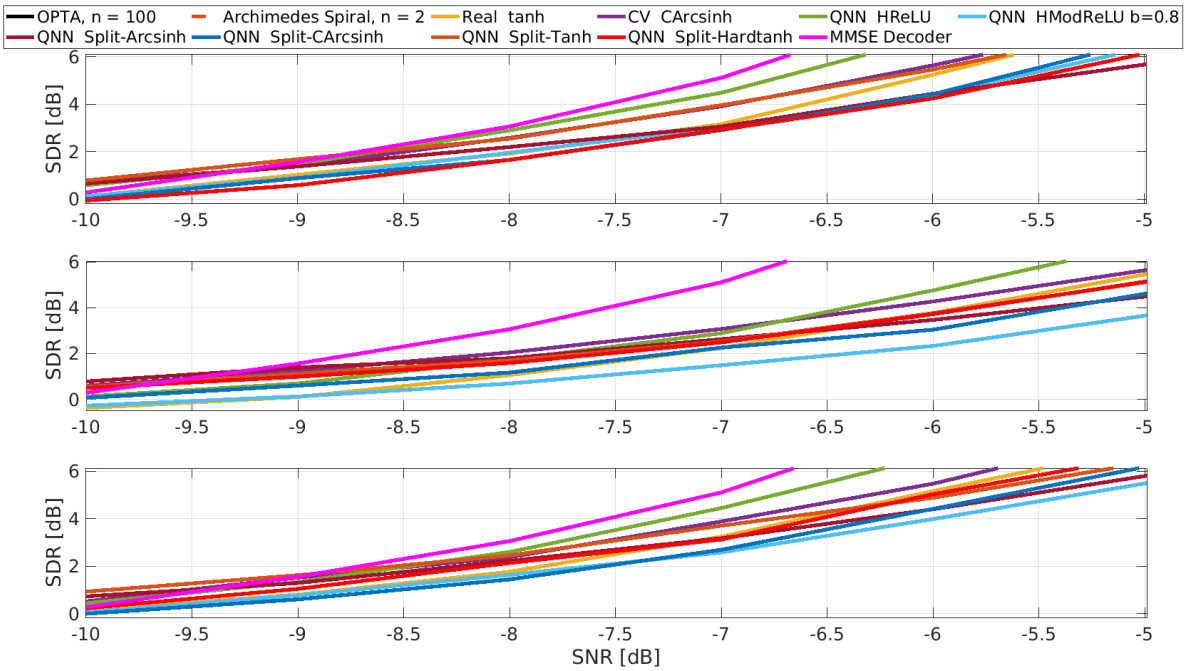

Figure 11. As in Fig. 10 but for low SNR. The QNN decoder out-performs the Bayesian MMSE decoder at very low SNR.

of noise immunity. The combination of torus projection and $\mathbb{H}$ModReLU activation seems to work effectively to suppress noise. For the $\mathbb{H}$ModReLU parameter b, we tried $b = 0.5$ and $b = 0.8$, where the latter yielded better results, and hence it was used in the QNN simulations. For Split-HardTanh, we used the thresholds $b_{min} = -1$ and $b_{max} = +1$. The training of the QNN used a batch size of 256. A geometrical explanation to why the QNN performs excellently in decoding C3T above 0 dB SNR could be that 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. The extracted angles after up-scaling can be greater than 2π. Fig. 12 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.

Using $b = 1$, we applied the Huber M-estimator to the extracted angles and observed an improved distribution of angles, see Fig. 12, and significantly better RMS errors, see Fig. 13. At low SNR, the performance of robust torus projection is significantly better than C3T. Considering the performance improvements by scaling the source to low SNR, we propose a dual mode C3T code, whose operation depends on the SNR level.

VII. Conclusions and Future Research

We propose a scheme for robust decoding of C3T codes. The sample 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

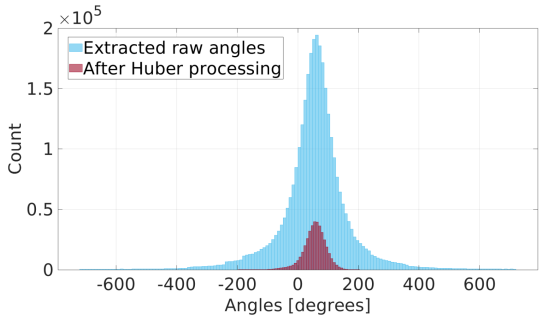

Figure 12. Histogram of extracted and Huber angles for a 60-degree source transmitted over an AWGN channel with SNR 0 dB and C3T $n = 8$.

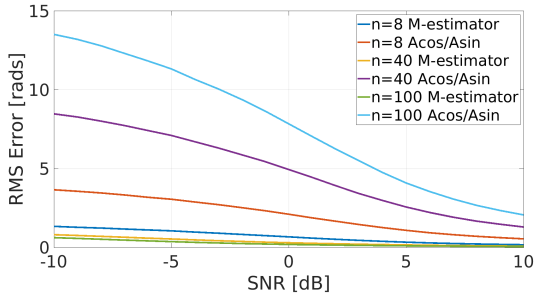

Figure 13. The RMS error of the extracted angles with and without the Huber M-estimator for C3T dimensions $n \in 8{,}40{,}100$.

source by $n/2$ ensures that transmitted angles can be recovered and processed with a Huber M estimate. 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.

