Neural Network Filter Bank Methods for Spectrum Sensing

Harmeet Gill
Departament of Electrical and Computer Engineering
San Diego, University of California
La Jolla, CA
hsgill@eng.ucsd.edu

The successful detection of radio signals serves a fundamental need for many applications (Cognitive Radios, Radar, Military Comms, etc...). This paper focuses on spectrum sensing using neural networks, with a filter-bank-based preprocessing step to obtain better Signal to Noise ratios (SNR). The performance of this sensing architecture is compared to conventional Energy Detection and Cyclostationary detection under various noise uncertainty models.

Keywords—Energy Detector, Cyclostationary Detector, Neural Network, Probability of Detection, Radio Frequency (RF), Signal to Noise Ratio (SNR), Low Probability of Interception/Detection (LPI/LPD)

I. INTRODUCTION

Recent years have shown a growing interest in the use of advanced computing techniques for communication links. At the physical layer, smart algorithms are being applied to create a new class of radios called cognitive radios. These radios can make decisions regarding its transmitted characteristics based on knowledge of the environment (location, spectrum, etc..). With the use of spectrum sensing algorithms, applications such as cognitive radios can detect co-existing transmissions to protect them from interference, and better utilize unused frequency bands (known as spectrum holes).

Without an *a priori* knowledge of where signals are being transmitted and at which carrier, radios must scan the RF spectrum and make decisions regarding which frequency bands are being utilized. Here, we can see the obvious use of filter banks. With communication bandwidths spanning specific frequency ranges, efficient spectrum scanning can be achieved by splitting the frequency range into multiple bands using a filter bank. The filtering of signals into specific bands is important in comms links to reduce the noise power. Better SNR allows for larger ranges between communication nodes, and for cognitive radios, a higher probability to detect weak signals and avoid their transmission bands.

Once filtered properly, the detection of signals can be implemented using several algorithms. Energy detectors are the simplest form of spectrum sensing and require little computational loads. Energy detectors can, however, be less reliable when high variance noise data is introduced to the signal path. Cyclostationary detection is another conventional algorithm that seeks to extract statistical features from radio signals. Although more computationally complex, the cyclostationary detection method offers more robust detection at lower SNRs due to its autocorrelation algorithm effectively reducing noise power correlation. Cyclostationary detectors,

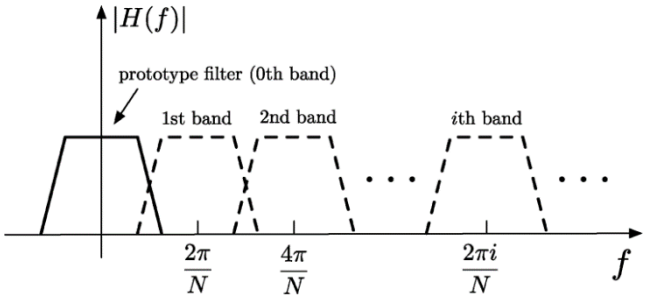

Figure 1: Graphical representation of a filter bank (DFT) [1]

however, also exhibit limitations regarding signal detection. If a signal suffers channel fading (decoherence of signal phase, typically caused by multipath issues in the signal path shown in Figure 2), statistical patterns begin to breakdown, making cyclostationary detection less effective. Thanks to the progression of computing power, Neural Network based signal detection have been proposed and shown to be effective in this domain. Neural Networks are optimized to extract patterned features from data in a highly non-linear manner and can combine both the power detection and statistical feature detection used in the conventional algorithms. (As well as higher order feature detection)

The goal of this paper is to explore the performance of conventional and machine learning based spectrum sensing, as well as relate these performances to the proposed filtering technique.

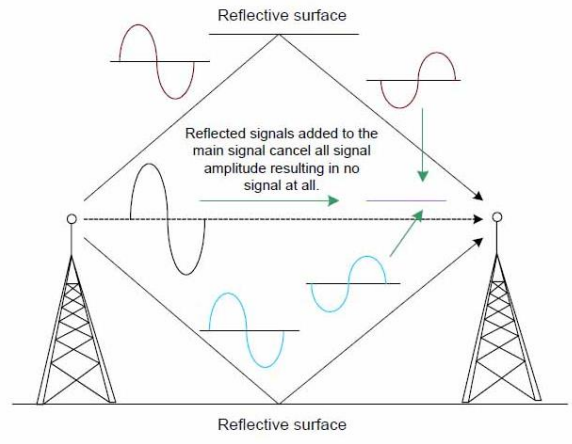

Figure 2: Illustration of multipath causing channel decoherence

II. Spectrum Sensing Algorithms

A. Energy Detectors

Energy detection involves the simplest form of spectrum sensing and consists of summing the magnitude of signal over a sample set and determining whether the squared-sum value is above a certain calibrated threshold.

$$E = \frac{1}{N_s}\sum_{n=1}^{N_s}|y_n|^2$$

Here, y_n refers to the complex RF signal that's captured by the radio in the time domain (referred to as IQ data, satisfying the relationship $y_n = I_n + jQ_n$). The decision of what threshold value to pick involves capturing the RF spectrum when no signal is present and deciding a power value based on a system's accepted probability of false alarm rate.

