Understand symbol alignment

Symbol Alignment The OFDM Sync Long block is also responsible for symbol alignment. Each OFDM symbol spans 80 samples, consisting of 16 samples of cyclic prefix and 64 data samples. The task of symbol alignment is to calculate where the symbol starts, to extract the data symbols, and to feed them to an algorithm doing a Fast Fourier Transformation (FFT). This alignment is done with the help of the long training sequence, which is composed of a 64 sample long pattern that repeats 2.5 times. As this block needs only act on a subset of the incoming sample stream, and as the alignment has to be very precise, we employ matched filtering for this operation. In Figure showed above, is presented a typical graph showing the correlation of the input stream with the known sequence is reproduced. The two characteristic peaks are very dominant and narrow, thus allowing very precise symbol alignment. We calculate the indices of the highest three peaks as

$$N_{\mathcal{P}} = \underset{n \in \{0,\ldots,N_{\text{preamble}}\}}{\arg\max_3} \sum_{k=0}^{63} \mathrm{s}[n+k]\,\overline{\mathrm{LT}}[k],$$

where N_{preamble} corresponds to the added length of the short and long preambles, LT is the repeating pattern of the long training sequence spanning 64 samples, and arg max_3 returns the top 3 indices maximizing the expression. The first data symbol thus starts at sample index

$$n_p = max(N_p) + 64,$$

as the latest peak of the matched filter output is 64 samples before the end of the long training sequence. With the relative position of the first data symbol known, this block can extract the data symbols, then pass chunks of data samples that correspond to one symbol to subsequent blocks in the flow graph. The first symbol of each OFDM frame is tagged, so that the following blocks are able to recognize the frame start. Knowing the start of the data symbols, we can remove the cyclic prefix by subsetting the data stream and grouping the samples that correspond to individual data symbols as

$$\mathrm{s} \leftarrow \Big(\underbrace{\mathrm{s}[n_{\mathcal{P}}+16],\ldots,\mathrm{s}[n_{\mathcal{P}}+79]}_{\text{first symbol}},\underbrace{\mathrm{s}[n_{\mathcal{P}}+80+16],\ldots}_{\text{second symbol}}\Big)$$

Long-training symbol synchronization in IEEE 802.11 receivers is achieved by correlating the incoming signal with a known reference sequence. The goal is to determine the sample index at which the long-training symbol (LTS) begins.

The mathematical expression N_p defines this operation as a **64-point sliding correlation** between the received baseband samples $s[n]$and the conjugated LTS reference $LT[k]$. The receiver computes this correlation for every possible offset n within the preamble region and selects the value of n that maximizes it.

In the implementation, this procedure is executed by the GNU Radio block ieee802_11.sync_long. This block receives the aligned signal from the short-training synchronization stage and performs the correlation internally. The resulting peak index is used to accurately locate the OFDM symbol boundary, enabling correct FFT demodulation in subsequent processing stages.

Why Matched Filtering Is Used for Symbol Alignment but Not for Frame Detection

In OFDM–based systems such as IEEE 802.11a/g/p, the receiver must perform two distinct tasks during the preamble processing: **frame detection** and **symbol alignment**. Although both are forms of synchronization, they rely on different signal properties and therefore require different signal-processing techniques.

1. Frame Detection

Frame detection aims to determine **whether a valid OFDM transmission is present** and to obtain a coarse estimate of the packet start. For this purpose, IEEE 802.11 employs **autocorrelation** of the Short Training Field (STF), which contains periodic repetitions every 16 samples.

Autocorrelation is preferred because:

- **It does not require time alignment.** The structure of the STF allows detection even when the start of the packet is unknown.
- **It is robust to carrier-frequency offset (CFO).** The CFO induces the same phase rotation on both signals being compared, so the autocorrelation peak remains strong.
- **It has low computational cost.** Only a sliding comparison between $r[n]$and $r[n + 16]$is required.

Due to these advantages, matched filtering is not used for frame detection; it would require prior alignment and is computationally heavy and sensitive to CFO at this stage.