

DESIGN OF A NON-DATA-AIDED DIFFERENTIAL-QAM MODEM FOR REAL-TIME SPEECH COMMUNICATION: A SOFTWARE-DEFINED-RADIO PC-BASED APPROACH

Jeng-Kuang Hwang*, Yu-Lun Chiu, and Rih Lung Chung

Institute of Communication Eng., Yuan-Ze Univ.

Chung-Li City, Taiwan 32026, R.O.C.

*Email : eejhwang@saturn.yzu.edu.tw

ABSTRACT

In this paper, a non-data-aided differential-QAM (NDA-DQAM) system is proposed to solve the phase ambiguity problem which arises in continuous speech transmission. Moreover, the system design and test are done by the software-defined radio (SDR) approach which is based only on the PC soundcard and Matlab[®] software for a minimum-cost implementation of the NDA-DQAM system. In so doing, first, the speech signal is converted into bit stream by the adaptive delta modulation. Then using two differential angles mapped by Gray-coded dibits, a passband 16-DQAM waveform can be generated. It is shown that the 16-DQAM system incurs only about 0.5-dB performance degradation as compared to the coherent 16-QAM system. With the SDR approach, all the system parameters and TX/RX algorithms are embodied in high-level Matlab programs. Especially, a free-running T/2-spaced receiver structure is used with interpolation-based timing recovery and Costas carrier synchronization. Finally, experimental results are included to show how real-time speech transmission can be readily achieved. Hence this paper also suggest an efficient and cost-effective approach to teach a design-oriented communication lab course.

I. INTRODUCTION

Among various digital modulation schemes, quadrature amplitude keying (QAM) is one of the most often adopted scheme in various standards. However, considering a situation where continuous non-data-aided transmission of speech signal is desired without using the preamble or unique word, ordinary QAM will suffer from the phase ambiguity problem. Although such a problem can be easily tackled by applying differential coding to PSK-type modulation or star-QAM [1], it is seldom studied for the QAM signal with square constellation. This motivates us to devise the so called NDA-DQAM modulation scheme proposed in this paper.

Besides the DQAM coding and decoding schemes, a complete SDR design and illustrative PC-based implementation of the proposed system are also presented, which follows the canonical top-down system design flow that starts from setting the specifications, design of block diagrams and algorithms, computer simulation and performance evaluation, HW/SW implementations, to overall system test and verification. For instructional purpose, we only employ the PC soundcard and Matlab software as the SDR implementation platform. This

makes the system integration, testing, and verification process rather easily done at the minimum cost, and therefore such a system can serve as a highly valuable example for an innovative communication design lab courseware [2-3].

This paper is organized as follows. In Section II, the differential encoding and decoding schemes of the NDA-DQAM system is described in terms of a Solar System analogy. In Section III, we present the top-down SDR design and PC-based implementation of the NDA-DQAM transceiver. Section IV includes some real-world experimental results. Finally, some conclusions are made in Section V.

II. DIFFERENTIAL CODING AND DECODING FOR THE NDA-DQAM SYSTEM

In considering the tradeoff between bandwidth efficiency and BER performance, the M-QAM modem is seen as a good candidate for real-time voice transmission. When it is equipped with the non-data-aided (NDA) synchronizer, the phase ambiguity problem must be solved in order not to breaking down the BER performance. In this section, we propose a Differential-QAM modulation scheme which uses K/2 differential angles to represent the transition of consecutive signal points, where $K=\log_2 M$ is the number of bits per symbol. It is also shown that such a scheme can also be uniquely demodulated by a reverse differential decoding. In the following, the coding and decoding processes are explained by taking the 16-DQAM constellation as an example.

A. Differential Encoding for DQAM System

Here we describe the method to generate the 16-DQAM symbol from a sequence of input bits. For $M=16$, a group of four bits is to be mapped into a symbol. The 16-DQAM symbol is constructed by two differential angles $\Delta\theta_1$ and $\Delta\theta_2 \in \{0, \pi/2, 3\pi/2, \pi\}$, where $\Delta\theta_1$ is determined by the first two-bit part (dibit #1) of the 4-bit symbol, and $\Delta\theta_2$ is determined by the last two-bit part (dibit #2). Listed in Table 1 is the relation between the dibit (b_1, b_2) and the corresponding differential angle $\Delta\theta$.

