Testing different Channel Estimation Techniques in Real-Time Software Defined Radio Environment

Ponnaluru Sowjanya[1], Penke Satyanarayana[2]
Department of Electronics and Communication Engineering
Koneru Lakshmaiah Educational Foundation, Vaddeswaram, Andhra Pradesh, India

***Abstract*—In modern wireless communication to maximize spectral efficiency and to minimize the bit error rate OFDM (Orthogonal frequency-domain multiplexing) is used. OFDM is used broadly in networks using various protocols, including wireless vehicular environment IEEE 802.11p, IEEE 802.16d/e Wireless Metropolitan Area Networks, Long-Term Evolution 3GPP networks and IEEE 802.11a/g/n Wireless Local Area Networks. The main challenges involved when using OFDM for wireless communications are short channel-coherence bandwidth and the narrow coherence time, and both have a major effect on the reliability and latency of data packet communication. These properties increase the difficulty of channel equalization because the channel may change drastically over the period of a single packet. Spectral Temporal Averaging is an enhanced decision-directed channel equalization technique that improves communication performance (as far as the frame delivery ratio (FDR) and throughput) in typical channel conditions. This paper reports tests of Spectral Temporal Averaging channel equalization in an IEEE 802.11a network, compared with other channel equalization techniques in terms of the FDR in a real-time environment. Herein, a software defined Radio (SDR) platform was used for estimating the channel. This proves that the system can provide over 90% of delivery ratio at 25 db of Signal to Noise Ratio (SNR) for various digital modulation techniques. For this purpose, an experimental setup consisting of software-defined radio, Universal Software Radio Peripheral (USRP) N210 along with wide bandwidth daughter board as hardware and GNU radio is used.**

Keywords*—*Channel Estimation; GNU Radio Companion (GRC); Orthogonal frequency-domain multiplexing (OFDM); software-defined radio (SDR); Spectral Temporal Averaging (STA); Universal Software Radio Peripheral (USRP)

I. Introduction

The latest developments in wireless communications like phones, protocols and applications are impressive. For large data high data rates applications and low energy usage are important for upcoming wireless technologies. [1]. When the number of users with common Mobile Bandwidths expands, traffic congestion will be a harder task, and increased flexibility in the requirements of transmission will be required if multiple access contention is to be handled. Such problems are often faced by wireless communication protocols. The new mobile technology Standard Long-Term Evolution (LTE) focuses a lot on experimental and evaluation of wireless protocols.

A widely used modulation technique by means of which the flow of symbol is split in several narrowband subcarriers which are modulated progressively, and which have very close connections for high data transmission rates is OFDM. This method eliminates the frequency-selective loss signal. OFDM generates several non-selective parallel frequency streams, thus reducing inter-symbol interference (ISI). The decoder process is utilized to gain information on the channel state Information (CSI) and to account for channel effects such as delay spread and lower Doppler distribution and this process makes the OFDM system reliable.

SDR may provide flexible, upgradeable and longer lifetime radio equipment for the military and for civilian wireless communications infrastructure as shown in Fig. 1. SDR may also provide more flexible and possibly cheaper multi-standard-terminals for end users. It is also important as a convenient base technology for the future context-sensitive, adaptive and learning radio units referred to as cognitive radios. SDR also poses many challenges, however, some of them causing SDR to evolve slower than otherwise anticipated. Transceiver development challenges include size, weight and power issues such as the required computing capacity, but also software architectural challenges such as waveform application portability. SDR has demanding implications for regulators, security organizations and business developers. In a multicarrier technology such as OFDM, it is crucial to maintain orthogonality among all the subcarriers to prevent the Inter Carrier Interference (ICI) which if not done leads to significant performance degradation. In a high mobility environment, such as aerial vehicle communication, multi carrier transmission leads to severe ICI due to Doppler shift.

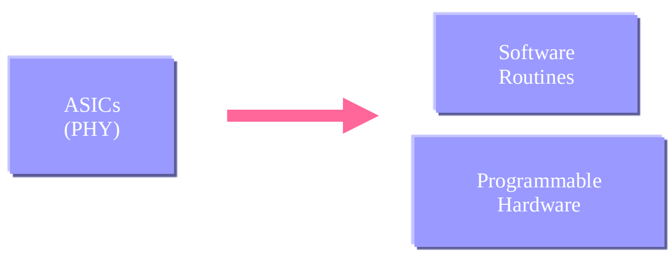


Fig. 1. Software Defined Radio.

II. Literature Review

Several authors have tried to use several channel estimations schemes to implement OFDM in real-time. with an emphasis on the Least Square (LS) channel estimation scheme and signal capture they, experimentally investigated the preamble detection, transmitter sensing and physical layer-capture effect [2]. Another team developed and evaluated a channel estimation system for pilot subcarriers with limited complexity, high precision, little pilot bandwidth and buffer-free data flow [3], and Great strength [4]. Using Field-Programmable Gate Arrays (FPGAs), that require changeable-power loading system of three levels, such experiments have been executed in real-time OFDM transceivers. To this end, a high-performance, radio-based software platform (CuSora) was developed as a modern processor for GPuS high-speed signal processing [5]. This software supports development of different protocols using an entire hardware and software framework. Signal processing components for the 802.11a and 802.16 protocols were tested in CuSora, and 2-dimensional linear interpolation was implemented for channel estimation in these tests. This simple algorithm modulates signals along both the frequency axis (FA) and the time axis (TA). Along the FA, the major technique is the evaluation of received samples and pilots within the frequency period between two adjacent pilot packets. Along the TA, the main determinant of transmission performance is channel estimation. Armour et al. discussed the potential trade-offs between efficiency and complexity when using the decision directed Least Mean Square (LMS) algorithm with the Hiperlan E and IEEE P802.1la standards [6].

SDR is an ideal solution for rapid prototyping and testing of wireless communication protocols [7][8]. The tools included in SDR platforms allow for easy reconfiguration of transceiver devices to quickly implement standards and protocols. SDR solutions can be used to test wireless sensor networks [9], satellite communications [10], and many other applications. Currently, research on implementing different systems using SDR technology is being conducted. In [11], the authors integrated their design of a joint DA-ML estimator using SDR combined with FPGA. And finally, they tested their prototype in a real-time environment. In [12], the author used USRP N200/N210 as a frontend and proved that the experimentation is possible at 60 GHz using SDR. In [13], using SDR platform, the authors designed, implemented, and evaluated a MIMO system with eight antennas. They chose LTE parameters as their system parameters. In [14], the authors detected the human activity using SDR platform. They extracted the channel state information using two USRPs as transmitter and receiver using 64-FFT point's OFDM modulation technique. In [15], the author implemented some of the communication features like pulse shaping, demodulation, and synchronization in a real=time environment; therefore, they have chosen the SDR platform and USRPs are used as SDR front end. Through these references, it can be concluded that SDR is a present trending technology.

For signal processing GRC is an open source platform. On a regular PC, this allows for the installation of a general-purpose processor. To support device networking, it is one of the best tools for SDR. This offers several projects to make a flexible software radio system utilizing various software libraries. In [16], The researcher received a 3-30MHz frequency band experimental SDR receiver with the aid of GRC / FPGA programmable hardware. In [17], Writer suggested a wide-band frame-level adaptation that can be simply implemented in various protocols. They showed their viability with GRC / SDR platform. In [18], Compatible range, instead of channel condition maintaining that energy-constrained systems scale down their sampling levels is called Sampless Wi-Fi. GRC / USRP was used to test their SDR system. In [19] and [20], the anti-jamming strategy was suggested in various methods and their approach was tested using GRC. In [21], The author has created a mechanism called TREKS to facilitate communication with Spread Spectrum without pre-shared information. They used a GRC and USRP test bed to evaluate this process. In [22], To research OFDM quality in the 802.11 standard, they built an SDR-based test bed. Consequently, GRC / USRP was chosen as their perfect platform. In [23], Two separate graphics processing unit implementation methods in a Software Defined Radio set-up were evaluated and it was found that only one proposed method was environmentally friendly. In [24], A new PHY / MAC protocol, known as Diversity-aware Wi-Fi, is developed and tested using USRP / GRC platform, and compared to existing methods. Despite this variety of features, GRC can be considered to be one of the strongest SDR apps [25].

In this paper, understanding the challenges of, dynamically changing channels, however, will have serious problems with Doppler and delay spread, which cause multipath fading channel effects. Therefore, we implemented a decision-directed spectral temporal averaging (STA) channel estimator [26] in real-time data transmissions considering the IEEE 802.11a protocol. We compared the STA scheme with LS, LMS, and comb-pilot linear interpolation schemes. We prepared a custom testbed for this study comprising GRC/USRP as their software and frontend.

We designed and tested an IEEE 802.11a transceiver with a selection of channel-estimation methods in real-time environment. The results show that the STA channel-estimation scheme achieves the best frame data rate of the techniques that we tested. Here, real-time signal constellations are also observed to find the effect of multipath propagation of signals through flat fading channels. Both medium access control (MAC) and physical (PHY) layers were implemented in the SDR platform for testing. A known stream of data was transmitted and received using various modulation schemes, and the FDR performance was analyzed for comparison.

The remaining document is as follows:

Section 2 contains information on the testing of channel estimation schemes.

