Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

SDR Implementation of a Testbed for Real-Time Interference Detection with Signal Cancellation

CHRISTOS POLITIS[1], SINA MALEKI[1], JUAN CARLOS MERLANO DUNCAN[1], JEVGENIJ KRIVOCHIZA[1], SYMEON CHATZINOTAS[1], BJÖRN OTTERSTEN[1]

[1]University of Luxembourg, SnT, Luxembourg

Corresponding author: Christos Politis (e-mail: christos.politis@ses.com).

This work is supported by Fond National de la Recherche Luxembourg (FNR), under the CORE projects: SATellite SEnsor NeTworks for spectrum monitoring (SATSENT) and on-board PROcessing techniques for high throughput SATellite systems (PROSAT).

ABSTRACT Interference greatly affects the quality of service of wireless and satellite communications, having also a financial impact for the telecommunication operators. Therefore, as the interfering events increase due to the deployment of new services, there is an increasing demand for the detection and mitigation of interference. There are several interference detectors in the literature, evaluated by using extensive simulations. However, this paper goes one step further, designing, implementing and evaluating the performance of the developed interference detection algorithms experimentally using a software defined radio, and particularly the universal software radio peripheral platform. A realistic communication system is implemented, consisting of a transmitter, a channel emulator and a receiver. Based on this system, we implement all the appropriate communications features such as pulse shaping, synchronization and demodulation. The real-time system implementation is validated and evaluated through signal and interference detection. We observe that the interference detection threshold is critical to the functioning of the system. Several existing interference detection techniques fail in practice due to this fact. In this paper, we propose a robust and practically implementable method the selection of threshold. Finally, we present real-time experimental results for the probabilities of false alarm and detection in order to verify the accuracy of our study and reinforce our theoretical analysis.

INDEX TERMS SDR, USRP, interference detection, real-time testbed, satellite

I. INTRODUCTION

INTERFERENCE has been identified as a major threat for wireless and satellite communications with an important financial impact on the operators [1]. There are various steps that can be performed for the proper management of interference, such as interference monitoring; interference detection and isolation; interference classification; interference localization; and interference mitigation [1]. In this paper, we focus on the detection of interference. The common spectrum sensing techniques, in terms of the way that interference can be detected in a single input-single output (SISO) system are matched filter detection [2], energy detection [3]- [6] and cyclostationary detection [7]. Matched filter detection is an optimal detection approach, however it requires a priori information of the interfering signal, e.g., modulation, coding and etc., which is often not available in practice. Furthermore, cyclostationary detection needs the knowledge of the cyclic frequencies of the interfering signal, and increases the complexity, which make it difficult for practical implementation. On the other hand, the energy detector is a blind technique, as it does not require a priori knowledge of the interfering signal and it is the most popular detector due to its simplicity, resulting in low complexity algorithms. The main drawback of the energy detector for the detection of interference is its sensitivity to the noise variance and desired signal power uncertainties [8], which results is difficulty in setting the decision threshold. However, the detection of low values of interference is crucial, for example the satellite industry where the satellite operators have reported that the type of interference with the major contribution in the interfering events is the VSAT (very small aperture terminals which transmit low power signals) interference [9].

To address this concern, in [8] and [10], we have proposed two on-board interference detection algorithms based on the idea of an energy detector with imperfect signal cancellation exploiting the pilot and data symbols, respectively, which obtain significantly better detection performance than the conventional energy detector (CED). While the developed algorithms have been evaluated extensively using realistic simulations, our objective in this paper is to go one step further and design, implement and evaluate the real-time performance of the developed algorithms experimentally using software defined radio (SDR), reinforcing and validating with this way the theoretical analysis. There are works such as [11], [12], which build testbeds using SDR in order to further explore and verify the potential algorithms. An SDR is a communications platform that uses software for fast prototyping of digital communications algorithms, while allowing analog transmissions over a physical medium. Here, we use the National Instrument (NI) universal software radio peripheral (USRP) platforms as the SDR.

The USRPs [13], [14] are inexpensive programmable radio platforms which can be used in a plethora of applications such as, spectrum monitoring, record and playback, communications, cognitive radio, physical layer prototyping and wireless communications teaching and research. The USRP consists of two main features: 1) the hardware; and 2) the software. The used hardware platform is the NI USRP-2954R, which is considerable choice for wireless communication system designers in terms of cost and performance. This USRP consists of a 2x2 multiple input, multiple output (MIMO) RF transceiver with a LabVIEW programmable digital signal processor (DSP) oriented Kintex-7 field programmable gate array (FPGA), proper for high-rate and low-latency applications [15] and its hardware characteristics are summarized in Table 1. Furthermore, the software platform that we use in order to program the USRPs is the LabVIEW Communications System Design Suite 2.0 [17], [18].

In this study, we design and implement a real-time communication system for the detection of interference with USRPs. In this context, the contributions of the paper are three-fold:

- We build a real communication system for the detection of interference, using USRPs as SDR platform, that contains all the appropriate communication features such as pulse shaping, synchronization and demodulation. Then, we apply real-time physical layer signal processing for the derivation of the decision threshold and the detection of interference.
- Furthermore, we observe that the choice of detection threshold is critical and most theoretically developed thresholds fail in practice (due to unrealistic assumptions). In this paper, we propose a robust and practically implementable method for threshold selection.
- In addition, we evaluate and validate the interference detection algorithms experimentally through the real-time visualization of the probabilities of false alarm and detection and also the detection of interference.

Parameter	Value
Frequency range	10 MHz to 6 GHz
Maximum output power (P_{out})	50 mW to 100 mW (17 dBm to 20 dBm)
Maximum input power (P_{in}	-15 dBm
Maximum instantaneous real-time bandwidth	160 MHz
Maximum I/Q sample rate	200 MS/s
Digital-to-analog converter	2 channels, 200 MS/s, 16 bit
Analog-to-digital converter	2 channels, 200 MS/s, 14 bit

TABLE 1. Hardware characteristics of NI-USRP-2954R [16].

The rest of the paper is organized as follows. In Section II, the system model and the considered interference detection algorithms are discussed. Section III presents a real-time communication system, while Section IV goes one step further including the presence of interference in the previous real-time communication system. Experimental results are depicted in Section V. Finally, Section VI concludes the paper.

Notation: Bold-face lower case letters are used to declare vectors. The superscript $(\cdot)^T$ represents the transpose of $(\cdot)$. $|\cdot|$ is the absolute value.

II. BASIC DESCRIPTION

In this section, we describe the signal model and present the considered algorithms of the paper for interference detection.

A. SIGNAL MODEL

We assume a common telecommunication interference scenario, where the receiver, the desired transmitter and the interferer are equipped with one antenna. Then, this interference detection problem can be formulated as the following binary hypothesis test:

$$\mathcal{H}_0: \quad \mathbf{y}{=}h\mathbf{s}{+}\mathbf{w}, \tag{1}$$

$$\mathcal{H}_1: \quad \mathbf{y}{=}h\mathbf{s}{+}\mathbf{w}{+}\mathbf{i}, \tag{2}$$

where $y = [y(1) \cdots y(N)]^T$, h, $s = [s(1) \cdots s(N)]^T$, $w = [w(1) \cdots w(N)]^T$ and $i = [i(1) \cdots i(N)]^T$ denote the total received signal, the scalar flat fading channel, the transmitted signal by the desired terminal, the additive white Gaussian noise (AWGN) and the interfering signal, respectively. The desired transmitted signal s with power P_s is a modulated signal consisted of an amount of N_p number of pilot symbols $\mathbf{s}_p$, interleaved with an N_d number of data streams $\mathbf{s}_d$. Therefore, $N = N_p + N_d$, with N denoting the total number of samples. Furthermore, it holds that, $\mathbf{w} \sim \mathcal{CN}\left(0, \sigma_w^2 \mathbf{I}_N\right)$ and $\mathbf{i} \sim \mathcal{CN}\left(0, \sigma_i^2 \mathbf{I}_N\right)$ with σ_w^2 and σ_i^2 denoting the variance of the AWGN and Gaussian interference, respectively. Similar signal models are frequently used in the literature [19] - [21] for cases without knowledge of the symbols of the signal as is the case for the interfering signal. It is worth mentioning that the most important factor in the design of a detection scheme is the proper selection of the decision threshold, which is derived in maximizing the probability of detection (P_D) for a specific probability of false alarm (P_{FA}) constraint.

B. INTERFERENCE DETECTION ALGORITHMS

1) Conventional energy detector

The CED is used as a benchmark of our work. The CED computes the energy of the received baseband signal, compares it with a properly selected threshold and decides if the interference is present or not. The probability of false alarm and probability of detection, in this case $P_{FA_{ced}}$ and $P_{D_{ced}}$, can be expressed in closed form as:

$$P_{FA_{ced}} = Q_N\left(\sqrt{\rho_{\mathcal{H}_0}}, \sqrt{\frac{2\gamma_{ced}}{\sigma_w^2}}\right), \tag{3}$$

$$P_{D_{ced}} = Q_N\left(\sqrt{\rho_{\mathcal{H}_1}}, \sqrt{\frac{2\gamma_{ced}}{\sigma_i^2+\sigma_w^2}}\right), \tag{4}$$

where the non-centrality parameter is given by $\rho_{\mathcal{H}_0} = \frac{2|h|^2 E_s}{\sigma_w^2}$ under the hypothesis $\mathcal{H}_0$ and $\rho_{\mathcal{H}_1} = \frac{2|h|^2 E_s}{\sigma_w^2+\sigma_i^2}$ for the hypothesis $\mathcal{H}_1$, respectively, and E_s denotes the energy of the desired transmitted signal.

2) Energy detector with imperfect signal cancellation (EDISC) exploiting the pilot symbols

We proposed this algorithm in [8] in order to overcome the difficulties of the CED in the detection of low values of interference. It performs the following steps:

1) Frame synchronization for the extraction of the received data related to the position of the pilot symbols.
2) Channel estimation.
3) Subtraction of the original pilots multiplied by the estimated channel from the extracted signal.
4) Energy detector in the remaining signal.