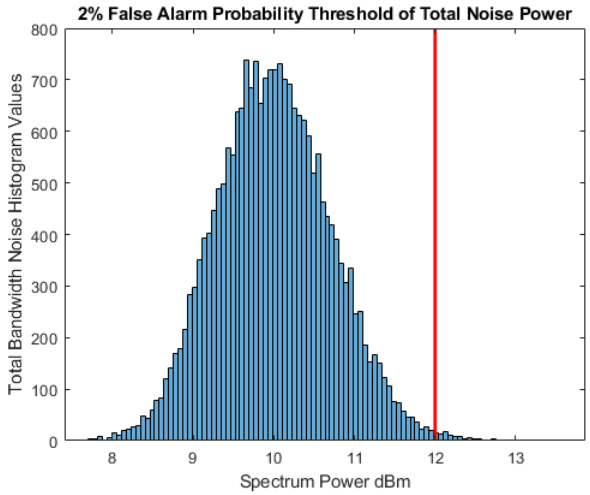

Figure 3: Simulated Noise Power Calibration

Figure 3 shows an example of noise variance used in the detection simulation, with a variance of 1dBm. Under this noise profile, an energy detection threshold of 12dBm was picked, which would cause a false alarm rate of 2 percent. When signal power is introduced, this gaussian profile shifts rightwards over the threshold, and produces an S curve in the probability of detection. (shown later in Figure 13)

B. Cyclostationary detectors

Cyclostationary detectors take an input sample spectrum of complex data and perform autocorrelation on the sample set with various autocorrelation values. (cyclic frequencies)

$$R_x^\alpha(k) \triangleq \lim_{N\to\infty}\frac{1}{2N+1}\sum_{n=-N}^{N}\left[x(n+k)e^{-i\pi\alpha(n+k)}\right]$$

$$\cdot\left[x(n)e^{i\pi\alpha n}\right]^*$$

$$S_x^\alpha(f) \triangleq \sum_{k=-\infty}^{\infty}R_x^\alpha(k)e^{-i2\pi fk}$$

The autocorrelation function, $R_x^\alpha(k)$, performs a convolution of the input signal and introduces a frequency offset, $e^{i\pi\alpha n}$, where α represents the cyclic frequency of the signal. Once calculated, a Fourier transform is performed to obtain the spectral correlation function, $S_x^\alpha(f)$. If the cyclic frequency is zero, once can see that the typical power spectral density of a signal is recovered. Under non-zero cyclic frequencies, a spectral correlation is obtained that contains statistical features of signals.

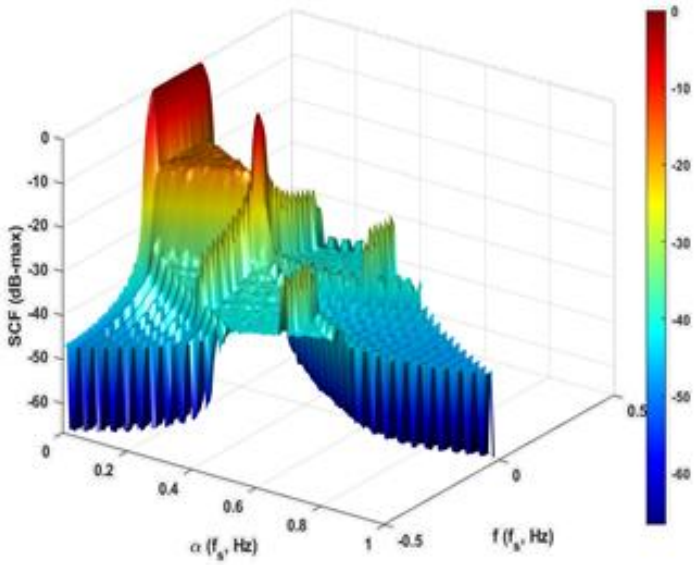

Figure 4: Cyclostationary results of a QPSK Signal

Figure 4 shows the cyclostationary algorithm ran with a high SNR for a simulated QPSK modulated signal, up sampled by a factor of 4. Under this configuration, feature spikes can be observed at $\alpha = 0.25, 0.5$ which indicates the presence of an intelligent signal. If a cyclostationary algorithm was ran on pure noise, one would expect that noise power would not autocorrelated with itself (completely random) and thus have very little spectral correlation features for nonzero cyclic frequencies. Indeed, this behavior is observed in figure 5, where

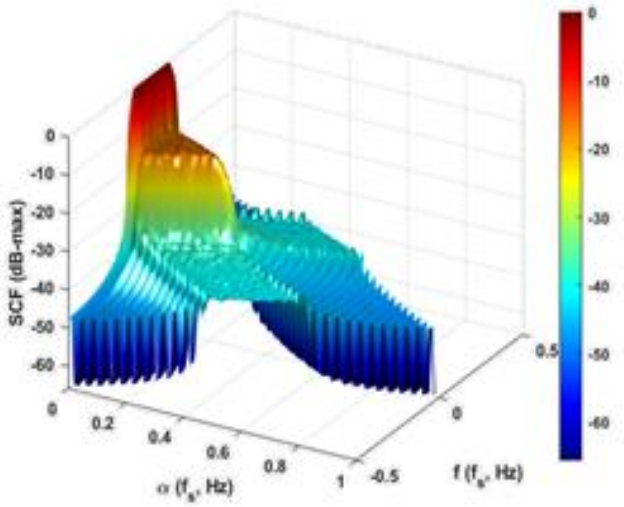

Figure 5: Cyclostationary results of noise power

the lack of cyclical signals causes no significant correlation to occur.

Given these cyclostationary results, a similar thresholding concept can be applied where one picks a correlation magnitude representative of noise and indicative of a certain probability of false alarm. Once picked, subsequent cyclostationary results can be decided to have a signal present depending on whether it's correlation is above the threshold.