Section 3 explains how we used the GRC platform and USRPs to implement this network in a real-time environment.

Section 4 introduces the channel-estimation techniques that we tested for comparison against the STA technique.

Section 5 describes the outcomes of tests and compares the outputs of the different channel estimates and Section 6 includes our conclusions.

III. Channel Estimation Techniques

Channel approximation is the analysis of a predetermined mathematical model of communication channels.The two variables that define the computational network design are short-term CSI / instant CSI and long-term CSI / statistical CSI.

The CSI provided statistical data for the long term, including statistical distribution and average channel gain. The only element used in the short-term CSI is channel impulse response. In OFDM systems the channel impulses were detected by the time / domain signal and channel frequency response, respectively, before and after DFT transmission, and they had been defined by frequency / domain channel estimation.[27]. Three methods can be used to approximate a channel: pilot-assisted, blind stream and Decision Directed Channel Estimation (DDCE).

The pilot-aided channel approach is one of the most standard way of calculating channels. A sender transfers in this system well-known pilot data used by both sender and recipient as proof. Pilot symbols are computationally somewhat complicated, but they are used in every wireless communication device. That method, though, decreases the bit rate since some symbols are used instead of information for pilots and the network space is lost. Even if the number of pilots has decreased, it is a challenge to estimate the channel precisely. As shown in Fig. 1, using blocks or combs for pilot assignment. A pilot block model is ideal for a slow fading stream, where the signal moves slowly. A comb pilot distribution is therefore suitable for the quick fading flow, since the pilots are arranged equally throughout the symbol sequence. To decide the channel response of the information symbols and consider this more vulnerable method to frequency-selective channels, interpolation between frequency-domain is required.

The structure of the block pilot is shown in Fig. 2. At the start of each subcarrier, a pilot data is there in the OFDM symbol. The time-domain interpolations are used to approximate the channel using these pilots. Since the opposite of the Doppler rate, f_{dp} in the channel provides continuity time, the pilot symbol duration will comply with the following variation:

$$S_t \leq \frac{1}{f_{dp}} \tag{1}$$

The comb-pilot's design is also shown in Fig. 4. Frequency-domain interpolation stream pilot tones are incorporated in every OFDM symbol between subcarriers on a regular basis. Since the bandwidth of coherence of the reverse of the delay spread σ_{max}. maximum, the pilot symbol duration should satisfy the below change:

$$S_f \leq \frac{1}{\sigma_{max}} \tag{2}$$

Estimation of the blind channel does not require pilot symbols and relies instead on intrinsic data received from symbols. Although no bandwidth of the signal is consumed by this approximation method, the computations are much more complicated and lead to higher bandwidth. For illustration, to estimate one channel coefficient, almost 100 symbols are needed. For this reason, this blind channel estimation method is seldom used in real-world wireless communication systems.

DDCE, which is our subject below, uses both observed channel approximation information symbols and pilot symbols. The estimated values are updated as diagrammed in Fig. 3. Thus, DDCE offers superior performance than pilot-aided channel estimation.

In our study, data were transmitted and received in real time to allow testing of channel-estimation techniques in a realistic context. We assumed that the channels are positioned dynamically. We tested four channel-estimation techniques: LS, LMS, comb-pilot linear interpolation, and STA. Of these, LMS and STA are decision-directed interpolation schemes.

A. Least Squares Equalizer

In modern hardware implementations, the basic LS equalizer algorithm is frequently used as a regular method [28]. For estimating the channel, in IEEE 802.11p the long training sequence is treated as block pilots. Let us say that after the start of the frame the two long-term preambles are referred to as

$$\underbrace{y_G[n_p-128],\ldots,y_G[n_p-65]}_{Long\ Preamble\ 1},\underbrace{y_G[n_p-64],\ldots,y_G[n_p-1]}_{Long\ Preamble\ 2}$$

And they're named T1 and T2. T1[n] and T2[n] are the time domain symbols for approximation of LS channels are derived from these two LPs. Instead they measure their N-point DFTs are represented below:

$$Y_1(k) = \sum_{n=0}^{N-1} T_1[n]\, e^{-\frac{2\pi jkn}{N}} \tag{3}$$

$$Y_2(k) = \sum_{n=0}^{N-1} T_2[n]\, e^{-\frac{2\pi jkn}{N}} \tag{4}$$

The N-points between two learning symbols are the same, i.e. X1(k)= X2(k)= X(k). The estimate of the LS for H(k) is as follows:

$$\hat{H}(k) = \frac{(Y_1(k)+Y_2(k))}{2X(k)} \tag{5}$$

The data in the packet will be equalized after channel estimation is completed. The obtained signal's DFT is defined as Y(k). Through equalizing the obtained DFT function, the transmitted information is calculated.

$$\hat{X}(k) = \frac{Y(k)}{H(k)} \tag{6}$$

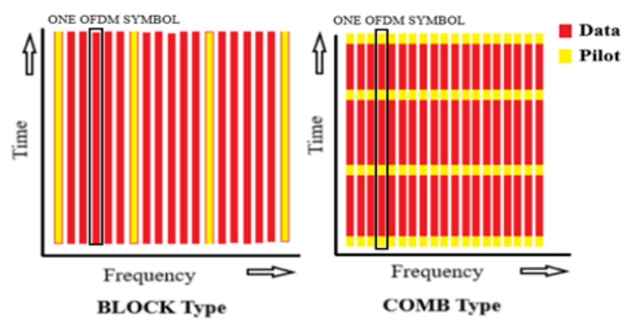


Fig. 2. Types of Pilot Assignments.

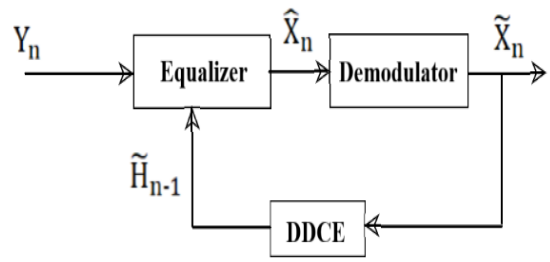


Fig. 3. Decision-Directed Channel Estimation.

A basic one-taps equalizer approach is used by each subcarrier in the frequency domain. The least square equalizer fits all the symbols in the packet. We cannot accurately represent the channel estimate H(k), if the channel drastically varies over the period of a packet. Also, equalization could falsify the received signal rather than making correction. So, a perfect and effective means of tracing the channel is needed.

B. Least Mean Squares

The LMS algorithm disables this constraint in channel tracing by acclimating the channel estimates while the signal is received. The original channel estimate is automatically adjusted to the time-variant properties of the communication channel by the LMS algorithm. The mean square error among the desired equalizer output and the actual equalizer output will be minimized using this algorithm. As the LS equalizer, beginning with a similar introductory decision after the ith OFDM symbol update the channel utilizing the constellation point $\hat{X}_i$ onto where the obtained symbol Yi was deallocated:

$$\widehat{H}_i(k) = (1-\alpha)\widehat{H}_{i-1} + \alpha \frac{Y_i(k)}{\hat{X}_i(k)} \quad (7)$$

To average the coefficients of the time-domain channel that finds α, a low-pass filter is used. Within the equalizer filter length constraints, the signal-to-distortion ratio at its output is maximized in LMS equalizer. This approach is constrained because if the signal obtained is longer than the propagation time, the equalizer cannot decrease this distortion.

C. Comb-Pilot Interpolation

None of the LS or LMS algorithms average symbols across the frequency domain but consider each subcarrier individually instead. The received values from each pilot sub-carrier are initially obtained in a comb-pilot interpolation in the frequency domain and each symbol then demodulated. [29]. The four-element vector Y_p is used to choose these values. The protocol defines DFT values for known sent pilots in these subcarriers from the four-element matrix, X_p. At every pilot subcarrier the LS estimate is designed as

$$H_p(k) = \frac{Y_p(k)}{X_p(k)} \quad (8)$$

The above equation represents a four-element vector which denotes the regularly spaced channel estimations. The end points are attached to the vector to obtain the estimates as follows:

$$H_p^{|} = \left[m_{H_p}\, H_p^T\, m_{H_p}\right] T \quad (9)$$

where m_{H_p} is the mean of H_p. Instead of extrapolation from the subcarriers −21 and 21, this mean is used for the endpoints because the actual channel response at the edge frequencies cannot be resolved. For every OFDM symbol, this interpolation is done. In the time domain of sorting, a low-pass filter such as formula (7) can be used.

