INNOVATIVE COMMUNICATION DESIGN LAB BASED ON PC SOUND CARD AND MATLAB : A SOFTWARE-DEFINED-RADIO OFDM MODEM EXAMPLE

Jeng-Kuang Hwang

Inst. of Communication Eng.
Yuan-Ze University
Chung-Li City, Taiwan 32026
Email : eejhwang@saturn.yzu.edu.tw

Abstract

In this paper, an innovative software-defined radio (SDR) approach to advanced communication design lab courseware is proposed and illustrated by an audio-band OFDM transceiver design example. To implement this new approach, it only needs the most common PC sound card and Matlab software, making such a lab course extremely easy to setup at a minimum cost. Moreover, all system parameters and algorithms are embodied in the form of high-level RX and TX programs, which can be modified and tested very flexibly. In this aspect, this approach has all the advantages of simulation-based experiments. However, it further touches the real-world implementational issues. A real-world passband signal in audio band can be physically generated and simultaneously recorded by the full-duplex sound card, and then the received digitized signal can be demodulated by an SDR receiver which implements all the required DSP algorithms in Matlab code.

I. Introduction

Nowadays, the technology renewal pace in the field of wireless communications has become faster and faster, and thus a strong educational need for the university teacher is to show the students how the communication principles can be easily and quickly applied to realistic design and implementation of some experimental communication systems. To this goal, it calls for an innovative communication system design lab course which can clearly illustrate the complete canonical top-down system design flow from setting the specifications, design of block diagrams and algorithms, computer simulation and performance evaluation, HW/SW implementations, to system test and verification. At a first glance, to set up such a design lab course may often cost a lot of budget by resorting to existing commercially available educational equipments. Besides, many existing lab materials are usually presented in an inflexible and less-creative style that can hardly cultivate the students to gain a system designer's experiences, ranging from setting up the system specifications and parameters, system architecture design, modeling and computer simulation, DSP implementation, to system integration. Thus, the above existing approach is not cost-effective for training a system designer.

Aware of the above needs as well as the budget problem, we have undertaken a novel approach at Yuan-Ze Univ. to develop a communication design lab courseware for our EE senior and graduate students. It is attempted to achieve all the above educational purposes, while the equipment requirement is so little which only consists of an ordinary PC with sound card support and the basic Matlab package for SDR development, as is shown in Fig.1. To be more specific, this courseware has the following merits :

1. Minimum budget and equipments : Only one PC with a student-version Matlab package is required.
2. Broadest coverage of topics : The new lab materials can cover the designs and functions of virtually unlimited signal processing blocks for generic point-to-point communication link, such as various modulation formats, coding schemes, baseband and passband processing, synchronization techniques, digital receiver design, and channel models.
3. Greatest flexibility : All system parameters can be adjusted in the TX/RX Matlab programs, and the resulting effect can be readily investigated. Hence we also refer to the approach as a kind of software-defined radio (SDR) lab, and this is highly coincides with the technology trend [1].
4. Highest instruction efficiency: The students can gain a complete understanding of the top-down design approach, and link the textbook principles to practice. Furthermore, by generate and record real-world audio-band modulated signals, the students gain a sense of realistic engineering problems.
5. Newest technology taught : For example, an advanced SDR OFDM modem lab will be demonstrated in Section II.
6. Real-time receiver option provided: If the Matlab S/W includes a Data Acquisition Toolbox (Daq Tbx) [2], then real-time receiver experiment can be performed by using the real-time data logging capability of the toolbox.

Currently, a lot of illustrative labs have been developed or are under development, mainly in the field of digital communications. There are SDR modems for many different modulation formats, such as standard AM, AM-DSB, FM, Grayed-coded QPSK, π/4-DQPSK, 8-PSK, 16-QAM, GMSK based on linearized PAM model [3], spread-spectrum, and OFDM modulation formats. In regard to the synchronization issue, there are two possible operation modes. If continuous transmission is considered, then we often use non-data-aided (NDA) synchronization techniques such as recursive digital Costas loop for carrier synchronization, and early-late gate for symbol timing synchronization [4]. On the other hand, if burst-mode transmission with a known preamble is under consideration, we often first use a double sliding-window method to detect the start of a burst, and then exploit the

preamble part by using correlation-based techniques for symbol timing synchronization and carrier offset estimation [5]. Besides, channel estimation and equalization can also be studied if some dispersive channel is emulated at the TX side.

