Deep Learning-Based CS under IQ imbalance For Channel Estimation of mmWave phased arrays

I. Introduction

Ultra-high frequency bands, such as the Terahertz (Tz) band, in the context of information systems, provides enormous spectrum resources for ultra-high speed wireless communications, leading to a theoretical data rate on the order of 1Tbps [1] as well as providing physical gains by allowing a decrease in the size of antenna's, fascillitating the implementation of near-invisible and tiny communication devices.

With these advantages also come disadvantages, mainly in the realm of transceiving the signal. Of these challenges, one can name the non-linearity of power amplifier (PA), the imbalance between in-phase/quadrature (IQ) branches and the phase noise (PN) of local oscillator [1].

Due to an imbalance in the in-phase and quadrature modulating signals of the local oscillator (LO) at the RX, the effects of IQ imbalance (IQI) can occur. For a mathematical description of this modulation, we refer the reader to Appendix A. The main problem with IQ imbalance is insufficient image rejection due to the IQI. We refer the reader to [2] for a clear description of this imbalance and some nice graphics.

A more specific problem with mmWave systems, is the problem of channel estimation for beam alignment. In these systems, beams are formed directionally. Configuring these arrays is however complex, with restrictions on the hardware (due to restrictions in the phases of the phased array to restricted resolutions) as well as a requirement for few measurements to reduce overhead. Typical exhaustive beamscanning techniques require significant overhead [3].

II. Motivation and Contribution

A. Motivation

In the case of IQI, physically, the complexity of matching phase and amplitude between the in-phase and quadrature signals increases as the frequency of the LO increases. In low-cost applications, this may even become infeasible. For these cases, digital signal processing techniques can be preferred as an economical substitute. By pre-distorting the signal at the TX side and signal compensation at the RX side, the effects of the IQ-imbalance can be minimized thereby reducing the need for exact and robust hardware for the generation and mixing of the LO signal. Some existing methods include channel estimation [1] and self-interference coordination [2].

For the beamforming problem of mmWave phased arrays, compressed sensing (CS) is valuable in reducing the overhead and has been shown to lead to improvements when compared to exhaustive beamscanning [4]. The signal to be acquired is sparse in the Fourier representation and thus CS techniques can be applied in reconstructing the channel from fewer measurements than channel dimensions. Additionally, if one were to restrict the values of the CS matrix, one could achieve a CS matrix corresponding to a low-cost and complexity configuration, able to be deployed on a wide range of TX/RX setups.

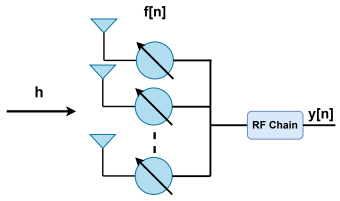

Fig. 1: Diagram of Phased Array Setup

B. Contribution

This paper proposes to attack the problem from a CS/neural network point of view. By taking advantage of both the sparse representation of the signal in the time domain, we apply CS techniques to reduce the sensing overhead. We furthermore apply an autoencoder based CS to not only efficiently find an optimal CS matrix, taking advantage of the structure of the IQ-imbalance, we additionally find a neural network decoder to negate the effects of the IQ-imbalance, and finally restrict the CS matrix to a discrete set to simulate the restricted resolution of the phased arrays enabling us to estimate the channel effectively, while additionally circumventing the effects of IQ imbalance.

Additionally, previous work has shown that deep learning techniques have been shown to reconstruct a sparse signal faster than classical optimization approaches such as OMP/ CoSAMP/ IHT/ Reweighted ℓ_1, ℓ_2 min/ SBL, further motivating the implementation of a neural network approach.

III. Proposed Methodology

In this section we discuss the methodology for the deep learning approach to IQ-imbalance.

A. Dataset and problem

In our problem we consider phased-array setup as shown in Figure 1. The channel is received by the phased array antenna at timestep n and processed by the RF chain as in Figure 5. The dataset is then given as follows.

