# Introduction

Interference is a major impediment to robust communications. In this set of functions we address the mitigation of Narrowband interference (NBI), interference with a bandwidth that is substantially more narrow than the signal of interest. Specifically, we are interested here in a signal of interest that is a single carrier wideband (relative to the interference) signal contaminated with interference that has a bandwidth that is at least a factor of 10 smaller. These functions do not directly address spread spectrum, Ultra Wide Band (UWB), or OFDM. Such solutions could be implemented various ways, but we focus on three techniques. Ultimately, we hope to provide 3-4 methods, but currently provide two techniques: a linear time domain approach (described in a companion data sheet) and a non-linear frequency domain approach that is presented here. Two additional techniques are under development that will be mentioned here (one time domain and one frequency domain), but not described in detail.

## Features

* Accepts a general complex signal which contains a single-carrier signal of interest contaminated by a narrowband interferer and noise.
* Does not depend on the modulation scheme of the signal of interest, although this aspect has not yet been tested thoroughly.
* Does not depend on the specific frequency of the interfering signal.
* Does not depend on time synchronization but assumes that the incoming signal is in complex baseband (which requires some level of rough frequency synchronization).
* Assumes that the signal not of interest is significantly more narrow than the signal of interest (tested on tone interference to date).

# Interface Description

The cancellation flow chart is shown in Figure 1. The main processing routine will call two functions in succession. The first function called is calculate\_threshold.m which determines a cancellation threshold to be applied in the frequency domain. The determined threshold is calculated and sent to the primary cancellation function FreqDomainCancel.m which applies the threshold to mitigate the interference and the complex time domain samples are output for further processing.

Main processing routine

Signal samples

threshold

threshold

Signal samples

Main

FreqDomainCancel.m

calculate\_threshold.m

Signal samples

Input samples

Output samples

**Figure 1: Flow chart for FreqDomainCancel.m (primary cancellation function)**

## Generics

The signal model assumed for the signal of interest, is a single-carrier linearly modulated signal with standard pulse shaping. The specifics of the modulation and pulse shaping should not matter, but have not been thoroughly tested. The signal of interest is assumed to be converted to baseband, although perfect frequency synchronization is not assumed. The NBI here is defined as a single tone with relatively high amplitude, and it is additive. Various SIRs have been tested as will be shown below.

y = FreqDomainCancel(x,threshold);

threshold = calculate\_threshold(x);

## Inputs

The signal inputs to the two functions are defined in Table 2.

**Table 1: Inputs for FreqDomainCancel.m and calculate\_threshold.m**

|  |  |  |
| --- | --- | --- |
| Input Name | Type | Description |
| x |  | complex vector of input samples contaminated with narrowband  interference. |
| threshold |  | threshold used for frequency domain cancellation |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

## Outputs

The outputs for the two functions are defined in Table 3.

**Table 2: Outputs**

|  |  |  |
| --- | --- | --- |
| Output Name | Type | Description |
| y |  | complex vector which is the same length as the input vector and has had interference mitigation applied |
| threshold |  | scalar real value that represents the threshold for frequency domain interference cancellation |
|  |  |  |

## File List

Table 5 lists the main files included with the project. Function files be called are also included in Table 6.

**Table 3: Main File List**

|  |  |
| --- | --- |
| File Name | Description |
| FreqDomainCancel.m - | This function performs the frequency domain cancellation. The input signal is converted to the frequency domain and compared to a provided threshold. All values above the threshold are clipped but while keeping the phase consistent. |
| calculate\_threshold.m | This function determines the adaptive threshold for interference mitigation. Specifically, it divides the frequency range into 10 sub bands, with the assumption that NBI is isolated to one sub band or possibly two if it straddles to sub bands. The 8 bands with the lowest energy are used to calculate the threshold since they should represent the signal only. |
| TestNbiFreqDomain.m | This is a script which tests the frequency domain cancellation technique and demonstrates how to call the functions. |

