**A Comparison Algorithms for Sensing the Spectrum Using Deep Learning Techniques**

**ABSTRACT**

In 5G Wireless communications Spectrum sensing is a key function of cognitive radio to prevent the harmful interference with licensed users and identify the available spectrum for improving the spectrum’s utilization. In this paper a deep learning-based approach is introduced to sense the spectrum for availability of the primary user. If the primary user is absence, then a secondary user can be allocated in the spectrum. In ultra-dense networks and in dense network traffic conditions there is a scope for continuous utilization of the spectrum by the licensed user. Along with this approaches we will implement spectrum sensing using Energy detection technique, Entropy technique, and optimized features. Telecommunications can expand its services to huge volume of users without any disturbances. This process will show better results when compared to state of art methods.

**CHAPTER 1**

**INTRODUCTION**

A cognitive radio (CR) is a [radio](https://en.wikipedia.org/wiki/Radio) that can be programmed and configured dynamically to use the best [wireless channels](https://en.wikipedia.org/wiki/Wireless_channel) in its vicinity to avoid user interference and congestion. Such a radio automatically detects available channels in [wireless spectrum](https://en.wikipedia.org/wiki/Radio_spectrum), then accordingly changes its [transmission](https://en.wikipedia.org/wiki/Transmission_(telecommunications)) or [reception](https://en.wikipedia.org/wiki/Telecommunication) parameters to allow more concurrent [wireless communications](https://en.wikipedia.org/wiki/Wireless_communications) in a given spectrum band at one location. This process is a form of [dynamic spectrum management](https://en.wikipedia.org/wiki/Dynamic_spectrum_management).

In response to the operator's commands, the cognitive engine is capable of configuring radio-system parameters. These parameters include "[waveform](https://en.wikipedia.org/wiki/Waveform), protocol, operating frequency, and networking". This functions as an autonomous unit in the communications environment, exchanging information about the environment with the networks it accesses and other cognitive radios (CRs). A CR "monitors its own performance continuously", in addition to "reading the radio's outputs"; it then uses this information to "determine the [RF](https://en.wikipedia.org/wiki/Radio_frequency) environment, channel conditions, link performance, etc.", and adjusts the "radio's settings to deliver the required quality of service subject to an appropriate combination of user requirements, operational limitations, and regulatory constraints".

Some "smart radio" proposals combine [wireless mesh network](https://en.wikipedia.org/wiki/Wireless_mesh_network)—dynamically changing the path messages take between two given nodes using [cooperative diversity](https://en.wikipedia.org/wiki/Cooperative_diversity); cognitive radio—dynamically changing the frequency band used by messages between two consecutive nodes on the path; and [software-defined radio](https://en.wikipedia.org/wiki/Software-defined_radio)—dynamically changing the protocol used by message between two consecutive nodes.

J. H. Snider, Lawrence Lessig, David Weinberger, and others say that low power "smart" radio is inherently superior to standard broadcast radio.

The concept of [cognitive](https://en.wikipedia.org/wiki/Cognition) radio was first proposed by Joseph Mitola III in a seminar at [KTH](https://en.wikipedia.org/wiki/Royal_Institute_of_Technology) (the Royal Institute of Technology in Stockholm) in 1998 and published in an article by Mitola and Gerald Q. Maguire, Jr. in 1999. It was a novel approach in wireless communications, which Mitola later described as:

The point in which wireless [personal digital assistants](https://en.wikipedia.org/wiki/Personal_digital_assistant) (PDAs) and the related networks are sufficiently computationally intelligent about radio resources and related computer-to-computer communications to detect user communications needs as a function of use context, and to provide radio resources and wireless services most appropriate to those needs.

[[1]](https://en.wikipedia.org/wiki/Cognitive_radio#cite_note-mitola_dissertation-1)

Cognitive radio is considered as a goal towards which a [software-defined radio](https://en.wikipedia.org/wiki/Software-defined_radio) platform should evolve: a fully reconfigurable wireless transceiver which automatically adapts its communication parameters to network and user demands.

Traditional regulatory structures have been built for an analog model and are not optimized for cognitive radio. Regulatory bodies in the world (including the [Federal Communications Commission](https://en.wikipedia.org/wiki/Federal_Communications_Commission) in the United States and [Ofcom](https://en.wikipedia.org/wiki/Ofcom) in the United Kingdom) as well as different independent measurement campaigns found that most [radio frequency](https://en.wikipedia.org/wiki/Radio_frequency) spectrum was inefficiently utilized. [Cellular network](https://en.wikipedia.org/wiki/Cellular_network) bands are overloaded in most parts of the world, but other frequency bands (such as military, [amateur radio](https://en.wikipedia.org/wiki/Amateur_radio) and [paging](https://en.wikipedia.org/wiki/Paging_(telecommunications)) frequencies) are insufficiently utilized. Independent studies performed in some countries confirmed that observation, and concluded that spectrum utilization depends on time and place. Moreover, fixed spectrum allocation prevents rarely used frequencies (those assigned to specific services) from being used, even when any unlicensed users would not cause noticeable interference to the assigned service. Regulatory bodies in the world have been considering whether to allow unlicensed users in licensed bands if they would not cause any interference to licensed users. These initiatives have focused cognitive-radio research on [dynamic spectrum access](https://en.wikipedia.org/wiki/Dynamic_spectrum_management).

The first cognitive radio wireless regional area network standard, [IEEE 802.22](https://en.wikipedia.org/wiki/IEEE_802.22), was developed by IEEE 802 LAN/MAN Standard Committee (LMSC) and published in 2011. This standard uses geolocation and spectrum sensing for spectral awareness. Geolocation combines with a [database of licensed transmitters](https://en.wikipedia.org/wiki/TV_White_Space_Database) in the area to identify available channels for use by the cognitive radio network. Spectrum sensing observes the spectrum and identifies occupied channels. IEEE 802.22 was designed to utilize the unused frequencies or fragments of time in a location. This white space is unused television channels in the geolocated areas. However, cognitive radio cannot occupy the same unused space all the time. As spectrum availability changes, the network adapts to prevent interference with licensed transmissions.

Depending on transmission and reception parameters, there are two main types of cognitive radio:

* Full Cognitive Radio (Mitola radio), in which every possible parameter observable by a wireless node (or network) is considered.
* Spectrum-Sensing Cognitive Radio, in which only the radio-frequency spectrum is considered.

Other types are dependent on parts of the spectrum available for cognitive radio:

* Licensed-Band Cognitive Radio, capable of using bands assigned to licensed users (except for unlicensed bands, such as the [U-NII](https://en.wikipedia.org/wiki/U-NII) band or the [ISM band](https://en.wikipedia.org/wiki/ISM_band)). The [IEEE 802.22](https://en.wikipedia.org/wiki/IEEE_802.22) working group is developing a standard for wireless regional area network (WRAN), which will operate on unused television channels, also known as TV [white spaces](https://en.wikipedia.org/wiki/White_spaces_(radio)).
* Unlicensed-Band Cognitive Radio, which can only utilize unlicensed parts of the radio frequency (RF) spectrum. One such system is described in the [IEEE 802.15](https://en.wikipedia.org/wiki/IEEE_802.15) Task Group 2 specifications, which focus on the coexistence of [IEEE 802.11](https://en.wikipedia.org/wiki/IEEE_802.11) and [Bluetooth](https://en.wikipedia.org/wiki/Bluetooth).
* Spectrum mobility: Process by which a cognitive-radio user changes its frequency of operation. Cognitive-radio networks aim to use the spectrum in a dynamic manner by allowing radio terminals to operate in the best available frequency band, maintaining seamless communication requirements during transitions to better spectrum.
* Spectrum sharing[]](https://en.wikipedia.org/wiki/Cognitive_radio#cite_note-ieeexplore.ieee.org-11): Spectrum sharing cognitive radio networks allow cognitive radio users to share the spectrum bands of the licensed-band users. However, the cognitive radio users have to restrict their transmit power so that the interference caused to the licensed-band users is kept below a certain threshold.
* Sensing-based Spectrum sharing: In sensing-based spectrum sharing cognitive radio networks, cognitive radio users first listen to the spectrum allocated to the licensed users to detect the state of the licensed users. Based on the detection results, cognitive radio users decide their transmission strategies. If the licensed users are not using the bands, cognitive radio users will transmit over those bands. If the licensed users are using the bands, cognitive radio users share the spectrum bands with the licensed users by restricting their transmit power.
* Database-enabled Spectrum Sharing, In this modality of spectrum sharing, cognitive radio users are required to access a [white space database](https://en.wikipedia.org/wiki/TV_White_Space_Database) prior to be allowed, or denied, access to the shared spectrum. The white space database contains algorithms, mathematical models and local regulations to predict the spectrum utilization in a geographical area and to infer on the risk of interference posed to incumbent services by a cognitive radio user accessing the shared spectrum. If the [white space database](https://en.wikipedia.org/wiki/TV_White_Space_Database) judges that destructive interference to incumbents will happen, the cognitive radio user is denied access to the shared spectrum.

Although cognitive radio was initially thought of as a [software-defined radio](https://en.wikipedia.org/wiki/Software-defined_radio) extension (full cognitive radio), most research work focuses on spectrum-sensing cognitive radio (particularly in the [TV](https://en.wikipedia.org/wiki/TV) bands). The chief problem in spectrum-sensing cognitive radio is designing high-quality spectrum-sensing devices and algorithms for exchanging spectrum-sensing data between nodes. It has been shown that a simple energy detector cannot guarantee the accurate detection of signal presence, calling for more sophisticated spectrum sensing techniques and requiring information about spectrum sensing to be regularly exchanged between nodes. Increasing the number of cooperating sensing nodes decreases the probability of false detection.

Filling free RF bands adaptively, using [OFDMA](https://en.wikipedia.org/wiki/OFDMA), is a possible approach. Timo A. Weiss and Friedrich K. Jondral of the [University of Karlsruhe](https://en.wikipedia.org/wiki/University_of_Karlsruhe) proposed a [spectrum pooling](https://en.wikipedia.org/wiki/Spectrum_pooling) system, in which free bands (sensed by nodes) were immediately filled by [OFDMA](https://en.wikipedia.org/wiki/OFDMA) sub bands. Applications of spectrum-sensing cognitive radio include [emergency-network](https://en.wikipedia.org/wiki/Professional_Mobile_Radio) and [WLAN](https://en.wikipedia.org/wiki/Wireless_LAN) higher [throughput](https://en.wikipedia.org/wiki/Throughput) and [transmission](https://en.wikipedia.org/wiki/Transmission_(telecommunications))-distance extensions. The evolution of cognitive radio toward [cognitive networks](https://en.wikipedia.org/wiki/Cognitive_network) is underway; the concept of cognitive networks is to intelligently organize a network of cognitive radios.

