Pacific Science Review A: Natural Science and Engineering 17 (2015) 79–82

HOSTED BY

ELSEVIER

Contents lists available at ScienceDirect

Pacific Science Review A: Natural Science and Engineering

journal homepage: www.journals.elsevier.com/pacific-science-review-a-natural-science-and-engineering/

Pseudohexadecimal near maximum likelihood detector

Muhanned AL-Rawi [a,*], Muaayed AL-Rawi [b]

[a] Department of Electrical Engineering, University of Ibb, Yemen
[b] Department of Electrical Engineering, AL-Mustansiryia University, Iraq

ARTICLE INFO

Article history:
Available online 22 March 2016

Keywords:
Intersymbol interference
Near-maximum likelihood detector
QAM modem

ABSTRACT

This paper introduces two newly developed near-maximum likelihood detectors, named pseudohexadecimal and modified pseudohexadecimal near-maximum likelihood detectors. These two detectors are tested against a pseudoquaternary near-maximum likelihood detector using data transmission at 9.6 kb/s over a telephone channel. Simulation results demonstrate that the performance of the pseudohexadecimal detector is better than the performance of the modified pseudohexadecimal detector, and the latter is better than the pseudoquaternary detector.

1. Introduction

In digital data transmission systems, the communication channel introduces different types of impairments, one of which is intersymbol interference (ISI). Adaptive linear or nonlinear (decision-feedback) equalizers are used to handle ISI at the receiver end [1]. It is well known that a maximum likelihood sequence estimation (MLSE), implemented with the Viterbi algorithm, can provide a significant improvement in detection performance compared with equalization techniques [2,3].

When the sampled impulse response of the channel contains a large number of components, the Viterbi algorithm requires both an excessive amount of storage and an excessive number of operations per received data symbol. Considerable research has been carried out to realize the performance of the MLSE at reduced complexity [4–20].

A promising technique for overcoming this problem is to modify the Viterbi algorithm itself but without greatly reducing the tolerance of the detector to noise. Such systems are called near-maximum likelihood detectors [21–27]. These detectors operate similarly to the Viterbi algorithm but use a different selection process for the stored sequences of possible data symbol values, and very few of these sequences are stored with the corresponding costs.

2. Model of data transmission

The data transmission system considered here is shown in Fig. 1. The input to the transmitter is data symbols $\{s_i\}$, which are statistically independent and equally likely to have any of (m) given values. In the application over a telephone channel to be considered here, $m = 16$, and possible values of s_i are given by all combinations of $\pm a$, $\pm b$, and $\pm ja$, $\pm jb$ where $a = 1$, $b = 3$, and $j = \sqrt{-1}$. The resulting output of the transmitter is a quadrature amplitude modulation (QAM) signal with a carrier frequency of 1800 Hz and symbol rate of 2400 baud, yielding an information rate of 9.6 kb/s to be sent over the telephone channel, which adds ISI and additive white Gaussian noise (AWGN) to the QAM signal. The receiver demodulates the QAM signal, and its output is fed to the estimator to estimate the impulse response of the baseband telephone channel; finally, the detector model combats ISI and detects the original data.

3. Detector model

3.1. Pseudoquaternary and pseudohexadecimal detectors

The pseudoquaternary near-maximum likelihood detector (QD) found in Ref. [23] and the new detector called the pseudohexadecimal near-maximum likelihood detector (HD) are described as follows.

* Corresponding author.
E-mail addresses: muhrawi@yahoo.com (M. AL-Rawi), muaayed@yahoo.com (M. AL-Rawi).

Peer review under responsibility of Far Eastern Federal University, Kangnam University, Dalian University of Technology, Kokushikan University.

http://dx.doi.org/10.1016/j.psra.2015.12.003

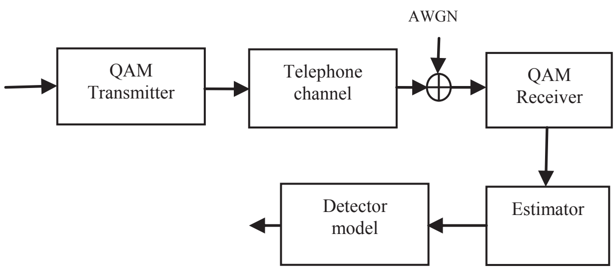

Fig. 1. Model of data transmission.

Just prior to the receipt of the signal sample r_i at time $t = iT$, the detector holds in store k (n-component) vectors Q_{i-1} as given below ($k = 4$ for pseudoquaternary, and $k = 16$ for pseudohexadecimal):

$$Q_{i-1} = [x_{i-n}x_{i-n+1}\ldots x_{i-1}] \tag{1}$$

where x_i is a possible value of s_i

Each stored vector is associated with a cost U_{i-1} given by

$$U_{i-1} = \sum_{j=0}^{i-1}\left|r_j - \sum_{h=0}^{g} x_{j-h}v_h\right|^2 = \sum_{j=0}^{i-2}\left|r_j - \sum_{h=0}^{g} x_{j-h}v_h\right|^2 + w_{i-1} = U_{i-2} + w_{i-1} \tag{2}$$

where $\{v_h\}$ is the sampled impulse response of the baseband telephone channel with length of $(g+1)$, and w_{i-1} is the corresponding estimate of the noise component in the received sample r_{i-1}.