# Functional Description

## FFT Thresholding Technique

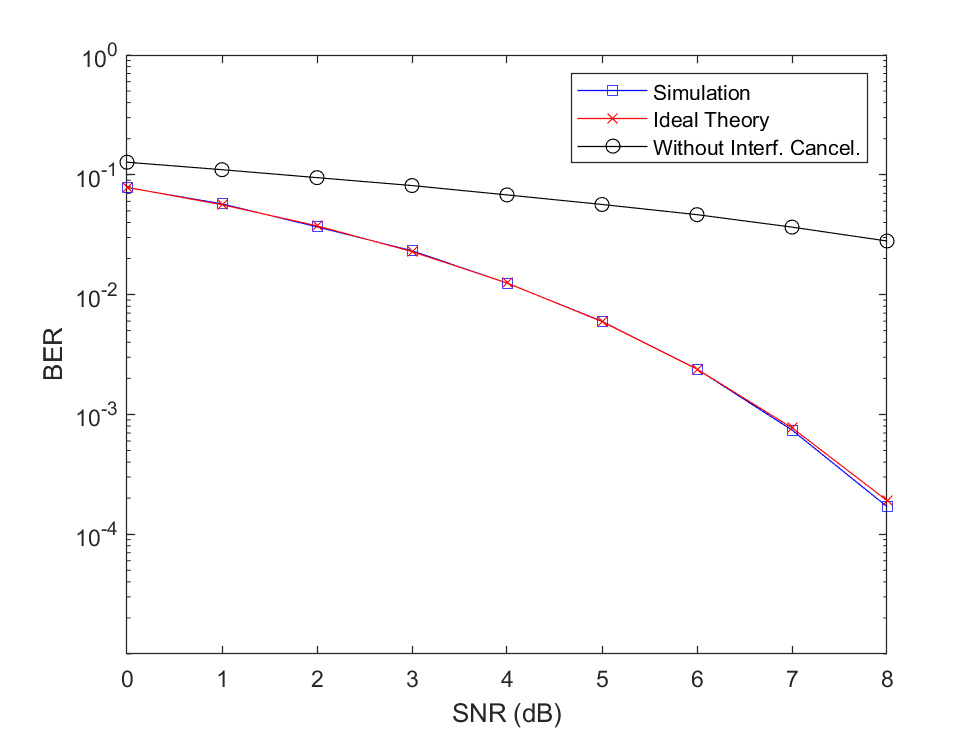
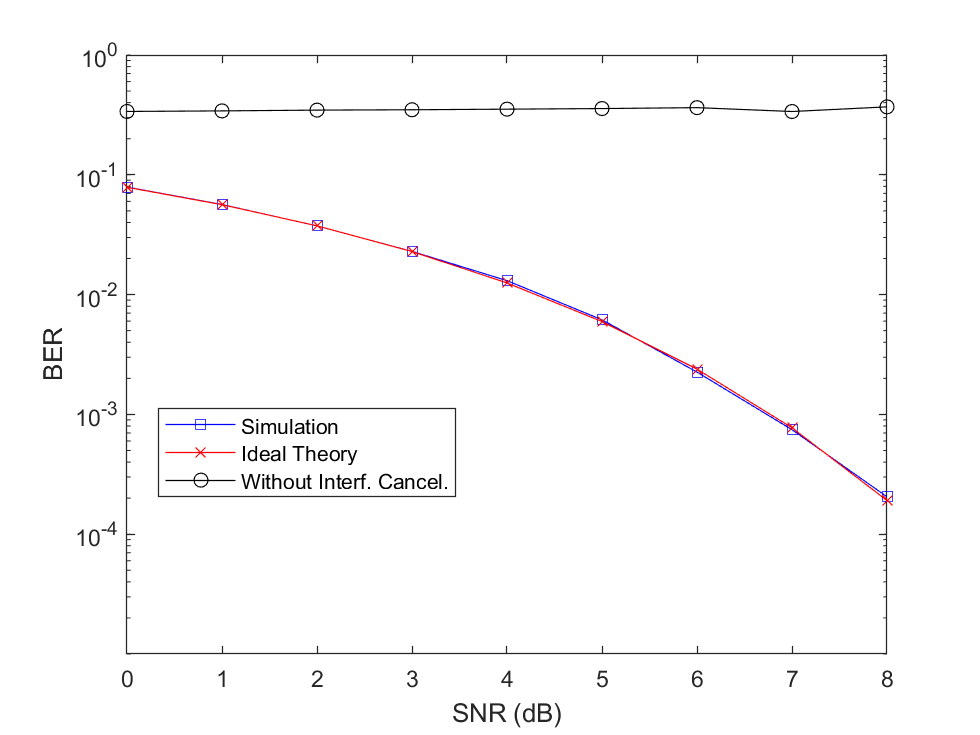
This approach is based on the simple property that NBI has significantly higher magnitude at a specific frequency than the signal of interest. A straightforward way to mitigate this interference is to simply clip the signal in the frequency domain. The threshold is based on the signal of interest. Specifically, the signal of interest is divided into 10 sub-bands with the assumption that the interference contaminates no more than two of those sub-bands.