The main functions of cognitive radios are:

* Power Control: Power control is usually used for spectrum sharing CR systems to maximize the capacity of secondary users with interference power constraints to protect the primary users.
* Spectrum sensing: Detecting unused spectrum and sharing it, without harmful interference to other users; an important requirement of the cognitive-radio network is to sense empty spectrum. Detecting primary users is the most efficient way to detect empty spectrum. Spectrum-sensing techniques may be grouped into three categories:
  + Transmitter detection: Cognitive radios must have the capability to determine if a signal from a primary transmitter is locally present in a certain spectrum. There are several proposed approaches to transmitter detection:
    - [Matched filter](https://en.wikipedia.org/wiki/Matched_filter) detection
    - Energy detection: Energy detection is a spectrum sensing method that detects the presence/absence of a signal just by measuring the received signal power. This signal detection approach is quite easy and convenient for practical implementation. To implement energy detector, however, noise variance information is required. It has been shown that an imperfect knowledge of the noise power (noise uncertainty) may lead to the phenomenon of the [SNR](https://en.wikipedia.org/wiki/Signal-to-noise_ratio) wall, which is a SNR level below which the energy detector cannot reliably detect any transmitted signal even increasing the observation time. It has also been shown that the SNR wall is not caused by the presence of a noise uncertainty itself, but by an insufficient refinement of the noise power estimation while the observation time increases.
    - [Cyclostationary](https://en.wikipedia.org/wiki/Cyclostationary_process)-feature detection: These type of spectrum sensing algorithms are motivated because most man-made communication signals, such as [BPSK](https://en.wikipedia.org/wiki/Phase-shift_keying), [QPSK](https://en.wikipedia.org/wiki/Phase-shift_keying), [AM](https://en.wikipedia.org/wiki/Amplitude_modulation), [OFDM](https://en.wikipedia.org/wiki/Orthogonal_frequency-division_multiplexing), etc. exhibit cyclostationary behavior. However, noise signals (typically [white noise](https://en.wikipedia.org/wiki/White_noise)) do not exhibit Cyclostationary behavior. These detectors are robust against noise variance uncertainty. The aim of such detectors is to exploit the cyclostationary nature of man-made communication signals buried in noise. Their main decision parameter is comparing the non-zero values obtained by CSD of the primary signal. Cyclostationary detectors can be either single cycle or multicycle cyclostatonary.
* Wideband spectrum sensing: refers to spectrum sensing over large spectral bandwidth, typically hundreds of MHz or even several GHz. Since current ADC technology cannot afford the high sampling rate with high resolution, it requires revolutional techniques, e.g., compressive sensing and sub-Nyquist sampling.
  + Cooperative detection: Refers to spectrum-sensing methods where information from multiple cognitive-radio users is incorporated for primary-user detection
  + Interference-based detection
* Null-space based CR: With the aid of multiple antennas, CR detects the null-space of the primary-user and then transmits within the null-space, such that its subsequent transmission causes less interference to the primary-user
* Spectrum management: Capturing the best available spectrum to meet user communication requirements, while not creating undue interference to other (primary) users. Cognitive radios should decide on the best spectrum band (of all bands available) to meet [quality of service](https://en.wikipedia.org/wiki/Quality_of_service) requirements; therefore, spectrum-management functions are required for cognitive radios. Spectrum-management functions are classified as:
  + Spectrum analysis
  + Spectrum decision

The practical implementation of spectrum-management functions is a complex and multifaceted issue, since it must address a variety of technical and legal requirements. An example of the former is choosing an appropriate sensing threshold to detect other users, while the latter is exemplified by the need to meet the rules and regulations set out for radio spectrum access in international (ITU radio regulations) and national (telecommunications law) legislation.

Applications

Cognitive Radio (CR) can sense its environment and, without the intervention of the user, can adapt to the user's communications needs while conforming to [FCC](https://en.wikipedia.org/wiki/Federal_Communications_Commission) rules in the United States. In theory, the amount of spectrum is infinite; practically, for propagation and other reasons it is finite because of the desirability of certain spectrum portions. Assigned spectrum is far from being fully utilized, and efficient spectrum use is a growing concern; CR offers a solution to this problem. A CR can intelligently detect whether any portion of the spectrum is in use, and can temporarily use it without interfering with the transmissions of other users. According to Bruce Fette, "Some of the radio's other cognitive abilities include determining its location, sensing spectrum use by neighboring devices, changing frequency, adjusting output power or even altering transmission parameters and characteristics. All of these capabilities, and others yet to be realized, will provide wireless spectrum users with the ability to adapt to real-time spectrum conditions, offering regulators, licenses and the general public flexible, efficient and comprehensive use of the spectrum".

Examples of applications include:

* The application of CR networks to emergency and public safety communications by utilizing white space
* The potential of CR networks for executing dynamic spectrum access (DSA)
* Application of CR networks to military action such as chemical biological radiological and nuclear attack detection and investigation, command control, obtaining information of battle damage evaluations, battlefield surveillance, intelligence assistance, and targeting.
* They are also proven to be helpful in establishing Medical Body Area Networks which can be utilized in omnipresent patient monitoring that aids in immediately notifying the doctors regarding vital information of patients such as sugar level, blood pressure, blood oxygen and electrocardiogram (ECG), etc. This gives the additional advantage of reducing the risk of infections and also increases the patient's mobility.
* Cognitive radio is practical also to wireless sensor networks, where packet relaying can take place using primary and secondary queues to forward packets without delays and with minimum power consumption.

Simulation of CR networks

At present, [modeling & simulation](https://en.wikipedia.org/wiki/Modeling_and_simulation) is the only paradigm which allows the simulation of complex behavior in a given environment's cognitive radio networks. Network simulators like [OPNET](https://en.wikipedia.org/wiki/OPNET), [NetSim](https://en.wikipedia.org/w/index.php?title=NetSim&action=edit&redlink=1), [MATLAB](https://en.wikipedia.org/wiki/MATLAB) and [ns2](https://en.wikipedia.org/wiki/Ns_(simulator)) can be used to simulate a cognitive radio network. CogNS is an open-source NS2-based simulation framework for cognitive radio networks. Areas of research using network simulators include:

1. Spectrum sensing & incumbent detection
2. Spectrum allocation
3. Measurement and/or modeling of spectrum usage
4. Efficiency of spectrum utilization

SPECTRUM SENSING

Spectrum sensing is the ability to measure, sense and be aware of the parameters related to the

radio channel characteristics, availability of spectrum and transmit power, interference and

noise, radio’s operating environment, user requirements and applications, available networks

(infrastructures) and nodes, local policies and other operating restrictions. It is done across Frequency, Time, Geographical Space, Code and Phase.

Spectrum Sensing Methods

A number of different methods are proposed for identifying the presence of signal transmission all of which are in early development stage. They are:

 Energy – Detection Based

 Waveform Based

 Cyclostationary – Based

 Radio Identification Based

 Matched filtering Based

We will be dealing with Energy detection Wavelet Packet based spectrum sensing.

ENERGY DETECTION BASED SPECTRUM SENSING

Energy Detection is the most common way of spectrum sensing because of its low computational

and implementation complexities. It is a more generic method as the receivers do not need any

knowledge on the primary user’s signal. The signal is detected by comparing the output of the

energy detector with a threshold which depends on the noise floor. The important challenge with

the energy detector-based sensing is the selection of the threshold for detecting primary users.

The other challenges include inability to differentiate interference from primary users and noise

and poor performance under low signal-to-noise ratio values.

PD (probability of detection) and PF (probability of false alarm) are the important factors for

energy based detection which gives the information of the availability of the spectrum.

WAVELET TRANSFORM

The Wavelet Transform has recently gained a lot of popularity in the field of signal processing

due to its capability of providing both time and frequency information simultaneously, hence

giving a time-frequency representation of the signal.

The traditional Fourier Transform can only provide spectral information about a signal and only

works for stationary signals whereas in many real-world applications, the signals are non

stationary and needs to be processed in real time. The problem with Short Time Fourier

Transform (STFT) goes back to the Heisenberg uncertainty principle which states that it is

impossible for one to obtain which frequencies exist at which time instance but, one can obtain

the frequency bands existing in a time interval. Also, the window used in STFT is of constant

length whereas with Wavelet transform we can have multi resolution analysis i.e. we can

 Analyze the signal at different frequencies with different resolutions.

 Have good time resolution and poor frequency resolution at high frequencies.

 Have good frequency resolution and poor time resolution at low frequencies.

Also, it is more suitable for short duration of higher frequency and longer duration of lower

frequency components.

WAVELET PACKET TRANSFORM

For application of interest noise is primarily of high frequency and the signal of interest is

primarily of low frequency. The wavelet transform decomposes the signal into approximation

(low frequency) and details (high frequency) coefficients, the detail coefficients containing much

noise. The simple method to denoise the signal is to simply reduce the size of the detail

coefficients before using them to reconstruct the signal. This approach is called thresholding.

The detail coefficients cannot be made zero since they contain some important features of the

original signal. The two different approaches which are usually applied to denoise are hard

thresholding and soft thresholding.

Wavelet packet transform is a generalization of wavelet transform which keep splitting both low

pass and high pass sub-bands at all scales in the filter bank approximation and implementation.

Hence it is suitable to finely identify the information in both high and low frequency bands and

thus, is an ideal processing tool for non-stationary time-variable signal.

The following figure is the wavelet packet decomposition tree.

Wavelet packet analysis allows the signal S to be represented as A1 + AAD3 + DAD3 + DD2.

This is an example of a representation that is not possible with ordinary wavelet analysis but

made feasible only with Wavelet Packet Transform.

ENERGY DETECTION MODEL BASED ON WAVELET

PACKET TRANSFORM

The block diagram is similar to the simplest energy-based detector but most importantly a

Wavelet Packet Transform (WPT) block has been introduced which estimates the current noise

and signal power, which is very important for settling threshold. The analog signal x(t) after

being converted into digital signal x(n) is decomposed for a certain level related to the resolution

required and then is reconstructed by wavelet packet decomposition coefficients. And hence the

noise power and reconstructed signal power is estimated.

CYCLOSTATIONARY SPECTRUM SENSING

When a transmitted signal is modulated with a sinusoidal carrier, cyclic prefixes (as in OFDM),

code or hopping sequences (as in CDMA); cyclostationarity is induced i.e. mean, autocorrelation

show periodic behavior. This feature is exploited in a Cyclostationary Feature Detector that

measures a signal property called Spectral Correlation Function.

ADVANTAGES & DISADVANTAGES OF CYCLOSTATIONARY

SPECTRUM SENSING

Cyclostationary Spectrum Sensing performs better than Energy detection because of its noise

rejection ability. This occurs because noise is totally random and does not exhibit any periodic behavior. When we have no prior knowledge about primary user’s waveform which is the

scenario in real life, then best technique is cyclostationary feature detection.

The disadvantage with cyclostationary spectrum sensing is its high complexity which results in

high cost.

**CHAPTER 2**

**LITERATURE SURVEY**

**R. Tandra and A. Sahai:** In this paper we consider the problem of detecting whether a frequency band is being used by a known primary user. We derive fundamental bounds on detection performance in low SNR in the presence of noise uncertainty — the noise is assumed to be white, but we know its distribution only to within a particular set. For clarity of analysis, we focus on primary transmissions that are BPSK-modulated random data without any pilot tones or training sequences. The results should all generalize to more general primary transmissions as long as no deterministic component is present. Speciﬁcally, we show that for every ‘moment detector’ there exists an SNR below which detection becomes impossible in the presence of noise uncertainty. In the neighborhood of that SNR wall, we show how the sample complexity of detection approaches inﬁnity. We also show that if our radio has a ﬁnite dynamic range (upper and lower limits to the voltages we can quantize), then at low enough SNR, any detector can be rendered useless even under moderate noise uncertainty.

**S.V. Nagaraj:** In this paper, we present a simple technique for detection of primary users in cognitive radio networks with unknown noise and interference levels. We will show that the likelihood ratio test for detecting the primary user can be approximated to a formulation that compares the estimated entropy of the received signal to a suitable threshold. This formulation is also intuitive since for a given variance, the entropy of a stochastic signal is maximized if it is Gaussian. If the received signal contains the primary user's digitally modulated component, the entropy is reduced. Although the proposed approach is applicable under any scenario, we will specifically consider matched-filter-based detection in this paper, with its underlying assumption that the cognitive radio knows the primary user signaling waveform. We will consider the case where the Gaussian noise and interference levels in the region are unknown, which renders traditional matched-filtering and energy-based detection approaches unfeasible. The probabilities of successful detection and false alarm are characterized for both classical and Bayesian scenarios.

