Using Spread Spectrum with Narrow Band Channels

George Elmasry

The problem:

Spread spectrum is beneficial for many of Raytheon's communications channels, as it reduces detectability, mitigates various forms of fading, and enhances resilience to external and self-interference. Please see reference [1] for how wireless radio communications use a combination of spread spectrum and frequency hopping to achieve resilience against eavesdropping and combat jamming. However, engineers avoid using spread spectrum for narrowband channels because spreading a single modulated symbol over a wideband spectrum is not supported by such channels.

RTX offers current and future products that use narrowband channels (e.g., for underwater sensors). These products can benefit from the targeted mathematical solution.

In this paper, we will use the notations "*Bob*" and "*Alice*" to illustrate how Raytheon's narrow-band communication channel "*Bob*" faces the challenge of avoiding a malicious or benign interferer or eavesdropper "*Alice*".

Review of relevant definitions

Figure 1 illustrates how a typical RTX wideband channel employs a combination of frequency hopping and spread spectrum. The channel divides time into repeated time epochs and each epoch into time slices. The channel also divides the wideband frequency into smaller frequency sub-bands (f_1 through f_8 in Figure 1). The transmitter and the receivers are time-synchronized to a pseudo-random frequency-hopping pattern represented by the gray squares in Figure 1. Within each frequency sub-band, the channel spreads the modulated symbol to utilize the entire sub-band frequency range.

Mixing frequency hopping and spread spectrum offers resilience against eavesdroppers and jammers. If Bob is the one using this technique, Alice finds it hard to synchronize to the frequency sub-band utilized by Bob at any moment in time. Even if Alice succeeded in synchronizing to a frequency sub-band at a given moment in time, spread spectrum would make the signal look to Alice like background noise. Spread spectrum spreads the signal over the frequency band, causing its power spectral density at any time to be very low, so that it can be confused with background noise, as detailed below.

The mix of spread spectrum with frequency hopping requires a wideband frequency. Not all the RTX applications have this wideband frequency. If the channel has a very narrow band of frequency (i.e., in the order of a few 10s of Hertz), there is no room for using this technique.

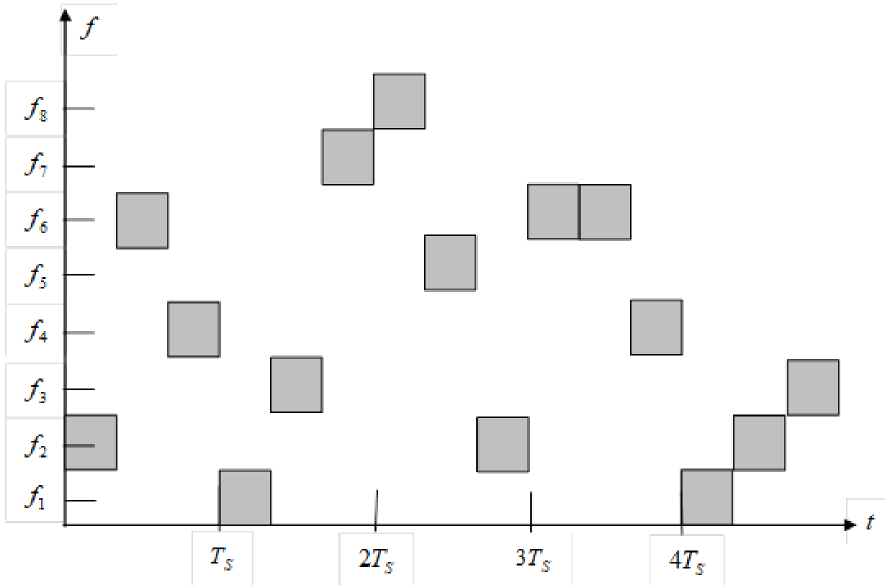

Figure 1:

Figure 2 illustrates how spread spectrum can make the signal that Bob sends seem like background noise to Alice. In Figure 2, the center of the sub-band frequency mentioned above is f_c. The sub-band has a frequency range of $2R_c$. If a modulated symbol is sent without spread spectrum, its power spectral harmonics would peak at the center frequency f_c at a high-power spectral density of $\frac{A^2T_b}{4}$. The existence of harmonics is a telltale of a modulated symbol. On the other hand, if the modulated symbol is sent using the proper spread spectrum, the peak power spectral density at the center frequency becomes much less at $\frac{A^2T_c}{4}$. Notice that increasing the spreading factor means increasing the ratio of $\frac{T_c}{T_b}$. Choosing the proper spreading factor lines the main (center) harmonic of the spread signal within the sub-band frequency range, such that spreading makes the signal as close to low-power noise as possible.