Given is a dataset, denoted by $\{h\}_K$ of cardinality K, where each element in $\{h\}_K$, $h \in \mathbb{R}^{N\times 1}$ satisfies the condition that it's inverse Fourier transform

$$x = F^* h$$

is sparse. Here F denotes the unitary discrete Fourier transform (DFT) matrix, given by

$$W = (\frac{\omega^{jk}}{\sqrt{N}})_{j,k=0,\dots,N-1}$$

Given this dataset, the challenge is constructing an autoencoder which brings the given (dense) data into a lower dimensional representation, and which is able to reconstruct the original (sparse) signal with minimal NMSE. From the lens of the phased array antenna, the CS matrix is the phase configuration of the RX over a certain time range N, which will make up the $s_{baseband}$ signal. This signal is assumed to be modulated as shown in the Appendix, received at the RX, demodulated and the decoder attempts to reconstruct the signal from this demodulated version. Between the encoder and the decoder noise, as well as IQ-imbalance will affect the signal, the exact specifications of which are discussed later.

The final dataset is generated as stated above. We train the neural networks on a set of $K = 10000$ randomly generated vectors of size $N = 100$.

We refer to the next section for the architecture of the problem.

B. Autoencoder architecture

The autoencoder was trained and tested in the Pytorch Deep Learning Library [5].

We construct an autoencoder as follows. First, the encoder is restricted to a complex unitary matrix, defined by F_m as

$$F_m = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_M \end{bmatrix} \in \mathbb{C}^{M\times N}, w = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_M \end{bmatrix}, w_i \sim N(0, Q)$$

where each entry of f_i is restricted to

$$f_{i,j} = e^{jq_{i,j}},\ q_{i,j} \in [-\pi, \pi]\ \forall\ i\ \in 1, 2, \dots, M, j \in 1, 2, \dots N$$

to emulate the phased-array antenna structure. Note for now that we assume infinite resolution possible for the antenna.

Additionally, noise as well as IQ-imbalance is added to the compressed signal. We denote the noisy signal as

$$\mathbf{y_{noise}} = F_m h + w, \quad w \sim \mathcal{N}(0, Q)$$

where Q is a diagonal matrix with all equal terms. The signal affected by IQ-imbalance will be denoted as y_{IQ} and each entry modeled by

$$\mathbf{y_{IQ,j}} = K_1 \cdot y_j + K_2 \cdot y_j^*$$

The decoder is constructed from multiple linear neural network layers with ReLu activation functions, whose size grows with each layer. Note that in the Pytorch software, complex operations are not typically supported. To work around this issue, the complex and real components of the dataset were stacked on top of one another in a single vector. All layers whose size would be N in typical complex operations, were doubled to $2N$ in the neural network architecture to deal with these complex operations. Additionally, the complex unitary encoding layer was restricted to purely possible complex operations i.e. given the original matrix F_m, a new matrix was constructed for the neural network as

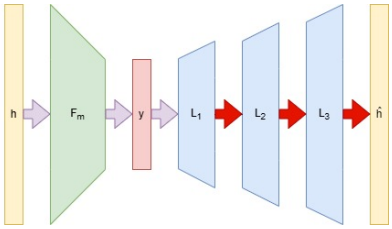

Fig. 2: Autoencoder architecture. Green denotes a complex unitary matrix operation, blue denotes linear layers and red arrows denote ReLu activation functions

$$F_{torch} = \begin{bmatrix} \Re(F_m) & -\Im(F_m) \\ \Im(F_m) & \Re(F_m) \end{bmatrix}$$

where $\Re(F_m), -\Im(F_m)$ denote the real and imaginary parts of the matrix respectively. A schematic representation of the autoencoder is given in Figure 2

In Appendix B one can find our hyperparameters for training purposes.

C. Sparsity of x

As discussed before, the dataset $\{h\}_N$ is generated by generating a sparse x and defining the DFT of x as h. that is to say

$$h = Fx$$

the sparsity of x was varied in order to explore how the effect of noisy signals or signals impacted by an IQ imbalance differ when the underlying structure of the incoming signal is more or less complex. The sparsity ranges that were tested were $[(3-5), (5-7), (7-9), (10-30)]$. These sparsities were chosen to show the effect of different sparsities on models subject to gaussian additive noise or IQ imbalances. As well as the effect of different compressed dimensions when the compressed vector has different complexities of underlying structure.

