Windowed overlapped frequency domain block filtering approach for direct sequence signal acquisition

Ebrahim Karami\textsuperscript{a,}\textsuperscript{*}, Harri Saarnisaari\textsuperscript{b}

\textsuperscript{a} Department of Engineering and Applied Sciences, Memorial University, Canada
\textsuperscript{b} Centre for Wireless Communications, University of Oulu, Finland

\textbf{A B S T R A C T}

This study applies a windowed frequency domain overlapped block filtering approach to acquire direct sequence signals. As a novel viewpoint, the windows not only allow pulse shaping without front-end pulse-shaping filter, but also improve the performance of the spectrum sensing unit, which can efficiently be implemented into this frequency domain receiver and may further be used for spectrum sensing in cognitive radios or narrowband interference cancellation in military radios. The proposed receiver is applicable for the initial time synchronization of different signals containing a preamble. These signals include single carrier, constant envelope single carrier, multicarrier, and even generalized multicharrier signals, making the proposed receiver structure a universal unit. Furthermore, the receiver can be used to perform filtering with long codes and compute the sliding correlation of an unknown periodic preamble. The receiver can further be modified to handle large Doppler shifts. We will also demonstrate herein the computational complexity and analysis of the acquisition performance in Rayleigh and Rician fading channels.

1. Introduction

Initial synchronization or the acquisition of a Direct Sequence (DS) signal appears to be a common first step that a communication receiver must perform after switching on the power because many wireless standards use either DS signaling or their preamble, which is used for synchronization purposes, is a DS signal. These standards include GSM, LTE, UMTS, GPS, GALILEO, WIMAX, Zigbee, and many others [1–4]. LTE systems use two DS signals (i.e., a 62-long Zadoff-Chu sequence and a 31-long M-sequence), as primary and secondary synchronization signals [5]. Meanwhile, military communication systems require a robust acquisition. One method of increasing the synchronization robustness is to use Interference Cancellation (IC) signal processing. The notch filters are a well-known example of these. Another use of these IC units is spectrum sensing in cognitive radios. A notch filter may be a separate stand-alone unit in the front of a conventional receiver, but it may also be integrated into a frequency domain receiver, which reduces complexity because needed transformations may be shared. Frequency domain receivers have somewhat recently attained interest because they offer complexity reduction (e.g., in filtering). One particularly interesting filtering is matched filtering, which allows fast acquisition [6,7]. In traditional frequency domain filtering, where the filter is in one piece, OverLap-Save (OLS) or OverLap-Add (OLA) methods must be acquired to properly handle the convolution process [8]. In addition, the frequency domain receivers may be of interest as multipurpose, or universal, receivers because they, by their very nature, can be used to receive multi-carrier signals, like Orthogonal Frequency Division Multiplexing (OFDM), its variants as MC-CDMA [9], and Generalized Multi Carrier (GMC) [10] signals. As have been well known, they can also receive single carrier signals [11].

Some systems have rather long DS codes, which require long filters that are difficult to implement. In such a case, filtering must be divided into blocks. The required filtering process must be performed using block or partitioned filtering [12]. This technique is well known in audio signal processing [13,14], and even overlapping blocks may be used [14,15]. Block filtering may also be adopted to acquire larger Doppler shifts than sole filters (see Ref. [16] for the time domain approach). Block filtering is equal not only to Discrete Fourier Transform (DFT) filter banks (multirate filters) and Linear Periodic Time Varying (LPTV) filtering [12], but also to Short-Time Fourier Transform (STFT)-based filtering [17]. The STFT adds windows, which are not used in DFT filter banks or LPTV filters, to the overall picture. The windows may be used to perform the pulse shape filtering to match the filter frequency response to that of the signal and improve the performance of notch filters by reducing the spectral...
leakage. However, although essential for the proper performance of notch filters, windowing is known to cause Signal-to-Noise Ratio (SNR) losses of up to 3 dB for good windows. This loss may be reduced almost down to zero dB using overlapping segments, which are also elementary for the STFT-based processing [18]. An STFT-based correlator DS receiver has been presented in Ref. [19]. In addition to the data demodulation investigated in Ref. [19], the receiver may be used for serial search acquisition, which is known to result to a slower acquisition than the matched filtering acquisition investigated herein.

