GNSS Interference Detection Using a Compressed Sensing Analog to Information Converter Approach

Alexander Rügamer*, Ivana Lukčin*, Günter Rohmer*, Jörn Thielecke†
*Fraunhofer Institute for Integrated Circuits IIS, Nuremberg, Germany
†Friedrich-Alexander University of Erlangen-Nuremberg, Germany

BIOGRAPHY

Alexander Rügamer received his Dipl.-Ing. (FH) degree in Electrical Engineering from the University of Applied Sciences Wuerzburg-Schweinfurt, Germany in 2007. Since the same year he works at the Fraunhofer Institute for Integrated Circuits IIS in the field of GNSS front-end receiver development. He was promoted to Senior Engineer in February 2012. His main research interests focus on GNSS multi-band reception, integrated circuits and immunity to interference.

Ivana Lukčin has received her MSc. degree in Mathematics from the University of Zagreb, Croatia in 2012. Since then, she works at the Fraunhofer Institute for Integrated Circuits in the field of compressed sensing and snapshot positioning.

Günter Rohmer received his Dipl.-Ing. degree in Electrical Engineering in 1988 and the PhD in 1995 from the University of Erlangen, Germany. Since 2001 he is head of a department at the Fraunhofer Institute for Integrated Circuits dealing with the development of components for satellite navigation receivers, indoor navigation and microwave localization systems.

Jörn Thielecke received his Dipl.-Ing. and PhD degrees in Electrical Engineering from the Friedrich-Alexander University of Erlangen-Nuremberg, Germany. From 1991 to 2003 he was first with Philips Kommunikations Industrie, later with Ericsson where he led a research group on radio access for cellular mobile radio communication systems. From 2003 until 2004 he was responsible for the department of communications at the Research Establishment for Applied Science FGAN. Since 2004 he is professor at the Friedrich-Alexander University of Erlangen-Nuremberg focussing now on localisation and navigation systems.

ABSTRACT

GNSS signals are extremely susceptible to all types of interference. Therefore the GNSS bands should be constantly monitored to detect possible threats. Since the GNSS bands are up to 100 MHz wide, receivers sampling at a Nyquist rate (i.e. at least twice the bandwidth) have very stringent requirements in terms of data rates and data storage. In this paper, a compressed sensing random demodulator analog-to-information converter (AIC) architecture for interference detection is proposed. In this approach, only a sub-Nyquist sampling rate is required which greatly reduces the ADC requirements and the size of the raw data output file. For interference detection and recovery the AIC output is tested against several given interference models (assumed as a priori knowledge) using specific reconstruction algorithms. If successful, the original interference signal can be recovered. The structure of the AIC, the signal recovery, and the modeling of the interference source are described in detail. Moreover the performance of this method is assessed for three types of interference (CW, chirp, and pulse). Specifically, the interference detection probability is evaluated as a function of the interferences' signal-to-noise ratio (SNR) and of the downsampling factor used in the AIC.

INTRODUCTION

GNSS Bands and Interference

Figure 1 shows the L-band spectrum of the current and planned GPS and Galileo global navigation satellite system (GNSS) signals with the notation of their modulation names and carrier frequencies. The red signals are classified (e.g. signals for military purpose only), the blue ones are open signals. All current and upcoming signals are within the protected Radio Navigation Satellite Services (RNSS) band but only the L1/E1 and L5/E5 bands are within the even better protected spectrum allocated to Aeronautical Radio Navigation Services (ARNS). The other

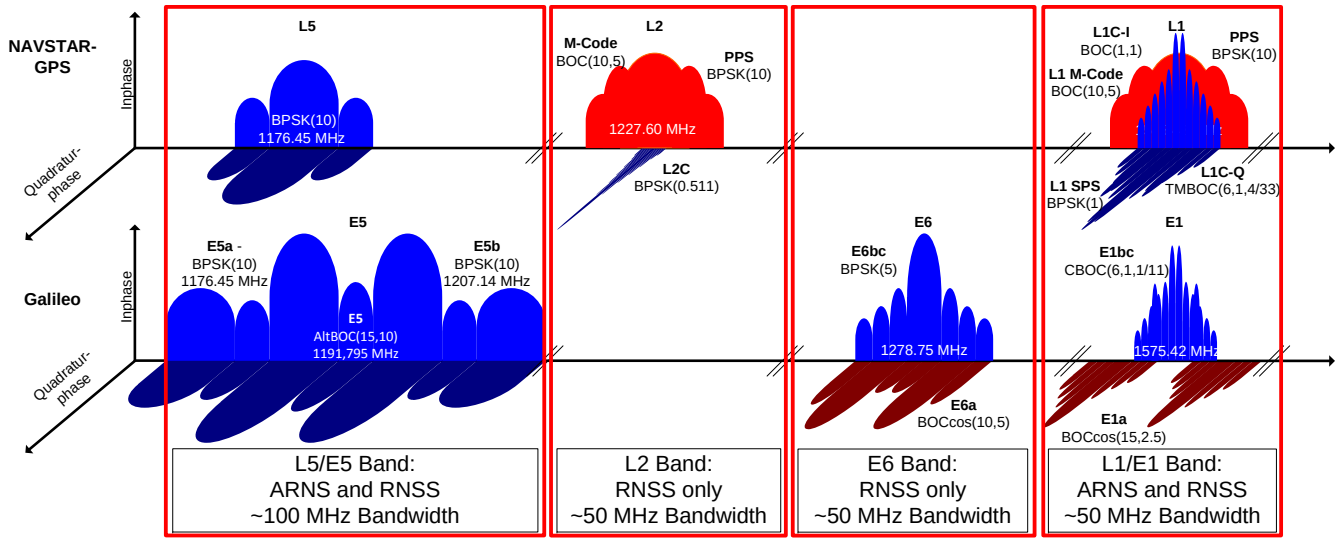


Figure 1. GPS and Galileo signals and their bands

two GNSS bands, E6 and L2, only protected through the RNSS, suffer from radar, military transmissions, and other potentially strong interferences.

Due to the inherently low power of GNSS signals (approx. -127 dBm received signal power on earth), the GNSS bands are dominated by white Gaussian noise. The noise is about hundred to a few thousand times stronger than the GNSS signal itself. As a consequence, the GNSS signals are extremely susceptible to all types of interference. The interferences can be unintentional like the harmonics of certain oscillators that translate into single continuous wave (CW)-tones or multitones in the GNSS spectrum. Moreover in the L5/E5 band, the GNSS service is sharing its bandwidth with systems like Distance Measuring Equipment (DME) and Tactical Air Navigation (TACAN) appearing as Gaussian shaped pulse signals. In the L2/E6 band strong military radar signals can appear. But, there are also more and more intentional interferences - so called jammers - readily available on the market, mostly sold over the internet, even if their use is illegal in most countries. The commercial jammers can often be characterized by a chirp signal. All these interferences have in common that a very small output power is sufficient to exceed the thermal noise floor and therefore to effectively affect the GNSS signals.

Since effective interference mitigation techniques - like array processing, or frequency domain adaptive filtering - are mostly unavailable to mass-market GNSS receivers and still relatively uncommon in professional receivers, it is necessary to monitor the GNSS bands of interest for later interference detection and elimination.

State of The Art

Different kinds of GNSS bands monitoring networks were installed, are currently under development or planned: [1], [2], [3]. [4]. To monitor a GNSS band, basically the complete broadcast bandwidth (around 50 MHz for Galileo E1, GPS L2 and Galileo E6, 100 MHz for the complete E5 band, see Figure 1) has to be supervised.

The Nyquist-Shannon sampling theorem states that the sampling rate F_s has to be at least twice the bandwidth of the signal to be digitized in order to avoid aliasing effects. The bandwidth of interferer signals are typically much smaller than that of the overall frequency band but their location is unknown. So, without exploiting any a priori knowledge about the interferer the complete GNSS signal bandwidth to be monitored has to be digitized.

