Hardware Emulations

using 5G Toolkit and SDRs: Hands-on

GIGAYASA

For academia only

Contributors:

Editor:	Simulator Design Team, GIGAYASA
Contributing Authors:	Aman Mishra, Vijaya Mareedu and Vikram Singh, GIGAYASA

Grants and Funding

Gigayasa is supported by:

- Indian Institute of Technology, Madras Incubation Center (IITM-IC).
- Startup India.
- Department of Telecommunication (DoT), India.
- Center of Excellence in Wireless Technology (CEWiT), IITM.
- Ministry of Electronics and Information Technology (MEITy), India.

Contents

1 | Channel Estimation and Equalization for SSB/PBCH using PBCH-DMRS

The objective of the wireless communication system is to decode the transmitted data at the receiver. Before decoding the data, the receiver must estimate the channel between the transmitter and itself. However, wireless channels are dynamic and subject to various environmental factors such as multipath fading, shadowing, and interference. By accurately estimating the wireless channel, the receiver can decode the transmitted data without any error . Error-free decoding of data results in higher throughput and lower BLER, thus making accurate channel estimation in wireless communication crucial. In this experiment, we will discuss different techniques for channel estimation and use this estimated channel to decode the SSB.

1.1 | Why Estimate Wireless Channel?

As radio signal propagate from transmitter to receiver, the signal propagate through the multi paths to reach the receiver. The signal received at the k^{th} subcarrier is given by,

$$Y[k] = H[k]X[k] + N[k], \tag{1.1}$$

where $Y[k]$, $H[k]$, $X[k]$ and $N[k]$ are the received signal, channel between transmitter and receiver, transmitted signal and noise at the receiver respectively. As discussed, before decoding the data, the receiver must obtain the accurate estimate of wireless channel. The channel estimation is performed using the pilot sequences, which are known both at the transmitter and receiver. These pilots are called Demodulation Reference Signal (DMRS) in 5G. As discussed in **??**, OFDM modulation is performed on 2D time-frequency grid as shown in figure 1.1 carrying DMRS and payload. The resources allocated to DMRS and PBCH symbols are demonstrated in figure 1.1.

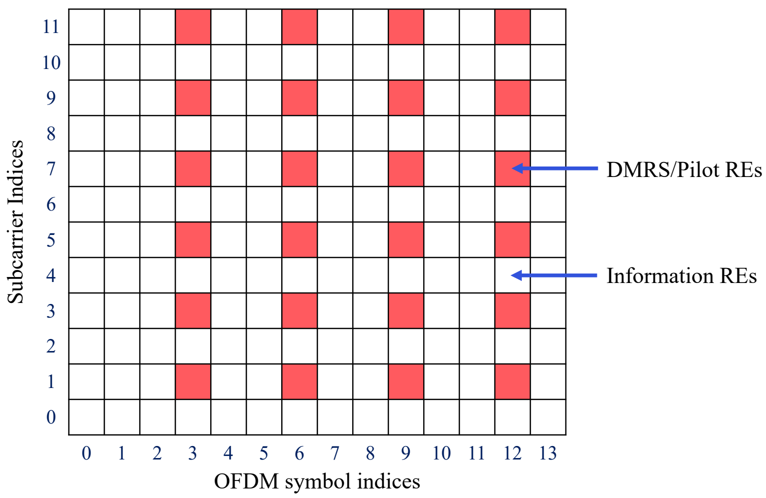


Figure 1.1: 5G Resource Grid

1.2 | Channel Estimation Techniques

The two most commonly used channel estimation technique in wireless communication are,

1. least square (LS) based channel estimation
2. minimum mean square error (MMSE) based channel estimation

The channel estimation is performed in two steps, as explained below.

1. The channel is estimated at pilot locations using known DMRS sequences.
2. Interpolate the channel at data locations using the channel estimated at DMRS locations. Different interpolation schemes can be used based on the rate of change in channel across time and frequency. 5G Toolkit supports nearest neighbour (NN), linear, cubic and spline interpolation. The interpolation techniques are discussed in detail in section 1.3.

1.2.1 | Least Square Based Channel Estimation

The LS estimation framework tries to minimize the error between the estimated channel and the actual channel based on first order approximation. The generalized LS channel estimate is given by,

$$\hat{H[k]} = Y[k](X[k]^H X[k])^{-1} X[k]^H, \tag{1.2}$$

where $\hat{H[k]}$ is the estimated channel matrix, $Y[k]$ is the received matrix and $X[k]$ is the transmitted pilot matrix. For a single layer transmission equation 1.2 reduces to,

$$\hat{H[k]} = Y[k]/X[k] \tag{1.3}$$

1.2.2 | Minimum Mean Square Error(MMSE) Channel Estimation

The MMSE estimator provides an estimate of the channel matrix $H[k]$ that minimizes the mean square error (MSE) between the estimated channel and the true channel. Mathematically, the MMSE estimator is given by,

$$\hat{H}_{MMSE} = R_{h,y} R_y, \tag{1.4}$$

Where $\hat{H}_{MMSE}$ is the MMSE estimate of the channel matrix, $R_{h,y}$ is the cross-correlation matrix between the true channel and the received signal, R_y is the autocorrelation matrix of the received signal. $R_{h,y}$ and R_y is estimated from DMRS.
The choice between LS and MMSE technique depends on following tradeoffs:

1. **Performance:** The MMSE estimator minimizes the mean square error between the estimated channel and the true channel. It takes into account the statistics of the noise and the channel, resulting in potentially better performance, especially in scenarios where there's significant noise or fading. Whereas, the LS estimator minimizes the sum of squared errors between the estimated and observed signals without considering the noise statistics. It is simpler and computationally less intensive but may suffer from higher error rates, particularly in noisy or frequency-selective fading channels.

2. **Robustness to Noise:** MMSE estimator considers noise statistics, it tends to be more robust in noisy environments. It can mitigate the effects of noise and provide better estimation accuracy. While LS estimator may be more sensitive to noise as it doesn't explicitly account for noise characteristics during estimation. MMSE is more robust to ill-conditioned channel matrix.

3. **Computational Complexity:** Generally, MMSE estimation involves more complex calculations, including matrix inversions and multiplications, making it computationally more expensive compared to LS estimation. While LS estimation involves simple matrix operations and is computationally less expensive compared to MMSE estimation.

1.3 | Channel Interpolation Techniques

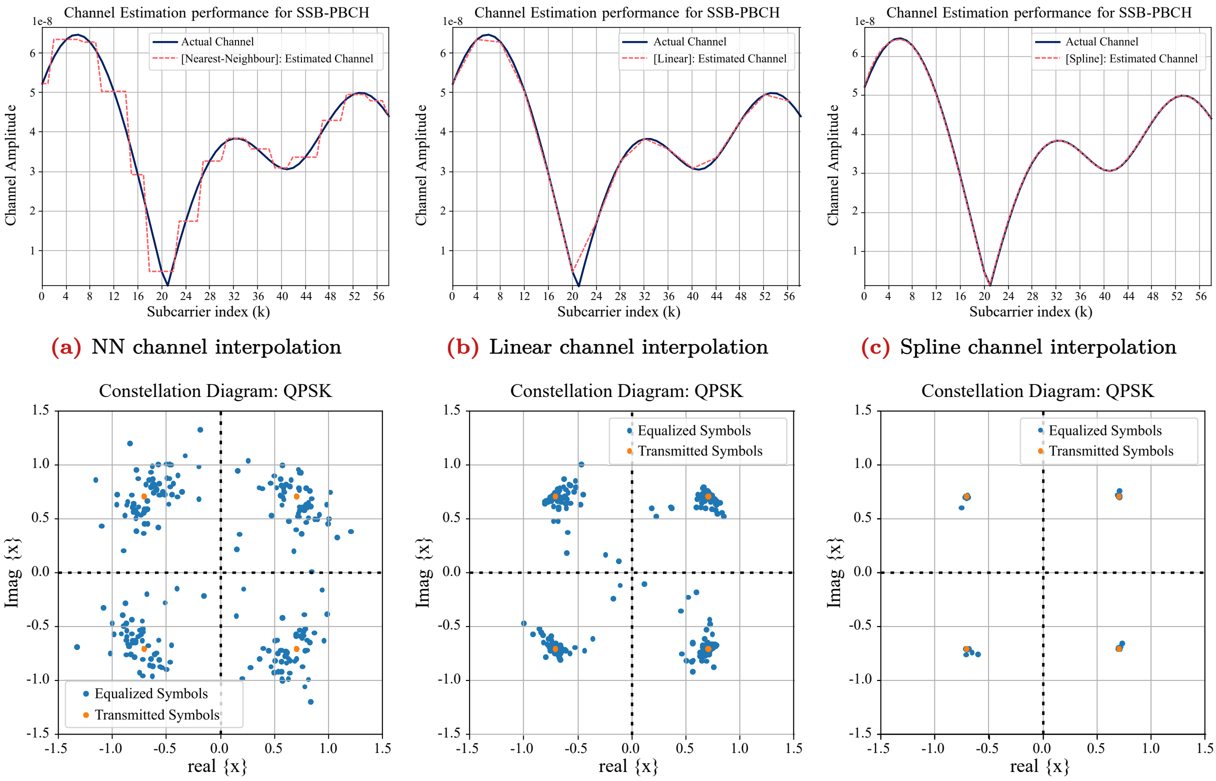
The 5G Toolkit employs four types of channel interpolators to accommodate different channel scenarios. The channel interpolators are listed below,

1. Nearest Neighbour (NN): It is the simplest channel interpolator with the least computational complexity. Once the channel is estimated at pilot locations, applying the nearest neighbour interpolator results in the channel estimate at data locations being equal to the channel estimated at the nearest pilot. NN works best for flat faded channels and lower subcarrier spacing. An example of NN interpolator is shown in figure 1.2a below. As observed from figure 1.2a, the NN interpolator estimated channel is not able to keep track of actual channel accurately.