When creating spread-spectrum codes, engineers strive to create orthogonal codes. Making the spread spectrum orthogonal minimizes self-interference. If Bob sends two signals at the same time using the same frequency sub-band, the two signals would not interfere with each other because the spreading codes used are orthogonal to each other. Each seems like background noise to the other. The Hadamard matrix is a form of maximizing the number of orthogonal spreading codes for a given spreading factor.

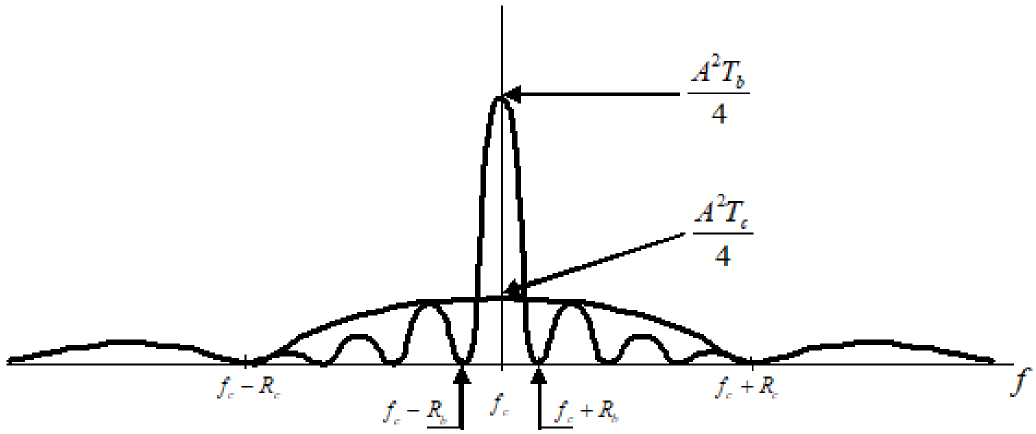

Figure 2: Spread v non-spread signal in the frequency domain

A narrowband is unsuitable for frequency-hopping with spread spectrum. For a channel with a bandwidth in the range of 10s of Hertz, the spreading factor we can use is very small and can be meaningless. There is no room for frequency hopping. Alternatively, we can increase the number of orthogonal spreading codes and be able to hop (cycle) through spreading codes for evasion.

<u>The challenge:</u>

Find a way to use spread spectrum with low-bandwidth communications channels so that these channels can leverage the advantages of spread spectrum, including a form of hopping between spread codes. There exist solutions that can generate a single polynomial or a limited number of polynomials [2], given the channel bandwidth and the number of bits per symbol, for limited use. However, these polynomials are known mathematically and can result in the following limitations:

1- "Alice" can discover "Bob's" transmitted signal as the generating polynomial is known[1].
2- Interference can occur when different transmitters use the same spreading code and their RF footprints overlap. This is self-interference. These polynomials generate a limited number of spread codes for "Bob's" applications. It also does not offer enough entries to rely on hopping between a large pool of spreading codes to evade "Alice".

[1] Polynomials such as Kasami's are known in the literature. An eavesdropper can use correlation techniques with all known polynomials, which are limited, and decipher the signal.

The solution

We are seeking a matrix-based, not a polynomial-based, solution. Our goal is to maximize spreading regardless of the channel bandwidth limitations. We want to generate a large matrix, e.g., a Hadamard matrix. This matrix has a large set of orthogonal spreading codes for a long spreading factor. Although this matrix may not be suitable for the narrow-band channel, we know that within this large matrix, we can find subsets of orthogonal spreading codes that can survive the channel's limitations. Note that we are not relating the transmitted chip rate per second to the size of the matrix. We have the following considerations:

1- The matrix can map to a rate per 2 seconds, a rate per 4 seconds, and so on. Some of Bob's applications can transmit a signal every n seconds where $n > 1$.
2- We have room for correcting lossy compression. The large matrix allows the receiver's cross-correlation to mitigate some chip loss due to channel effects.

This solution will:

1- Enable "Bob" to hop between different spreading codes to evade "Alice".
2- Maximize the channel resilience to varied forms of fading and interference.
3- Allow the RF footprints of different transmitters to overlap, which is the case for many of Bob's applications.

In other words, we aim to use a large spreading factor, way beyond the channel capacity of bits per second, to generate a large matrix. From this matrix, we want to find the maximum number of spreading codes that can survive the channel impact for given channel parameters.