The required Nyquist sampling rate of an analog-to-digital converter (ADC) used to digitize a 50 or 100 MHz bandwidth is therefore at least 100 or 200 mega samples per second (MSPS), respectively. Moreover a high dynamic range is required since the GNSS signals are around the thermal noise floor power while the interferences can easily reach 80 dB or more. So, a 14 or even 16 bit ADC is needed. Looking at the current state of the art, sophisticated 16 bit ADCs with 200 MSPS are available to digitize the complete E5 band but they are expensive and have very high power consumption and stringent jitter requirements. They also generate very high data rates resulting in very large files that need to be stored for post-processing.

E.g. the data acquisition system used in [5] to characterize different GNSS jammers features 16-bit I/Q samples with 62.5 MHz resulting in a raw data rate of 2 GBit/s or 250 MByte/s which is already too high for most hard drives for constant data recording. Such a system is not only very expensive but also produces very large amounts of raw data making storage and post-processing very cumbersome.

Thinking about building a regional interference monitoring network, a certain amount of stations have to be present to capture the raw signal and transmit their measurements

(e.g. the raw data) to a central server. This server can then process the raw-snapshots, detect and possibly also localize the interference source [6]. The high raw-snapshot sizes of current Nyquist sampling data acquisition systems make the data transfer and storage very demanding since e.g. no mobile network connection can be used.

New Approach

In this paper a much more efficient approach is presented to detect all kinds of known interference sources using a compressed sensing analog-to-information converter (AIC) featuring a sub-Nyquist rate sampling process. The requirements for this technique to work are some a priori knowledge about the interference signals to be detected (e.g. CW-tones, certain chirp or pulse signals) and the existence of a domain where the interference signal's representation is sparse. Then the ADC's sampling rate can be decreased to approx. O($k \log^6 W$), where k is the number of interferences and W being the bandwidth to be monitored [7]. As an example: instead of the 200 MSPS that would normally be required, the compressed sensing AIC technique could detect one CW-tone or one kind of chirp interferer within the 100 MHz E5-band, using approximately 330 kSPS (a reduction factor of more than 600!).

The extremely low ADC sampling rate in the AIC helps to minimize the raw-snapshot size: The signal is "compressed" since only a certain signature and not the signal itself is stored. This enables an interference monitoring network where the compressed raw-samples are shared using an inexpensive low-rate mobile network connection. The compressed raw-samples are then post-processed on a central server with high computational power and a catalog of characterized interference templates that can then be detected and optionally also localized.

AIC INTERFERENCE DETECTION RECEIVER

Compressed Sensing

Compressed sensing is a data acquisition protocol which samples at a sub-Nyquist rate and later reconstructs the original data from an incomplete set of measurements.

Let $\mathbf{f} \in \mathbb{C}^n$ be the signal to be recovered. Instead of $\mathbf{f}$ n measurements are stored in the vector $\mathbf{y} \in \mathbb{C}^n$. The sensing matrix $\mathbf{\Phi}$ makes the connection between signal and measurements by forming a linear system

$$\mathbf{y} = \mathbf{\Phi f}. \tag{1}$$

For the compression only $m \ll n$ out of all n measurements are used. So the measurement system is

$$\hat{\mathbf{y}} = \mathbf{R}\mathbf{y} = \mathbf{R}\mathbf{\Phi}\mathbf{f} \in \mathbb{C}^m, \tag{2}$$

where $\mathbf{R}$ is an $m \times n$ matrix that samples m out of n measurements. This system is generally ill-posed because it has no unique solution or, in other words, underdetermined with an infinite number of solutions.

The theory of compressed sensing claims that this ill-posed problem can be solved and the original signal $\mathbf{f}$ recovered if a proper base representation matrix $\mathbf{\Psi}$ is chosen in which this signal has a unique sparse representation [8], [9].

The signal $\mathbf{f}$ can be expanded in an orthonormal basis $\mathbf{\Psi}$ where it has a sparse representation $\mathbf{x}$:

$$\mathbf{f} = \mathbf{\Psi}\mathbf{x} \tag{3}$$

Moreover an incoherency between the sensing matrix $\mathbf{\Phi}$ and the representation matrix $\mathbf{\Psi}$ is required. The coherence can be measured with [8]

$$\mu(\mathbf{\Phi}, \mathbf{\Psi}) = \sqrt{n} \cdot \max_{1 \le k,j \le n} |\langle \phi_k, \psi_j \rangle| . \tag{4}$$

The low coherence can be obtained e.g. by using an independent identically distributed Gaussian random sensing matrix $\mathbf{\Phi}$ with ± 1 binary entries and any fixed representation for $\mathbf{\Psi}$. The incoherency ensures that the small number of taken samples still contains enough information for a successful reconstruction afterwards.

Under these and some further assumptions it can be shown that the undetermined linear system can be solved thanks to the sparsity of the signal in the appropriately chosen representation basis.

Possible Hardware Implementation

For the interference detection receiver described hereafter, the sampling process is performed by an AIC hardware converter using a random demodulator.

The basic idea behind the random demodulator is that if a signal, which has to have a sparse representation in a known domain, is spread with a pseudorandom noise (PN) sequence, a distinct signature of the signal is present at all frequencies. If then only a small portion of the spectrum is recorded (e.g. using a low-pass filter), the amount of information acquired is still sufficient to enable the full reconstruction of the signal.

The reconstruction algorithm then uses the information of the hardware AIC - namely the filter response, the pseudorandom sequence used, the compression or downsampling ratio - and the information about the domain in which the signal to be reconstructed has a sparse representation to recover the signal.

Figure 2 shows the block diagram of a possible hardware implementation of the AIC interference detection receiver.

An active multiband antenna is used to receive the GNSS band to be monitored. The first low noise amplifier (LNA) guarantees a good overall noise figure for the receiver according to Friis' Formula. The RF bandpass filter then attenuates all unwanted out-of-band frequencies.

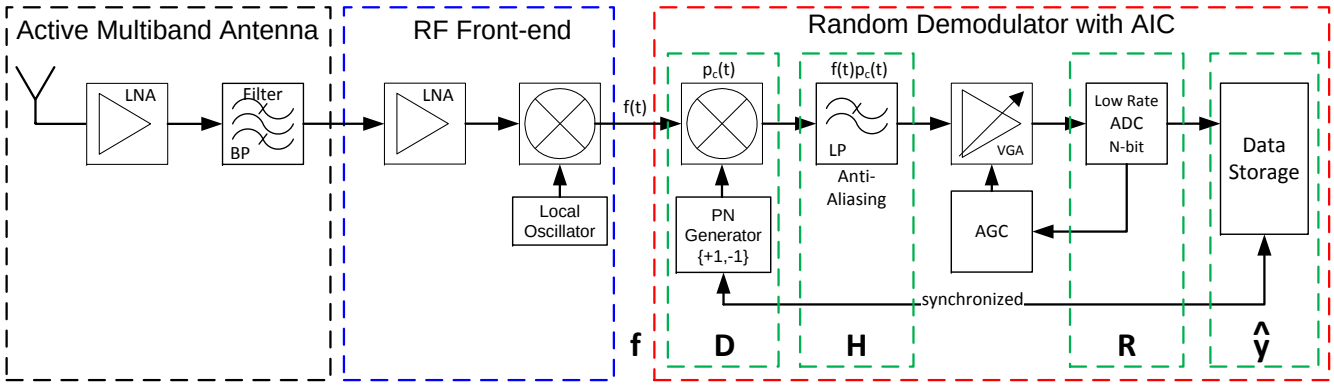


Figure 2. Block diagram of the proposed AIC interference detection receiver

In the RF front-end the antenna output signal is amplified again and down-converted using either a real mixer or a complex I/Q demodulator to an intermediate frequency or complex baseband signal $f(t)$, respectively.

The random demodulator with AIC consists of a pseudorandom sequence demodulator, an anti-aliasing low-pass filter, an analog-to-digital converter with an appropriate automatic gain control (AGC), and a data storage device. The AGC uses a variable gain amplifier (VGA) to ensure that the ADC's input signal is within the full scale range providing the maximum effective number of bits.

