A Crash Course on Data Compression

FMCW radar data compression

Giulio Meucci[1,2], **Francesco Mancuso**[1,2]

[1] *Department of Information Engineering (DII), University of Pisa*
[2] *Radar and Surveillance System National Laboratory (RaSS), CNIT*

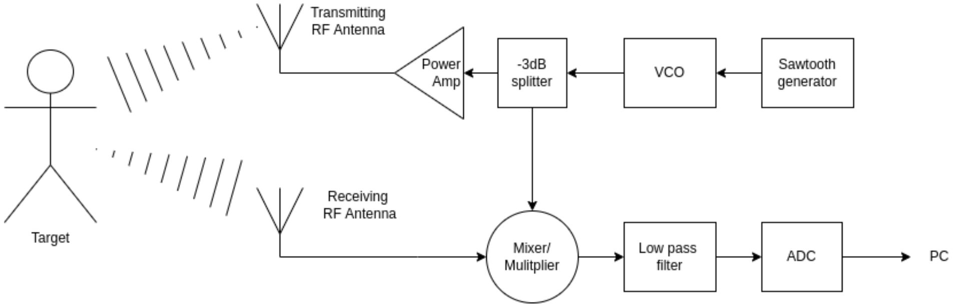
`{giulio.meucci, francesco.mancuso} @phd.unipi.it`

Overview

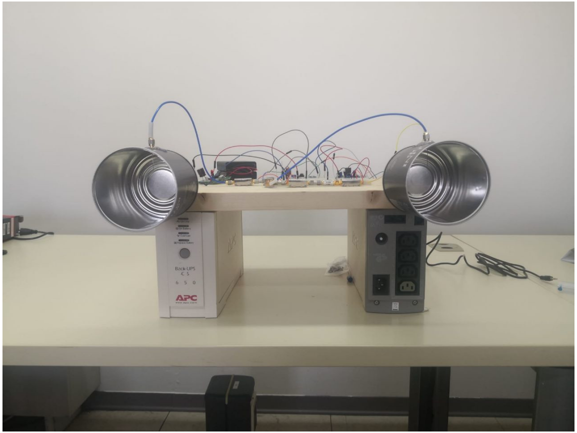
- FMCW radar
- Linear Predictive Coding
- Huffman encoding
- Processing scheme
- Results
- Future works

FMCW radar (I)

Frequency Modulated Continuous Wave (FMCW) radar are among the simplest kind of radar system available. In its basic form, it consist of a modulator, a multiplier, a low-pass filter and two separate RF chain for continuously transmitting and receiving signals.



FMCW radar (II)



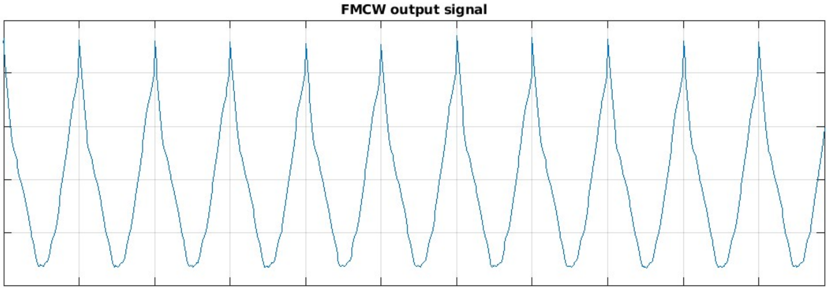
Recently MIT published online a DIY project for building a simple FMCW radar[1] from few discrete components, two coffee cans used as antennas, and a common audio ADC for signal acquisition.

The radar transmits on the $2.4\,\mathrm{GHz}$ ISM band with a frequency span of nearly $80\,\mathrm{MHz}$.

[1] Build a Small Radar System

FMCW radar (III)

The output data is a mixture of sinusoidal signal which frequency depends on the distance of the reflecting object. Such a signal is characterized by strongly periodic components.