**D. Bhargavi and C. R. Murthy:** This paper presents a comprehensive performance comparison of energy detection, matched-filter detection, and cyclostationarity-based detection, the three popular choices for spectrum sensing by cognitive radios. Analytical expressions for the false alarm and detection probability achieved by all the detectors are derived. For cyclostationarity-based detection, two architectures that exploit cyclostationarity are proposed: The Spectral Correlation Density (SCD) detector, and the Magnitude Squared Coherence (MSC) detector. The MSC detector offers improved performance compared to existing detectors, and this is demonstrated using the 802.22 RF capture database. It is also shown that the cyclostationarity-based detectors are naturally insensitive to uncertainty in the noise variance, as the decision statistic is based on the noise rejection property of the cyclostationary spectrum. Simulation results plotting the receiver operating characteristics corroborate the theoretical results, and enable visual comparison of the performance.

**Y. L. Zhang, Q. Y. Zhang, and T. Melodia**: Sensitivity to noise uncertainty Isa fundamental limitation of current spectrum sensing strategies in cognitive radio networks (CRN). Because of noise uncertainty, the performance of traditional detectors such as matched ﬁlters, energy detectors, and even cyclostationary detectors deteriorates rapidly at low Signal-to-Noise Ratios (SNR). To counteract noise uncertainty, a new entropy-based spectrum sensing scheme is introduced in this letter. The entropy of the sensed signal is estimated in the frequency domain with a probability space partitioned into ﬁxed dimensions. It is proven that the proposed scheme is robust against noise uncertainty. Simulation results conﬁrm the robustness of the proposed scheme and show 6dB and 5dB performance improvement compared with energy detectors and cyclostationary detectors, respectively. In addition, the sample size is signiﬁcantly reduced compared to an energy detector.

**A. Kortun, T. Ratnarajah, M. Sellathurai, C. Zhong, and C. B. Papadias:** In this paper, the distribution of the ratio of extreme eigenvalues of a complex Wishart matrix is studied in order to calculate the exact decision threshold as a function of the desired probability of false alarm for the maximum-minimum eigenvalue (MME) detector. In contrast to the asymptotic analysis reported in the literature, we consider a finite number of cooperative receivers and a finite number of samples and derive the exact decision threshold for the probability of false alarm. The proposed exact formulation is further reduced to the case of two receiver-based cooperative spectrum sensing. In addition, an approximate closed-form formula of the exact threshold is derived in terms of a desired probability of false alarm for a special case having equal number of receive antennas and signal samples. Finally, the derived analytical exact decision thresholds are verified with Monte-Carlo simulations. We show that the probability of detection performance using the proposed exact decision thresholds achieves significant performance gains compared to the performance of the asymptotic decision threshold.

**M. Jin, Y. Li, and H.-G. Ryu:** Owing to no need for prior knowledge of signal, blind spectrum sensing has received wide attention. Covariance Absolute Value (CAV) detection algorithm, one of the most popular blind sensing algorithms, considers the correlation of signal samples. However, its detection performance is restricted by the uncertain threshold calculation. To optimize the performance of CAV, we propose a new method based on a new statistic and goodness of fit test. The statistic is constructed from the off-diagonal of covariance matrix firstly, then Anderson-Darling (AD) test is used to estimate the existence or absence of primary user. The proposed method not only achieves blind detection but also improves the sensing performance of CAV. Experimental results manifest the effectiveness of the proposed scheme.

**N. Zhao**: Spectrum sensing is a key problem in cognitive radio. However, traditional detectors become ineffective when noise uncertainty is severe. It is shown that the entropy of Gauss white noise is constant in the frequency domain, and a robust detector based on the entropy of spectrum amplitude was proposed. In this paper a novel detector is proposed based on the entropy of spectrum power density, and its performance is better than the previous scheme with less computational complexity. Furthermore, to improve the reliability of the detection, a two-stage entropy-based cooperative spectrum sensing scheme using two-bit decision is proposed, and simulation results show its superior performance with relatively low computational complexity.

**Prieto, Angel G. Andrade, Daniela M. Martinez, and Guillermo Galvanize:** n this work, the evaluation of a spectrum sensing strategy based on the Frequency Domain Entropy applied to cognitive radio networks is presented. Entropy estimation is performed using Bartlett periodogram. A tradeoff between variance and the spectral resolution for Bartlett periodogram is figured out. This tradeoff affects the probability of detection and false alarm of the spectrum sensing strategy under environments with low signal-to-noise ratio and noise uncertainty. The Entropy detector is optimal when the product of the number of segments and the number of points used is equal to the number of available samples of the received signal.

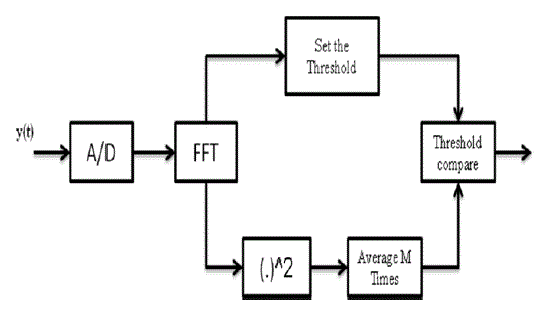
**Drawback:**

* 1. Probability of detection is low.
  2. Computational complexity is high.
  3. Accuracy is low.

**CHAPTER 3**

**EXISTING METHOD**

**Energy Detection:** In this process the data is received from the signal is check with the pre estimated with cut off rang and detect the existence of the user. This threshold can be selected either statistically or dynamically. But this method shows degraded results when noise and interference is high. But to provide high bit error rate the threshold is select dynamically using the noise level present in the signal.

****

**Figure 2: FFT-Based Energy Detector**

signal is converted into the digital using the A/D converters, and fast Fourier transform of the signal is obtained and finally these N samples are averaged by squaring the FFT coefficients [7-10, 26], this can be explain by using the below equation:

Threshold =…….. (3)

This energy Threshold is then compared to a pre-defined threshold to obtain the sensing decision as follows:

Threshold < : nonappearance of PU signal

Threshold >: appearance of PU signal

The overall performance of the detection technique is identified the use of the probability of detection PD and the probability of false alarm Pfa. This can be identified using the ratio of number of correct detections and number of trails used to estimate the energy detection. Meanwhile, the possibility of fake apprehension is approximate using the ratio of number of times, that the primary user is falsely detected to total number of trails. These can be explained using the below equation:

= (Threshold >) …… (4)

= (Threshold >) …… (5)

= non-appearance of Primary User Signal

= appearance of Primary User Signal

Where Threshold corresponds to the energy of N samples given by equation 1 and is the sensing threshold.

The chance [2] of detection and the prospect of false alarm are

= Q …… (6)

= Q …… (7)

Where Q-Function Q(x) = , are the standard deviation of noise and primary user signal and N is number of samples.

Then detection of threshold is [17] given by

= …… (8)

1. **Noise Estimation by using Dynamic Threshold Technique**

A novel technique is introduced to estimate the threshold dynamically using the noise level in organize to defeat the drawback of the auto correlation, method [16-17, 21-22, 24, 27].

In our approach the dynamic threshold is estimated using the noise level in received signal other than the previous method based on the Eigen values. of the covariance matrix. The noise levels are calculated by means of the power of the transmitted and noise signals [4-5, 13, 15].

**Noise Estimation:**

Noise is estimated by using Eigen values of covariance matrix of a received signal [4-5, 9, 19].

Judge the established signal is ‘y’ and it is articulated in the form of N×L matrix,

y= …… (9)

Mutually noise and indication are to be independent and suppose noise is additive white Gaussian noise (AWGN) whose mean and variance are 0, , then equation 1&2 becomes as

: = …… (10)

: = …… (11)

Consider the available bandwidth is ‘B’ and bandwidth occupied is ‘b’ within the sample covariance matrix of the Eigen values. Then the available bandwidth is in use into consideration as in the range 1 to L in which 1 to K is concerned as transmitted signal and rest is noise.

The statistical covariance matrices of noise, transmitted samples and received samples are

= …… (12)

…… (13)

…… (14)

Where

Noise statistical covariance matrix

Transmitted signal covariance matrix

Received Signal covariance matrix

(.)H= Complex Conjugate of Transport Operator

= Noise Variance

= L-Order identity matrix

Then

Ʃw …… (15)

Consider the eigen values λy of Ʃy and λx of Ʃx in a descending order, then the equations are

= + for i=1,2,…K …… (17)

= for i=K+1,K+2…L …… (18)

The approximation of the received statistical covariance matrix can be computed as

= …… (19)

The value of ‘K’ is predictable with the help of the minimal graphic period principle. After estimate the value of K, the covariance matrix [4-5, 9, 19] of eigen values is calculated. Then K value is given by

K= …… (20)

Where

= …… (21)

= …… (22)

Where

L= Number of Eigen values

N= Number of samples

= Set of Eigen values

Based on these Eigen values the noise variance is measured. By using this noise level, the threshold is estimated to detect presence or absence of primary user accurately

**CHAPTER 4**

**PROPOSED METHOD**

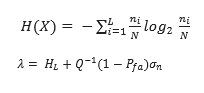
**Proposed Method-1**

Spectrum sensing can be modeled as a binary hypothesis test problem. The hypotheses under test are:

H 0 ∶ x (n) = ω (n) …… (1)

H 1 ∶ x (n) =s (n) +ω (n) …. (2)

n = 0,1…. N-1, Hypotheses H0 and H1 is for idle and busy frequency band respectively. In (1) and (2), x(n), s(n), ω(n) correspond to the received signal, the modulated signal, and the noise samples respectively. Shannon’s entropy, denoted by H, is a measure of the uncertainty present in the random variable. It can be quantified by the following equation:

….. (3,4)

Where

…… (5)

λ- Threshold

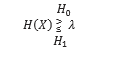
Pfa-Probability of false alarm is set as 0.2 in our paper

γ - Euler-Mascheroni constant

Q-1-Inverse Q function

σn -Standard deviation

Let be the number of elements in X falling inside the bin such that, ∑ = as given in. Entropy is calculated separately for H0 and H1. A threshold is calculated using equation (4) as per and probability of detection is simulated for various SNR Values. For EnBD, the detection strategy consists of testing the entropy obtained as:

…. (6)

H0 - Absence of primary user

H1- Presence of primary user

In previous detection techniques, the performance is acceptable, whose entropy is calculated from the amplitude spectrum. The result is robust to noise uncertainty. In this work, the entropy is calculated using histogram. Threshold is calculated using the equation (4). Probability of detection is calculated by comparing this threshold with the entropy of H1 and H0 as given in equation (6). The performance of spectrum sensing is evaluated by a metric: The Probability of detection Pd. The Pd quantifies the ability to correctly detecting the presence of a primary signal.

Initially, a discrete signal of length 32 is generated in a random manner for and all 32 zeros for H0 hypotheses. This discrete signal is mapped to BPSK and passed through an additive white gaussian noise channel. Entropy is calculated for the random signal using equation (3), [8]. A threshold is determined to calculate the presence or absence of user using the condition in equation (6), as per. Then finally probability of detection is calculated.

PROPOSED ALGORITHM

x Generation of random discrete data of length N=32.

x Map the generated data to Binary Phase Shift Keying.

x Calculate the number of samples in each bin using histogram function in MATLAB.

x Evaluate the average information of 32 sample Using H(X)=∑ log, where L=16 as in.

x Calculate the entropy for the hypothesis H0 and H1 using equation (3).

x A threshold is calculated using the equation (6).