# Modeling and Simulation

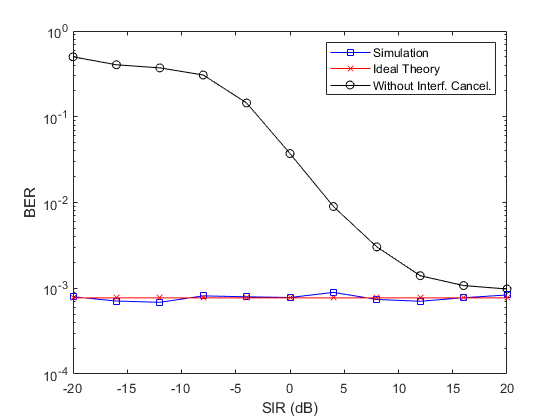
The plots below provide simulated performance of the technique in Matlab. The simulated performance in Figure 2 is plotted versus SNR and compared to (a) AWGN theoretical performance (“Ideal Theory”) and (b) the performance without NBI mitigation (“Without Interf. Cancel”). We can see that the mitigation is essentially ideal and provides a substantial gain. The simulated performance in Figure 3 is plotted versus SIR and again shows performance that is essentially ideal. Note that no assumptions are made regarding signal synchronization for cancellation to be applied. It should also be noted that the performance was tested both before and after down-sampling (i.e., with five samples per symbol and one sample per symbol) and the performance was found to be the same.

Two other issues that could potentially impact the performance of the algorithm: FFT size and frequency offset. FFT size impacts the frequency resolution. The performance with fft size ranging from 0.2k to 200k is shown in Figure 4 for an SNR of 7dB and SIR values of -20dB, 0dB and 10dB. It can be seen that at very low values (200 samples) the performance suffers slightly, but otherwise the performance is unaffected. It should be noted that these plots all assume cancellation occurs *before* downsampling (i.e., with 4 samples per symbol). Additionally, the presence of interference can impact frequency offset estimation and removal. Thus, we desire an interference mitigation technique that is not impacted if the desired signal suffers from a frequency offset. The plot in Figure 5 demonstrates that frequency offset (at values up to 10% of the sample rate) have no impact on performance, as expected. Note that it is assumed in this plot that the offset can be successfully removed *after* interference mitigation occurs.