Linear Predictive Coding (I)

Linear Predictive Coding (LPC) [5] is a popular technique for **parametric** (i.e., model-based) **analysis** of certain physical signals. LPC assumes an all-pole model [4] (aka, *autoregressive* model), thus the signal is modeled by resonances only and then it is given as a linear combination of past values and some input u_n (G is a gain factor):

$$s_n = -\sum_{k=1}^{p} a_k s_{n-k} + G u_n \tag{1}$$

Now let's see the problem in terms of **prediction** [3]. Here we assume that the input u_n is totally unknown and we would like to predict s_n using the last *p* samples:

$$\tilde{s}_n = -\sum_{k=1}^{p} a_k s_{n-k} \tag{2}$$

Linear Predictive Coding (II)

The error between the real value s_n and our estimation $\tilde{s}_n$ is given by:

$$e_n = s_n - \tilde{s}_n = s_n + \sum_{k=1}^{p} a_k s_{n-k}, \tag{3}$$

which is also known as **residual**. The problem now is to determine the predictor coefficients a_k in order minimize the mean or the total squared error w.r.t. each of the parameters (i.e., *least square error* approach). Let E be the energy of the error signal:

$$E = \sum_n e_n^2 = \sum_n \left(s_n + \sum_{k=1}^{p} a_k s_{n-k} \right)^2, \tag{4}$$

Linear Predictive Coding (III)

The values of a_k that minimize E are found by setting:

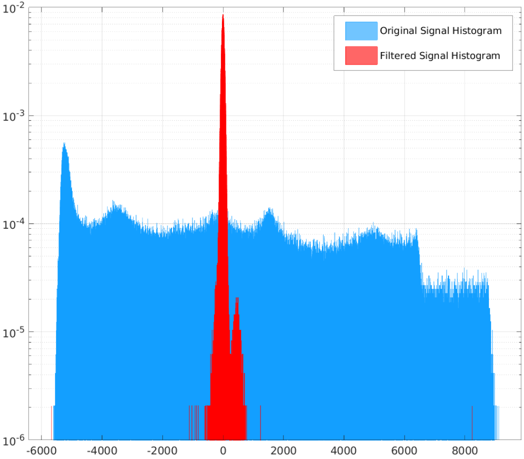
$$\frac{\partial E}{\partial a_i} = 0, \quad 1 \leq i \leq p, \tag{5}$$

and from Equations (4) and (5) we obtain:

$$\begin{gathered}\sum_{k=1}^{p} a_k \sum_n s_{n-k} s_{n-i} = -\sum_n s_n s_{n-i}, \quad 1 \leq i \leq p \\ \sum_{k=1}^{p} a_k R(i-k) = -R(i), \quad 1 \leq i \leq p\end{gathered} \tag{6}$$

where $R(i)$ is the autocorrelation function of s_n. By solving this set of p equations [1], the coefficients a_k that minimize the energy of the error are found. ■

Huffman encoding (I)



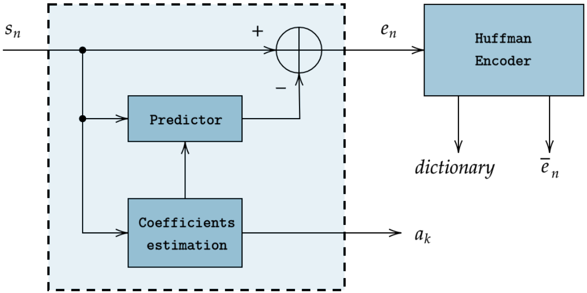
Due to the structure of our signal, we can strongly reduce its dynamic using LPC, storing all the "highly correlated information" in a few coefficients (i.e., the FIR prediction filter), and after that we can encode the residual e_n using Huffman.

Huffman encoding (II)

Huffman coding [2] is a lossless statistical data compression algorithm which always produce optimal prefix-free code. Using a bottom-up strategy, it assigns variable-length codes to input symbols (`int16`) whose lengths are based on the frequency of occurrence of the corresponding symbols.

SYMBOLS	CODE WORD
-5678	0011111101110011101
-1114	0011111101110011100
-1030	0011111101110011111
⋮	⋮
-1	0010111
0	0011100
1	0100110
⋮	⋮
784	000101111111100010001
1239	000101111111100010000
8244	00010111111110001001