This study presents a frequency domain, windowed, overlapped block filtering approach for DS signal acquisition. In addition to introducing the filtering and acquisition concept, other possible applications in radio communications will also be briefly discussed here. These applications include the following points: i) addition of a particular notch filter method [20] into the receiver chain; ii) processing of different signals, like conventional DS, constant envelope DS, OFDM (WiMAX), MC-CDMA, and GMS; iii) adaption of the receiver to handle large Doppler frequency uncertainties; and iv) possible changes when the receiver is turned to the demodulation phase after acquisition. Furthermore, this study includes an analysis of the computational complexity of the receiver compared to the conventional (non-block) matched filter implementation in the time or frequency domain as well as an analysis of the acquisition probabilities in Additive White Gaussian Noise (AWGN) and Rayleight flat fading channels, of which the latter denotes novel results. The probabilities include conventional detection and false alarm probabilities, maximum search-based probabilities, and maximum search followed by threshold detection-based probabilities offering a very comprehensive picture of the receiver’s performance. These probabilities may be used to set the detection threshold and predict the receivers performance in practice. In summary, this study introduces a flexible baseband architecture that may be used with most existing and future signals and offers a spectrum sensing or narrowband interference rejection capability with a low additional cost. Therefore, the proposed receiver structure is a candidate receiver architecture for future multiwaveform platforms.

The remainder of the paper is organized as follows: Section 2 introduces the filtering concept; Section 3 presents the applications and modifications; Section 4 discusses the analysis of the acquisition process; Section 5 shows the simulation results confirming the analysis; and finally, Section 6 draws the conclusions.

2. Block filtering

This section first discusses the blockwise convolution to provide an insight on how block filtering works. Subsequently, its mathematical frequency domain basis, which is the STFT-based time-varying filtering, is presented.

2.1. Example

A simple example is probably the best way to explain how block filtering differs from the conventional one. Let \(x_1, x_2, x_3, x_4\) be the signal block to be filtered. In the conventional case, the signal is step-by-step put through the filter with an impulse response of \(h_1, h_2, h_3, h_4\). Consequently, the response sequence is \(x_1 h_1, x_1 h_2 + x_2 h_1, x_1 h_3 + x_2 h_2 + x_3 h_1, x_1 h_4 + x_2 h_3 + x_3 h_2 + x_4 h_1\) (desired phase in acquisition), \(x_2 h_4 + x_3 h_3 + x_4 h_2 + x_1 h_4\). The block-wise convolution should end up to the same response.

In block processing, the signal and the filter are divided into blocks using equal division. The division of the signal in the example could be presented as follows (the filter is divided correspondingly):

\[
\begin{bmatrix}
  x_3 & x_1 \\
  x_4 & x_2
\end{bmatrix}
\]

where the block size \(M = 2\), and the totality is a \(2 \times 2\) matrix. The signal would include zero blocks in both sides in the absence of noise. In other words, the signal matrix stream can be presented as:

\[
\begin{bmatrix}
  0 & x_3 & x_1 & 0 \\
  0 & x_4 & x_2 & 0
\end{bmatrix}
\]

which is divided into \(2 \times 2\) matrices by discarding the oldest data and taking a new block in. Therefore, the input matrices, the first on the right, are:

\[
\begin{bmatrix}
  x_1 & 0 & x_1 & x_1 & 0 & x_1 \\
  x_2 & 0 & x_4 & x_2 & 0 & x_4
\end{bmatrix}
\]

In block processing, an input matrix is processed at a time instant, called a filtering cycle. Each block (column) of an input matrix is convolved with the corresponding block (column) of a filter. The results are then summed up. Subsequently, the next input matrix in the next cycle is taken in, and the operations are repeated, such that the \(M\) responses are calculated in one time cycle. The operation must be repeated for all \(L\) possible cycles to obtain the whole response. The block convolution length is \(2M - 1\). Hence, the tails have to be added to the corresponding convolutions in the next cycle. This aspect is clarified next. Each block (column) of the signal (matrix) is assumed to pass a filter block from down to top. The corresponding convolution results are added together from each filtering cycle. The cycles are separated by bars, and the tails are below the dotted lines.

\[
\begin{align*}
x_1 h_1 & \quad x_1 h_1 + x_1 h_1 & \quad x_1 h_1 \\
x_1 h_2 + x_2 h_1 & \quad x_1 h_2 + x_2 h_1 + x_2 h_3 & \quad x_1 h_2 + x_2 h_3 \\
x_1 h_3 & \quad x_1 h_3 + x_2 h_2 + x_3 h_1 & \quad x_1 h_3 + x_2 h_2 + x_3 h_1 \\
x_1 h_4 & \quad x_1 h_4 + x_2 h_3 + x_3 h_2 + x_4 h_1 & \quad x_1 h_4 + x_2 h_3 + x_3 h_2 + x_4 h_1
\end{align*}
\]

