Quasi Real-Time Intermodulation Interference Method: Analysis and Performance

Mohammad Malkawi, Khaldoon Al-Zoubi, and Ahmed S. Shatnawi

Jordan University of Science and Technology (Software Engineering), Jordan

***Abstract*:** Establishing interference-free wireless networks has become essential requirement with the advent of 5G networks as wireless carriers becoming eager to start transmitting high volume of voice and data to meet enterprise demands and consumers for dispersed information. Indeed, reduction of interference level now emerges as one of the most viable solutions for the service provider networks, as 5G networks use is increasingly on the rise. Therefore, a comprehensive interference solution is needed to enable service providers (e.g., cellular, PCS, Wi-Fi and Broadband, and LTE wireless LAN/WAN services) to identify and resolve network interference quickly and reliably. However, currently existing methods and tools, used to conduct interference detection and analysis lack the necessary performance required in the RF engineering and spectrum optimization fields, and hence, fast, and reliable networks. This paper presents a new high-performance method for detecting and mitigating intermodulation interference in various wireless networks. Clearly, reducing the level of interference in wireless communication contributes directly to having an improved and reliable signal to noise ratio (C/I). We will discuss the complexity of the intermodulation interference problem. We will further compare the performance of the new approach with different existing interference detection methods. We will show that the presented method intermodulation complexity is reduced to a near linear one, compared to the current non-linear methods.

***Keywords*:** Intermodulation, Interference, Complexity analysis, Wireless communication

1. Introduction

As wireless carriers begin transmitting voice and data over the network to meet enterprise customers and consumers' demands for dispersed information, creating an interference-free network has become imperative. The advancement in wireless technology manifested by the introduction of 5G networks can be hindered by relative increase in interference due to intermodulation. To meet customer requirements, providers of cellular, PCS, broadband and WLAN (Wireless LAN) services need a comprehensive interference solution that enables them to identify and resolve interference quickly and reliably.
Voice quality and reliable high data transmission rates are dependent on Carrier/Interference (C/I) ratio and Received Signal Strength Indicator (RSSI) [3][6][7]. It is common for an "adequate" signal to be present and still have poor voice quality and low data transmission rates. When this occurs, it is often due to the presence of external intermodulation interference, resulting in reduced C/I. The C/I ratio must be at least 17 dB for quality voice signal [1][6][8][9][10].
It is a competitive necessity that wireless service providers deliver reliable, high-speed data if they are to make a successful foray into the enterprise market, fast, reliable access to information and communication is mission critical. Delivery of higher data transmission rates requires the use of either a wider spectrum or a different modulation and coding scheme and packet data techniques. While the former is expensive and not feasible for many service providers, the latter proves to be a more realistic and cost-effective solution. In fact, carriers, e.g., T-Mobile and others, are embracing the latter approach [2][3] to offer high data rate solutions. To transmit at higher modulation or coding scheme 1024 to 4096 QAM, the C/I must be strong (stronger than required for GPRS, Wi-Fi/Broadband or GSM) [4]. Given the fact that most wireless networks are up-link limited, the C/I ratio can be increased by building more sites, reducing the interference level, or both.
For example, the buildup of new sites is expensive and often presents new challenges. Specifically, additional sites result in more signals contributing to a rise in the spectrum noise floor and limiting frequency re-use in FDMA (Frequency Division Multiple Access) and TDMA (Time Division Multiple Access) and CDMA (Code Division Multiple Access) systems [11][14][15]. Consequently, the reduction of interference level merges as one of the most viable solutions for service provider networks, especially with the dramatic increase in the number of service providers in given locality. Current methods and tools, which are used to conduct interference detection and analysis lack the necessary performance required in the various RF fields. Consequently, new methods and tools for interference detection, analysis, and resolution are necessary for the purpose of spectrum optimization, and hence, fast, and reliable networks.
As new wireless network technologies emerge and the number of transmitters increases, there is a corresponding increase in external interference. Wi-Fi and WLAN is a technology with a growing presence in the high-data rate space. Wi-Fi not only poses a threat to service providers' market share, but it also threatens the very quality of voice and data throughput on providers' networks, due to significant interference with existing carrier signals.
WLAN, especially within the 802.11(x) spectrum (~ 2.5 GHz) faces serious interference challenges [16][20]. Another area where interference is highly sighted as a potential safety and quality problem is the medical equipment RF environment [13, 17, 18, 19].
RF interference can occur in any radio system. Interference depends on interactions within the systems that are dependent on the system equipment, system design parameters, and site antenna configurations [24]. One of the significant sources of interference is intermodulation [5]. For a single system installation, the interference analysis and control are comparatively simple since site equipment, and antenna configurations can be restricted to tried and tested setups. For multiple collocated systems, the scenarios are too complex to predict [24]. Besides providing broad guidelines for collocation, there is no option but to perform the comprehensive interference (intermodulation) studies on a site-by-site basis. As the number of collocated sites