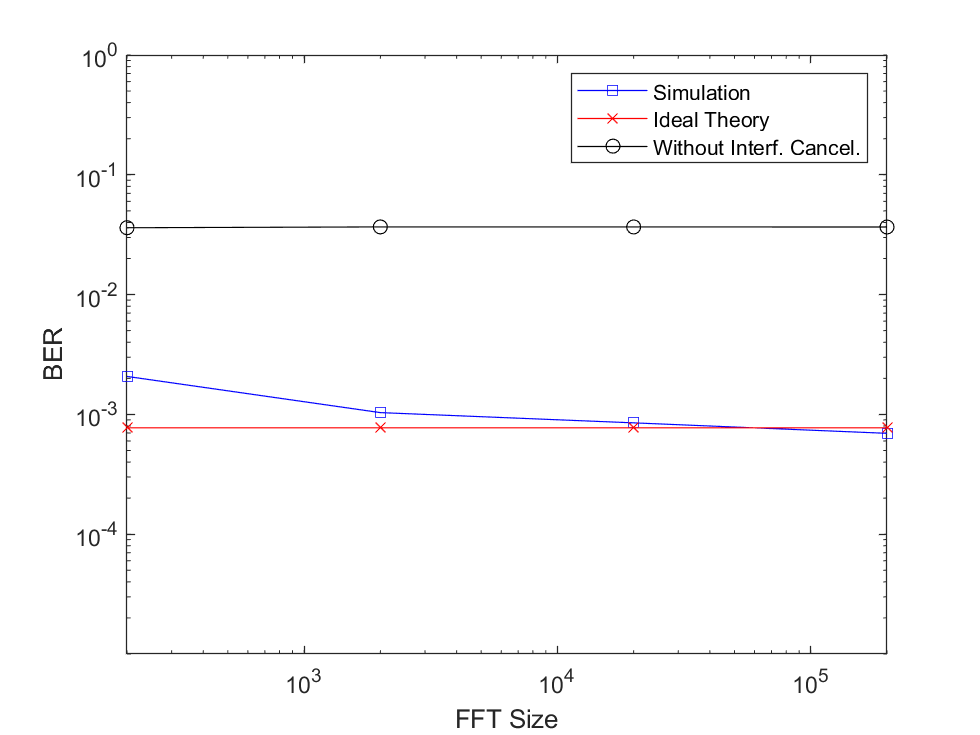
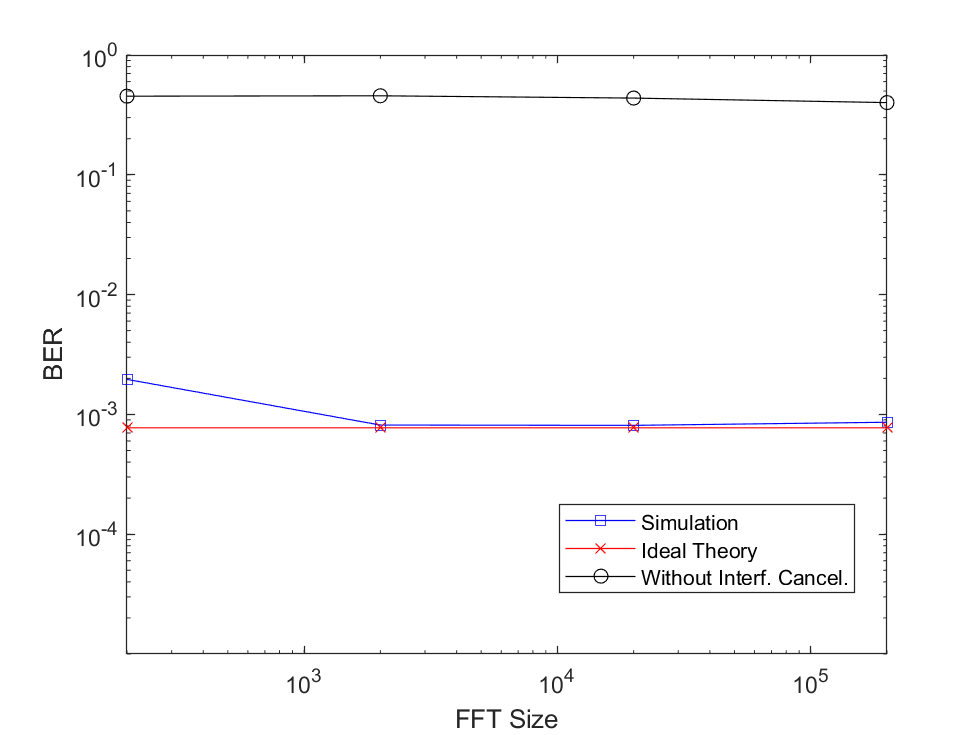
Lastly, the performance versus interference frequency is plotted in Figure 6. As expected the center frequency of the narrowband interference does not impact performance (at least up to 20% of the sampling frequency).