D. Spectral Temporal Averaging

The STA channel-estimation scheme is detailed below. From the training preamble as in equation (5), the initial estimate of the channel is first obtained. The first symbol in the packet goes through this primary estimation. After completion of this symbol demodulation a channel estimate is framed:

$$H_i(k) = \frac{Y_i(k)}{X_i(k)} \quad (10)$$

At symbol i, X_i, $Y_i(k)$, and H_i are the decided constellation point, demodulated subcarrier values, and the resulting estimate respectively. This estimation will then be determined on average over the symbol frequencies as a standard moving average so that the approximation at subcarrier λ is

$$H_{update}(\lambda) = \sum_{k=-\beta}^{\beta} w_k H_i(\lambda + k) \quad (11)$$

where β is an integer that determines the number of terms involved in the average and $\sum_{k=-\beta}^{\beta} w_k = 1$. From this averaging operation, absent subcarriers (subcarriers that do not contain data) are omitted. For example, subcarrier 26 and β = 3, the only used average subcarriers are 26, 25, 24, and 23, and the weights are corrected consequently. Since the information on the null subcarrier is not transmitted, the value of $H_i(0)$ is substituted with the average of subcarrier −1 ($H_i(-1)$) and subcarrier 1 ($H_i(1)$). For all 52 subcarriers, the frequency averaging is be performed, and subsequently, the channel estimate is restructured using the following equation:

$$H_{STA,t} = \left(1 - \frac{1}{\alpha}\right) H_{STA,t-1} + \frac{1}{\alpha}\left(H_{update}\right) \quad (12)$$

Where α defines the time-domain parameter of the moving average. $H_{STA,0}$ defines the initial estimate of the channel obtained during the estimation of the preamble. For equalizing the next symbol, the estimate is applied, and the whole packet is demodulated by iterating the process. In the present study, by observing we get the best performance by selecting the parameters α and β as 0.5 and 2, respectively. In LMS equalizer, the same value of α was used to facilitate easy comparison.

IV. Methodology

GRC is a technology framework open source that offers signal processing frames to radio applications. It can be used with a wide variety of hardware components compared to other SDR frameworks. GRC is not a fixed application-oriented environment, so it provides a solid foundation for the use of nearly any hardware components. The setup of the system used in this research is illustrated in Fig. 4.

For the transmitter and receiver attached to the GRC program, we created a SDR with N210 USRPs in real time (Fig. 5). Here we find a space indoors in which no signals of Wi-Fi are being sent, and the transmitter and the receiver are roughly 1 meter away. The SDR architecture comprises three sections of the baseband, Intermediate Frequency (IF) and Radio Frequency (RF). The RF signal is sent to the USRP, which includes the daughterboard, Analog to Digital converter (ADC)/Digital to Analog Converter (DAC), Field Programmable Gate Array (FPGA)s, Digital Signal Processing (DSP) and Application-Specific Integrated Circuit (ASIC)s by an intelligently designed antenna. For versatile baseband signal processing GRC modules are used [30].

Fig. 4. Experimental Setup.

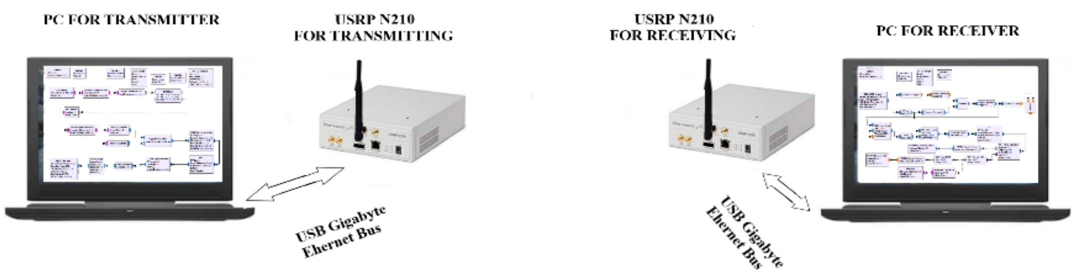


Fig. 5. Image of the Test Set-up for a Real-Time Radio System.

In order to implement the real-time radio system, two PCs were used to operate the SDR program. The used components of software and hardware are included in Table I and Table II. The PHY and MAC layers are both implemented in GRC modules, which allow us to change the layers according to specific requirements and easily analyze the results. The USRP hardware driver (UHD) is required for connecting the USRP frontends to the PCs. UHD provides common transmitting and receiving interfaces for the two USRP devices detailed in Fig. 6 and 7.

A. *Transmitter*

To implement an IEEE 802.11a LAN in GRC, an Out-Of-Tree (OOT) was used [29]. OOT modules are extended custom software blocks that are used for implementing application-specific functionalities. The transmitter implementation in GRC is shown in Fig. 8. Table III details the used parameters of IEEE 802.11a PHY.

TABLE. I. PC COMPONENTS USED IN THE SETUP

PC Component	Type
CPU	Intel core i7-8550U
RAM	16 GB
Operating system	16.04 LTS
Software	Ubuntu-Version 3.7
UHD Version	Version 003 011 000 000

TABLE. II. HARDWARE MODULES MENTIONED IN OUR IMPLEMENTATION

Hardware Component	Type
SDR	USRP N210
Daughter Board	WBX
Transmitting Antenna	VERT 400
Receiving Antenna	Dipole Antenna
SNR	0-30 dB

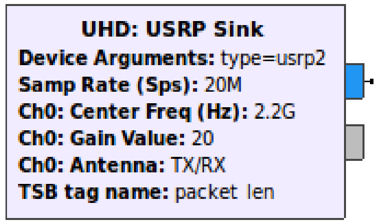


Fig. 6. Transmitting Block in UHD.

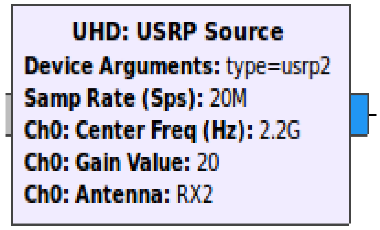


Fig. 7. Receiving Block in UHD.

TABLE. III. PHY VARIABLES USED IN THE IMPLEMENTATION OF OFDM

Parameter	Measurement
Bandwidth	10 MHz
OFDM subcarrier	64
Subcarrier Spacing	312 KHz
OFDM Symbol time	4 µs
Guard time	1.6 µs
Comb-pilot spacing	4.4 MHz
Center frequency	2.2 GHz

B. Receiver

The standard IEEE 802.11 has been extended to support typical channel propagation with a latest operating model that supports instantaneous contact without setting up a previous link and An Modified Physical Layer (PHY) focused on Orthogonal Frequency-Division Multiplexing (OFDM), close to IEEE 802.11a but doubled with all timings. It switches converts the IEEE 802.11a's 20MHz signal into the new 10MHz signal for vehicular applications. The change to 10MHz, in a nutshell, renders the signal better for delay propagation but more responsive to Doppler simultaneously and channel period variations. It indicates the transition has was not clearly improved. It simply the trade-off was rebalanced and was not without any question. The receiver implementation in GRC using IEEE 802.11 OOT modules is illustrated in Fig. 9. OFDM receiver has synchronization and channel-estimation modules that help with data recovery. These modules rely on the preamble data that are appended to every frame. For this purpose, we use equations (13), (14), and (15) to detect the beginning of a frame, and the equations are related as the block diagram in Fig. 10 illustrates.

$$a[n] = \sum_{k=0}^{N_{win}+15} y_G[n+k]\overline{y_G}[n+k+16] \tag{13}$$

$$p[n] = \sum_{k=0}^{N_{win}-1} y_G[n+k]\overline{y_G}[n+k] \tag{14}$$

$$c(n) = \frac{|a[n]|}{p[n]} \tag{15}$$

Equation (15) is used for detecting the beginning of the frame. The modules “WiFi Sync Short” and “WiFi Sync Long” handle frame detection, frequency offset correction, and channel estimation. While moving from “WiFi Sync Short” to “WiFi Sync Long,” only long-preamble data were transferred, and short-preamble data were detected and discarded in the “WiFi Sync Short” block. The decoded data were collected by the “Wireshark connector” module and displayed in Wireshark, which gives various information about the signal. This output is shown in Fig. 11.

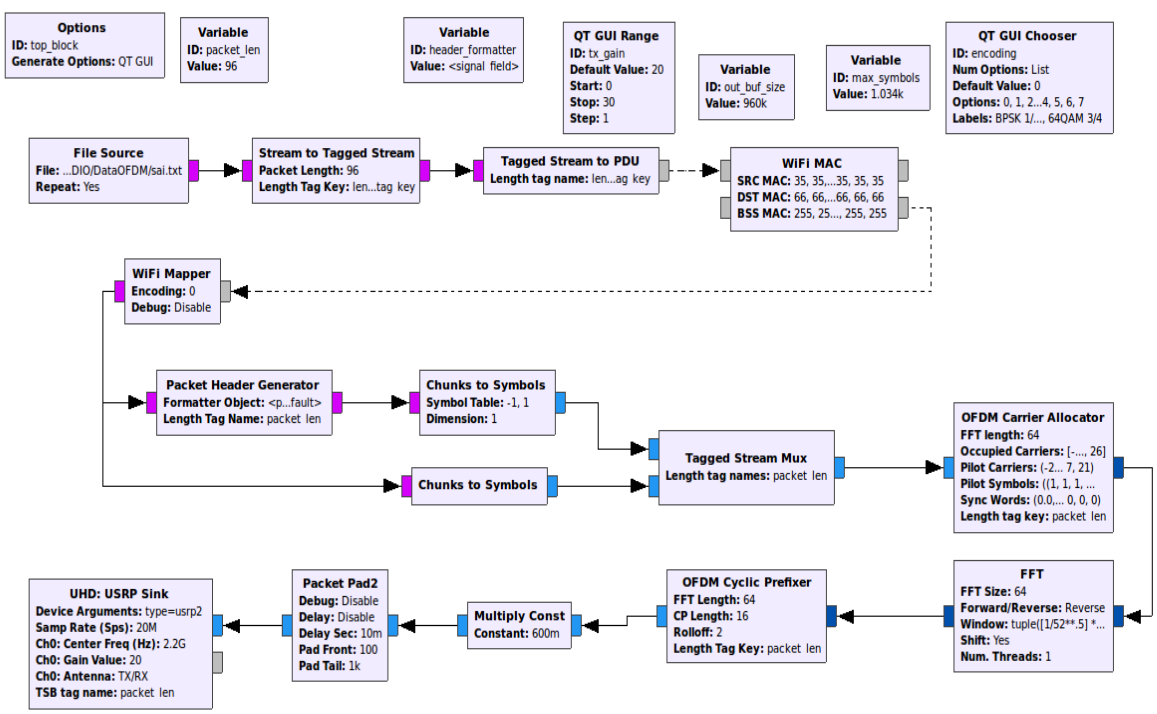


Fig. 8. OFDM Transmitter using GRC.