The pseudorandom sequence demodulator uses a predefined ± 1 data sequence $p(t)$ with a sampling rate P_c being the Nyquist rate for the signal band of interest. The length of the pseudorandom sequence is conditioned by the number of samples n to be acquired. Moreover the pseudorandom sequence is synchronized to the data storage device which is required for the later reconstruction phase. Since only multiplications with ± 1 are necessary, the random demodulator can be efficiently implemented in hardware e.g. using simple flip-flops.

Referring to the theoretical model, the signal $\mathbf{f}$ is first modulated with a pseudorandom sequence $p = (p_1, ..., p_n)$ and $p_i \in \{-1, 1\}$ resulting in $\mathbf{f}_{\text{spread}} = \mathbf{Df}$, for a pseudorandom matrix $\mathbf{D} \in \mathbb{R}^{n \times n}$

$$\mathbf{D} = \begin{bmatrix} p_1 & 0 & \cdots & 0 \\ 0 & p_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \cdots & p_n \end{bmatrix}.$$

The spread output of the random demodulator is filtered by an anti-aliasing low-pass filter whose bandwidth is set according to the Nyquist frequency of the subsequent low-rate ADC. It is assumed that the filter behavior can be described with filter coefficients in $h(t)$ having an order l.

Using the low-pass filter matrix $\mathbf{H} \in \mathbb{R}^{n \times n}$

$$\mathbf{H} = \begin{bmatrix} h(1) & 0 & \cdots & \cdots & \cdots & 0 & 0 \\ h(2) & h(1) & \cdots & \cdots & \cdots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & h(l-1) & \cdots & h(1) & 0 \\ 0 & 0 & \cdots & h(l) & \cdots & h(2) & h(1) \end{bmatrix}$$

the system can be rewritten as

$$\mathbf{y} = \mathbf{H}\mathbf{f}_{\text{spread}} = \underbrace{\mathbf{HD}}_{\mathbf{\Phi}}\mathbf{f} = \mathbf{\Phi f}. \tag{5}$$

Between the pseudorandom generator's sampling or chipping rate P_c and the ADC sampling rate F_s, a compression ratio or downsampling factor DSF can be defined. It basically provides the advantage of the whole concept since instead of n acquired samples, only m samples have to be stored with $m \ll n$:

$$DSF = P_c/F_s = n/m \tag{6}$$

For a rational compression ratio, the compression matrix $\mathbf{R} \in \mathbb{R}^{m \times n}$ can be described as

$$\mathbf{R} = \begin{bmatrix} 1 \cdots 0 & \cdots & \cdots & \cdots & 0 \\ \underbrace{0 \cdots 0}_{\lfloor \text{DSF} \rfloor - 1} & 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 \cdots 0 & \cdots & \cdots & \cdots & \end{bmatrix}.$$

After this downsampling or compression the systems provides the measurement output as

$$\hat{\mathbf{y}} = \mathbf{R\Phi f}. \tag{7}$$

Signal Recovery

The goal of the signal recovery process is to reconstruct a discretized version of the original signal $f(t)$ with a sampling rate P_c using the compressed measurements $\hat{\mathbf{y}}$, the

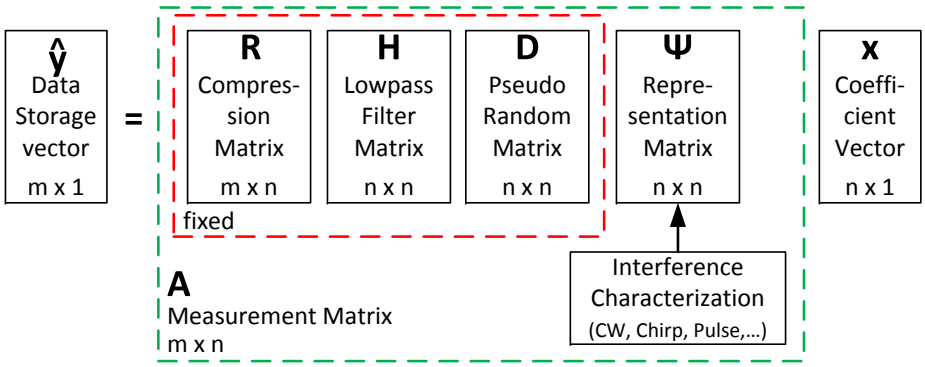


Figure 3. Block diagram of the signal recovery with $m \ll n$ samples

information of the random demodulator $\mathbf{R\Phi}$, and the a priori knowledge of the signal to be recovered, e.g. in which base $\mathbf{\Psi}$ the signal $\mathbf{f}$ has a sparse representation coefficient vector $\mathbf{x}$. Putting all this information together the measurement matrix $\mathbf{A}$ is introduced, see also Figure 3:

$$\hat{\mathbf{y}} = \underbrace{\mathbf{R\Phi\Psi}}_{\mathbf{A}}\mathbf{x} = \mathbf{Ax}. \tag{8}$$

Having solved this equation for $\mathbf{x}$, a discretized version of the original signal $f(t)$ can easily be found since $\mathbf{f} = \mathbf{\Psi x}$.

The challenge of the recovery is that Equation 8 is underdetermined and therefore has an infinite number of solutions and cannot be solved uniquely. But with the constraint of sparsity - that only the solution with the minimal number of nonzero coefficients is the right one - the problem can be solved uniquely.

Different signal recovery algorithms for this can be found in the literature e.g. greedy pursuits, convex relaxation, combinatorial algorithms classes [10], [11], [12], [13], [14]. Convex relaxation algorithms are very successful in reaching a good solution but are computationally expensive. The greedy pursuits algorithms are known for their speed but there is also a group of the combinatorial algorithms that are even faster than greedy algorithms with very strict requirements on the measurement matrix [15].

Two signal recovery algorithms are used for the simulations presented later on in this paper: CoSaMP (Compressive Sampling Matching Pursuit) and a brute force algorithm.

The CoSaMP, as described in [14], is based on a greedy pursuit approach that incorporates ideas from combinatorial algorithms to guarantee speed and to provide rigorous error bounds. The iterative CoSaMP first identifies the locations of nonzero elements in $\mathbf{x}$ and then estimates its values. CoSaMP is relatively fast, can recover K-sparse solutions, and preserves amplitudes.

The performance of the CoSaMP algorithm largely depends on the restricted isometry property (RIP). For each integer $K = 1, 2, ...$, the isometry constant δ_K of a matrix $\mathbf{A}$ is defined as the smallest number for which

$$(1 - \delta_K)||\mathbf{x}||_{l_2}^2 \leq ||\mathbf{Ax}||_{l_2}^2 \leq (1 + \delta_K)||\mathbf{x}||_{l_2}^2 \tag{9}$$

holds for all K-sparse vectors $\mathbf{x}$. The vector $\mathbf{x}$ is said to be K-sparse if it has only K nonzero elements.

It can be loosely stated that a matrix $\mathbf{A}$ obeys the RIP of order K if δ_K is not to close to one. In other words if $\mathbf{A}$ obeys the RIP than $\mathbf{A}$ approximately preserves the Euclidean length of K-sparse signals. RIP is equivalent to the fact that all subsets of K columns taken from $\mathbf{A}$ are actually nearly orthogonal.

The CoSaMP starts with the initial approximation $\mathbf{x} = \mathbf{0}$ in order to recover the K-sparse solution. At each step the residual vector

$$\mathbf{r} = \mathbf{Ax} - \hat{\mathbf{y}} \tag{10}$$

and the correlation vector

$$\mathbf{u} = \mathbf{A}^*\mathbf{r} = \mathbf{A}^*(\mathbf{Ax} - \hat{\mathbf{y}}) \tag{11}$$

are computed.

Thanks to the RIP, for the K-sparse signal $\mathbf{x}$ the vector $\mathbf{A}^*\mathbf{Ax}$ can serve as a proxy for the signal because the energy in each set of K components of the mentioned vector approximates the energy in the corresponding K components of $\mathbf{x}$. This is the main reason behind the identification of the $2K$ largest components in $\mathbf{u}$, or the components with the largest energy, whose indexes are put in the set I (which was at the beginning an empty set). On this set a least squares problem is solved

$$\hat{\mathbf{y}} = \mathbf{A}_I\mathbf{x}_I \tag{12}$$

where the matrix $\mathbf{A}_I$ is the restriction of $\mathbf{A}$ to set I by just taking the columns at positions defined in I. The solution at each step is equal to $\mathbf{x}_I$ for positions in I and elsewhere 0. The process is repeated until a certain number of steps is done or until CoSaMP reaches a predefined tolerance on the residual.