D. Signal to noise ratio

The signal to noise ratio is defined as the ratio of the power of the signal to the power of the noise. As these are randomly generated signals with 0 mean, the power of the signal is equal to the expectation squared, which in turn is equal to the

variance. The signal to noise ratio can further be expressed in decibels, that leads to:

$$SNR_{dB} = 10log_{10}\left(\frac{P_{signal}}{P_{noise}}\right) = 10log_{10}\left(\frac{\sigma^2_{signal}}{\sigma^2_{noise}}\right)$$

the set of SNR_{dB} tested was $[2, 5, 8, 11, 14, 17, 20]$. These ratios were chosen, as it gives a complete overview of noise levels, from barely more signal than noise, to practically no noise. Additionally the effects of these noise levels on autoencoders were tested by Cayamcela et al. [6]

E. IQ imbalance

As discussed in the Appendix A, the IQ imbalance is defined as

$$\mathbf{y_{IQ,j}} = K_1 \cdot y_j + K_2 \cdot y_j^*$$

where:

$$K_1 = \frac{1}{2}(1 + \epsilon e^{-j\phi}), \qquad K_2 = 1 - K_1^*$$

where ϵ denotes the amplitude imbalance and ϕ denotes the phase imbalance between the IQ branches.

This corresponds to a rotation in the imaginary plane when y is an imaginary number. ϵ was varied on the interval: $0.8 < \epsilon \leq 1$ and ϕ on $0 < \phi < \frac{\pi}{8}$. Due to time constraints, ϕ and ϵ were varied together and not independently according to:

$$\epsilon = 1 - 0.2 \times shift, \quad \phi = shift \times \frac{\pi}{8}$$

with $shift$ varying from 0 to 1 in 6 steps.

F. Discrete sensing matrix

As an extension to the restrictions given on the sensing matrix by restricting the values of q to the continuous set $R = [-\pi, \pi]$, further restrictions can be imposed on the sensing matrix, namely if we restrict ourselves to the discrete set

$$R_D = \{0, \pm\frac{1}{2}\pi, \pm\pi\}$$

This corresponds to the use of 5-bit low-resolution phase shifters that further result in low cost and complexity hardware. The (non-restricted) optimization problem can be defined as

$$\begin{aligned} &\min_{q,w,b} \frac{1}{N}\|h - f(h, w, b, q)\|_2^2 \\ &s.t. \quad q_{i,j} \in \{0, \pm\frac{1}{2}\pi, \pm\pi\} \end{aligned} \tag{1}$$

where $f(h, w, b, q)$ denotes the neural network with w, b, q denoting weights, biases and q-values respectively. We can rewrite the constraint equivalently as

$$g(q) = |q(q - \frac{\pi}{2})(q - \pi)(q + \frac{\pi}{2})(q + \pi)| = 0$$

We can then formulate the optimization problem as

$$\min_{q,w,b} \frac{1}{N}\|h - f(h, w, b, q)\|_2^2 + \lambda g(q) \tag{2}$$

Resulting in an unconstrained optimization problem suitable for the Pytorch architecture. We finally map the values to their closest discrete counterparts (in the ℓ_2 sense) to ensure a fully discrete matrix.

IV. Results

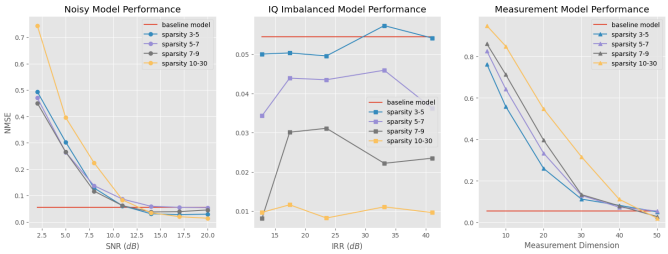

Fig. 3: NMSE of trained models for continuous phase set. Varied SNR (left), IQ imbalance (middle), Measurement dimensions (right)

In the leftmost graph in figure 3 It can be seen that all models perform much better with a larger SNR than with a lower SNR, which is in line with expectations. It can also be noted that for models corresponding to x's with sparsities: $[3-5,\ 5-7,\ 7-9]$, the performance does not differ much. While the model corresponding to an x with sparsity $(10-30)$ performs significantly worse for $SNR \leq 7.5$, and comparable for $SNR > 10$, even reaching the lowest NMSE at the highest SNRs. This might be due to the fact that, when there h is build from a less sparse x the signal will have a greater variance, as a result, the absolute amount of added noise will be greater.

