A Novel SNR Estimator for DS-UWB Wireless Sensor Network

Bo Li¹, Qing-zhi LIU², Zhen-dong YIN²,* and Zhi-lu WU²

¹School of Electronics and Information Engineering, Harbin Institute of Technology, China Eastern Route Corporation of South-to-North Water Diversion, Harbin, China
²School of Electronics and Information Engineering, Harbin Institute of Technology, Harbin, China

*Corresponding author

Keywords: DS-UWB, Wireless sensor network, SNR estimate, Regression analysis, LSTM network.

Abstract. In this technical survey, we introduce the state-of-the-art signal-to-noise rate (SNR) estimation techniques for direct sequence ultra-wide bandwidth (DS-UWB) system. The SNR estimation is important to improve the system performance in a multiuser wireless sensor network (WSN). We demonstrate the effectiveness of using regression analysis of features extracted from code mapping to derive a new SNR estimator. And compared the result with another estimator derived by long short-time memory (LSTM) recurrent neural networks (RNN) in terms of their root-mean-squared error. Numerical result show that with a proper length of sequence the proposed method can reach an excellent performance when operating in a DS-UWB system.

Introduction

Ultra-wide bandwidth (UWB) technology has attracted considerable interest in the research and standardization communities, due to its promising ability to provide high data rate at low cost with relatively low power consumption[1]-[3]. However the FCC’s new regular brings UWB technology not only opportunity but challenge too. When come to use, the UWB system must aware the quality of the communication channel before it can decide the transmission power. The knowledge of signal-to-noise rate (SNR) can help in deciding the proper emission power.

In [4], the problem of SNR estimation for UWB when independent sequences of noisy sign and pure noise is considered from the perspective of Birnbaum’s sequential sampling method. In paper [5], the author proposed an algorithm for estimating the pulse energy-to-noise ratio of an UWB sign through utilization of output samples from a correlated in the receiver and evaluate the performance of the proposed algorithm through computer simulation. In [6], PSO algorithm not rely on a specific single variable initialization and is less complex. However, most of these estimators are for a conventional narrowband system and doesn’t take the properties of the UWB signals into account.

In this paper, we proposed a SNR estimation method based on the regression analysis of features extract from code mapping. The remainder of this paper is organized as follows. Section II describes the framework of the proposed method, which includes transmitted signal, system model and the SNR estimator designed. Simulation results are provided in Section III. Finally, in Section IV conclusions of the paper are presented.

Framework of the Proposed Method

Signal Model

In this paper, the transmit signal of wireless sensor network is based on DS-UWB system where each transmitter employs binary phase-shift key (BPSK) modulation. The kth user’s transmit signal can be written as:
Where the BPSK symbols \( b_l(i) \in \{-1,1\}_l^{M} \) are spared with the specific pseudorandom (PN) codes \( c_i(t) \in \{+1,-1\} \). The BPSK symbols are modulated by the pulse signal \( p(t) \) with a duration of nanosecond. \( M \) represents the length of transmitted bits. Assume the symbol duration is denoted by \( T_s \) and each BPSK symbol can be divided into \( N_c \) chips each with duration of \( T_c \). What we should aware is that \( N_c = T_s / T_c \) and in each chip the \( p(t) \) lasts \( T_p (T_c \gg T_p) \).

At the receiver, there is a K-user DS-UWB MA system and the transmission channel is the additive Gaussian noise (AWGN) channel with multiple access interference (MAI). The model of the received signal in such a channel is expressed as follows:

\[
r(t) = \sum_{k=1}^{K} x_k(t) + n(t)
\]  

(2)

Where \( x(t) \) is the received signal of different user and \( n(t) \) is the additive white Gaussian noise. The MAI is caused by several user signal overlapped on both time and frequency domain. For kth user signal, other \( K - 1 \) user’s signal are MAI. If we want to get the kth user’s signal, using the correlation receivers (sometimes called matched filter receivers) can detect different user’s signal and get the desired user’s signal.

**System Model**

Figure 1. shows the block diagram of the proposed system model. Because there are \( K \) users, the number of transmitter is \( K \). Each of the transmitter corresponding to one of the users. The received signal \( r(t) \) will go through the match filter to remove the Gaussian white noise and MAI. For all \( K \) users the result of the matched filter can be expressed as a vector \( y \) with \( K \) elements. The kth user’s output of the filter can be written as:

\[
y_k(i) = \int_{-iT_f}^{(i+1)T_f} x_k(t)r(t)dt
\]  

(3)

It is the output corresponding to the \( x_k(t) \) in the ith information symbol interval.
If we ignore the interval and make the formula clear and general. Which means that all the information symbol intervals are the same to the filter receiver. By this way the element \( y_k \) can replace all the time sequence \( y_k(t) \). And the equation of the output of the filter can be expressed as equation (4).