Then, the probabilities of false alarm and detection, in this case P_{FA_p} and P_{D_p}, can be expressed in closed form as:

$$P_{FA_p} = \frac{\Gamma\left(N_p-1, \frac{\gamma_p}{\sigma_{w_p}^2}\right)}{\Gamma\left(N_p-1\right)}, \tag{5}$$

$$P_{D_p} = \frac{\Gamma\left(N_p-1, \frac{\gamma_p}{\sigma_{w_p}^2+\sigma_{i_p}^2}\right)}{\Gamma\left(N_p-1\right)}. \tag{6}$$

3) Energy detector with imperfect signal cancellation (EDISC) exploiting the data

We proposed this algorithm in [10], in case that the interference is intermittent during the frame or a larger number of pilots supported from the standards is required for the detection of low values of interference. It performs the following steps:

1) Frame synchronization for the extraction of the received data related to the position of the pilot symbols to be used for channel estimation.
2) Demodulation of the received signal.

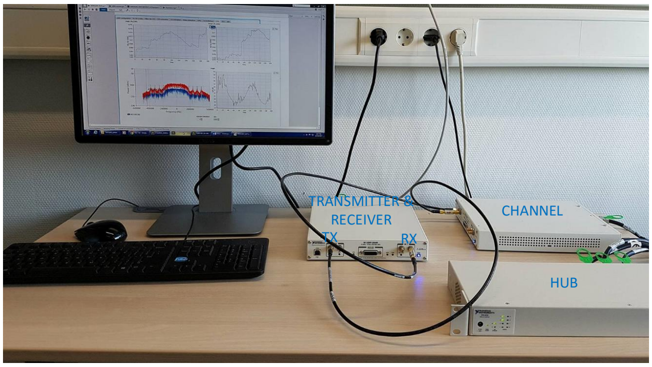


FIGURE 1. Experimental set-up, where the SDR platform used for transmitter, channel emulator and receiver is the NI USRP-2954R.

3) Subtraction of the demodulating signal multiplied by the estimated channel from the total received signal.
4) Energy detector in the remaining signal.

Then, the probability of false alarm for the QPSK scenario, in this case P_{FA_Q} and P_{D_Q}, can be expressed in closed form as:

$$P_{FA_Q} = \sum_{k=0}^{2N} \begin{pmatrix} 2N \\ k \end{pmatrix} P_{k_Q} P_{e_Q}^k \left(1-P_{e_Q}\right)^{2N-k}, \tag{7}$$

where k denotes the number of wrong decoded bits, $P_{e_Q} = Q\left(\sqrt{\frac{\gamma P_s}{\sigma_w^2}}\right)$ is the probability of bit error for QPSK [22] and P_{k_B} is the probability of false alarm for the case that k bits are decoded wrongly, which can be approximated as follows:

$$P_{k_Q} = Q\left(\frac{\varepsilon - \mu_{\mathcal{H}_{0_Q}}}{\sqrt{V_{\mathcal{H}_{0_Q}}}}\right), \tag{8}$$

where $\mu_{\mathcal{H}_{0_Q}}$ and $V_{\mathcal{H}_{0_Q}}$ are the mean and variance of the test statistic, respectively. Similar expression is expressed for the probability of detection.

III. BUILDING A REAL COMMUNICATION SYSTEM WITH USRPS

In this section, we build a demonstrator of a real communication system with USRPs. The demonstrator consists of one transmitter, one channel emulator and one receiver. A picture of this experimental set-up is depicted in Figure 1. Now, we will analyze it with more details starting by building up the transmitter, then the channel emulator and we will conclude with the receiver. Table 2 shows a summary of the parameters of this demonstrator. Here, we should mention that the selected carrier frequencies of the transmitter and receiver are set to 10 MHz in order to decrease the mutual coupling between the transmission and reception links.

A. TRANSMITTER

The full transmitter is shown in the flowchart in Figure 2. It consists of two parts: 1) the components generated by the host with the aid of LabVIEW; and 2) the components which

Parameter	Value
Modulation type	QPSK
Sample rate	1M samples/second
Bandwidth	400KHz
TX to Emulator carrier frequency	10 MHz
Emulator to RX carrier frequency	10 MHz
RX carrier frequency	10 MHz
Over-sampling factor	4
Pulse-shaping filter	root-raised cosine with 0.5 roll-off
Multi-tone preamble	320 samples
Data	500 symbols or 2000 data samples

TABLE 2. Experimental parameters for the transmission and reception in a single input-single output (SISO) system.

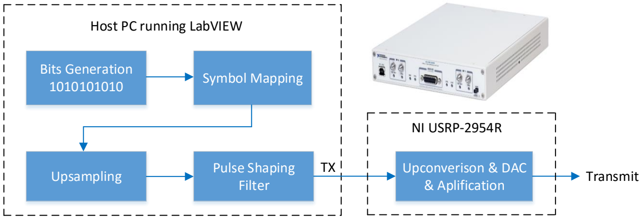


FIGURE 2. Flowchart showing the generation of a digitally modulated waveform and the transmitted signal.

take place in the NI USRP-2954R. Starting on the left of Figure 2, we generate a random sequence of bits. Once we have the bits, we are ready to map them to our modulation symbols, where QPSK is the chosen modulation scheme. Then, we design a pulse-shaping filter for a given length, sample rate, symbol rate and roll-off factor. The pulse-shaping filter is used to combat the intersymbol interference (ISI) [22]. Nyquist has developed a criterion for choosing a filter that is guaranteed to have zero ISI. One such filter is the root-raised cosine filter [22]. The modulated complex baseband waveform after the use of pulse-shaping is depicted in Figure 3. After the host has synthesized the the baseband in-phase (I) and quadrature-phase (Q) signals, they are ready to pass to the transmitted device over a standard PCIe connection. There, the digital up-converter mixes, interpolates and filters the signal, while the digital-to-analog converter (DAC) converts it to analog signal. Then, the signal is upconverted and finally, is amplified and transmitted through the cables to the channel emulator.

Preamble

In Figure 3, we can notice that there are two distinct signals: 1) the signal related to frame synchronization; and 2) the data. One of the most important functions of the receiver is the synchronization. As we will discuss later in this paper, the synchronization issues have two aspects: 1) time; and 2) frequency synchronization. One method to address the time synchronization is to insert a known preamble at the beginning of a transmission. A proposed method is the design of a multi-tone preamble, which offers robustness in frequency-selective fading channels [23], [24], [25]. Therefore, this is the reason that we can see the two different waveforms in Figure 3, one for the multi-tone preamble and another one for the information data. At the receiver, the time synchronization is obtained by finding the instant at which the cross-correlation of the known multi-tone preamble and the received signal reaches in its peak.

B. CHANNEL EMULATOR

The transmitted signal is sent to the channel emulator, which injects to the signal an AWGN with a controlled power and then is sent to the receiver. The channel functionalities are implemented in an FPGA which is integrated to the SDR platform.

The channel emulator implements a hardware complex multiplier for each of the emulated channel chains in order to control the amplitude and phase of the signal. By means of this multiplier, the channel emulator has the possibility to recreate time-varying fading patterns. The AWGN uses a gold sequence generator with the two polynomials of order 63, which are associated to Galois Linear Feedback registers:

$$P1(X) = X^{63} + X^{62} + 1 \quad \text{and} \quad P2(X) = X^{63} + 1, \tag{9}$$

which generate the periodic sequence $Z(k)$, for the index k with a period L. Two offset versions of the sequence $Z(k)$ are used to generate the sequence number $R(k) = [0, 1, 2, 3]$ with

$$R(k) = Z(k) + 2Z(\text{modulo}(k + O, L)), \tag{10}$$

where O is a given integer offset.

The output waveform is generated by applying the discrete Fourier transform (DFT) procedure to the sequence

$$Q(k) = e^{(j\pi/2R(k)+pi/4)}, \tag{11}$$

for sections of the sequence $Q(k)$ given by the selected DFT length. The transformation length is chosen as a function of the number of quantization bits used in the FPGA for the noise generator.

The output pseudo random noise is the concatenation of the outputs of the subsequent DFT operations. The pseudo random noise injected to the signal is also multiplied by a weight coefficient by means of a complex multiplier in order

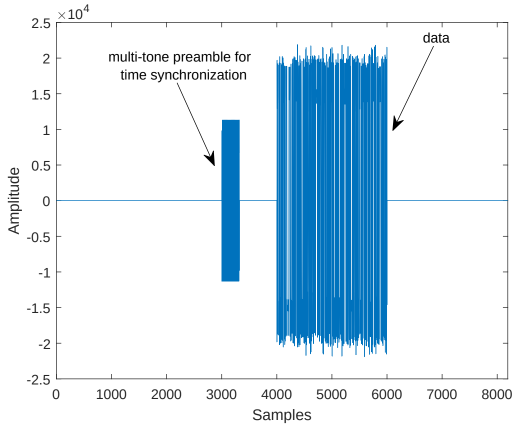


FIGURE 3. Transmitted waveform before passing to the USRP.

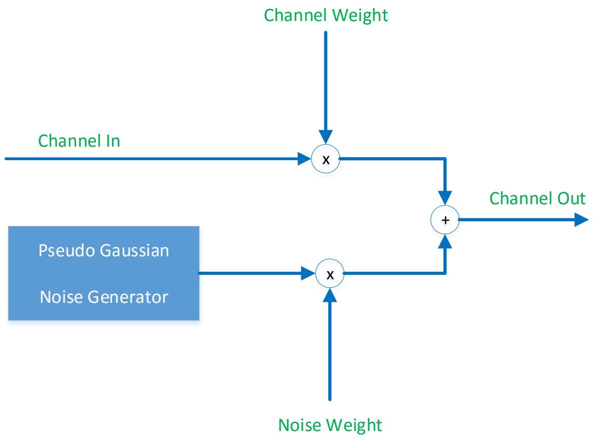


FIGURE 4. Flowchart showing the generation of a digitally modulated waveform and the transmitted signal.