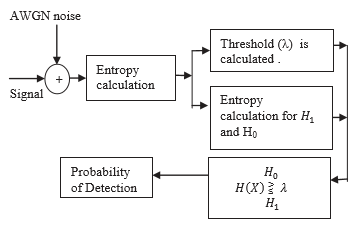
x the probability of detection is calculated by comparing threshold value λ with the entropy of H1 and H0.

x the presence of primary user is identified when entropy of H1 is less than the threshold value λ.

x the absence of primary user is identified when the entropy of H0 is greater than the threshold value λ.

x Thus probability of detection is calculated and the same procedure is iterated.

x A plot for probability of detection is simulated for various values of SNR.

****

**Fig: Block diagram of proposed method**

**Advantages:**

* 1. Probability of detection is high.
  2. Computational complexity is low.
  3. Accuracy is high.

**Proposed Method-2**

The energy is detected and is compared with the dynamic threshold in Rayleigh fading environment in order to increase the probability of detection. Rayleigh fading is a reasonable model when there are many objects in the environment that scatter the radio signal before it arrives at the receiver. The central limit theorem holds that, if there is sufficiently much scatter, the channel impulse response will be well modeled as a Gaussian process irrespective of the distribution of the individual components. If there is no dominant component to the scatter, then such a process will have zero mean and phase evenly distributed between 0 and 2π radians.

The envelope of the channel response will therefore be Rayleigh distributed. The probability of detection will increases rapidly in Rayleigh fading environment when compared to AWGN environment. Hence, the signal which is better when compared to all the signals appeared at the receiver is selected using peak detection process and probability of detection is estimated for that signal which is explained below. The input data is assumed based on the given parameters. This data is modulated using pulse shift keying.

Phase Shift Keying (PSK) is the digital modulation technique in which the phase of the carrier signal is changed by varying the sine and cosine inputs at a particular time. PSK technique is widely used for wireless LANs, bio-metric, contactless operations, along with RFID and Bluetooth communications. Then inverse fast Fourier transforms is applied to the modulated data and the signal is optimized in order to reduce noise in the data.

Then the power is calculated for both the optimized data and the selected data, then the results specify that optimized data can consume less power. Then finally the optimized data is send through the Rayleigh fading channel and energy is estimated and probability of detection is estimated for the data which is optimized. In this process the energy and power required for detection is less when compared to state of art methods as we are processing entire operation on select data.

**Phase-shift keying (PSK)** is a [digital modulation](https://en.wikipedia.org/wiki/Digital_modulation) process which conveys [data](https://en.wikipedia.org/wiki/Data_(computing)) by changing (modulating) the [phase](https://en.wikipedia.org/wiki/Phase_(waves)) of a constant [frequency](https://en.wikipedia.org/wiki/Frequency) reference [signal](https://en.wikipedia.org/wiki/Signal) (the [carrier wave](https://en.wikipedia.org/wiki/Carrier_wave)). The modulation is accomplished by varying the [sine](https://en.wikipedia.org/wiki/Sine_wave) and [cosine](https://en.wikipedia.org/wiki/Cosine_wave) inputs at a precise time. It is widely used for [wireless LANs](https://en.wikipedia.org/wiki/Wireless_LAN), [RFID](https://en.wikipedia.org/wiki/RFID) and [Bluetooth](https://en.wikipedia.org/wiki/Bluetooth) communication.

Any digital modulation scheme uses a finite number of distinct signals to represent digital data. PSK uses a finite number of phases, each assigned a unique pattern of [binary digits](https://en.wikipedia.org/wiki/Bit). Usually, each phase encodes an equal number of bits. Each pattern of bits forms the [symbol](https://en.wikipedia.org/wiki/Symbol_rate) that is represented by the particular phase. The [demodulator](https://en.wikipedia.org/wiki/Demodulator), which is designed specifically for the symbol-set used by the modulator, determines the phase of the received signal and maps it back to the symbol it represents, thus recovering the original data. This requires the receiver to be able to compare the phase of the received signal to a reference signal – such a system is termed coherent (and referred to as CPSK).

CPSK requires a complicated demodulator, because it must extract the reference wave from the received signal and keep track of it, to compare each sample to. Alternatively, the phase shift of each symbol sent can be measured with respect to the phase of the previous symbol sent. Because the symbols are encoded in the difference in phase between successive samples, this is called differential phase-shift keying (DPSK). DPSK can be significantly simpler to implement than ordinary PSK, as it is a 'non-coherent' scheme, i.e. there is no need for the demodulator to keep track of a reference wave. A trade-off is that it has more demodulation errors.

A convenient method to represent PSK schemes is on a [constellation diagram](https://en.wikipedia.org/wiki/Constellation_diagram). This shows the points in the [complex plane](https://en.wikipedia.org/wiki/Complex_plane) where, in this context, the [real](https://en.wikipedia.org/wiki/Real_number) and [imaginary](https://en.wikipedia.org/wiki/Imaginary_number) axes are termed the in-phase and quadrature axes respectively due to their 90° separation. Such a representation on perpendicular axes lends itself to straightforward implementation. The amplitude of each point along the in-phase axis is used to modulate a cosine (or sine) wave and the amplitude along the quadrature axis to modulate a sine (or cosine) wave. By convention, in-phase modulates cosine and quadrature modulates sine.

In PSK, the [constellation points](https://en.wikipedia.org/wiki/Constellation_diagram) chosen are usually positioned with uniform [angular](https://en.wikipedia.org/wiki/Angle) spacing around a [circle](https://en.wikipedia.org/wiki/Circle). This gives maximum phase-separation between adjacent points and thus the best immunity to corruption. They are positioned on a circle so that they can all be transmitted with the same energy. In this way, the moduli of the complex numbers they represent will be the same and thus so will the amplitudes needed for the cosine and sine waves. Two common examples are "binary phase-shift keying" ([BPSK](https://en.wikipedia.org/wiki/Phase-shift_keying#Binary_phase-shift_keying_(BPSK))) which uses two phases, and "quadrature phase-shift keying" ([QPSK](https://en.wikipedia.org/wiki/Phase-shift_keying#Quadrature_phase-shift_keying_(QPSK))) which uses four phases, although any number of phases may be used. Since the data to be conveyed are usually binary, the PSK scheme is usually designed with the number of constellation points being a [power](https://en.wikipedia.org/wiki/Power_(mathematics)) of two.

**Proposed Method-3**

**CNN (Convolutional Neural Network):**

When it comes to Machine Learning, artificial neural network performs really well. Artificial Neural Networks are used in various classification tasks like image, audio, words. Different types of Neural Networks are used for different purposes, for example for predicting the sequence of words we use Recurrent Neural Networks more precisely an LSTM, similarly for image classification we use Convolution Neural Network. In this we are going to build basic building block for CNN. A convolutional neural network can consist of one or multiple convolutional layers. The number of convolutional layers depends on the amount and complexity of the data.

Before diving into the Convolution Neural Network, let us first revisit some concepts of Neural Network. In a regular Neural Network, there are three types of layers:

**1. Input Layers:** It’s the layer in which we give input to our model. The number of neurons in this layer is equal to total number of features in our data (number of pixels in case of an image).

2.  **Hidden Layer:** The input from Input layer is then feed into the hidden layer. There can be many hidden layers depending upon our model and data size. Each hidden layer can have different numbers of neurons which are generally greater than the number of features. The output from each layer is computed by matrix multiplication of output of the previous layer with learnable weights of that layer and then by addition of learnable biases followed by activation function which makes the network nonlinear.

3**. Output Layer:** The output from the hidden layer is then fed into a logistic function like sigmoid or softmax which converts the output of each class into probability score of each class.

The data is then fed into the model and output from each layer is obtained this step is called feed forward, we then calculate the error using an error function, some common error functions are cross entropy, square loss error etc. After that, we back propagate into the model by calculating the derivatives. This step is called back propagation which basically is used to minimize the loss.

A Convolutional neural network (CNN) is a neural network that has one or more convolutional layers and is used mainly for image processing, classification, segmentation and also for other auto correlated data. A convolution is essentially sliding a filter over the input. One helpful way to think about convolutions is this quote from Dr Prasad Samarakoon: “A convolution can be thought as “looking at a function’s surroundings to make better/accurate predictions of its outcome.” Rather than looking at an entire image at once to find certain features it can be more effective to look at smaller portions of the image. The most common use for CNNs is image classification, for example identifying satellite images that contain roads or classifying hand written letters and digits. There are other quite mainstream tasks such as image segmentation and signal processing, for which CNNs perform well at. CNNs have been used for understanding in Natural Language Processing (NLP) and speech recognition; although often for NLP Recurrent Neural Nets (RNNs) are used.

A CNN can also be implemented as a U-Net architecture, which are essentially two almost mirrored CNNs resulting in a CNN whose architecture can be presented in a U shape. U-nets are used where the output needs to be of similar size to the input such as segmentation and image improvement. Each convolutional layer contains a series of filters known as convolutional kernels. The filter is a matrix of integers that are used on a subset of the input pixel values, the same size as the kernel. Each pixel is multiplied by the corresponding value in the kernel and then the result is summed up for a single value for simplicity representing a grid cell, like a pixel, in the output channel/feature map. These are linear transformations; each convolution is a type of affine function. In computer vision the input is often a 3 channel RGB image. For simplicity, if we take a greyscale image that has one channel (a two-dimensional matrix) and a 3x3 convolutional kernel (a two-dimensional matrix). The kernel strides over the input matrix of numbers moving horizontally column by column, sliding/scanning over the first rows in the matrix containing the images pixel values. Then the kernel strides down vertically to subsequent rows.

**Padding:**

To handle the edge pixels there are several approaches:

* Losing the edge pixels
* Padding with zero value pixels
* Reflection padding

Reflection padding is by far the best approach, where the number of pixels needed for the convolutional kernel to process the edge pixels are added onto the outside copying the pixels from the edge of the image. For a 3x3 kernel, one pixel needs to be added around the outside, for a 7x7 kernel then three pixels would be reflected around the outside. The pixel added around each side is the dimension, halved and rounded down.

Traditionally in many research papers, the edge pixels are just ignored, which loses a small proportion of the data and this gets increasing worse if there are many deep convolutional layers. For this reason, I could not find existing diagrams to easily convey some of the points here without being misleading and confusing stride 1 convolutions with stride 2 convolutions.

With padding, the output from a input of width w and height h would be width w and height h (the same as the input with a single input channel), assuming the kernel takes a stride of one pixel at a time.

**Strides:**

It is common to use a stride two convolution rather than a stride one convolution, where the convolutional kernel strides over 2 pixels at a time, for example our 3x3 kernel would start at position (1, 1), then stride to (1, 3), then to (1, 5) and so on, halving the size of the output channel/feature map, compared to the convolutional kernel taking strides of one. With padding, the output from an input of width w, height h and depth 3 would be the ceiling of width w/2, height h/2 and depth 1, as the kernel outputs a single summed output from each stride.

For example, with an input of 3x64x64 (say a 64x64 RGB three channel image), one kernel taking strides of two with padding the edge pixels, would produce a channel/feature map of 32x32.

The first step of creating and training a new convolutional neural network (Convnet) is to define the network architecture. This topic explains the details of Convent layers, and the order they appear in a ConvNet. For a complete list of deep learning layers and how to create them, see List of Deep Learning Layers. To learn about LSTM networks for sequence classification and regression, see Long Short-Term Memory Networks. To learn how to create your own custom layers, see Define Custom Deep Learning Layers. The network architecture can vary depending on the types and numbers of layers included.

**Image Input Layer:**

Create an image input layer using image input layer. An image input layer inputs images to a network and applies data normalization. Specify the image size using the input Size argument. The size of an image corresponds to the height, width, and the number of color channels of that image. For example, for a grayscale image, the number of channels is 1, and for a color image it is 3.

**Convolutional layer:**

A 2-D convolutional layer applies sliding convolutional filters to the input. Create a 2-D convolutional layer using convolution2dLayer. The convolutional layer consists of various components. Filters and Stride a convolutional layer consist of neurons that connect to sub regions of the input images or the outputs of the previous layer. The layer learns the features localized by these regions while scanning through an image. When creating a layer using the convolution2dLayer function, you can specify the size of these regions using the filter Size input argument.