The tails of the convolution must be added to the head of the convolution in the next cycle. More precisely, let \(c_k = [h_k h_{k+1}]\) denotes the convolution result in cycle \(k\), where \(h_k\) is the head (first \(M\) samples), and \(h_{k+1}\) is the tail. In the next cycle \(c_{k+1} = [h_{k+1} h_{k+2}]\). This way the response of the block convolution becomes equivalent to that of the conventional convolution. In summary, the signal stream is block-by-block stepped through the filter. The columnwise convolution between the signal and the filter is then performed. The convolution results are added columnwise, and the tails must be added to the head of the next cycle. The convolution in the time domain may equally be done well in the frequency domain. Hence, the signal’s FFT (matrix) in each cycle can be elementwise multiplied by the filter’s FFT (matrix), then inverse-transformed to obtain the time domain convolution. The convolution results are then added together, and OLA processing is performed. However, only one FFT per incoming signal block must be calculated because these transformations flow matrix-wise through the filter.

Using a similar example, it would be easy to see that the response of the blockwise convolution in the over-lapping segment case (e.g., \(x_1, x_2; x_2, x_3; x_3, x_4\)) would not be equal to that of the conventional convolution. Instead, the original signal and its overlapped version must process separately, and the results must be added afterwards. The filter must be overlapped correspondingly.

2.2. STFT-based block filtering

All this data is put into the STFT framework as follows. Let: \(x(n), n = 0, \ldots, N - 1\) be a discrete signal with the following STFT [17]:

\[
X_m = \sum_{n=0}^{N-1} x(n)w(n - IR)e^{2\pi jmn/M}
\]  

(1)

where the analysis window \(w(n)\) has length \(M\), with the nonzero values being in the interval \(n = 0, \ldots, M - 1\). The signal is divided into blocks of \(M\) samples, and the blocks may overlap depending on the parameter \(R\).
No overlapping occurs if $R = M$, only consecutive blocks are observed. As a result of the analysis process (1), the signal is presented by an $M \times LM/R$ array of coefficients $X_{lm}$ assuming, for simplicity, that $N = LM$ and $M/R = 1, 2, 4, \ldots$. The case $M = R$ is called the critical sampling case. The selected restrictions yield to a simple implementation through FFT, but are still quite flexible. A more general case has been studied in Ref. [21], but without considering the signal acquisition.

Several alternatives can be used to recover the signal [17]. One particularly interesting form is:

$$x(n) = \sum_{l=0}^{L-1} g(n - IR) \sum_{m=0}^{M-1} X_{lm} e^{j2\pi nm/M}$$

(2)

where $g(n)$ is the synthesis window of length $M$. Assuming that $w(n)$ and $g(n)$ satisfy some restrictions [17], the signal $x(n)$ can be perfectly reconstructed (synthesized) from its STFT coefficients $X_{lm}$. In other words, the STFT columns are first inverse Fourier transformed (rightmost sum in Eq. (2)), then windowed and finally added together in OLA fashion. Note that the order of addition and (I)FFT can be changed because the (I)FFT is a linear operator. Thus, if the synthesis window is rectangular, the complexity may be reduced by performing addition before the IFFT. Naturally, this is a sensible operation only if the partial filtering results are not needed, like in Doppler processing or in filtering several symbols during a filtering cycle.

Let $H_{lm}$ be the STFT of the filter. The filter’s output [17] is the inverse STFT of $X_{lm} H_{lm}$ (elementwise product). Thus, filtering includes the multiplication of the signal’s STFT by that of the filter and the inverse transformation of the product. In the study’s case, the frequency response of the filter is zero outside an interval. Therefore, the output is computed by multiplying a finite portion of the signal’s STFT by the filter’s STFT. In addition, the FFT size must be $2M$ to properly handle the heads and the tails. The overlapping effect is considered by stepping the input signal STFT stream in steps of size $M/R$, which is the number of overlapping segments. Fig. 1 presents the filtering process. The described block FFT filtering method is reduced to the conventional FFT OLA filtering if $N = M = R$ [13].

2.3. Complexity comparison

The complexity of generic time, conventional frequency domain, and block filtering are compared herein in terms of Complex Multiplications (CM). In the time domain, each output needs $N$ CM and $N$ outputs exist, such that total complexity is $N^2$ CM. The conventional frequency domain OLA processing needs the FFT and the IFFT of size $2N$ and multiplication by the filter frequency response of size $2N$ yielding a total complexity of $2N(\log_2 N + 1)$ CM.