The main thresholding difference here versus energy detection comes from the choice of cyclic frequency to calibrate. As seen in Figures 4 and 5, spectral correlation depends on both normal and cyclic frequency. The normal frequency bands can be addressed when applying filter bank methods, however without knowledge of the signal characteristics one doesn't know which cyclic frequency value with hold the feature spike caused by the radio signal. This cyclic frequency is not fixed. For example, if the radio was BPSK modulated with an unsampled rate of 2, the features observed in figure 4 at $\alpha = 0.25$ would instead occur at $\alpha = 0.5$. Thus, cyclostationary detectors have an added complexity of scanning in the cyclic domain. For the purposes of this paper, we will use our *a priori* information of knowing the simulation contains a QPSK modulated signal with a up sample factor of 4 and calibrate against the cyclic frequency of $\alpha = 0.25$.

C. Neural Network Approach

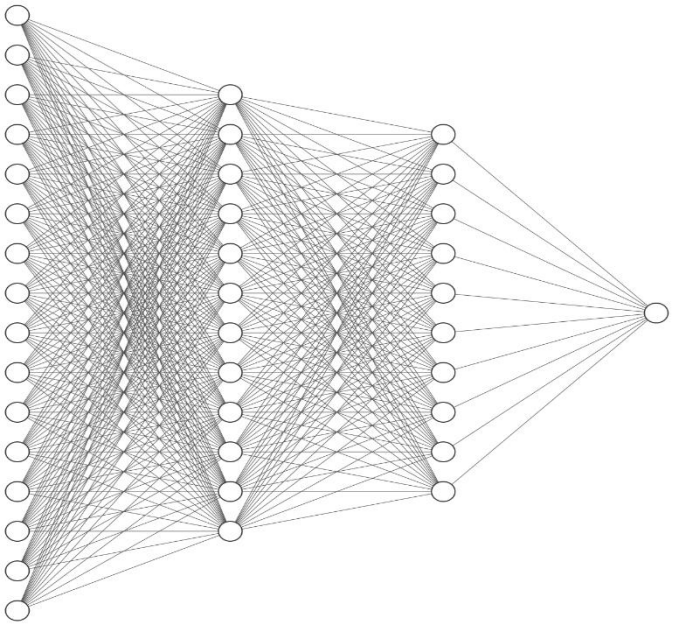

Figure 6: Neural Network Architecture Visualization

The neural network approach is different from conventional algorithms in that one cannot analytically write the detection method and trace input to output. Neural networks are highly non-linear and can have multiple hidden layers between input and output that hold decision weights. Figure 6 shows a typical visualization of a neural network (16 neurons to 12 neurons to 1 neuron). In this network, an input vector consisting of 16 values (could be an 8x8 pixel image) is passed trough the network to an output of one value. (a binary decision, for example whether the 8x8 image contains a specific pattern)

For our implementation, a neural network architecture is established and labeled data is generated. The training data constitutes a set of 1024x2 IQ samples (second dimension representing I and Q respectively) with and without signal in the presence of noise. The network architecture is as follows.

Table 1: Neural Network Architecture

Index	Network Architecture	Dimensions
1	Input	1024x2
2	1x8 Convolution 16	1024x2
3	Maxpool 2x1	512x2
4	1x8 Convolution 24	512x2
5	Maxpool 2x1	256x2
6	1x8 Convolution 32	256x2
7	Maxpool 2x1	128x2
8	1x8 Convolution 48	128x2
9	Maxpool 2x1	64x2
10	1x8 Convolution 64	64x2
11	Maxpool 2x1	32x2
12	1x8 Convolution 96	32x2
13	Averagepool 32x1	32x1
14	Fully Connected Layer	20x1
15	Fully Connected Layer	10x1
16	Fully Connected Layer	2x1
17	Classification Output	2x1

This architecture was picked based on some academic research [2] of spectrum sensing with machine learning, as well as trial and error. The 6 convolutional layers allow the network to relate the input vector to itself (somewhat emulating autocorrelation in cyclostationary detection). Each convolution takes a 1x8 window from the input IQ dataset and convolutes to produce a third dimension of layers. The Max pooling layers reduce the dimensions of the neural network, this is meant to choke the dataflow and allow fundamental features of the signal to be extracted and learned by the network. The output of the neural network produces two values, meant to represent whether the input data did or did not contain a signal. Later sections will explain how this network was trained.

III. FILTERING TECHNIQUES

For this application, our filter design does not care about the synthesis bank or perfect reconstruction conditions in the synthesis bank domain. We only care to have a filter bank with

significantly low stop-band rejection allowing noise power out of band to be kept minimal. Similarly, phase distortion of our filter bank design should be kept at significantly low dB values (<60 dB) to avoid any decoherence that could be used for detection. A simple M-point DFT analysis filter bank was constructed with M=3. The choice of 3 bands was arbitrarily picked to illustrate the effects of proper filtering on spectrum sensing. For the transmit channel of choice for normalized frequency bands was enough to match the transmission bandwidth against one of the channel frequency lengths (matched filtering). In real applications the number of channels could be significantly higher.