2. Linear: Linear interpolation is a method of curve fitting using linear polynomials to construct new data points within the range of a discrete set of known data points. The computational complexity of linear interpolator is higher than that of NN interpolator. Linear interpolation works well for moderately varying channels and for lower subcarrier spacing. The magnitude plot of linear interpolator is shown in figure 1.2b below, the linear interpolated estimated channel is able to track the actual channel for most part except for some subcarriers.

3. Cubic: The cubic interpolator estimates the values between known data points by fitting a cubic polynomial curve through those points. The computational complexity of the cubic interpolator is higher than that of NN and linear interpolator. However, it can track rapidly varying channels and works satisfactorily for higher subcarrier spacing.

4. Spline: The spline interpolator involves fitting a piecewise polynomial function to the data points in such a way that the resulting curve passes through each data point smoothly. This smoothness is typically achieved by enforcing continuity of derivatives up to a certain order across adjacent polynomial segments. There are different types of spline interpolators, with cubic splines being one of the most common. Cubic splines use cubic polynomials between each pair of adjacent data points and ensure continuity of the first and second derivatives at each data point. This creates a smooth curve that passes through all the given data points without oscillations. The spline interpolator are best interpolators which can track rapidly varying channel and works best for higher subcarrier spacing. The magnitude plot of cubic spline interpolator is shown in figure 1.2c below. As observed, the cubic spline interpolated channel gives the best performance as it is able to track the channel of all the subcarriers. It must be noted that the cubic spline interpolator shown in figure 1.2c is equivalent to cubic interpolator. In general, the spline interpolators can be of higher order, however, it is observed that the the channel estimation performance of third order spline, a.k.a cubic spline is best and saturates for higher order splines.



(a) NN channel interpolation (b) Linear channel interpolation (c) Spline channel interpolation

(d) Constellation diagram: NN (e) Constellation diagram: Linear (f) Constellation diagram: Spline

Figure 1.2: Performance evaluation of different channel interpolation schemes for $\Delta f = 15$ kHz.

1.4 | What is Symbol Equalization?

Symbol equalization or channel equalization refers to a signal processing technique used to mitigate the effects of channel impairments on transmitted symbols. In wireless communication systems, signals transmitted from a transmitter to a receiver undergo various distortions and impairments due to factors such as multipath fading, shadowing, interference, etc. These phenomena results in the distortion and corruption in the transmitted symbols. Symbol equalization aims to neutralize these effects and recover the original symbols. By performing symbol equalization, the receiver can improve the accuracy of symbol decoding, thereby enhancing the overall performance of the communication system. This is particularly important in 5G systems, which aim to provide high data rates, low latency, and reliable connectivity in diverse and challenging deployment scenarios. Using LS based channel equalization, the symbols can be retrieved using the equation below,

$$\hat{X}[k] = (\hat{H}[k]^H \hat{H}[k])^{-1} \hat{H}[k]^H Y[k] \tag{1.5}$$

When receiving only single layer, equation 1.5 can be simplified to:

$$\hat{X}[k] = Y[k]/\hat{H}[k] \tag{1.6}$$

The complete process of channel estimation, interpolation, and channel equalization is illustrated in Figure 1.3.

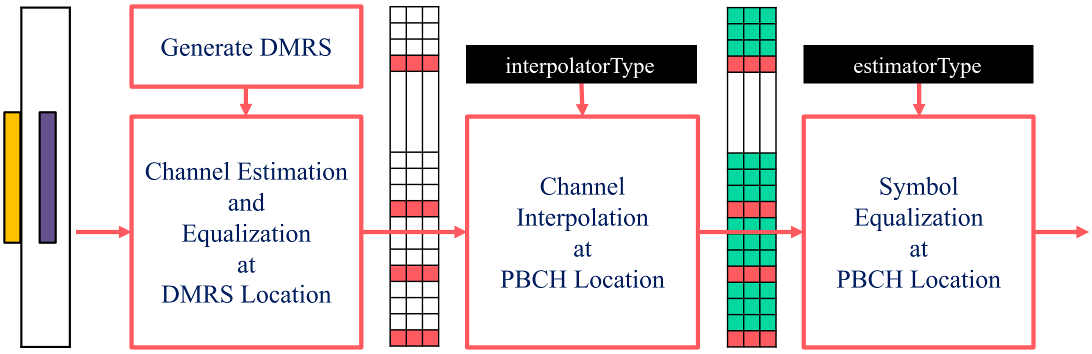


Figure 1.3: Channel estimation and equalization for PBCH.