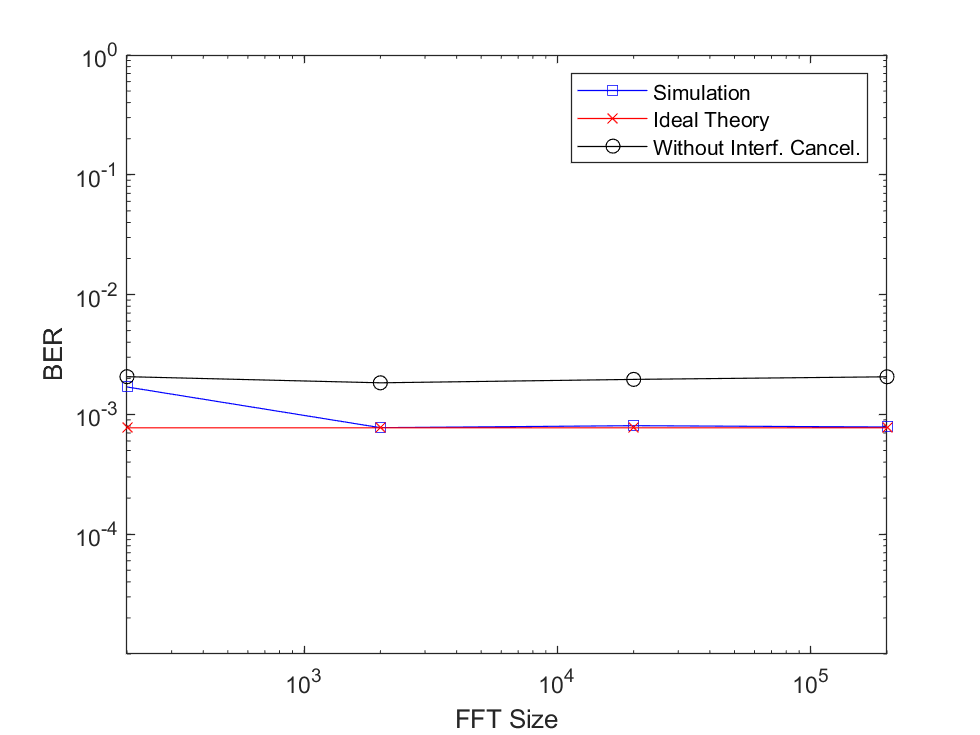
 

**Figure 2 : BER versus SNR for SIR = 0dB (left) and SIR = -10dB (right)**

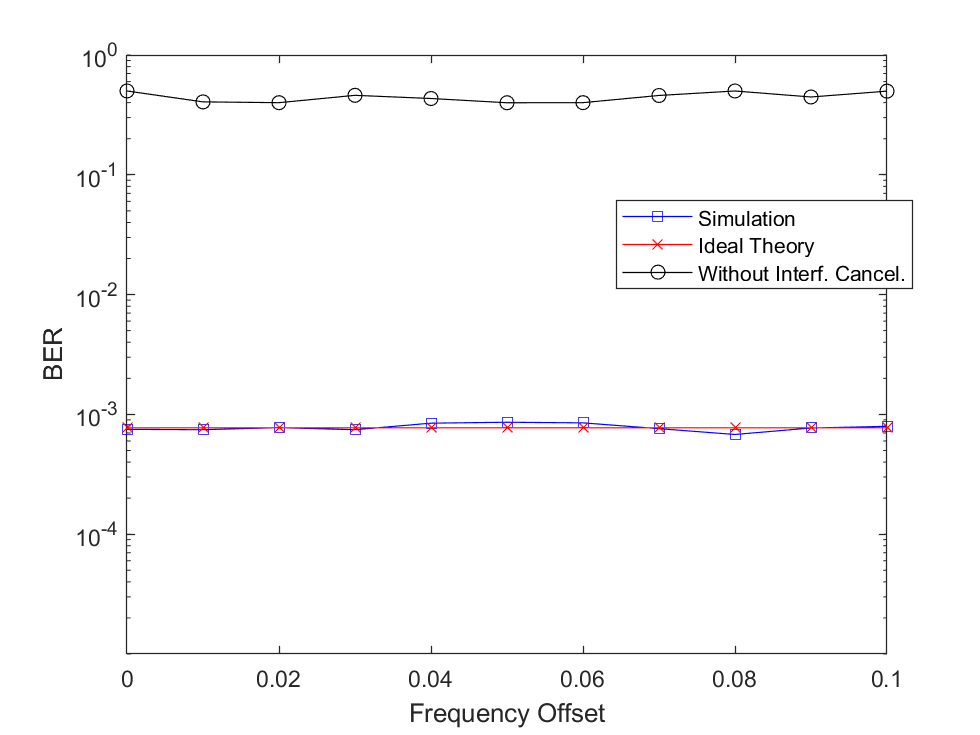


**Figure 3 : Performance vs. SIR for SNR = 7dB**





**Figure 4 : Performance vs. FFT Size for SNR = 