The experimental solution:

We used MATLAB to generate a large Hadamard matrix. With this matrix, we spread the signal beyond the channel's per-second bitrate limitations. We experimented with the entries of this matrix. We were able to:

1- Identify a subset of spreading codes that can't survive the channel even when the channel does not introduce noise (noiseless channel). This subset is discarded. This subset can't survive the lossy compression introduced by the limited bandwidth channel.
2- Assign to the rest of the entries in the matrix a rank based on each entry's ability to survive the channel's bandwidth limitation and a given signal-to-noise ratio (SNR).

This means that if we know the channel's SNR, the transmitter and receiver can switch (hop) between all the orthogonal spreading codes that can survive the channel's SNR.

Further definition of the problem

The mathematical solution we seek includes considerations such as:

1- **Lossy compression**. When a highly spread signal is transmitted over a limited-bandwidth channel, the spreading code goes through lossy compression. Some spreading chips are altered/erased at the receiver. However, because we started with a large matrix, we have long orthogonal spreading codes with maximum separation. This will allow the receiver to estimate the correct transmitted spreading sequence for certain entries of the large matrix, even with the impact of lossy compression.

2- **Harmonics.** The spread signal can be expressed in the transform domain as a polynomial with power levels (coefficients) at the transform domain harmonics. The spreading codes identified as usable by MATLAB simulation are those with coefficients concentrated at the lower harmonics. Varying power spectral densities per harmonics enables ranking a spreading code based on the surviving SNR.

The ideal mathematical solution

Just as Hadamard matrices [3] are known and can be found in MATLAB for given parameters, the solution should be implementable in MATLAB, where we define:

1- The Hadamard matrix with the spreading factor we desire. We want to maximize spreading, to increase the pool of usable spread codes. Recall that we can work with the rate per n seconds where $n > 1$.

2- The channel's bandwidth

3- A given SNR range. SNR is dynamic, making the available pool of spreading codes also dynamic.

4- Other parameters, such as fading.

That solution will identify the suitable subset from the large Hadamard matrix and how the available pool of spreading code varies with SNR.

The tradeoffs

Certainly, one can't keep increasing the dimension of the Hadamard matrix indefinitely. There should be a saturation point where the gain obtained from increasing the size of the Hadamard matrix becomes negligible. The DSP processing power in the product should limit spreading. We want the solution to show how the gain from increasing the spreading matrix size diminishes. This will help us select the proper hardware for the product.

Analogy

Claud Shannon, the father of information theory, proved that there is no point in joining source and channel coding or modulation and coding [4], given that an optimum solution is used at each stage. However, practical solutions proved otherwise. The ideal optimization at each stage is achievable by joining the stages. There are gains to be obtained from joint source and channel coding and joint modulation and coding. This is an analogy to what we are trying to do. We do not want the channel bandwidth limitation to prevent us from using the much-needed spread spectrum in Bob's products. What we seek is a joint solution. We join:

1- The message's time length. The message rate can be per n seconds where $n > 1$.
2- The channel bandwidth
3- The lossy compression caused by the channel, which alters/erases some chips.
3- The channel's SNR
4- Other channel parameters, such as fading

This is merely an analogy to avoid bringing up the known argument that narrow-band channels can't use spreading. Our MATLAB experiments showed that we can use a large pool of spreading codes and vary them as the channel's SNR varies.

Preliminary Math Analysis

The analysis below is for illustration. We did not derive a complete mathematical solution. This is to bring the reader to some thoughts that were considered.

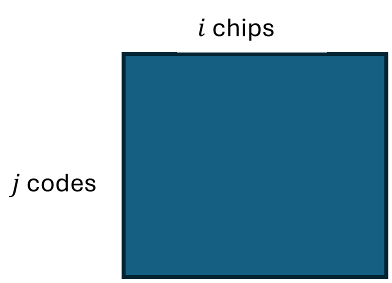

Figure 3: Hadamard Matrix of size N*N

For a given Hadamard Matrix of size $(N * N)$, There are N spreading codes. Each code j, where $j = (1, 2, \ldots N)$ can be expressed in the time domain as:

$$f(it) = \sum_{i=1}^{N} \pm 1(i\frac{\tau}{2}) \quad (1)$$

Where τ is the spreading code pulse width, and $i\frac{\tau}{2}$ is the middle of the i^{th} pulse. The energy of each pulse is normalized to 1 and can be positive or negative. The pulse being positive or negative is associated with the energy at the pulse's center.