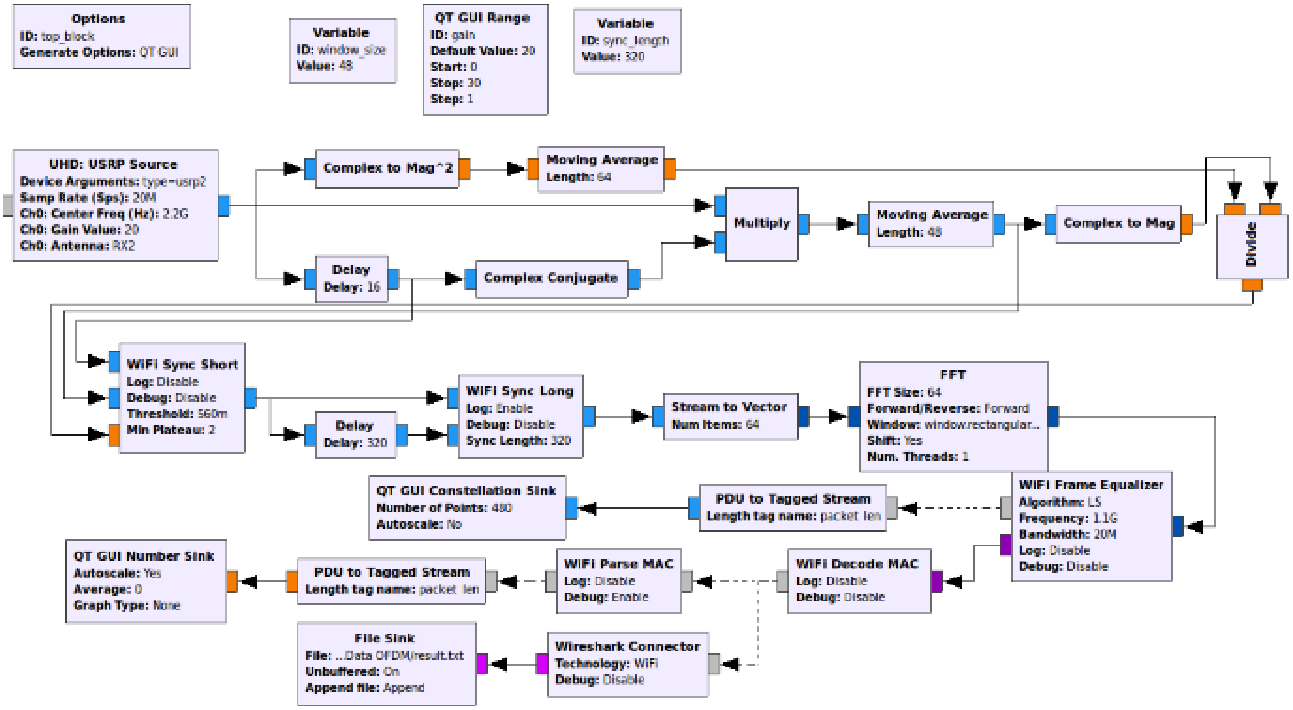


Fig. 9. OFDM Receiver using GRC.

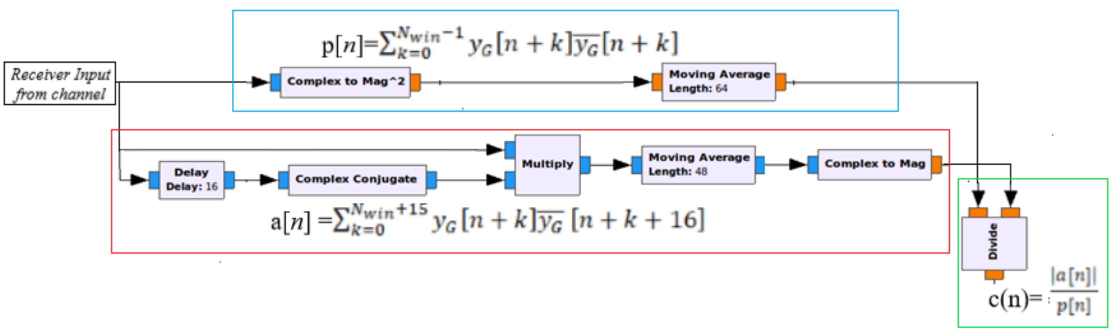


Fig. 10. Detection of Frame Starting.

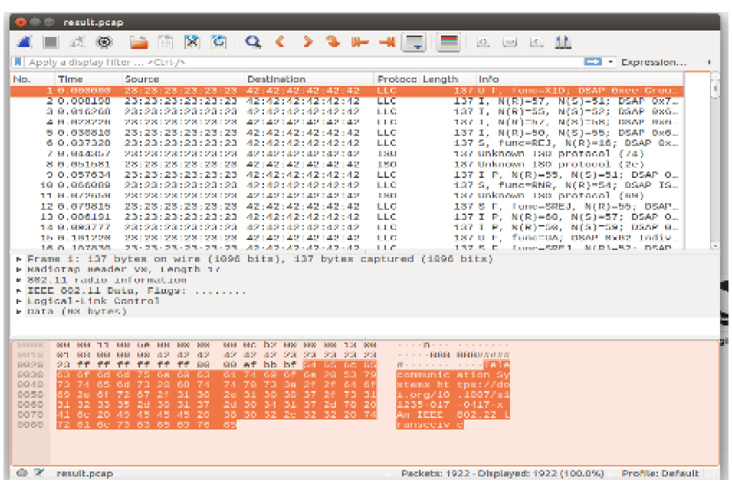


Fig. 11. Wireshark Connector Output.

V. Results and Discussion

We tested four different channel-estimation techniques in a real-time implementation of the IEEE 802.11a standard [31]. This section gives constellation plots as measured from the various modulation schemes at different transmitting powers, along with the FDR results. We highlight the systems applicability for real-time environment by determining the frame delivery ratio. This gives the percentage of successfully delivered frames by transmitted frames. In this comparison test between different channel estimations we considered 100 frames per run for each transmitting power from 0dB to 30dB. After testing different system factors (i.e., α, and β, as applicable) we select the best performing scheme and the results are generated from that system.

We trust that on-road situations can be reflected by this evaluation procedure. This structural proof uses genuine experimental data that are gathered from field measurements. The accuracy of the channel models is undoubted in this process. This method permits us to compare various equalization methods. This approach is faithful, easy to use, and repeatable.

We extended our tests by comparing different modulation schemes using their constellation plots at the receiver. Here we see the pilot information as the learning signals that enable us to measure the changes in the frequency-domain channel for a given test package in each subcarrier. We can generate the 802.11a waveforms using the stored frequency-time channel response. The performance of 802.11a at each 100 frames is done using this method. Results of this study are shown below.

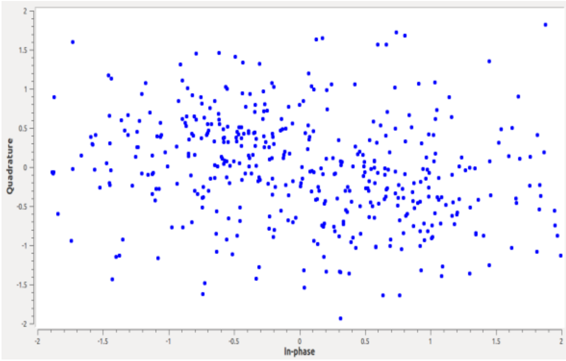
A. *Comparison of Frame Delivery Ratio (FDR) for different Modulation Schemes*

Simulations show that noise and interference in network simulators could be treated similarly. To provide evidence for these findings through measurements, we installed radio transceivers in our laboratory. Below, the FDR results from all four modulation schemes are plotted for several SNRs in dB. Constellation plots and FDR results for different modulations and different coding rates in the real-time testbed environment are shown in Fig. 12, Fig. 13, Fig. 14, Fig. 15, and Fig. 16 and Fig. 17 respectively. Our results show that technological considerations such as system parameters have a lesser impact on the signal than the transmission of multipaths. Under practical propagation conditions, such a learning environment allows us to determine the importance of the evaluated variables to a localization process. We demonstrate that propagation conditions vary even in LOS conditions in each SNR and that the radio signals undergo distinct propagation conditions in specific SNRs. When we found in each situation in the constellation diagrams, the received symbols form into clusters, each of which has its ideal point when the SNR is at 30dB but at 0dB, the received symbols scatter randomly over the constellation diagram. Therefore, the optimal points for the obtained signs are difficult to identify, and demodulation errors may occur frequently. These plots show that the BPSK modulation performs the best, irrespective of the encoding rate. All the results shown in this section were recorded when using the STA channel-estimation technique and the Schmidl-cox synchronization technique. Here we considered 100 frames per run for each transmitting power. Observing these results, we can conclude that for typical channel conditions we can get best results at BPSK.