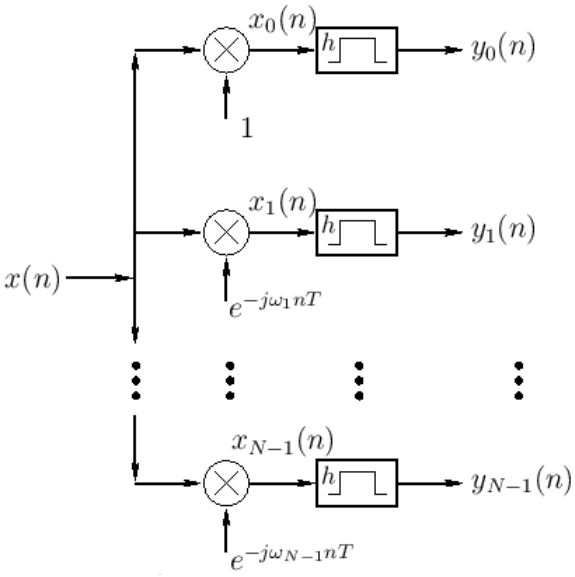

Figure 7: DFT Filter Band Diagram (Frequency Modulated)

With this realized structure, the finite impulse response (FIR) of each branch is expressed with the following equation.

$$y_k(n) = \sum_{m=1}^{N-1} h_0(m)x(n-m)e^{-i\omega_k m}$$

With $h_0(m)e^{-i\omega_k m}$ representing the effective filter response values for each branch. The design parameters are as specified.

Table 2: Filter Design Parameters

Filter Bank	Frequency Modulated DFT
ω_k	0, $\pi/2$, π
Filter length	36 taps
Impulse Response	Equiripple
Stopband Rejection	-70dB

Because our application considers detection ranges between -10dB to 0dB SNR, optimization iterations of stopband rejection or phase distortion was kept low since matching the signal bandwidth was known to have enough of an effect on the detecting the communication channel. Subsequent optimization iterations might consider a value function of signal detection for these filter banks.

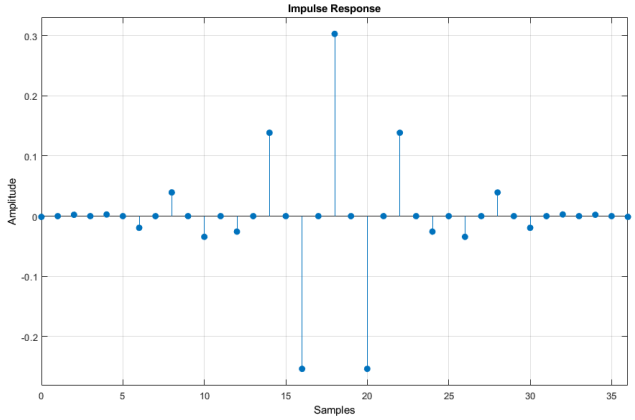

Figure 8: Simulated Passband impulse response

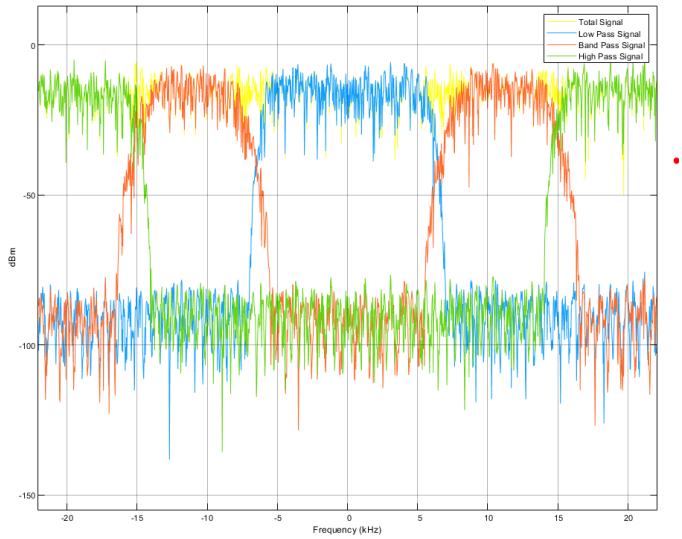

Figure 9: Magnitude response of filter bank

Figures 8 and 9 show the impulse and frequency response of the filter bank in Simulink.

IV. Simulink Radio & Simulation Setup

To perform analysis of both filtering and detection techniques, a Simulink radio (shown in Figure 10) was created to generate data.

This radio consists of a QPSK modulated signal, transmitting baseband and up sampled by a factor of 4. Both the up-sampling and pulse shaping happened in the "Square root" filter block, where the sampling factor and root-raised cosine filter are applied. This is a common radio transmitting technique and is done to prevent inner-symbol interference. The transmitting signal is then amplified and added with a "Random source" block, used to set a gaussian noise floor (as shown in Figure 3). The radio transmits and receives at 200 samples per frame, and after 100 seconds of simulation, produces 120Mb of IQ data. This general approach to very useful for generating data at the physical layer.

