End-to-end OFDM system to reduce BER using Machine Learning Driven approaches

Rishhabh Naik, Rohan Sequeira, Sudharsan Babu Srikanthan, Vibhuti Ravi
rishhabh, rohseque, sudhsrik, vibhutir

May 15, 2022

Abstract

We first implement an end-to end system on GNU radio using two USRP X310 for the transmitter and the receiver in an indoor setting. We then build a comprehensive model using the components individually rather than the built-in OFDM transmitter and receiver. We use two different approaches to incorporate machine learning to enhance aspects of the communication system, in a bid to improve channel estimation, in the presence of multi-path fading and frequency offset. Each of our models offers a different advantage. With the first model, JCES performs joint channel estimation and symbol detection. It offers a performance comparable to that of the MMSE estimator even in the case of no cyclic prefix being used. The Cs-CNN (cascaded CNN model) explicitly performs channel estimation and estimates the channel response. This is crucial as it would allow us to further try and maximize the data transfer while keeping the bit error rate as low as possible. Using this model, we hope to use dynamically updating CP-OFDM in the future. Since, synchronization is such an important aspect, we explore a bare bone implementation.

1 OFDM

In Orthogonal frequency-division multiplexing , the data to be transmitted is split among several several closely spaced narrowband subchannel frequencies instead of a single wideband channel frequency. In a traditional single-channel modulation scheme, each data bit is sent serially or sequentially one after another. In OFDM, several bits can be sent in parallel, or at the same time, in separate substream channels. This enables each substream's data rate to be lower than would be required by a single stream of similar bandwidth. This makes the system less susceptible to interference and enables more efficient data bandwidth. It has recently been applied widely in wireless communication systems due to its high data rate transmission capability with high bandwidth efficiency and its robustness to multipath delay.

The main advantages of OFDM technology are as follows:

- The frequency spectrum usage ratio is high. Because OFDM technology uses orthogonal sub-carriers as sub-carrier channels, not only does the bandwidth of OFDM sub-carriers do not need to be maintained, but the spectrum can be overlapped, using spectrum data to a great extent. Moreover, the sub-carrier in OFDM also selects a multi-system modulation method, which can deepen the ratio of spectrum usage.

- Strong ability to resist multipath interference. OFDM technology implements serial-to-parallel conversion of data symbols waiting to be sent and converts serial information with a higher rate into N columns of parallel information with a lower rate. As a result, the rate of the symbol is reduced, thereby ensuring that the period of the symbol increases.

  Therefore, scientifically weakens the interference caused by the multipath effect and the crosstalk between codes. Also, the OFDM technology points out the addition of the cyclic prefix CP as a way to maintain the interval. As a result, it can not only ensure that the orthogonal characteristics between the channels do not interfere but also can minimize or even eliminate the interference existing between codes.

- Strong ability to resist weakness. For a single carrier transmission system, if weakening or interference occurs, all the transmission links may be invalid. However, the OFDM modulation style is suitable for high-rate information transmission in multipath and weakened channels. If fading or interference occurs, only the carrier and the data it carries that fall in the frequency recess will have interfered, and other sub-carriers will not be destroyed.

- The allocation of resources is more sensitive. In a simple modulation and demodulation process, IDFT and DFT are used in the OFDM system to complete orthogonal modulation and demodulation. For systems with a large number of subcarriers, FFT can be used to complete. Because digital signal processing (DSP, Digital Signal Processing) technology and digital integrated circuit (CPLD, Complex PLD) technology are developed at a very fast speed, it is very easy to complete a series of tasks using IFFT and FFT to complete calculations.

- The asymmetric type has a faster-conveying rate. The OFDM system can adjust the transmission speed of the uplink and downlink by selecting a different number of sub-channels and can implement asymmetric transmission of the uplink and downlink speeds. Also, OFDM can place information in a higher rate channel through a loading method, so the OFDM system can complete asymmetric information transmission at a higher rate.

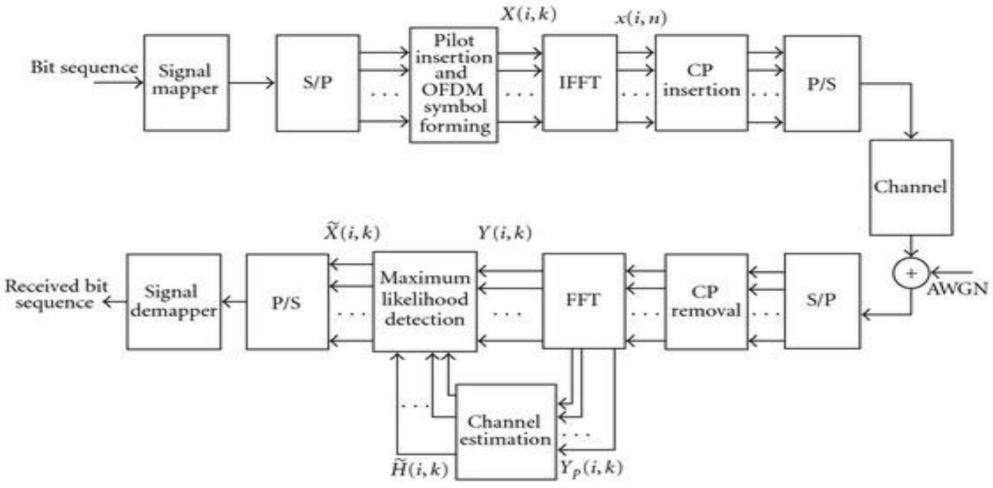

Figure 1: Block Diagram of an OFDM System

Due to these advantages of OFDM technology, it has been widely used in many technical fields. These include, high-speed wireless local area networks (WLAN) radio interfaces IEEE 802.11a, g, n, ac, ah and HIPERLAN/2. OQAM-OFDM technology which has a lower out-of-band spectrum leakage, has become an important part of 5G key technology.

The disadvantages of OFDM system is as follows:

- The complexity of the system is relatively high. In the OFDM system, each sub-channel determines its transmission power and data rate, so to effectively implement the allocation of signal power and information, the loading algorithm is introduced. However, this complicates the transceiver equipment. If the terminal moves at a very high speed, it will not be suitable for adaptive modulation technology.

- The frequency is extremely sensitive. The OFDM system is sensitive to noise and carrier frequency deviation and is easily interfered with by frequency errors. Because the frequency of the sub-channels that cover each other changes with time, the orthogonal characteristics of the sub-carriers get damaged, and the frequency spectrum has deviates, causing interference between sub-channels.