1.5 | Results

The parameters used for hardare emulation are stated in table-1.1.

Table 1.1: Simulation parameters of PBCH channel estimation and symbol decoding

Parameters	Value
Carrier frequency (f_c)	1000 MHz
Bandwidth (B)	5/10/15 MHz
FFT size (N_{FFT})	1024
subcarrier spacing (Δf)	15/30/60* KHz
Transmitter-Receiver separation (d)	1 m
Channel estimator	zero forcing (ZF)
Channel interpolator	"NN"/"Linear"/"Cubic"

The above parameters hold unless explicitly stated. The tutorial explores channel estimation, channel interpolation, and symbol equalization for the physical broadcast channel (PBCH), which is the simplest and most important channel for communication in 5G networks. PBCH is carried by the synchronization signal block (SSB) in 5G networks, transmitting crucial network information. Further details about this channel will be discussed in the next chapter (Chapter-**??**).
The process at the receiver begins with time synchronization, followed by carrier frequency offset (CFO) correction. Once synchronization is achieved and the effects of hardware impairments are mitigated, the OFDM resource grid is reconstructed using the OFDM demodulator. From this resource grid, the SSB grid is extracted, which appears as shown in Fig-[1.4].
The channel estimation for PBCH requires two components: the demodulation reference signal (DMRS) sequence and the physical cell ID ($N_{\text{ID}}^{\text{cell}}$). The cell ID helps the UE identify the time-frequency locations where PBCH-DMRS is loaded for channel estimation. The cell ID is computed using two components: cell-ID-1 (N_{ID}^{1}) and cell-ID-2 (N_{ID}^{2}), which are detected using time synchronization (PSS detection) and synchronization signal (SSS) detection procedures respectively, as detailed in the Python code. The channel estimation is performed using DMRS, which is generated using the SSB index, half-frame index, and cell ID. These parameters are detected/estimated using the DMRS parameter detection module.

Observation-1: *Errors in the detection of cell-ID, SSB index, or half-frame index will result in failure to decode the synchronization signal block (SSB) and, ultimately, the master information block (MIB).*

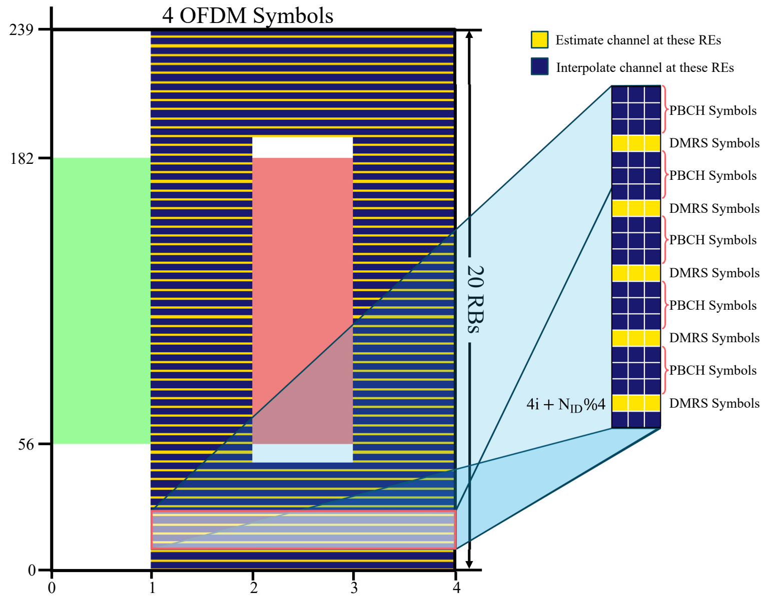


Figure 1.4: 5G Synchronization Signal Block (SSB).

The SSB, as shown in Fig-1.4, is deigned very high DMRS density in time and frequency which provides robustness against time and frequency selective channels. As the distance between the transmitter and receiver increase, the link budget deteriorates and frequency selectivity increases. However, its can be seen from the results in Fig-1.7q, that SSB can be detected even in harsh link budget conditions.

Observation-2: *The SSB is robust even in weak links, highly frequency-selective channels, and with high-mobility users.*

As the distance between the transmitter and receiver increases, the received power by the UE decreases, leading to significantly degraded signal-to-noise ratios, as depicted in Fig-[1.7a], [1.7e], [1.7i], [1.7m], [1.7q], and [1.8a]. Furthermore, Fig-[1.7m] and Fig-[1.8a] clearly show that SNRs are much lower in NLoS links compared to LoS links, alongside higher frequency selectivity.