Table 1

$b_1 b_2$	$\Delta\theta$
0 0	0
0 1	$\pi/2$
1 1	π
1 0	$3\pi/2$

Next, referring to Fig. 1, the i -th QAM symbol $S(i)$ can be obtained from the $(i-1)$ -th symbol $S(i-1)$ and the two differential angles $\Delta\theta_1(i)$ and $\Delta\theta_2(i)$ as follows. Let the quadrant center for symbol $S(i)$ be denoted as $C(i)$, and rewrite $S(i)$ as the sum of the quadrant center $C(i)$ and a displacement vector $\Delta S(i)$:

$$S(i) = C(i) + D(i) \quad (1)$$

Then the differential 16-QAM encoding rule can be described as two recursive updating formula of the quadrant center and the displacement vector:

$$C(i) = C(i-1)e^{j\Delta\theta_1(i)} \quad (2a)$$

$$D(i) = D(i-1)e^{j\Delta\theta_2(i)} \quad (2b)$$

The initial symbol $S(0)$ can be arbitrarily set as

$$C(0) = Re^{j\pi/4}$$

$$D(0) = re^{j\pi/4}$$

where

R denotes the distance between the original point and the Quadrant center,

r is the distance between the Quadrant center and the constellation point.

The above encoding scheme can be easily explained in terms of a Solar System analogy. Let the signal space origin represents the *Sun* which is stationary. Then the *Earth* revolving around the *Sun* can be represented by the four possible quadrant centers. And naturally, the *Moon* revolving around the *Earth* is represented by four possible constellation points in each quadrant. With the above analogy, the transition from $S(i-1)$ to $S(i)$ can be simply viewed as the two relative revolving angles $\Delta\theta_1(i)$ and $\Delta\theta_2(i)$ between the *Sun*, *Earth*, and *Moon*. To be more specific, the first differential angle $\Delta\theta_1(i)$ rotates the *Earth* with respect to the *Sun*, and the second differential $\Delta\theta_2(i)$ rotates the *Moon* with respect to the *Earth*. Note that the above differential coding scheme can be easily extended to higher-level QAM such as $M=64=2^6$, in which case the symbol vector will be decomposed into three displacement vectors which are rotated by three differential angles, respectively.

In what follows, we give a simple example to illustrate the above 16-DQAM encoding scheme.

Example: Generation of 16-DQAM symbols

Let the data bits = [01 01 11 10] which correspond to two 16-DQAM symbols.

From Table 1, we obtain the following differential angles for the two symbols

$$\Delta\theta_1(1) = \frac{\pi}{2} ; \Delta\theta_2(1) = \frac{\pi}{2}$$

$$\Delta\theta_1(2) = \pi ; \Delta\theta_2(2) = \frac{3\pi}{2}$$

According to (2a) and (2b), the two 16-DQAM symbols can be readily obtained, where in Fig. 2(a) we show the transition from $S(0)$ to $S(1)$, and in Fig. 2(b) the transition from $S(1)$ to $S(2)$.

B. Differential Decoding of the DQAM Symbols with Phase Ambiguity

For 16-QAM system, the phase offset can be compensated by using a decision-directed recursive Costas loop. However,

there remains in the compensated DQAM constellation a common phase ambiguity of multiples of $\pi/2$. In such a case, the proposed differential decoding of the DQAM system can still give the correct bit sequence. This can be easily seen from equations (2a) and (2b) of the differential encoding process. For example, assuming a phase ambiguity of $\phi \in \{0, \pi/2, \pi, 3\pi/2\}$, we have the noise-corrupted received symbol as

$$X(i) = S(i)e^{j\phi} + N(i) \quad (3)$$

where $N(i)$ denotes the noise component, and the received constellation has been adjusted to correct level by using AGC. Substituting (2a) and (2b) into (3), we have

$$X(i) = C(i)e^{j\phi} + D(i)e^{j\phi} + N(i) = C_p(i) + D_p(i) + N(i)$$

where the subscript p denotes the rotation caused by the phase ambiguity ϕ . Since the rotated quadrant center can be easily decided as:

$$\hat{C}_p(i) = R[\text{sgn}(\text{real}[X(i)]) + j\text{sgn}(\text{imag}[X(i)])] \quad (4)$$

