An Autoencoder for Transmission of Vocoder Features over Radio Channels

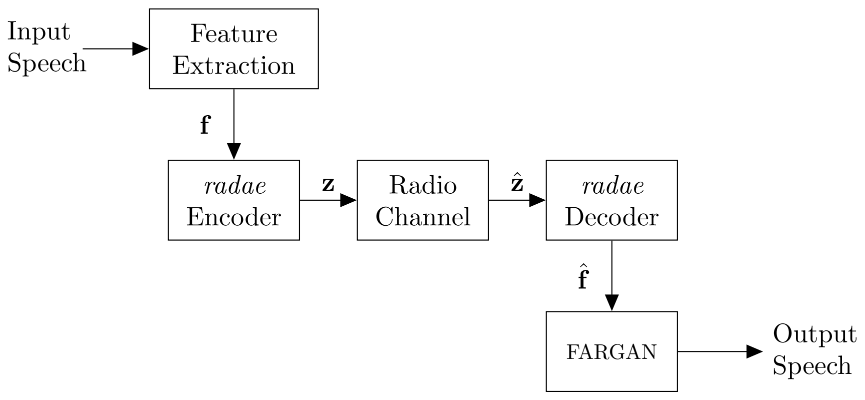
May 22, 2024

1 Introduction

This report presents an autoencoder derived from RDOVAE [3] to send speech over radio channels. Our goal is to determine if reasonable speech quality can be obtained over a channel of bandwidth $B < 3000$ Hz and SNR around 0dB, roughly the lower limit of Single Side Band (SSB) - a popular power and bandwidth efficient form of speech communication.

This document describes the Radio Autoencoder *radae* system, discusses channels simulation and calibration, and describes how we combined ML with classical DSP to create a practical proof of concept system that can send speech over real world VHF and HF radio channels. Finally, we compare the capabilities of the Radio Autoencoder to existing systems for speech communication over HF and VHF/UHF radio.

Figure 1: Radio Autoencoder Concept



The encoder (Figure 1) takes as input a typical set of vocoder features (short term spectrum, pitch, voicing), then applies time based prediction and transforms to produce a set of parameters that can be sent over a channel. This is similar to vocoders using classical DSP, except Machine Learning (ML) allows us to learn non-linear transforms and prediction, which tend to be more powerful.

In conventional digital speech systems, after the transformation/prediction stage we then quantise to a low bit rate, then use Forward Error Correction (FEC) and modems to send the bits over a channel. However our work takes a novel twist – we train the autoencoder to generate PSK symbols that we send over the channel. It effectively combines quantisation, channel coding, and modulation. The symbols from the autoencoder tend to cluster around +/-1 like BPSK but are continuously valued, so can be considered discrete time, continuously valued PSK.

The encoder and decoder are trained together as an autoencoder with the loss function $L(\mathbf{f}, \hat{\mathbf{f}})$ applied to the vocoder features. We employ the FARGAN vocoder for speech analysis and synthesis, however the concept is applicable to any neural and even classical vocoder with a similar feature set.

Given a vector of vocoder features $\mathbf{f}$, we use an encoder to map them to a dimension d latent vector $\mathbf{z}$ where d is even. Unlike digital modulation, each element z_i of $\mathbf{z}$ is continuously valued and not constrained to a discrete set of points. For bandwidth efficient transmission over the channel the elements of $\mathbf{z}$ are mapped to $d/2$ complex symbols $\mathbf{q}$. Compared to classical digital modulation, the elements of $\mathbf{z}$ can be considered BPSK symbols (continuously valued, analog bits), and the elements of $\mathbf{q}$ analog QPSK symbols.

2 Simulation of AWGN Channels

The autoencoder output $\mathbf{z}$ is updated every $T_z = 1/R_z$ seconds, giving a BPSK symbol rate of:

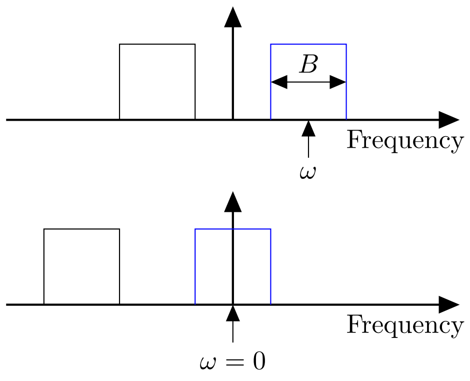
$$R_b = d/T_z \tag{1}$$

For example with $T_z = 0.04, d = 80, R_b = 2000$ symbols/s. The QPSK symbol rate is given by:

$$R_q = \frac{d}{2T_z} \tag{2}$$

For example with $T_z = 0.04, d = 80, R_q = 1000$ symbols/s.

Figure 2: Real sampled off-air signal. We are interested in the blue bandpass interval of bandwidth B, which is single sided and hence complex valued. After shifting to baseband, it's power is unchanged, and it remains complex valued.



We wish to simulate an AWGN channel with a user-defined E_b/N_0, where E_b is the energy of each BPSK symbol, and N_0 is the noise power per unit bandwidth. Consider a real valued signal sampled off air (Figure 2). We will follow convention and define signal and noise power in the "single sided" bandpass interval of the frequency spectrum with bandwidth B centred on ω. As the interval is single sided, we must use complex valued quantities to represent it.