Another approach is the brute force recovery. It may be used in special cases where the reconstruction phase is not

blind. Using the a priori information the signal may be efficiently recovered with low computational cost.

In the special case when only one interferer shall be recovered (the sparsity is therefore one and only a single element of $\mathbf{x}$ is one), the brute force method calculates all possible residuals r_k with $k = 1, ..., n$ over all n combinations by

$$r_k = \|\mathbf{A}(:, k) - \hat{\mathbf{y}}\|_2. \tag{13}$$

The minimum of all r_k, $r_{\min}$ is the most likely solution. By using a certain threshold between the minimum $r_{\min}$ and the second minimum of the residuals, it is possible to decide if the tested interferer was present or not.

INTERFERENCE CHARACTERIZATION

Before the interference can be reconstructed or detected from the AIC output data, the interference source must be carefully characterized. This a priori knowledge is necessary for this compressed sensing interference detection method to work.

As depicted in Figure 3, the product of the $\mathbf{RHD}$ matrices stays the same for each recovery process. The representation matrix $\mathbf{\Psi}$ has to be adapted for each kind of interferer. This adaption is happening within the interference characterization step described in detail in the following. To conclude, for each interference to be checked a measurement matrix $\mathbf{A}$ (or its conjugate transpose) has to be calculated and can then be stored e.g. in an interference template catalog. In the detection and recovery process the compressed samples then have to be checked against the different catalog entries to determine if one of the characterized interferences was present in the captured signal.

The easiest kind of interference is a continuous wave (CW) tone interference which could e.g. be unintentional (a harmonic from some local-oscillator of a different system) or intentionally placed e.g. on the main lobes of the GNSS signal to be jammed. Moreover, also multitones are possible. The presented interference detection method detects both the frequency and the amplitude of the interference signal within the observation bandwidth. The frequency resolution f_{res} is principally limited by the number of acquired samples n and the sampling rate F_s according to $f_{\text{res}} = \frac{F_s}{n}$.

Recently some research institutes have characterized different kinds of jammers sold over the Internet, from small cigar-lighter type ones to very sophisticated multi-band jammers [16], [5]. In summary, most of them can be modeled as some kind of chirp generator distinguished by their parameters in terms of starting/stopping frequency and sweep period time. Therefore for each of the jammers, one input in the interference template catalog is necessary.

Known pulse interference coming from Distance Measuring Equipment (DME) and Tactical Air Navigation (TACAN) systems are present in the L5/E5 band. The different pulse signals can also be modeled in their time and frequency behavior and filed in the interference template catalog.

The first task for a successful detection is to find a certain representation base that exhibits the sparsity of the interference source to be detected. In the following, the construction of representation matrices $\mathbf{\Psi}$ that provide the required signal sparsity is described for each of the three types of the interferences: CW, chirp and pulse.

Tones and Multitones

As it can be seen in Figure 4(a), CW tones are sparse in the frequency domain. Thus, an appropriate basis function for the signal's sparse representation is an inverse Fourier transform. For time invariant tones and multitones the representation matrix is an $n \times n$ inverse discrete Fourier transform (IDFT) matrix

$$\mathbf{\Psi}_{\text{CW}} = \frac{1}{\sqrt{n}} \cdot \begin{bmatrix} 1 & 1 & \cdots & \cdots & 1 \\ 1 & e^{-\frac{2\pi i}{n}} & \cdots & \cdots & e^{-\frac{2\pi i(n-1)}{n}} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ & & \cdots & e^{-\frac{2\pi i(n-2)(n-2)}{n}} & e^{-\frac{2\pi i(n-2)(n-1)}{n}} \\ 1 & e^{-\frac{2\pi i(n-1)}{n}} & \cdots & e^{-\frac{2\pi i(n-1)(n-2)}{n}} & e^{-\frac{2\pi i(n-1)(n-1)}{n}} \end{bmatrix},$$

where n is the number of acquired samples.

To be able to guarantee the good performance of the CoSaMP, it would have to be proven that the measurement matrix $\mathbf{A} = \mathbf{RHD\Psi}$ satisfies the RIP property. In the literature the RIP property has been proven for special kinds of filter matrices $\mathbf{H}$ [17], [7]. This property has also been proven for the Toeplitz matrices, which is exactly the structure of the filter matrix $\mathbf{H}$ used in this paper, but with a property that the vector used in its construction is a Gaussian vector [18].

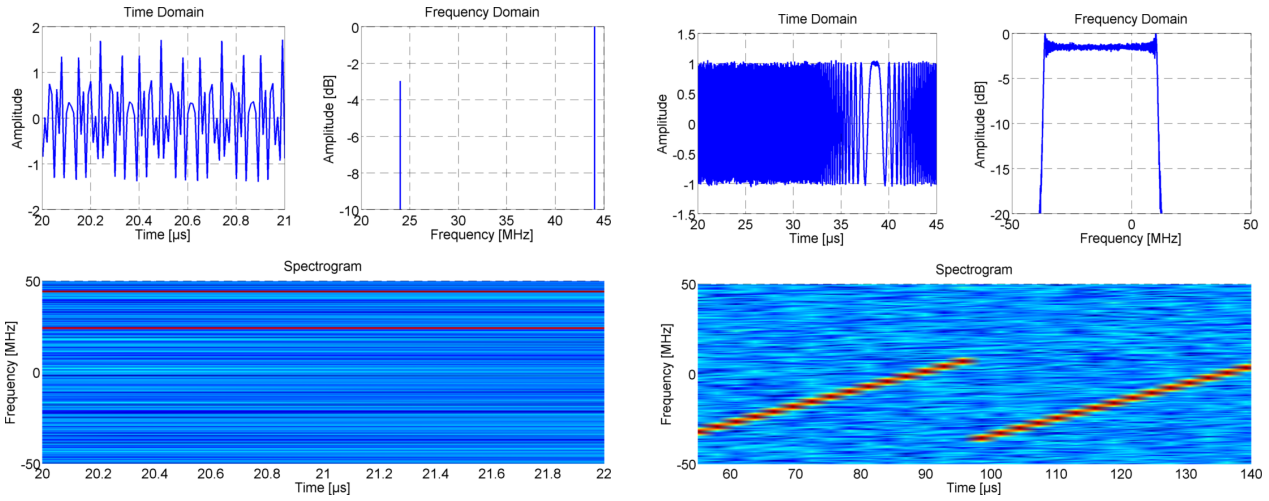
Even though the constructed matrix $\mathbf{A}$ used in this paper does not belong to any of these two groups, it was concluded that, from a numerical point of view, the RIP is sufficiently satisfied (otherwise the reconstruction would not be possible).

Chirp Signals

By definition, a chirp signal is a signal in which the frequency increases or decreases with a certain slope over the time. Types of a chirp signal are e.g.:

Linear chirp: frequency $f(t)$ varies linearly with time

$$f(t) = f_0 + kt,$$



(a) Exemplary multitone signal, SNR=30 dB, f_1=44 MHz, f_2=24 MHz, f_2 with half the power of f_1

(b) Exemplary chirp signal, SNR=30 dB, linear sweep, f_{start}=−37 MHz, f_{stop}=+11 MHz, period=50 µs

Figure 4. Time, Frequency and Spectrogram representation of CW and Chip signals

where f_0 is the starting frequency (at time $t = 0$) and k is the rate of frequency increase or chirp rate. The time-domain function for a sinusoidal linear chirp is the sine of a phase in radians

$$x(t) = \sin\left[2\pi\left(f_0 t + \frac{k}{2}t^2 + \phi_0\right)\right],$$

where ϕ_0 is the initial phase.