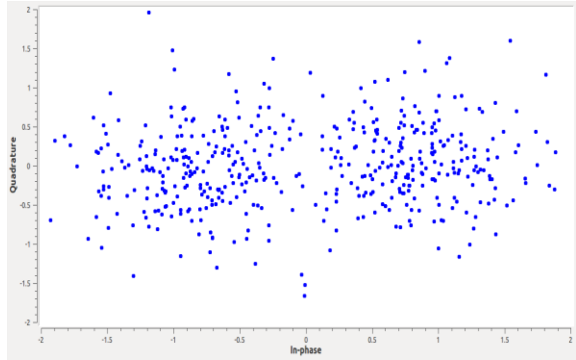
B. *Comparison of FDR for different Channel-Estimation Techniques*

To compare the channel-estimation techniques in real-time transmissions, data were sent repeatedly in the form of frames, and the FDR was observed at the receiver using the ratio of received frames and transmitted frames. Among the four channel estimation methods we tried, these outcomes show that the STA channel estimator offers the best execution. The difference in the algorithms' performance is small in general, but under dynamic channel conditions, STA will perform the best. In this case the receiver and transmitter are in Line of Sight (LOS). But fading may happen even there is a presence of a LOS due to the reflection, scattering etc. of the transmitted signal from the ground and surrounding area objects. The receiving antenna receives a signal which depends on the frequency and bandwidth of the transmission signal propagation, and which can vary widely either in amplitude or even phase. This estimation technique is generally used in IEEE 802.11p protocols, since with vehicle-to-vehicle communications, the relative positions of the transmitter and receiver can change very quickly, which leads to Doppler spread and delay that can exacerbate fading effects. Under such channel conditions, a DDCE-based adaptive channel equalizer must be used.

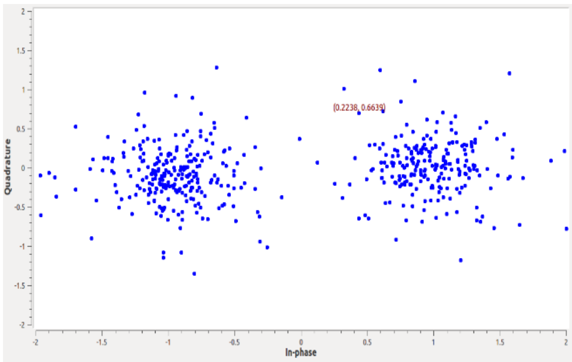
In Fig. 18, one can see the difference between the constellation's plots measured with different channel-estimation techniques in our real-time environment. These four constellation plots were measured with transmissions using BPSK (1/2) modulation, transmitting power of 30 dB, and receiving power of 20dB. Fig. 19 shows FDR plots of data transmitted using the four channel-estimation techniques. These data also show that the STA channel estimator offers the best FDR at low SNR and that it reaches the highest FDR of all channel estimators that we tested.



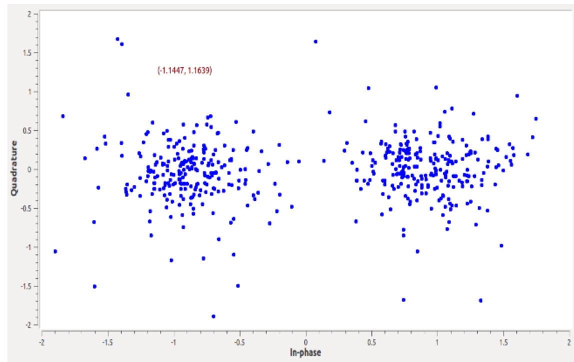
AT 0dB SNR.



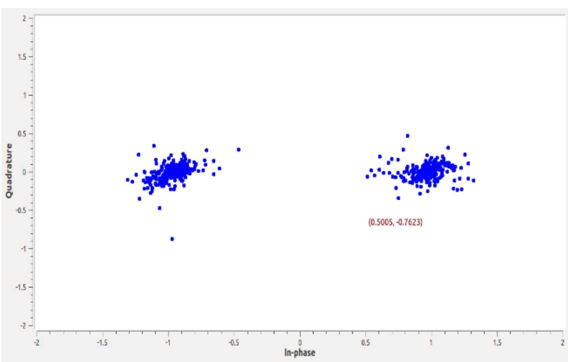
AT 5dB SNR.



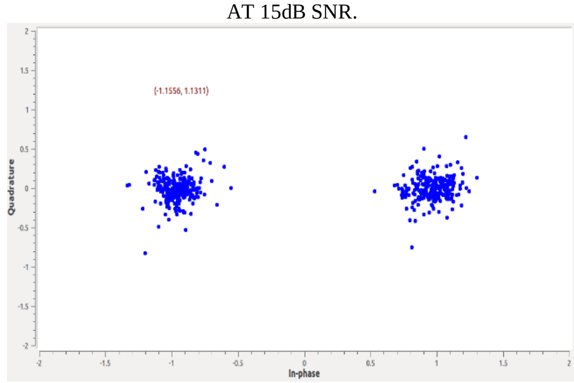
AT 10dB SNR.



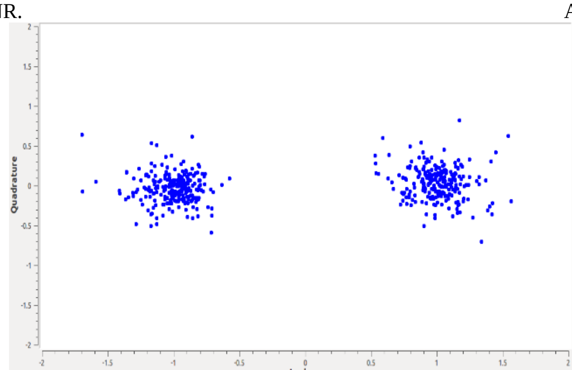
AT 15dB SNR.



AT 20dB SNR.

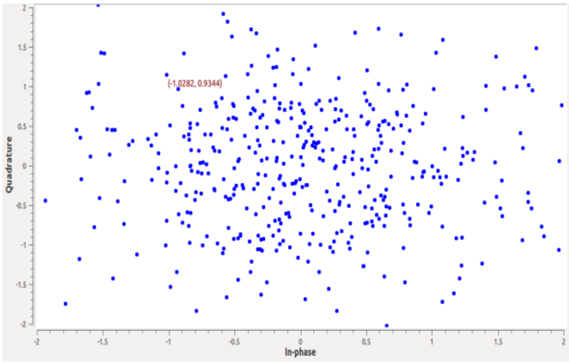


AT 25dB SNR.

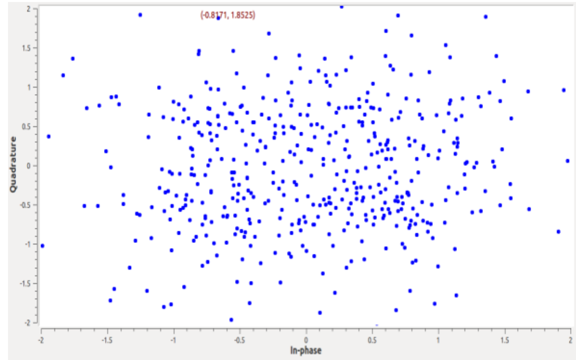


AT 30dB SNR.

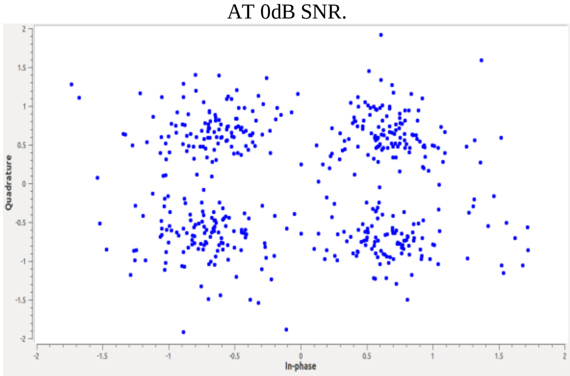
Fig. 12. Constellation Plots for BPSK of SNR from 0 to 30 dB.



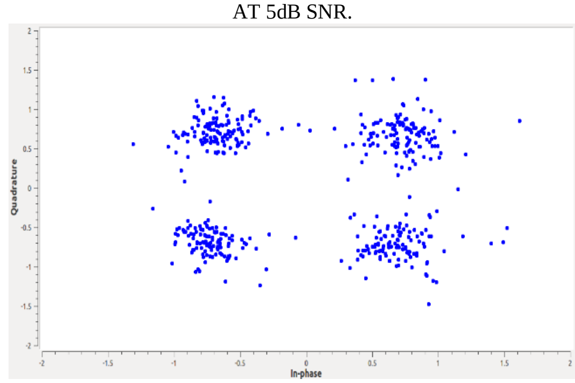
AT 0dB SNR.