We wish to simulate a bandpass AWGN channel at baseband ($\omega = 0$). This implies a frequency shift of the complex valued signal, but the signal remains complex valued and it's power is unchanged. The negative frequency component on the LHS of Figure 2 is redundant and after frequency shifting can be removed by filtering.

Note that even at baseband we must use complex valued quantities for the signal and noise to represent a bandpass signal of bandwidth B. As a counter-example, given a fixed sample rate B and noise power N, a real valued noise sequence can only represent a bandwidth of $B/2$ which results in doubling the noise density $N_0 = N/(B/2) = 2N_0$ compared to a complex valued noise sequence with the same power.

The energy of each PSK symbol E_q is the signal power S divided by the PSK symbol rate R_q. The noise per unit bandwidth is the total noise power N divided by the bandwidth B of the system. If we are simulating at one sample per PSK symbol, $B = R_q$:

$$\begin{aligned} \frac{E_q}{N_0} &= \frac{S/R_q}{N/R_q} \\ &= \frac{S}{N} \\ &= \frac{A_q^2}{\sigma^2} \end{aligned} \tag{3}$$

where A_q is the amplitude of each PSK symbol and $\sigma^2 = N$ is the variance of the complex valued noise (mean noise energy per sample). Given a set point E_q/N_0:

$$\sigma = \frac{A_q}{\sqrt{E_q/N_0}} \tag{4}$$

If we use a BPSK system model, we note the amplitude of each BPSK symbol $A_b = A_q/\sqrt{2}$, and the energy of each BPSK symbol $E_b = E_q/2$. Subsituting into 4:

$$\sigma = \frac{A_b}{\sqrt{E_b/N_0}} \tag{5}$$

Which allows us to simulate using a set point E_b/N_0. The complex noise sample r_i can be generated as:

$$r_i = \frac{\sigma}{\sqrt{2}}(\mathcal{N}_{2i}(0,1) + j\mathcal{N}_{2i+1}(0,1)) \tag{6}$$

where $\mathcal{N}_i(0,1)$ is the $i-th$ sample of a unit variance, zero mean, real Gaussian noise source. Note the noise power is split evenly between the real and imaginary arms. Our symbols passing through an AWGN channel can be simulated at complex baseband as:

$$\begin{aligned} \hat{z}_i &= z_i + r_i \\ \hat{q}_i &= q_i + r_i \end{aligned} \tag{7}$$

If the noise is zero mean, we can estimate σ^2 over K noise samples r_i as:

$$\sigma^2 = E[|r_i|^2] = \frac{1}{K}\sum_{i=0}^{K-1} |r_i|^2 \tag{8}$$

2.1 SNR Measurement

In order to compare with other methods of speech communication that have varying bandwidths B, it is useful to formulate expressions for estimating SNR from the BPSK and QPSK symbols. The Signal to Noise ratio (SNR) is given by:

$$\begin{aligned} \frac{S}{N} &= \frac{E_b R_b}{N_0 B} \\ &= \frac{E_q R_q}{N_0 B} \end{aligned} \tag{9}$$

A noise bandwidth B needs to be selected; common choices are $B = R_b$, in which case $S/N = E_b/N_0$; for HF radio $B = 3000$ Hz to compare with existing analog and digital voice waveforms; or $B = 1$ to obtain a normalised C/N_0 carrier power to noise density ratio - useful for comparing waveforms with different bandwidths.

At one sample per symbol, the power, the mean energy of each QPSK symbol over a window of K samples is given by:

$$E_q = E[|q_i|^2] = \frac{1}{K}\sum_{i=0}^{K-1}|q_i|^2 \tag{10}$$

Note the variance function should not be used to calculate E_q, as we cannot guarantee q_i is zero mean. As each QPSK symbol contains 2 BPSK symbols, the energy is split evenly:

$$E_b = E_q/2 \tag{11}$$

For example if the symbol amplitude is $A = 1, E_b = A^2 = 1$, then $E_q = 1+1 = 2$.

For transmission over multipath channels using OFDM we arrange the QPSK symbols as N_c parallel carriers, each running at a symbol rate of $R_s = R_q/N_c$ symbols/s, where R_s is chosen based on delay spread considerations. Typical values for HF modems are $N_c = 20$ and $R_s = 50$ Hz. However the OFDM carriers are arranged such that the total symbol rate over the channel remains constant. So for a given signal power E_q and E_b remain constant (Table 1).

Waveform	N_c	R_s	R_q	R_b	E_q	E_b
Single Carrier BPSK	1	-	-	2000	-	$S/2000$
Single Carrier QPSK	1	-	1000	2000	$S/1000$	$S/2000$
OFDM QPSK	20	50	1000	2000	$S/1000$	$S/2000$

Table 1: E_b and E_q examples for single and multi-carrier OFDM waveforms for constant carrier power S

2.2 Calibration and Testing

In order to evaluate the ML system early in the development process it is important to ensure the noise is correctly calibrated. The expressions above can be used to check the noise injection process:

1. Set a target E_b/N_0 for the simulation run, and calculate σ using (4).
2. Establish the equivalent target SNR from (9) evaluated using the target E_b/N_0.
3. After the simulation run measure $E_q = E[|q_i|^2]$ over a sample of transmitted symbols. Note that in general $E_q \neq 2$ as the encoder outputs continuous values.
4. Calculate measured SNR using (9) and compare.

