Channel estimation and equalization

# Channel estimation:

The pilot symbols in LTE are assigned positions within a subframe depending on the eNodeB cell identification number and which transmit antenna is being used, as shown in the following figure.

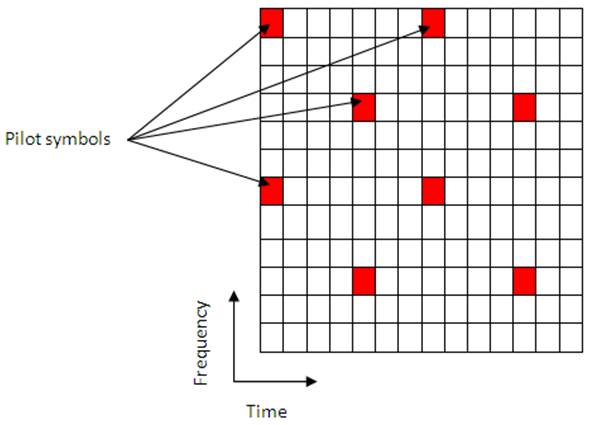


Figure 1: pilot positions

The unique positioning of the pilots ensures that they do not interfere with one another and can be used to provide a reliable estimate of the complex gains imparted onto each resource element within the transmitted grid by the propagation channel.

Both transmit and receive chains and the propagation channel model are shown in the following block diagram.

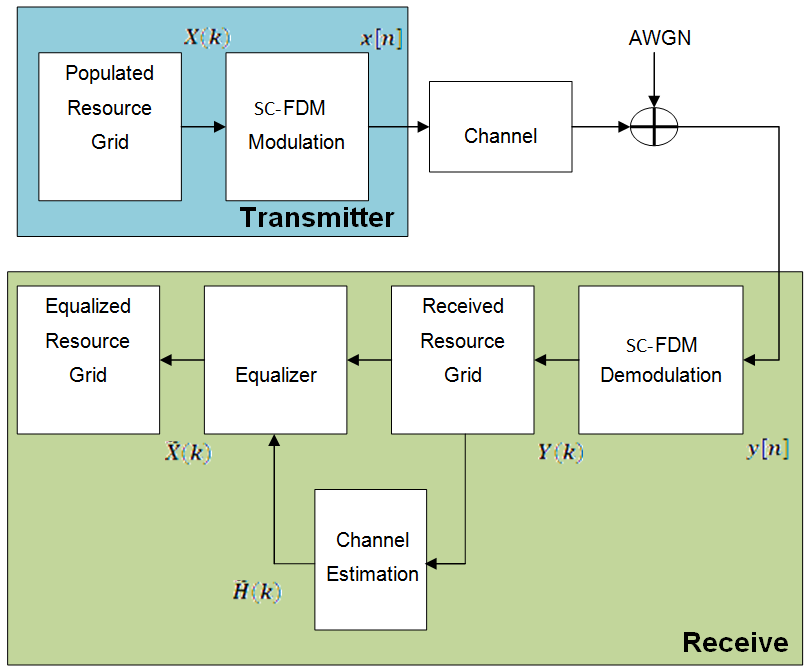


Figure 2: Both transmit and receive chains

The populated resource grid represents several subframes containing data. This grid is then SC-FDM modulated and passed through the model of the propagation channel. Channel noise in the form of additive white Gaussian noise (AWGN) is added before the signal enters the receiver. Once inside the receiver the signal is SC-OFDM demodulated and a received resource grid can be constructed. The received resource grid contains the transmitted resource elements which have been affected by the complex channel gains and the channel noise. Using the known pilot symbols to estimate the channel, it is possible to equalize the effects of the channel and reduce the noise on the received resource grid.

