# Unwrapping Codec 2 Data Manifolds

December 31, 2023

> Uncrumpling paper balls is what machine learning is all about: finding neat representations for complex, highly folded data manifolds
>
> Francois Chollet [3]

## 1 Introduction

Speech coding can be defined as the art of representing speech using a small number of parameters that can be efficiently sent over a channel.

Codec 2 700C resamples the pitch dependent, time varying $L = F_s/(2F_0), L = 20..80$ harmonic magnitude samples $20log_{10}(A_m), m = 1..L$ to a set of fixed length $K = 20$ samples $B_k, k = 1..K$. These are then vector quantised $\hat{B_m} = Q(B_m)$. Harmonic phases are synthesised using a minimum phase model. The recovered phase spectra and hence energy distribution across the pitch cycle is dependent on the magnitude spectra available at the receiver.

There is some evidence that the distribution of energy over a pitch cycle is important for speech perception [2]. To adequately represent male speech, the formant energy must be spread in time. As Codec 2 synthesises phase using a minimum phase model - this implies narrow formant bandwidths must be maintained [1]. Narrow formant bandwidths lead to long decay times for filters "ringing" with formant energy. When low resolution spectral sampling is employed (e.g. smoothed vector quantised spectra or low bit rate LSP quantisation of LPC models), male speech becomes buzzy, muffled and less intelligible. Speech with a short pitch period (e.g. female) has less time to decay between pitch cycles so is less sensitive to formant bandwidths, the short pitch period naturally leads to a more uniform distribution of energy over time.

At the Codec 2 receiver, a post filter $P()$ is applied which improves the perceptual quality, in particular for low pitched (male) speakers. The post

[1] Other ways of spreading time domain energy are possible, for example generating phases through heuristics, non-minimum phase models, or time domain compression [2]

filter is a form of ad-hoc non-linear filtering, using experimentally derived constants. It raises formants and lowers anti-formants, effectively reducing formant bandwidth and more widely distributing energy in time (each formant "rings" longer).

Neural codecs have shown high quality speech can be synthesised using vectors of sparsely sampled frequency domain samples (MFCCs) [4]. Applying linear transformations (such as IDCT and interpolation) to recover $\{A_m\}$ from MFCCs results in poor quality synthesised speech in sinusoidal codecs. This implies neural codecs are employing a non-linear transformation, made possible by modern deep learning techniques.

From an information theory perspective there is no reason to believe there is more information in speech from low pitch speakers than high pitched speakers. Therefore with an appropriate transformation we should be able to synthesise equivalent quality speech regardless of pitch.

This document describes two experiments to determine if narrow bandwidth formants can be preserved for low pitch speakers using non-linear transformations, resulting in reasonable quality speech when synthesised using Codec 2 (without the use of an ad-hoc post filter).

## 1.1 Vector Quantisation

Vector quantisers can take advantage of linear and non-linear dependencies to reduce bit rate [5]. They can efficiently exploit information that is uncorrelated in a linear sense (no linear transformation exists), but is statistically dependant.

The 700C resampling to a fixed length $K << L_{max}$ is a linear transformation aimed at fixing the dimension of the data; reducing the distortion with multi-stage VQ where residuals tend towards independent Gaussians; and reducing VQ storage requirements. However the resampling is a filtering/smoothing/aliasing operation, so some information is lost, for example narrow bandwidth formants cannot be recovered by a linear transformation.

In this experiment we will upsample the variable rate $L$ vectors to a fixed length $K = 80$, then vector quantise. In this way no information is lost from the source vectors. The aim is to utilise the non-linear dependency matching properties of VQ to preserve high quality speech, while still maintaining a similar bit rate to $K = 20$ quantisation. We will ignore storage concerns for this experiment.

## 1.2 Machine Learning

In this approach we will attempt to take a $K = 20$ vector and resample it to $K = 80$ using a small neural network. We hope the network will discover any non-linear dependencies, and produce narrow bandwidth formants and (when synthesised using Codec 2) reasonable quality speech for low pitched speakers.

We will include pitch as a feature (perhaps via an embedding network), arguing that the formant bandwidth is a function of pitch ($F_0$). This is information the VQ in Section 1.1 does not consider, which implies better expected results

from the ML approach. Using a fixed length target vector simplifies any issues around variable vector length. This is similar to the decoder side of an autoencoder network. Autoencoder designs could therefore be used as candidate architectures.

We argue that essentially the same information (in the form of MFCCs) is used for high quality speech synthesis with neural vocoders, therefore these networks must be performing a similar non-linear mapping of coarsely sampled, smoothed spectral information to speech spectra that includes narrow bandwidth formants. A counter argument is a simple neural network may not be capable of representing the non-linear function that maps between the two vectors. However we are not trying to synthesise speech here - just the speech spectral envelope. Other differences: the network proposed here outputs linear values using regression, more sophisticated neural vocoders employ sophisticated conditional probability models and output a PDF; this network considers just one frame, rather than utilising an autoregressive design that considers many past samples.

### 1.3 Evaluation Plan

Evaluation methods considered:

1. Informal listening tests and ranking across a small number of samples.

2. Evaluation of the results using objective measures such as Spectral Distortion (SD), however the relationship to subjective quality may be complex. For a given bit rate SD may be larger at $K = 80$ but the synthesised speech may sound better.