In the center graph in figure 3 it can be seen that there is no consistent trend for different levels of IQ imbalance. This means that the decoder is able to accurately compensate for the IQ imbalance, regardless of the size of the imbalance. It is noteworthy that for decreasing sparsities of x, the performance of the models increases. This can be explained by the fact that we evaluate the models on normalized MSE. The datasets built from less sparse x have a larger variance, and are thus divided by a larger normalizing term.

Lastly in the rightmost graph in figure 3 it can be seen that all models perform better for larger compression dimensions, or larger sensing matrices. The smallest compression dimension explored is 5, while only the sparsest x has a sparsity of 5, thus for the less sparse x's, the compressed representation can never capture all of the underlying structure of the corresponding h and there will always be a loss of information. This loss of information is reduced for larger sensing matrices. In

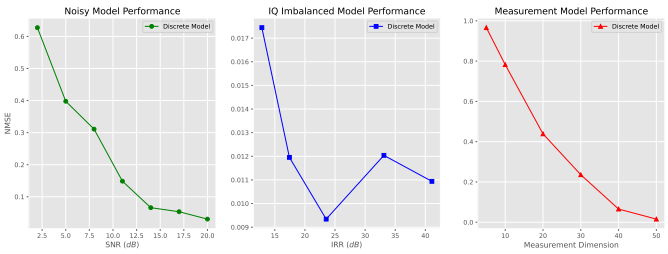

Fig. 4: NMSE of models with 5-bit resolution phased-array antenna for various SNR (left), IQ imbalance (middle) and measurement dimensions (right)

fact, for the largest measurement dimension of 50, the model corresponding to an x with a sparsity of $10-30$ performs best. This is likely due to one of two effects. The first is that the MSE of all models at that compression dimension becomes very similar, but due to the larger variance of the h built from a less sparse x, the corresponding model will have a smaller NMSE. The second effect might be overfitting. It is known that the minimum amount of measurements needed to be able to recover all s-sparse vectors is $2s$ [7]. When the encoding dimension (corresponding to the number of columns in our measurement matrix) becomes much larger than this $2s$, this leaves a lot of room in the compressed representation of our vector to encode weaker patterns present in the dataset purely by chance. As a result the models encoding a very sparse x in a very large vector, might be more prone to overfitting.

Figure 4 shows the performance of the same autoencoders as before, but with the entries of the encoding matrix constrained to the set $\{\pm\pi, \pm\frac{1}{2}\pi, 0\}$. These models are trained on a dataset corresponding to an x with sparsity of $(7-9)$. It is good to note that the same trends are present in terms of the performance with varying SNRs, IQ imbalances and encoding dimensions. Furthermore, compared to a model corresponding to an x with the same sparsity, but without the discrete constraint on the encoding matrix, there is very little reduction in performance. the unconstrained model reaches an NMSE of ≈ 0.45 at a SNR of $2dB$, increasing to NMSE ≈ 0.05 at $SNR = 20dB$, while the constrained model reaches performances of NMSE ≈ 0.06 and NMSE ≈ 0.05 for the same noise levels. For varying encoding dimensions a very similar pattern emerges, with the NMSE of the constrained model being roughly 0.15 higher for the very smallest encoding dimensions, and converging to the same score for higher encoding dimensions. This is likely due to the fact, that the individual elements of the encoding or measurement matrix are constrained to discrete values, but when the size of the matrix increases, so does the flexibility of the model. For the case of varying IQ imbalances, the constrained model actually performs better than the unconstrained model. However, as mentioned before, the decoder is able to effectively compensate for the IQ imbalance at all levels. Combined with the fact that the performance of the constrained and unconstrained models lie quite close together, this difference might purely be by change, as differences in convergence always emerge when training machine learning models.