**Dilated Convolution:**

A dilated convolution is a convolution in which the filters are expanded by spaces inserted between the elements of the filter. Specify the dilation factor using the 'Dilation Factor' property.Use dilated convolutions to increase the receptive field (the area of the input which the layer can see) of the layer without increasing the number of parameters or computation.The layer expands the filters by inserting zeros between each filter element. The dilation factor determines the step size for sampling the input or equivalently the up-sampling factor of the filter. It corresponds to an effective filter size of (Filter Size – 1). \* Dilation Factor + 1. For example, a 3-by-3 filter with the dilation factor [2 2] is equivalent to a 5-by-5 filter with zeros between the elements.This image shows a 3-by-3 filter dilated by a factor of two scanning through the input. The lower map represents the input and the upper map represents the output

**Feature Maps**

As a filter moves along the input, it uses the same set of weights and the same bias for the convolution, forming a *feature map*. Each feature map is the result of a convolution using a different set of weights and a different bias. Hence, the number of feature maps is equal to the number of filters. The total number of parameters in a convolutional layer is ((*h*\**w*\**c* + 1)\**Number of Filters*), where 1 is the bias.

**Zero Padding**

You can also apply zero padding to input image borders vertically and horizontally using the 'Padding' name-value pair argument. Padding is rows or columns of zeros added to the borders of an image input. By adjusting the padding, you can control the output size of the layer. This image shows a 3-by-3 filter scanning through the input with padding of size 1. The lower map represents the input and the upper map represents the output.

**Rectified Linear Unit (ReLU):**

A Rectified Linear Unit is used as a non-linear activation function. A ReLU says if the value is less than zero, round it up to zero. Create a ReLU layer using reluLayer. A ReLU layer performs a threshold operation to each element of the input, where any value less than zero is set to zero. Convolutional and batch normalization layers are usually followed by a nonlinear activation function such as a rectified linear unit (ReLU), specified by a ReLU layer. A ReLU layer performs a threshold operation to each element, where any input value less than zero is set to zero, that is, The ReLU layer does not change the size of its input. There are other nonlinear activation layers that perform different operations and can improve the network accuracy for some applications. For a list of activation layers, see Activation Layers.

**Batch normalization layer:** Batch normalization has the benefits of helping to make a network output more stable predictions, reduce over fitting through regularization and speeds up training by an order of magnitude.Batch normalization is the process of carrying normalization within the scope activation layer of the current batch, subtracting the mean of the batch’s activations and dividing by the standard deviation of the batch’s activations. Create a batch normalization layer using batch Normalization Layer. A batch normalization layer normalizes each input channel across a mini-batch. To speed up training of convolutional neural networks and reduce the sensitivity to network initialization, use batch normalization layers between convolutional layers and nonlinearities, such as ReLU layers.

The layer first normalizes the activations of each channel by subtracting the mini-batch mean and dividing by the mini-batch standard deviation. Then, the layer shifts the input by a learnable offset β and scales it by a learnable scale factor γ. β and γ are themselves learnable parameters that are updated during network training. Batch normalization layers normalize the activations and gradients propagating through a neural network, making network training an easier optimization problem.

To take full advantage of this fact, you can try increasing the learning rate. Since the optimization problem is easier, the parameter updates can be larger and the network can learn faster. You can also try reducing the L2 and dropout regularization. With batch normalization layers, the activations of a specific image during training depend on which images happen to appear in the same mini-batch. To take full advantage of this regularizing effect, try shuffling the training data before every training epoch. To specify how often to shuffle the data during training, use the 'Shuffle' name-value pair argument of training Options.

**Max and Average Pooling Layers:**

A max pooling layer performs down-sampling by dividing the input into rectangular pooling regions, and computing the maximum of each region. Create a max pooling layer using maxPooling2dLayer.An average pooling layer performs down-sampling by dividing the input into rectangular pooling regions and computing the average values of each region. Create an average pooling layer using averagePooling2dLayer.Pooling layers follow the convolutional layers for down-sampling, hence, reducing the number of connections to the following layers. They do not perform any learning themselves, but reduce the number of parameters to be learned in the following layers. They also help reduce over fitting.

A max pooling layer returns the maximum values of rectangular regions of its input. The size of the rectangular regions is determined by the pool Size argument of maxPoolingLayer. For example, if pool Size equals [2, 3], then the layer returns the maximum value in regions of height 2 and width 3. An average pooling layer outputs the average values of rectangular regions of its input. The size of the rectangular regions is determined by the pool Size argument of averagePoolingLayer. For example, if pool Size is [2, 3], then the layer returns the average value of regions of height 2 and width 3.

Pooling layers scan through the input horizontally and vertically in step sizes you can specify using the 'Stride' name-value pair argument. If the pool size is smaller than or equal to the stride, then the pooling regions do not overlap. For non-overlapping regions (Pool Size and Stride are equal), if the input to the pooling layer is n-by-n, and the pooling region size is h-by-h, then the pooling layer down-samples the regions by h. That is, the output of a max or average pooling layer for one channel of a convolutional layer is n/h-by-n/h. For overlapping regions, the output of a pooling layer is (Input Size – Pool Size + 2\*Padding)/Stride + 1.

**Dropout Layer**

Create a dropout layer using dropout layer. A dropout layer randomly sets input elements to zero with a given probability. At training time, the layer randomly sets input elements to zero given by the dropout mask rand (size(X)) <Probability, where X is the layer input and then scales the remaining elements by 1/ (1-Probability). This operation effectively changes the underlying network architecture between iterations and helps prevent the network from over fitting. A higher number results in more elements being dropped during training. At prediction time, the output of the layer is equal to its input. Similar to max or average pooling layers, no learning takes place in this layer

**Fully Connected Layer:**

Create a fully connected layer using fully connected layer. A fully connected layer multiplies the input by a weight matrix and then adds a bias vector. The convolutional (and down-sampling) layers are followed by one or more fully connected layers. As the name suggests, all neurons in a fully connected layer connect to all the neurons in the previous layer. This layer combines all of the features (local information) learned by the previous layers across the image to identify the larger patterns. For classification problems, the last fully connected layer combines the features to classify the images. This is the reason that the output Size argument of the last fully connected layer of the network is equal to the number of classes of the data set. For regression problems, the output size must be equal to the number of response variables.

You can also adjust the learning rate and the regularization parameters for this layer using the related name-value pair arguments when creating the fully connected layer. If you choose not to adjust them, then train Network uses the global training parameters defined by the training Options function. For details on global and layer training options, and train the neural network. A fully connected layer multiplies the input by a weight matrix *W* and then adds a bias vector *b*. If the input to the layer is a sequence (for example, in an LSTM network), then the fully connected layer acts independently on each time step. For example, if the layer before the fully connected layer outputs an array *X* of size *D*-by-*N*-by-*S*, then the fully connected layer outputs an array *Z* of size output Size-by-*N*-by-*S*. At time step *t*, the corresponding entry of *Z* is, where denotes time step *t* of *X*.

**Output Layers:**

**Softmax and Classification Layers:** A softmax layer applies a softmax function to the input. Create a softmax layer using softmax layer. A classification layer computes the cross-entropy loss for multi-class classification problems with mutually exclusive classes.

Create a classification layer using classification Layer. For classification problems, a softmax layer and then a classification layer must follow the final fully connected layer. The softmax function is also known as the normalized exponential and can be considered the multi-class generalization of the logistic sigmoid function. For typical classification networks, the classification layer must follow the softmax layer. In the classification layer, train Network takes the values from the softmax function.

**Regression Layer:**

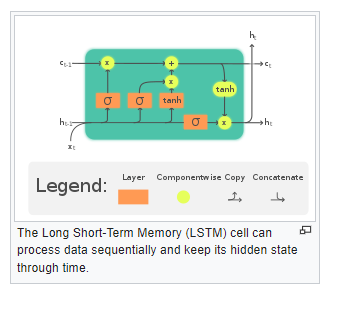
Create a regression layer using the regression layer. A regression layer computes the half-mean-squared-error loss for regression problems. For typical regression problems, a regression layer must follow the final fully connected layer.For a single observation, the mean-squared-error is given by:where *R* is the number of responses, *ti* is the target output, and *yi* is the network’s prediction for response *i*.For image and sequence-to-one regression networks, the loss function of the regression layer is the half-mean-squared-error of the predicted responses, not normalized by *R*:For image-to-image regression networks, the loss function of the regression layer is the half-mean-squared-error of the predicted responses for each pixel, not normalized by *R*.

**Long Short Term Memory:**

**Long short-term memory (LSTM)** is an artificial [recurrent neural network](https://en.wikipedia.org/wiki/Recurrent_neural_network) (RNN) architecture used in the field of [deep learning](https://en.wikipedia.org/wiki/Deep_learning). Unlike standard [feed forward neural networks](https://en.wikipedia.org/wiki/Feedforward_neural_network), LSTM has feedback connections. It can process not only single data points (such as images), but also entire sequences of data (such as speech or video). For example, LSTM is applicable to tasks such as unsegmented, connected [handwriting recognition](https://en.wikipedia.org/wiki/Handwriting_recognition), [speech recognition](https://en.wikipedia.org/wiki/Speech_recognition) and anomaly detection in network traffic or IDSs (intrusion detection systems).

A common LSTM unit is composed of a cell, an input gate, an output gate and a forget gate. The cell remembers values over arbitrary time intervals and the three gates regulate the flow of information into and out of the cell.

LSTM networks are well-suited to [classifying](https://en.wikipedia.org/wiki/Classification_in_machine_learning), [processing](https://en.wikipedia.org/wiki/Computer_data_processing) and [making predictions](https://en.wikipedia.org/wiki/Predict) based on [time series](https://en.wikipedia.org/wiki/Time_series) data, since there can be lags of unknown duration between important events in a time series. LSTMs were developed to deal with the [vanishing gradient problem](https://en.wikipedia.org/wiki/Vanishing_gradient_problem) that can be encountered when training traditional RNNs. Relative insensitivity to gap length is an advantage of LSTM over RNNs, [hidden Markov models](https://en.wikipedia.org/wiki/Hidden_Markov_models) and other sequence learning methods in numerous applications.



In theory, classic (or "vanilla") [RNNs](https://en.wikipedia.org/wiki/Recurrent_neural_network) can keep track of arbitrary long-term dependencies in the input sequences. The problem with vanilla RNNs is computational (or practical) in nature: when training a vanilla RNN using [back-propagation](https://en.wikipedia.org/wiki/Back-propagation), the long-term gradients which are back-propagated can ["vanish"](https://en.wikipedia.org/wiki/Vanishing_gradient_problem) (that is, they can tend to zero) or "explode" (that is, they can tend to infinity), because of the computations involved in the process, which use [finite-precision numbers](https://en.wikipedia.org/wiki/Round-off_error). RNNs using LSTM units partially solve the [vanishing gradient problem](https://en.wikipedia.org/wiki/Vanishing_gradient_problem), because LSTM units allow gradients to also flow unchanged. However, LSTM networks can still suffer from the exploding gradient problem.

**CHAPTER 5**

**MATLAB**

**INTRODUCTION TO MATLAB**

**What Is MATLAB?**

MATLAB is an elite dialect for specialized registering. It incorporates calculation, representation, and programming in a simple to-utilize condition where issues and arrangements are communicated in natural numerical documentation. Run of the mill utilizes incorporate

• Math and calculation

• Algorithm advancement

• Data obtaining

• Modeling, reenactment, and prototyping

• Data examination, investigation, and representation

• Scientific and designing illustrations

• Application advancement, including graphical UI building

MATLAB is an intuitive framework whose important records thing is an exhibit that doesn't require dimensioning. This allows you to address several specialized processing issues, specifically people with framework and vector data, in a small quantity of the time it might take to compose a program in a scalar non intuitive dialect, as an instance, C or FORTRAN.