LTE assigns each antenna port a unique set of locations within a subframe to which to map reference signals. Because no other antenna transmits data at these locations in time and frequency, channel estimation for multi-antenna configurations can be performed. The channel estimation algorithm extracts the reference signals for a transmit/receive antenna pair from the received grid. The least squares estimates of the channel frequency response at the pilot symbols are calculated as described in On Channel Estimation in SC-OFDM Systems [1]. The least squares estimates are then averaged to reduce any unwanted noise from the pilot symbols. Because it is possible that no pilots are located near the subframe edge, virtual pilot symbols are created to aid the interpolation process near the edge of the subframe. Using the averaged pilot symbol estimates and the calculated virtual pilot symbols, interpolation is then carried out to estimate the entire subframe. This process is demonstrated in the following block diagram.

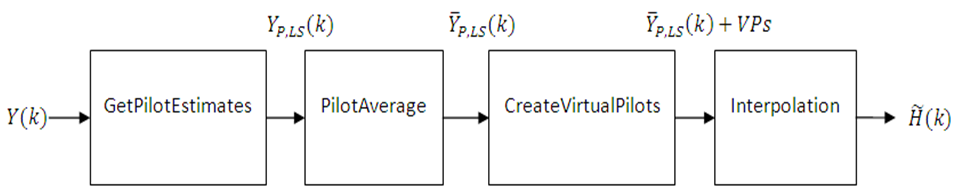


Figure 3: channel estimation block diagram

## Sub-block 1: Get Pilot Estimates Subsystem

The first step in determining the least squares estimate is to extract the pilot symbols from their known location within the received subframe. Because the value of these pilot symbols is known, the channel response at these locations can be determined using the least squares estimate. The least squares estimate is obtained by dividing the received pilot symbols by their expected value.

Y(k)=H(k) \* X(k) + noise

Where:

Y(k) is a received complex symbol value.

X(k) is a transmitted complex symbol value.

H(k) is a complex channel gain experienced by a symbol.

Known pilot symbols can be sent to estimate the channel for a subset of REs within a subframe. In particular, if pilot symbol XP(k) is sent in an RE, an instantaneous channel estimate ˜HP(k) for that RE can be computed using:

˜HP(k) = = HP(k) + noise

Where:

YP(k) represents the received pilot symbol values.

XP(k) represents the known transmitted pilot symbol values.

˜HP(k) is the true channel response for the RE occupied by the pilot symbol.

## Sub-block 2 : Pilot Average Subsystem

To minimize the effects of noise on the channel estimates, the least square estimates are averaged using an averaging window. This simple method produces a substantial reduction in the level of noise found on the pilot REs.

**Averaging Method:**

This method uses the approach described in TS 36.141 [2], Annex F.3.4. Time averaging is performed across each subcarrier that contains a pilot symbol, resulting in a column vector containing an average amplitude and phase for each subcarrier that is carrying a reference signal.

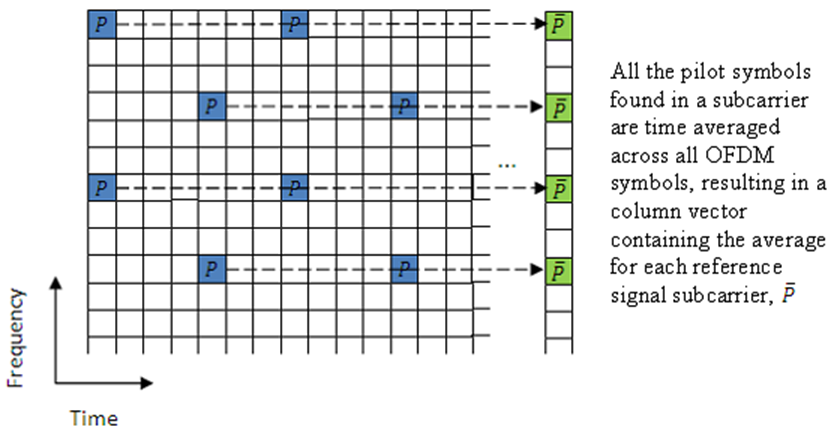


Figure 4: time averaging

The averages of the pilot symbol subcarriers are then frequency averaged using a moving window of maximum size 19.