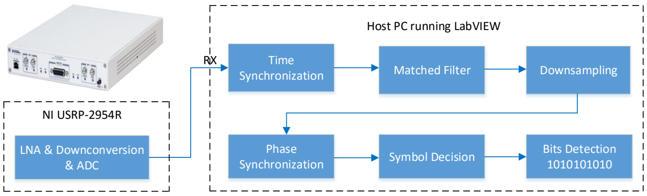


FIGURE 5. Flowchart showing the demodulation process of a digitally modulated waveform.

to control its power. Figure 4 shows how the output signal is generated, having the capability to accurately adjust the SNR.

C. RECEIVER

The functionality of the receiver is shown in Figure 5. It is the reverse process of the transmitter with the additional pieces of synchronization. Again, the receiver consists of two parts: 1) the components which take place in the NI USRP-2945R; and 2) the components generated by the host and the aid of LabVIEW. At the receiver, the incoming signal is amplified by an LNA and downconverted to the baseband in-phase and quadrature phase components. Then, the signal passes through the analog-to-digital converter (ADC) and transported to the host computer. In the host, the first thing the receiver needs to do is the time synchronization in order to know where the waveform begins and where to start the pulse shaping filter. Then, we downsample the signal and finally, we attempt to recover the transmitted waveform and detect the symbols and bits.

Received signal from the USRP

The incoming signal in the RX port of the USRP is amplified, downconverted and analog-to digital converted inside the USRP, where finally, the samples are transported to the host computer for further processing. This received signal at the host is presented in Figures 6 and 7 for two different signal-to-noise ratio (SNR): 1) 30 dB; and 2) 8 dB, respectively. In both cases, we can see that the beginning of the transmitted signal is different compared to that of the received signal in Figure 3. Therefore, the time synchronization is necessary.

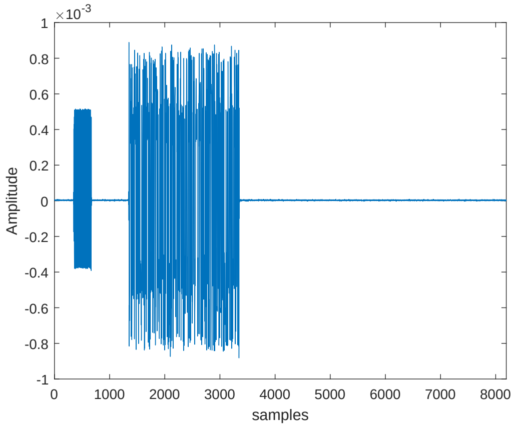


FIGURE 6. Received signal at the host with SNR=30 dB.

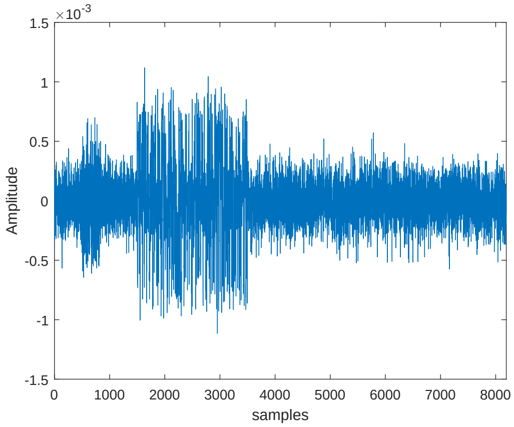


FIGURE 7. Received signal at the host with SNR=8 dB.

Synchronization

Synchronization is a very important part in digital communications [22] as the receiver needs to know where the waveform begins and where to start the matched filter. If the receiver starts receiving somewhere in the middle of a transmission or if we do not know which symbol was in the beginning of a transmission, we understand that the proper recovery of the bits is almost impossible. There two important aspects related to synchronization: 1) time; and 2) frequency synchronization.

To overcome the issue of time synchronization, a signal, known as preamble, can be embedded at the beginning of a transmission [26], [27]. This signal should be different from our data signals, otherwise we will get false matches. Therefore, we should choose a very distinct signal, which has almost no correlation with any of our data symbols and is known and just append that signal to the beginning of every transmission. Then, a matched filter is used at the receiver, where the peak of its output will give a good estimate about the position of our embedded synchronization symbol and

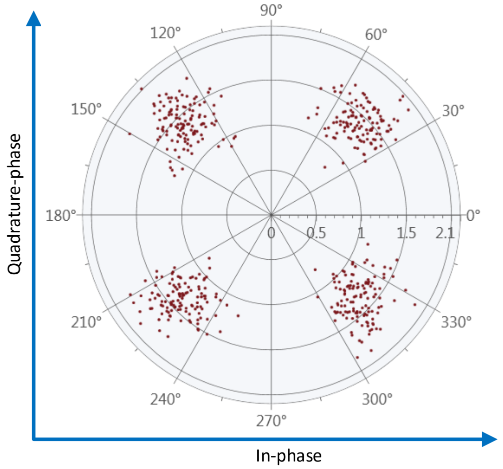


FIGURE 8. QPSK RX constellation for SNR=8dB with time and phase synchronization.

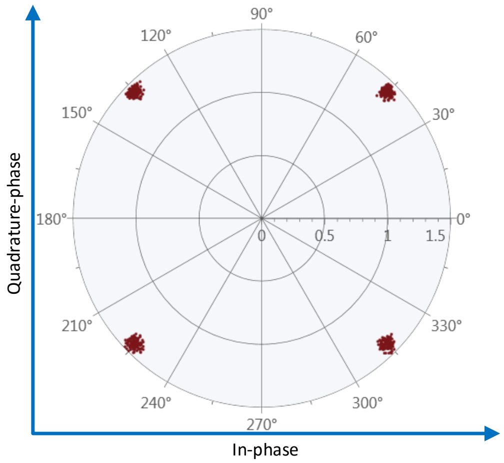


FIGURE 9. QPSK RX constellation for SNR=30dB with time and phase synchronization.

thus the beginning of our transmission [18].

As for the frequency synchronization, it is as important as the time synchronization mentioned previously. Any frequency error will result in a phase rotation on the recovered symbols. For the estimation of the frequency error, we can embed a specific symbol at known places in our transmission. These symbols will have known phases, which can be employed as references to compute the phase error of the recovered symbols. Then, the frequency offset can be estimated by adding the phase error over the duration of the transmission.

Matched filter and downsampling

As discussed earlier, the choice of the root-raised cosine filter at the transmitter satisfies the Nyquist criterion and therefore guarantees ISI-free transmission. However, at the receiver side, the best option is the use of a matched filter, which guarantees optimal error performance at the receiver output, maximizing the SNR [22]. Next we downsample the signal to arrive at what should be the transmitted symbols.

Phase synchronization and symbol decision

Finally, we have come to the point where we can attempt to recover our transmitted waveform. Figures 8 and 9 present the constellation of the time and phase aligned downsampled complex waveform, namely the constellation of our signal after the process of matched filtering and time and phase synchronization for SNR=8 dB and SNR=30 dB, respectively.

IV. BUILDING A REAL COMMUNICATION SYSTEM FOR INTERFERENCE DETECTION USING USRPS

In this section, we build a demonstrator of a real communication system for the detection of interference using USRPs. The demonstrator consists of one transmitter, one interferer, one channel emulator and one receiver. A picture of this experimental set-up is depicted in Figure 10. The NI USRP-2954R has two RF transmitters. Therefore, as shown in Figure 10, the TX1 port is used for the generation of the desired signal, while the TX2 port is used for the generation of the interference. Furthermore, the desired signal is sent to the channel emulator, which injects to the signal AWGN noise with a controlled power. With this controlled artificial noise, we can adjust the desired SNR more efficiently for evaluation purposes. Then, this signal and the interference added in their analog waveforms through a connector and the resulting signal is sent to the RX port of the USRP for further processing.

Most of the pieces in Figure 10 were described in the previous section, so here, we focus on the design of the interference and the implementation of the developed algorithms for its detection. The experimental parameters are same as in Table 2.

A. IMPLEMENTATION FOR THE CALCULATION OF THE PROBABILITY OF FALSE ALARM IN REAL TIME

The goal of this paper is to implement some novel algorithms for the detection of weak interference. The most important factor for the design of these detection algorithms is the proper derivation of the decision threshold, which is indepen-

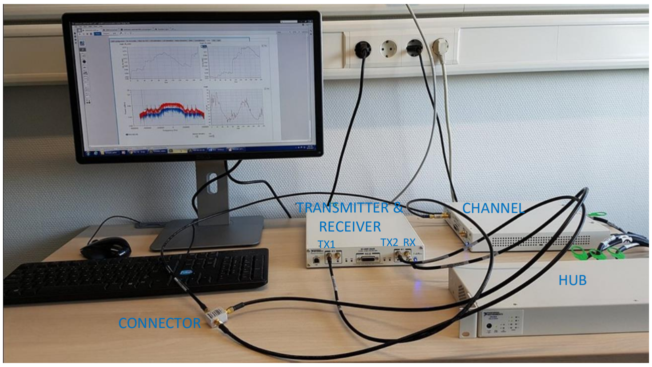


FIGURE 10. Experimental set-up for interference detection, where the SDR platform used for transmitter, interferer, channel emulator and receiver is the NI USRP-2954R.