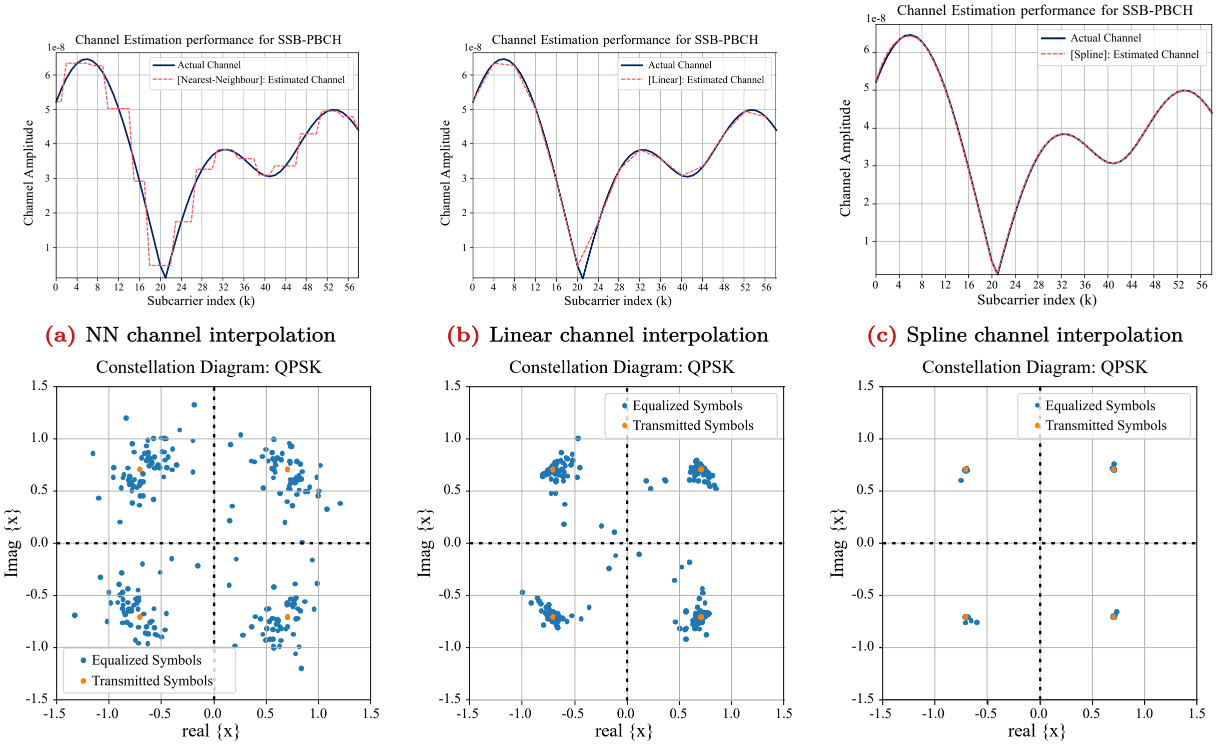
In LoS scenarios where links are typically strong and channels are nearly flat, the nearest neighbor interpolator delivers the best performance along with the least squares estimator. However, the results in Fig-[1.7d], [1.7h], [1.7l], [1.7p], [1.7t], and [1.8d] clearly demonstrate that linear interpolators are the most robust performers for most cases, even in weak frequency-selective channels. However, if the channel is highly frequency selective but the SNR is high, which occurs rarely, the spline interpolator delivers the best performance, as demonstrated by Fig-[1.2].

Observation-3: *The nearest neighbor interpolator is suitable when the channels are mostly flat, but higher-order interpolators such as linear or spline interpolation are required to track the rapid variations in wireless channels resulting from frequency selectivity.*

As the subcarrier spacing increases, the variations in the channel become more abrupt due to higher sample rate and wider frequency bin in frequency domain as shown Fig-[1.2] and Fig-[1.6]. In such scenarios, spline interpolators are the most suitable when the SNRs are high, while linear interpolators are more appropriate when the SNR is low.

Observation-4: *As the subcarrier spacing increases, the channel amplitude varies much more rapidly compared to lower subcarrier spacing, requiring higher-order interpolators for accurate channel estimation and equalization.*