3. Another candidate objective measure is Peak Average Power Ratio (PAPR). This tends to be higher when the formant bandwidths are not preserved.

4. Visual inspection of speech spectra and waveforms of speech synthesised from both methods (prior to post filtering) would indicate if narrow formants have been preserved for males.

Success would be indicated by evidence of formant bandwidth being preserved, and higher quality speech from male speakers compared to linear $K = 20$ approaches. Speech should be synthesised using phases recovered from the output magnitude spectra using the minimum phase model - the degradation in quality is less obvious when original harmonic phases are used.

For controls we could use:

1. Speech resampled through the $K = 20$ "bottleneck" (without postfiltering).

2. Speech VQed at $K = 20$ and $K = 80$.

3. Codec 2 3200 - this uses finely quantised LSPs that preserve formant shapes (and an ad-hoc post filter).

## 2 $K = 80$ VQ Results

$K = 20$ and $K = 80$ 2x12 bit (24 bit total VQs) were trained on 114,000 vectors using the LBG algorithm. A single male and female sample were processed under a variety of conditions (see `README`). The $K = 80$ male sample sounded better than the $K = 20$ sample, however there is still significant distortion. A significant artefact is a clicky/buzzy attribute after VQ, which is much more obvious at $K = 20$. The female sample was approximately the same for both $K = 20$ and $K = 80$.

Figure 1 is a plot of a single frame from the male sample. The SD was significantly higher for $K = 80$ than $K = 20$, the reverse of the subjective results. Note the narrow bandwidth of the formants - in this frame F1 is defined by a single harmonic around 10dB higher than adjacent harmonics. While $K = 80$ preserves the spectral detail well, the VQ is very noisy due to the difficulty with large dimension, multi-stage VQ. The largest errors (black) are around the formants - precisely the opposite of what we would like.

Visual inspection of the male time domain synthesised waveforms (Figure 2) demonstrates the energy distribution issue. The process of dimension reduction and quantisation leads to spectral smoothing; narrow formants become wider. A wider filter in the frequency domain leads to more concentration of energy in time (faster decay of the time domain envelope), which we believe is the source of the unpleasant buzzy artefact. The human ear does not like clicks. The objective CPAPR measure does not rank the samples correctly, it should be lower when the time domain energy is spread more uniformly over time (e.g. lower for plots 1 and 3 that use the original amplitudes and phases).

These experiments suggest the $K = 80$ VQ is better at preserving spectral detail such as narrow bandwidth formants and provides a better time domain energy distribution than $K = 20$. However large dimension multi-stage VQ with acceptable quality is difficult, as the residual error vectors from each stage tend towards independent Gaussians.

The problem remains - higher pitched speakers sound acceptable at low spectral resolution such as $K = 20$, so it should be possible to encode low pitch speech with acceptable quality at the same dimension.

## 3 ML Design

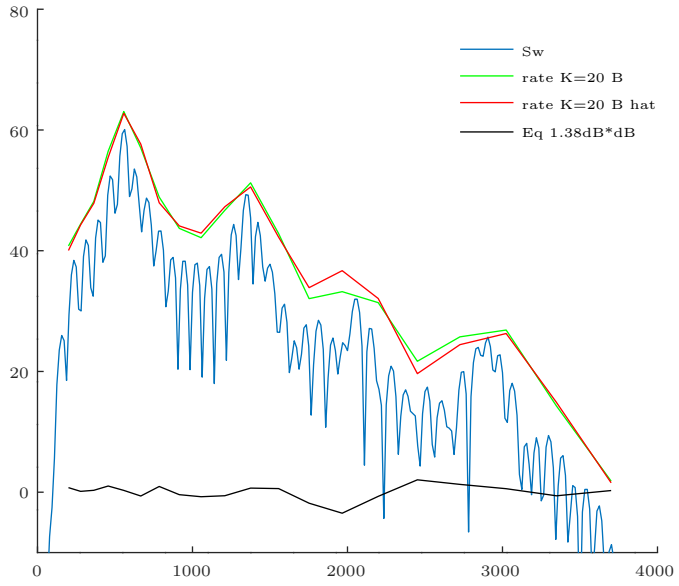
In order to develop a ML system, we need to develop network model and loss functions. To gain some insight into the energy distribution of speech across a pitch period a damped second order system was studied (Appendix A).

Our problem is more complex that the second order example:

1. Rather that a transfer function we have a smoothed (wide bandwidth formant) version of the amplitude spectrum $B_k, k = 1..K$ and pitch $F_0$ available.

2. From $\{B_k\}$ we wish to obtain a set of amplitude and phase spectra $\{A_m\}, \{\theta_m\}, m =$

Figure 1: Voiced frame 165 frequency domain from male speaker for $K = 20$ and $K = 80$, red is 24 bits/frame VQ. Objective VQ distortion is higher at $K = 80$ however formants are sharper and formants better preserved. F3 is poorly represented in both cases. Note the formants F1-F3 are defined by single harmonics, indicating narrow bandwidth.