Exponential chirp: frequency $f(t)$ varies exponentially with time

$$f(t) = f_0 k^t,$$

where f_0 is the starting frequency (at time $t = 0$) and k is the rate of exponential increase in frequency. The time-domain function for a sinusoidal exponential chirp is the sine of a phase in radians

$$x(t) = \sin\left[2\pi f_0 \frac{k^t - 1}{\ln(k)} + \phi_0\right],$$

where ϕ_0 is the initial phase.

Figure 4(b) shows an example of a linear chirp signal with increasing frequency and a given repetition period in its time and frequency domain together with its spectrogram. The picture shows that the chirp is neither sparse in the time domain nor in the frequency domain.

For the construction of the representation base for chirps an a priori knowledge about a special chirp signal is used. The basic requirement is that the chirp signal's parameters are stable and repeated in a periodic way. To obtain an independence of time, so that the signal recording position does not have to be synchronized to the period of the chirp, exactly one period of the chirp is used for the representation base. A circulant matrix using a vector that contains exactly the samples of the signal is constructed. An advantage with this approach is that the a priori knowledge of the sparsity one can and will be used in the reconstruction phase.

A basic scheme can be seen on a small example where n samples $[v_1, \cdots, v_n]$ are taken and the size of the representation matrix is $n \times n$:

$$\mathbf{\Psi}_{\text{chirp}} = \begin{bmatrix} v_1 & v_n & \cdots & v_3 & v_2 \\ v_2 & v_1 & \ddots & \vdots & v_3 \\ \vdots & v_2 & \ddots & \vdots & \vdots \\ v_{n-1} & \vdots & \ddots & \vdots & v_n \\ v_n & v_{n-1} & \cdots & v_2 & v_1 \end{bmatrix}.$$

The set in which the solution $\mathbf{x}$ can be found is:

$$S = \{\mathbf{z} | \text{supp}(\mathbf{z}) = \{k\} \text{ and } \mathbf{z}|_{\text{supp}(\mathbf{z})} = 1, k = 1, \ldots, n\}$$

A brute force reconstruction can be used where every possibility from the possible solution set S will be tested to find the best match. Using the a priori knowledge that with this kind of representation matrix the sparsity level is exactly one, the brute force reconstruction algorithm even performs faster than CoSaMP.

Pulses

In Figure 5 one special kind of pulse is exemplary plotted in its time and frequency domain together with a spectrogram. It can be seen that this pulse has a sparse representation in the time domain.

If the pulse is in the form of a unique spike then it already has a sparse representation in the time domain and a proper representation base would be the spike base. But often

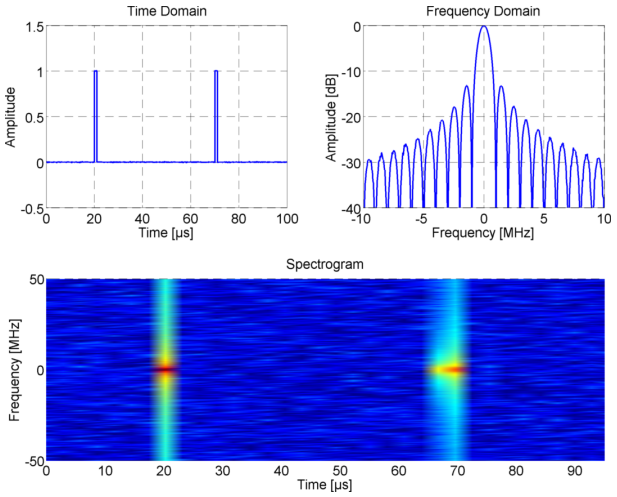


Figure 5. Exemplary Rectangular shaped pulse signal, SNR=30 dB, length $= 1$ µs

pulses are not received in that form: e.g. the DME/TACAN pulses have a certain Gaussian shaping with a distinct repetition frequency on a well defined frequency offset or channel. Then an approach similar to that previously described for chirp signals can be used where the pulse time behavior is described in the representation matrix $\mathbf{\Psi}$ obtaining independence in the time domain.

BASIC REQUIREMENTS

Based on the analysis made in the previous section some requirements for a reliable reconstruction have to be made: The signal to be detected has to be periodic and the number of acquired samples n must be large enough to include the period of the interference signal. Moreover n depends on the bandwidth to be observed: The higher the bandwidth, the larger the reconstruction matrices for a given time interval are.

Therefore n basically sets the matrix size within the reconstruction process and directly influences the computational complexity.

A trade-off has to be made between the computational power and the snapshot size of the monitored bandwidth. The CoSaMP is reported to have a complexity of $\mathrm{O}(n \log^2 n)$ [14]. Moreover it is important to find a proper input setting of the algorithm in accordance to the targeted application.

The other mentioned brute force reconstruction algorithm has a linear complexity of $\mathrm{O}(n)$ and performs therefore much faster but can only recover a sparsity of one.

SIMULATION RESULTS

To investigate the impact of the signal-to-noise ratio (SNR) - where signal actually refers to the interference to be detected - and the downsampling (DSF) or compression factor on the detection probability, Monte Carlo (MC) simulations were done. Each combination of SNR and DSF was simulated 1000 times, and the success rates are given in percentage. The position of the interference signal was chosen in a random uniformly distributed way.

The scenario in the simulation setup was to observe a bandwidth of 100 MHz as it is needed for the Galileo E5 band. A conventional Nyquist sampling approach would have to use e.g. at least a 100 MSPS I/Q ADC. In the proposed AIC approach, only the sampling rate P_c of the PN generator has to fulfil the Nyquist criteria. Therefore a low-rate ADC with a sampling rate of only 100/DSF MSPS can be used. The preceding anti-aliasing filter was set to have a 3 dB bandwidth according to the Nyquist frequency of the low-rate ADC. A simple FIR filter with an order l of 50 was used to that purpose. The acquired sample size n was 1000 while the stored sample size m was determined according to Equation 6.

MC simulations were also carried out to verify that the proposed methods could reliably distinguish between different kind of interferences that belong to the same group but have different parameters (e.g. between two chirp signals with different start and/or stop frequency) and between interferences from different groups (e.g. between a chirp and a CW or a pulse).

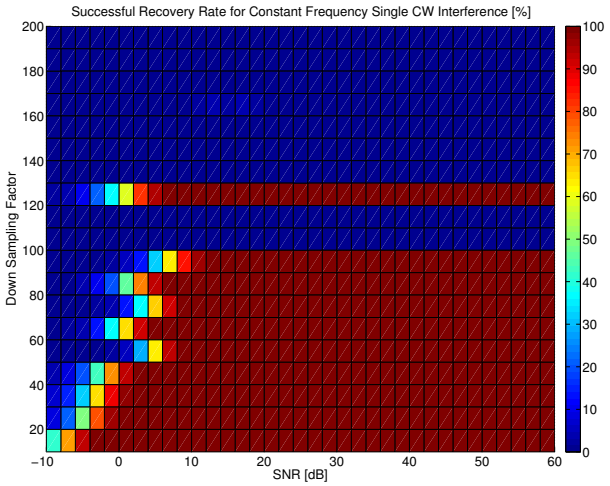
Single Tone Simulation

The reconstruction and detection of tones and multitones was done using CoSaMP in dependence of the downsampling factor and SNR.

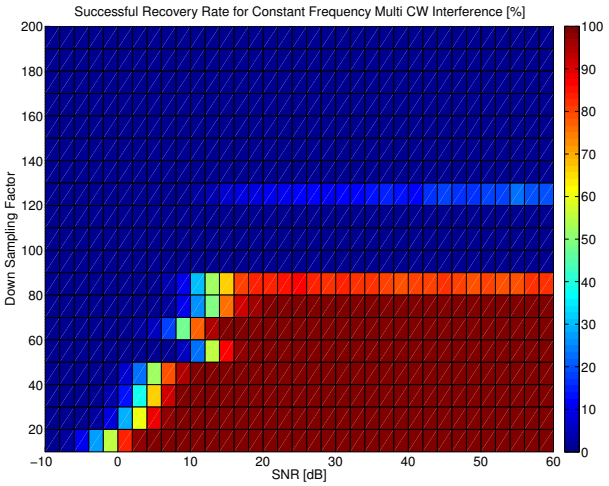
In Figure 6(a) the recovery rate of a constant single tone with $f_1 = 44$ MHz was evaluated. As expected, the detection probability is better for higher SNR and lower downsampling factors. What was not expected is the single line of excellent detection probability for a downsampling factor of 120. This artifact appears to origin from the combination of the PN sequence used and the fixed CW interference frequency. To prove this dependence, in Figure 6(c) the same constant CW interference was used but now the PN sequence was chosen afresh each MC iteration. Now this artifact disappears.