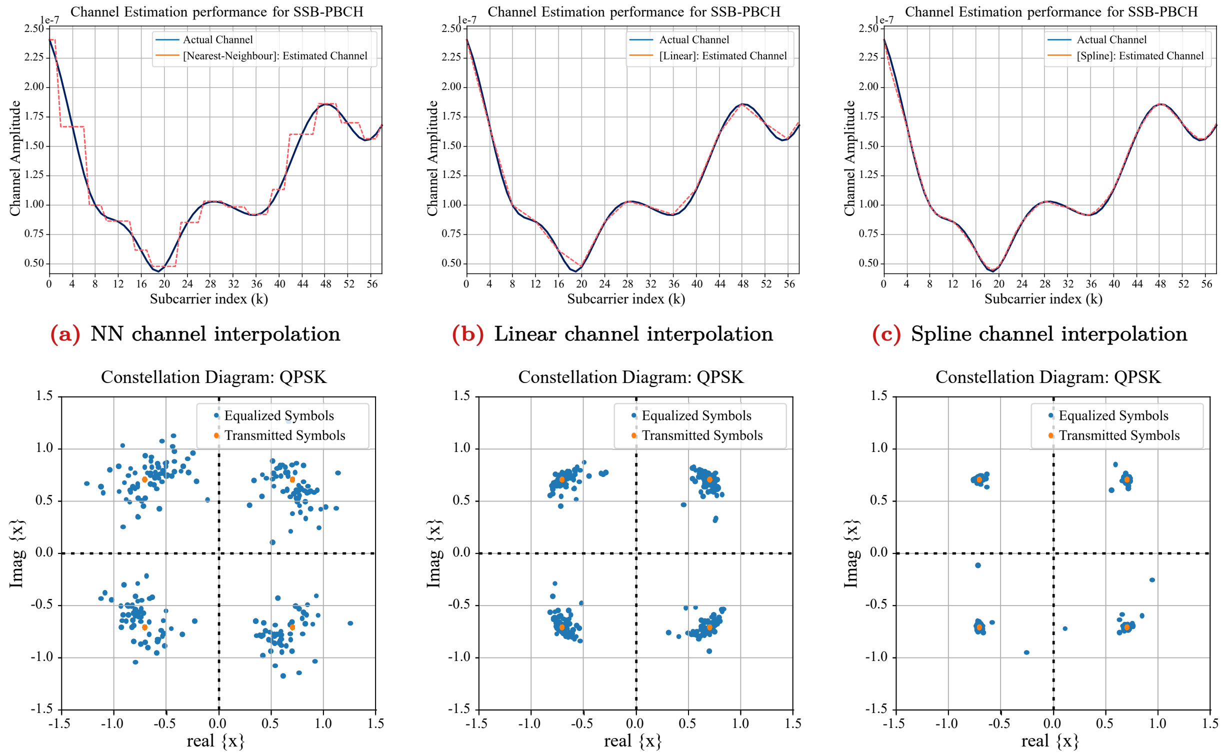
1.6 | Appendix



(a) NN channel interpolation (b) Linear channel interpolation (c) Spline channel interpolation

(d) Constellation diagram: NN (e) Constellation diagram: Linear (f) Constellation diagram: Spline

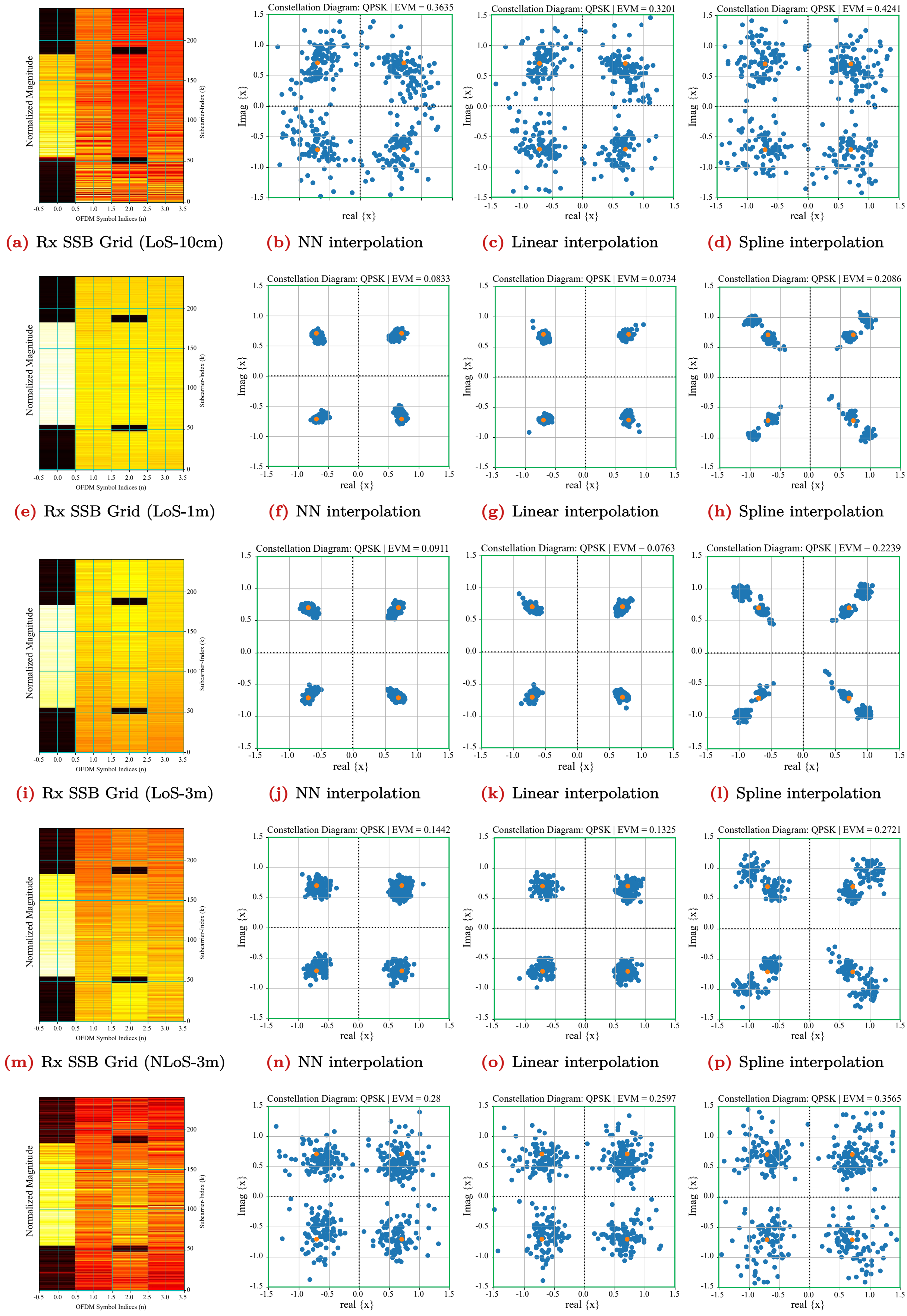
Figure 1.5: Performance evaluation of different channel interpolation schemes for $\Delta f = 30$ kHz.