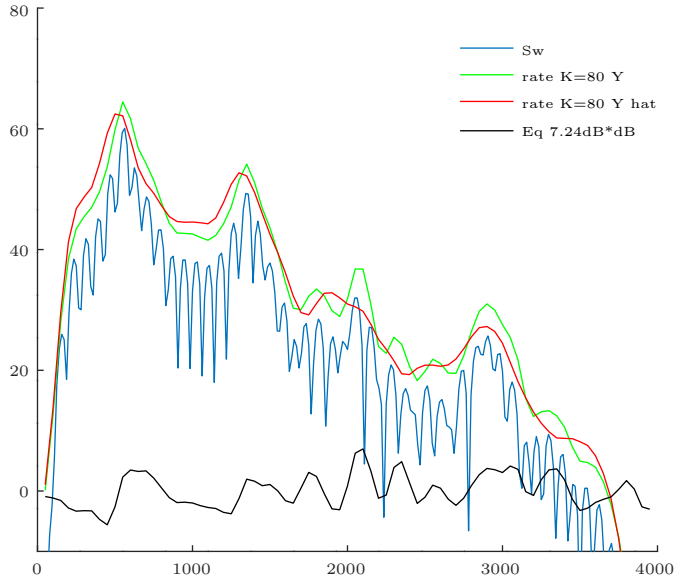
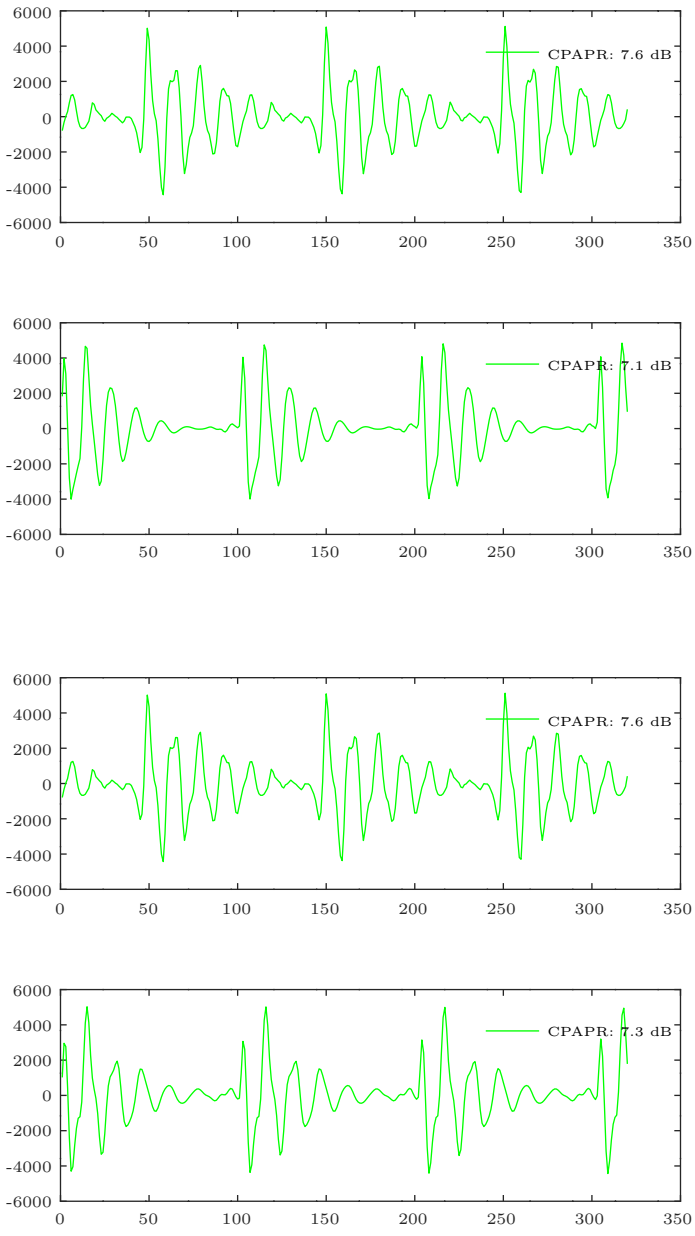



Figure 2: Voiced frame 165 time domain synthesised speech from male speaker. Plots 1 and 3 are the speech synthesised with original harmonic magnitudes and phases, presented as a control. Plot 2 is speech synthesised from the $K = 20$ model, plot 4 $K = 80$. The difference in energy distribution over the pitch cycle is clear.



1..$L$ such that the energy distribution across the pitch period is similar to the input speech.

However the second order example gives us some hints on what we would expect to see from a viable ML system: a narrowing of formants for males, an increase in phase shift across each formant, and energy distributed across the pitch cycle.

### 3.1 ML Design candidate

Let the $K = 20$ $N_b$ filtered vector $\mathbf{b} = \left[B_1, B_2, \ldots B_K\right]$, and the $K = 80$ unfiltered, oversampled vector $\mathbf{y} = \left[Y_1, Y_2, \ldots Y_K\right]$, where both are resampled versions of $log_{10}A_m$ scaled to have unit (linear) energy. The network and loss functions are:

$$\begin{aligned} \hat{\mathbf{y}} &= F(\mathbf{b}, \omega_0) \\ L(\mathbf{y}, \hat{\mathbf{y}}) &= \sum_{k=1}^{K=80} |(10^{Y_k} - 10^{\hat{Y_k}})W(k)|^2 \\ &= \sum_{k=1}^{K=80} |(10^{Y_k} - 10^{\hat{Y_k}})10^{-Y_k(1-\gamma)}|^2 \end{aligned} \tag{1}$$

where $W(k) = 10^{-Y_k\gamma}$ is a weighting filter that de-emphasises spectral peaks and emphasises spectral valleys, with $0 < \gamma < 1$ chosen by experiment ($\gamma = 1$ is unweighted). The loss function compares the spectrally weighted linear energy. This candidate has the following features:

1. Experience has shown us that if formant bandwidths are preserved, the $K = 80$ spectrum can be used to generate speech with a good energy distribution using phase obtained by a minimum phase (Hilbert) transform.