The proposed block filtering (assuming rectangular windows) requires the FFT of size $2M$, which must be repeated for $LM/R$ times by multiplying by the filter of size $2M \times LM/R$ that must be repeated for $LM/R$ times. The conventional form still needs the $LM/R$ IFFTs of size $2M$, whereas the simpler form only has $L$ IFFTs. Therefore, the total complexity of the conventional form is

$$\frac{2LM}{R} \left( \log_2 2M + \frac{L}{R} \right)$$

CM, while that of the simpler form is

$$\frac{2LM}{R} \left( \frac{L}{R} + \frac{N}{C0} \right) \log_2 2M + \frac{L}{R}$$

CM. The complexity of block filtering becomes equal compared to that of the conventional OLA filtering if $N = M = R$.

The complexity comparison shows that block filtering is more complex than the conventional one, but the complexity increase is marginal without overlapping. In addition, both the frequency domain versions are simpler compared to the generic time domain implementation. However, possible windowing increases the complexity.

3. Applications

3.1. Matched filtering for acquisition

Symbol or chip synchronization is conventionally performed by correlation. However, this process results in a slow synchronization phase (see Ref. [6]). One method of speeding up is to implement several correlators in parallel to compute a number of test variables (search cells) at the same time [22]. If the signal to be synchronized comprises $N$ symbols or chips, the receiver usually has $qN$ search cells in the time domain, where $q$ is the oversampling factor. In addition, search cells might be found in the frequency, as will be seen later. This section only investigates the time uncertainty. The receiver usually first includes a pulse shape filter with a response fed to the correlator for one sample per symbol or chip basis. In the oversampling case, the response must be split into $q$ steams. Each stream is then processed separately [23]. Alternatively, the pulse shape may be considered in the correlation [24]. In this case, the receiver does not include a separate pulse-shaping filter. This processing may be called the waveform based correlation. The other processing may be called the (training) symbol or chip sequence-based correlation. The STFT-based block filtering may adopt both ways. In the former, the analysis window may indeed be matched to form a suitable response. Another method to speed up is to calculate the test variables through matched filtering either in time or frequency domain [22,24,25].

In the serial search-matched filter (MF) acquisition, the outputs of the MF (computed in any possible way) are compared to a threshold in a serial fashion. In the maximum search, a period of outputs is calculated, and the maximum is found. This maximum is then compared to the threshold. As has been claimed, the signal is present in both cases, and symbol or chip synchronization is acquired if the threshold is exceeded.

Example with $L=2$ and $R=M/2$

![Fig. 1. Windowed and overlapped block filtering approach.](image-url)
This finding will be considered more in detail in Section 4.

3.2. Different modulations

OFDM systems are a special case of the synthesized signal (2) (i.e., the window is rectangular, and \( L = 1 \)). WIMAX synchronization is conventionally performed in the time domain. The preamble DS code in WIMAX is put into even subcarriers. Meanwhile, the odd ones are zero, thereby making the time domain signal periodic with two periods of size \( N/2 \), where \( N \) is the number of subcarriers. The acquisition unit performs a sliding correlation between two consecutive \( N/2 \) blocks [26]. Other variants, as well as the possibility that the matched filter and frequency domain processing is used, exist [27]. Note that frequency domain processing may be used to compute the sliding correlation. However, sample-by-sample sliding results to a high complexity for both the time and frequency domains. Therefore, the sliding step may be larger (e.g., \( N/4 \) or \( N/8 \)) and is closely related to the overlap processing.

The Generalized MultiCarrier (GMC) transmission technique presented in Refs. [11,21,28,29] is a possible candidate for future wireless communication systems because of its better time-frequency localization properties, which may reduce intersymbol and intercarrier interference and remove the need for the cyclic prefix needed in conventional OFDM systems [10]. The papers being referred to considered different aspects of the GMC signal, but not synchronization. Ref. [11] mentioned that synchronization may be performed on a subcarrier basis. The GMC signal may be explained as follows: the STFT coefficients \( X_n \) are the transmitted data symbols. In this case, signal (2) is called the GMC signal, or, if considered during interval 0, …, \( N - 1 \), a GMC symbol corresponding to the OFDM symbol definition. If a GMC system uses a known preamble symbol or symbols, its acquisition can be performed just as described herein. In the authors' knowledge, this is the first study considering the acquisition of GMC signals.

Linearly modulated single carrier signals can be obtained by setting \( M = 1 \). However, one might also want to receive these signals using the frequency domain processing, instead of the conventional time domain processing, to keep the receiver universal. This is possible because filtering, which is essential to all the receivers, may be done either in the time or frequency domain. This study has readily shown how this is done using frequency domain block filtering. Please note that constant envelope DS signals may also be received using a conventional matched filter [30], and thus, the proposed frequency domain block filter.