The name MATLAB stays for grid studies facility. MATLAB modified into to start with composed to offer simple access to framework programming created thru the LINPACK and EISPACK ventures. Today, MATLAB vehicles fuse the LAPACK and BLAS libraries, putting the cutting element in programming for network calculation.

MATLAB has advanced over a time of years with contribution from several customers. In college situations, it's miles the same old educational system for early on and propelled courses in arithmetic, designing, and technological know-how. In industry, MATLAB is the tool of selection for high-profitability research, development, and exam.

MATLAB highlights a collection of extra utility-precise preparations called device booths. Important to most clients of MATLAB, tool kits allow you to examine and practice particular innovation. Tool compartments are exhaustive accumulations of MATLAB capacities (M-records) that extend out the MATLAB situation to take care of precise instructions of problems. Territories in which device stash are reachable incorporate flag dealing with, manage frameworks, neural systems, fluffy purpose, wavelets, recreation, and several others.

**The MATLAB System:**

The MATLAB machine includes five predominant parts.

**Development Environment:**

This is the set of tools and facilities that assist you use MATLAB functions and files. Many of those system are graphical consumer interfaces. It includes the MATLAB computing device and Command Window, a command history, an editor and debugger, and browsers for viewing help, the workspace, files, and the hunt direction.

**The MATLAB Mathematical Function:**

This is a wonderful collection of computational algorithms starting from essential features like sum, sine, cosine, and complex mathematics, to extra sophisticated features like matrix inverse, matrix eigen values, Bessel competencies, and fast Fourier transforms.

**The MATLAB Language:**

This is a excessive-degree matrix/array language with manipulate go with the go with the flow statements, competencies, statistics systems, input/output, and object-oriented programming talents. It permits every programming in the small to hastily create quick and dirty throw-away programs, and programming in the big to create entire massive and complicated software applications.

**Graphics:**

MATLAB has huge facilities for showing vectors and matrices as graphs, further to annotating and printing those graphs. It consists of immoderate-degree capabilities for two-dimensional and three-dimensional facts visualization, image processing, animation, and presentation images. It additionally includes low-stage functions that allow you to completely customize the appearance of image graphs further to assemble whole graphical consumer interfaces on your MATLAB packages.

**The MATLAB Application Program Interface (API):**

his is a library that lets in you to write C and Fortran packages that engage with MATLAB. It includes centers for calling workout routines from MATLAB (dynamic linking), calling MATLAB as a computational engine, and for analyzing and writing MAT-documents.

**5.2 MATLAB WORKING ENVIRONMENT:**

## MATLAB DESKTOP:

Matlab Desktop is the precept Matlab utility window. The computing device consists of five sub windows, the summon window, the workspace software, the prevailing catalog window, the order records window, and at the least one figure home windows, which are confirmed simply whilst the patron indicates a realistic.

The order window is the vicinity the consumer types MATLAB orders and expressions on the provoke (>>) and where the yield of these costs is shown. MATLAB characterizes the workspace because the association of factors that the purchaser makes in a work consultation. The workspace application demonstrates these elements and a few information approximately them. Double tapping on a variable in the workspace software dispatches the Array Editor, which can be applied to get statistics and wage instances regulate certain homes of the variable.

The gift Directory tab over the workspace tab demonstrates the substance of the prevailing registry, whose way is seemed within the gift index window. 1For case, in the home windows working framework the manner may be as consistent with the subsequent: C:MATLABWork, demonstrating that registry work is a subdirectory of the number one catalog MATLAB, that's added in force C. Tapping on the bolt inside the gift index window demonstrates a rundown of as of late utilized approaches. Tapping at the capture to at least one aspect of the window allows the customer to exchange the prevailing catalog.

MATLAB utilizes an inquiry way to discover M-information and other MATLAB associated documents, which might be kind out in catalogs within the PC report framework. Any report hold going for walks in MATLAB must reside in the ebb and go with the flow registry or in an index this is on are looking for manner. Of course, the facts provided with MATLAB and math works device kits are integrated into the inquiry manner. The least traumatic approach to look which indexes are at the inquiry manner. The best method to peer which catalogs are quickly the search way, or to consist of or adjust an inquiry manner, is to pick out set way from the File menu the computing device, and after that utilization the set way change container. It is extremely good exercise to feature any normally applied catalogs to the pursuit manner to preserve a strategic distance from again and again having the exchange the existing index.

The Command History Window incorporates a file of the orders a customer has entered in the rate window, together with each gift and beyond MATLAB classes. Already entered MATLAB orders may be selected and re-done from the fee history window by using right tapping on a sum on or association of orders. This interest dispatches a menu from which to choose exclusive picks however executing the orders. This is helpful to pick special choices notwithstanding executing the summons. This is a precious element even as attempting different things with distinct orders in a work consultation.

**Using the MATLAB Editor to create M-Files:** The MATLAB manager is both a word processor specific for making M-information and a graphical MATLAB debugger. The proofreader can display up in a window without everybody else, or it could be a sub window inside the desktop. M-statistics are supposed by means of the growth .M, as in pixelup.m. The MATLAB editorial manager window has diverse draw down menus for errands, as an example, sparing, seeing, and troubleshooting files. Since it plays out some primary tests and furthermore makes use of shading to separate among exceptional components of code, this content material device is suggested because the apparatus of choice for composing and changing M-capacities. To open the proofreader, sort adjust at the incite opens the M-report filename. In a supervisor window, organized for changing. As referred to before, the file has to be within the momentum catalog, or in an index inside the pursuit way.

**Getting Help:**

The important approach to get help online is to utilize the MATLAB help program, opened as a different window either by tapping on the question mark image (?) on the desktop toolbar, or by writing help program at the provoke in the order window. The assistance Browser is a web program coordinated into the MATLAB desktop that shows a Hypertext Markup Language (HTML) records. The Help Browser comprises of two sheets, the assistance pilot sheet, used to discover data, and the show sheet, used to see the data.

**CHAPTER 6**

**COMMUNICATION**

Communications System Toolbox™ offers algorithms and gear for the layout, simulation, and analysis of communications systems. These capabilities are furnished as MATLAB ® features, MATLAB System gadgets™, and Simulink ® blocks. The machine toolbox includes algorithms for source coding, channel coding, interleaving, modulation, equalization, synchronization, and channel modeling. Tools are supplied for bit blunders charge evaluation, producing eye and constellation diagrams, and visualizing channel characteristics. The machine toolbox additionally

provides adaptive algorithms that allow you to version dynamic communications structures that use OFDM, OFDMA, and MIMO techniques. Algorithms support fixed-point facts arithmetic and C or HDL code era.

Key Features

▪ Algorithms for designing the physical layer of communications systems, which includes supply coding, channel coding, interleaving, modulation, channel fashions, MIMO, equalization, and synchronization

▪ GPU-enabled System objects for computationally intensive algorithms together with Turbo, LDPC, and Viterbi decoders

▪ Interactive visualization equipment, consisting of eye diagrams, constellations, and channel scattering capabilities

▪ Graphical tool for evaluating the simulated bit mistakes rate of a machine with analytical outcomes

▪ Channel models, consisting of AWGN, Multipath Rayleigh Fading, Rician Fading, MIMO Multipath Fading, and

LTE MIMO Multipath Fading

▪ Basic RF impairments, along with nonlinearity, section noise, thermal noise, and section and frequency offsets

▪ Algorithms available as MATLAB features, MATLAB System objects, and Simulink blocks

▪ Support for fixed-point modeling and C and HDL code technology

System Design, Characterization, and Visualization

The layout and simulation of a communications gadget requires analyzing its reaction to the noise and interference inherent in real-world environments, reading its behavior the usage of graphical and quantitative manner, and determining whether the resulting overall performance meets requirements of acceptability. Communications System Toolbox implements a selection of obligations for communications machine layout and simulation. Many of the functions, System objects™, and blocks inside the device toolbox perform computations associated with a specific thing of a communications gadget, consisting of a demodulator or equalizer. Other talents are designed for visualization or evaluation.

System Characterization

The system toolbox offers several standard methods for quantitatively characterizing system performance:

▪ Bit error rate (BER) computations

▪ Adjacent channel power ratio (ACPR) measurements

▪ Error vector magnitude (EVM) measurements

▪ Modulation error ratio (MER) measurements

Because BER computations are fundamental to the characterization of any communications system, the system toolbox provides the following tools and capabilities for configuring BER test scenarios and accelerating BER simulations:

BERtool — A graphical user interface that enables you to analyze BER performance of communications systems. You can analyze performance via a simulation-based, semi analytic, or theoretical approach.

Error Rate Test Console — A MATLAB object that runs simulations for communications systems to measure error rate performance. It supports user-specified test points and generation of parametric performance plots and surfaces. Accelerated performance can be realized when running on a multicore computing platform.

Multicore and GPU acceleration — A capability provided by Parallel Computing Toolbox™ that enables you to accelerate simulation performance using multicore and GPU hardware within your computer.

Distributed computing and cloud computing support — Capabilities provided by Parallel Computing Toolbox and MATLAB Distributed Computing Server™ that enable you to leverage the computing power of your server farms and the Amazon EC2 Web service. Performance Visualization The system toolbox provides the following capabilities for visualizing system performance:

Channel visualization tool — For visualizing the characteristics of a fading channel

Eye diagrams and signal constellation scatter plots — For a qualitative, visual understanding of system behavior that enables you to make initial design decisions

Signal trajectory plots — For a continuous picture of the signal’s trajectory between decision points

BER plots — For visualizing quantitative BER performance of a design candidate, parameterized by metrics such as SNR and fixed-point word size

Analog and Digital Modulation

Analog and digital modulation strategies encode the facts circulation into a sign this is appropriate for transmission. Communications System Toolbox presents some of modulation and corresponding demodulation abilities. These talents are available as MATLAB features and gadgets, MATLAB System Modulation sorts provided by the toolbox are:

Analog, including AM, FM, PM, SSB, and DSBSC

Digital, including FSK, PSK, BPSK, DPSK, OQPSK, MSK, PAM, QAM, and TCM



Source and Channel Coding

Communications System Toolbox affords source and channel coding talents that can help you develop and compare communications architectures fast, enabling you to discover what-if eventualities and avoid the need to create coding competencies from scratch.

Source Coding

Source coding, also referred to as quantization or signal formatting, is a manner of processing facts a good way to lessen redundancy or prepare it for later processing. The system toolbox offers a diffusion of styles of algorithms for imposing source coding and interpreting, inclusive of:

▪ Quantizing

▪ Commanding (µ-law and A-law)

▪ Differential pulse code modulation (DPCM)

▪ Huffman coding

▪ Arithmetic coding

Channel Coding

▪ Orthogonal area-time block code (OSTBC) (encoder and decoder for MIMO channels)

▪ Turbo encoder and decoder examples

The gadget toolbox offers application functions for developing your personal channel coding. You can create generator polynomials and coefficients and syndrome deciphering tables, in addition to product parity-take a look at and generator matrices.