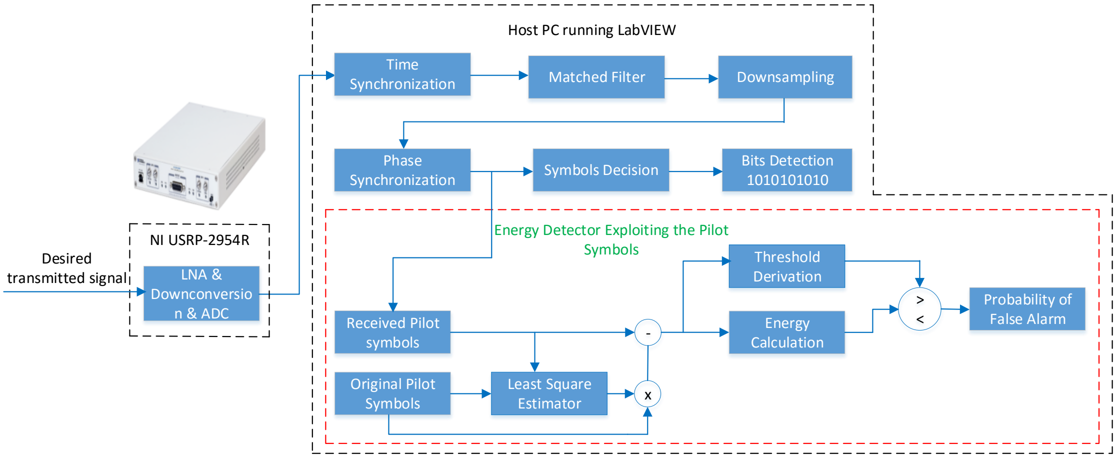


FIGURE 11. Flowchart showing the methodology for the calculation of the probability of false alarm of the EDISC exploiting the pilots.

dent from the distribution of the interfering signal. Therefore, for the calculation of the decision threshold we focus on the hypothesis $\mathcal{H}_0$, where the interference is absent. Hence, in the beginning (until the derivation of the decision threshold), the TX2 port of Figure 10 does not generate interference and then, the adopted experimental set-up is more similar with that in Figure 1.

Now, using this set-up, we implement the methodology for the derivation of the threshold and the probability of false alarm of three algorithms. First, we start with the energy detector with imperfect signal cancellation exploiting the pilot symbols, then we continue with the energy detector with imperfect signal cancellation exploiting the data and we conclude with the conventional energy detector.

1) Probability of false alarm for the EDISC exploiting the pilot symbols

Once we acquire the time and phase aligned downsampled complex waveform, we know the positions of the pilots in this received signal. Therefore, we extract the samples related to the position of the pilot symbols. Then, we estimate the channel using a least square estimator and remove the original pilots from the extracted received signal. Finally, we apply an ED in the remaining signal. This methodology is described in the block diagram of Figure 11.

Regarding the block of the threshold derivation, two methods have been used: 1) derivation of the decision threshold based on the theoretical formula; and 2) derivation of the decision threshold based on a more practical approach.

Implementation for the derivation of the decision threshold based on the theoretical formula

The probability of false alarm of the energy detector with signal cancellation exploiting the pilot symbols is given by (5). The derivation of the threshold γ_p based on (5) requires the inverse incomplete gamma function. However, the implementation of the latter in LabVIEW is very difficult. For this reason, we derive an approximated formula for the P_{FA_p} using the central limit theorem (CLT). Then, the approximated probability of false alarm of the energy detector with signal cancellation exploiting the pilot symbols, in this case $P_{FA_{p_{apr}}}$, is given by

$$P_{FA_{p_{apr}}} = Q\left(\frac{\gamma_p - \left(N_p-1\right)\sigma^2_{w_p}}{\sqrt{\left(N_p-1\right)\sigma^2_{w_p}\sigma^2_{w_p}}}\right), \qquad (12)$$

and the threshold is derived as follows:

$$\gamma_p = Q^{-1}\left(P_{FA_{p_{apr}}}\right)\sqrt{\left(N_p-1\right)\sigma^2_{w_p}\sigma^2_{w_p}} + \left(N_p-1\right)\sigma^2_{w_p}. \qquad (13)$$

Therefore, we have to implement one function in LabVIEW, which calculates the inverse Q function.

Furthermore, from (13), it is obvious that the threshold γ_p depends on the noise variance $\sigma^2_{w_p}$, which is unknown in practice, and hence has to be estimated. As mentioned earlier, after the frame synchronization, the original pilot symbols are removed from the received samples related to the position of the pilot symbols and then, the hypothesis $\mathcal{H}_0$ is written as:

$$\mathcal{H}'_{0_p} : \quad \mathbf{y}'_p\text{=}\mathbf{w}_p\text{-}\varepsilon_{\mathcal{H}_0}\mathbf{s}_p. \qquad (14)$$

For a large number of pilots ($N_p > 100$) the channel estimation is almost accurate, hence the channel estimation error $\varepsilon_{\mathcal{H}_0}$ is negligible and the above hypothesis can be simplified into the following one:

$$\mathcal{H}'_{0_p} : \quad \mathbf{y}'_p\text{=}\mathbf{w}_p. \qquad (15)$$

In this case, the log-likelihood function (LLF) under $\mathcal{H}'_{0_p}$ can be expressed as

$$\ln p\left(\mathbf{y}'_p|\mathcal{H}'_{0_p}, \sigma^2_{w_p}\right) = -\frac{N_p}{2}\ln\left(2\pi\sigma^2_{w_p}\right) - \frac{1}{2\sigma^2_{w_p}}\sum_{n=1}^{N_p}\left|y'_p\left(n\right)\right|^2. \qquad (16)$$

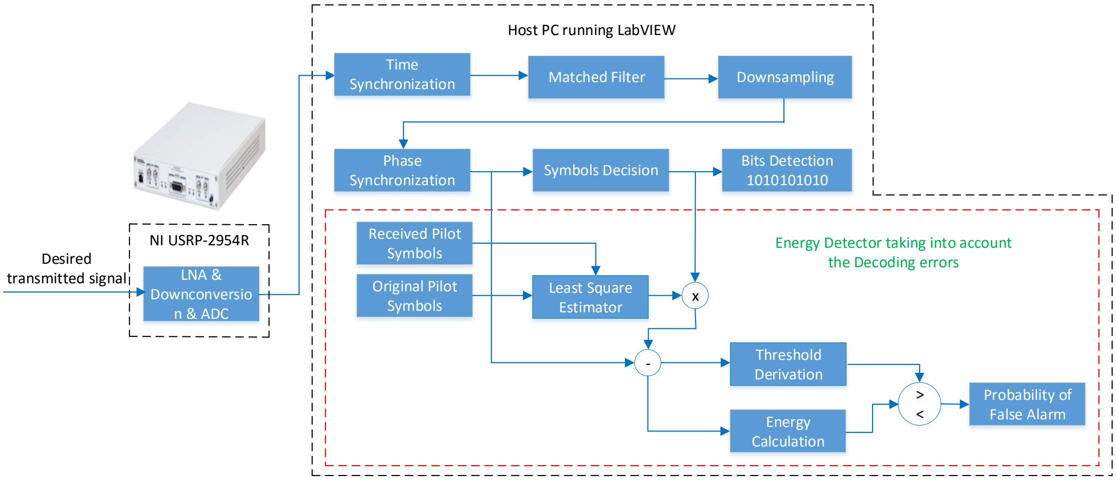


FIGURE 12. Flowchart showing the methodology for the calculation of the probability of false alarm of the EDISC exploiting the data.

The maximum likelihood estimate (MLE) of $\sigma^2_{w_p}$ under $\mathcal{H}'_{0_p}$ minimizes (16) and is given by

$$\hat{\sigma}^2_{w_p} = \frac{1}{N_p} \sum_{n=1}^{N_p} \left| y'_p(n) \right|^2. \tag{17}$$

Therefore, from (17) we can see that the estimation of the noise variance is obtained by taking the summation of N_p squared samples (remaining samples after the signal cancellation) and then dividing by the number N_p. In practice estimation of noise variance is the bottleneck and one solution is to try to find a reference band where we are almost sure there is no interference or signal. In our lab setup in order to achieve a more reliable estimation of the noise variance, we devote a large number of frames N_f. Namely, we estimate the noise variance in each frame and when the devoted period expires, we derive the averaged estimated noise variance. Then, the latter and (13) are utilized for the derivation of the decision threshold. In the next frames, the TX2 port of Figure 10 starts to transmit an interfering signal and then, we apply the EDISC exploiting the pilots and use this derived threshold to detect the presence or absence of interference.

Implementation for the derivation of the decision threshold based on a more practical approach

Here, we derive the decision threshold based on a practical approach. This method can be used in the case that the theoretical expression for the derivation of the decision threshold is unknown.

In this method, same as before, we devote a number of frames N_f only for the derivation of the decision threshold. In each frame, we calculate the energy of the remaining samples after the signal cancellation and save it in a buffer. At the end of the devoted period, we find the maximum and minimum value in the buffer and we apply line search in order to determine the threshold γ_p, which guarantees a given P_{FA_p}.

Probability of false alarm in real time

After the derivation of the decision threshold, we can calculate the probability of false alarm in real time. In each frame where the energy of the remaining signal after the pilots cancellation is higher than the decision threshold, we increase an initialized to zero counter by one and then, we divide the new result by the current number of frame. Therefore, in each frame we update the probability of false alarm and provide a visualization of it in real time.

2) Probability of false alarm for the EDISC exploiting the data

In this algorithm, we estimate the channel by using again the pilot symbols, then we decode the received signal and we remove the decoded signal from the total received signal. Finally, we apply an ED in the remaining signal. This methodology is described in the block diagram of Figure 12. Then, we calculate the probability of false alarm and derive the threshold based on the practical approach that described earlier.

3) Probability of false alarm for the CED

In this algorithm, there is no need for signal cancellation, therefore we determine the threshold immediately after the time synchronization (there is need to know the beginning of the frame). Then, we apply an ED in the total received signal. This methodology is described in the block diagram of Figure 13. In this algorithm, we calculate the probability of false alarm, derive the threshold based on the methodology of the practical approach and implement them in LabVIEW, same as before.

B. GENERATION OF INTERFERENCE AND IMPLEMENTATION FOR THE CALCULATION OF THE PROBABILITY OF DETECTION IN REAL TIME

After the derivation of the decision threshold, the TX2 port of Figure 10 starts to generate interference. In this paper, we assume that the interference follows a Gaussian distribution.