2. Operating in the magnitude domain with fixed $K = 80$ avoids the problems around dealing with phase and time varying number of parameters in the network and loss function.

3. Features in the network $F()$ are unit energy and in the $log_{10}()$ domain to reduce dynamic range.

4. A loss function in the linear (rather than $log_{10}()$) domain will emphasise spectral match (and hence formant bandwidth preservation) in the high energy formants.

5. Weighting means it will pay some attention to interformant regions and less to very low energy regions at low and high frequencies (due to input filtering of the source speech).

6. Using pre-emphasised speech to extract $\{A_m\}$ will also help formant matching and energy distribution of higher frequency formants (assuming the input speech has flat filtering).

7. For synthesis we could choose to use the amplitudes as well as the phases derived from $\hat{\mathbf{y}}$, or just the phases.

8. The weighting filter means energies will be matched in the weighted domain. This may not be the same as matching energy in the synthesised speech. A loss term could be introduced to ensure an energy match is mantained, despite modification of the spectral shape.

Figure 3: ML Experiment Block Diagram. All operations (including $F$) operate at the sinusoidal vocoder 10ms frame rate.

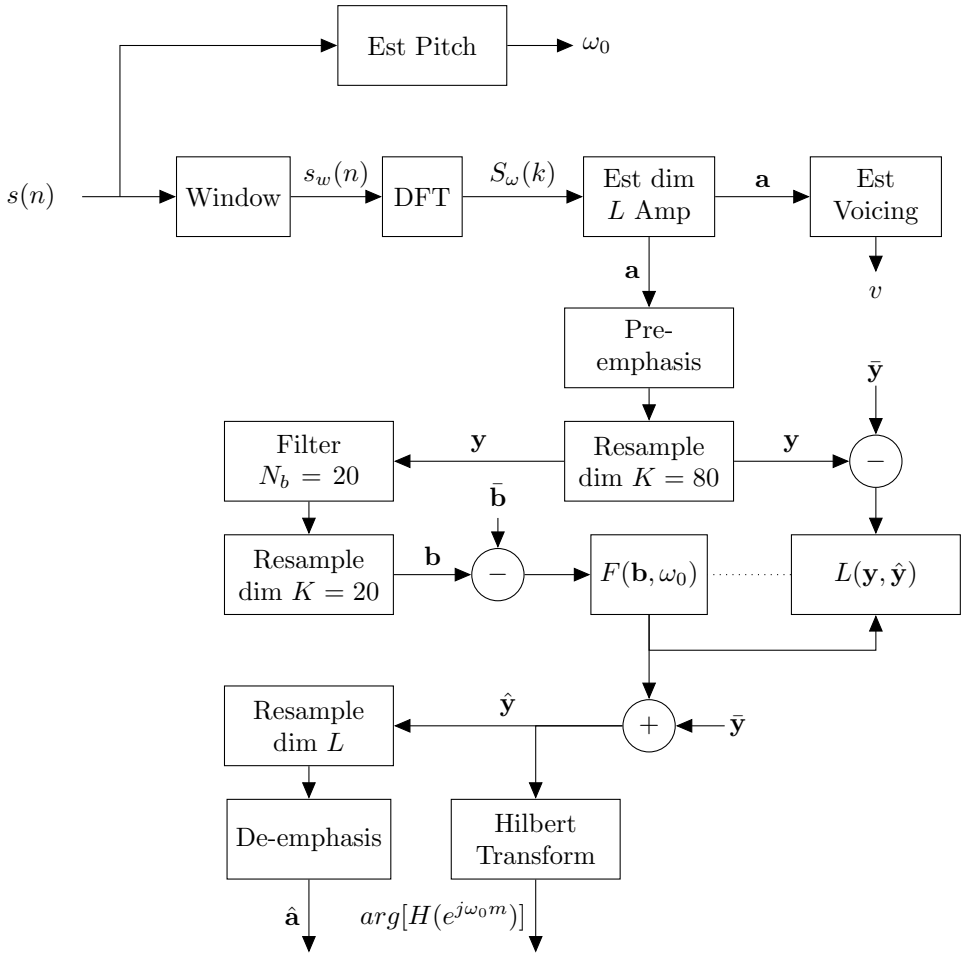



Figure 3 is a block diagram of the ML experiment signal processing. The loss function $L(\mathbf{y}, \hat{\mathbf{y}})$ only used during training. The algorithm outputs the

dimension $L$ harmonic amplitudes $\mathbf{a}$ and phases $arg[H(e^{j\omega_0 m})]$ for sinusoidal synthesis [1]. Pre-emphasis and de-emphasis is applied in the frequency domain. The ML network $F$ and loss function $L$ operates on normalised energy vectors as we are interested in matching spectral shape, not gain (see also manifold.py comments and implementation). It can be viewed as an autoencoder with the bottleneck being the smoothed dimension $K = 20$ vector $\mathbf{b}$. Only the decoder side of the autoencoder is implemented with a nueral network - the encoding is performed using traditional, linear DSP.