The system toolbox additionally presents block and convolutional interleaving and deinterleaving functions to reduce facts errors as a result of burst mistakes in a conversation machine:

Block, including General block interleave, algebraic interleave, helical scan interleaves, matrix interleave, and random interleave

Convolutional, including General multiplexed interleaved, convolutional interleaver, and helical interleaver

Channel Modeling and RF Impairments

Channel Modeling

Communications System Toolbox provides algorithms and tools for modeling noise, fading, interference, and different distortions which might be commonly found in communications channels. The system toolbox supports the subsequent styles of channels:

▪ Additive white Gaussian noise (AWGN)

▪ Multiple-enter multiple-output (MIMO) fading

▪ Single-enter single-output (SISO), Rayleigh, and Rician fading

▪ Binary symmetric

A MATLAB channel object provides a concise, configurable implementation of channel models, enabling you to

specify parameters such as:

▪ Path delays

▪ Average path gains

▪ Maximum Doppler shifts

▪ K-Factor for Rician fading channels

▪ Doppler spectrum parameters

For MIMO systems, the MATLAB MIMO channel object expands these parameters to also include:

▪ Number of transmit antennas (up to 8)

▪ Number of receive antennas (up to 8)

▪ Transmit correlation matrix

▪ Receive correlation matrix

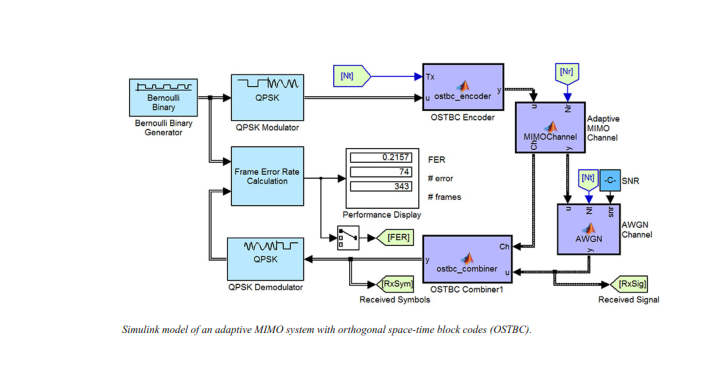
To combat the effects noise and channel corruption, the system toolbox provides block and convolutional coding and decoding techniques to implement error detection and correction. For simple error detection with no inherent correction, a cyclic redundancy check capability is also available. Channel coding capabilities provided by the system toolbox include:

▪ BCH encoder and decoder

▪ Reed-Solomon encoder and decoder

▪ LDPC encoder and decoder

▪ Convolutional encoder and Viterbi decoder



RF Impairments

To model the effects of a nonideal RF front end, you can introduce the following impairments into your communications system, enabling you to explore and characterize performance with real-world effects:

▪ Memoryless nonlinearity

▪ Phase and frequency offset

▪ Phase noise

▪ Thermal noise

You can include more complex RF impairments and RF circuit models in your design using SimRF™.



Equalization and Synchronization

Communications System Toolbox lets you discover equalization and synchronization strategies. These techniques are usually adaptive in nature and tough to design and symbolize. The machine toolbox affords algorithms and tools that will let you swiftly select the proper approach on your communications machine. Equalization To compare one-of-a-kind techniques to equalization, the device toolbox offers you with adaptive algorithms which include:

▪ LMS

▪ Normalized LMS

▪ Variable step LMS

▪ Signed LMS

▪ MLSE (Viterbi)

▪ RLS

▪ CMA

These adaptive equalizers are available as nonlinear decision feedback equalizer (DFE) implementations and as

linear (symbol or fractionally spaced) equalizer implementations.

Synchronization

The device toolbox provides algorithms for each service segment synchronization and timing phase synchronization. For timing section synchronization, the machine toolbox presents a MATLAB Timing Phase Synchronizer object that offers the following implementation techniques:

▪ Early-late gate timing method

▪ Gardner’s method

▪ Fourth-order nonlinearity method

Stream Processing in MATLAB and Simulink

Most verbal exchange structures cope with streaming and frame-primarily based statistics using a aggregate of temporal processing and simultaneous multi frequency and multichannel processing. This form of streaming multidimensional processing can be visible in superior communication architectures consisting of OFDM and MIMO. Communications System Toolbox enables the simulation of advanced communications structures via helping move processing and frame-based simulation in MATLAB and Simulink. In MATLAB, circulate processing is enabled by way of System items™, which use MATLAB objects to symbolize time-based and facts-driven algorithms, sources, and sinks. System objects implicitly manipulate many information of flow processing, including information indexing, buffering, and management of set of rules state. You can mix System gadgets with fashionable MATLAB functions and operators. Most System items have a corresponding Simulink block with the identical abilties. Simulink handles circulation processing implicitly with the aid of coping with the float of information thru the blocks that make up a Simulink model. Simulink is an interactive graphical environment for modeling and simulating dynamic systems that uses hierarchical diagrams to symbolize a machine version. It includes a library of widespread-reason, predefined blocks to represent algorithms, resources, sinks, and device hierarchy.

Implementing a Communications System

Fixed-Point Modeling Many communications systems use hardware that requires a fixed-point representation of your design.

Communications System Toolbox supports fixed-point modeling in all relevant blocks and System objects™ with tools that help you configure fixed-point attributes.

Fixed-point support in the system toolbox includes:

▪ Word sizes from 1 to 128 bits

▪ Arbitrary binary-point placement

▪ Overflow handling methods (wrap or saturation)

▪ Rounding methods: ceiling, convergent, floor, nearest, round, simplest, and zero

Fixed-Point Tool in Simulink Fixed Point™ facilitates the conversion of floating-point data types to fixed point. For configuration of fixed-point properties, the tool tracks overflows and maxima and minima.

Code Generation

Once you've got advanced your set of rules or communications device, you can robotically generate C code from it for verification, rapid prototyping, and implementation. Most System gadgets, functions, and blocks in Communications System Toolbox can generate ANSI/ISO C code the use of MATLAB Coder™, Simulink Coder™, or Embedded Coder™. A subset of System gadgets and Simulink blocks also can generate HDL code. To leverage present highbrow belongings, you can choose optimizations for specific processor architectures and integrate legacy C code with the generated code.

You can also generate C code for both floating-point and f

ixed-point data types.

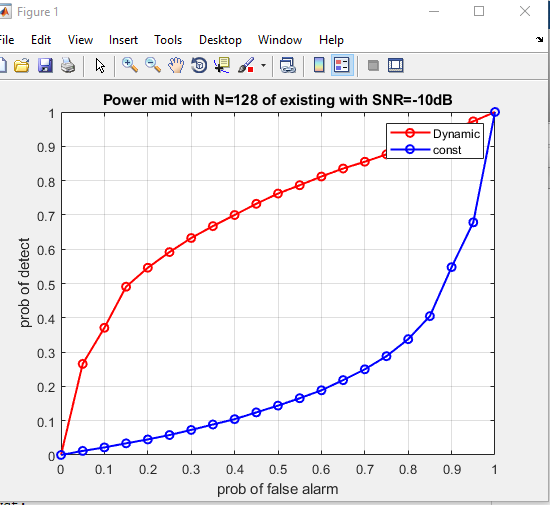
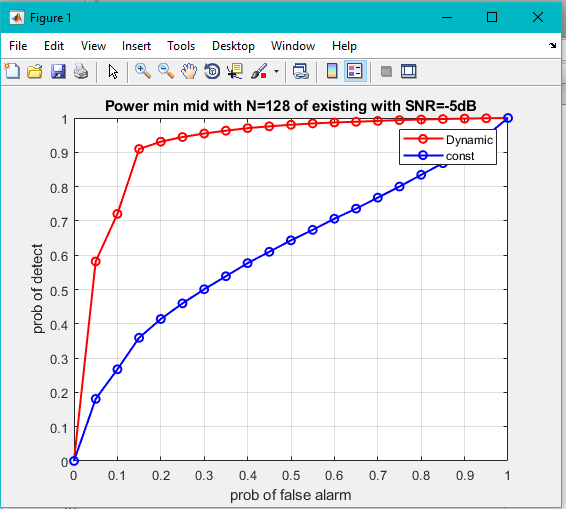
DSP Prototyping’s are used in communication system implementation for verification, rapid prototyping, or final hardware implementation. Using the processor-in-the-loop (PIL) simulation capability found in Embedded Coder, you can verify generated source code and compiled code by running your algorithm’s implementation code on a target processor. FPGA Prototyping

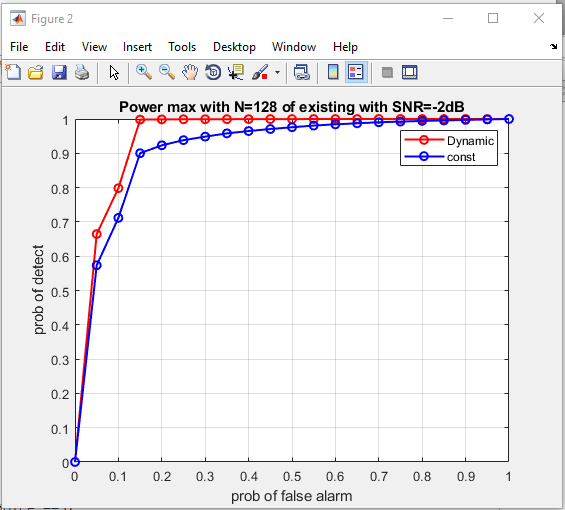
FPGAs are used in communication systems for implementing high-speed signal processing algorithms. Using the FPGA-in-the-loop (FIL) capability found in HDL Verifier™, you can test RTL code in real hardware for any existing HDL code, either manually written or automatically generated HDL code.

**CHAPTER 7**

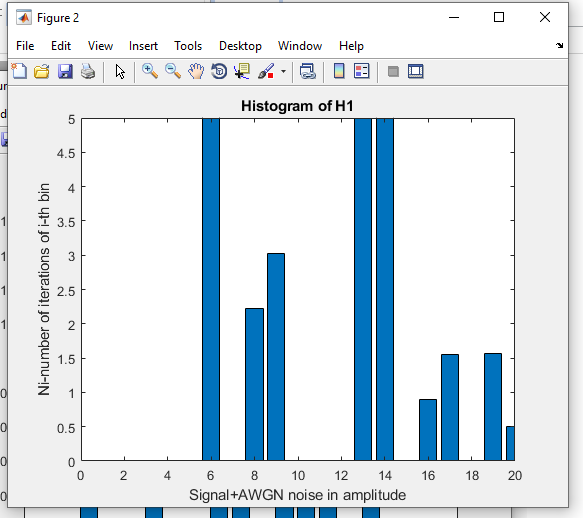
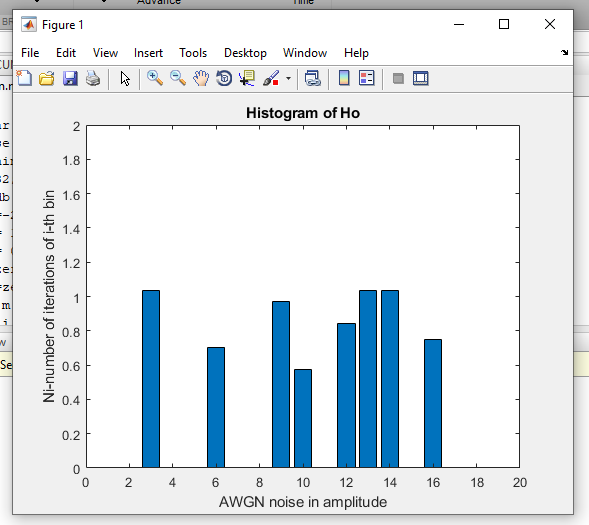
**RESULTS**