- The crest factor is too high. Since the derived signal of the OFDM system is superimposed by several sub-channels and consists of several separately modulated sub-carriers, the accumulation of the same phase of multiple signals will also cause the composite signal to have an extremely high peak power. The instantaneous power of the system as a whole will be far greater than the average power of the system, so a very large Peak to Average Power Ratio (PAPR) will be generated, which will reduce the power transmission efficiency. We use clipping to keep PAPR under good control in the Machine Learning model described in Section 4.

The OFDM system is implemented by combining the different blocks as shown in Fig. 1. The input bits are grouped and mapped to data symbols that are complex numbers representing the modulation constellation point (e.g., the BPSK or QAM symbols that would be present in a single sub-carrier system). These data symbols and a set of pre-defined pilot symbols (used for channel estimation) are loaded onto the sub-carriers. In general, there are two basic preparation of the pilot symbols:

- Block-type pilot (as shown in Figure 2 [left]), by including a pilot symbol into all sub-carrier within a certain time within a specific time period. In addition to the previous recipient

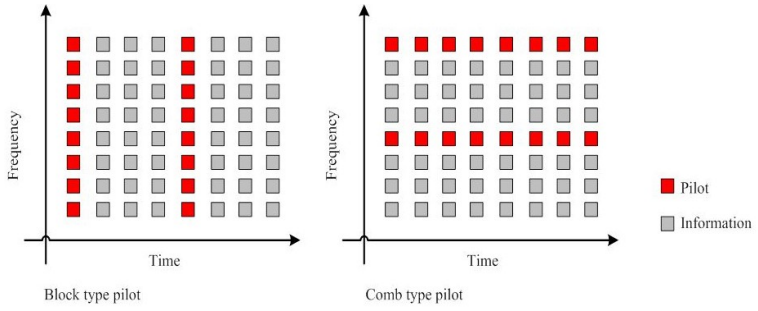

Figure 2: Two Types of Pilot Symbol Arrangement: Block-type Pilot Arrangement, Comb-type Pilot Arrangement.

already knows the value of pilot symbols, the receiver also has to know when the pilot symbols transmitted simultaneously. This type is used for slow fading channels.

- Comb type pilot (as shown in Figure 2 [right]), that is provided a special allocation of frequencies used to transmit pilot symbols every time. The sender determines the sub-carrier which has previously been used to transmit pilot symbols. This type is used for fast fading channels.

Now, the OFDM carriers can be transformed to the time-domain by means of the IDFT operation. Subsequently, we add a cyclic prefix to the symbol. This operation concatenates a copy of the last CP samples of the OFDM time domain signal to the beginning. This way, a cyclic extension is achieved. The CP fulfills two tasks: (i) It isolates different OFDM blocks from each other when the wireless channel contains multiple paths, i.e. is frequency-selective. (ii) Sensitivity to time synchronization is achieved. When the length of the guard interval is longer than the duration of the channel impulse response, ISI can completely be removed. Next, the signal is sent to the antenna and sent over the air to the receiver. At the receiver, the CP is removed from the signal and the signal is transformed back to the frequency domain, in order to have the received value on each sub-carrier available. As the next step, the wireless channel needs to be estimated. This is done using pilot-aided channel estimation techniques explored in Section 3. Once the channel is estimated at all carriers, we can use this information in the channel equalizer step where the influence of the channel is removed. Finally, we obtain the symbols back and send the complex values to the de-mapper, to transform the constellation points back to bit groups.

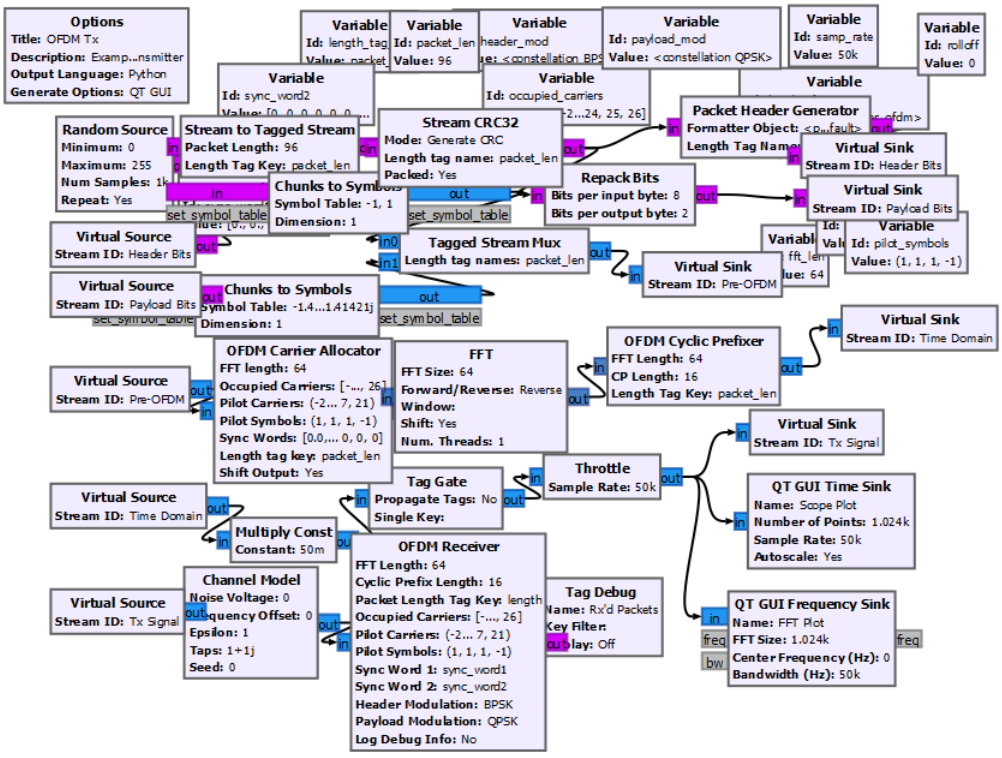

Figure 3: Transmitter Block Diagram

This OFDM block diagram is used as a backbone for our work in the later sections.