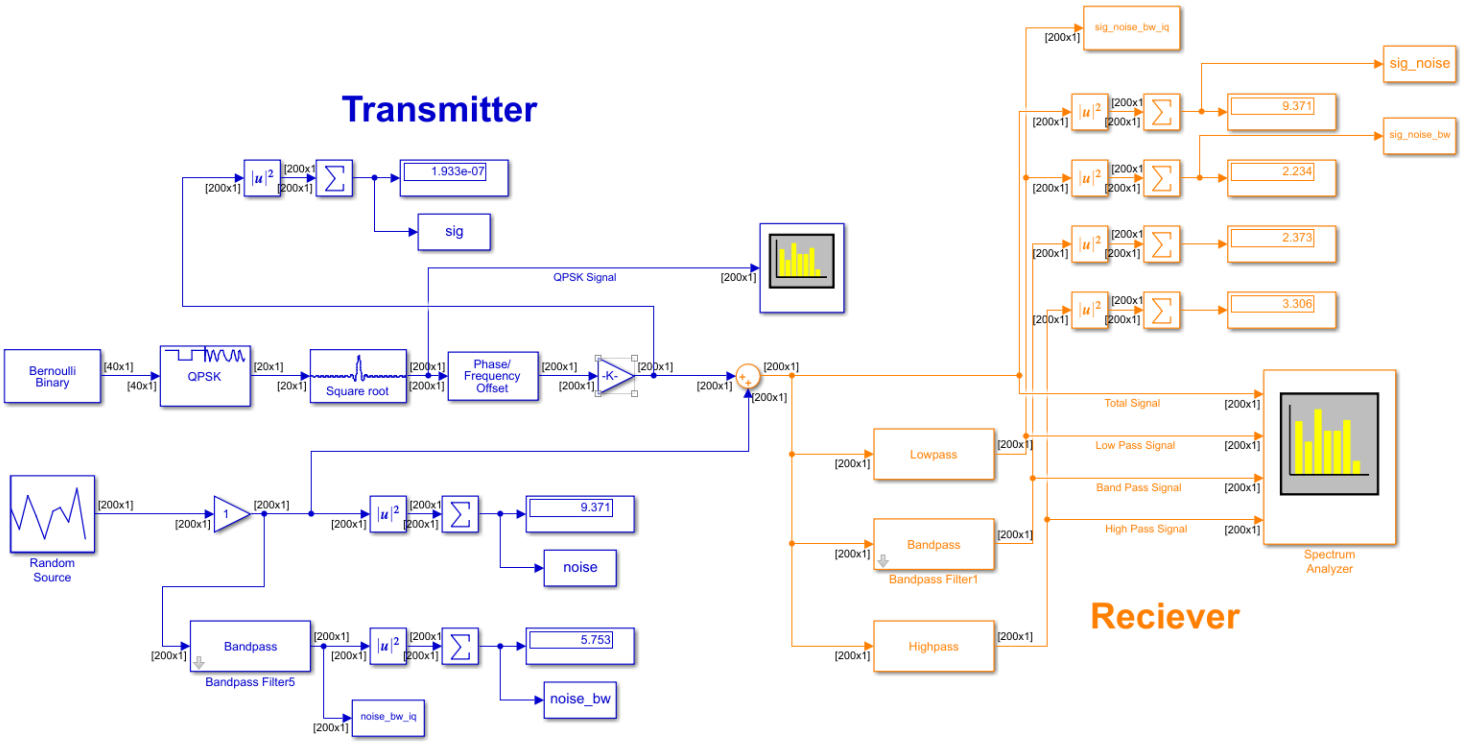

Figure 10: Simulink Based Radio with QPSK Transmitter and DFT Filter Bank Receiver

This radio served useful in generating both IQ sample data as well as sum-squared values over a frame for quick noise calibration.

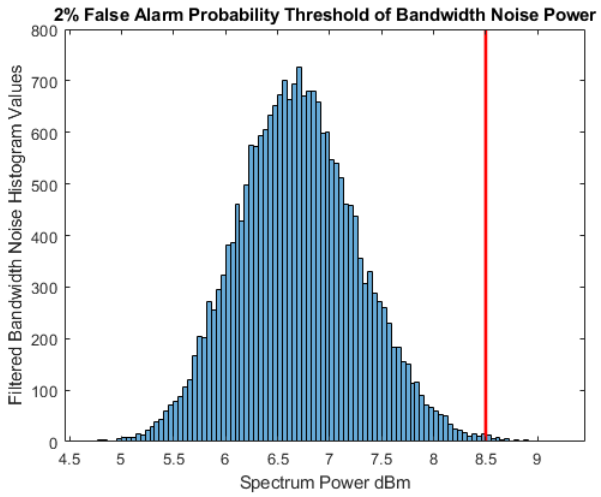

Figure 11: Simulated noise variance of bandpass filtered branch

Figure 11 shows noise floor variance in absence of signal power for the bandpass branch. The gaussian curve takes a similar shape to the total noise power shown in Figure 3, except with the absolute power at a lower level. This indicates the obvious intuition that less noise power is coming out of the branch. If signal energy is perfectly matched against this specific passband range, then SNR is higher than that of the overall system. (As shown in Figure 12) From here, a 2% false alarm rate is set by choosing a threshold value roughly 3 sigma away from the noise mean (8.5 dBm).