From the two complex number $\hat{C}_p(i)$ and $\hat{C}_p(i-1)$, the first differential angle $\Delta\theta_1(i)$ and the corresponding dibit can be easily detected as

$$\hat{C}_p(i) \times (\hat{C}_p(i-1))^* = \begin{cases} R^2, & \text{say } \Delta\theta_1(i) = 0 \\ jR^2, & \text{say } \Delta\theta_1(i) = \pi/2 \\ -R^2, & \text{say } \Delta\theta_1(i) = \pi \\ -jR^2, & \text{say } \Delta\theta_1(i) = 3\pi/2 \end{cases} \quad (5)$$

Once the $\Delta\theta_1(i)$ has been detected, the rotated displacement vector $D_p(i)$ can be estimated likewise:

$$\hat{D}_p(i) = r[\text{sgn}(\text{real}[X(i) - \hat{C}_p(i)]) + j\text{sgn}(\text{imag}[X(i) - \hat{C}_p(i)])] \quad (6)$$

Finally, the second differential angle $\Delta\theta_2(i)$ can be similarly detected as

$$\hat{D}_p(i) \times (\hat{D}_p(i-1))^* = \begin{cases} r^2, & \text{say } \Delta\theta_2(i) = 0 \\ jr^2, & \text{say } \Delta\theta_2(i) = \pi/2 \\ -r^2, & \text{say } \Delta\theta_2(i) = \pi \\ -jr^2, & \text{say } \Delta\theta_2(i) = 3\pi/2 \end{cases} \quad (7)$$

Therefore, regardless of the phase ambiguity ϕ , the differential angles can always be correctly detected. For higher-level QAM, similar procedure can be applied.

In Fig.3, the BER performance of the 16-DQAM system is compared to the coherent 16-QAM system under the AWGN channel, it is seen that the 16-DQAM system degrades by only about 0.5 dB.

III. THE TOP-DOWN SDR DESIGN AND IMPLEMENTATION OF THE NDA-DQAM SYSTEM

As is shown in Fig.4, the audio-band SDR platform needs only ordinary PC with soundcard support and the Matlab software with DAQ toolbox. To be more specific, it is composed of two essential parts:

1. Matlab command window #1 for running the transmitter-plus-channel (TX+CH) SDR program, which uses the soundcard SPK output jacket for sending out the modulated signal.

2. Matlab command window #2 for running the receiver (RX) SDR program which acquires its input signal from the LINE IN jacket.

Besides, an audio wire is used to connect the SPK and the LINE IN jackets. Oscilloscope can also be used to monitor the real-world audio signal during transmission.

A. System Block Diagram Design and System Parameters

Setting

Fig. 5 shows the block diagram of the audio-band 16-DQAM modem, in which Fig. 5(a) shows the transmitter, and Fig. 5(b) is the associated receiver. At the transmitter, the analog voice signal is first over-sampled and converted to source bit sequence at the rate of 44.1 kbps by adaptive delta modulation (ADM), and then mapped to the 16-DQAM symbol sequence. At the receiver, real-time data processing is utilized, where the non-data-aided (NDA) synchronizer includes (1) the T/2-spaced interpolator combined with early-late gate for symbol timing recovery, and (2) the recursive digital Costas loop synchronizer for carrier synchronization [5-6]. The detailed system parameters setting is listed below.

1. The Transmitter Parameters

- ADM voice-coding parameters

Source bit rate (sampling frequency)	$R_b = 44.1$ kbps
Step size	$\delta = 0.05$
Step-size amplification factor	$K_g = 0.1$

- 16-DQAM parameters

Symbol rate	$R_s = R_b/4 = 11.025$ kbps
Over sampling rate	OVR=4
TX D/A sampling rate	$f_{st} = \text{OVR} * R_s = 44.1$ kbps
TX carrier frequency	$f_c = 10$ kHz
SRRC filter roll-off factor	$\alpha = 0.5$

2. The NDA-DQAM Receiver Parameters

RX A/D passband sampling rate	$f_{sr} = 44.1$ kbps
RX quadrature LO frequency	$f_{cr} = f_c + f_0$ (A carrier offset f_0 can be added deliberately)
Matched filter and decimation ratio	0.5-SRRC with 2:1 decimation

- T/2-spaced Interpolation Timing Synchronizer

Step size of the 1 st -order tracking loop for fractional timing offset	$K_0 = 0.1$
--	-------------

- Recursive Costas Loop

2 nd -order phase tracking loop	$\gamma = 0.038, \rho = 0.85$
--	-------------------------------

B. The DQAM Transmitter

As is shown in Fig 5(a), the 16-DQAM symbols are divided into I/Q branches with the squared-root- raised cosine (SRRC) pulse shaping, and up-converted to the transmitted passband signal as follows

$$s_T(nT_s) = s_I(nT_s) \cos(2\pi f_c nT_s) + s_Q(nT_s) \sin(2\pi f_c nT_s)$$

Next, use the Matlab built-in function sound to send the signal to the SPK OUT jacket

sound(s_tx,44100)

Then with a 3.5mm audio wire, the output signal is then connected back to the LINE IN jacket of the soundcard for demodulation purpose.