2 End to end OFDM system implementation in USRP/ GNU radio

Theoretical simulations show that OFDM provides good error performance and high data rate transmission compared to single carrier techniques. However, channel conditions play a major role in synchronization and channel estimation and can lead to poor performance in terms of BER. Therefore, evaluation of the practical performance of OFDM systems is very important to verify the system's performance in realistic propagation conditions, understand the practical limitations of the hardware and to obtain a deeper understanding of the effects of synchronization and channel estimation techniques to the error performance. This section explains how an OFDM testbed for real time transmission was implemented using two USRPs of model X310 as transmitter and receiver with an open source of GNURadio as a software.

The transmitter side is implemented as shown in Figure 3. The random source block generates 1k samples of binary data that gets transformed into streamed information via the Tagged Stream block. The stream tagged utilizes tags to define the boundaries of packets. The Stream CRC-32

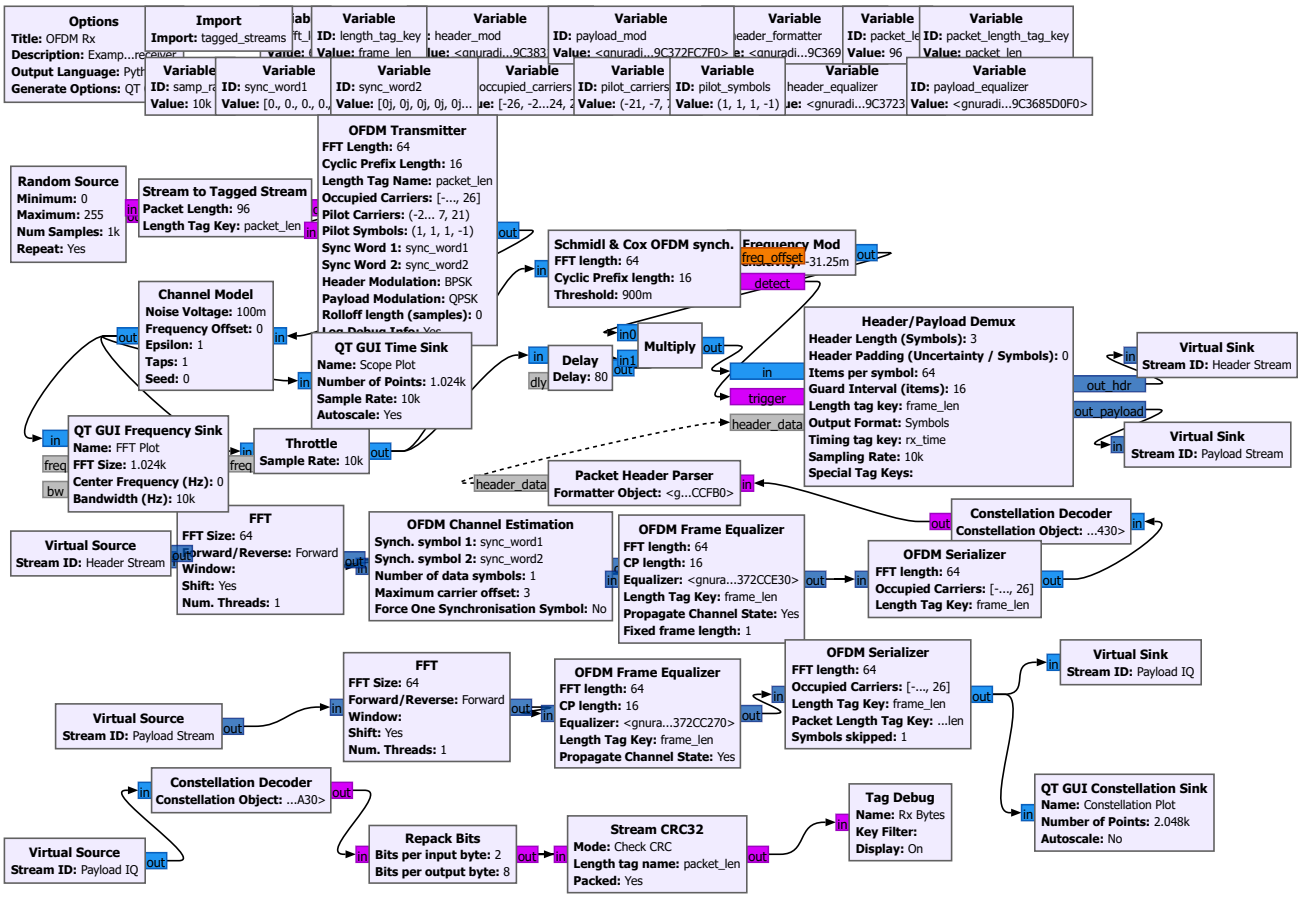

Figure 4: Receiver Block Diagram

block attaches a 32-bit cyclic redundancy (CRC-32) control to the tagged stream. The Packet Header block generates one header per packet. The Repack Bits block converts 8 bits per byte to 2 bits per byte and modulates the payload by Quadrature phase shift keying (QPSK). A block of Virtual Sink relates to the same ID to the Virtual Source block. BPSK modulates the header and QPSK modulates the payload in the block of chunks to symbols. In the Tagged Stream Mux block, the header and payload are concatenated. The OFDM Carrier Allocator block assigns modulated information symbols to the relevant sub-carriers, inserts pilot sub-carriers, and attaches the short preamble and long preamble. The parameters of this block are Occupied Carriers, Pilot Carriers, Pilot Symbols, and Sync Words. The parameter for Occupied Carriers contains the range of subcarriers used. The parameter Pilot Carriers assigns pilot subcarrier IDs and the parameter Pilot Symbols assigns pilot subcarrier modulated symbols. The Sync Words parameter provides the modulated symbol sequences for the short and long preambles which are the same as the IEEE 802.11a standard but has half the length of 802.11a preambles. The FFT block operates in the TX flow-graph as an inverse fast Fourier transform (IFFT). The OFDM Cyclic Prefixer attaches an OFDM symbol with the guard interval (GI). The Multiply Const block multiplies the signal by 50m as the transmit signal should be within the transmitter's dynamic range. The block Tag Gate removes the tags from the recorded information. Additionally, a UHD: USRP Sink block is added that controls USRP and UHD for the transmission of digital radio IQ information.