3.3. Doppler processing

In ideal Doppler processing, the input signal is transformed into different frequency offsets corresponding to the possible Doppler values. In a simpler solution, the filter is divided (partitioned) into blocks, and the outputs of these blocks are Fourier-transformed as shown in Ref. [16] (and references therein). The reference uses time domain processing. However, as already shown, the partitioned matched filtering can also be done in the frequency domain. If the Doppler is changing (because of accelerated motion) between the blocks, one may possibly search over all possible (but sensible) Doppler filters in the resulting time-frequency uncertainty grid. The acquisition probabilities concerning the Doppler processing are analyzed in Ref. [16] and not repeated in this paper. Another method of increasing the Doppler resistance is to combine the partial responses either non-coherently or in a differentially coherent way [31].

3.4. Long codes

Some systems have a basic long code, and a direct implementation of a filter matched to it may be infeasible. Block filtering is a possible solution with shorter elements that are feasible to implement. Another case, where the block matched filters may be needed, is that when a long code is divided into subintervals, each subinterval contains a symbol. In this case, the responses of the partitioned matched filters are the variables used for the symbol demodulation (naturally sampled at a symbol synchron). This may be needed in a long code system, where the data rate is adjusted using code partitioning. However, for some reasons, short DS codes are not will be used.

A possible example where block filtering may be applied is the UMTS system, where the uplink preamble consists of several scrambled repeats of a short code [32].

3.5. Spectrum sensing

Frequency domain processing allows the easy adaption of spectrum sensing algorithms because the FFT is readily included into the processing chain. Spectrum sensing may be applied in cognitive radios to find available spectrum holes. Another application is interference cancellation (IC), which is needed especially in military systems. In these cases, the process is known as notch filtering. However, in both cases, the technique is basically the same. The window inherent to the proposed receiver is helpful because it reduces spectral leakage. However, a drawback of the windowing is the SNR loss, which may be 3 dB. Overlapping, which is also inherent to the receiver, reduces this loss almost down to zero dB. An automatic narrowband signal detector is the CME algorithm [20] or its extended version, which is the LAD algorithm that clusters the found signals [33].

An important aspect to note when doing spectrum sensing or IC is that if the desired underlaying signal is not flat, or white, in the frequency domain, it may also be detected (if SNR is high enough) or, worse, canceled. The receiver should be designed using one sample per symbol/ chip processing (i.e., the receiver should have a traditional pulse-shaping filter at the front and parallel processing of the oversampled streams) to avoid these unpleasant phenomena. In this case, we may lose an advantage of the windows, but the complexity remains (almost) the same.

3.6. Demodulation

Once the acquisition is finished, the receiver turns its attention to tracking and data demodulation. In this turn, the receiver may continue matched filtering if the signal has a DS component. The block filtering allows different code lengths; the short codes are needed at high data rates, while the long ones are used in low data rates or when the DS processing gain is needed for interference tolerance. In addition, the time varying nature [17] of the filter allows despreading of scrambled signals. However, in this case, the filter’s or correlator’s frequency response must be frequently updated. Alternatively, the receiver uses a correlation in the DS component case, pure FFT in the OFDM case, or frequency domain pulse shape filtering in the single carrier case. The filter in the latter may filter several symbols at one filtering cycle, and the filter’s frequency response is only the pulse shape. This pulse shaping goal may also be achieved using a suitable analysis window.

4. Acquisition analysis

One usually wants to use detectors that are not sensitive to the signal level variations and are often called ‘constant false alarm rate (CFAR) detectors’. These CFAR detectors may also be derived using generalized likelihood ratio detectors [34]. A CFAR detector is presented in Refs. [35, 36]. Let \( \tilde{y}_k = a_k \tilde{s} + n_k \) denote the \( k \)th received signal including \( N \) samples, where \( a_k \) is a channel amplitude, \( \tilde{s} \) a preamble signal, such that \( || \tilde{s} || = 1 \) (|||| denotes the Euclidean norm), and \( n_k \) a complex white Gaussian noise with variance \( \sigma^2 \). In addition, let \( r(n) \) be an output of the MF (a test variable). \( a_k = 0 \) if the signal is not present. The detector is presented as:

\[
|r(n)|^2 > \gamma || \tilde{s} ||^2 || \tilde{y}_k ||^2,
\]
where \( \gamma \) is a parameter depending on the desired false alarm rate. The average signal power in the right-hand side makes the detector a CFAR detector. Basically, it is an estimator of the thermal noise level, but it also makes the detector insensitive to interference. Note that if an IC unit is used, the average signal power must be measured after the IC unit (i.e., after mitigation), such that mitigation may remove the interference that, otherwise, could deny detection. In other words, the mean signal power would be too high.