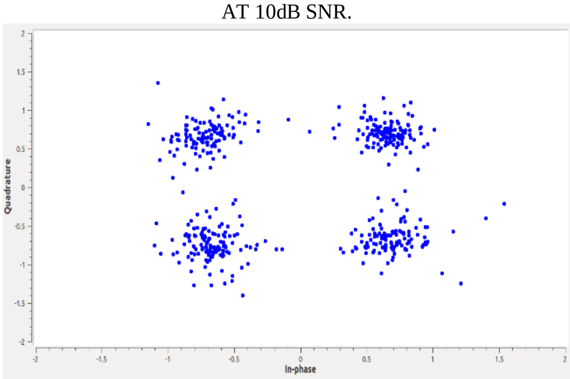
AT 5dB SNR.



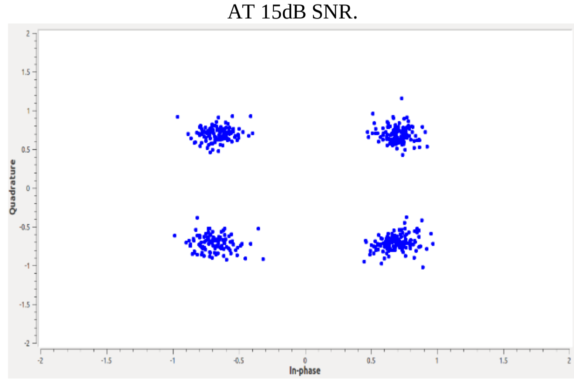
AT 10dB SNR.



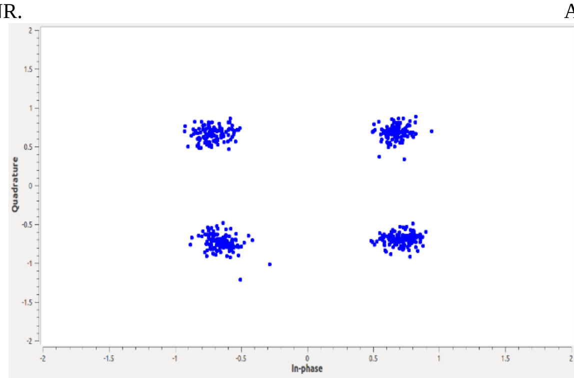
AT 15dB SNR.



AT 20dB SNR.

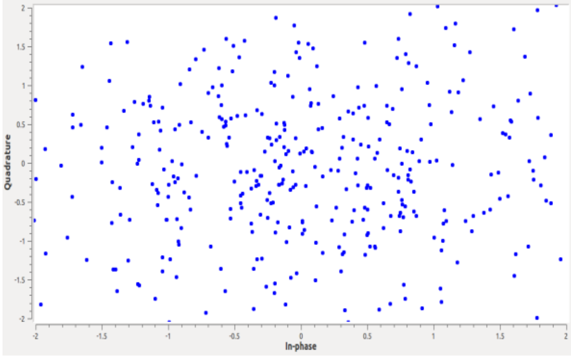


AT 25dB SNR.

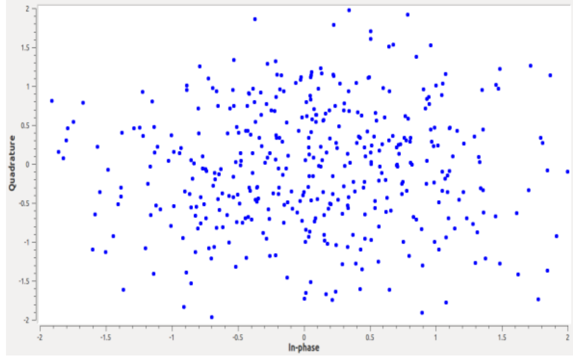


AT 30dB SNR.

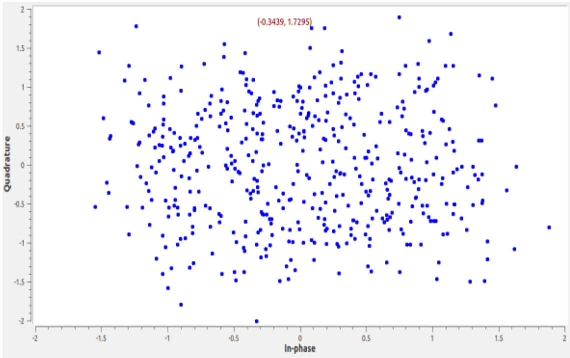
Fig. 13. Constellation Plots for QPSK of SNR from 0 to 30 dB.



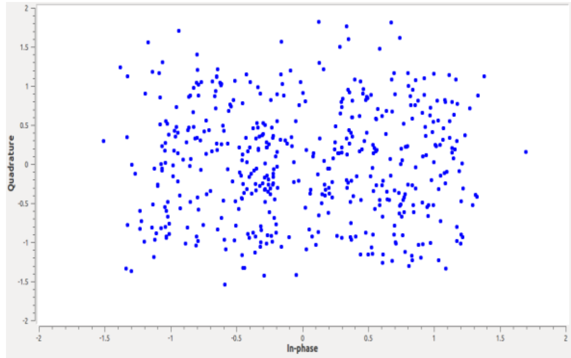
AT 0dB SNR



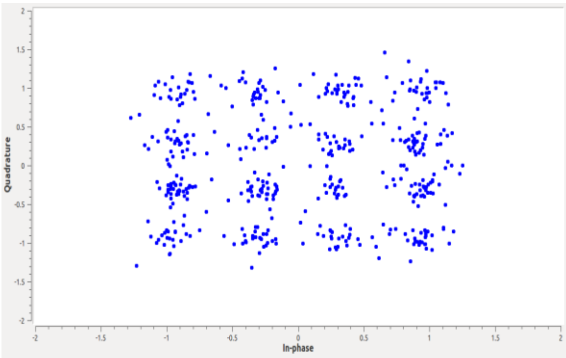
AT 5dB SNR



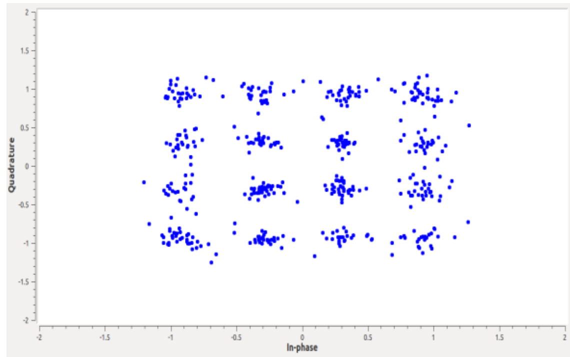
AT 10dB SNR.



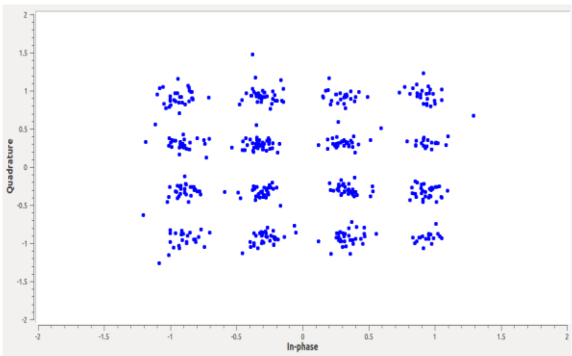
AT 15dB SNR.



AT 20dB SNR.

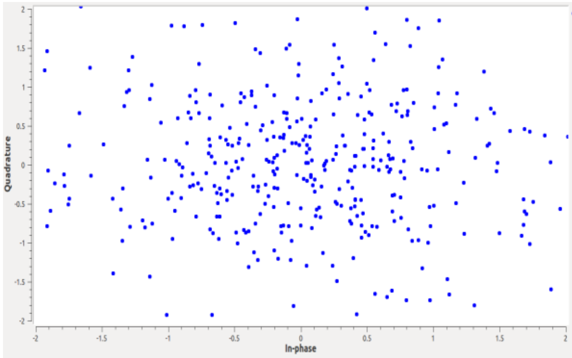


AT 25dB SNR.

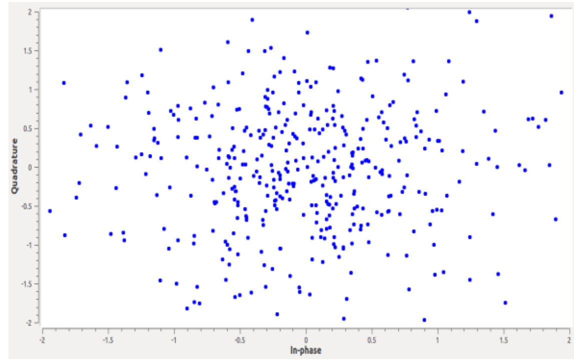


AT 30dB SNR.

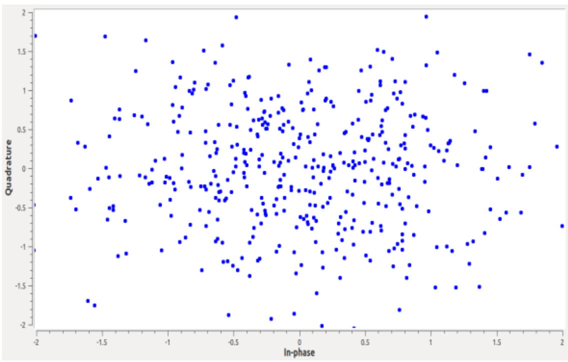
Fig. 14. Constellation Plots for 16QAM of SNR from 0 to 30 dB.