The receiver side is implemented as shown in Figure 4. A UHD: the Source Block of the USRP is added, which controls the digital IQ on the radio. The symbol and the frequency offset of the RX signal is removed by the Schmidl & Cox OFDM Synch. block. The block Delay delays 1 OFDM symbol in the RX signal. Frequency offset is modulated via the Frequency Mod block and compensated by a Multiplexing block for the late RX signal. The RX is divided into headers and payload streams via the Demux header/payload block. Signal processing is performed in the header stream before the payload stream. The FFT Block is an FFT that matches a demodulator of the OFDM. The estimate block for each sub-carrier with the prevalent lengthy preamble and the

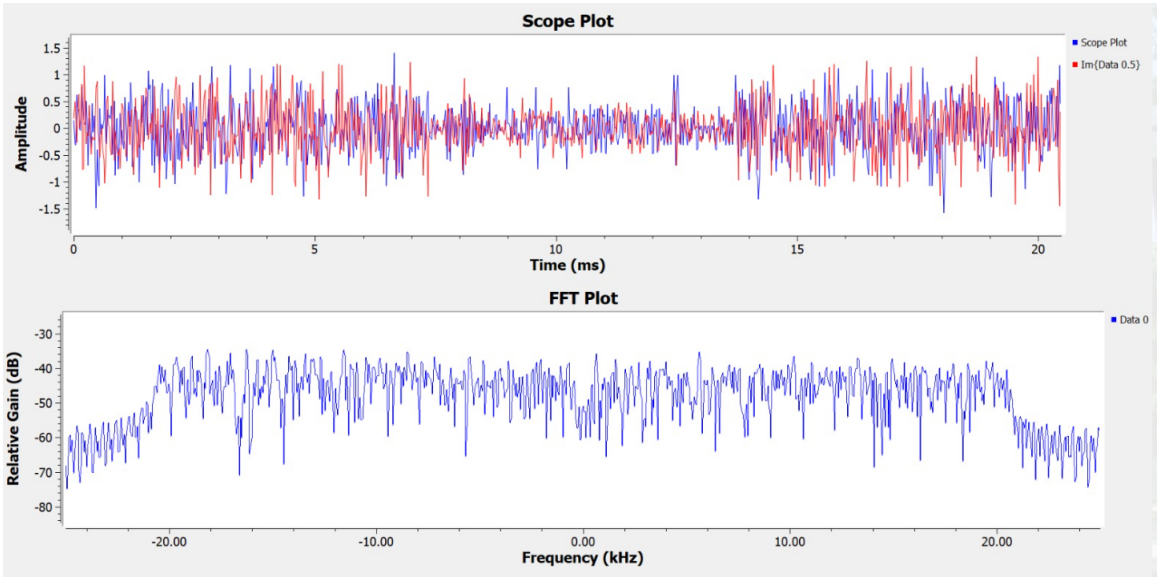

Figure 5: Scope plot and FFT Plot of the received signal

equalizer for the channel state data for the header works with the estimated CSI. OFDM channel estimating block estimates the channel state data (CSI). When the OFDM symbol stream is send to the OFDM frame equalizer, it equalizes the raw frequency offset of the carrier in one or two sizes for an OFDM frame. The OFDM Serializer block is used to remove and convert pilot subcarriers into serial. The Decoder Constellation block demodulates the BPSK header. The Parser Block Packet Header transforms header information as tags into packet information. Except for obtaining estimated CSI from the tags and demodulating payload, the signal processing in payload stream is almost identical to that in the header stream. The Bit repack transforms the byte order bits into 8 bits per byte. Stream CRC-32 transmits RX information if there is no CRC-32 error. The packet sequence number on a console is displayed in the Tag Debug block.

Figure 5 shows the scope and FFT plots obtained at the USRP Reciever. Bursts of OFDM signals can be seen in the scope plot where the lower amplitude part of the signal is the preamble used for synchronization purposes. The higher amplitude part of the signal consists of the OFDM symbols with Cyclic Prefixers. The FFT Plot gives a highest frequency of about 20kHz, as expected, which is less than half of the sampling frequency (50kHz) of the system.

3 Channel

Across both our DL based approaches in the following sections, we modelled our multipath fading channel to train the both the models and test their performance. We chose Rician fading for our channels.

1. An LOS component leads to a received signal with non-zero mean. The Rician distribution models signal envelope in this case, with K factor dictating the relative power of the LOS component: $p_Z(z) = \frac{z}{\sigma^2} \exp\left[\frac{-(z^2+s^2)}{2\sigma^2}\right] I_0\left(\frac{zs}{\sigma^2}\right), \quad z \geq 0.$

2. The average received power in the Rician fading is $P_r = \int_0^\infty z^2 p_Z(z) dz = s^2 + 2\sigma^2$.

3. The Rician distribution is often described in terms of a fading parameter K, defined by $K = \frac{s^2}{2\sigma^2}$. The distribution in terms of K is: $p_Z(z) = \frac{2z(K+1)}{P_r} \exp\left[-K - \frac{(K+1)z^2}{P_r}\right] I_0\left(2\sqrt{\frac{K(K+1)z}{P_r}}\right)$

4 Estimation Techniques

In general, there are many methods used in estimating channel. The number of methods is based on the reduction of error that occurred by comparing the pilot symbols that are initially sent and received. Besides the method of least squares, the method of channel estimation that is widely used is the minimum mean square (MMSE), best linear unbiased estimator (BLUE), and adaptive boosting (AdaBoost).

When demodulating an OFDM burst, the equalizer response computed from the burst preamble is used to correct many imperfections in the OFDM signal. Because the equalizer response is not perfect, pilot tracking is used to correct for imperfections in the equalizer response and for imperfections that change over the length of the burst. In each OFDM symbol, four of the subcarriers are dedicated to pilot signals in order to make the coherent detection robust against frequency offsets and phase noise. Pilot subcarriers transmit with a known data sequence. This information is

used to determine the difference, or error, between an ideal signal and the actual received signal. Because the data is complex, the VSA calculates phase, amplitude, and timing error data. The error data can then be used to correct both pilot and data subcarrier imperfections, producing a more accurate demodulation. The following sub-sections describe the various channel estimation techniques in consideration.

4.1 Least Squares (LS)

Least Squares in prefered in many scenarios because it is easy and very simple to apply. The difference technique is based on an algorithm that takes the CIR value of the comparison of known pilot symbol. In OFDM systems, transmitters modulate a series of bits into symbols PSK / QAM, perform an IFFT operation on the symbol to turn it into a signal in time domain, and further send it through the channel. Received signal is usually distorted by the channel characteristics. To repair the bits sent, the effects of channel estimation should be expected. The equation of the received signal to the channel impulse response can be written into the equation,

$$Y = HX + W \tag{1}$$

where Y is the received signal, H is the impulse response of channel, W is the noise, and X is the signal sent and each is written by searching channel impulse response estimation $\hat{H}_p^{LS}$.