(a) NN channel interpolation (b) Linear channel interpolation (c) Spline channel interpolation

(d) Constellation diagram: NN (e) Constellation diagram: Linear (f) Constellation diagram: Spline

Figure 1.6: Performance evaluation of different channel interpolation schemes for $\Delta f = 60$ kHz.



(a) Rx SSB Grid (LoS-10cm) (b) NN interpolation (c) Linear interpolation (d) Spline interpolation

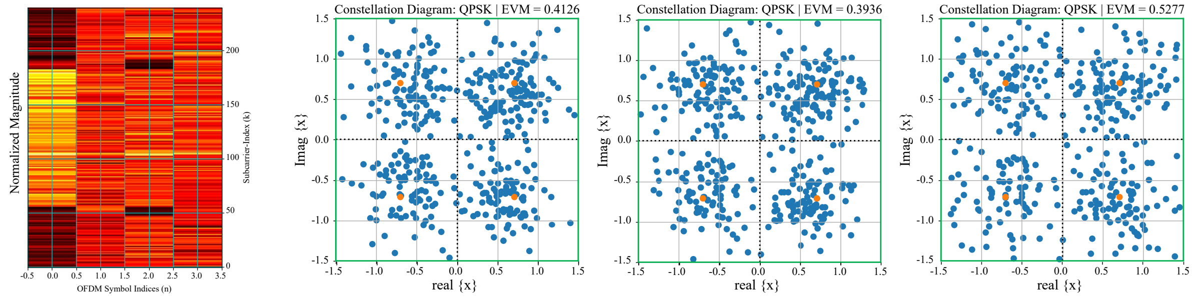
(e) Rx SSB Grid (LoS-1m) (f) NN interpolation (g) Linear interpolation (h) Spline interpolation

(i) Rx SSB Grid (LoS-3m) (j) NN interpolation (k) Linear interpolation (l) Spline interpolation

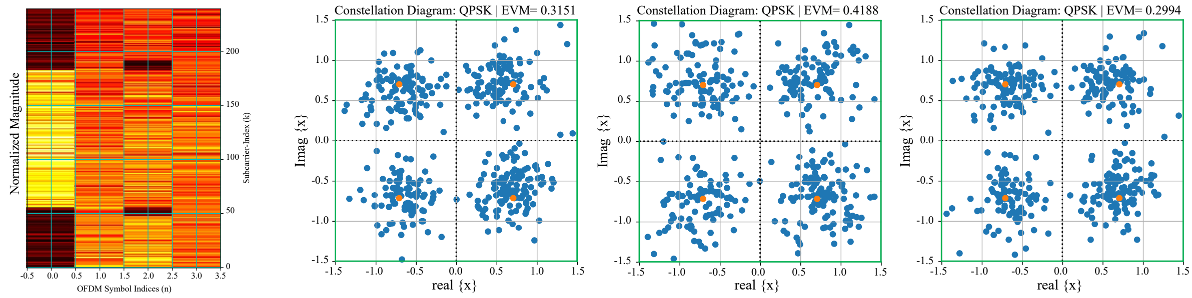
(m) Rx SSB Grid (NLoS-3m) (n) NN interpolation (o) Linear interpolation (p) Spline interpolation

(q) Rx SSB Grid (LoS-10m) (r) NN interpolation (s) Linear interpolation (t) Spline interpolation