II. Design of the SDR OFDM Modem

A. System Block Diagram Design

Fig.2 shows the block diagram of our audio-band OFDM modem, which lends itself from the PHY layer of IEEE 802.11a OFDM WLAN TX/RX [6]. In this lab, students can not only learn all the basic principles behind the OFDM system, but can also get familiar with the practices of a real-world commercial standard. Once they leave the school, such experience will be highly valuable in industry R/D works.

B. Specifications and SDR OFDM Transmitter Design

This lab begins with specifying the system specifications and parameters. By referring to the general rules of OFDM system design, we have written an OFDM design program to find the following specifications such as to match the sound card :

1. System parameters Setting

- Subcarrier spacing : Δ_f =62.5000 Hz
- No. of subcarriers (without the DC term): N_d =24
- No. of FFT points : N_{fft} =32
- OFDM Data duration : T_d =1/d_f=0.0160 s
- Cyclic guard time : T_g =T_d/4=0.0040 s
- OFDM symbol (block) duration : T_{ofdm}=T_d+T_d=0.02 s
- Passband bandwidth : B =N_d*d_f=1500 Hz
- Passband span: [1000-750 1000+750]= 250 ~1750 Hz
- Subcarrier Moduation : $S_k \in \{\pm 1/\sqrt{2} \pm j/\sqrt{2}\}$ (QPSK)
- Overall bit rate : R_b=2×N_d/T_{ofdm}=2400 bps
- IFFT output sampling period: T_s=T_d/N_{fft} =0.5ms
- Oversampling ratio : OVR=4
- Interpolation sampling period:T_{s2} =T_s/OVR= 0.125ms
- Waveform sampling rate : f_{s2}=1/T_{s2}=8000sps (This matches to the standard sampling rate of PC sound card.)
- Roll-off factor of Nyquist Interpolation filter : α=0.15
- Carrier frequency : f_c =1000 Hz

2. Packet Structure Design

Simplified from the IEEE 802.11a packet, our packet (time slot) structure is shown in Fig.3, which includes the preamble and payload data fields. The preamble consists of 10 short training symbols (t_1~t_{10}) of 8 ms each, two long training symbols (T_1 and T_2) of 32ms each, and a cyclic prefix (GI2) of 16ms for the long training symbols, making the total preamble duration being 160ms. The preamble is designed to help a lot of synchronization tasks stated below :

- t_1 ~ t_7 : packet start detection, AGC
- t_8 ~ t_{10} : symbol timing synchronization and coarse carrier frequency offset estimation
- GI2 and T_1 ~ T_2 : fine frequency offset estimation and channel estimation

The individual short training symbol and long-training symbol are chosen as :

$P_{-12:12}=\sqrt{5/3}\times$[-1-j 0 0 0 1-j 0 0 0 -1+j 0 0 0 0 0 0 -1+j 0 0 0 1-j 0 0 0 1+j]; (1)

$L_{-12:12}=\sqrt{2/3}\times$[1 -1 -1 1 1 -1 1 -1 1 1 1 1 0 1 -1 -1 1 1 -1 1 -1 1 -1 -1 -1]; (2)

respectively, where each symbol has 25 elements, in which the middle zero element is the additional null DC subcarrier. Moreover, their corresponding baseband samples can be generated according to the following equations :

$$r_{SHORT}(nT_s) = \sum_{k=-N_d/2}^{N_d/2} P_k \exp(j2\pi k\Delta_f nT_s), \quad n = 0,1,..,15 \qquad (3)$$

$$r_{LONG}(nT_s) = \sum_{k=-N_d/2}^{N_d/2} L_k \exp(j2\pi k\Delta_f nT_s), \quad n = 0,1,..,64 \qquad (4)$$

Note that the sample spacing of the above waveforms is T_s=0.5ms, so the preamble part has a total length of 320 complex (I/Q) samples. After the preamble part, we append the data part of 100 OFDM symbols, each of 20 ms. Thus the overall packet duration is T_{pt}=160 + 20×100= 2160 ms, in which a total of 100×24×2=4800 bits is transmitted.