The least square estimates (LS) of the channel at the pilot subcarriers given can be obtained by the following,

$$\hat{H}_p^{LS} = (X_p)^{-1} Y_p \tag{2}$$

where $\hat{H}_p^{LS}$ represents the least-squares (LS) estimate obtained over the pilot subcarriers. We assume the entire subcarrier as orthogonal and that ICI did not occur between them.

LS has simple structure and low complexity characteristics but its performance is not satisfactory.

4.2 Linear Mean Minimum Square Error (LMMSE)

LMMSE is designed to minimize the estimation MSE. The LMMSE estimate of the channel response is,

$$\hat{H}_p^{LMMSE} = R_{HH_p}(R_{H_pH_p} + \sigma_\mu^2 (XX^H)^{-1})\hat{H}_p^{LS} \tag{3}$$

R_{HH_p} represents the crosscorrelation matrix between all subcarriers and the subcarriers with reference signals. $R_{H_pH_p}$ represents the autocorrelation matrix of the subcarriers with reference signals.The high complexity of the LMMSE estimator is due to the inversion matrix lemma. Every time data changes, inversion is needed. The complexity of this estimator can be reduced by averaging the transmitted data. Therefore, the term $(XX^H)^{-1}$ is replaced with its expectation $E[(XX^H)^{-1}]$.

The simplified LMMSE estimator becomes,

$$\hat{H}_p^{LMMSE} = R_{HH_p}(R_{H_pH_p} + \frac{\beta}{SNR} I_p)^{-1} \hat{H}_p^{LS} \tag{4}$$

where β is scaling factor depending on the signal constellation. SNR is the average signal-to-noise ratio, and I_p is the identity matrix.

MMSE uses the prior knowledge to improve its performance but the complexity is too high.

4.3 Our Deep-Learning Based Channel Estimation

Deep learning (DL) has emerged as an effective tool for channel estimation in wireless communication systems, especially under some imperfect environments. In this report, we have focused on SISO systems with a single transmitter and receiver. As deep neural networks (DNN) with rectified linear unit (ReLU) activation function is mathematically equivalent to a piecewise linear function, the corresponding DL estimator can achieve universal approximation to a large family of functions by making efficient use of piecewise linearity. We demonstrate that DL based channel estimation does not restrict to any specific signal model and asymptotically approaches to the minimum mean-squared error (MMSE) estimation in various scenarios without requiring any prior knowledge of channel statistics. Therefore, DL based channel estimation outperforms or is at least comparable with traditional channel estimation, depending on the types of channels. We detail our models and their estimation with architecture in the following sections. The overall idea is to use a training based adaptive algorithm applied at the receiver to reduce the error rate due to the multipath fading effect in the indoor setting.

4.3.1 JCES (Joint Channel Estimation and Symbol Detection) Model

Different from existing OFDM receivers that first estimate channel state information (CSI) explicitly and then detect/recover the transmitted symbols using the estimated CSI, the proposed deep learning based approach estimates CSI implicitly and recovers the transmitted symbols directly. To address channel distortion, a deep learning model is first trained offline using the data generated from simulation based on channel statistics and then used for recovering the online transmitted data directly. The deep learning based approach can address channel distortion and detect the transmitted symbols with performance comparable to the minimum mean square error (MMSE) estimator.

The architecture of the OFDM system with deep learning based channel estimation and signal detection is illustrated in Fig. 6. The baseband OFDM system is the same as the conventional ones. On the transmitter side, the transmitted symbols inserted with pilots are first converted to a paralleled data stream, then the inverse discrete Fourier transform (IDFT) is used to convert the signal from the frequency domain to the time domain. After that, a cyclic prefix (CP) is inserted to

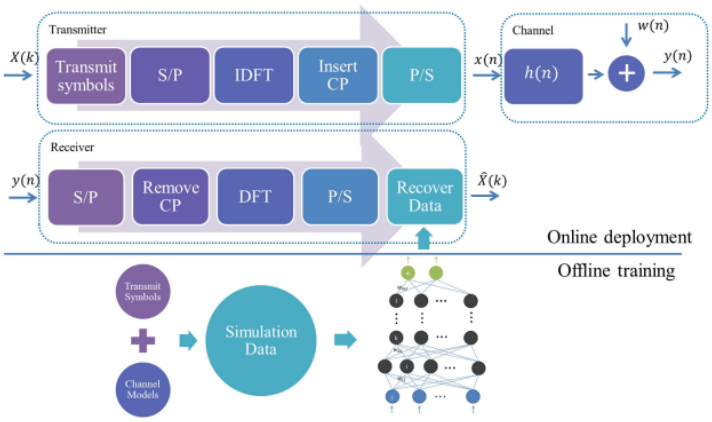

Figure 6: JCES Model Architecture

mitigate the inter-symbol interference (ISI). The length of the CP should be no shorter than the max delay spread of the channel. We assume that the pilot symbols are in the first OFDM block while the following OFDM blocks consist of the transmitted data. Together they form a frame. The channel can be treated as constant spanning over the pilot block and the data blocks, but change from one frame to another. The DNN model takes as input the received data consisting of one pilot block and one data block , and recovers the transmitted data in an end-to-end manner.

To obtain an effective DNN model for joint channel estimation and symbol detection, two stages are included. In the offline training stage, the model is trained with the received OFDM samples that are generated with using 5 randomly generated taps per OFDM frame.

The models are trained by viewing OFDM modulation and the wireless channels as black boxes. Historically, researchers have developed many channel models for CSI that well describe the real channels in terms of channel statistics. With these channel models, the training data can be obtained by simulation. In each simulation, a random data sequence is first generated as the transmitted symbols and the correspondent OFDM frame is formed with pilot symbols. The current random channel state is simulated based on the channel models. The received OFDM signal is obtained based on the OFDM frames undergoing the current channel distortion, including the channel noise. The received signal and the original transmitted data are collected as the training data. The model is trained to minimize the difference between the output of the neural network and the transmitted data. The difference can be portrayed in several ways. In our settings, we chose the L_2 loss,

$$L_2 = \frac{1}{N}\sum_{k}(\hat{X}(k) - X(k))^2$$

where $\hat{X}(k)$ is the prediction and $X(k)$ is the supervision message, which is the transmitted symbols in this situation. The DNN model we use consists of five layers, three of which are hidden layers. The numbers of neurons in each layers are $256, 500, 250, 120, 16$. The input number corresponds to the number of real parts and imaginary parts of 2 OFDM blocks that contain the pilots and transmitted symbols, respectively. Every 16 bits of the transmitted data are grouped and predicted based on a single model trained independently, which is then concatenated for the final output. The Relu function is used as the activation function in most layers except in the last layer where the Sigmoid function is applied to map the output to $[0, 1]$.