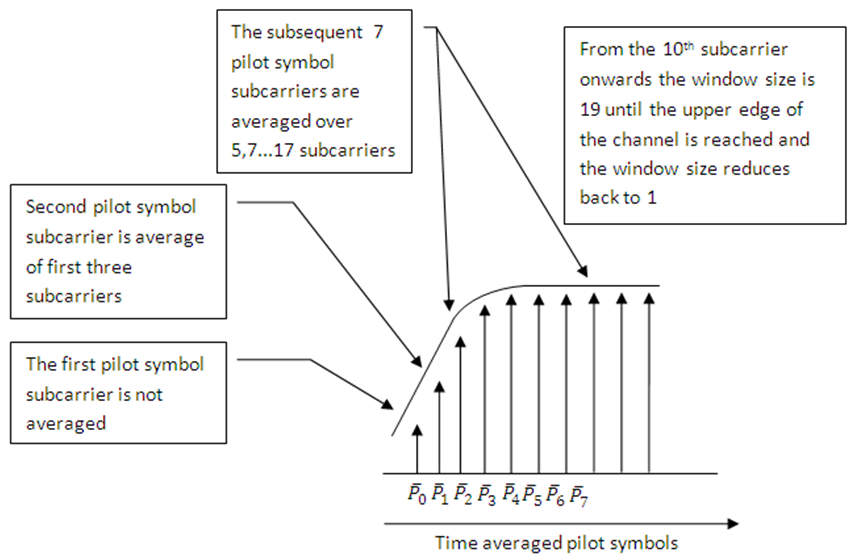


Figure 5: frequency averaging

## Sub-block 3: Create Virtual Pilots Subsystem

In many instances, edges of the resource grid do not contain any pilot symbols. This effect is shown in the resource grid in the following figure.

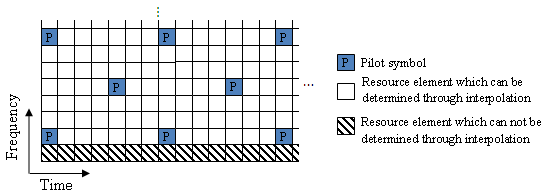


Figure 6: symbols at edge

In this case, channel estimates at the edges cannot be interpolated from the pilot symbols. To overcome this problem, virtual pilot symbols are created.

**Virtual Pilot Placement**

Virtual pilot symbols are created as shown in the following figure.

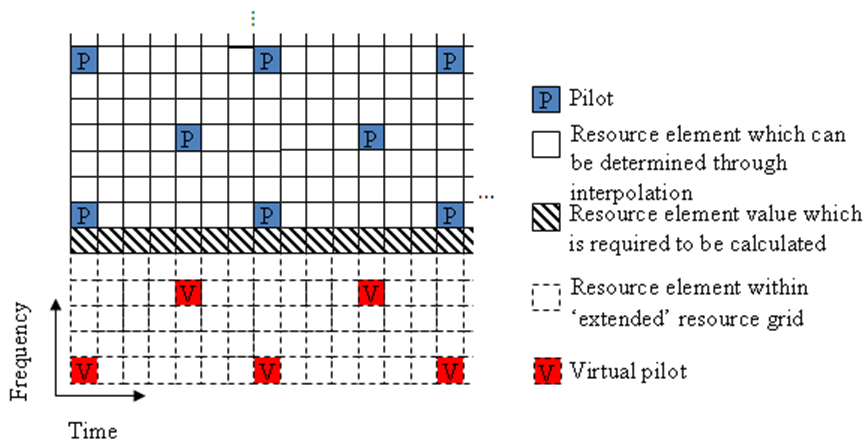


Figure 7: virtual pilot placement

In this system, the resource grid is extended, with virtual pilot symbols created in locations which follow the original reference signal pattern. The presence of virtual pilot symbols allows the channel estimate at the resource elements, which previously could not be calculated by interpolation, to be calculated by interpolation using the original and virtual pilot symbols.

Calculating Virtual Pilot Symbol Values

The virtual pilot symbols are calculated using the original pilot symbols. For each virtual pilot symbol, the value is calculated following these steps:

The closest 10 ordinary pilots in terms of Euclidian distance in time and frequency are selected. The search is optimized to consider 10 these pilots, rather than checking all possible pilots. Based on the possible configurations of the cell RS, using 10 pilots provides sufficient time and frequency diversity in the pilots for the virtual pilot calculation.