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. In particular, 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

[1] L. U. Khan, W. Saad, D. Niyato, Z. Han, and C. S. Hong, "Digital-twin-enabled 6G: Vision, architectural trends, and future directions," *IEEE Communications Magazine*, vol. 60, no. 1, pp. 74–80, 2022.

[2] Y. Polyanskiy, H. V. Poor, and S. Verdú, "Channel coding rate in the finite blocklength regime," *IEEE Transactions on Information Theory*, vol. 56, no. 5, pp. 2307–2359, 2010.

[3] A. M. Buvarp, R. M. Taylor, K. M. Mishra, L. Mili, and A. Zaghloul, "Constant curvature curve tube codes for low-latency analog error correction," *IEEE Transactions on Information Theory*, vol. 69, no. 12, pp. 7738–7754, 2023.

[4] F. Hekland, P. Floor, and T. Ramstad, "Shannon-Kotelnikov mappings in joint source-channel coding," *IEEE Transactions on Communications*, vol. 57, no. 1, pp. 94–105, 2009.

[5] E. Akyol, K. Rose, and T. Ramstad, "Optimal mappings for joint source channel coding," in *IEEE Information Theory Workshop*, 2010, pp. 1–5.

[6] N. Wernersson, M. Skoglund, and T. Ramstad, "Polynomial based analog source-channel codes," *IEEE Transactions on Communications*, vol. 57, no. 9, pp. 2600–2606, 2009.

[7] K. Zhang, J. Jiao, Z. Huang, S. Wu, and Q. Zhang, "Finite block-length analog fountain codes for ultra-reliable low latency communications," *IEEE Transactions on Communications*, vol. 68, no. 3, pp. 1391–1404, 2020.

[8] M. Kleiner and B. Rimoldi, "Asymptotically optimal joint source-channel coding with minimal delay," in *IEEE Global Telecommunications Conference*, 2009, pp. 1–5.

[9] C. Bian, Y. Shao, H. Wu, and D. Gündüz, "Space-time design for deep joint source channel coding of images over mimo channels," in *International Workshop on Signal Processing Advances in Wireless Communications*, 2023, pp. 616–620.

[10] J. Xu, T.-Y. Tung, B. Ai, W. Chen, Y. Sun, and D. Gündüz, "Deep joint source-channel coding for semantic communications," *IEEE Communications Magazine*, vol. 61, no. 11, pp. 42–48, 2023.

[11] S. F. Yilmaz, E. Ozyilkan, D. Gunduz, and E. Erkip. (2023) Distributed deep joint source-channel coding with decoder-only side information. arXiv 2310.04311.

[12] O. Gonzalez and J. H. Maddocks, "Global curvature, thickness and the ideal shapes of knots," *Proceedings of the National Academy of Sciences*, vol. 96, no. 9, pp. 4769–4773, 1999.

[13] T. Cover and J. Thomas, *Elements of Information theory*. John Wiley & Sons, 1991.

[14] V. A. Vaishampayan and S. I. R. Costa, "Curves on a sphere, shift-map dynamics, and error control for continuous alphabet sources," *IEEE Transactions on Information Theory*, vol. 49, no. 7, pp. 1658–1672, 2003.

[15] R. A. Fisher, "Theory of statistical estimation," *Mathematical proceedings of the cambridge philosophical society*, p. 700–725, 1925.

[16] R. Maronna, R. Martin, V. Yohai, and M. Salibián-Barrera, *Robust Statistics: Theory and Methods (with R)*, ser. Wiley Series in Probability and Statistics. Wiley, 2019.

[17] J. A. Fishbone and L. Mili, "Highly robust complex covariance estimators with applications to sensor array processing," *IEEE Open Journal of Signal Processing*, vol. 4, pp. 208–224, 2023.

[18] M. N. A. H. Abolmasoumi and L. Mili, "Robust dynamic mode decomposition," *IEEE Transactions on Communications*, vol. 10, 2022.

[19] S. I. R. Costa, "On closed twisted curves," *Proceedings of the American Mathematical Society*, vol. 109, no. 1, pp. 205–214, 1990.

[20] S. I. R. Costa, M. Muniz, E. Agustini, and R. Palazzo, "Graphs, tessellations, and perfect codes on flat tori," *IEEE Transactions on Information Theory*, vol. 50, no. 10, pp. 2363–2377, 2004.

[21] S. V. Zhidkov, "Analysis and comparison of several simple impulsive noise mitigation schemes for OFDM receivers," *IEEE Transactions on Communications*, vol. 56, pp. 5–9, 2008.

[22] P. J. Huber, "Robust estimation of a location parameter," *The Annals of Mathematical Statistics*, vol. 35, no. 1, p. 73–101, 1964.