4.3.2 Cascaded CNN Model for Channel Estimation

We focused on one link between a pair of Tx and Rx antennas, i.e., we have Single-input, Single-output (SISO) communication link. For this link, the channel time-frequency response matrix $\mathbf{H}$ ($N_S \times N_D$) between a transmitter and a receiver, which has complex values, can be represented as two 2 D-images (one 2D-image for real values and another one for imaginary values). An example of the normalized real/imaginary 2D-image for a sample channel time-frequency grid with $N_D = 14$ time slots and $N_S = 72$ subcarriers, based on Long-Term Evolution (LTE) standard is shown in Figure 7.

The goal is to estimate the whole time-frequency of the channel using the transmitted pilots. We have used a comb type arrangement for pilot transmission. The estimated value of the channel at the pilot locations $\hat{\mathbf{h}}_p^{\mathrm{LS}}$ (which might be noisy) is considered as the LR and noisy version of the channel image.

To obtain the complete channel image, a two stage training approach is presented:

1. In the first stage, an Super Resolution (SR) network is implemented which takes $\hat{\mathbf{h}}_p^{\mathrm{LS}}$ as the vectorized low resolution input (once the real-part and then the imaginary-part) and estimates the unknown values of channel response $\mathbf{H}$.

2. In the second stage to remove the noise effects, a denoising Image Restoration (IR) network is cascaded with the SR network.

In this method, the time-frequency grid of the channel response is modeled as a 2D-image which is known only at the pilot positions. This channel grid with several pilots is considered as a low-resolution (LR) image and the estimated channel as a high-resolution (HR) one. A two-phase approach is presented to estimate the channel grid. First, an image super-resolution (SR) algorithm is used to enhance the resolution of the LR input. Afterwards, an image restoration (IR) method is utilized to remove the noise effects. For the SR and IR networks, we have used two recently developed CNN-based (Convolutional Neural Network) algorithms, SRCNN (Super-resolution convolutional neural network) and DnCNN (denoising convolutional neural network), respectively. SR recovers high-resolution (HR) images from low-resolution (LR) images. It is an important class of image processing techniques in computer vision and image processing and enjoys a wide range of real-world applications, such as medical imaging, satellite imaging, astronomical imaging, amongst others.

For training the model, let us denote the set of all network parameters by $\Theta = \{\Theta_S, \Theta_R\}$, where the Θ_S and Θ_R denote the set of parameter values for SR and IR networks, respectively. The input to this DL based channel estimation model is the pilot values vector $\hat{\mathbf{h}}_p^{\mathrm{LS}}$ and the output is the estimated channel matrix is denoted as $\hat{\mathbf{H}}$:

$$\hat{\mathbf{H}} = f\left(\Theta; \hat{\mathbf{h}}_p^{\mathrm{LS}}\right) = f_R\left(f_S\left(\Theta_S; \hat{\mathbf{h}}_p^{\mathrm{LS}}\right); \Theta_R\right),$$

where f_S and f_R are the SR and IR functions, respectively. The total loss function of the network is the Mean square error (MSE) between the estimated and the actual channel responses calculated as follows:

$$C = \frac{1}{\|\mathcal{T}\|} \sum_{\mathbf{h}_p \in \mathcal{T}} \left\| f\left(\Theta; \hat{\mathbf{h}}_p^{\mathrm{LS}}\right) - \mathbf{H} \right\|_2^2,$$

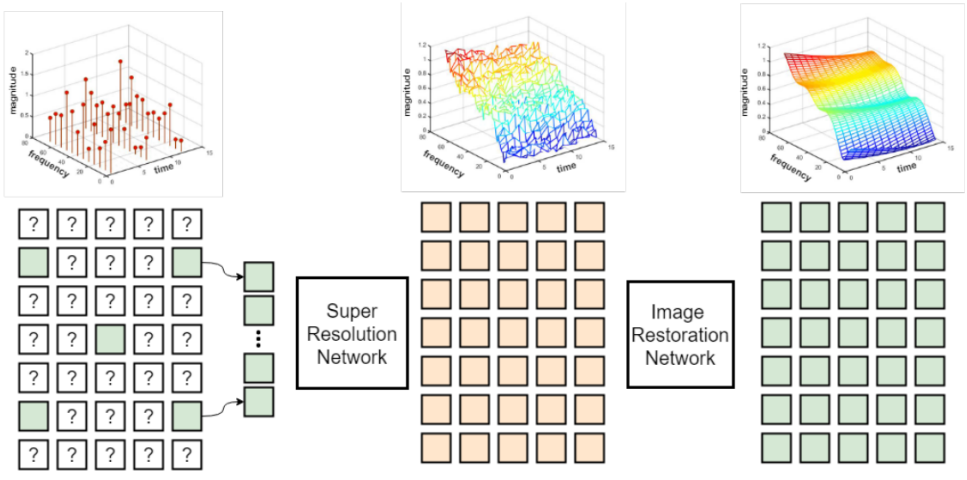

Figure 7: Pipeline for this DL model

where $\mathcal{T}$ is the set of all training data and H is the perfect channel. $\|\mathcal{T}\|$ is the size of the training set.

To simplify the training process, we use a two stage training algorithm. Where in the first stage we minimize the the loss of the SR network, C_1 :

$$C_1 = \frac{1}{\|\mathcal{T}\|} \sum_{\mathbf{h}_p \in \mathcal{T}} \|\mathbf{Z} - \mathbf{H}\|_2^2,$$

where $\mathbf{Z} = f_S\left(\Theta_S; \hat{\mathbf{h}}_p^{\mathrm{LS}}\right)$ is the output of the SR network. In the second stage, we freeze the weights of the SR network and find the parameters of the denoising network by defining $\hat{\mathbf{H}} = f_R\left(\mathbf{Z}; \Theta_D\right)$ and minimizing the loss function C_2 :

$$C_2 = \frac{1}{\|\mathcal{T}\|} \sum_{\mathbf{h}_p \in \mathcal{T}} \|\hat{\mathbf{H}} - \mathbf{H}\|_2^2$$

Note that, similar to image-based techniques, the optimal weights of the network is dependent on the value of the SNR; thus, to have a complete solution we have to re-train the network for each SNR value.