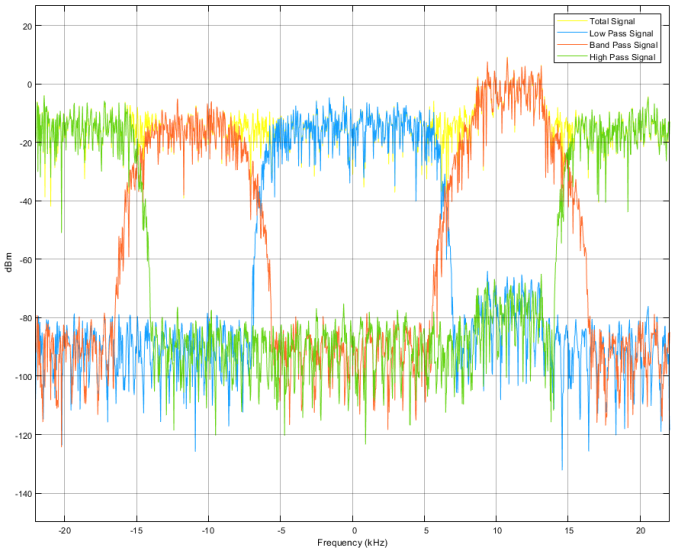

Figure 12: Spectrum with 10 dBm signal power introduced at bandpass

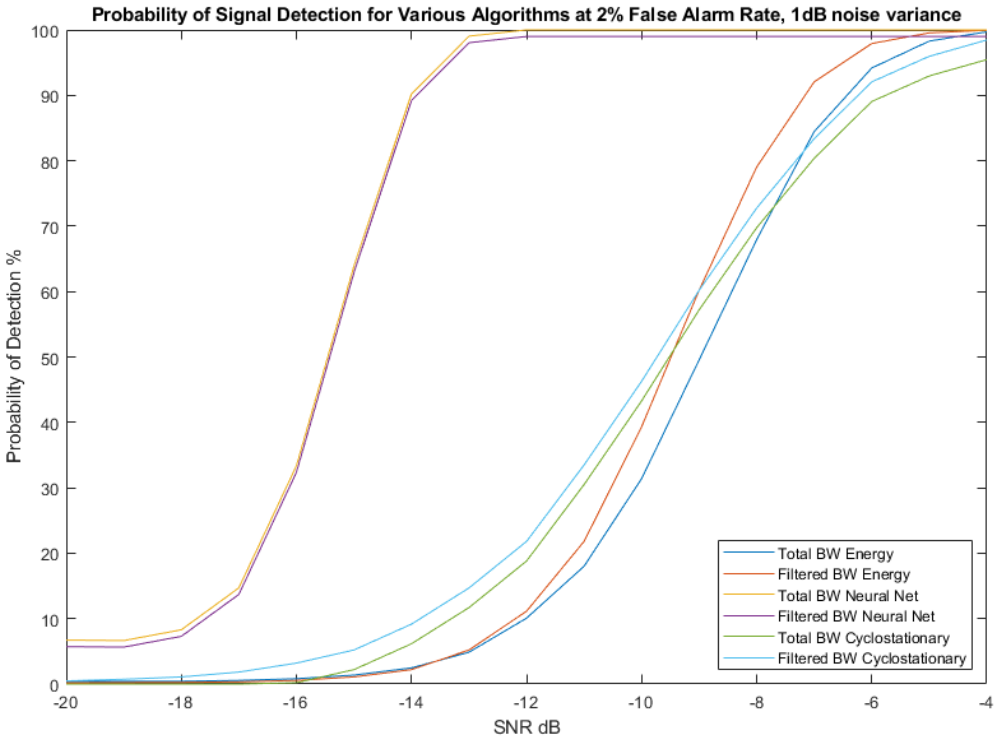

Figure 13: Radio operating curve for various algorithms in Simulink radio

V. SIMULTION RESULTS

A. Neural Network Training Process

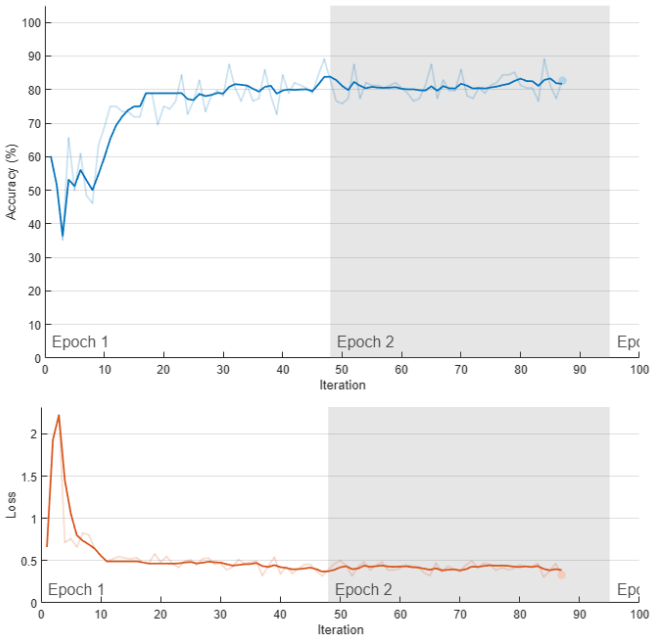

Figure 14: Network Training at -15dB SNR

The neural network training process involved generating IQ data at various signal amplification values. On the simulink transmit side in Figure 10, signal power is amplified by a gain block before it is added with noise power (sum block in Figure 10) and received by the filter bank. To generate training data, the gain block is set to below -50dB gain to kill signal power. With this, a pure noise signal is captured (4200 IQ sets of sample size 1024). Once saved, the gain block is set such that the signal to noise ratio produces an 80% probability of detection from the network training process (Figure 14). This SNR was ultimately determined as being -15dB and was done iteratively by training various SNR signals. This SNR was used as the benchmark dataset to train the network, the reasoning behind this is we don't want to give the network an obvious mixture of no signal/signal data, (for example a high SNR signal) otherwise the network wouldn't learn how to extract features at lower SNRs.

With these two data sets captured (no signal present and low SNR signal present), the datasets were then labeled, (0 for no signal and 1 for signal present) mixed and fed into the network for training using an Adam optimizer. The validation dataset produced roughly 80% accuracy (same as training dataset), indicating that overfitting did not occur. (several dropout layers were also introduced in the architecture to prevent overfitting)

B. Radio Operating Curve Results

Figure 13 shows the probability of detection curve given the 2% false alarm threshold (commonly referred to as a radio operating curve). This curve shows the performance of signal detection with total noise power and bandpass filtered power of the various algorithms mentioned.