V. Theoretical Guarantees

In this section we provide some theoretical insights into the "quality" of the found CS matrices. Note we show only these guarantees for the limited resolution CS matrices. Some options include Spark, Null-space property (NSP), restricted isometry property (RIP) and coherence. Due to the arguably more insightful nature of RIP than Spark and NSP (as NSP is implied by RIP), we show the restricted isometry constraint (RIC) of a selection of matrices. Due to the combinatorial nature of finding the RIC, we show only the RIC of matrices corresponding to sparsities of 3, thus checking $\binom{100}{3} = 3921225$ combinations of columns of the matrix F_m. We additionally show the coherence μ. Both are given in I and II respectively. We see that the RIP property is satisfied for almost all matrices, except for higher encoding dimensions. The RIP property satisfaction guarantees the NSP property which in turn guarantees ℓ_1 norm algorithm reconstruction. Of course, this is slightly irrelevant for our algorithm since we use an autoencoder approach but the magnitude of the IRC does indicate the "quality" of the sensing matrix. It is remarkable that then the quality is so poor, as lower values correspond to better CS matrices. Additionally, we note that we see similar results for coherence, with the higher encoding dimensions corresponding to high μ whereas in the case of IQ imbalance only slight changes are observed. The same can be seen for the SNR as well.

VI. Discussion

As can be seen in the results section, a learned autoencoder is able to effectively compensate for an IQ imbalance applied to the compressed representation of the input vector h. While a learned decoder is able to compensate for a constant IQ imbalance, it is unable to compensate for additive gaussian noise applied to the compressed version of the input. This can be seen as the increasing NMSE on the validation set with increasing SNR. We also note a positive correlation between NMSE and the dimension of the compressed vector. This points to a tradeoff between the level of compression and the quality of the recovered signal. Lastly, we note that in the case of a discrete sensing matrix, we are still able to build a model that performs comparable to models without any discrete constraints on the sensing matrix. In fact, in some cases the constrained model performs even better than the unconstrained models, although this might only be by chance.

A. Recommendations

At the edges of the explored ranges, some behaviors emerge which are not in line with the rest of the findings of this paper. It would be worthwhile to explore what happens at even larger encoding dimensions, to explore what causes the model trained on less sparse x to perform better.

Most interesting perhaps is the fact that our results point to the fact that using a sensing matrix populated with discrete values, signals can still be reconstructed with a high degree of accuracy, in the case that there the noise present in the measurement is not too great, more research into the noise robustness of compressed sensing under IQ imbalance with discretely constrained sensing matrix is needed. On top of this, it would be worthwhile to explore these same trends with fewer constraints on the training data, for instance, what happens to the quality of the models when the sparsity of x becomes larger, or when we do not assume that noise is added to the compressed but to the uncompressed representation of the signal.

APPENDIX

A. IQ imbalance model

For the purposes of this paper, we regard the direct-conversion architecture of signal transceiving. In the direct-conversion architecture, a baseband signal, denoted by $s_{baseband}(t)$, has a real and imaginary part, $I(t)$ and $Q(t)$.

$$s_{baseband}(t) = A(t)e^{j\phi(t)} = I(t) + jQ(t) \tag{3}$$

Another way of viewing these would be the in-phase $I(t)$ and quadrature $Q(t)$ parts of the original signal. The in-phase and quadrature parts of the signal are then separated and modulated with a sine and cosine function of a specific frequency respectively. Formally,

$$s_{RF}(t) = Re\{s_{baseband}(t)e^{j2\pi f_c t}\} = \tag{4}$$

$$I(t)\cos(2\pi f_c t) - Q(t)\sin(2\pi f_c t) \tag{5}$$

Following this modulation, the problem is then to down-convert this signal back into the baseband. The way that this is typically handled is via a Local Oscillator (LO). The local oscillator oscillates at the frequency at which the signal is modulated. By then taking this signal, and adding a $90°$ phase-shift we get a cosine and sine term. By then multiplying our original signal with these modulation terms, we find the following results

$$I_{received}(t) = s_{RF}(t) \cdot \cos(2\pi f_c t) = \tag{6}$$

$$A(t)\frac{1}{2}[\cos(\phi(t)) + \cos(4\pi f_c t + \phi(t))] \tag{7}$$

$$Q_{received}(t) = s_{RF}(t) \cdot -\sin(2\pi f_c t) = \tag{8}$$

$$A(t)\frac{1}{2}[\sin(\phi(t)) - \sin(4\pi f_c t + \phi(t))] \tag{9}$$

Notice how if a low-pass filter is applied to both signals, the original baseband signal is recovered completely. We refer to Figure 5 for a visual representation of this process. However, if even a slight imbalance occurs in either the phase or amplitude of the signal produced by the LO, the signal quality is tremendously effected and can severly degrade receiver performance. For brevity reasons, we refer the reader to [2] for a complete description, but if we represent the LO signals as