Using this set of the 10 pilots, the closest three pilot symbols are selected. These three symbols must occupy at least two unique subcarriers and two unique OFDM symbols.

Using this set of three pilots, two vectors are created. One vector between the closest and furthest pilot symbols, and one vector between the second closest and furthest pilot symbols.

The cross-product of these two vectors is calculated to create a plane on which the three points reside.

The plane is extended to the position of the virtual pilot to calculate the value based on one of the actual pilot values.

This diagram shows the virtual pilot calculation.

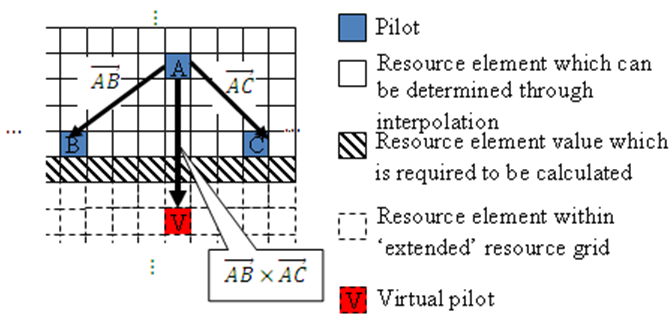


Figure 8: virtual pilot calculation

Linear interpolation is used in our implemented estimator so no virtual pilots are used

## Sub-system 4: Interpolation Subsystem

Once the noise has been reduced or removed from the least squares pilot symbol averages and sufficient virtual pilots have been determined, it is possible to use interpolation to estimate the missing values from the channel estimation grid.

The pilot averaging method described in TS 36.141 [1], Annex F.3.4, requires the use of simple linear interpolation on the time-averaged and frequency-averaged column vector. The interpolation is one-dimensional, since it only estimates the values between the averaged pilot symbol subcarriers in the column vector. The resulting vector is then replicated and used as the channel estimate for the entire resource grid.

## Noise Estimation:

The performance of some receivers can be improved through knowledge of the noise power present on the received signal. The noise power can be determined by analyzing the noisy least squares estimates and the noise averaged estimates.

The noisy least-squares estimates from the Get Pilot Estimates Subsystem and the noise averaged pilot symbol estimates from the Pilot Average Subsystem provide an indication of the channel noise. The least-squares estimates and the averaged estimates contain the same data, apart from additive noise. Simply taking the difference between the two estimates results in a noise level value for the least squares channel estimates at pilot symbol locations. Considering again.



Averaging the instantaneous channel estimates over the smoothing window, we have



where *S* is the set of pilots in the smoothing window and *|S|* is the number of pilots in *S*. Thus, an estimate of the noise at a particular pilot RE can be formed using:



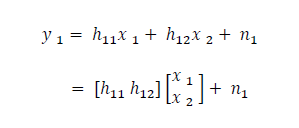
In practice, it is not possible to remove all the noise using averaging. Because it is only possible to reduce the noise, only an estimate of the noise power can be made.

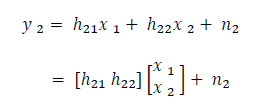
Using the value of the noise power found in the channel response at pilot symbol locations, the noise power per resource element (RE) can be calculated by taking the variance of the resulting noise vector. The noise power per RE for each transmit and receive antenna pair is calculated and stored. The mean of this matrix is returned as the estimate of the noise power per RE.

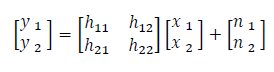
# Channel Equalization:

## Zero Forcing Equalizer:

In this method, the ISI component at the output of equalizer is forced to zero by using appropriate linear time invariant filter having suitable transfer function.