Several interesting results can be seen with Figure 13. An obvious result from this set is that the bandpass filtered algorithms performed slightly better than the total signal algorithms (roughly 0.5dB better). Additionally, the neural network beat both Energy and Cyclostationary detectors (filtered and unfiltered) by roughly 6dB. Although the performance of the neural network was expected to be better then traditional algorithms, the 6dB of magnitude difference is a surprising gap. Further investigation will be needed to test the corner cases of this neural network performance.

Another interesting yet obvious result is that the filtered neural network performed as well as the unfiltered neural network. This indicates that the network itself is taking on the task of filtering the input signal and is doing so at an effectiveness close to that of the DFT filter bank. This result, however, can also be explained by the lack of diversity in training data. To test the permeability of this auto filtering by the neural network, a more diverse spectrum of signals at various carrier frequencies, power levels, and bandwidths is needed to train the network against realistic scenarios.

VI. Hardware Implementation and Results

Figure 15: Image of Xilinx ZCU111 RFSoC

A. Hardware Information and Design

Using the same process of capturing IQ data, calibrating the noise floor, and training the neural network against a low SNR case, the proposed detection algorithm was testing using IQ data from a Xilinx ZCU111 RFSoC Evaluation board. The ZCU111 is a prototyping radio packaged with mixers, ADC/DACs, and FPGA space, making it a versatile software defined radio (SDR). The design parameters of the radio are as follows.

Table 3: SDR Design Parameters

Modulation	QPSK
Carrier Frequency	1100 MHz
Baseband Upsample Rate	4
ADC/DAC Sampling	4096 MHz
Baseband Clock Rate	125 MHz
Signal Bandwidth	32 MHz

The radio was designed for specific carrier frequency modulation/demodulation, and therefore had a filter component on transmit and receive that was matched against the transmission (Root raised cosine filter on both transmit and receive baseband channels). Because we are leveraging this readily implemented design, the DFT Filter bank technique cannot be compared, however, neural network performance can still be confirmed against energy detection methods.

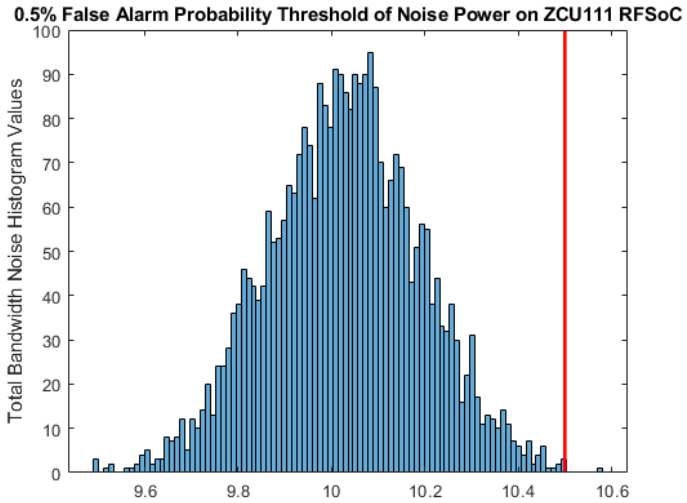

Figure 17: Noise floor calibration of ZCU111

B. Hardware Results

To obtain a noise floor that would allow for a SNR sweep of -30 to -5 dB, a noise source generator was injected between the transmit and receive RF path using a 3-way splitter. Figure 17 shows a histogram of the receive noise power on the ZCU111 under such configuration. The specific noise source used generates a noise variance close to 0.2 dB. With this, a 0.5% probability of false alarm was chosen and energy detection threshold value set to 10.5dBm. The proposed neural network architecture was then trained using radio IQ data with no signal present and -15dB SNR of signal producing a training result as shown in Figure 18.