Upon the receipt of the signal sample r_i, the detector expands every vector Q_{i-1} into four $(n+1)$ components vectors $\{P_i\}$, as given below, with minimal cost. The selection of P_i is achieved through the use of simple threshold-level comparison and does not involve the computation of any costs.

$$P_i = [x_{i-n}x_{i-n+1}\ldots x_i] \tag{3}$$

The first n components of P_i are as shown in the original vector Q_{i-1}, and the last component x_i takes on any one of the four different values of its 16 possible values. In pseudoquaternary, the number of expanded vectors is 16 (see Fig. 2), whereas in pseudohexadecimal, it is 64 (see Fig. 3). The detector then evaluates for each expanded vector P_i its cost given by

$$U_i = U_{i-1} + \left|r_i - \sum_{h=0}^{g} x_{i-h}v_h\right|^2 = U_{i-2} + w_{i-1} + w_i \tag{4}$$

The detector then selects the vector P_i with the smallest cost and takes its first component x_{i-n} as the detected value s'_{i-n} of the data symbol s_{i-n}.

All vectors $\{P_i\}$ for which $s'_{i-n} \neq x_{i-n}$ are discarded and the first components of all remaining vectors are omitted to give the corresponding n-component vectors $\{Q_i\}$, where

$$Q_i = [x_{i-n+1}x_{i-n+2}\ldots x_i] \tag{5}$$

The detector then selects from the resulting vectors $\{Q_i\}$ the k vectors with the lowest costs $\{U_i\}$. The k vectors $\{Q_i\}$ together with their costs are stored in preparation for the next detection cycle.

After each detection process and to prevent overflow due to the increase in costs over any transmission, the smallest cost is subtracted from the cost of each vector so that the value of the smallest cost is always reduced to zero.

The starting up procedure for the detector begins with k stored vectors $\{Q_{i-1}\}$ that are all the same and correct. A zero cost is allocated to one of the k vectors, and a very high cost is assigned to each of the remaining vectors. After a few received samples, the detector holds k vectors, which are all different and are all derived from the original vector with zero cost.

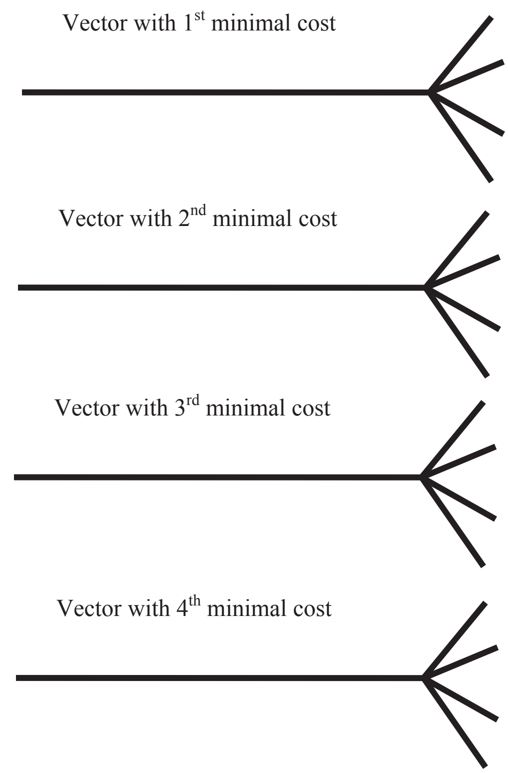

Fig. 2. Configuration of pseudoquaternary detector.

3.2. Modified pseudohexadecimal detector

The second new detector is named the modified pseudohexadecimal detector (MHD). In MHD, the detector holds in the stored 16 (n-component) vectors Q_{i-1} given by Equation (1). Each stored vector is associated with a cost U_{i-1} given by Equation (2). Upon the receipt of the sample r_i, the detector expands the 1st, 2nd, 3rd, and 4th vectors into four (n+1) component vectors $\{P_i\}$ with minimal cost and expands the 5th, 6th, 7th, and 8th vectors into three (n + 1) component vectors with minimal cost. The 9th, 10th, 11th, and 12th vectors are expanded to two (n + 1) component vectors with minimal cost, whereas the 13th, 14th, 15th, and 16th vectors are expanded to one (n + 1) component vector with minimal cost. Thus, the number of expanded vectors is 40 (see Fig. 4) instead of 64; therefore, the complexity of MHD is lower than the complexity of HD. The complexity is proportional to the number of stored and expanded vectors.

The detector then evaluates for each expanded vector its cost given by Equation (4). The detector selects the vector P_i with the smallest cost and takes its first component x_{i-n} as the detected value s'_{i-n} of the data symbol s_{i-n}. All vectors $\{P_i\}$ for which $s'_{i-n} \neq x_{i-n}$ are discarded, and the first components of all remaining vectors are omitted to give the corresponding n-component vectors $\{Q_i\}$.

The detector then selects from the resulting vectors $\{Q_i\}$ the 16 vectors with the lowest costs $\{U_i\}$. The 16 vectors $\{Q_i\}$ together with their costs are stored in preparation for the next detection cycle.