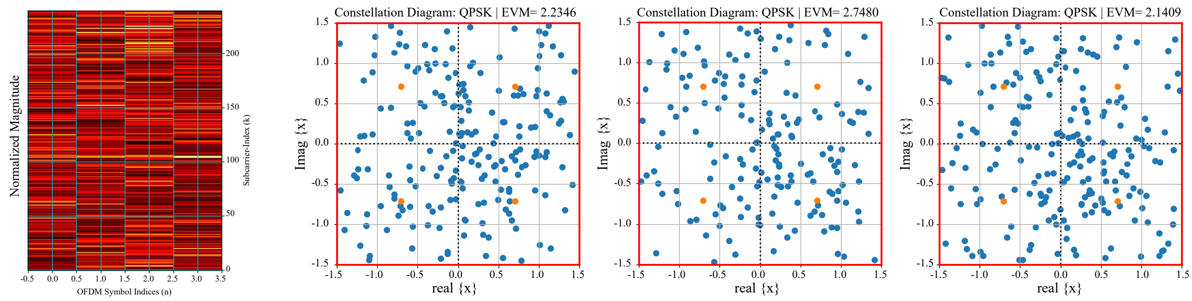
Figure 1.7: Performance evaluation of least square channel estimator combined with nearest neighbour (NN), linear and spline interpolator for different link state (NLoS/LoS) and Tx-Rx separations.



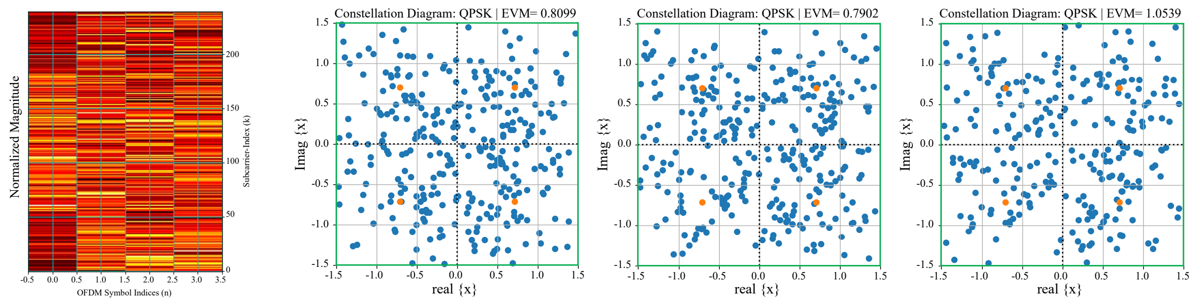
(a) Rx SSB Grid (NLoS-10m) **(b)** NN interpolation **(c)** Linear interpolation **(d)** Spline interpolation



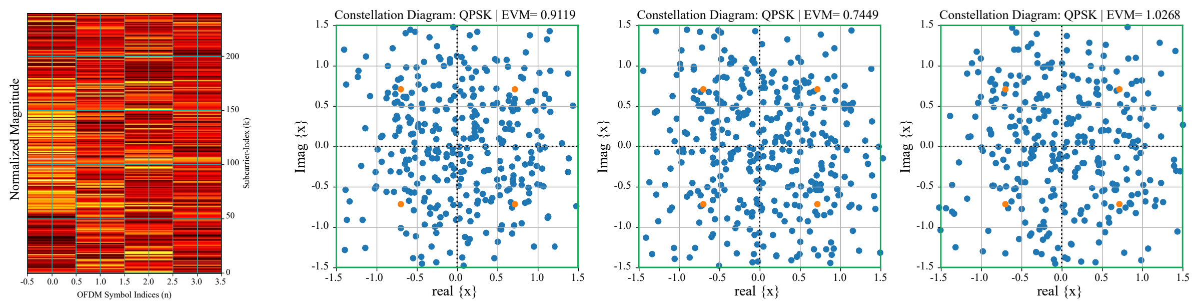
(e) Rx SSB Grid (LoS-15m) **(f)** NN interpolation **(g)** Linear interpolation **(h)** Spline interpolation



(i) Rx SSB Grid (NLoS-15m) **(j)** NN interpolation **(k)** Linear interpolation **(l)** Spline interpolation



(m) Rx SSB Grid (LoS-50m) **(n)** NN interpolation **(o)** Linear interpolation **(p)** Spline interpolation



(q) Rx SSB Grid (LoS-85m) **(r)** NN interpolation **(s)** Linear interpolation **(t)** Spline interpolation

Figure 1.8: Performance evaluation of least square channel estimator combined with nearest neighbour (NN), linear and spline interpolator for different link state (NLoS/LoS) and Tx-Rx separations.

1.7 | Further Reading

- Read more on Interpolation.
- Read about PBCH
- Read about PBCH DMRS

2 | References