$$cos(2\pi f_c t), \quad -\epsilon sin(2\pi f_c t + \phi)$$

where ϵ denotes the amplitude mismatch and ϕ the phase mismatch the effect of this IQ imbalance can be modelled as

$$s_{IQI}(t) = K_1 s_{baseband}(t) + K_2 s^*_{baseband}(t) \tag{10}$$

where

$$K_1 = \frac{1}{2}(1 + \epsilon e^{-j\phi}), \quad K_2 = \frac{1}{2}(1 - \epsilon e^{j\phi})$$

from which follows

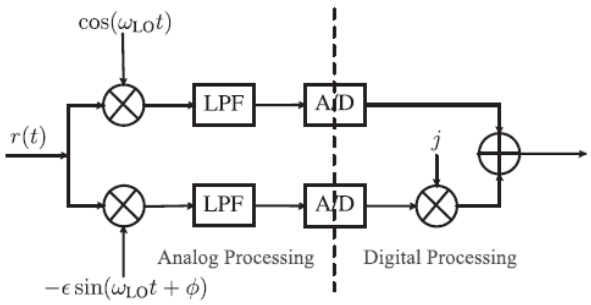

Fig. 5: IQI model from [2]. LPF denotes low-pass filter, A/D denotes the analog-to-digital converter. Note that we are working with the reconstructed signal on the right, and we assume perfect LPF and A/D processing.

$$K_1 = 1 - K_2^*$$

B. Training Hyperparameters

For training purposes, the Adam optimizer [8] was used with learning rate 0.001, $\beta_1 = 0.9, \beta_2 = 0.999$. The standard MSE loss function was used, where we compared $\hat{h}$ (outputs) to h (targets).

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(h_i - \hat{h}_i)^2$$

C. Discrete sensing matrix

The problem of restricting the values to this discrete set is interesting as of course it is non-trivial to find gradients for discrete variables, and optimizing this is also combinatorial in nature. One approach may be using the Pytorch Quantization functionality, however this functionality is not made for our use case and is still in beta and additionally does make use of the neural network optimization functionality.

By doing so we relax the problem. λ is then a hyperparameter which decides how "important" the discrete set is in the optimization problem. If one were to set this too high, the problem converges to solutions with (close-to) discrete values for the F_m matrix, but with terrible loss. Setting this parameter too low results in solutions with no discrete values.

Of course, regardless of how large or small lambda is, due to the nonconvex nature of the optimization problem, the solution will never converge to a truly discrete set, thus the values of q are mapped in a final stage to their closest discrete counterparts which was found to work well.

D. Reconstruction of signals

Figure 6 shows the DFT of the input signal and the reconstructed signal plotted over each other for a model trained on noiseless data. Similarly figure 7 shows the same for a model trained with gaussian noise added to the compressed representation of the vector. In both cases, there is some of all frequencies present in the output signal, which were not

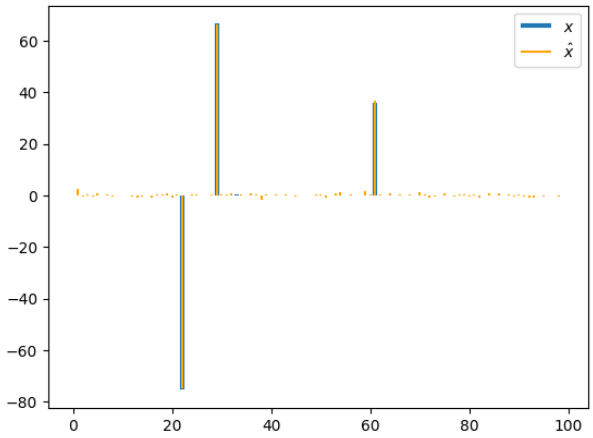

Fig. 6: In blue, the sparse DFT of an input signal h. In orange, the DFT of of the reconstructed signal $\hat{h}$ made with a model trained on noiseless data