The calibration of the noise injection can be checked by replacing the encoder output z_i with discrete PSK symbols to create a digital modem, then measuring the BER at E_b/N_0 points. The theoretical BER over an AWGN channel is:

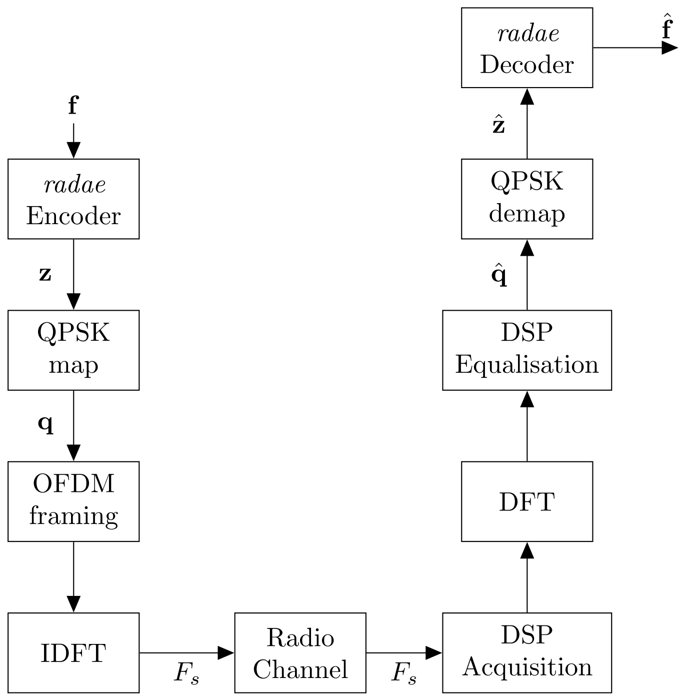
$$BER = 0.5erfc(\sqrt{E_b/N_0}) \tag{12}$$

For a multipath channel:

$$BER = 0.5\left(1 - \sqrt{\frac{E_b/N_0}{E_b/N_0 + 1}}\right) \tag{13}$$

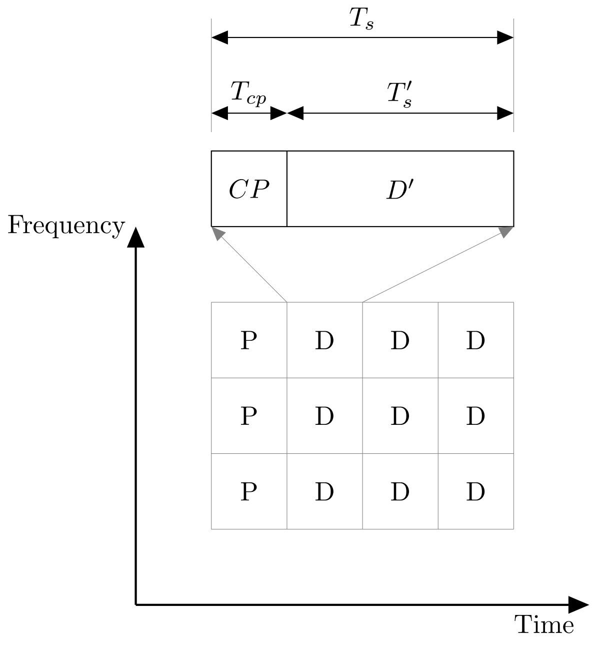
3 Proof of Concept Radio Frequency Tests

Figure 3: ML combined with OFDM and classical DSP synchronisation. Transmitter (Tx) is on the LHS, at RHS is the Receiver (Rx). The sample rate over the channel is $F_s = 8000$ Hz.



Impressive results have been obtained from symbol rate simulations of an OFDM modem, which assumed ideal synchronisation. We would like to verify these results using real radio signals in Over The Cable (OTC) and Over the

Figure 4: OFDM modem frame, P denotes pilot symbol, D payload data symbols. This example has a modem frame of $N_s = 4$ symbols, and $N_c = 3$ carriers. Each symbol D or P is comprised of a T_{cp} second Cyclic Prefix and T'_s second symbol D'.



Air (OTA) tests. This requires building up a rate F_s system, and synchronisation subsystems. For our proof of concept system, the choice was made to use OFDM to handle HF multipath channel, and classical DSP pilot symbol based synchronisation, although we acknowledge potential for ML based synchronisation in future iterations. The combined ML and OFDM/DSP system is illustrated in Figure 3.

The goal is to compare speech quality to SSB at $E_b/N_0 = 0dB$ (approx -3dB SNR in a 3000Hz BW), and work through any issues that prevent the system working over real radio channels. PAPR optimisation will be ignored for the first iteration, as our initial goal is to verify the low E_b/N_0 results suggested by the symbol rate simulations.

The general design of the OFDM frame in Figure 4. QPSK symbols are mapped to a matrix of parallel carriers at a relatively low symbol rate in order to successfully pass through the target HF multipath channel. Pilot symbols are periodically inserted into each OFDM carrier. At the receiver the pilots are used to estimate the time varying phase of the channel (equalisation), and for initial acquisition (coarse frequency, and frame sync) of the received signal. The disadvantage of pilot based schemes is they consume carrier power that would otherwise be available for the data symbols, and require the symbol rate and hence overall RF bandwidth to be increased to maintain the payload data rate. After the IDFT stage a cyclic prefix is inserted to accommodate inter symbol interference. The cyclic prefix also consumes carrier power and requires an increase in RF bandwidth. The OFDM waveform details, including overheads, are explained in detail in the waveform design spreadsheet [2]. The *Candidate 2* waveform has a RF bandwidth of approximately 1500 Hz, with 500Hz due to synchronisation overhead.