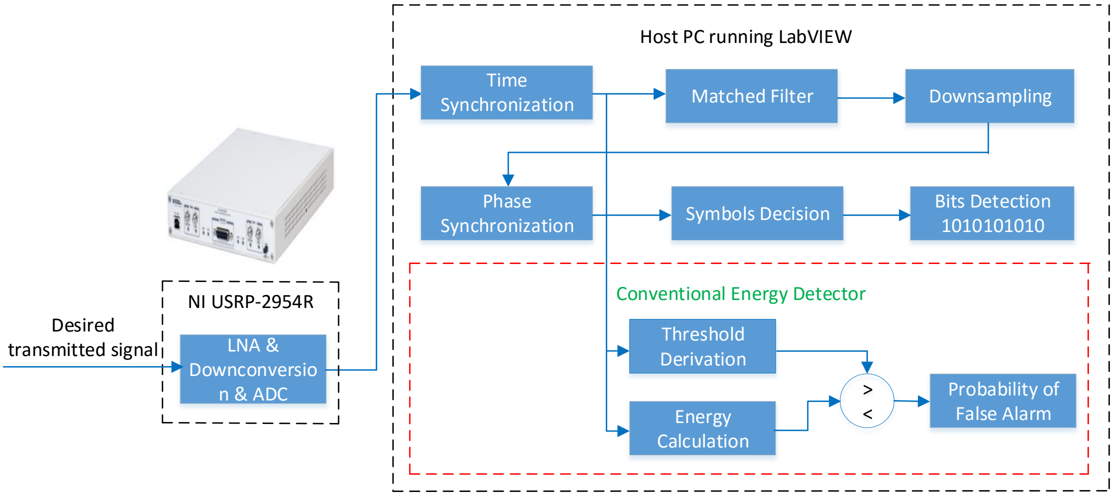


FIGURE 13. Flowchart showing the methodology for the calculation of the probability of false alarm of the CED.

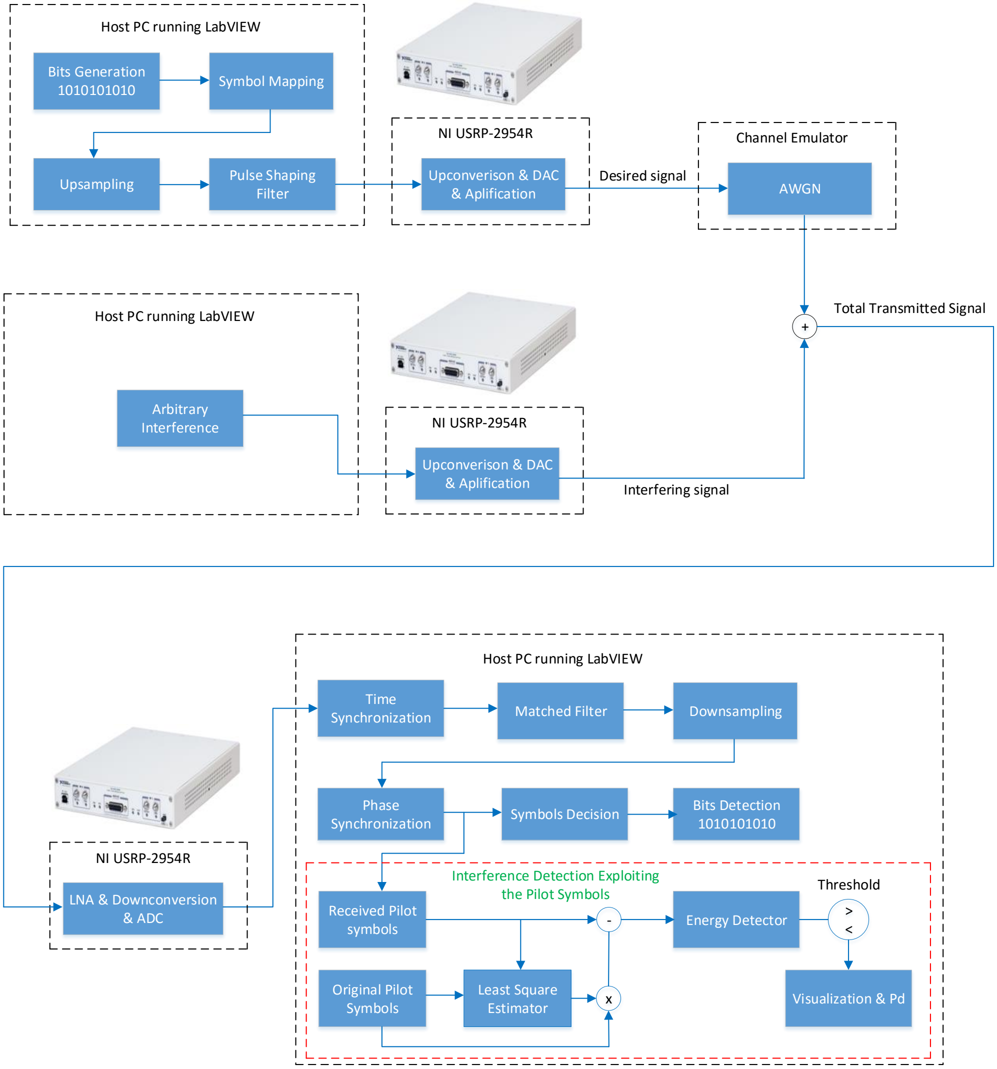
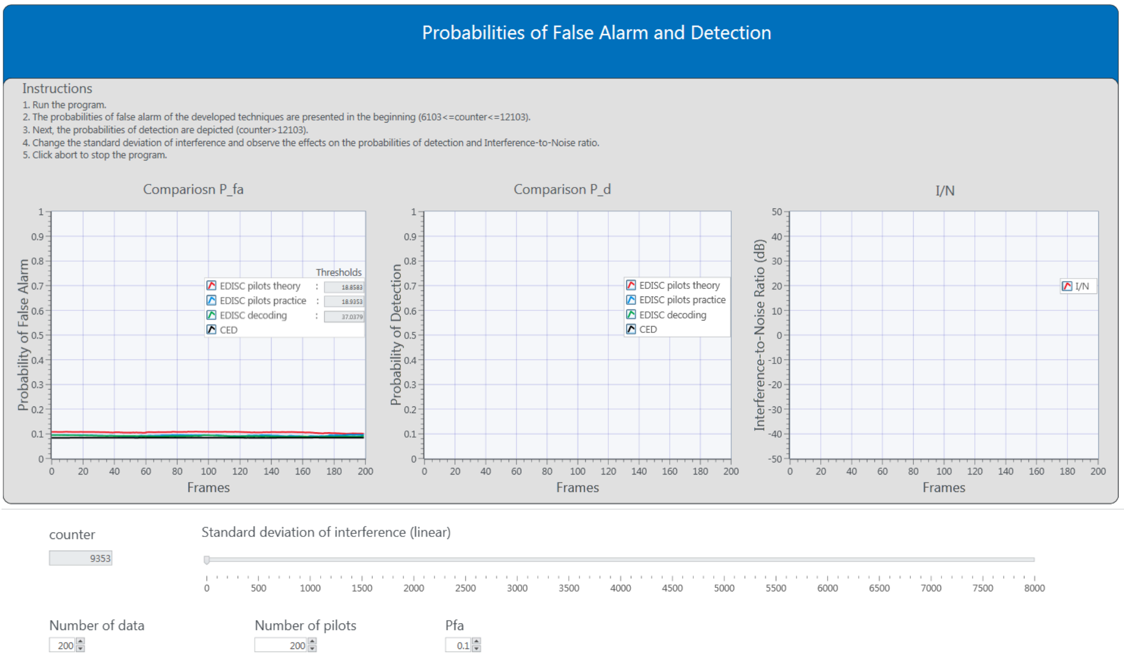


FIGURE 14. Flowchart of the complete interference detection system.



Probabilities of False Alarm and Detection

Instructions
1. Run the program.
2. The probabilities of false alarm of the developed techniques are presented in the beginning (6103<=counter<=12103).
3. Next, the probabilities of detection are depicted (counter>12103).
4. Change the standard deviation of interference and observe the effects on the probabilities of detection and Interference-to-Noise ratio.
5. Click abort to stop the program.

Thresholds
EDISC pilots theory : 18.8583
EDISC pilots practice : 18.9353
EDISC decoding : 37.0379

counter: 9353

Standard deviation of interference (linear)

Number of data: 200

Number of pilots: 200

Pfa: 0.1

FIGURE 15. Visualization of the probability of false alarm of EDISC with pilots, EDISC with decoding errors and CED for $SNR \approx 8$ dB.

The block diagram of this experimental set-up for the EDISC exploiting the pilot symbols is presented in 14.

1) Probability of detection in real time

As mentioned earlier, after the derivation of the decision threshold, we assume the experimental set-up of Figure 10, where the TX2 port transmits interference which is added to desired signal generated by the TX1 port. Then, we can apply the three aforementioned algorithms and the calculation of the P_D in real time is obtained following the same methodology as for the case of the P_{FA}.

In the beginning we focus on the EDISC exploiting the pilot symbols. Then, in each frame where the energy of the remaining signal after the pilots cancellation is higher than the decision threshold, we increase an initialized to zero counter by one and then, we divide the new result by the current number of frame. Therefore, in each frame, we update the probability of detection and provide a visualization of it in real time. Similar methodology and implementation performed for the EDISC exploiting the data and the CED.