5 Our Schmidt and Cox Synchronization Implementation

The symbol timing recovery relies on searching for a training symbol with two identical halves in the time domain. This remains identical after passing through the channel, except that there is a phase difference between them caused by the carrier frequency offset. The two halves of the training symbol are made identical (in time order) by transmitting a pseudonoise (PN) sequence on the even frequencies, while zeros are used on the odd frequencies. This means that at each even frequency, one of the points of a QPSK constellation is transmitted. In order to maintain an approximately constant signal energy for each symbol, the frequency components of this training symbol are multiplied at the transmitter, or the four points of the QPSK constellation are selected from a larger constellation, such as 64-QAM, so that points with higher energy can be used. Transmitted data will not be mistaken as the start of the frame since any actual data must contain odd frequencies. Note that an equivalent method of generating this training symbol is to use an IFFT of half the normal size to generate the time domain samples. The repetition is not generated using IFFT, so instead of just using the even frequencies, a PN sequence would be transmitted on all of the subcarriers to generate the time domain samples which are half a symbol in duration. These time-domain samples are repeated (and properly scaled) to form the first training symbol.

Consider the first training symbol where the first half is identical to the second half (in time order), except for a phase shift caused by the carrier frequency offset. If the conjugate of a sample from the first half is multiplied by the corresponding sample from the second half ($T/2$ seconds later), the effect of the channel should cancel, and the result will have a phase of approximately $\phi = \pi T \Delta f$. At the start of the frame, the products of each of these pairs of samples will have approximately the same phase, so the magnitude of the sum will be a large value.

Let there be L complex samples in one-half of the first training symbol (excluding the cyclic prefix), and let the sum of the pairs of products be,

$$P(d) = \sum_{m=0}^{L-1} \left(r_{d+m}^{*} r_{d+m+L} \right) \tag{5}$$

which can be implemented with the iterative formula,

$$P(d+1) = P(d) + \left(r_{d+}^{*} L_{d+2L} \right) - \left(r_{d}^{*} r_{d+L} \right), \tag{6}$$

Note that there is a time index corresponding to the first sample in a window of $2L$ samples. This window slides along in time as the receiver searches for the first training symbol. The received energy for the second half-symbol is defined by

$$R(d) = \sum_{m=0}^{L-1} \left| r_{d+m+L} \right|^2 \tag{7}$$

The timing metric can be defined as,

$$M(d) = \frac{|P(d)|^2}{(R(d))^2} \tag{8}$$

6 Results

Figure 8 shows the SNR vs BER results for Block type 64 and 8 pilots using our JCES model. Figure 9, we have shown the results of the implementation of the Least Squares (LS) and Minimum Mean Squares Estimators (MMSE). The LS estimation was done by using the pilot values at the transmitter and receiver and using linear interpolation to fill in the missing values. The MMSE estimate was obtained from the correlation matrices obtained from the channel statistics. In Figure 10, we see that our JCES model gives good bit error rates even in the case of multipath fading (we have modelled Rician fading). We observe from the graph that there is a considerable improvement in performance when an entire block of 64 pilots is used as opposed to 8. In this model, a block of pilots is sent before another block of 64 data carriers. It is to be noted that we used this model for a slow fading channel. We performed a bare bone implementation of the Schmidt and Cox from scratch to fully visualize and understand the detection of the end of the cyclic prefix and the detection of the data. Our synchronizer results pertain to 64 carriers, with 30 of them turned off and a cyclic prefix of 16. In Figure 10, we compare the results of the implementation of the JCES scheme for comb type OFDM with the block type implementation. The above graph shows that this scheme performs better on block type OFDM implementations as compared to the comb type. This can be attributed to the fact that in block type implementations, the Deep Learning model has access to the pilot values of all frequencies as opposed to the comb type implementation which has access to pilot values at only a few frequencies. Since comb type OFDM has pilot information of only a few frequency bands, it is difficult to design Deep Learning models for comb type OFDM as this would give the model an additional task of inferring the frequency effects at non-pilot frequencies along with the channel estimation task. Figure 11 shows the BER performance of our Channel Estimation model using Super Resolution Deep Neural Network in comparison to that of Ls and MMSE estimates on a OFDM system having multi-path fading channel. As shown, the DNN model performs much better than LS estimates but isn't close to the MMSE estimates.