A pilot based sync system was built in PyTorch, and is used for coarse and fine timing, phase and amplitude equalisation. Unlike classical PSK, the ML network is likely to be sensitive to amplitude variations. Phase equalisation also allows small frequency offsets (± 2 Hz) to be handled, sufficient for tests with short samples.

Several phase estimators were prototyped, and evaluated using BER measurements. Maintaining low loss synchronisation at low E_b/N_0 is challenging. Using per-carrier phase estimation makes the system less dependant on fine timing accuracy and gives us the ability to handle multipath, but has higher loss than algorithms that consider all carriers at the same time. As further work a lower latent dimension d and higher E_b/N_0, would allocate more power to pilots, and result in less carriers.

Figure 5 and 6 plots the simulated performance of the OFDM pilot based synchronisation system on AWGN and multipath channels. The *genie* curve is the baseline rate F_s OFDM system with ideal sync, and matches the theoretical BPSK BER curve. Using this baseline system to send ML symbols, we obtain intelligible speech at $E_b/N_0 = -6$, which corresponds to BER=0.24 in a digital modem. We can use the BER=0.24 line to estimate the synchronisation loss at this operating point, which for the *mean6* and *LS* algorithms under realistic conditions is around 2 to 2.4 dB on AWGN, and 2dB on the multipath channel. The *LS* works better on fast fading multipath channels, so was chosen for the HF OTA trials.

Table 2 summarises the sync losses. These are comparable to classical DSP OFDM data modems. Combined with L_p, we estimate a total sync loss of 4dB for this first pass of the ML system combined with classical pilot based synchronisation.

Symbol	Loss (dB)	Description
L_p	0.97	Pilot symbol overhead
L_{cp}	0.97	Cyclic prefix overhead
L_{eq}	2.00	*LS* equalisation (multipath)

Table 2: *Candidate 2* OFDM waveform design sync losses, around 4dB total.

Figure 7: Over The Cable (OTC) VHF test. The *radae* transmitter output is upsampled and shifted to 144.5 MHz using a HackRF. The RF receive power and hence SNR is set by a step attenuator before being received by a RTL-SDR and *gqrx* SDR application. Wave files recorded off air by *gqrx* are presented to the *radae* receiver.

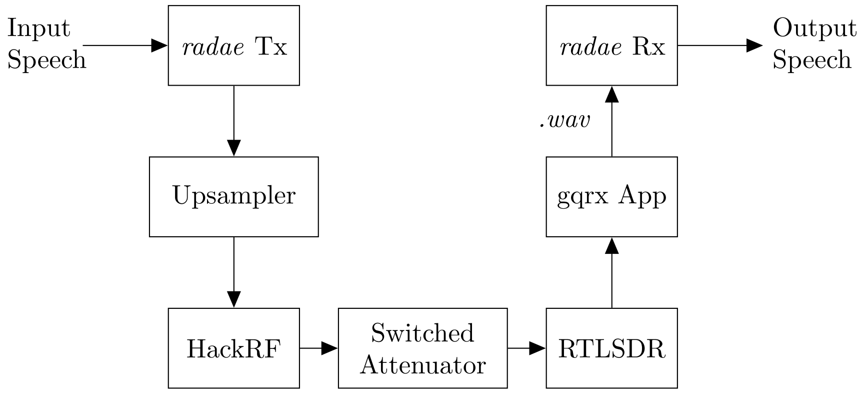
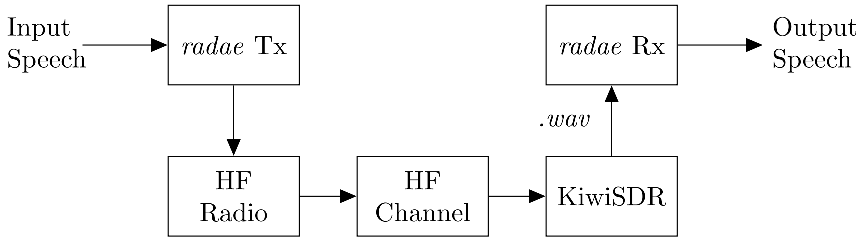


Figure 8: Over The Cable (OTA) HF test. The *radae* transmitter output is fed to the USB sound interface of a COTS HF Radio where it is shifted to HF, amplified, and transmitted over various real world HF channels to a remote KiwiSDR receiver. Wave files recorded off air by the KiwiSDR are presented to the *radae* receiver.



3.1 March 2024 Results

In March 2024, the system was tested over a VHF OTC AWGN path and OTA on a variety of real world Australian HF multipath paths. Figure 7 and Figure 7 describe the experimental configurations. Wave files containing a sine wave tone, compressed SSB, and the *radae* signal were sent over the same channel at the same time. The speech quality obtained was consistent with the simulations and competitive with SSB sent over the same channel at the same time.

Take aways:

1. We have combined ML vocoder, ML autoencoder and classical DSP OFDM to build a system capable of sending speech over radio channels. It is robust to AWGN and multipath channel impairments. Our initial simulation, VHF OTC, and HF OTA tests suggest performance competitive with the analog SSB at the same C/N_0.

2. The total "sync losses" due to pilot, cyclic prefix overheads and non-ideal equalisation are around 4dB. It is conceivable these can be reduced using ML rather than classical DSP.