V. RESULTS

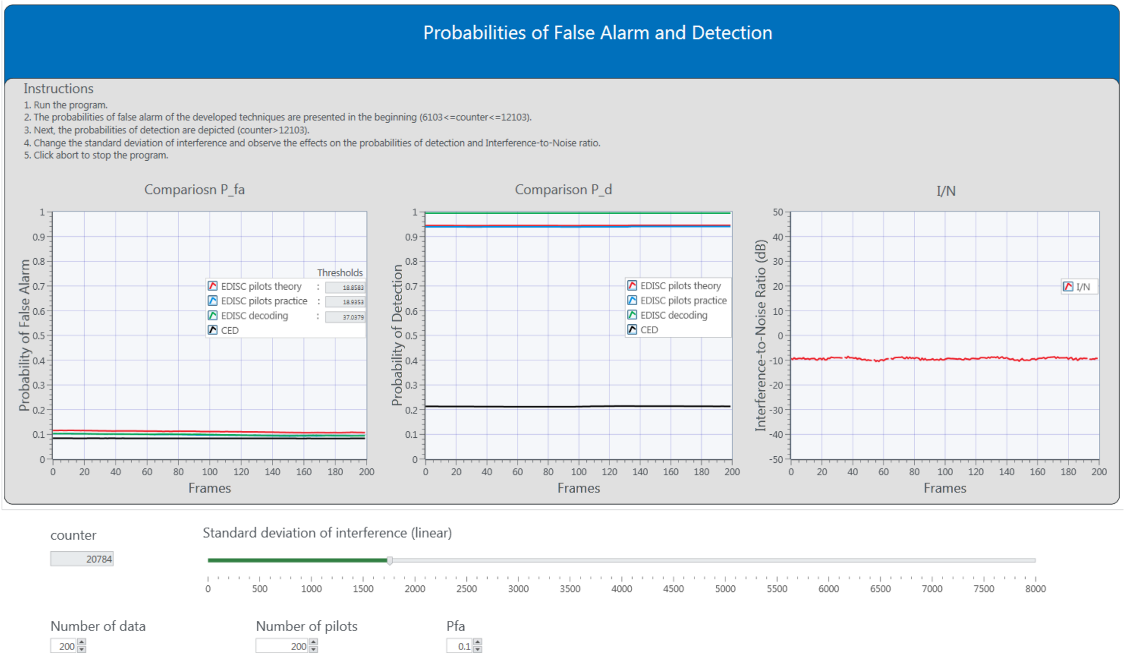
In this part, we present experimental results in order to:

1) verify that the theoretical and practical approaches for the derivation of the decision threshold can guarantee $P_{FA} = 0.1$;
2) depict the probabilities of false alarm and detection in real time; and
3) present the detection or not of interference through squared light emitting diodes (LEDs).

As mentioned earlier, the TX1 port of Figure 10 generates the desired transmitted signal, which is QPSK modulated. In order to set a specific SNR, the desired signal is sent to the channel emulator, which injects on it artificial AWGN noise. This SNR is set to 8 dB. Then, the TX2 port of Figure 10 generates Gaussian interference and these signals are added in their analog waveforms and sent to RX port of Figure 10 for further processing. For the evaluation of the algorithms, the number of used pilots is set to $N_p = 200$, while the number of data symbols $N_d = 200$. Here, we have to mention that the EDISC exploiting the pilots uses N_p samples, while the EDISC with data and the CED use $N_p + N_d$ samples. Furthermore, the probability of false alarm is set to $P_{FA} = 0.1$.

A. EVALUATION OF THE DECISION THRESHOLD AND PERFORMANCE ANALYSIS WITH RESPECT TO P_{FA} AND P_D

Figure 15 presents the first panel for the evaluation of the probabilities of false alarm and detection of the aforementioned algorithms. Our first step is to run the program and then, we devote a number of frames, in this case $N_f = 6102$, in order to estimate the averaged noise variance and derive the decision threshold of the detectors. For the EDISC exploiting the pilots, we derive the threshold based on both the theoretical and practical method. However, for the EDISC exploiting the data and the CED, we implement only the practical approach which is more accurate. As we see, in Figure 15, when $N_f = 6102$, the thresholds have been found. Furthermore, it is observed that the derived theoretical threshold is very close to the derived threshold with the



Probabilities of False Alarm and Detection

Instructions
1. Run the program.
2. The probabilities of false alarm of the developed techniques are presented in the beginning (6103<=counter<=12103).
3. Next, the probabilities of detection are depicted (counter>12103).
4. Change the standard deviation of interference and observe the effects on the probabilities of detection and Interference-to-Noise ratio.
5. Click abort to stop the program.

Thresholds
EDISC pilots theory : 18.8583
EDISC pilots practice : 18.9353
EDISC decoding : 37.0379

counter: 20784

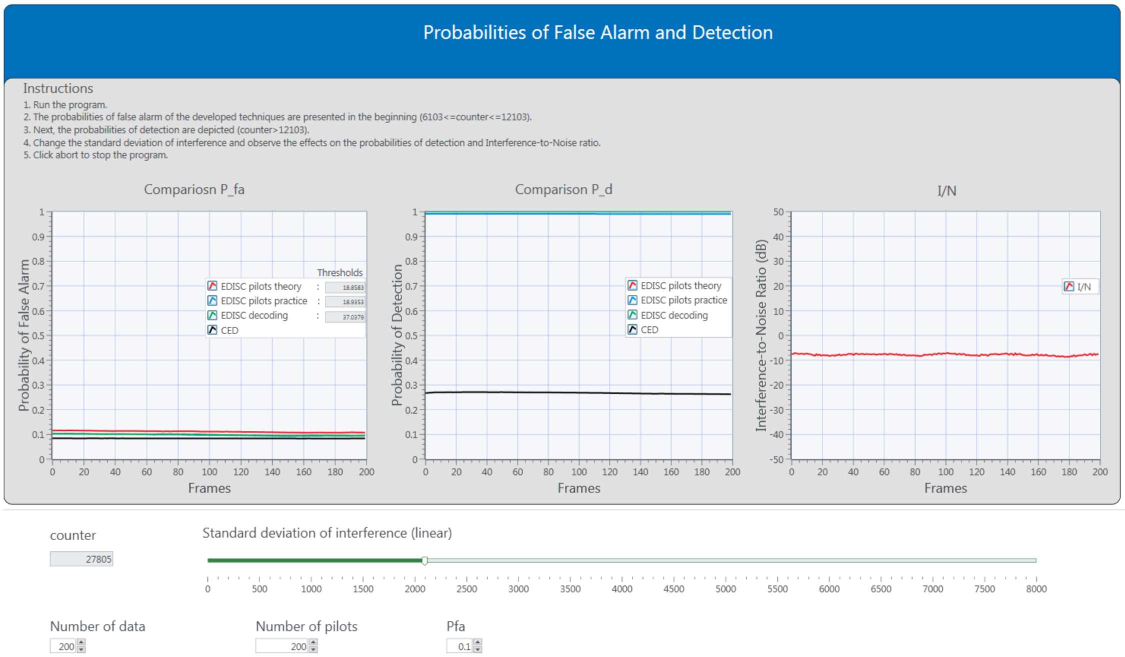
Standard deviation of interference (linear)

Number of data: 200

Number of pilots: 200

Pfa: 0.1

FIGURE 16. Visualization of the probability of detection of EDISC with pilots, EDISC with decoding errors and CED for $INR \approx -10$ dB.



Probabilities of False Alarm and Detection

Instructions
1. Run the program.
2. The probabilities of false alarm of the developed techniques are presented in the beginning (6103<=counter<=12103).
3. Next, the probabilities of detection are depicted (counter>12103).
4. Change the standard deviation of interference and observe the effects on the probabilities of detection and Interference-to-Noise ratio.
5. Click abort to stop the program.

Thresholds
EDISC pilots theory : 18.8583
EDISC pilots practice : 18.9353
EDISC decoding : 37.0379

counter: 27805

Standard deviation of interference (linear)

Number of data: 200

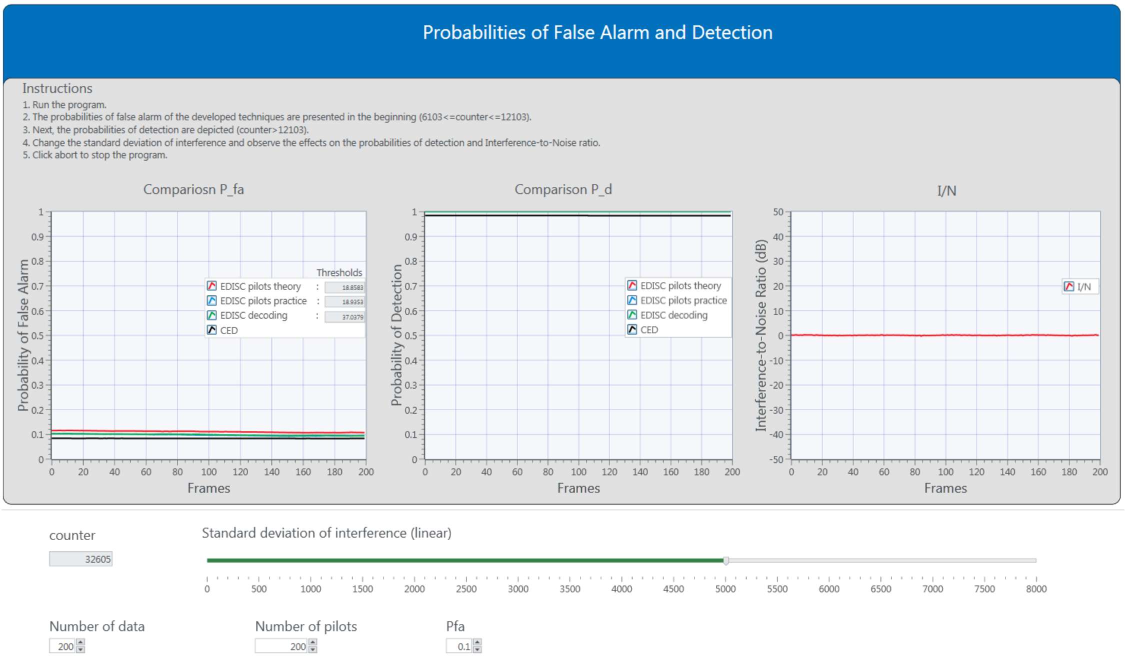
Number of pilots: 200

Pfa: 0.1

FIGURE 17. Visualization of the probability of detection of EDISC with pilots, EDISC with decoding errors and CED for $INR \approx -8$ dB.

practical approach. After the derivation of the thresholds, we devote again a new number of frames in order to calculate the probability of false alarm in real time. Figure 15 shows that the derived thresholds can guarantee $P_{FA} = 0.1$. Obtaining this goal, we verify the reliability of the methods for the derivation of the decision thresholds. Finally, we see that the theoretical formula for the P_{FA} of the EDISC exploiting the pilots can be applied in a practical system offering a reliable decision threshold.