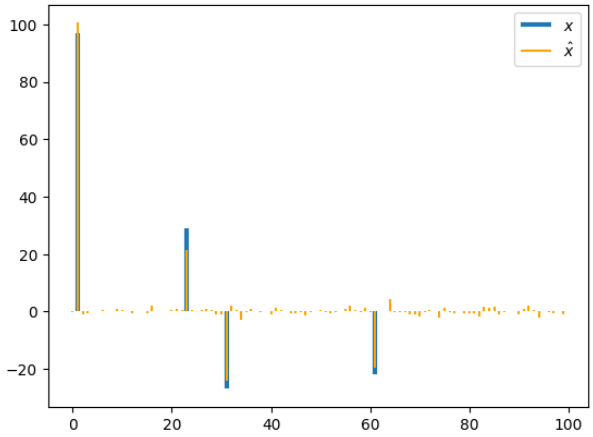

Fig. 7: A visual representation of the sparse DFT of the original signal h and a reconstructed $\hat{h}$ made with a model trained on noiseless and noisy data.

present in the input data. Furthermore, it can be noted that the noiseless model is able to recover the original frequency distribution much more accurately, which can be seen as the blue bars overlapping almost perfectly with the orange bars, whereas in the noisy version, the overlap is much less accurate. The presence of the incorrect frequencies could probably be removed quite simply in both cases, whereas the error on the true frequencies is much more problematic.

E. RIC

Here we showcase the RIC for a range of the matrices found for the discrete encoding case

F. Coherence μ

TABLE I: RIC values for different encoding dimensions, SNR, and IQ imbalance

Encoding Dim		SNR		IQ Imbalance	
Value	RIC	Value	RIC	Value	RIC
50	0.99999994	20	0.99999994	Infinite	0.99999994
40	1.0161078	17	0.99999994	41.09	0.9999999
30	1.0	14	0.9999999	33.12	0.99999994
20	1.1904368	11	0.99999994	23.53	0.99999994
10	1.8087626	8	0.9999999	17.43	0.9999999
5	1.9999998	5	1.0	12.84	0.99999994
		2	0.99999994		

TABLE II: Coherence μ values for different Encoding Dimensions, SNR, and IQ Imbalance (Discrete model)

Encoding Dim		SNR		IQ Imbalance	
Value	μ	Value	μ	Value	μ
50	0.6746851	20	0.6675328	Infinite	0.65115285
40	0.7762087	17	0.6746852	41.09	0.66272163
30	0.76011705	14	0.6511529	33.12	0.66753286
20	0.86313385	11	0.66753286	23.53	0.7299314
10	1.0000001	8	0.6511529	17.43	0.6811755
5	0.99999994	5	0.6905071	12.84	0.70256686
		2	0.6811755		

REFERENCES

[1] Z. Sha and Z. Wang, "Channel estimation and equalization for terahertz receiver with rf impairments," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 6, pp. 1621–1635, 2021.

[2] A.-A. A. Boulogeorgos, V. M. Kapinas, R. Schober, and G. K. Karagiannidis, "I/q-imbalance self-interference coordination," *IEEE Transactions on Wireless Communications*, vol. 15, no. 6, pp. 4157–4170, 2016.

[3] N. J. Myers, A. Mezghani, and R. W. Heath, "Spatial Zadoff-Chu Modulation for Rapid Beam Alignment in mmWave Phased Arrays," in *2018 IEEE Globecom Workshops (GC Wkshps)*. IEEE, pp. 1–6. [Online]. Available: https://ieeexplore.ieee.org/document/8644177/

[4] N. J. Myers, Y. Wang, N. González-Prelcic, and R. W. Heath, "Deep learning-based beam alignment in mmwave vehicular networks," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 8569–8573.

[5] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. [Online]. Available: http://arxiv.org/abs/1912.01703

[6] M. E. Morocho-Cayamcela, J. Njoku, J. Park, and W. Lim, "Learning to communicate with autoencoders: Rethinking wireless systems with deep learning," 02 2020.

[7] R. Gribonval and M. Nielsen, "Highly sparse representations from dictionaries are unique and independent of the sparseness measure," *Applied and Computational Harmonic Analysis*, vol. 22, no. 3, pp. 335–355, 2007. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1063520306001175

[8] D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. [Online]. Available: http://arxiv.org/abs/1412.6980