C. The SDR Design of the NDA-DQAM Receiver

1. Signal recording and real-time receiver processing

Since the sound card is full-duplex, while the transmit signal is sending, it can record LINE IN signal at 44.1 kHz through the soundcard's A/D converter. Herein we use the Matlab DAQ toolbox [4] to record the signal, and FIFO data buffer is allocated such as to facilitate the real-time segment-by-segment processing of the whole data record. The term "real-time" is used to stand for the receiver's capability of demodulating a short segment (say, 0.1 sec) of the signal right after it is received.

Fig. 6 illustrates the procedure of real-time data processing. As one data segment is recorded in the FIFO buffer, the data processing is launched immediately, which includes I/Q down-conversion, matched filtering, synchronization, differential decoding for symbol decision, ADM source decoding, and speech signal reconstruction.

2. I/Q down-converter

As is shown in Fig. 5(b), in the I/Q demodulation, we deliberately set the local oscillator frequency as $f_{cr} = f_c + f_0$, which introduces a carrier frequency offset f_0 for testing the subsequent Costas loop. After passing through a halfband filter and down-sampling by two, the baseband T/2-spaced I/Q samples can be finally obtained from a 0.5-SRRC matched filtering.

3. T/2-spaced interpolator and early-late synchronizer

As is plotted in Fig. 7, the symbol timing control is accomplished by using a variable interpolation FIR filter with the following impulse response $h(n; \mu)$

$$h(n; \mu) = \text{sinc} \left[\frac{\pi}{T_s} (nT_s + \mu T_s) \right] ; n = -L, \dots, -1, 0, 1, \dots, L \quad (8)$$

where $\mu \in [0, 1]$ is a fractional timing offset to be adjusted by the timing loop, and $T_s = T/2$. Then the k -th interpolated sample at $kT_s + \mu T_s$ is just a weighted combination of its nearby samples, i.e.

$$y(k) = x(kT_s + \mu T_s) = \sum_{n=-L}^L x[(k-n)T_s] h(n; \mu) \quad (9)$$

And according to the non-data-aided early-late-gate (NDA-ELG) synchronizer, the timing error signal $e_r(k)$ is formed as [5]

$$e_r(k) = \text{Re} \{ y(k) [y(k+1) - y(k-1)] \} \quad (10)$$

With the timing error signal, the fractional timing offset μ can be recursively updated as follows

$$\mu(k+1) = \mu(k) + K_0 e_r(k) \quad (11)$$

where K_0 is a step-size constant.

4. Recursive digital Costas-loop synchronizer

The Costas loop is used to lock the received signal

constellation to its grid. But it remains an unsolved phase ambiguity term. Based on the LMS-like adaptive algorithm

$$\begin{pmatrix} \text{Updated} \\ \text{estimate} \end{pmatrix} = \begin{pmatrix} \text{Old} \\ \text{estimate} \end{pmatrix} + \begin{pmatrix} \text{Step-size} \\ \text{parameter} \end{pmatrix} \begin{pmatrix} \text{Error} \\ \text{signal} \end{pmatrix}$$

and with the phase error signal as

$$e(k) = \text{Im}[\hat{a}_k^* \tilde{x}_k e^{-j\hat{\theta}(k)}] \quad (12)$$

a 2nd-order Costas loop is used to produce a carrier phase track as follows [6]

$$\theta_1(k+1) = \theta_1(k) + \gamma e(k) \quad (13a)$$

$$\hat{\theta}(k+1) = \hat{\theta}(k) + \rho \theta_1(k) \quad (13b)$$

where $\hat{\theta}(k)$ is the former carrier phase estimate, $\hat{\theta}(k+1)$ is the new carrier phase estimate, and γ and ρ are the loop parameters. The above algorithm can be implemented as shown in Fig. 8. After obtaining a new carrier phase estimation, the received and interpolated complex baseband signal is de-rotated by the phase correcting term $\exp(-j\hat{\theta}(k))$.