3. The ML algorithms run at a 40ms frame rate (with some DSP at the OFDM 20ms symbol rate), resulting in modest CPU requirements. The PyTorch simulation code runs several times faster than real time in inference mode.

4. Simulation and initial HF OTA results show surprisingly good performance on multipath channels where the period of the fading (100's of ms) is large compared to the 40ms analysis window of the ML code. Classical DSP would require a 1000-2000 ms interleaver (introducing a algorithmic delay of the same order) for similar robustness to fading on these channels.

5. Unlike regular PSK, we require magnitude estimation and correction, due to the limited dynamic range of ML systems and the wide dynamic range of radio signals.

6. The *Candidate 2* OFDM frame design [2] contains three latent vectors z, so introduces an algorithmic delay of 120ms, due to OFDM framing considerations. This is tolerable in a PTT radio system, but ideally should be reduced.

7. Substituting classical PSK digital symbols and measuring BER is a very useful way to test sync subsystems, and to allows us to verify E_b/N_0 using BER measurements during development.

8. Much of our development effort to date has concerned with the "is this too good to be true" question; to date significantly more effort was put into noise calibration, synchronisation algorithms and testing than the actual ML.

Further work:

1. Try a low dimension latent vector, e.g. $d = 40$, and see if similar speech quality can be obtained at 3dB higher E_b/N_0. This would result in lower sync losses and RF bandwidth for the same channel SNR or C/N_0, as the E_b/N_0 of the pilots would be increased. Does the encoder output still resemble BPSK, or is it training to a higher order constellation?

2. There is significant synchronisation loss from the classical DSP OFDM waveform and associated sync algorithms. Attempt to use the ML network

to perform frequency, phase and amplitude equalisation, with or without passing the pilots to the decoder. An initial attempt (model07) without pilots showed some ability to correct phase, frequency, and magnitude offsets, but resulted in some performance degradation. However this may be acceptable if comparable to the pilot based sync losses. Some pilot or unique word injection may still be required to perform coarse and fine timing estimation using classical DSP running at the sample rate.

3. We may be able to improve on algorithmic delay using ML sync techniques that are less dependant on traditional OFDM framing considerations.

4. Some form or interleaving, or a long time window for the ML network, may improve performance.

5. Work to improve the current classical DSP sync, e.g. from Figure 5 a feedback loop to track out frequency offsets is worth 1 dB.

6. Include PAPR optimisation and rate F_s multipath channels in the training.

7. To better model SSB, locate a better analog compressor, a 3rd party reference implementation in software form would be useful.

8. Definition of lower limit link closure for this use case, for example "CQ CQ, and callsign, enough to produce a QSO report".

9. For a practical implementation, coarse amplitude and timing need to be updated regularly. As we are testing short samples, a single block estimate is used at present.

4 Comparison with Other Speech Waveforms

We wish to compare the potential of our radio autoencoder with existing waveforms used for speech transmission over radio channels. This section assumes the *radae* system can send intelligible speech at an E_b/N_0 of -6dB, with a PAPR of 1dB, both results have been demonstrated in simulation (but not at the time of writing together over real world radio channels). We include a 3dB synchronisation overhead, anticipating modest improvements over the 4dB obtained with our proof of concept system described above.

We start with the assumption that we have a transmitter of C watts, and an AWGN channel with a spectral noise density of N_0 watts/Hz. As the speech waveforms being considered vary in bandwidth we will choose C/N_0 as the SNR metric.

The C/N_0 (in dBHz) at the demodulator input of a terrestrial radio receiver is given by:

$$\frac{C}{N_0} = P - PAPR - L_{path} - NF + 174 \tag{14}$$

where P is the maximum output power of the transmitter power amplifier, $PAPR$ is the Peak to Average Power Ratio of the waveform, L_{path} is the path loss, NF is the noise figure of the receiver. For example consider a 400 MHz FM hand held radio over a 1km urban (non line of site) path. The radio has a 1W (30 dBm) power output, $L_{path} = 120$ dB, with noise dominated by ambient EMI such that $NF = 10$ dB. $C/N_0 = 30 - 0 - 120 - 10 + 174 = 74$ dBHz, sufficient for good quality speech (Table 6).

Note that C/N_0 at the demodulator is a function of the waveform PAPR. With all other link properties (e.g. peak PA power, noise figure, path loss) being equal, a high PAPR reduces the C/N_0 available at the receiver. We effectively "back off" the transmitter power from the maximum P by the PAPR. We assume the PA is capable of sustaining P watts indefinitely, i.e. it is only the waveform choice that lowers the average power. As PAPR varies by waveform and has an impact on the C/N_0 available the receiver, it should be included in any metric for comparison of waveforms. We define P/N_0 as:

$$P/N_0 = C/N_0 + PAPR \tag{15}$$

A waveform that delivers intelligible speech at a low P/N_0 is the target. A low PAPR waveform has other desirable properties, such as greater PA efficiency, longer battery life, and the use of low cost semiconductors in the radio power amplifier electronics.