In the equation (4) the first part \( A_k b_k \) is the ideal result of the \( k \)th matched filter. The second part of the equation (4) is the MAI to the \( k \)th user and \( r_{ik} \) is the cross-correlation coefficient between the user \( i \) and user \( k \). The last part \( n_k \) is the AWGN in the system channel that interference the \( k \)th user. The vector representation of matched filter can be expressed as:

\[
y = RAb + n
\]

In equation (5) vector \( y \) is the made up by output of all matched filters \( y = [y_1, y_2, \ldots, y_K]^T \). The vector \( R \) denotes the cross-correlation matrix composed of \( r_{ik} (i, k = 1, 2, \ldots, K) \), and \( A \) is the matrix \( A = \text{diag}\{A_1, A_2, \ldots, A_K\} \), vector \( b = [b_1, b_2, \ldots, b_K]^T \), vector \( n = [n_1, n_2, \ldots, n_K]^T \).

Then judge the output signal of the matched filter through sign detector and map the received bits. The sign detector result can be written as vector

\[
b = \text{sgn}(y)
\]

In paper [7], the results of code mapping when the MAI is the main interference source are shown as follows:

\[
L(b_i) = \sum_{j=1}^{K} A_j A_j r_{ij} b_j - A_i y_i \quad i = 1, 2, \ldots, K
\]

(7)

Considering that \( b_i (i \in [1, K], \ i \in K) \) be the wrong code and other codes are correct, we substituted the equation (7) as follows:

\[
L(b_i) = \sum_{j=1}^{K} A_j A_j r_{ij} b_j - A_i y_i + A_i^2 r_{i} b_i = \sum_{j=1}^{K} A_j A_j r_{ij} b_j - A_i y_i + A_i^2 r_{i} (-b_i) + 2A_i^2 r_{i} b_i
\]

\[= 2A_i^2 r_{i} b_i - A_i n_i
\]

What’s more, if \( k \) is not equal to \( i \), we will get:

\[
L(b_k) = 2A_k A_k r_{ik} b_i - A_i n_i \quad k = 1, 2, \ldots, K, k \neq i
\]

(9)

The code mapping will distinguish the MAI and AWGN of the received bits. The first part of the equation (9) is the MAI of the \( k \)th user. While the second part of the equation (9) is the AWGN interference. The mapping result will send to SNR estimator for further study.

**SNR Estimator Using Polynomial Regression**

In order to observe the differences directly, we calculate the mean value and standard deviation of the mapping value of received bits. The mean value and standard deviation decline as the SNR
increases. In other words, the mapping value contains obvious features about SNR in an AWGN channel. Both the mean value and standard deviation have a negative correlation with SNR.

The relation between mean value, standard deviation and SNR is nonlinear. In this case, polynomial regression can be adapted as a helping method [8]-[9]. In general, we can model the expected value of $y$ as an $m$th degree polynomial, yielding the general polynomial regression model as follows:

$$
\hat{y} = b_0 + b_1 x + b_2 x^2 + \ldots + b_m x^m 
$$

(10)

In our case study, the polynomial of 6rd is the best to choose and the centerpiece for linear regression is the functions:

$$
SNR = b_0 + b_1 (\text{meanvalue}) + b_2 (\text{meanvalue}^2) + \ldots + b_6 (\text{meanvalue}^6)
$$

(11)

$$
SNR = \beta_0 + \beta_1 (\text{standard}) + \beta_2 (\text{standard}^2) + \ldots + \beta_6 (\text{standard}^6)
$$

(12)

The equation (11) and equation (12) are fit formula for mean value and standard deviation respectively. The Figure. 2. Shows the mean value and standard deviation of the mapping value at different SNR. The lower one is mean value while the other one is the standard deviation. From the graphic blow, it is obvious that the SNR has a negative correlation with mean value and standard deviation. We can use the both features to estimate the SNR in the system. The simplest way to do so is take the average of two features estimation result.

![Figure 2. Mean value and standard deviation of the mapping value at different SNR.](image)

**Simulation Result and Discussion**

**Comparison Method and Simulation Parameters**

In this section, we propose a SNR estimator based on regression analysis of features extract after code mapping for BPSK signals in DS-UWB systems. Another SNR estimation method using long short-time message (LSTM) network is chosen as the contrast method. LSTM RNNs are well-suited to process and learn features from time series [10]-[11]. The structure of LSTM RNNs we build for estimating SNR is listed in Table 2. We build LSTM structures with the length of sequences of 100 time steps with 10 layers. In this paper, the LSTM RNN is implemented by Keras with TensorFlow as its backend. Furthermore, the simulation is implemented on a desktop computer with Intel Core i7-6700K CPU 4.00GHz, 16GB RAM and NVIDIA GeForce GTX 1070 GPU. The simulation parameters of the communication system are shown in Table 3.
Table 2. Structure of LSTM with length of sequence 100.