7dB and SIR = -20dB (top), 0dB (middle), and 10dB (bottom)**



**Figure 5 : Performance vs. Frequency Offset for SNR = 7dB and SIR = -20dB**

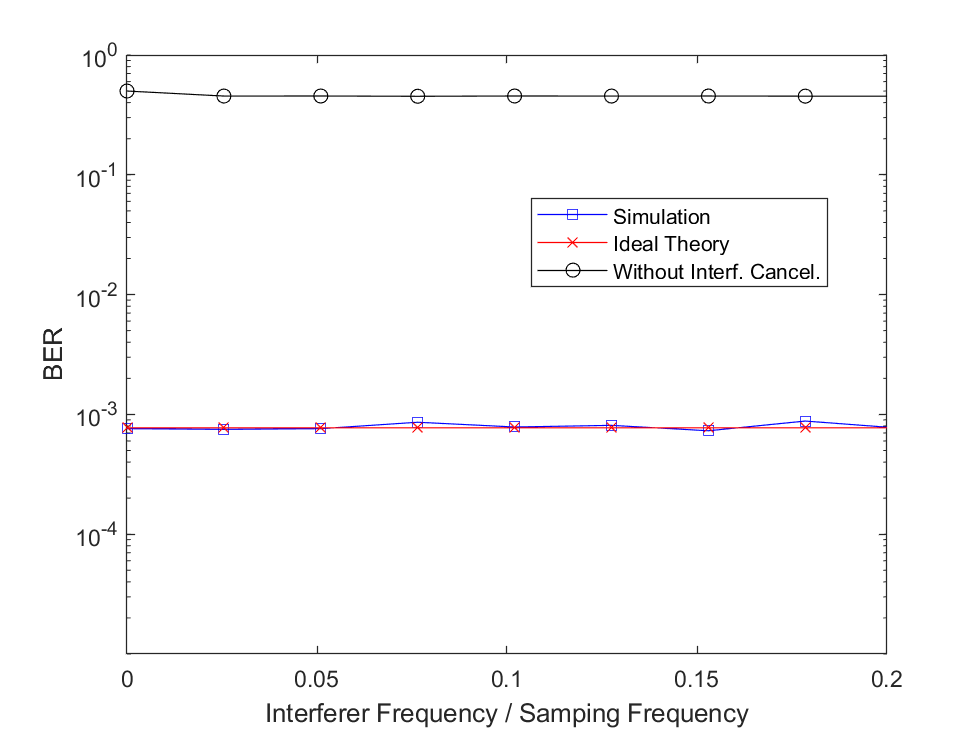


Figure 6 – Performance vs. Interference Frequency for SNR = 7dB and SIR = -20dB