Another concern is the effect of the window to the threshold because it affects the signal power, which is especially so if the input signal is windowed and the reference (filter) is not. Let \( \mathbf{w}_u \) and \( \mathbf{w}_r \) denote the window vectors used for the signal and filter analysis. One then has to use the power difference of the windows as a normalizing factor. In addition, overlapping means that the response is computed and replicated for \( M/R \) times, which is also considered. Consequently, the signal power should be modified as follows:

\[
\| S \|^2 = \frac{M}{R} \| \mathbf{w}_u \|^2 \| \mathbf{w}_r \|^2
\]

Ref. [35,36] showed that the false alarm probability \( P_{FA} \) (i.e., the probability that the threshold is exceeded even though the signal is not present) can be approximated as follows:

\[
P_{FA} = e^{-gN}
\]

from which \( g \) can easily be obtained as \( g = \frac{1}{g} \ln(P_{FA}) \). Another false alarm probability that is of interest herein is the probability that the maximum exceeds the threshold [35,36]:

\[
P_{FA,M} = 1 - (1 - P_{FA})^N
\]

The probability of detection \( P_d \) (i.e., the probability that the test cell exceeds the threshold when the actual synchro position is investigated) can be approximated [35,36] as follows:

\[
P_d = Q_b \left( \sqrt{2\mu} \cdot \sqrt{2\gamma(N + \mu)} \right)
\]

where \( \mu = |a_k|/\sigma^2 \) is the SNR of the preamble signal, and \( Q_b(a, b) \) is the generalized Marcum's Q-function [8].

Another interesting probability herein is the probability \( P_m \) that the maximum occurs at the actual synchro position. The approximation in Refs. [35,36] is not very accurate. Therefore, another attempt that will result to a closer approximation is provided. The analysis tool in Refs. [35,36] takes the distribution of \( r(n) \) as it is, and assumes that \( \| Y_\alpha \|^2 \) converges to its average. This simplifies the analysis because only one random variable must be considered. However, the method still has its roots on probability and statistics [37]. Now, at the synchro position \( r(n) \) is a complex Gaussian variable with mean \( a_k \) and variance \( \sigma^2 \). Thus, \( \| r(n) \|^2 \) has a non-central chi-square distribution. Assuming the insignificant sidetones on the autocorrelation function of the preamble signal, the non-synchro positions are zero mean Gaussian variables with variance \( \sigma^2 \). The probability of interest now becomes:

\[
P_m = P(\| r(\text{synchro}) \|^2 > \| r(\text{non-synchro}) \|^2, \forall i).
\]

A direct application of the analysis principle yields to the result presented in Refs. [35,36]. However, this probability is equivalent to the probability that the decision variable at the synchro position is larger than that of the largest non-synchro position. It is well known that 98% of the Gaussian variables are within 2.33 standard deviations from the mean. Thus, the novel approximation is presented as follows:

\[
P_m = Q_b \left( \sqrt{2\mu} \cdot \sqrt{2(2.33)^2} \right)
\]

The confidence probability for very long (large \( N \)) preamble signals may be higher (e.g., 99.5%) because it is natural that then, on average, the largest test variable at the non-synchro positions may be larger than in short signals. See Ref. [38] for another solution to this problem.

One more interesting probability is the probability that the maximum exceeds the threshold, independent of the fact of whether it is the synchro position or not. This is presented as [36]:

\[
P_{D_M} = 1 - (1 - P_{FA})^{N-1}(1 - P_d)
\]

Finally, the probability that the maximum is at the synchro position and exceeds the threshold is provided as \( P_m = P_mP_{D_M} \).