Expressing Equation (1) in a transformation domain can be:

$$F(j\omega) = \sum_{k=1}^{\infty} P_k . e^{-2\pi k\omega} \tag{2}$$

Equation 2 is a generic representation of the transform domain polynomial in terms of harmonics, representing the spread signal in Equation (1). That is, this expression is for harmonics with a coefficient at each harmonic, regardless of the transformation methodology used. We avoid using the Fourier Transform expression because the solution may use other transformation domains.

In Equation (2), $\omega = \frac{1}{\tau}$, and P_k is the power spectral density of the k^{th} harmonic (coefficient). Note that the coefficients can be normalized such that the transform domain is a power spectral density that adds up to 1. That is, $\sum_{k=1}^{\infty} P_k \, e^{-2\pi k\omega} = 1$

Equation (2) states that there can be an infinite number of harmonics that represent the j^{th} code. These harmonics are notated as $k\omega$. An infinite number of harmonics are theoretically needed to express the j^{th} code completely in the transformation domain. Practically, some of these harmonics have zero or near-zero coefficients in some spreading codes.

To select a subset of n codes, where $n < N$, that suits our channel, we need to impose conditions on the polynomial expressed in Equation (2). The goal here is to select the subset of n codes that most suit the channel. The goal of the imposed condition can be summarized in the theory below.

Note that the channel's characteristics will ultimately determine the appropriate subset. The solution is not for one channel; rather, it states: Give me a channel's parameters, and I will give you the subset of codes. The channel parameters are dynamic, making the subset of spreading codes dynamic. Unlike the Gold Code obtained from the Kasami polynomial, which is a single set with maximum orthogonality, the proposed approach creates varied subsets, where Bob can switch from one subset to another when the channel's condition changes.

Theory:

- In the Hadamard matrix of size $(N * N)$, all spreading codes are defined as a set H. There exists a subset of orthogonal codes (Subset $A \in H$) that can't survive the channel even when it is noiseless. Subset A is discarded.

- There exists a subset of orthogonal codes (subset $B \in H$) that concentrates power spectral density in low harmonics to overcome the narrow-band channel boundaries.
- The entries in the subset B can also survive the lossy compression of the channel.
- A dynamic subset $C \in B$ can be identified given the varying channel conditions.

The question now is: What decides the subsets A, B and C? What conditions should we impose on the polynomial in Equation (2) to create these subsets?

If we assume that the target BW is β, around a pilot frequency f_c. The conditions we need to apply to Equation 2 can be expressed as achieving Equation 3:

$$P = \sum_{k=f_c}^{f_c+\frac{\beta}{2}} P_k . e^{-2\pi k\omega} + \sum_{k=f_c-\frac{\beta}{2}}^{f_c} P_k . e^{-2\pi k\omega} \sim 1 \tag{3}$$

Equation (3) pertains to double sideband and states that the conditions we impose on Equation (2) will ensure a concentration of the power spectral density of the used spreading code within the narrow band.

Notice that for the given narrow-band channel, the most critical criterion is the result of the cross-correlation of the received chip sequence with all the entries of the Hadamard table. This is the lossy compression aspect of the solution that must be considered.

Let us assume that the chip code selected for transmission is Tc_j, and the received chip code is Rc_j. For a spreading code to fit the channel, Equation (4) below must be met

$$Rc_j \otimes Tc_j > Rc_j \otimes Xc_m \tag{4}$$

Equation (4) states that at the receiver, the received chip sequence yields the highest correlation with the corresponding transmitted sequence, and correlation with all other entries in the Hadamard matrix results in a lower correlation outcome.

That is, we want to use the receiver's correlations to maximize the probability of overcoming the channel's lossy compression.

It is important to note that regardless of the chip rate used, we must ensure that the bit rate transmitted over the channel does not violate the channel capacity. Channel capacity is a physical limitation that can't be violated, or else the decoded bit stream at the receiver will be garbled. We are not claiming that we can violate the channel capacity limit. We are maximizing the use of spread spectrum over the narrow-band channel and the ability to hop between spread codes.

References:

[1] G.F. Elmasry, *"Tactical Wireless Communications and Networks, Design Concepts and Challenges,"* first edition, Wiley, October 2012, ISBN: 978-1-1199-5176-6.

[2] https://www.mathworks.com/help/comm/ref/kasamisequencegenerator.html

[3] https://www.mathworks.com/help/matlab/ref/hadamard.html

[4] R. Blahut, "Principles and Practice of Information Theory," Second Edition, Wiley, 1987, ISBN: 0-201-10709-0.