## 3.2 ML Results

Table 3.2 describes the simple neural network that implements $F$. The 22 input features are the vector $\mathbf{b}$, $log_{10}(\omega_0)$, and the binary voicing decision $v$. After some experimentation, training was performed on voiced vectors ($v = 1$) above an energy threshold to remove silence frames. It was found that removing the pitch feature $log_{10}(\omega_0)$ made little difference to the final loss, suggesting $\mathbf{b}$ alone is sufficient.

| Layer | Input Features | Output Features | Output Activation |
|---|---|---|---|
| 1 | 22 | 512 | ReLU |
| 2 | 512 | 512 | ReLU |
| 3 | 512 | 79 | ReLU |

Table 1: The function $F$ consists of three fully connected layers, the size and topology was arrived at by experiment. It has around 300,000 floating point weights.

Figures 4 and 5 show the ML system operating in the time and frequency domain. Table 3.2 presents an objective results for 4 samples outside of the training data. Spectral distortion between $\mathbf{y}$ and $\hat{\mathbf{y}}$ is defined as:

$$E_q = \frac{1}{K} \sum_{k=0}^{K-1} |Y_k - \hat{Y}_k|^2 \tag{2}$$

The four samples were subjectively evaluated using headphones and laptop speakers. Samples included speech synthesised with;

1. original amplitude and phases, the best case Codec 2 speech.
2. phase0 model, $N_b = 20$ filtering, equivalent to speech synthesised directly from the $\mathbf{b}$ vectors.
3. phase0 model, $\mathbf{y}$, equivalent to ideal output from the ML network.
4. phase0 model, $\hat{\mathbf{y}}$, output from the ML network.
5. the fully quantised Codec 2 3200 mode.

All male samples synthesised with the ML network sounded better than the $N_b = 20$ and Codec 2 3200 samples, and through laptop speakers they were hard to distinguish from the original amplitudes/phases. The female $N_b = 20$ and $\hat{\mathbf{y}}$ sounded the same. A slight amount of buzziness could be heard when comparing the output from $\mathbf{y}$ and $\hat{\mathbf{y}}$ through headphones, but not the laptop speakers. The subjective results support the theory that $\hat{\mathbf{y}}$ can be expressed as a (probably non-linear) function of $\mathbf{b}$, and that the information of formant bandwidths is somehow encoded in $\mathbf{b}$.

| Sample | Gender | $E_q\ dB^2$ | $\sqrt{E_q}\ dB$ |
|---|---|---|---|
| big_dog | M | 1.38 | 1.18 |
| two_lines | F | 1.02 | 1.01 |
| hts1a | M | 0.40 | 0.64 |
| kristoff | M | 0.37 | 0.61 |

Table 2: Objective results with one female and 3 low pitched male samples. The low $Eq$ values suggest the formant bandwidth information in $\mathbf{y}$ is represented in the lower dimension vector $\mathbf{b}$, despite the smoothing and undersampling. All samples were outside of the training data, however the first two were part of the same database and shared the same recording conditions as the training data. These results demonstrate successful inference outside of the training data and with varying recording (source filtering) conditions.

# 4 Conclusions

This work ties together modern neural vocoders and earlier sinusoidal vocoders, by showing how low pitched speech with a suitable energy distribution can be extracted from smoothed, low dimension feature vectors. The ML experiment can be considered an autoencoder with the smoothed, low dimension vector $\mathbf{b}$ as the bottleneck. The output is a high resolution spectral envelope, that is used to generate a suitable phase spectra via a Hilbert Transform. Inference has been demonstrated working outside of the training database. It was found that the inclusion of pitch feature wasn't significant, suggesting the formant bandwidths are encoded by the smoothed spectral information alone. Extending this framework to include quantisation may lead to low complexity communications quality speech coding using small neural networks and linear DSP.

Further work:

1. The weighted loss function approach may also be useful for quantisation, e.g. a VQ search. It may be possible to use smaller dimension bottlenecks than $K = 20$.

2. It may be possible to integrate the linear resampling steps with NN, however their complexity is low, and NNs are often inefficient when implementing well understood DSP operations.

Figure 4: Voiced frame 165 frequency domain from male speaker for ML experiment. The neural network output $\hat{\mathbf{y}}$ has significantly narrower formants than the smoothed, dimension $K = 20$ $\mathbf{b}$.