Waveform	Threshold
Single Sideband	0dB SNR in 3000Hz noise BW, 2400Hz audio bandwidth, Tx speech compressor with 6dB PAPR
Frequency Modulation	-120 dBm quoted for many NBFM radios, 54dB above -174dBm/Hz noise floor
FreeDV 700D	10% PER threshold at -2dB SNR in 3000Hz noise BW
Radio Autoencoder	Intelligible speech at $E_b/N_0 = -6$ dB, $R_b =$ 2000 symbols/s, 3dB sync overhead, a PAPR of 1dB

Table 3: Thresholds for speech link closure for each waveform. The link is considered closed when the speech is barely intelligible to a trained listener.

TODO: include 1st gen VHF/UHF digital voice - I think they go down to -123 dB (5%) BER, but speech quality is sub FM.

Waveform	Threshold C/N_0 calculations (dBHz)
Single Sideband	$0 + 10log_{10}(3000) = 35$
Frequency Modulation	$-120 + 174 = 54$
FreeDV 700D	$-2 + 10log_{10}(3000) = 33$
Radio Autoencoder	$-6 + 10log_{10}(2000) + 3 = 30$

Table 4: Threshold C/N_0 calculations.

Waveform	Abbr	RF BW	PAPR	C/N_0	P/N_0	Δ
Single Sideband	SSB	2400	6	35	41	-10
Frequency Modulation	NBFM	16000	0	54	54	-23
FreeDV 700D	700D	1100	4	33	37	-6
Radio Autoencoder	radAE	1400	1	30	31	0

Table 5: Comparison of link closure by waveform over AWGN channels. A lower P/N_0 is better.

Waveform	Audio BW	C/N_0	P/N_0	Δ
Radio Autoencoder	8000	36	37	0
Frequency Modulation	3000	64	64	-27
Single Sideband	3000	55	61	-24

Table 6: Comparison of good quality "arm chair copy" by waveform over AWGN channels. They are ranked in terms of maximum achievable speech quality. FreeDV 700D has been omitted because of its low speech quality even at high SNR. Even at 20dB SNR there is noticeable noise in received SSB, although DSP based noise reduction may help. Only the radio autoencoder delivers wideband (8000 Hz) speech.

5 CPU and Memory Requirements

Module	MMACS	Memory (kbyte)
Feature Extraction	-	-
RADAE Encoder	80	1000
OFDM Tx	-	-
Totals	80	1000

Table 7: Estimates of RADAE transmitter (Tx) CPU and Memory Requirements.

Module	MMACS	Memory (kbyte)
FARGAN vocoder	300/175	800
RADAE Decoder	80	1000
OFDM Rx	-	-
Totals	380/255	1800

Table 8: Estimates of RADAE receiver (Rx) CPU and Memory Requirements.

Tables 7 and 8 contain estimates of the CPU and memory resources. Typical PTT radio uses-case are half duplex, so the Tx and Rx would not be required to run at the same time. The receiver dominates, due to the FARGAN vocoder.

MMACS is the number 8-bit multiply and accumulates per second divided by 1E6. The memory is mainly ML weights which are read-only. However memory access needs to be fast to maintain the MMAC rate - each MMAC is assumed to include any required loads. A non-SIMD floating point implementation (where multiple and accumulate are separate operations) would take twice the number of operations when measured in MFLOPS, and four times the memory.

The FARGAN author recommends a 64-bit ARM with NEON extensions as a minimum machine (Cortex-M is likely too small). The current FARGAN vocoder requires 300 MMACs, a slightly lower quality version is available that uses 175 MMACS. The classical DSP Feature Extraction uses comparatively small amounts of CPU and memory, but currently requires floating point hardware.

The RADAE ML CPU hasn't been measured accurately but runs in real time in Python, and is less than the FARGAN decoder (i.e. the FARGAN vocoder dominates the CPU). For an initial estimate we have approximated the RADAE CPU as 25% of FARGAN. The RADAE encoder and decoder require approximately the same CPU, and similar memory (ML weights). It may be possible to reduce the number of weights and CPU, as the optimal size of the RADAE network has not been studied.

The classical DSP OFDM processing requires less resources but hasn't been accurately measured at this time. Comparable modems at similar symbol rates run on a fraction of Cortex-M micro-controllers. Unlike the ML modules, it currently requires floating point hardware. For a first approximation we assume OFDM resource requirements are small compared to the ML modules.

6 Training and Testing Low PAPR Models

6.1 Mixed Rate Model

We wish to train a model to minimise PAPR over multipath channels. Our previous models (e.g. *model05*) were trained in the frequency domain at rate R_s with the bottleneck applied to the magnitude of real valued symbols z_i. Due to the neat properties of OFDM, we could ignore issues like phase/amplitude equalisation and ISI during training, and just apply magnitude only fading to each PSK symbol. After training, classical DSP techniques were then wrapped around the core ML encoder to develop a practical, rate F_s digital voice system.

We reasoned that a time domain bottleneck applied to the magnitiude of the complex time domain signal would encourage the network to maximise the RMS power (and hence minimse the PAPR) given the channel noise and peak power constraint of the bottleneck.

To train a model to optimise PAPR we need to apply a bottleneck in the time domain, and simultaneously apply a multipath channel model. Applying the multipath model in the time domain introduces phase rotations and ISI, which would then require equalisation and removal inside the training loop. While possible this would require significant additional CPU and algorithmic complexity using classical DSP or ML based equalisation, and require training at the higher rate F_s sample rate ($F_s/R_s = 160$ in our use case).

An alternative, "mixed rate" training model was therefore developed. The transmit PSK symbols are IDFTed to the time domain, the bottleneck applied, then immediately DFTed back to the frequency domain. The multipath model is then applied on the frequency domain PSK symbols. Training using this model resulted in signals with a PAPR of around 0.5 dB and S/N performance close to the reference non-PAPR optimised 1-D bottleneck *model05*.