[23] F. R. Hampel, "Contributions to the theory of robust estimation," Ph.D. dissertation, University of California, 1968.

[24] L. Mili, "Lecture notes in robust estimation theory," 2006.

[25] T. Kim and T. Adalı, "Approximation by fully complex multilayer perceptrons," *Neural Computation*, vol. 15, no. 7, pp. 1641–1666, 2003.

[26] A. Campello, C. Torezzan, and S. I. R. Costa, "Curves on flat tori and analog source-channel codes," *IEEE Transactions on Information Theory*, vol. 59, no. 10, pp. 6646–6654, 2013.

[27] T. Clarke, "Generalization of neural networks to the complex plane," in *International Joint Conference on Neural Networks*, vol. 2, 1990, pp. 435–440.

[28] A. Hirose, *Complex-Valued Neural Networks*, 2nd ed. Springer, 2012.

[29] S. Suresh, N. Sundararajan, and R. Savitha, *Supervised Learning with Complex-Valued Neural Networks*, 1st ed. Springer, 2013.

[30] C. Trabelsi, O. Bilaniuk, Y. Zhang, D. Serdyuk, S. Subramanian, J. F. Santos, S. Mehri, N. Rostamzadeh, Y. Bengio, and C. J. Pal, "Deep complex networks," in *International Conference on Learning Representations*, 2018.

[31] N. Guberman. (2016) On complex valued convolutional neural networks. arXiv 1602.09046.

[32] W. R. Hamilton, *Quaternion quotient lines tridimensional space time*. Hodges and Smith, 1853.

[33] M. Arjovsky, A. Shah, and Y. Bengio. (2016) Unitary evolution recurrent neural networks. arXiv 1511.06464.

[34] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *International Conference on Machine Learning*, vol. 37, p. 448–456, 2015.

[35] T. Berger and D. Tufts, "Optimum pulse amplitude modulation Part I: Transmitter-receiver design and bounds from information theory," *IEEE Transactions on Information Theory*, vol. 13, no. 2, pp. 196–208, 1967.

[36] I. Nazarov and E. Burnaev, "Bayesian Sparsification of Deep C-valued Networks," in *International Conference on Machine Learning*, vol. 119, 2020, pp. 7230–7242.

[37] M. W. Matthès, Y. Bromberg, J. de Rosny, and S. M. Popoff, "Learning and avoiding disorder in multimode fibers," *Physical Review X*, vol. 11, 2021.

[38] S. Costa, F. Oggier, A. Campello, J.-C. Belfiore, and E. Viterbo, *Lattices applied to coding for reliable and secure communications*. Springer, 2018.

[39] T. Parcollet, M. Ravanelli, M. Morchid, G. Linarès, C. Trabelsi, R. D. Mori, and Y. Bengio, "Quaternion recurrent neural networks," in *International Conference on Learning Representations*, 2019.

Anders M. Buvarp received an M.S. degree in communications engineering in 1993 from Lulea University of Technology, Sweden and, the Ph.D. degree in electrical engineering at Virginia Tech in 2023. His research interests include robust estimation theory, statistical signal processing, reconfigurable intelligent surfaces, electromagnetic information theory, time/frequency synchronization, massive MIMO, and channel estimation.

Lamine Mili (S'82-M'88-SM'93) received a B.S. degree from the Swiss Federal Institute of Technology, Lausanne, in 1976, and the Ph.D. degree from the University of Liege, Belgium, in 1987. He is presently a Professor of Electrical and Computer Engineering at Virginia Tech. His research interests include robust statistics, robust signal processing, radar systems, and power systems analysis and control.

Amir I. Zaghloul (S'68-M'73-SM'80-F'92-LF'11) is with the US Army Research Laboratory, Adelphi, MD, and has been with the Electrical and Computer Engineering Departments at Virginia Tech since 2001. He was at COMSAT Laboratories for 24 years performing and directing R&D efforts on satellite communications and antennas. He is a Life Fellow of the IEEE, Fellow of the Applied Computational Electromagnetics Society (ACES), Associate Fellow of The American Institute of Aeronautics and Astronautics (AIAA), and Member of Commissions A, and B, and Chair of Commission C of the US National Committee (USNC) of the International Union of Radio Science (URSI). He was the general chair of the 2005 "IEEE International Symposium on Antennas and Propagation and USNC/URSI Meeting," held in Washington, D.C., and served as an Ad Com member of the IEEE AP Society in 2006-2009. He also served on the IEEE Publication Services and Products Board (PSPB), the Editorial Board of "The Institute," and the IEEE History Committee. He was a Distinguished Lecturer for the IEEE Sensors Council. He received several research and patent awards, including the Exceptional Patent Award at COMSAT and the 1986 Wheeler Prize Award for Best Application Paper in the IEEE Transactions on Antennas and Propagation. Dr. Zaghloul received the Ph.D. and M.A.Sc. degrees from the University of Waterloo, Canada in 1973 and 1970, respectively, and the B.Sc. degree (Honors) from Cairo University, Egypt in 1965, all in electrical engineering.