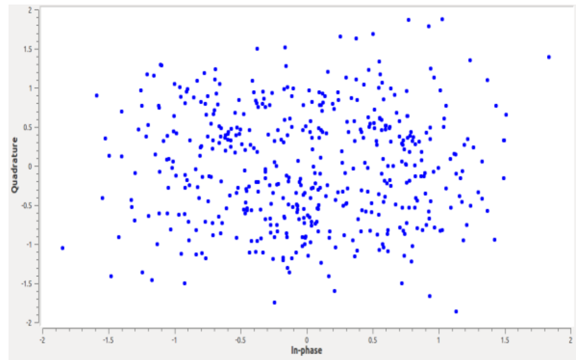
AT 0dB SNR.



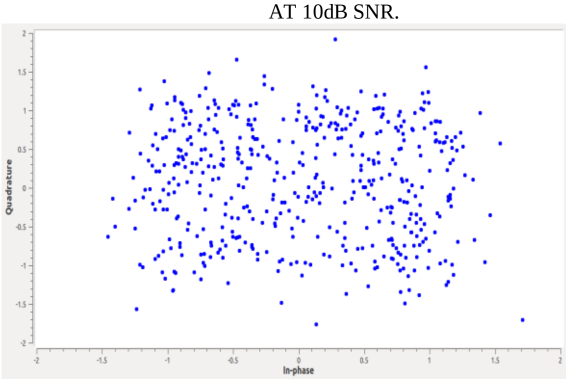
AT 5dB SNR.



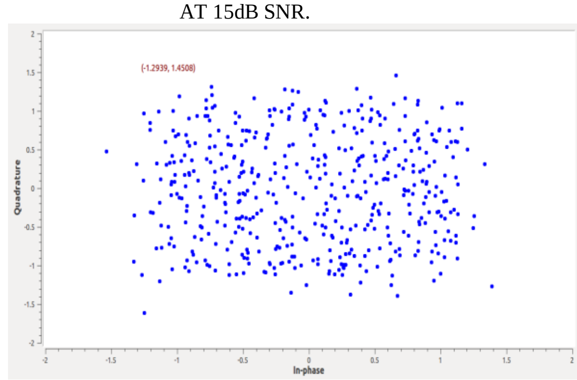
AT 10dB SNR.



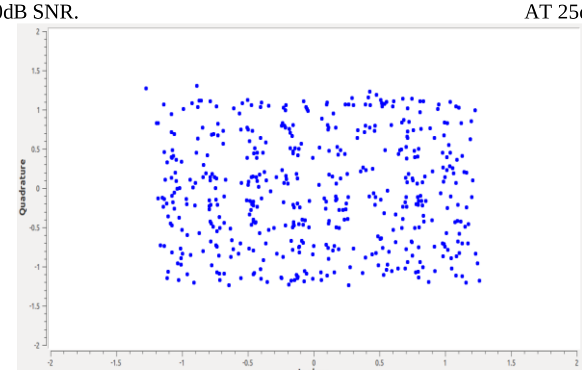
AT 15dB SNR.



AT 20dB SNR.



AT 25dB SNR.



AT 30dB SNR.

Fig. 15. Constellation Plots for 64QAM of SNR from 0 to 30 dB .

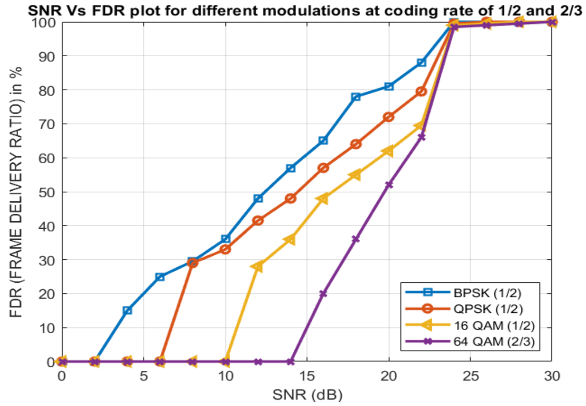


Fig. 16. SNR vs FDR for BPSK (1/2), QPSK (1/2), 16 QAM (1/2), and 64 QAM (2/3) Coding rate.

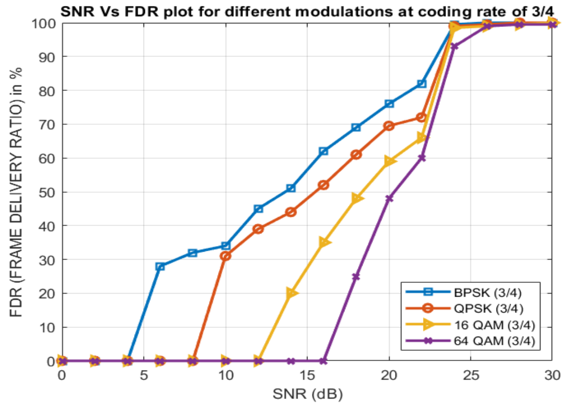
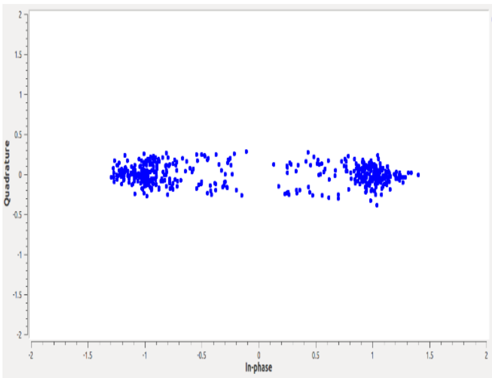
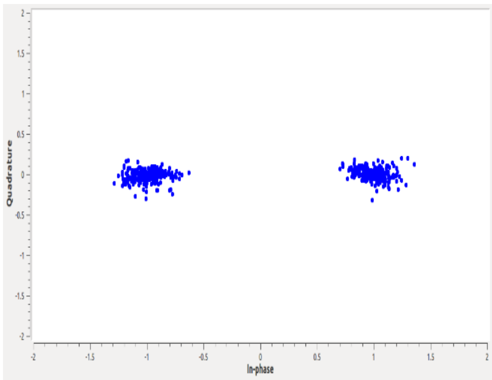


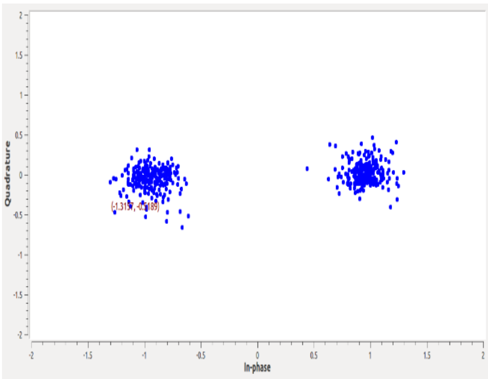
Fig. 17. SNR vs FDR for BPSK, QPSK, 16 QAM, and 64 QAM at ¾ Coding rate.



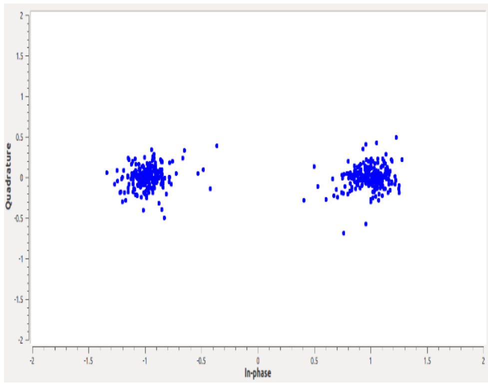
LS Channel Estimation.



Linear Comb Channel Estimation.



LMS Channel Estimation.



STA Channel Estimation.

Fig. 18. Constellation Plots for different Channel-Estimation Techniques.

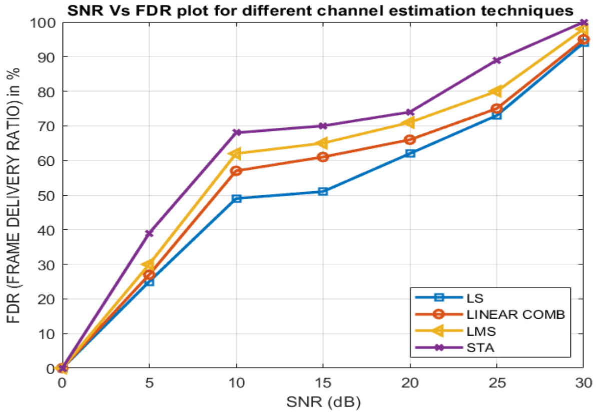


Fig. 19. SNR vs FDR for different Channel-Estimation Techniques.