Finally, Figure 6(e) shows the results when an arbitrary chosen single tone frequency was used in each MC run. In this simulation the PN sequence was again not refreshed, like it is the case in a real hardware implementation.

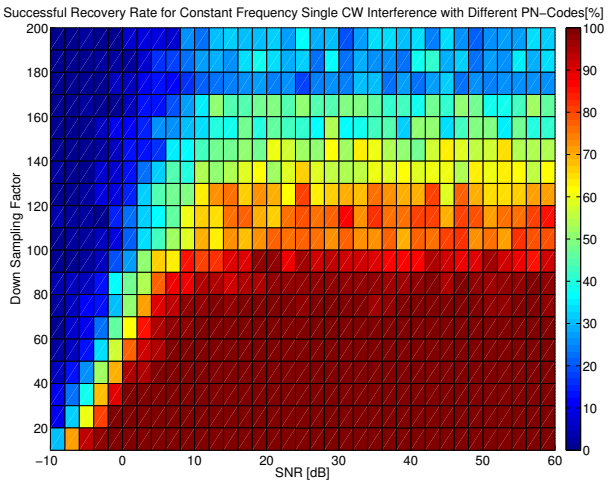
It can be seen that a downsampling factor of 80 still provides a high detection probability when the SNR is larger than 10 dB. In other words, instead of using an I/Q ADC with sampling rates of 100 MHz, a low-rate I/Q ADC with 1.25 MHz would be sufficient. Additionally, the snapshot size is inversely proportional to the downsampling factor.



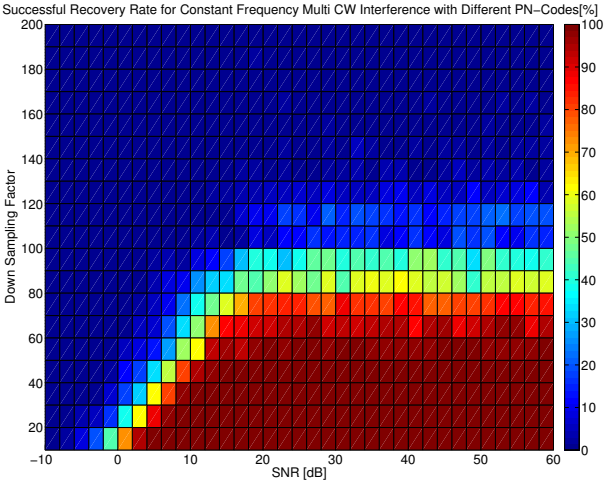
(a) Constant frequency single tone with the same PN sequence



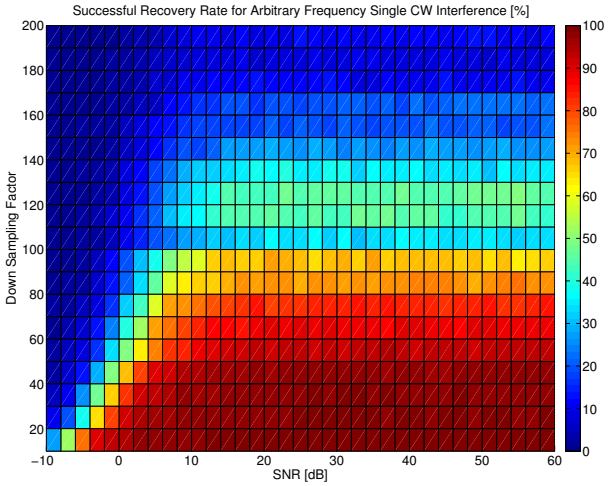
(b) Constant frequency two tones with the same PN sequence



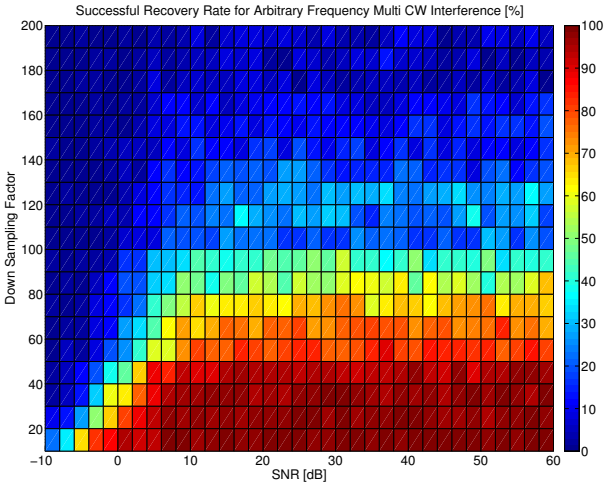
(c) Constant frequency single tone with different PN sequences



(d) Constant frequency two tones with different PN sequences



(e) Arbitrary frequency single tone with the same PN sequence



(f) Arbitrary frequency two tones with the same PN sequence

Figure 6. Monte Carlo simulation results of the recovery rate of single and multiple tone signals using CoSaMP in dependence of the downsampling factor and SNR

Multitones Simulation

In the same way as for a single tone interferer, a multitone scenario as depicted in Figure 4(a) with a first frequency of f_1=44 MHz and a second frequency f_2=24 MHz with f_2 having half the power of f_1 was simulated.

The detection was declared successful if the recovery algorithm reports both frequencies correctly. Moreover also the correct amplitude was monitored. In general the amplitude was always correct (within less then 0.1 dB variation) when the correct frequencies were recovered. This demonstrates the capability of the used CoSaMP recovery algorithm to preserve the recovered signal's amplitude.

The simulation results in 6(b) show similar results as the single tone simulations before. Since the first fixed frequency was the same as for the single CW test, also the artifact of a higher detection probability for a downsampling factor of 120 appears. When choosing the PN sequence afresh for each MC iteration the effect disappears, as shown in Figure 6(d).

In 6(f) both tone frequencies were chosen in a random way for each MC iteration.

Chirp Simulation

The recovery of a chirp signal with a start frequency of -37 MHz and a stop frequency of +11 MHz (as depicted in Figure 4(b)) was simulated while the chirp start period was uniformly randomly chosen within the recorded chirp period.

The used brute force signal recovery technique relies on the knowledge that the chirp representation matrix constructed as explained in Subsection Chirp Signals provides a sparsity of exactly one in $\mathbf{x}$ where all other values are zero. The brute force approach tests all possible solutions and returns the best match as explained in Subsection Signal Recovery and Equation 13.

The simulation results in Figure 7 show a perfect recovery rate even with downsampling factors of up to 200 and SNR values above 10 dB. These excellent results came from the brute force recovery approach and the fact the representation matrix and recovery algorithms are insensitive to the chirp's amplitude (it is always assumed to be normalized) in contrast to what CoSaMP is doing.

In practice it is necessary to distinguish between different kind of chirp signals - e.g. the different chirp jammers characterized in a database. Therefore it is necessary to reconstruct the result with different representation bases and to identify the most probable one. When the chirp is not included in the recovery database, then all the computed residuals should be similarly high because none of them is the solution.