3. Multipath Channel Emulation and I/Q Modulation

To fully test the OFDM receiver, a dispersive channel effect is emulated on the I/Q samples in the above packet at the TX side. Here we use a static two-ray multipath channel having the impulse response as :

$$h(t) = A_1\delta(t-\tau_1) + A_2\,\delta(t-\tau_2) \qquad (5)$$

where the delay spread $\tau_d = \tau_2 - \tau_1$ is less than the guard time T_g. Then we apply a root Nyquist interpolation filter for up-sampling both the channel output samples $\{s_I(nT_s), s_Q(nT_s)\}$ by a factor of four, and use a quadrature modulator to up covert the I/Q waveforms to the carrier frequency of 1000 Hz, resulting in an audio, passband, and channel-included, transmitted signal as follows :

$$s_T(nT_{s2}) = s_I(nT_{s2})\cos(2\pi f_c nT_{s2}) + s_Q(nT_{s2})\sin(2\pi f_c nT_{s2}) \qquad (6)$$

Finally, after adding some AWGN noise of power σ^2 to the passband signal, an audible signal `s_tx` is sent out to the output SPK jack by using a simple Matlab statement :

```
sound(s_tx, 8000);
```

which play the software-generated transmitted signal at a basic sampling rate of 8000 sps. Through a simple extension wire, the output signal is then connected back to the LINE IN jack of the sound card.

C. SDR OFDM Receiver Design

1. Signal recording and digital I/Q down-conversion

To record the transmitted signal with the same PC, another Matlab session should be opened for running the receiver program. Since the sound card is full duplex, we can record the TX signal for 4 seconds and store to data to an array `sig_rec` by the following Matlab statements :

```
r=audiorecord(8000,16,1); %open the recorder
r.record;                 % begin recording
pause(4);                 % last for 4 sec
r.stop;                   % stop recording
sig_rec=r.getaudiodata('single');%get buffer data
```

After recording the signal in digital format, we perform digital down-conversion by using the RX local cos and sin carriers at

frequency f_{cr} and a matched root Nyquist filter for low pass filtering, resulting in a baseband complex received signal $x(nT_{s2})= x_I(nT_{s2}) + jx_Q(nT_{s2})$. Then the output signal is decimated by four. Finally we have a sampled signal which can be modeled as

$$x(nT_s) = s_T(nT_s)e^{j(2\pi f_o nT_s+\theta)} + w_o(nT_s) \quad (7)$$

where $s_T(nT_s) = s_I(nT_s)+js_Q(nT_s)$ represents the complex envelope samples of the passband signal, $w_o(nT_s)$ is the sampled discrete-time AWGN noise, θ is a random phase due to time delay, and $f_o= f_{cr}- f_c$ is the carrier frequency offset.

2. Packet detection and symbol timing estimate

The packet detection is performed by a simple double-sliding window method [5]. First, the pre- and post-window outputs are computed as

$$a(nT_s)=\sum_{m=0}^{M-1} x((n-m)T_s)x^*((n-m)T_s) \quad (8)$$

$$b(nT_s)=\sum_{m=0}^{M-1} x((n+m)T_s)x^*((n+m)T_s) \quad (9)$$

respectively. Then the decision variable is forms as

$$m(nT_s)= b(nT_s)/a(nT_s) \quad (10)$$

If the decision variable exceeds a given threshold, then the presence of the packet is detected and the peak of the $m(nT_s)$ can give the estimate of packet start position.

If the above estimate is to be further refined for symbol timing, we can calculate the cross-correlation between the received signal $x(nT_s)$ and a known reference $g(nT_s)$; for example, the t_8-t_{10} segment. This yields

$$\hat{t}_s = \arg\max_n \left|\sum_{k=0}^{L-1} x((n+k)T_s)g^*(kT_s)\right|^2 \quad (11)$$

as the sample index at which the reference segment begins.

3. Carrier frequency synchronization

To estimate the carrier frequency offset f_o, the known preamble structure can be exploited [5]. Let D be the delay between the identical samples of two repeated training symbols, either short or long. Then we find the auto-correlation of the received signal at the sample lag D :

$$Z=\sum_{m=0}^{L-1} x(nT_s)x^*((n+D)T_s) \quad (12)$$

Under noise-free situation, it can be easily shown from eq.(7) that the following relation holds

$$Z = e^{-j2\pi f_o DT_s}\sum_{m=0}^{L-1}\left|s_T(mT_s)\right|^2 \quad (13)$$

Hence a frequency offset estimator is given by

$$\hat{f}_o = -\frac{1}{2\pi DT_s}\angle Z \quad (14)$$

It should be noted here that OFDM receiver is quite sensitive to the residual error in frequency offset estimation. To maintain a good performance for rather long data field, pilot subcarriers may be embedded to the packet format to help subsequent carrier phase tracking, and this is adopted in the 802.11a standard [6].