The earlier results are derived in the additive White Gaussian Noise (AWGN) case. The situation is different in fading channels. In the Rayleigh fading channels at the synchro position variable, \( r(n) \) follows a zero mean complex Gaussian distribution with variance \( \sigma^2 + \sigma^2 \), where \( \mu = E(|a_k|^2)/\sigma^2 = \sigma^2/\sigma^2 \) is the average SNR.

\[
P(-|r(n)|^2) = \frac{e^{-|r(n)|^2}}{(\sigma^2/\sigma^2)}
\]

If the analysis tool in Refs. [35,36] is adopted, it follows that:

\[
P_D = e^{-\gamma(N/\mu + 1)}
\]

whereas the study's approach yields:

\[
P_m = Q_b \left( \frac{2\mu}{\mu + 1} \cdot \sqrt{2(2.33)^2} \right)
\]

which reduces to that (7) in the AWGN channel (as it should) if the random element is weak because when \( \sigma^2 = 0 \), then \( \kappa = \infty \). Correspondingly,

\[
P_m = Q_b \left( \frac{2\mu}{\mu + 1} \cdot \sqrt{2(2.33)^2} \right)
\]

4.1. More exact analysis

This section provides a more exact analysis of \( P_m \) in the AWGN, Rayleigh, and Rician channels.

4.2. AWGN channel

\( P_m \) can be calculated as follows in an AWGN channel:

\[
P_m^{\text{AWGN}} = Q_b \left( \sqrt{2\mu}, a_N^{-1} \right)
\]

where \( a_N \) is defined as,

\[
a_N = \frac{E\{\max(|r(n)|^2, n = 0, \ldots, N - 1)\}}{E\{|r(0)|^2\}}
\]
\[ P_m = \int_0^\infty Q_0(\sqrt{2\mu}, \alpha_{k-1}) P(\mu) \, d\mu \]  

(17)

The integral must be solved, where \( P(\mu) \) is the distribution of the SNR in a channel.

### 4.3. Rayleigh channel

In a Rayleigh channel,

\[ P(\mu) = \mu \exp\left(-\frac{\mu}{\bar{\mu}}\right) \]  

(18)

where \( \bar{\mu} \) is the average SNR. The generalized Marcum's Q-function is replaced by its integral form to solve the above integral. We obtain the following after some manipulations.

\[ p_{\text{Ray}} = \left(\frac{K\mu}{K\mu + 1}\right)^{1-K} \exp\left(-\frac{K\alpha_{k-1}}{K\mu + 1}\right) \]  

(19)

where \( K \) is the number of PN sequences used for synchronization.

### 4.4. Rice channel

In a Rician channel,

\[ P(\mu) = \frac{\mu}{\bar{\mu}} \exp\left(-\frac{\mu}{\bar{\mu}} + \frac{\mu_0}{\bar{\mu}}\right) I_0\left(\sqrt{\frac{\alpha_{k-1} \mu}{\bar{\mu}}}\right) \]  

(20)

where \( \bar{\mu} \) is the average of the variable part of the SNR, \( \mu_0 \) is a fixed part of the SNR, such that \( \bar{\mu} = \mu_0 + \bar{\mu} \), and \( I_0(\cdot) \) is the zero order modified Bessel function. We solve the needed integral by replacing the generalized Marcum's Q-function with its equivalent integral form and replacing the Bessel function by its Taylor series expansion. This series of integrals result in:

\[ p_{\text{Rice}} = \sum_{n=0}^m \frac{n!}{\mu^{n+1}} I_0\left(\sqrt{\frac{\alpha_{k-1} \mu}{\bar{\mu}}}\right) \]  

(21)

where \( F(\ldots) \) is the hyper geometric function, and \( e_{n+K}(K\alpha_{k-1}) \) is the incomplete exponential function defined as follows:

\[ e_{n+K}(K\alpha_{k-1}) = \sum_{n=0}^{n+K-1} (K\alpha_{k-1})^n \]  

(22)

The solution of Eq. (21) slowly converges. This convergence can be sped up by manipulating (Eq. (21)) into the following form:

\[ p_{\text{Rice}} = 1 - \sum_{n=0}^m \frac{n!}{\mu^{n+1}} I_0\left(\sqrt{\frac{\alpha_{k-1} \mu}{\bar{\mu}}}\right) \]  

(23)

\[ F\left(n + 1, 1, \frac{\mu_0}{2\bar{\mu}(K\bar{\mu} + 1)}\right) \cdot (\exp(K\alpha_{k-1}) - e_{n+K}(K\alpha_{k-1})) \]

### 5. Numerical results

In this section, the proposed windowed frequency domain acquisition technique is simulated, then compared to the analytical results that provide bounds on the acquisition performance. As a reference, the conventional non-windowed, non-overlapped approach achieved the theoretical bounds in the AWGN channel. Herein, it is trusted to a “fact” that if the analysis holds in Rayleigh fading channels, it also holds in AWGN channels. Therefore, only the Rayleigh fading channels are used in the simulations. A 64 chip preamble sequence (i.e., a 63-chip gold code extended by one) is used in all the simulations. The signal is sampled at one sample per chip. The simulation results are averaged over 1000 independent trials. The SNR is expressed per preamble sequence. The desired false alarm rate is quite high at 10\(^{-2}\).