If transmitted symbol is represented by x1 and x2, h11 represent the channel from first transmitter to first receiver, h12 represent the channel from second transmitter to first receiver, h21 represent the channel from first transmitter to second receiver and h22 represent the channel from second transmitter to second receiver and n1,n2 represent noise on first and second receiver then the received symbol on first receiver is given by:

And the received symbol on second receiver is given by:

These two above equation can also be written as

It is clear from this equation that if h11, h12, h21, h22 and y1, y2 is known then it is easier for the receiver to compute the x1 and x2.

Now if we rewrite the above equation then

𝑌=𝐻𝑋+𝑛

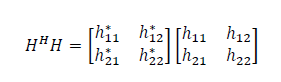
From here it is clear that in order to find x from above equation, we need to find out the matrix which is inverse of matrix H.

If W represent the inverse of H then it must satisfy the property

𝑊𝐻=𝐼

Where 𝐼 is the identity matrix

The matrix W which satisfy the above mentioned property is known as the zero forcing linear detector and is computed by following equation 𝑊=(𝐻𝐻𝐻)−1𝐻𝐻

In this equation the matrix 𝐻𝐻𝐻 is given by

From this matrix it is clear that the off diagonal term is non-zero and hence zero forcing equalizer cancel out the interference signal. It is reasonably simple and easy to implement but its main drawback is that it tends to amplify the noise and hence gives noisy output.

## MMSE Equalizer (Minimum Mean Square Error)

This type of equalizer uses the squared error as performance measurement. The receiver filter is designed to fulfill the minimum mean square error criterion. Main objective of this method is to minimize the error between target signal and output obtained by filter. The computation for this method is as follows.

1- Computes the coefficient of matrix W using MMSE algorithm which minimize the condition

𝐸{ [𝑊𝑦 − 𝑥][𝑊𝑦 − 𝑥] 𝐻 }

2- Solving the above equation gives

𝑊= (𝐻𝐻 𝐻 + 𝑁𝑜𝐼 )−1 𝐻𝐻

3- From that equation, it is clear that this equation is different from the equation of zero forcing equalizer by the term 𝑁𝑜𝐼. If we put 𝑁𝑜𝐼=0 in this equation, then MMSE equalizer becomes zero forcing equalizer.

# Propagation channel models:

## Multipath Fading Propagation Conditions

The multipath fading channel model specifies the following three delay profiles.

* Extended Pedestrian A model (EPA)
* Extended Vehicular A model (EVA)
* Extended Typical Urban model (ETU)

These three delay profiles represent a low, medium, and high delay spread environment, respectively. The multipath delay profiles for these channels are shown in the following tables.

**EPA Delay Profile**

| Excess tap delay (ns) | Relative power (dB) |
| --- | --- |
| 0 | 0.0 |
| 30 | –1.0 |
| 70 | –2.0 |
| 90 | –3.0 |
| 110 | –8.0 |
| 190 | –17.2 |
| 410 | –20.8 |

**EVA Delay Profile**

| Excess tap delay (ns) | Relative power (dB) |
| --- | --- |
| 0 | 0.0 |
| 30 | –1.5 |
| 150 | –1.4 |
| 310 | –3.6 |
| 370 | –0.6 |
| 710 | –9.1 |
| 1090 | –7.0 |
| 1730 | –12.0 |
| 2510 | –16.9 |

**ETU Delay Profile**

| **Excess tap delay (ns)** | **Relative power (dB)** |
| --- | --- |
| 0 | –1.0 |
| 50 | –1.0 |
| 120 | –1.0 |
| 200 | 0.0 |
| 230 | 0.0 |
| 500 | 0.0 |
| 1600 | –3.0 |
| 2300 | –5.0 |
| 5000 | –7.0 |

All the taps in the preceding tables have a classical *Doppler* spectrum. In addition to multipath delay profile, a maximum Doppler frequency is specified for each multipath fading propagation condition, as shown in the following table.

| **Channel model** | **Maximum Doppler frequency** |
| --- | --- |
| EPA 5Hz | 5 Hz |
| EVA 5Hz | 5 Hz |
| EVA 70Hz | 70 Hz |
| ETU 70Hz | 70 Hz |
| ETU 300Hz | 300 Hz |