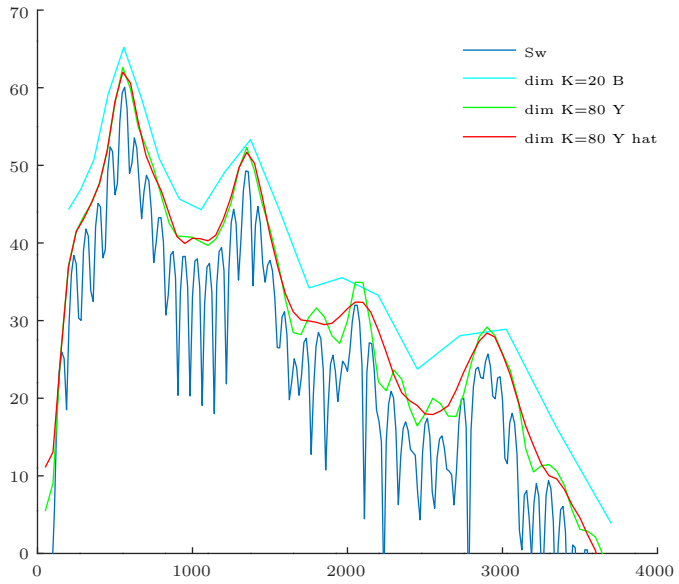
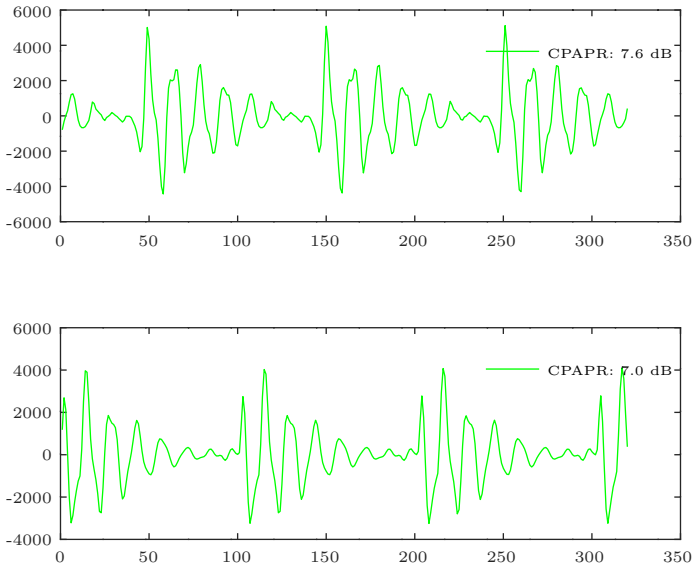


Figure 5: Voiced frame 165 time domain synthesised speech from male speaker. The upper plot is speech synthesised with original harmonic magnitudes and phases, presented as a control. The lower plot is speech synthesised from the ML model, and has a time domain energy distribution close to the original.



3. It may be possible to use MFCC feature vectors as the input, to avoid complexity and occasional errors around pitch dependant $\{A_m\}$ estimation. However there may be some benefit to the deconvolution (separation of excitation spectra from spectral envelope) inherent in the $A_m$ extraction process.

4. The weights of the network have not been studied, and are likely sparse, with significant weights clustered around frequency regions. It is likely the network can be reduced in size to make implementation possible on a micro-controller.

5. Other ML models are possible, for example a loss function that includes the time domain envelope, with time domain synthesis in the training loop.

6. A better objective measure of energy distribution would be useful, Complex Peak Average Power Ratio (CPAPR) is not effective. This should probably be comprised of several bandpass measures, to avoid F1 dominating.

# A Second Order Systems

Consider a damped 2nd order system approximating a vocal tract formant resonance at $\omega_f$ radians:

$$x(n) = e^{-\alpha n} e^{j\omega_f n} \quad n \geq 0 \tag{3}$$

As the sequence $x(n)$ is defined as analytical (complex valued), the magnitude as a function of time is simply:

$$|x(n)| = e^{-\alpha n} \tag{4}$$

Hence $\alpha$ is a time constant defining the distribution of energy across the pitch period. The Discrete Time Continuous Frequency Fourier Transform is:

$$\begin{aligned} X(\omega) &= \sum_{n=0}^{n=M} e^{-\alpha n} e^{j\omega_f n} e^{-j\omega n} \\ &= \sum_{n=0}^{n=M} e^{n(-\alpha + j\omega_f - j\omega)} \end{aligned} \tag{5}$$

Using the identity:

$$\sum_{n=0}^{M-1} r^n = \begin{cases} \frac{1-r^M}{1-r} & r \neq 1 \\ M & r = 1 \end{cases} \tag{6}$$

$$\begin{aligned} X(\omega) &= \frac{1 - e^{M(-\alpha + j\omega_f - j\omega)}}{1 - e^{-\alpha - j\omega_f + j\omega}} \\ &= \frac{1 - e^{-M\alpha} e^{M(j\omega_f - j\omega)}}{1 - e^{-\alpha} e^{j(\omega_f - \omega)}} \\ &= \frac{1}{1 - e^{-\alpha} e^{j(\omega_f - \omega)}} \end{aligned} \tag{7}$$

for large $M$. We can sample at $X(\omega_0 m)$ to obtain the phase and amplitude of each harmonic. This shows that for a simple system, we can obtain the harmonic phases as a function of the energy distribution defined by $\alpha$. The peak amplitude at $\omega = \omega_f$ is:

$$H(\omega_f) = \frac{1}{1 - e^{-\alpha}} \tag{8}$$

The bandwidth can be found by solving for the half power frequency:

$$\begin{aligned} |H(\omega)|^2 &= \frac{1}{2}|H(\omega_f)|^2 \\ \frac{1}{|1 - e^{-\alpha} e^{j(\omega_f - \omega)}|^2} &= \frac{1}{2|1 - e^{-\alpha}|^2} \\ \frac{1}{1 - 2e^{-\alpha} cos(\omega_f - \omega) + e^{-2\alpha}} &= \frac{1}{2(1 - 2e^{-\alpha} + e^{-2\alpha})} \\ cos(\omega_f - \omega) &= 2 - \frac{e^{\alpha}}{2} - \frac{e^{-\alpha}}{2} \end{aligned} \tag{9}$$