IV. SDR EXPERIMENTAL RESULTS

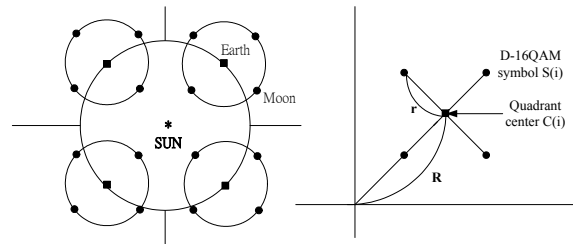
In the SDR experiment, a test voice signal is transmitted by the proposed NDA-DQAM system, and the relevant system parameters have been outlined in previous section. Here we set the frequency offset to $f_o=10\text{Hz}$. The received signal is processed in real time by dividing the signal into consecutive 0.1-sec segments. Fig. 9(a)-(b) show the tracking curves of the NDA-ELG symbol synchronizer and the Costas carrier synchronizer, respectively. It is seen that both the NDA synchronizers can lock the signal quickly. Fig. 10(a) shows the signal constellation after the NDA-ELG synchronizer, where the frequency offset gives rise to rotation of the signal constellation. Finally, Fig. 10(b) shows the constellation after the Costas loop for carrier phase correction. It is seen that the constellation is now aligned with the signal space axes. Although there still exists a unknown phase ambiguity, the previously proposed DQAM decoding scheme can solve the problem and finally detect all source bits correctly.

V. CONCLUSIONS

In this paper, we have proposed a differential-QAM system to solve the phase ambiguity problem for real-time speech transmission. It is shown that the differential coding/decoding scheme is very systematic and costs only a little extra computational load. As for the BER performance, the 16-DQAM system just loses 0.5dB in E_b/N_0 , as compared to the coherent 16-QAM system. Next, we presented a detailed design and implementation of an instructive SDR platform based on the PC soundcard and Matlab software. This audio-band 16-DQAM SDR modem can be very flexibly to adjust its parameters and even its signal processing algorithms. Hence, such a PC-based SDR approach can be very valuable in the system development process as a first start-up prototyping and verification workbench.

VI. REFERENCES

- [1] L. Hanzo et al : Single- and Multi-carrier Quadrature Amplitude Modulation, John Wiley, 2000.
- [2] J. K. Hwang (2003) : "Innovative Communication Design Lab based on PC Sound Card and Matlab : A Software-Defined-Radio OFDM Modem Example, "IEEE ICASSP 2003, Hong-Kong, April, 2003.
- [3] W. Tuttlebee Ed. : *Software Defined Radio*, John Wiley, 2002.
- [4] The Matlab Data Acquisition Toolbox Manual, MathWorks Inc., 2001.
- [5] H. Myer, M. Moeneclaey and S. A. Fechtel : *Digital Communication Receivers*, Wiley, 1998.
- [6] U. Mengali and Aldo N. D'Andrea: *Synchronization Techniques for Digital Receivers*, Plenum,1997.



- * : SUN= origin.
- : Earth= quadrant centers C(i).
- : Moon= 16-DQAM symbol S(i).

Fig.1 The composition of 16-DQAM constellation and its Solar System analogy

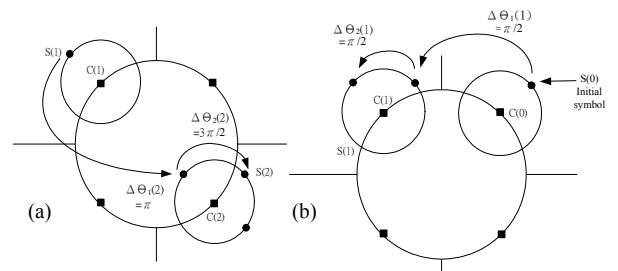


Fig.2 (a) The symbol transition from $S(0)$ to $S(1)$,
(b) The symbol transition from $S(1)$ to $S(2)$

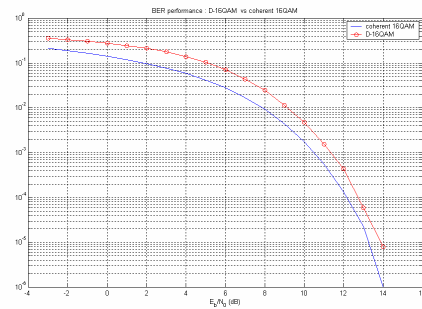


Fig.3 The BER performance comparison of 16-DQAM vs. coherent 16-QAM.