# Results:

Ber of full chain without channel encoding and using channel estimation and equalization in AWGN cahnnel:

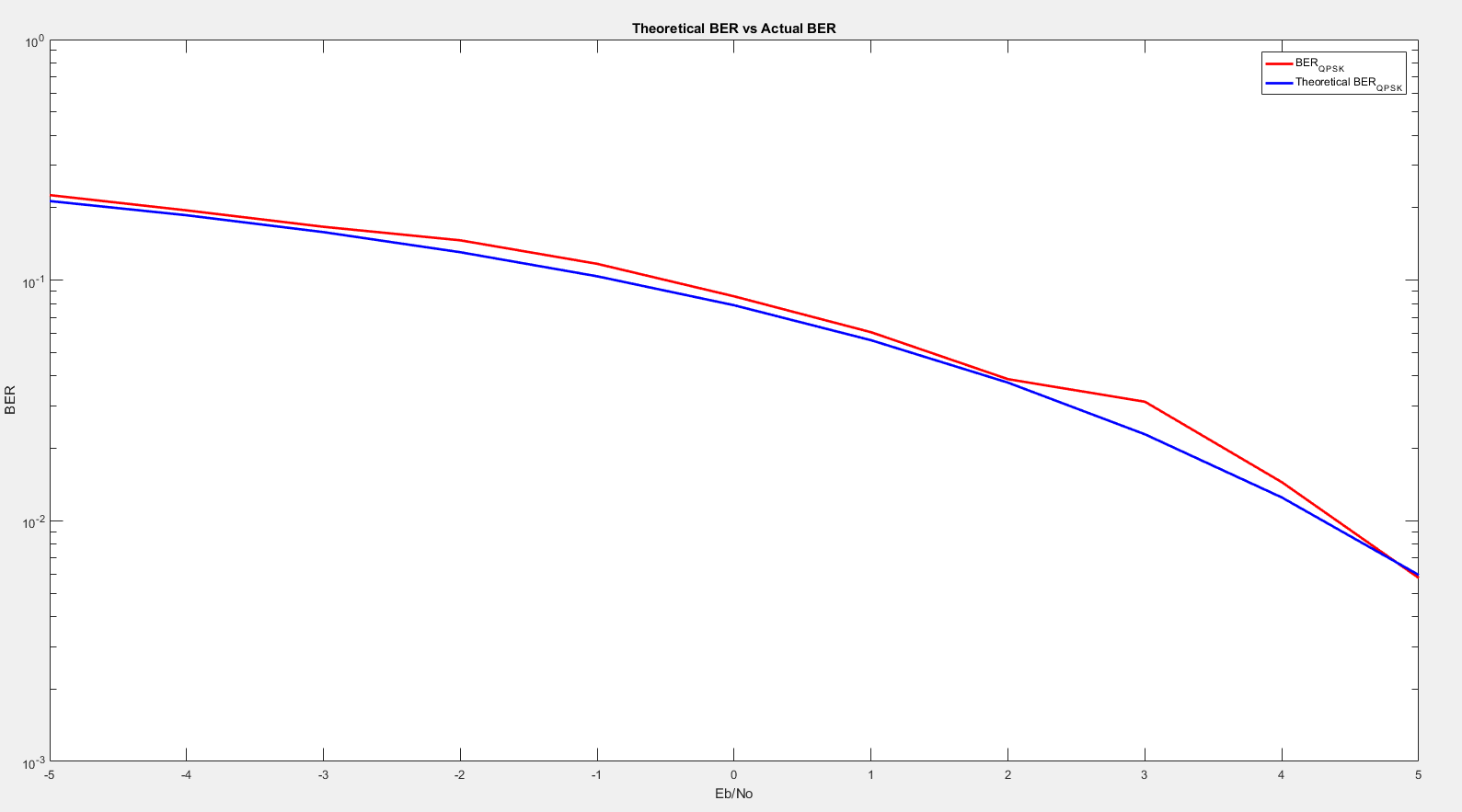


Figure 9: BER using channel equalization in AWGN channel

Ber of full chain without channel encoding and using channel estimation and equalization in Extended Pedestrian A model cahnnel:

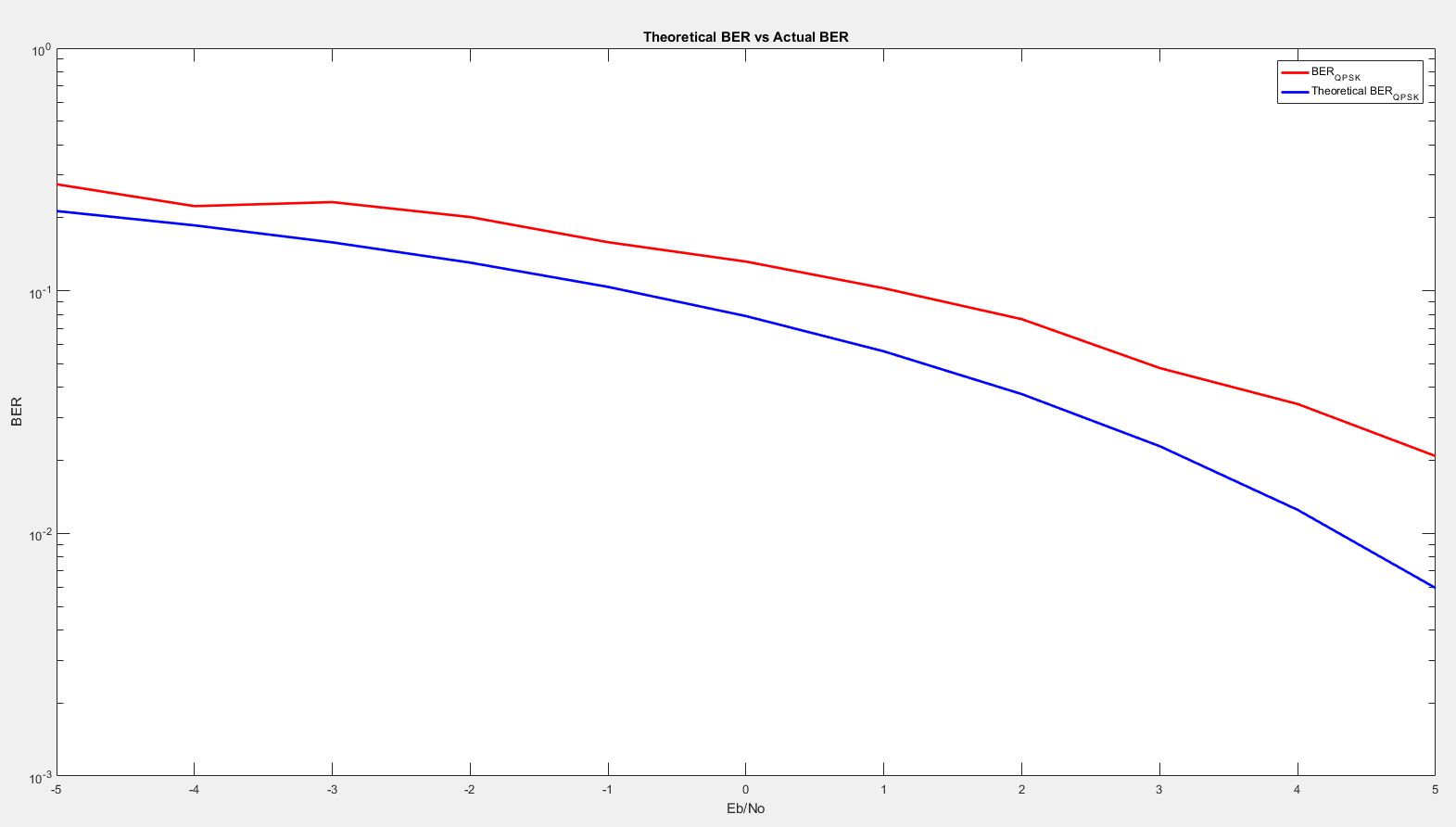


Figure 10: BER using channel equalization in EPA and AWGN channel