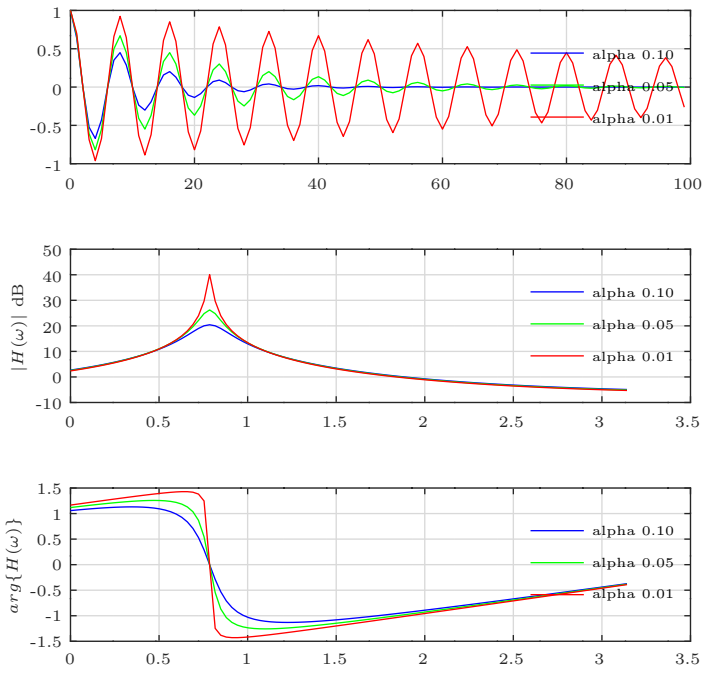
The bandwidth $B = 2(\omega_f - \omega)$. Using some approximations for $cos(a)$ when $a$ is small, and $e^x$ when $x$ is small:

$$\begin{aligned} cos(x) &\approx 1 - x^2/2 \\ e^x &= \sum_{k=0}^{\infty} \frac{x^k}{k!} \\ e^x + e^{-x} &= 2 + x^2 + \frac{x^4}{3} + ... \\ &\approx 1 + x^2 \\ B &\approx 2\sqrt{e^{\alpha} + e^{-\alpha} - 2} \\ &\approx 2\alpha \end{aligned} \tag{10}$$

To determine the phase, lets evaluate (7) at one of the half power frequencies:

$$\begin{aligned} X(\alpha) &= \frac{1}{1 - e^{-\alpha} e^{j(\omega_f - \alpha)}} \left( \frac{1 - e^{-\alpha} e^{-j(\omega_f - \alpha)}}{1 - e^{-\alpha} e^{-j(\omega_f - \alpha)}} \right) \\ &= \frac{1 - e^{-\alpha} e^{-j(\omega_f - \alpha)}}{1 - 2e^{-\alpha} cos(\omega_f - \alpha) + e^{-2\alpha}} \end{aligned} \tag{11}$$

Figure 6: Second order simulation for various values of $\alpha$. Top is $Re\{x(n)\}$ time domain for 100 samples, e.g. one pitch period with $F_s = 8000$ Hz and $F_0 = 80$ Hz.




To determine the phase at the upper half power frequency:

$$\begin{aligned} arg\{X(\omega_f + \alpha)\} &= -atan\left[\frac{1}{X(\omega_f + \alpha)}\right] \\ &= -atan\left[\frac{-e^{-\alpha}sin(\alpha)}{1 - e^{-\alpha}cos(\alpha)}\right] \end{aligned} \tag{12}$$

Figure 6 illustrates the second order system for various values of $\alpha$. We can observe that if the energy in the formant is declines slowly over time (small $\alpha$), the magnitude of the peak is high, the bandwidth narrow, and the rate of change of phase wrt frequency is greater.

# B ML Design Notes

Notes:

1. A first choice is $K = 80$ vectors as a target, and MSE between log spectra as the cost function. However like VQ, this will try to minimise the error across the entire spectra, and not focus on the critical formant bandwidths.

2. The VQ search uses MSE between log spectra, and often removes/rounds off spectral peaks and broadens formant bandwidths.

3. VQ training using kmeans also encourages averaging which de-emphasises vectors with tight spectral peaks (VQ doesn't consider $F_0$).

4. Matching the time domain waveform (or equivalently frequency domain amplitude and phase) is not critical, it's the time domain envelope that counts.

5. Lets separate the problem of quantisation from energy distribution. Use a low dimension, unquantised $K = 20$ or MFCC vector as input, and focus on reasonable quality low pitched speech as the output.

6. We are not tied to the current Codec 2 algorithm of converting magnitude spectra to phase spectra via a minimum phase Hilbert Transform. We can generate phase spectra using other means.

7. The cost function should be weighted against energy being confined to short time durations (clicks) in the synthesised speech. This implies we need a function $X(\{A_m\}, \{\theta_m\})$ that measures energy distribution.