After training, pilot symbols and a cyclic prefix are concatentaed with the ML signal carrying the payload voice information. Classical PAPR compression techniques (gain and compression) are applied to reduce the pilot signal PAPR to around 3dB. As the pilot sequence is much shorter than the payload sequence in each "modem frame", the overall PAPR remains beneath 1dB.

6.2 Calibrated Noise

To train for low PAPR we constrain the magnitude of the complex rate F_s time domain transmit signal $|x(n)| \leq 1$. We wish to train for a range of SNRs so require an expression to compute σ for a given target E_b/N_0.

Consider a hybrid ML-DSP system where we constrain $|x(n)|$ in the time domain, but perform all other processing at rate R_s in the frequency domain using classical DSP OFDM techniques. After the bottleneck is applied to $x(n)$, we DFT to transform back to the frequency domain, apply a magnitude only multipath model, than add AWGN noise to simulate the received frequency domain symbols $\hat{q}$.

Given this model, we wish to inject calibrated AWGN noise. Consider one time domain OFDM carrier $x(n)$ with magnitude B. We take the length M DFT to determine the magnitude A_q of the corresponding frequency domain PSK symbol:

$$\begin{aligned} x(n) &= Be^{j(2\pi nk/M+\theta)} \\ X(k) &= \sum_{n=0}^{M-1} x(n)e^{-j(2\pi nk/M)} \\ &= BMe^{j\theta} \\ |X(k)| &= A_q = BM \end{aligned} \tag{16}$$

We have N_c carriers, and the bottleneck constrains the total time domain power of all carriers to be approximately 1:

$$\begin{aligned} N_c B^2 &= 1 \\ B &= \frac{1}{\sqrt{N_c}} \end{aligned} \tag{17}$$

Using (4) we can find an expression for σ given E_b/N_0:

$$\begin{aligned}\sigma &= \frac{A_b}{\sqrt{E_b/N_0}} \\ \sigma &= \frac{A_q}{\sqrt{2E_b/N_0}} \\ \sigma &= \frac{M}{\sqrt{2N_cE_b/N_0}}\end{aligned} \tag{18}$$

The factor of $\sqrt{2}$ on the denominator accounts for A_q being the amplitude of a PSK symbol, whereas (4) is configured for BPSK symbols. For example if $A_b = 1$, $A_q = \sqrt{2}$.

These expressions were used as a baseline for applying noise to PAPR optimised models. It was found the measured Eq/N_0 were around 2.5dB higher than the targets. Our theory is that in practice the symbol magnitudes A_q vary across carriers and time and are best described as a model-dependant distribution (probably Rayleigh) rather than a constant. Nevertheless the expressions in this section are close enough for practical training over a range of SNRs, and for inference where we measure and report the actual SNR.

6.3 Simulated Results

The *evaluate.sh* script was used to genereate samples for informal listening, and to compare to SSB (Table 9). E_b/N_0 controls the channel C/No but does not map exactly to it due to model-model variations. In practice we measure C/No and PAPR for each run of the RADAE simulation, then determine the amount of noise to inject for the SSB sample to obtain the target P/N_0. When applying a multipath model, the transmit power is normalised over the simulation run such that the average carrier power C is the same before and after the multipath model is applied.

The earlier, *model05* RADAE waveform has a PAPR of 9-10dB. Based on our experience of classical DSP OFDM waveforms we reason that by applying classical DSP PAPR reduction techniques this could be reduced to 5dB, with a 1dB loss in performance, leading to an effective PAPR of 6dB, similar to the SSB signal. We have not tested this configuration, but reason that *model17* will have a 6dB advantage over *model05* for the same transmitter peak power.

Waveform	Channel	E_b/N_0	PAPR	C/N_0	P/N_0	SNR
Radio Autoencoder	AWGN	16	0.80	48.21	49.01	13.44
Single Sideband	AWGN	16	6.66	42.35	49.91	7.58
Radio Autoencoder	MPP	16	0.80	48.21	49.01	13.44
Single Sideband	MPP	16	6.66	42.35	49.91	7.58

Table 9: Comparison of PAPR optimised RADAE *model17* to SSB. The low PAPR RADAE waveform shows an improvement of around 6dB in receiver SNR over SSB, although the relationship between speech quality and SNR is of course different for both receivers.

7 Glossary

Symbol	Explanation	Units
B	noise or signal bandwidth	Hz
C	RMS carrier power $C = S$ for this study	
C/N_0	Carrier power/spectral noise density	
d	dimension of latent vector $\mathbf{z}$	
E_b/N_0	energy per BPSK symbol on spectral noise density	
E_q/N_0	energy per QPSK symbol on spectral noise density	
N	total noise power	Watts
N_c	Number of carriers	
N_0	Noise power in 1 Hz of bandwidth	
P/N_0	Peak power/spectral noise density	
$\mathbf{q}$	vector of QPSK symbols	
q_i	single QPSK symbol, element of $\mathbf{q}$	
R_b	BPSK symbol rate	symbols/second
R_q	QPSK symbol rate	symbols/second
R_s	OFDM per carrier QPSK symbol rate	symbols/second
R_z	latent vector update rate	Hz
SNR	signal to noise Ratio	
S	total signal (carrier) power	Watts
T_b	BPSK symbol period	seconds
T_q	QPSK symbol period	seconds
T_s	OFDM per carrier QPSK symbol period	seconds
T_z	time between latent vector updates	seconds
r_i	noise sample	
$\mathbf{z}$	Autoencoder output latent vector	
z_i	single latent vector element of $\mathbf{z}$, a BPSK symbol	

Table 10: Glossary of Symbols

8 References

[1] Low SNR FreeDV Modes. https://github.com/drowe67/misc/blob/master/freedv_low/freedv_low.pdf.