Fig. 2 shows the results in a flat Rayleigh fading channel when \( M = R = 32 \) (i.e., overlapping is not used, and the window is rectangular). The results illustrate that the simulated and theoretical results coincide. In other words, the approximative analysis is proper.

Fig. 3 presents interesting results from the frequency selective Rayleigh fading channel, which has two equal power multipath components with one chip separation. The SNR is defined per path. In practice, the receiver does not know whether the detected signal sample is from the first or second path. Therefore, the probability that either the first or second path exceeds the threshold (\( P_{\text{fl}} \)) and the probability that either the first or second path provides the maximum (\( P_{\text{md}} \)) are reported. The results conclude that the diversity in the multipath channels is very beneficial for the synchronization. This benefit is lost if the second path is weak and the situation becomes close to that in a single path channel. The multipath propagation causes SNR losses to \( P_0 \) because of the nonzero autocorrelation sidelobes, which are inversely proportional to the preamble length. Another observation is that \( P_m \) becomes close to half. This finding is easily understood because half of the time, the second path is stronger than the first path if the paths have an equal power. Although not shown in the figure for clarity reasons, a good explanation for \( P_{\text{fl}} \), where \( i \) is either \( D \) or \( m \), is:

\[ P_{\text{fl}} = 1 - \prod_{k}(1 - P_2(\text{SNR}_k)) \]  

(24)

where the probabilities are expressed as a function of the SNR and \( \text{SNR}_k \) is the SNR of the \( k \)th path.

The last set of simulations is concerned with the effects of windowing and overlapping. The used analysis window is the Kaiser window with parameter 8, which has very low tail values. The window for the reference is rectangular. The overlapping is either none, 50%, or 75% (i.e., \( R = 64, 32, \) or 16), while \( M = N = 64 \). The channel is a flat fading Rayleigh channel. The simulated false alarm rates with the original threshold setting are 0.01 (the desired value as it should), 0.16, and 0.54 without the window and 4.7 \times 10^{-6} and 3.4 \times 10^{-5}, 0.05 with the window (640,000 samples). This finding shows that threshold tuning is needed if a desired false alarm rate is needed with windows and overlapping. The trend seems to be that a nonrectangular window decreases the false alarm rate, whereas overlapping increases it. Consequently, the simulations with the original threshold setting would not be fair with
respect to the false alarm rate. Therefore, a proper threshold (multiplier of the original) is determined by simulations for the overlapped and windowed cases. Fig. 4 shows the results with the equal false alarm rates. This findings illustrate that overlapping does not significantly affect the performance, but windowing does. The overlapping and windowing are even worse (by 2 dB) than just windowing the conventional non-blocked MF (M = N = R = 64). The windowing loss with the conventional MF is 2–3 dB with this window.

The last result is contrary to the expectation that overlapping reduces windowing losses. Therefore, the last simulations also use window for reference to see if it changes the situation. The results in Fig. 5 indicate that adding the reference window reduces the performance (decreases sensitivity), but the overlapping now does not further decrease it. The total loss compared to the theory is 5 dB. In other words, the windowed cases must have a higher false alarm rate if the same sensitivity is required. The mentioned expectation may have resulted from the fact that the overlapping increases sensitivity if the threshold is kept constant. Therefore, the study results might not be in contradiction to the earlier ones.

6. Conclusions

This study provided an insight into the windowed, overlapped, frequency domain block filtering approach by explaining it and showing some of its possible applications in radio communications. This filtering approach may be used as a universal baseband receiver in communication systems. Accordingly, a single baseband architecture was able to receive all kinds of signals. The approach is especially helpful in multi-purpose platforms, which can hereafter be based on a single architecture simplifying the design. Further investigations are needed to see if this would also reduce other aspects in the receivers, such as power consumption or the silicon area.

The proposed approach was particularly applied to signal acquisition with some novel analyses of the acquisition probabilities in fading channels. This application and the provided analysis and simulation results verified the usefulness of the architecture for a wide range of channel conditions. In addition, the simulations showed that windowing reduces sensitivity if a desired false alarm rate is the receiver design goal. Therefore, one must use windows with care in environments where they are really needed.

One future research topic with the proposed filter would be to discuss if a proper synthesis window could be used to reduce the sensitivity losses that the windows produce. Such a finding would improve the usefulness of the filter. One method of finding an answer might be the dual window. Another open question is the automatic detection threshold determination based on a given false alarm rate with overlapping blocks and windows.

References