To distinguish between different possible chirp signals the brute force approach also tests all possible solutions for every cataloged type of the chirp. The l_2-norm of the residual

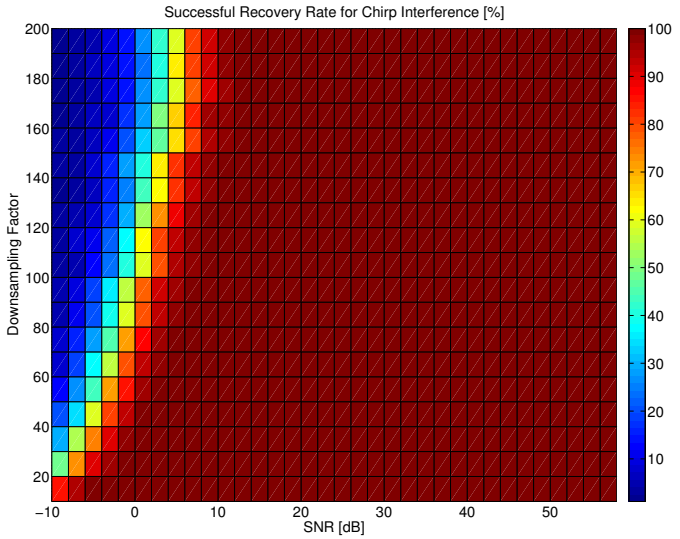


Figure 7. The recovery rate of a chirp signal with $f_{\text{start}} = -37\,\text{MHz}$, $f_{\text{stop}} = +11\,\text{MHz}$ using brute force in dependence of the downsampling factor and SNR

	Chirp_1	Chirp_2	Chirp_3	Chirp_4	Chirp_5
f_{start}	-1 MHz	2 MHz	0 MHz	2 MHz	-4 MHz
f_{stop}	2 MHz	-1 MHz	2 MHz	1 MHz	5 MHz
DSF=100 SNR [dB]	Chirp_1 [dB]	Chirp_2 [dB]	Chirp_3 [dB]	Chirp_4 [dB]	Chirp_5 [dB]
-10	0	0.35	0.38	0.26	0.40
-5	0	1.24	1.46	1.53	1.56
0	0	2.38	3.49	4.07	3.59
5	0	5.60	6.88	7.52	8.18
10	0	9.01	11.65	11.84	12.23
15	0	12.96	15.90	16.47	16.24
20	0	17.29	21.15	21.72	21.67
25	0	22.71	26.71	27.28	27.33
30	0	29.38	32.43	32.44	32.57
35	0	32.78	36.09	36.54	36.61
40	0	37.14	41.69	41.55	41.81
45	0	43.15	47.03	47.28	47.22
50	0	47.35	50.90	51.27	50.98

Table 1. The values of the $TH_i,\ i = 1, \ldots, 5$ for different SNR when Chirp_1 should be detected

$r_{i,k}$, where the subscripts i and k respectively represent the type of chirp and the solution set, is computed.

Then, for each chirp type the smallest value, e.g. $r_{i,\min} = \min_k r_{i,k}$ is taken. The distance to the other possible solutions is calculated in a logarithmic way and expressed as a normalized threshold TH:

$$TH_i = 20 \log \left[\min_i (1/r_{i,\min}) / (1/r_{i,\min}) \right]. \qquad (14)$$

A simulation with five different kind of chirps (different

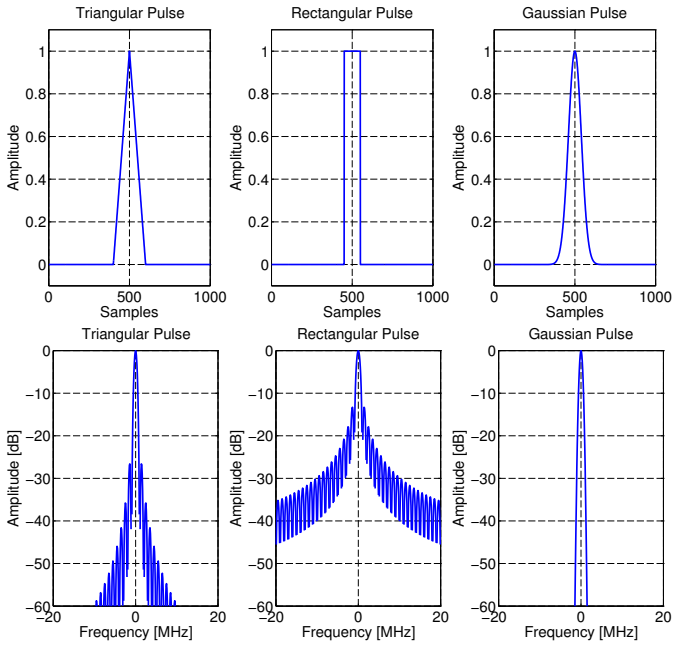


Figure 8. Different pulse shapes used in the simulation to be distinguished

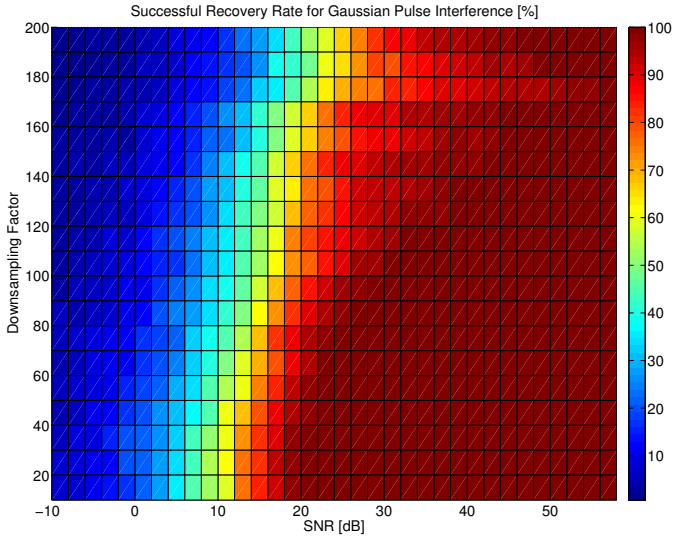


Figure 9. The recovery rate of a Gaussian pulse

start, stop frequencies and frequency rates) at a constant downsampling factor of 100 with different SNR was made and the threshold values TH was calculated according to Equation 14. The results are given in Table 1.

The conclusion based on Table 1 is that using a threshold TH of 5 dB, different kind of chirps with SNR values above 5 dB can reliably be detected at a constant downsampling factor of 100. This result is in accordance to the detection probability results given in Figure 7.

DSF=80, SNR [dB]	Triangular pulse [dB]	Rectangular pulse [dB]	Gaussian pulse [dB]
-10	0.15	0.10	0.00
-5	0.06	0.00	0.05
0	0.45	1.45	0.00
5	0.00	2.70	2.51
10	0.00	0.45	5.92
15	0.00	7.85	5.72
20	0.00	16.17	5.26
25	0.26	12.24	0.00
30	0.00	22.93	17.61
35	0.00	25.17	22.47
40	0.00	33.46	32.72
45	0.00	32.20	36.55
50	0.00	38.97	39.25

Table 2. The values of the TH_i, $i = 1, 2, 3$ for different SNR when a Triangular pulse should be detected

Pulse Simulation

In the following simulation, three different kind of shaped pulses (Triangular, Rectangular, Gaussian), all with the same signal power in relation to a 100 samples long, normalized rectangular pulse were tested. The different pulses are depicted in Figure 8 in their time and frequency domain. In the simulation their starting position was uniformly randomly chosen within the 1000 acquired samples n. As with the chirps, the brute force reconstruction approach is used here.

Looking at Figure 9 it can be seen that the pulse recovery does not perform as excellent as for the chirp signal. The results for Triangular, Rectangular pulses are similar and therefore not shown here. It is concluded that the success of the recovery does not only depend on the SNR and the downsampling factor, but also on the amount of the signal information used. For downsampling factors up to 100 and for SNRs above 20 dB the detection probability is very good.

To determine the possible threshold that would make it possible to distinguish between different pulses, the three main pulses shown in Figure 8 were used.

As was done for the chirp signals, the brute force approach tests all possible solutions for each type of cataloged pulse. The TH distance between the best match for each type of pulse is calculated as explained in Equation 14.

The Tables 2, 3, and 4 show the detection performance when Triangular, Rectangular, and Gaussian shaped pulses should be detected. A constant downsampling factor of 80 was used while the SNR was varied. Only one simulation run was used.