<table>
<thead>
<tr>
<th>Layer number</th>
<th>Layer type</th>
<th>Parameters (input dim, output dim)</th>
<th>Layer number</th>
<th>Layer type</th>
<th>Parameters (input dim, output dim)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>LSTM</td>
<td>(1, 200)</td>
<td>6</td>
<td>Dropout</td>
<td>0.2</td>
</tr>
<tr>
<td>2</td>
<td>Dropout</td>
<td>0.2</td>
<td>7</td>
<td>LSTM</td>
<td>(500, 300)</td>
</tr>
<tr>
<td>3</td>
<td>LSTM</td>
<td>(200, 500)</td>
<td>8</td>
<td>Dropout</td>
<td>0.2</td>
</tr>
<tr>
<td>4</td>
<td>Dropout</td>
<td>0.2</td>
<td>9</td>
<td>Dense</td>
<td>(300, 1)</td>
</tr>
<tr>
<td>5</td>
<td>LSTM</td>
<td>(500, 500)</td>
<td>10</td>
<td>Activation</td>
<td>tanh</td>
</tr>
</tbody>
</table>

Table 3. System parameters.

<table>
<thead>
<tr>
<th>System</th>
<th>DS-UWB</th>
<th>System</th>
<th>DS-UWB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modulation mode</td>
<td>BPSK</td>
<td>Communication channel</td>
<td>AWGN</td>
</tr>
<tr>
<td>DS codes</td>
<td>Kasami</td>
<td>Number of users</td>
<td>10</td>
</tr>
<tr>
<td>Length of DS codes</td>
<td>255</td>
<td>Range of SNR</td>
<td>0-15dB</td>
</tr>
</tbody>
</table>

Numerical Result and Discussion

Figure 3. Regression result for different sequence length. Figure 4. Result for the LSTM network and regression analysis.

Figure 3. shows the regression result for different sequence length in an AWGN channel with MAI. It is obvious that as the sequence length increases the estimation root mean square error (RMSE) decreases. The sequence length of 100, 500, 1000, 5000bits are tested and the 5000bits curve has the most accuracy result. The result can be explained by equation (11). As there’s more bits for regression analysis the MAI is not increasing as much as the AWGN at the receiver. That is to say, after code mapping with the increasing of sequence length the MAI weight less in total received energy. What’s more, the smoothness of curve is also varies for different curves. The 5000bits curve is the smoothest one which means that the estimation result is not sensitive to SNR variety. This guarantee the SNR estimator effective under different SNR channel situation.

Figure 4. shows the RMSE of SNR estimators based on LSTM network trained by original signals and mapping data in an AWGN channel with MAI. Obviously, the RMSE of LSTM with mapping data is much lower than that of original signals. And the SNR estimator based on mapping data have RMSEs of less than 0.3 at most SNRs. As we mentioned above, the mapping data which is used for training the LSTM RNN contributes to extraction of features about SNR. Consequently, the SNR estimator based on mapping data outperforms the SNR estimator based on original signal. What’s more, the proposed method do better than SNR estimators trained by LSTM RNN when the sequence length is above 1000. The polynomial regression SNR estimator for 5000bit sequence length have RMSEs of less than 0.2 at most SNRs. And the estimation result for mapping data trained LSTM SNR estimator is range from RMSE of 0.35 at 4dB to RMSE of 0.16 at 15dB. The estimation result for polynomial regression SNR estimator is ranged from RMSEs of 0.2 at 6dB to RMSE of 0.08 at 15dB.
Summary
In this paper, we have presented a new SNR estimation method using code mapping and polynomial regression. After investigating the mean value and standard deviation of the mapping result, there is a nonlinear relationship between SNR and these features. First, we use polynomial regression to figure out the relationship between mean value, standard deviation and SNR. The polynomial regression based SNR estimator’s accuracy gets higher with the increasing of bit sequence length. Then, the simulation result shows that compared with LSTM trained estimator the polynomial regression can achieve a better accuracy and the curve is smoother. What’s more, the training procedure for LSTM network cost much more time and computation resources. It is also interesting to note that proposed SNR estimator has a larger available range for polynomial regression is not sensitive to SNR variety. Finally, the comparison shows that the accuracy is very high which means that regression analysis of features extract after code mapping can do a good job in SNR estimation.

Acknowledgement
The work in this paper is supported by National Natural Science Foundation of China (No. 61571167, No. 61471142 and No. 61102084).

Reference