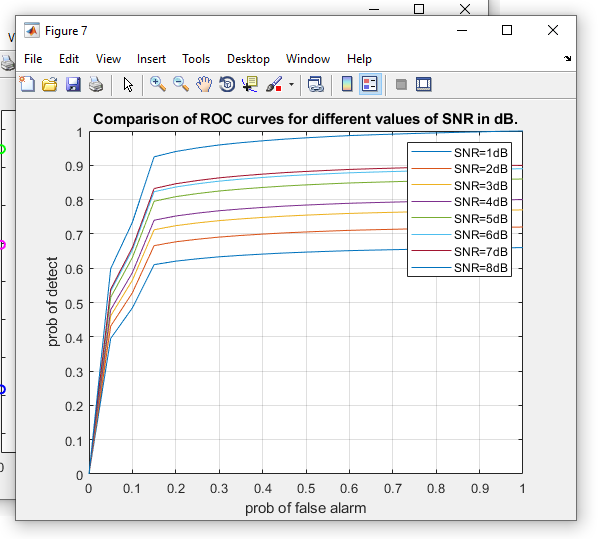
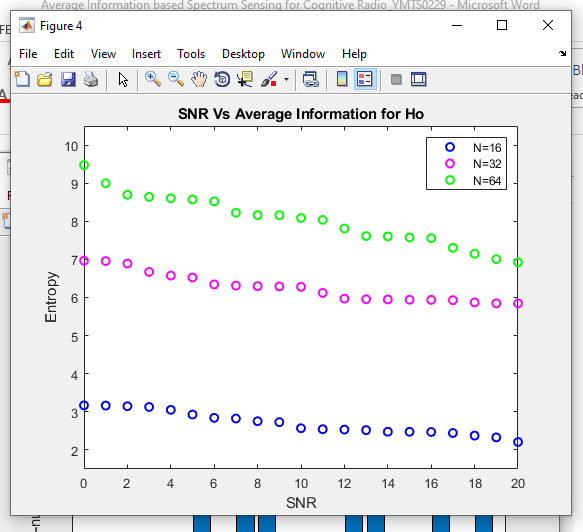
**Method: 1**

** **

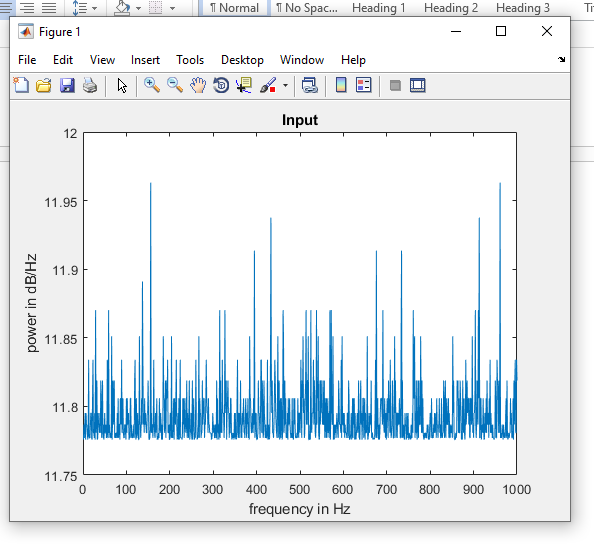
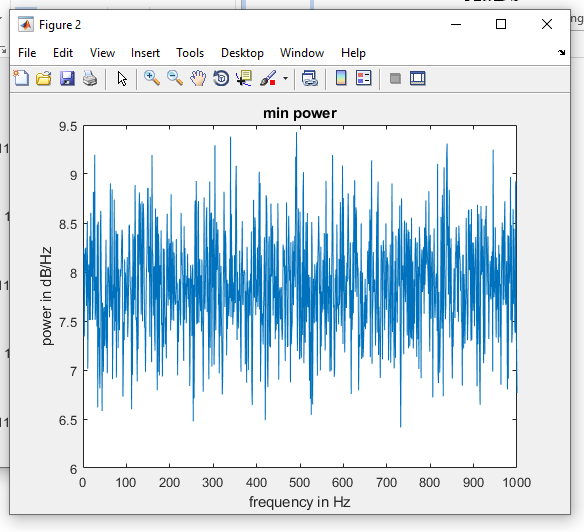
** **

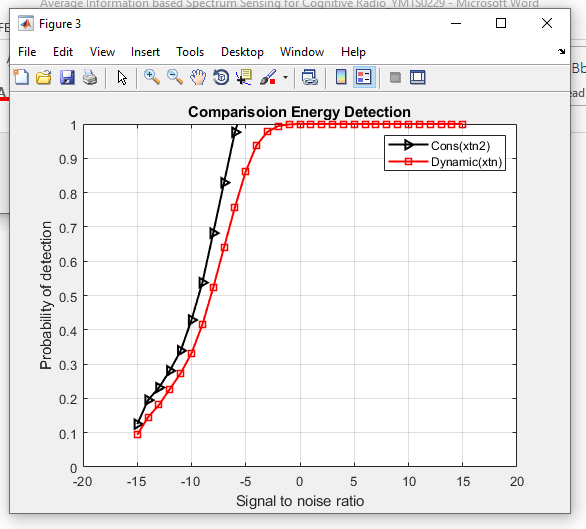
**Method 2:**



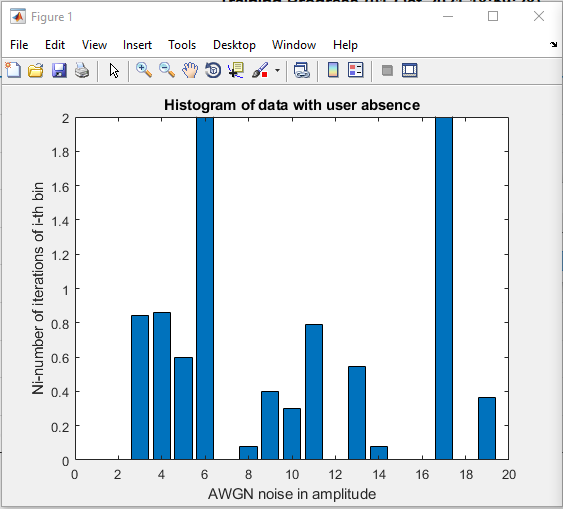
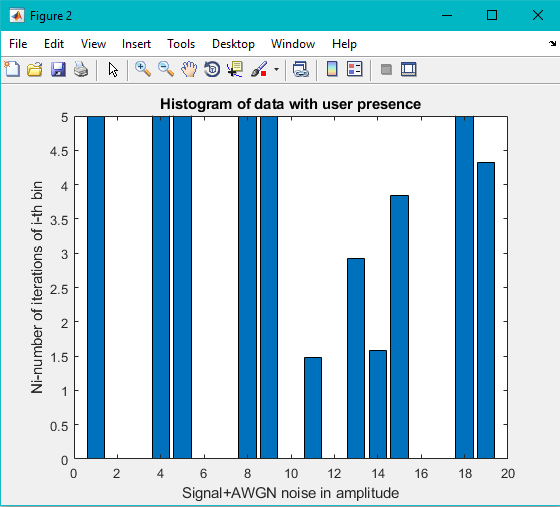


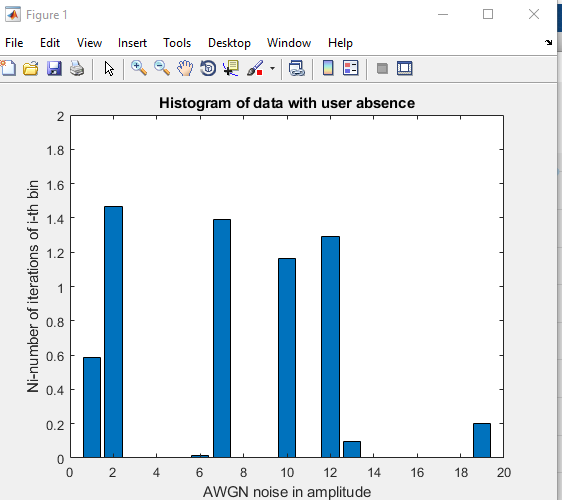
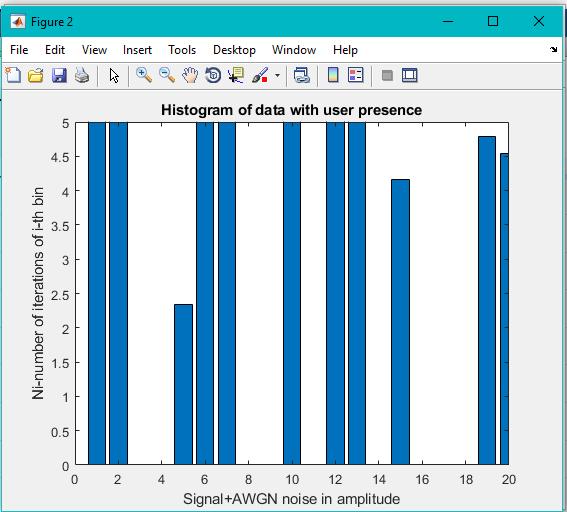
**Method: 3**



**Method: 4**

** **

** **

**CHAPTER 7**

**CONCLUSION**

In this work, the average information-based spectrum sensing is proposed. Histogram is used to find the number of samples in each bin. This method is used to detect the presence of primary users. Presence of primary user can be detected by considering the entropy value. If the entropy is higher, then the primary user is absent. If the entropy value is lower, then the primary user is present. The entropy for 16QAM modulation is calculated and it is found that entropy decreases as SNR increases. Simulations have been done for different signal lengths N=16, 32, 64 for 16QAM.The probability of the detection is significant even from low SNR of -8dB. Thus, the probability of detection is good even at lower SNR for average information-based spectrum sensing technique.

**CHAPTER 8**

**REFERENCES**

[1] R. Tandra and A. Sahai, “Fundamental limits on detection in low SNR under noise uncertainty,” in Proc. Int. Conf. Wireless Netw., Commun. Mobile Comput., vol. 1, Jun. 2005, pp. 464-469.

[2] S.V. Nagaraj, “Entropy-based spectrum sensing in cognitive radio,” Signal Process., vol. 89, no. 2, pp. 174-180, 2009.

[3] D. Bhargavi and C. R. Murthy, “Performance comparison of energy, matched-filter and cyclostationarity-based spectrum sensing,” in Proc.IEEE 11th Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC), Jun. 2010, pp. 1-5.

[4] Y. L. Zhang, Q. Y. Zhang, and T. Melodia, “A frequency-domain entropy-based detector for robust spectrum sensing in cognitive radio networks”, IEEE Commun. Lett., vol. 14, no. 6, pp. 533-535, Jun. 2010.

[5] A. Kortun, T. Ratnarajah, M. Sellathurai, C. Zhong, and C. B. Papadias, “On the performance of eigen value-based cooperative spectrum sensing for cognitive radio,” IEEE J. Sel. Topics Signal Process., vol. 5, no. 1, pp. 49-55, Feb. 2011.

[6] M. Jin, Y. Li, and H.-G. Ryu, “On the performance of cova-riance based spectrum sensing for cognitive radio,” IEEE Trans. Signal Process., vol. 60, no. 7, pp. 3670-3682, Jul. 2012.

[7] N. Zhao, “A novel two-stage entropy-based robust cooperative spectrum sensing scheme with two-bit decision in cognitive radio,” Wireless Personal Dec., vol. 69, no. 4, pp. 1551-1565, 2013. [8] Prieto, Angel G. Andrade,Daniela M. Martinez, and Guillermo Gavaviz, “On the Evaluation of an Entropy-Based Spectrum Sensing Strategy Applied to Cognitive Radio Networks,” IEEE Access, Vol. 6, pp. 64828-64835, Oct. 2018.

[9] Manish Gupta, Guaurav verma and Rahul Kumar Dubey, “Cooprative spectrum sensing for cognitive radio based on adaptive threshold,”IEEE International conference, pp.609-612, Aug. 2016.

[10] B.Suseela, Dr.D.Sivakumar, “Non- cooperative spectrum sensing techniques in cognitive radio- A survey,”IEEE Technological Innovation in ICT for Agriculture and Rural Development(TIAR),pp.127-133, Jul. 2015.