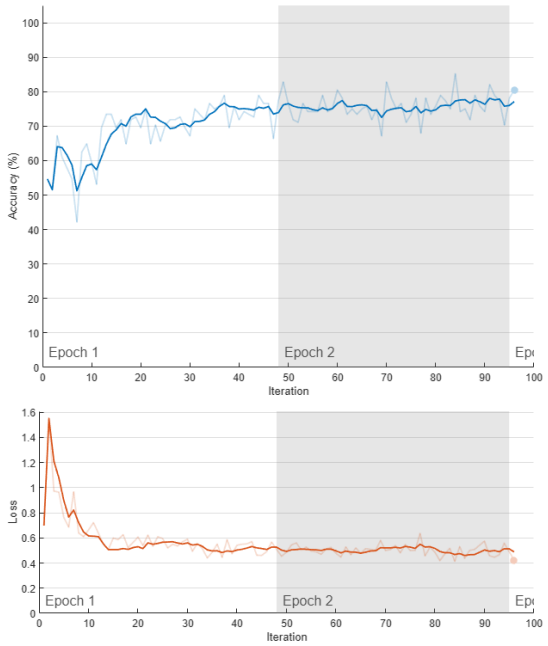

Figure 18: Neural Network Training on ZCU111 IQ Data

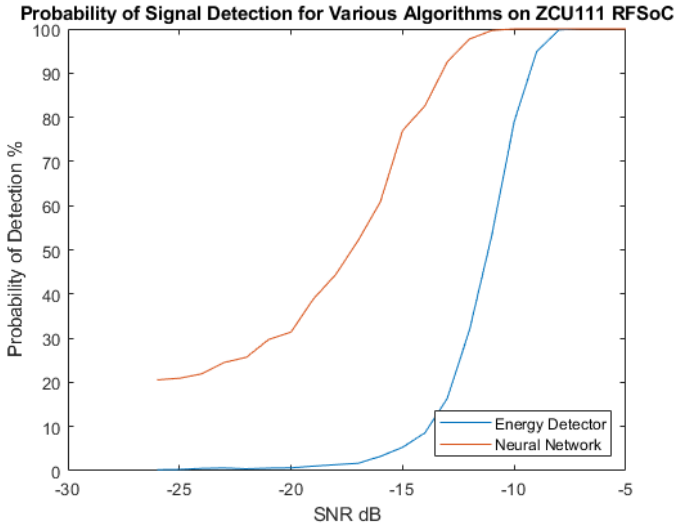

Figure 19: Radio operating curve for various algorithms on ZCU111 RFSoC

For the hardware results, the neural network algorithm defeats energy detection by approximately 4dB. This SNR gap is smaller than the simulated results, which could be explained by the smaller noise variance present on board. Another noteworthy result is that the neural network floats around 20% probability of detection with extremely low SNRs, suggesting that the network is unstable about the decision to declare no-signal. Further investigation is needed to optimize the algorithm.

VII. Conclusion

A machine learning algorithm with a DFT based filter bank architecture was presented and analyzed in this paper. The proposed network has been shown to outperform conventional energy and cyclostationary detection of signals by 6dB in simulation and 4dB in hardware. This performance increase is important to the operation of cognitive radios in detecting vacant frequency bands, as well as situations where low probability of detection is vital (Radar, military comms, etc..).

Future work will concern the characterization of the neural network with various noise power models as well as operational waveforms. From the filter bank side, a larger channel size with narrowband filters can be introduced to better detect signals of variable bandwidths against noise power. The success of machine learning techniques in these radios depend heavily on the user's ability to generate diverse and realistic data for the algorithms to learn properly against normal operation and corner cases. It is therefore a crucial next step to diversify the model in Simulink to include a host of radio operation scenarios as well as introduce these methods to SDRs (modulation types, power levels, bandwidths, burst waveforms, DSSS, etc..).

Acknowledgments

Thank you, professor, for teaching an awesome course this quarter! I learned a lot and gained a better understanding of DSP.

Thanks to the LPI/LPD team at NG for introducing me to the world of radios! (And saving me from my previously boring rotation 😊)

Source Code

https://github.com/Harmeetgg/ML-Spectrum-Sensing

Some of the code base is proprietary (anything relating to our cyclostationary detection algorithms and ZCU111 radio IQ data) and have therefore been omitted from the public repository.

References

[1] B. Farhang-Boroujeny, "Filter Bank Spectrum Sensing for Cognitive Radios," in IEEE Transactions on Signal Processing, vol. 56, no. 5, pp. 1801-1811, May 2008, doi: 10.1109/TSP.2007.911490.

[2] arXiv:1909.06020 [eecs] "Spectrum Sensing Based on Deep Learning Classification for Cognitive Radios"

[3] R. Mahesh, A. P. Vinod, C. Moy and J. Palicot, "A Low Complexity Reconfigurable Filter Bank Architecture for Spectrum Sensing in Cognitive Radios," 2008 3rd International Conference on Cognitive Radio Oriented Wireless Networks and Communications (CrownCom 2008), Singapore, 2008, pp. 1-6, doi: 10.1109/CROWNCOM.2008.4562506.

[4] S. Haykin, D. J. Thomson and J. H. Reed, "Spectrum Sensing for Cognitive Radio," in Proceedings of the IEEE, vol. 97, no. 5, pp. 849-877, May 2009, doi: 10.1109/JPROC.2009.2015711.

[5] T. Hunziker, U. U. Rehman and D. Dahlhaus, "Spectrum sensing in cognitive radios: Design of DFT filter banks achieving maximal time-frequency resolution," 2011 8th International Conference on Information,

Communications & Signal Processing, Singapore, 2011, pp. 1-5, doi: 10.1109/ICICS.2011.6173530.

[6] Trac Duy Tran and T. Q. Nguyen, "On M-channel linear phase FIR filter banks and application in image compression," in IEEE Transactions on Signal Processing, vol. 45, no. 9, pp. 2175-2187, Sept. 1997, doi: 10.1109/78.622942.

[7] T. T. Nguyen, M. Hoffmann, T. Kreul and T. Kaiser, "Experiments on multi-channel spectrum sensing with filter bank realization for cognitive LTE-A systems," 2015 International Conference on Advanced Technologies for Communications (ATC), Ho Chi Minh City, 2015, pp. 83-587, doi: 10.1109/ATC.2015.7388396.M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.

[8] R. Tandra and A. Sahai, "SNR Walls for Signal Detection," in IEEE Journal of Selected Topics in Signal Processing, vol. 2, no. 1, pp. 4-17, Feb. 2008, doi: 10.1109/JSTSP.2007.914879.

[9] R. Tandra and A. Sahai, "SNR Walls for Feature Detectors," 2007 2nd IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks, Dublin, 2007, pp. 559-570, doi: 10.1109/DYSPAN.2007.79.

[10] Y. Arjoune and N. Kaabouch, "On Spectrum Sensing, a Machine Learning Method for Cognitive Radio Systems," 2019 IEEE International Conference on Electro Information Technology (EIT), Brookings, SD, USA, 2019, pp. 333-338, doi: 10.1109/EIT.2019.8834099.