[2] David Rowe. FreeDV-032 Radio Autoencoder Waveform Design (spreadsheet).

[3] Jean-Marc Valin, Jan Büthe, and Ahmed Mustafa. Low-bitrate redundancy coding of speech using a rate-distortion-optimized variational autoencoder, 2023.

Figure 5: OFDM pilot based synchronisation algorithm performance, tested by measuring the BER obtained using discrete PSK symbols on an AWGN channel. With ideal sync, the autoencoder produces intelligible speech at $E_b/N_0 = -6$ dB which corresponds to BER=0.24. Several algorithms, combined with gain and frequency offsets were simulated.

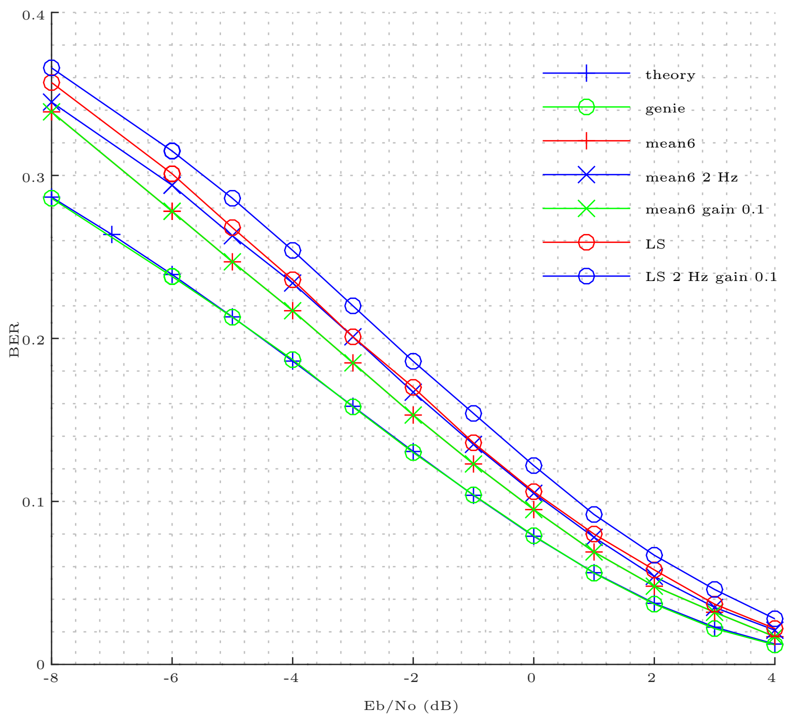
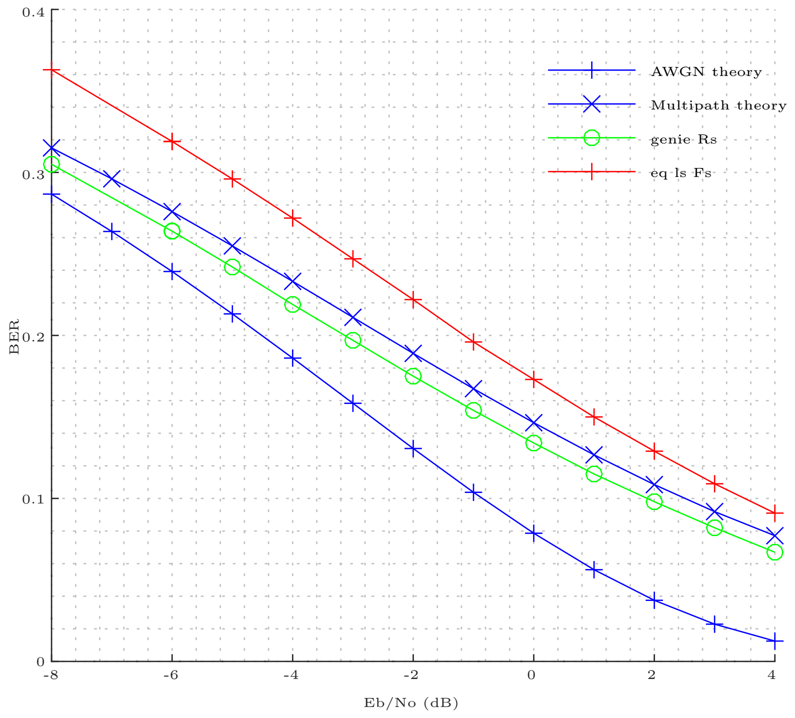


Figure 6: OFDM pilot based synchronisation algorithm performance, tested by measuring the BER obtained using discrete PSK symbols on an multipath (MPP) channel. The "genie" curve is slightly better than theory, probably due to an insufficiently long run. At the BER=0.24 threshold, the *LS* algorithm has a 2dB loss compared to the "genie" curve


AWGN theory
Multipath theory
genie Rs
eq ls Fs
BER
Eb/No (dB)