8. PAPR does not appear to be satisfactory (confirm/explore why).

9. We would like formant bandwidth to be a function of $F_0$, as it matters for low pitched speakers (e.g. males) than females. Having said that - females undersample the spectra which just one harmonic per formant so formant bandwidth is perhaps irrelevant to females. We do want to preserve the power of the harmonic(s) in that formant (something the current ad-hoc post filter doesn't explicitly do). A narrow bandwidth formant may work for both male and female speech.

10. Smooth post filtering across frames is important, we don't want abrupt changes in formant energy or discontinuous phases.

11. For UV speech mistakenly classified at V, we should also have energy distributed - we should never synthesise clicks. UV speech tends to have broad, flat spectra with no sharp peaks, so if synthesised as V will have noticeable clicks.

12. When formants are broadened, multiple harmonics of similar level are synthesised, creating a "two tone" time domain waveform envelope, rather than a damped exponential (bandpass figures).

13. Idea: can we "fit" damped exponentials to spectra, like VQ-VAE fits Gaussians?

Requirements for cost function:

1. Models energy distribution of real world speech, and way ear integrates energy.
2. Good time domain distribution in each formant, this may not be obvious in non-preemph speech due to spectral tilt.
3. Formant energy should be the same after processing.
4. Formant energy should be smooth between frames. Avoid amplitude modulation at the frame rate.
5. Smooth transition between frames, don't break phase track.
6. Deals with time varying number of harmonics, e.g. via interpolation of sparse functions
7. Stretch goal: For UV speech mistakenly classifed as V, doesn't generate clicks.
8. Low implemenation complexity in ML, to minimise chances of implemenation mistakes with current ML experience.
9. Operate in the pre-emphasised domain to ensure the energy distribution of HF formants is treated equally with LF formants. Removes spectral tilt.

Candidates techniques:

1. Derive a loss function based on comparing 2nd order exponential $\alpha$.
2. Match spectra, especially around formants, and use a HT min-phase to get phases. $N_b$ filtering broadens peaks and chops off tops, several harmonics enter formant filter and $\alpha$ is reduced.
3. Modify phases, not amplitude spectra. Experiments show original phase and rate $K$ spectra sounds OK. Ear is not that sensitive to frequency as long as energy distribution is OK. Modifying spectra can lead to other artefacts as energy bounces around. OTOH, it might help intelligability.
4. Use a CELP style weighting filter. This will combine energy match of highest amplitude spectra, with some match of lower energy, and will ignore low energy HP/LP sections that have been filtered out.
5. Ratio of minima/maxima of magnitude envelope, which can be used to obtain $\alpha$. Someone must have worked out expressions for these? Can we solve anlaytically? Given Fourier series define by $\{A_m\}, \{\theta_m\}, m = 1..L$, can we obtain an expression for min/max or $\alpha$.
6. Can we derive an expression for mag envelope as a function of phases?

# C Notes and Further Work

Notes:

1. Do we need an embedding layer for pitch? Research embedding networks.
2. Should we test with some contrived data? Introduce expected non-linearity and see if network can train as a check.
3. Just include voiced frames? This seems like a good idea. For UV frames pitch may not be meaningful, and may cause random noise. Set UV frames to a constant $F_0$ (like 0) so at least the network "knows".
4. Make sure we use energy normalisation [2] when resampling to rate $K$.
5. Used mean removed vectors, just consider $200 < f < 3600$ Hz, for consistency with recent Codec 2 VQ work in [2].
6. The $K = 80$ are linear spaced, $K = 20$ Mel spaced.

Further work:

1. If fixed $K$ works, try to come up with a network that handles sparse vectors.
2. Are MFCCs good choices for input features to neural vocoders? They are smoothed, linear transformations of the input speech. Other latent spaces may be more suitable.
3. A linear transformation (like a DCT or PCA) followed by VQ is a good idea - truncating DCT coeffs will reduce the dimension, making storage easier, and VQ will capture any non-linear dependencies. Smaller dimension will help residual error problem in multistage VQ. Trade off between truncation and VQ distortion.
4. Can we avoid gradient shift/tendancy towards Gaussians with multistage VQ? Does mbest help?
5. Scripts supporting this study need refactoring.
6. It would be useful to look at LPCNet excitation and LPC filter spectra to see how males are handled, as that has a fairly coarse spectral represesntation. So we would expect excitation for males to have notches either side of formants.
7. Consider a model that represnts a 2D time-freq formant "blob". This is what the ear detects - energy at a certain freq for a certain amount of time. This works suggests time-freq distribution should be considered together, rather than a frame-frame breakdown.

# References


[1] Codec 2. `https://github.com/drowe67/codec2/doc/codec2.pdf`.

[2] FreeDV-015 Codec 2 Rate K Resampler. `https://github.com/drowe67/misc/ratek_resampler/ratek_resampler.pdf`.

[3] Francois Chollet. Deep learning with python. 2017. *Shelter Island, NY: Manning*, 2018.

[4] Steven Davis and Paul Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE transactions on acoustics, speech, and signal processing*, 28(4):357–366, 1980.

[5] John Makhoul, Salim Roucos, and Herbert Gish. Vector quantization in speech coding. *Proceedings of the IEEE*, 73(11):1551–1588, 1985.