The conclusion based on Tables 2, 3 and 4 is that with a threshold TH of 5 dB, different kinds of pulses with SNR

DSF=80, SNR [dB]	Triangular pulse [dB]	Rectangular pulse [dB]	Gaussian pulse [dB]
-10	0.00	0.00	0.12
-5	0.26	0.00	0.11
0	0.34	0.00	0.53
5	2.82	0.00	1.97
10	4.54	0.00	8.01
15	1.43	0.00	9.28
20	9.53	0.00	13.08
25	16.25	0.00	18.86
30	15.54	0.00	19.74
35	25.11	0.00	28.54
40	30.09	0.00	35.09
45	29.93	0.00	33.41
50	41.64	0.00	49.97

Table 3. The values of the TH_i, $i = 1, 2, 3$ for different SNR when the Rectangular Pulse should be detected

DSF=80, SNR [dB]	Triangular pulse [dB]	Rectangular pulse [dB]	Gaussian pulse [dB]
-10	0.31	0.44	0.00
-5	0.20	0.00	0.70
0	1.54	1.93	0.00
5	1.21	8.39	0.00
10	9.17	12.81	0.00
15	2.91	10.73	0.00
20	6.85	8.99	0.00
25	16.88	20.43	0.00
30	14.39	17.96	0.00
35	16.93	23.44	0.00
40	24.79	39.22	0.00
45	43.25	47.39	0.00
50	46.05	46.56	0.00

Table 4. The values of the TH_i, $i = 1, 2, 3$ for different SNR when the Gaussian Pulse should be detected

values above 20 dB can reliably be detected at a downsampling factor of 80 with only a few exceptions.

CONCLUSION

In this paper a new method for interference detection of previously characterized interferences was presented. It can be used to efficiently monitor e.g. the up to 100 MHz wide GNSS bands without having to use a Nyquist sampling receiver which would have very high data rates and high raw data storage requirements. Instead a compressed sensing random demodulator analog to information converter (AIC) architecture was proposed. Its key element is the random demodulator which operates at the signal band's Nyquist rate. The random demodulator spreads the signal's characteristic signature over a wide frequency band. A low-rate ADC is now sufficient to sample a small portion of the signature.

Using the knowledge about the interference detection receiver's architecture and the well characterized interference signals to be detected, it is possible to reconstruct the original signal out of very few samples. The key element of a successful recovery is the setup of an interference characterization database where all interferences to be detected are filed. The construction of the required representation matrices for the interfere types of CW, Chirps, and Pulses was explained in detail.

For the reconstruction of CW signals, a CoSaMP algorithm was used and was shown to be insensitive to the CW frequency and amplitude. For the reconstruction of chirps and pulses a very fast brute force approach was used. The brute force method uses the a priori knowledge of the interferer and its periodic nature. It is insensitive to the interferer appearance in time domain, but can only reconstruct a normalized version of the signal.

Monte Carlo simulations were carried out to investigate the detection probability for varying downsampling factors and signal-to-noise ratios. Additional simulations were performed to quantify the algorithm's ability to distinguish between several interferences belonging to the same family but having different parameters.

The simulation results show excellent detection rates and also demonstrate the possibility to distinguish between several interferences with a margin larger than 5 dB for

- Tones: for SNR higher than 10 dB and DSF of 80
- Chirps: for SNR higher than 10 dB and DSF of 200
- Pulses: for SNR higher than 20 dB and DSF of 80

To conclude, with this AIC approach a downsampling factor of 80 provides a reliable high detection probability if the SNR is larger than 20 dB for all cases. In other words, instead of using an I/Q ADC with sampling rates of 100 MHz, a low-rate I/Q ADC with 1.25 MHz is sufficient. The snapshot size and storage requirements decrease accordingly by a factor of 80.

Further work will be to implement a basic AIC hardware demonstrator setup. To this end, it will be necessary to investigate the impact of "real hardware" parameter such as quantization or mismatch between the AIC implementation and how it is modeled in the signal reconstruction.

REFERENCES

[1] A. G. Proctor, C. W. T. Curry, J. Tong, R. Watson, M. Greaves, and P. O. S. Cruddace, "Protecting the UK Infrastructure - A System to Detect GNSS Jamming and Interference," *insideGNSS*, vol. September/October, pp. 49 – 57, 2011.

[2] R. Bauernfeind, A. Sicramaz Ayaz, and B. Eissfeller, "GNSS Interference Monitoring Network Based on Detection in Automotive ITS Station Receivers," in *Proceedings of the 24th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS 2011)*, 2011.

[3] C.-C. Sun and S.-S. Jan, "Interference characteristics for the civil airport environment using time-frequency analysis," in *Proceedings of IEEE/ION PLANS 2012, Myrtle Beach, South Carolina , April 2012, pp. 347-354*, 2012.

[4] J. Wendel, C. Kurzhals, M. Houdek, and J. Samson, "An interference monitoring system for GNSS reference stations," in *Antenna Technology and Applied Electromagnetics (ANTEM), 2012 15th International Symposium on*, pp. 1 –5, june 2012.

[5] R. H. Mitch, R. C. Dougherty, M. L. Psiaki, S. P. Powell, B. W. O'Hanlon, J. A. Bhatti, and T. E. Humphreys, "Signal characteristics of civil GPS jammers," in 24*th International Technical Meeting of the Satellite Division of The Institute of Navigation, Portland OR*, 2011.

[6] J. Bhatti, T. Humphreys, and B. Ledvina, "Development and Demonstration of a TDOA-Based GNSS Interference Signal Localization System," in *Proceedings of IEEE/ION PLANS 2012, Myrtle Beach, South Carolina , April 2012, pp. 455-469*, 2012.

[7] J. Tropp, J. Laska, M. Duarte, J. Romberg, and R. Baraniuk, "Beyond nyquist: Efficient sampling of sparse bandlimited signals," *Information Theory, IEEE Transactions on*, vol. 56, pp. 520 –544, jan. 2010.

[8] E. J. Candès and M. B. Wakin, "An Introduction to Compressive Sampling," *IEEE signal processing magazine*, pp. 21 – 30, March 2008.

[9] E. J. Candès, "Compressive sampling," in *Proceedings of the International Congress of Mathematicians, Madrid, Spain, 2006*, 2006.

[10] E. Candès and J. Romberg, "l_1-MAGIC. Recovery of sparse signals via convex programming," October 2005.

[11] Y. Pati, R. Rezaiifar, and P. Krishnaprasad, "Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition," in *Signals, Systems and Computers, 1993. 1993 Conference Record of The Twenty-Seventh Asilomar Conference on*, pp. 40 –44 vol.1, nov 1993.

[12] D. Needell and R. Vershynin, "Uniform Uncertainty Principle and Signal Recovery via Regularized Orthogonal Matching Pursuit," *Found. Comput. Math.*, vol. 9, pp. 317–334, Apr. 2009.

[13] D. Donoho, Y. Tsaig, I. Drori, and J. Starck, "Sparse solution of undetermined systems of linear equations by stagewise orthogonal matching pursuit," *IEEE Transactions on information theory*, vol. 58, pp. 1094 – 1121, Februar 2012.

[14] D. Needell and J. A. Tropp, "CoSaMP: iterative signal recovery from incomplete and inaccurate samples," *Commun. ACM*, vol. 53, pp. 93–100, Dec. 2010.

[15] D. Needell, "Topics in Compressed Sensing," 2009.

[16] T. Kraus, R. Bauernfeind, and B. Eissfeller, "Survey of In-Car Jammers - analysis and modeling of the RF signals and IF samples (suitable for active signal cancellation)," in 24*th International Technical Meeting of the Satellite Division of The Institute of Navigation, Portland OR*, 2011.

[17] J. Romberg, "Compressive sensing by random convolution," *SIAM J. Img. Sci.*, vol. 2, pp. 1098–1128, Nov. 2009.

[18] W. U. Bajwa, J. D. Haupt, G. M. Raz, S. J. Wright, and R. D. Nowak, "Toeplitz-Structured Compressed Sensing Matrices," in *Proceedings of the 2007 IEEE/SP 14th Workshop on Statistical Signal Processing*, SSP '07, (Washington, DC, USA), pp. 294–298, IEEE Computer Society, 2007.