After the calculation of the probability of false alarm, we continue with the calculation of the probability of detection. In this step, we start to introduce interference by increasing the standard deviation of interference from the horizontal slider and observing the effects in the figures of



Probabilities of False Alarm and Detection

Instructions

1. Run the program.
2. The probabilities of false alarm of the developed techniques are presented in the beginning (6103<=counter<=12103).
3. Next, the probabilities of detection are depicted (counter>12103).
4. Change the standard deviation of interference and observe the effects on the probabilities of detection and Interference-to-Noise ratio.
5. Click abort to stop the program.

Thresholds
EDISC pilots theory : 18.8583
EDISC pilots practice : 18.9353
EDISC decoding : 37.0379

counter: 32605

Standard deviation of interference (linear)

Number of data: 200

Number of pilots: 200

Pfa: 0.1

FIGURE 18. Visualization of the probability of detection of EDISC with pilots, EDISC with decoding errors and CED for $INR \approx 0$ dB.

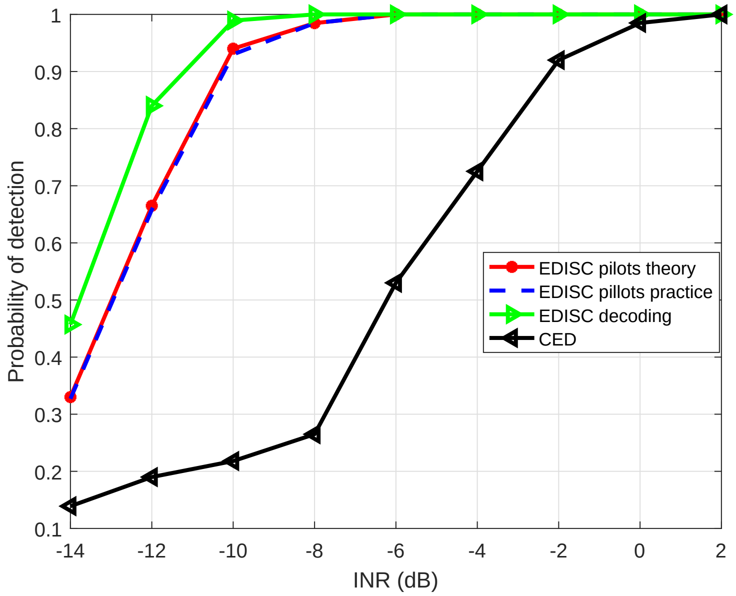


FIGURE 19. Probability of detection versus INR comparing EDISC with decoding, EDISC with pilots and CED.

the probability of detection and interference-to-noise ratio (INR). From Figure 16 is observed that the EDISC with pilots or data obtains much more reliable interference detection performance than the CED, particularly for low values of interference $INR \approx -10$ dB. Furthermore, we can notice that the EDISC with data performs better than the EDISC with pilots. It is explained by the fact that the EDISC with data has the ability to use the total number of symbols, while the EDISC with pilots is limited only on the number of pilots. Moreover, in Figure 17 we can see that the EDISC with pilots or data obtains $P_d \approx 1$ for $INR \approx -8$ dB, while the CED under the same scenario achieves $P_d \approx 0.27$. Furthermore, Figure 18 presents that the CED succeeds to obtain $P_d \approx 1$ when the $INR \approx 0$ dB. Therefore, it is obvious that our proposed algorithms can detect around 8 to 10 dB lower values of interference compared to CED. Finally, we validate that the theoretical P_D matches with the practical P_D under the considered EDISC with pilots. The same results are illustrated in Figure 19 for various values of INR (INR=-14:2 dB) and in Figure 20 for various values of

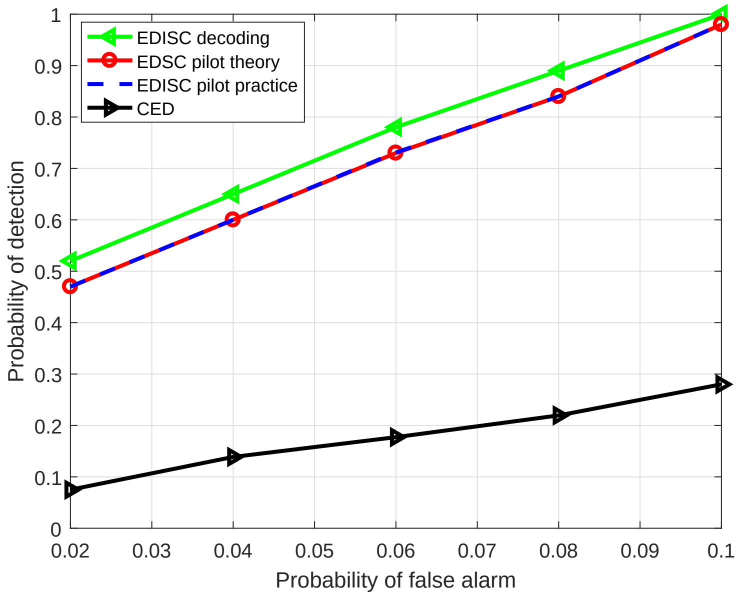


FIGURE 20. Probability of detection versus probability of false alarm comparing EDISC with decoding, EDISC with pilots and CED.

probability of false alarm ($P_{FA} = 0.02 : 1$).

B. VISUALIZATION PANEL FOR THE HYPOTHESIS TESTING

Figure 21 presents the second panel for the detection of interference. Except the probability of detection, another way to depict the detection of interference in real time is through a flush in squared LEDs. As shown in this figure, there are two LEDs which represent the case that the detection or not of interference is obtained using the CED and EDISC with the pilots. Both LEDs are white in the beginning. The white color corresponds in the absence of interference. In each frame the detectors try to detect the presence or not of the interference and if the interference is present the LEDs flush or change color and from white they become red. Therefore, the red color corresponds to presence of interference. Now, in the case where the interference is absent, the LEDs return to or keep the white color.

Figure 21 illustrates that when we change the standard deviation of interference from the horizontal slider and introduce strong interference, namely $INR \approx 0$ dB, both LEDs are red. However, if we reduce the level of the interference to $INR \approx -8$ dB, it is observed that the LED of the CED is most of the time white, while the LED of the EDISC with pilots is still red, as shown in Figure 22. Therefore, also from this panel with the flush of the LEDs, we can notice that the EDISC with pilots performs much better than the CED, especially for the detection of weak interference.

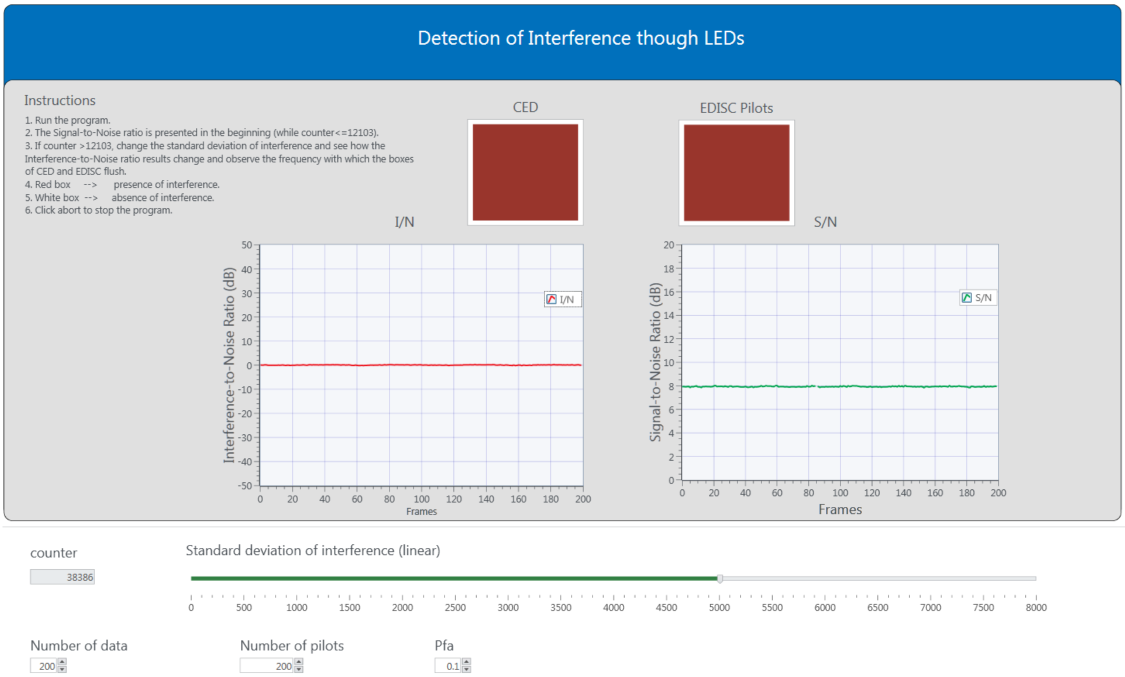
Finally, in this panel, we see that there is a graph which presents the estimated received signal-to-noise ratio, which is for the whole duration of the program 8 dB.