4. Channel estimation and channel equalization

From the received data $x(nT_s)$, let $\mathbf{x}_{C1}$ and $\mathbf{x}_{C2}$ denote the two 1×64 arrays (data segments) corresponding to the two long training symbols. Then the channel estimate can be found easily by first taking the FFT of $\mathbf{x}_{C1} + \mathbf{x}_{C2}$, yielding

$$\mathbf{R} = \text{FFT}(\mathbf{x}_{C1} + \mathbf{x}_{C2}) \quad (15)$$

Then we have

$$R_k = H_k L_k + W_{1,k} + H_k L_k + W_{2,k} \quad \text{k=-12,…,12} \quad (16)$$

where H_k is the channel response of subcarrier k. Since $|L_k|=1$, the channel estimate of H_k can be readily obtained as

$$\hat{H}_k = R_k L_k^* / 2 \quad \text{k=-12, …,12} \quad (17)$$

Then for each of the subsequent OFDM block, we first remove the CP, and then take a 32-point FFT to get

$$X_k = H_k S_k + W_k \quad (18)$$

where W_k is the noise terms. Thus the QPSK symbol S_k can be detected in terms of a one-tap ZF equalizer and a slicer as

$$\hat{S}_k = slicer\left\{ X_k / \hat{H}_k \right\} \quad (19)$$

Finally, a demapping from the detected QPSK symbol sequence to bit sequence completes our SDR OFDM receiver operation.

III. A Real-World Experiment

In this experiment, we set $A_1= 1e^{j\pi/4}$, $A_2= 0.5e^{-j\pi/3}$, SNR=30 dB, $\tau_d= 0.5T_g$ = 0.002s in the TX program, and one packet is played via the sound card. In the RX program, we set f_{cr}=1010 Hz to give an artificial frequency offset of f_o=10 Hz. Fig.4 shows the short- and long-training waveforms. Fig.5 shows the FFT magnitude spectrum of the passband audio signal. After receiving, figure 6 shows the packet detection result, Fig.7 shows the channel estimation result, and finally Fig.8 shows the equalized QPSK symbol estimates, with the frequency offset estimate being 10.0045 Hz. In this case, all the QPSK symbols and corresponding bits were correctly detected.

IV. Conclusions

In this paper, we have proposed an innovative software-defined radio (SDR) approach to design-oriented communication lab courseware. A sample experiment of audio-band OFDM modem design is used to illustrate the many merits of the proposed approach. As a summary, such a courseware can not only meet the current diverse educational and industrial requirements, but is also very cost-effective, efficient, instructive, flexible, and fruitful, as has been proven at YZU. Furthermore, some enhancements are possible. For example, if the Matlab software is equipped with a DAQ toolbox, then even real-time SDR receiver can be implemented and tested. On the other hand, a good follow-up lab course can be more focused on real-time implementation of the SDR modems using such tools as TI C6x DSK boards and CCS IDE.

References :

[1] W. Tuttlebee Ed. : *Software Defined Radio*, John Wiley, 2002.

[2] The Matlab Data Acquisition Toolbox Manual, MathWorks Inc., 2001.

[3] J. G. Proakis : *Digital Communications*, 4th Ed., McGraw-Hill, 2000

[4] H. Myer et al : *Digital Communication Receivers*, Wiley, 1998.

[5] J. Terry and J. Heiskala : *OFDM Wireless LANs*, SAMS Inc., 2002.

[6] IEEE Standard 802.11a : Pt. 11- Wireless LAN MAC and PHY Specifications in the 5 GHz band, 2000.