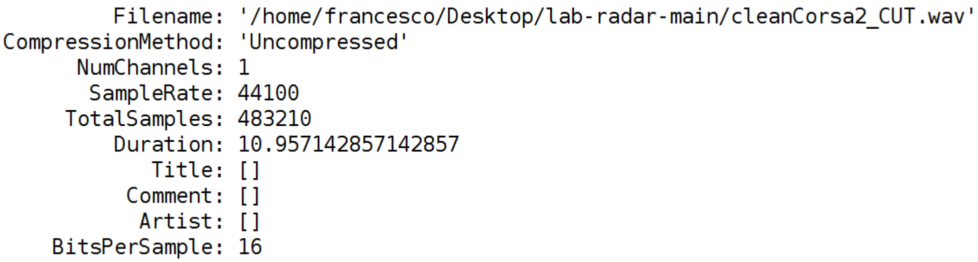
Processing scheme



The input s_n is the audio signal stored in a uncompressed *.wav* file, every sample is represented as an `int16`. The residual e_n is passed to the **Huffman Encoder**, that gives in output the compressed residual $\bar{e}_n$ and the obtained dictionary. Now we have to store p prediction coefficients ($64 \times p$ bit), the dictionary and $\bar{e}_n$.

Results (I)

The uncompressed file occupies $483\,210$ samples $\times\ 16\,\mathrm{bit} = 7\,731\,360\,\mathrm{bit} \approx 966\,\mathrm{kB}$

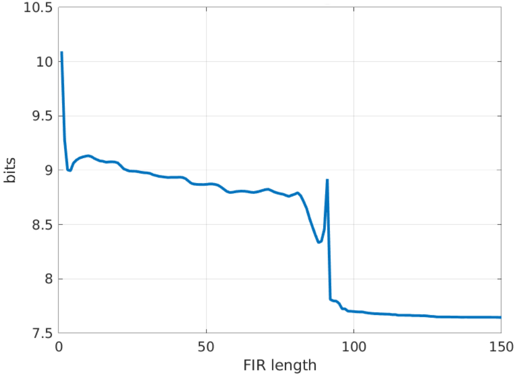


```
         Filename: '/home/francesco/Desktop/lab-radar-main/cleanCorsa2_CUT.wav'
CompressionMethod: 'Uncompressed'
      NumChannels: 1
       SampleRate: 44100
     TotalSamples: 483210
         Duration: 10.957142857142857
            Title: []
          Comment: []
           Artist: []
    BitsPerSample: 16
```

The size of the compressed file, using a FIR filter with 100 coefficients:

$$\text{length}\,(bitstream) + (64 \cdot 100) + \text{size}\,(dictionary) = 3\,760\,003\,\mathrm{bit} \approx 470\,\mathrm{kB}$$

Results (II)



We ran the compressor 150 times, changing the length of the FIR filter for every iteration. In the picture there are the results that we obtained. After 100 coefficients there is a plateau of the **average codeword length** at about 7.7 bit, this is why we chose 100 as filter length.

$$\text{Compression Ratio} = \frac{7\,731\,360}{3\,760\,003} = \mathbf{2.056}$$

Future works

We successfully reduced the size of the data using this approach, but the use of LPC on this kind of data raised some other questions, due to its nature of *spectrum-least-square-fitter*:

- The number of FIR coefficients that minimize the average codeword length is almost equal to the number of samples that belong to a single sweep. Does it have a direct physical meaning?
- Would be useful to apply a per-sweep compression approach, by estimating the FIR coefficients on single sweeps?

If the LPC filter works as an estimator for the stationary part of the signal, we could store only the residual because the target information is kept almost intact in e_n.

References

[1] **N. Levinson**. “The Wiener (Root Mean Square) Error Criterion in Filter Design and Prediction”. In: *Journal of Mathematics and Physics* 25.1-4 (1946), pp. 261–278. DOI: https://doi.org/10.1002/sapm1946251261.

[2] **David A. Huffman**. “A Method for the Construction of Minimum-Redundancy Codes”. In: *Proceedings of the IRE* 40.9 (1952), pp. 1098–1101. DOI: 10.1109/JRPROC.1952.273898.

[3] **J. Makhoul**. “Linear prediction: A tutorial review”. In: *Proceedings of the IEEE* 63.4 (1975), pp. 561–580. DOI: 10.1109/PROC.1975.9792.

[4] **S.M. Kay and S.L. Marple**. “Spectrum analysis—A modern perspective”. In: *Proceedings of the IEEE* 69.11 (1981), pp. 1380–1419. DOI: 10.1109/PROC.1981.12184.

[5] **D. O’Shaughnessy**. “Linear predictive coding”. In: *IEEE Potentials* 7.1 (1988), pp. 29–32. DOI: 10.1109/45.1890.