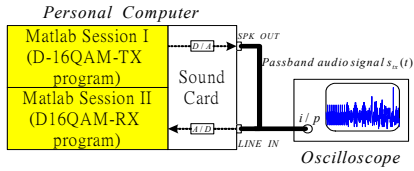


Fig.4 The setup of the SDR audio-band 16-DQAM modem lab based on PC soundcard and Matlab

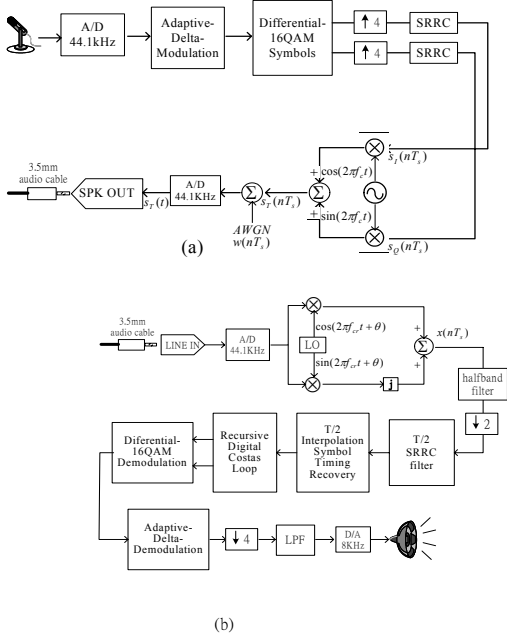


Fig.5 The Block diagrams of : (a)16-DQAM SDR TX, (b) NDA-DQAM SDR RX

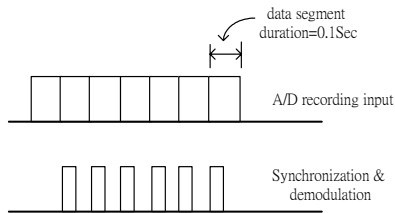


Fig.6 The illustration of real-time receiver processing.

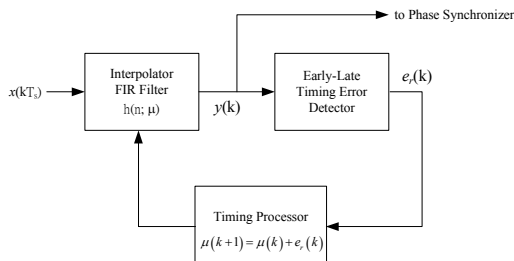


Fig.7 The Block diagrams of T/2-spaced sinc- interpolator and NDA-ELG symbol synchronizer.

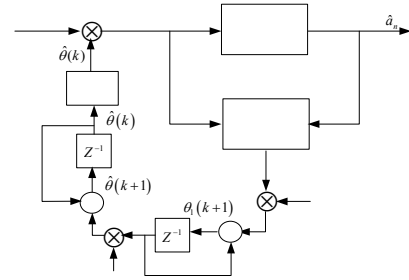


Fig.8 The Block diagrams of the recursive digital Costas-loop

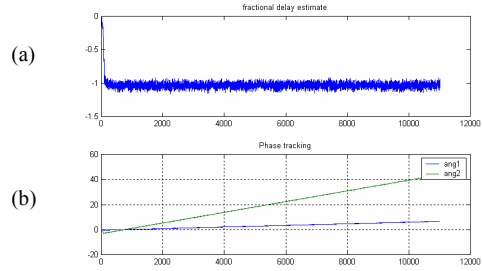


Fig.9 The tracking curves of : (a) the symbol synchronizer, and (b) the carrier synchronizers.

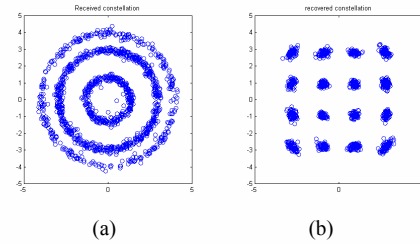


Fig.10 The 16-DQAM symbol constellation after (a) the symbol synchronizer, and (b) the carrier synchronizer.