VI. CONCLUSIONS

In this paper, we presented a real communication system for the detection of interference using software defined radios, particularly USRPs. We programmed the USRPs using the LabVIEW Communications System Design Suite 2.0. software tool of National Instruments. Furthermore, we explained how we implemented each piece of this detection system and also discussed the challenges that we faced, such as the frame synchronization and noise variance estimation. We gave little more attention in the implementation of the decision threshold, which is the most critical part in the design of a detector. Then, we evaluated our developed algorithms for the detection of interference and compared them with the conventional energy detector. Finally, we demonstrated two panels. The first one presented in real time the probabilities of false alarm and detection in relation to the interference-to-noise ratio and the second panel showed how we can detect the interference through squared LEDs and again in relation to interference-to-noise ratio. Both panels showed that our proposed and developed detectors perform significantly better than the CED in the detection of weak interference, offering the capability of detecting 8 to 10 dB lower values of interference.

REFERENCES

[1] *Technology Roadmap: Interference Monitoring, Detection/Isolation, Classification, Localisation and Mitigation*, ESA document, May 2015.
[2] D.Cabric, S. M. Mishra, R. W. Brodersen, “Implementation issues in spectrum sensing for cognitive radios,," *Proc. Asilomar Conf. Signals, Syst., Comput., Pacific Grove*, CA, USA, pp. 772-776, Nov. 2004.
[3] H. Urkowitz, “Energy detection of unknown deterministic signals," *Proc. of the IEEE*, vol. 55, no. 4, pp. 523-531, Apr. 1967.
[4] F. Digham, M.-S. Alouini, M.K. Simon, “On the energy detection of unknown signals over fading channels," *IEEE Trans. Commun.*, vol. 55, no. 1, pp. 21-24, Jan. 2007.
[5] R. Tandra, A. Sahai, “Fundamental limits on detection in low SNR under noise uncertainty," *Proc. IEEE Int. Conf. Wireless Netw., Commun. Mobile Comput.*, Maui, HI, USA, vol. 1, pp, 464-469, Jun. 2005.
[6] S. Atapattu, C. Tellambura, H. Jiang, “Performance of an energy detector over channels with both multipath fading and shadowing," *IEEE Trans. Wireless Commun.*, vol. 9, no. 12, pp. 3662-3670, Dec. 2010.
[7] P.D. Sutton, K.E. Nolan, L. E. Doyle, “Cyclostationary signatures in practical cognitive radio applications," *J. Sel. Areas Commun.*, vol. 26, no. 1, pp. 13-24, Jan. 2008.



Detection of Interference though LEDs

Instructions

1. Run the program.
2. The Signal-to-Noise ratio is presented in the beginning (while counter<=12103).
3. If counter >12103, change the standard deviation of interference and see how the Interference-to-Noise ratio results change and observe the frequency with which the boxes of CED and EDISC flush.
4. Red box --> presence of interference.
5. White box --> absence of interference.
6. Click abort to stop the program.

CED 🟥

EDISC Pilots 🟥

counter: 38386

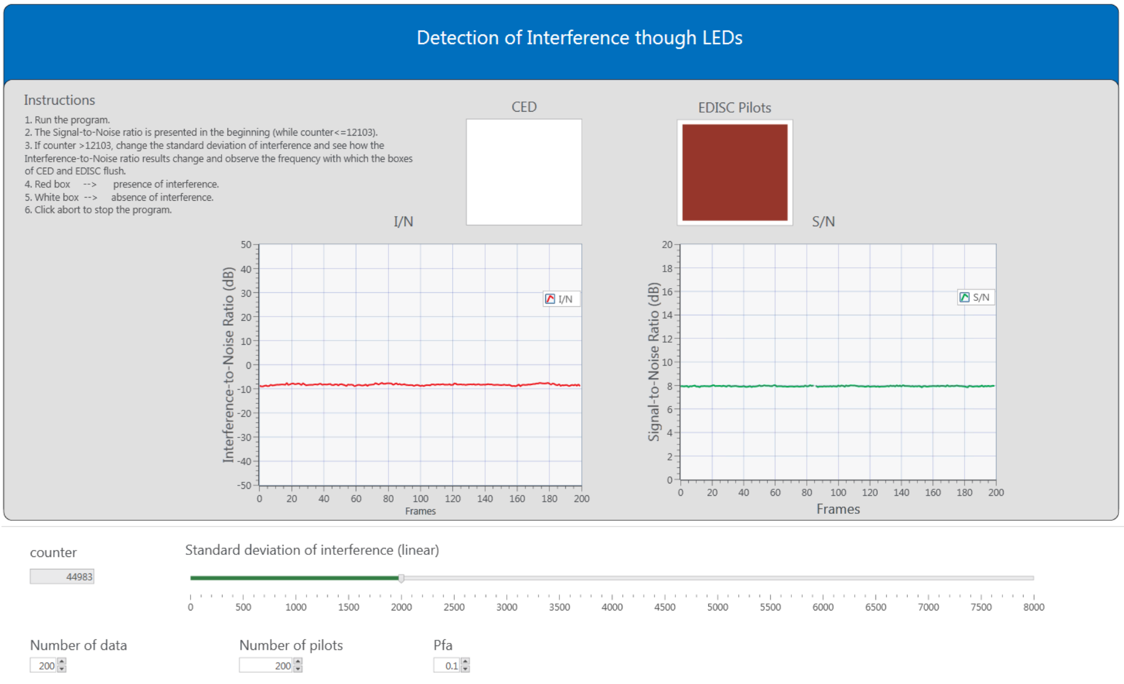
Standard deviation of interference (linear)

Number of data: 200

Number of pilots: 200

Pfa: 0.1

FIGURE 21. Interference detection through squared LEDs for $INR \approx 0$ dB.



Detection of Interference though LEDs

Instructions

1. Run the program.
2. The Signal-to-Noise ratio is presented in the beginning (while counter<=12103).
3. If counter >12103, change the standard deviation of interference and see how the Interference-to-Noise ratio results change and observe the frequency with which the boxes of CED and EDISC flush.
4. Red box --> presence of interference.
5. White box --> absence of interference.
6. Click abort to stop the program.

CED □

EDISC Pilots 🟥

counter: 44983

Standard deviation of interference (linear)

Number of data: 200

Number of pilots: 200

Pfa: 0.1

FIGURE 22. Interference detection through squared LEDs for $INR \approx -8$ dB.

[8] C. Politis, S. Maleki, C. Tsinos, S. Chatzinotas, B. Ottersten, “On-board the satellite interference detection with imperfect signal cancellation," *IEEE Intern. Workshop on Sig. Proc. Advanc.in Wirel. Comm.*, Edinburgh, Scotland, Jul. 2016.

[9] S. Smith, SES, “Satellite Interference Commercial Industry Views," presented at satellite Interference Reduction Group (sIRG), Jun. 2013.

[10] C. Politis, S. Maleki, C. Tsinos, S. Chatzinotas, B. Ottersten, “Weak interference detection with signal cancellation in satellite communications," *in IEEE International Conference on Acoustis, Speech, and Signal Processing (ICASSP)*, New Orleans, USA, Mar. 2017.

[11] O. Hiari, R. Mesleh, “A Reconfigurable SDR Transmitter Platform Architecture for Space Modulation MIMO Techniques," *in IEEE Access*, vol. 5, pp. 24214-24228, Oct. 2017.

[12] X. Lu, L. Ni, S. Jin, C-K Wen, W-J Lu, “SDR Implementation of a Real-Time Testbed for Future Multi-Antenna Smartphone Applications," *in IEEE Access*, vol. 5, pp. 19761-19772, Sep. 2017.

[13] National Instruments, “AN INTRODUCTION TO SOFTWARE DEFINED RADIO With NI LabVIEW and NI USRP."

[14] National Instruments, “USRP Software Defined Radio Device."

[15] National Instruments, “USRP-2954."

[16] National Instruments, “USRP-2954 Specifications."

[17] National Instruments, “LabVIEW Communications System Design Suite 2.0 Readme."

[18] C. L. Clark, *LabVIEW Digital Signal Processing and Digital Communications*, McGraw-Hill, 2005.

[19] D. Cabric, A. Tkachenko and R. W. Brodersen, “Experimental study of spectrum sensing based on energy detection and network cooperation," First international workshop on Technology and policy for accessing spectrum, 2006.

[20] D. Cabric, A. Tkachenko and R. W. Brodersen, “Spectrum Sensing Measurements of Pilot, Energy, and Collaborative Detection," IEEE Military Communications Conference, 2006. MILCOM 2006., pp. 1-7, Oct. 2006.

[21] R. Deng, J. Chen, C. Yuen, P. Cheng and Y. Sun, “Energy-Efficient Cooperative Spectrum Sensing by Optimal Scheduling in Sensor-Aided Cognitive Radio Networks," IEEE Transactions on Vehicular Technology, vol. 61, no. 2, pp. 716-725, Feb. 2012.

[22] J. G. Proakis, M. Salehi, *Digital Communications*, 5th Edition, McGraw-Hill, 2008.

[23] F. Jabbarvaziri, M. Nokhbeh-Zaeem, G. Moradi, “Preamble design for symbol synchronization in frequency-selective fading channels,"*in IET Communications*, Apr. 2014.

[24] H.-S. Chen, W. Gao, D. G. Daut, “Spectrum sensing for OFDM systems employing pilot tones," *IEEE Trans. Wireless Commun.*, vol. 8, no. 12, pp. 5862-5860, Dec. 2009.

[25] H.-S. Chen, W. Gao, “Spectrum sensing for TV white space in North America," *IEEE J. Select. Areas Commun.*, vol. 29, no. 2, pp. 316-326, Feb. 2011.

[26] Schmidl, T.M., Cox, D.C.: “Robust frequency and timing synchronization for OFDM", *IEEE Trans. Commun.*, vol. 45, no. 12, 1997, pp. 1613âĂŞ1621.

[27] Minn, H., Bhargava, V.K., Letaief, K.B.: “A robust timing and frequency synchronization for OFDM systems," *IEEE Trans. Wirel. Commun.*, vol. 2, no. 4, 2003, pp. 822âĂŞ839.

• • •