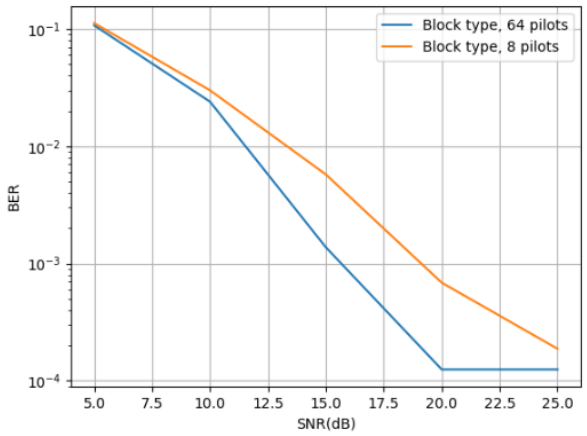

Figure 8: JCES Model SNR vs BER

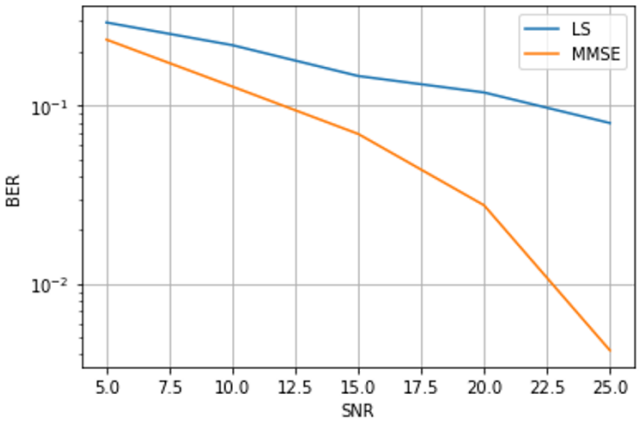

Figure 9: LS and MMSE

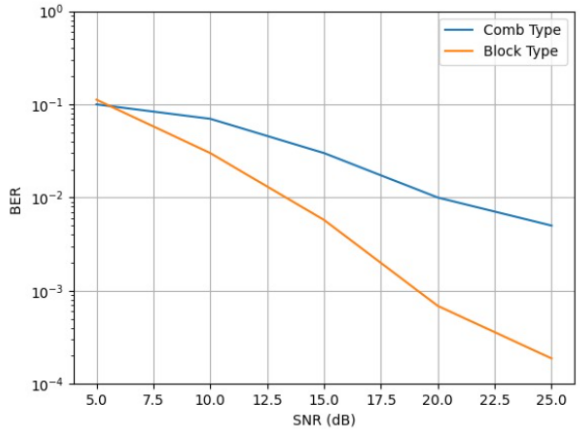

Figure 10: JCES Model

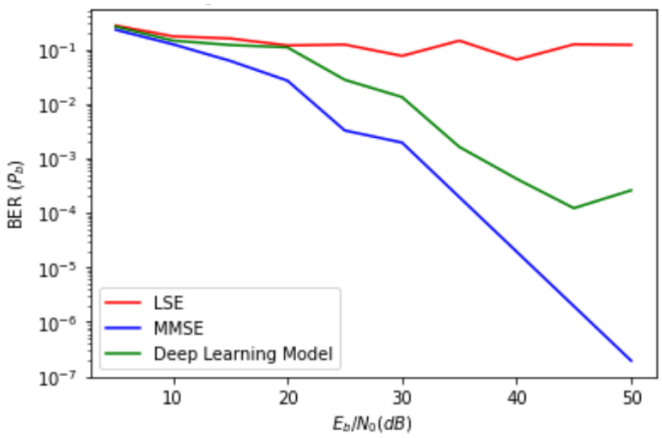

Figure 11: Cs-CNN Model

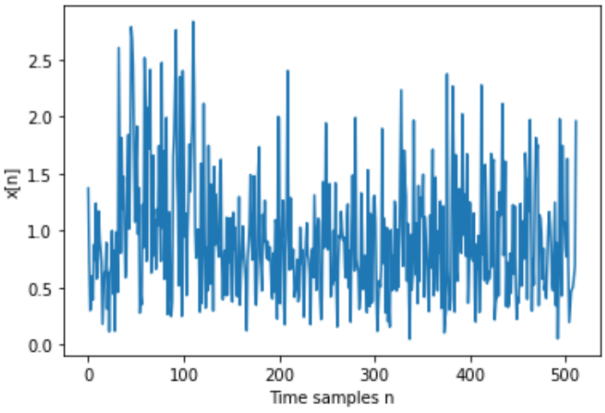

Figure 12: OFDM Frame

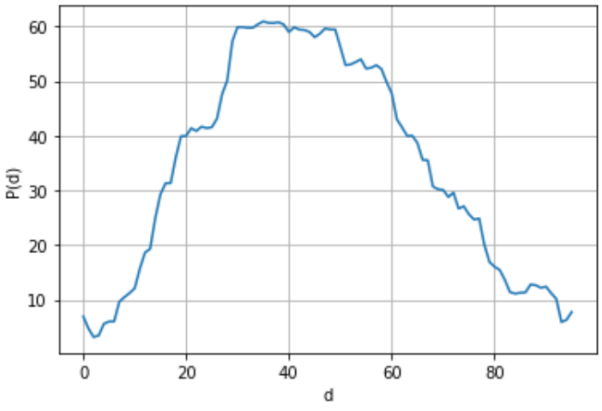

Figure 13: Timing offset

7 Conclusion and Future Work

We explored different DL based channel estimation techniques in this project. Building two models and completely coding it on our model was definitely a challenge. We hope to further carry out experiments with the models we built. For example, in the case of JCES, we know that the model should be able to perform symbol detection pretty close to the LMMSE even in the absence of cyclic prefix (CP). This is something we would like to investigate further. Keep in mind, that the models take around 3 days each to initially train. The great thing is that once our models are trained offline, we can easily perform good estimations in real time very well. With the second model, we explicitly estimate the channel response $\mathbf{H}$ and we plan to implement variable length cyclic prefix OFDM to maximize the data transfer while keeping the BER small as a function of SNR. We would do this by dynamically estimating the channel response for the frames, and then updating the value of the cyclic prefix depending on the rms delay spread of the channel as we know by the standard convention that the CP should be greater than the delay spread.

References

[1] C. Anjana, S. Sundaresan, T. Zacharia, G. Rajendran, and S. Kp. An experimental study on channel estimation and synchronization to reduce error rate in ofdm using gnu radio. *Procedia Computer Science*, 46:1056–1063, 12 2015.

[2] A. Marwanto, M. A. Sarijari, N. Fisal, S. K. S. Yusof, and R. A. Rashid. Experimental study of ofdm implementation utilizing gnu radio and usrp - sdr. In *2009 IEEE 9th Malaysia International Conference on Communications (MICC)*, pages 132–135, 2009.

[3] T. J. Mathews and Z. Han. Usrp2 implementation of compressive sensing based channel estimation in ofdm. In *2013 Third International Conference on Communications and Information Technology (ICCIT)*, pages 83–87, 2013.

[4] P. Pragna, P. Sahana, S. Naik, S. R. Charantimath, and V. Karjigi. Channel estimation using conventional methods and deep learning. In *2021 2nd Global Conference for Advancement in Technology (GCAT)*, pages 1–7, 2021.

[5] P. Sowjanya and P. Satyanarayana. Testing different channel estimation techniques in real-time software defined radio environment. *International Journal of Advanced Computer Science and Applications*, 11(2), 2020.

[6] K. Venkatesh, G. Nair, and S. Kulkarni. Implementation of a machine learning based ofdm detector using gnu radio. In *2021 International Conference on Communication information and Computing Technology (ICCICT)*, pages 1–5, 2021.

[7] D. Wu and W. Fan. A new channel estimation method based on pilot-aided and local adaptive least squares support vector regression in software radio ofdm system. In *2009 International Joint Conference on Artificial Intelligence*, pages 349–352, 2009.

[8] H. Ye, G. Y. Li, and B.-H. Juang. Power of deep learning for channel estimation and signal detection in ofdm systems. *IEEE Wireless Communications Letters*, 7(1):114–117, 2018.