VI. CONCLUSION

Short coherence time and narrow coherence bandwidth degrade the performance of the physical layer in typical channels. Preamble-based equalization is a traditional scheme, which cannot reimburse for these channel impacts. Data must be used to update the estimates of the channel because preamble-based standards do not sufficiently provide pilot-signal feedback. Thus, improvements in wireless communication system performance depend on channel-estimation techniques. This paper reports tests of channel-estimation techniques in a real-time environment that measured the FDR and proved that the system can provide over 90% of delivery ratio at 25 db of SNR for different digital modulation techniques using STA. Tests were performed with an implementation of the IEEE 802.11a standard protocol using the open-source GRC software to construct a novel SDR testbed that can be used with a wide variety of frontend hardware. Two N210 USRPs were used as frontend transmitter and receiver in our tests. This testbed offers a pathway for the investigation of various parameters in real time, and we performed a series of tests to validate the usefulness of the STA channel estimator in 802.11a networks. The results clearly show that STA outperforms other schemes, with the FDR clearly higher than the other four estimators that we tested.

REFERENCES

[1] Albreem MAM. “5G wireless communication systems: vision and challenges”. I4C, pp. 493–97, April 2015.

[2] Lee J, Ryu J, Lee S-J, Kwon T. “Improved modeling of IEEE 802.11a PHY through fine-grained measurements”, Computer Networks: The International Journal of Computer and Telecommunications Networking, Vol. 64, No.4, pp.641–5, March 2010.

[3] Giddings RP, Jin XQ, Tang JM. “First experimental demonstration of 6Gb/s real time optical OFDM transceivers incorporating channel estimation and variable power loading”. Optics Express, Vol.17, No.22, pp. 19727-19738, 2009.

[4] Jin XQ, Giddings RP, Tang JM. “Real-time transmission of 3Gb/s 16-QAM encoded optical OFDM signals over 75km SMFs with negative power penalties”. Optical Express, Vol.17, No.17, pp. 14574-14585, 2009.

[5] Li R, Dou Y, Zhou J, Deng L, Wang S. CuSora: “Real-time software radio using multi-core graphics processing unit”. Journal of Systems Architecture, Vol. 60, No.3, pp. 280-292, March 2014.

[6] Armour S, Nix A, Bul D. “Complexity evaluation for the implementation of a Pre-FFT Equalizer in an OFDM receiver”. IEEE Transactions on Consumer Electronics Vol. 46, No.3, pp. 428-437, June 2000.

[7] Pozza M, Rao A, Flinck H, Tarkoma S. “Network-in-a-box: a survey about on-demand flexible networks”. IEEE Communications Surveys & Tutorials, Vol. 20, No.3, pp. 2407-2428, February 2018.

[8] W. H. Tuttlebee, Software Defined Radio: Enabling Technologies. John Wiley & Sons, 2003.

[9] Luo H, Wu K, Ruby R, Liang Y, Guo Z, Ni LM. “Software-defined architectures and technologies for underwater wireless sensor networks: a survey”. IEEE Communications Surveys & Tutorials, Vol. 20, No.4, pp. 2855-2888, May 2018.

[10] Cai X, Zhou M, Xia T, Fong WH, Lee W-T, Huang X. “Low-power SDR design on an FPGA for intersatellite communications”. IEEE Transactions on Very Large-Scale Integration (VLSI) System, Vol. 26, No.11, pp. 2419–2430, July 2018.

[11] Haithem Haggui, Sofiène Affes and Faouzi Bellili. “FPGA-SDR Integration and Experimental Validation of a Joint DA ML SNR and Doppler Spread Estimator for 5G Cognitive Transceivers”, IEEE Access, Vol. 7, pp. 69464-69480, May 2019.

[12] Per Zetterberg and Ramin Fardi, “Open Source SDR Frontend and Measurements for 60-GHz Wireless Experimentation”, IEEE Access, Vol. 3, pp. 445-456, May 2015.

[13] Xintong Lu, Luyao Ni, Shi Jin, Chao-Kai Wen and Wen-Jun Lu, “SDR Implementation of a Real-Time Testbed for Future Multi-Antenna Smartphone Applications”, IEEE Access, Vol. 5, pp. 19761-19772, October 2017.

[14] Muhammad Bilal Khan, Xiaodong Yang, Aifeng Ren, Mohammed Ali Mohammed Al-Hababi, Nan Zhao, Lei Guan, Dou Fan and Syed Aziz Shah, “Design of Software Defined Radios Based Platform for Activity Recognition”, IEEE Access, Vol.7, pp. 31083-31088, March 2019.

[15] Christos Politis, Sina Maleki, Juan Merlano Duncan, Jevgenij Krivochiza, Symeon Chatzinotas and Björn Ottesten, “SDR Implementation of a Testbed for Real-Time Interference Detection with Signal Cancellation”, IEEE Access, Vol.6, pp. 20807-20281, May 2018.

[16] Leonardo A. Agüero Guzmán, Elias M. Ovalle, and Rodrigo A. Reeves, “Measurement of the ionospheric reflection height of an HF wave in vertical incidence with a resolution of minutes”, IEEE Geoscience And Remote Sensing Letters, Vol. 15, No.11, pp. 1637–1641, November 2018.

[17] Wei Wang, Yinejie Chen, Zeyu Wang, Jin Zhang, Kaishun Wu, and Qian Zhang, “Wideband Spectrum Adaptation Without Coordination” IEEE Transactions on Mobile Computing, Vol. 16, No.1, pp. 243-256, January 2017.

[18] Wang W, Chen Y, Wang L, Zhang Q, “Sample less Wi-Fi: bringing low power to Wi-Fi communications”, IEEE/ACM Transactions on Networking, Vol. 25, No.3, pp. 1663-1672, June 2017.

[19] Suman Bhunia, Edward Miles, Shamik Sengupta, and Felisa Vazquez-Abad, “CR-Honeynet: A Cognitive Radio Learning and Decoy Based Sustenance Mechanism to Avoid Intelligent Jammer”, IEEE Transactions on Cognitive Communications and Networking, Vol. 4, No.3, pp. 567-581, September 2018.

[20] Song Fang, Yao Liu, and Peng Ning, “Wireless Communications under Broadband Reactive Jamming Attacks”, IEEE Transactions on Dependable and Secure Computing, Vol. 13, No.3, pp. 394-408, June 2016.

[21] Cassola A, Jin T, Noubir G, Thapa B “Efficient spread spectrum communication without pre-shared secrets”, IEEE Transactions on Mobile Computing, Vol. 12, No.8, pp. 1669-1680, 2013.

[22] Vilches T, Dujovne D, “GNURadio and 802.11: performance evaluation and limitations”, IEEE Network, Vol. 28, No.5, pp. 27-31, October 2014.

[23] Horrein P-H,·Hennebert C, Pétrot F, “Integration of GPU computing in a software radio environment”, Journal of Signal Processing Systems, Vol. 69, No.1, pp. 55-65, October 2012.

[24] Lee S, Choi J, Yoo J, Kim C-K, “Frequency diversity-aware Wi-Fi using OFDM-based bloom filters”, IEEE Transactions On Mobile Computing Vol. 14, No.3, pp. 525-537, March 2015.

[25] Ponnaluru Sowjanya, Penke Satyanarayana, "Real-Time Data Transfer Based on Software Defined Radio Technique using Gnu radio/USRP", International Journal of Engineering and Advanced Technology, Vol. 9, No. 1, pp. 279-288, October 2019.

[26] Fernandez JA, Stancil DD, Bai F, "Dynamic Channel Equalization for IEEE 802.11p Waveforms in the Vehicle-to-Vehicle Channel". In 48th Annual Allerton Conference on Communication, Control, and Computing (Allerton), Monticello, IEEE, 542-551 2010.

[27] Kim H. Wireless Communications Systems Design. John Wiley & Sons; 2015.

[28] Van Nee R, Prasad R. OFDM wireless multimedia communications. Boston: Artech House; 2000.

[29] Bloessl B, Segata M, Sommer C, Dressler F, "Performance assessment of IEEE 802.11p with an open-source SDR-based prototype", IEEE Transactions on Mobile Computing, Vol. 17, No.5, pp. 1162-1175, May 2018.

[30] Ozdemir MK, Arslan H. "Channel estimation for wireless OFDM systems", IEEE Communications Surveys and Tutorials, Vol. 9, No.2, pp. 18-48, June 2007.

[31] IEEE, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," IEEE, STD 802.11-2016, Dec. 2016.

AUTHOR'S PROFILE

Ponnaluru Sowjanya completed her B.Tech (Electronics and Communication Engineering) in Sri Mittaplli Institute of Technology for Women, Tummalapalem, Guntur, India; M.Tech (Embedded Systems) in Nalanda Institute of Engineering and Technology, Kantepudi, Satenepalli, India. At present, she is pursuing a Ph.D. in Electronics and Communication Engineering at Koneru Lakshmaiah Educational Foundation, Vaddeswaram, India. Her research interest includes Wireless Communication and Signal processing.

Dr. Penke Satyanarayana received a B.Tech in Electronics and Communication Engineering (ECE) from Koneru Lakshmaiah College of Engineering, Vaddeswaram, Vijayawada, India; M.Tech in Computers and Communications Systems from JNTU, Hyderabad, India; and Ph.D. in ECE from JNTU, Kakinada, India. He is working as a Professor in the Department of ECE, K L E F, Guntur, India. He has published over 20 research papers in reputed international and national journals and conferences. He is a member in Indian Society of Technical Education (ISTE). His research interests include Wireless Communication and Signal processing, Embedded Systems and VLSI.