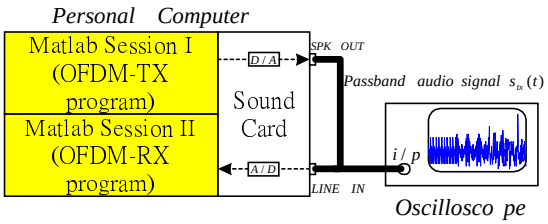

Fig.1 The setup of the SDR audio-band OFDM modem lab with only a single PC

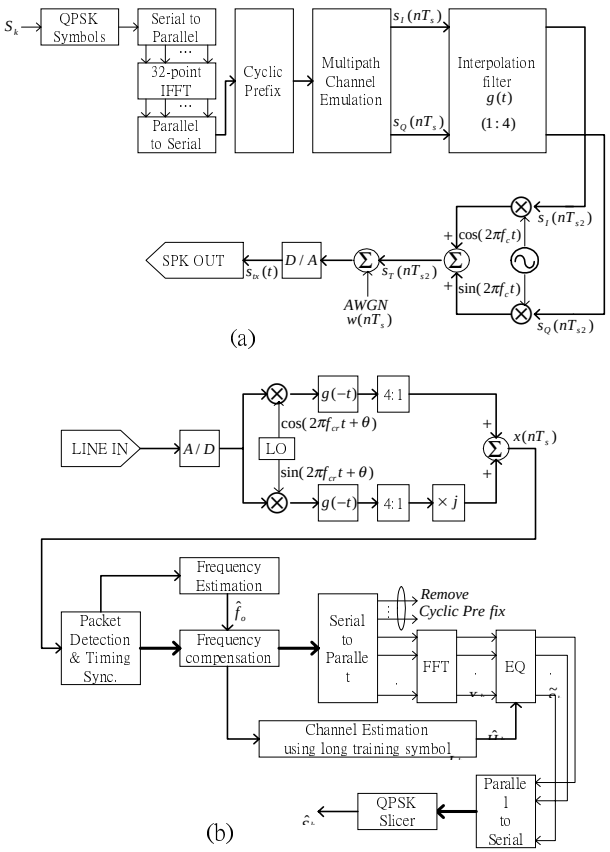

Fig.3 The Block diagrams of : (a) OFDM TX, (b) OFDM RX

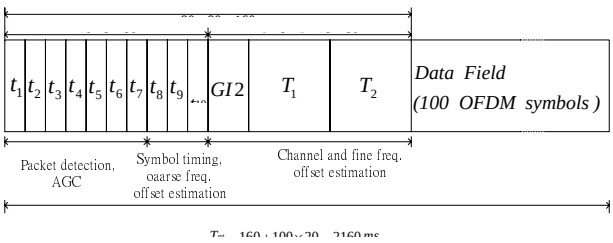

Fig.4 The packet structure, including the preamble field and payload data field

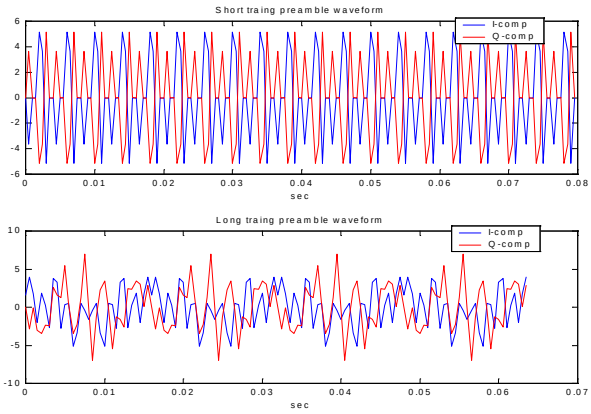

Fig.4 The preamble waveform: short- and long-training

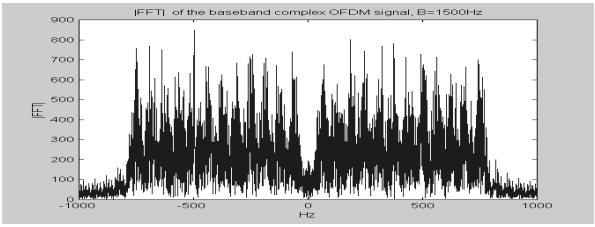

Fig.5 FFT magnitude spectrum of the baseband OFDM signal

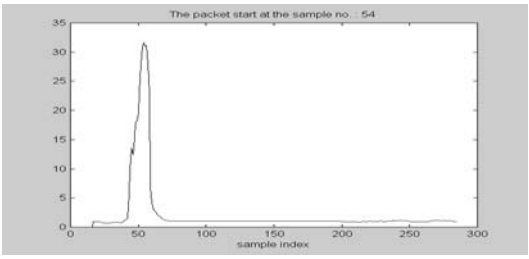

Fig. 6 Detection of packet start using double-sliding window techniques

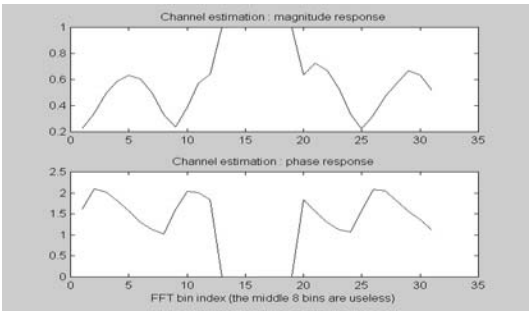

Fig.7 Channel estimation for the two-ray static multipath channel emulated at TX

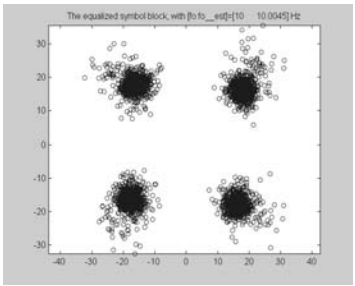

Fig.8 The equalized OFDM-QPSK constellation