increases, there is a growing need to understand and analyze all the issues in a unified manner. The main obstacle for this approach is the complexity of the analysis, which grows exponentially with the signals' growth that could cause interference to occur. Smith [24] recently reported that with the addition of two carrier channels to the existing 4 channels, the number of intermodulation products rose from 25 to 240. In the wireless industry, adding one carrier to a service-providing tower may introduce more than 300 frequency channels. For example, to appreciate the computational complexity involved in the detection and analysis of intermodulation product interference considers the following example. Assume a tower with four different carriers. The transmission spectrum of each carrier is 10 MHz with 30 KHz channel bandwidth. This configuration results in ~ 1300 transmission signals. Assume further that each carrier has one spectrum range for received signals. If we only consider 3rd order intermodulation, then there will be *5*[(1300*1299*1298)/6] = 1,826,610,500* different intermodulation products of the type that could interfere with 4 receiving frequency ranges. The time and space required to perform the search for all products that cause interference has been prohibitive. The current tools that exist in the market, such as Comsite Pro® and TAP/SoftWright®2, take more than 24 hours to complete such a task. The space requirement to solve such a problem is in excess of 8 GB of hard disk space (only to store the different product combinations).

The time and space complexity, until now, have been the main cause for not pursuing interference resolutions vigorously by service providers as well as by governmental agencies. Drop calls due to interference continues to be a hard to pursue issue, mainly because of the cost associated with detection and analysis. Note that in order to bypass complex computations, service providers need to spend expensive engineering hours for troubleshooting interference issues.

It is estimated that the annual loss of revenue and engineering cost due to unresolved interference is in excess of $400,000 ($300,000 lost revenue due to dropped calls and $100,000 is the average engineering cost for troubleshooting).

Assuming that only 4% of the total number of sites (there are more than 120,000 sites in the US alone) experience problems due to unresolved interference, the service providers' loss is close to $2 billion. This figure is expected to grow much larger as more sites tend to neighbor one another, more collocation occurs on the same site, and more cellular penetration is achieved. The need to address interference analysis in a timely manner is motivated by several factors:

1. The number of service providers continues to grow, due to the increased spectrum licenses auctioned worldwide.
2. Tower owners attempt to collocate more carriers on the same tower, for increased revenue. Some towers are already hosting 12 carriers. This phenomenon leads to more interference problems.
3. Tower farms are becoming more common, especially when service providers strive for more coverage along major highways as well as in highly populated areas.
4. Interference due to intermodulation can cause serious security issues, especially at the time of crisis. Any interference with government related signals during crisis can cause tremendous impact on security.
5. Medical institutions, by and large, continue to be worried that radio interference may jeopardize the safety of people. Lack of comprehensive interference analysis justifies this type of fear.
6. The growing utilization and use of wireless LAN (WLAN) bring interference problems to the small offices and homes.
7. The high cost associated with unresolved interference problems will contribute to the cost of cellular service for the end users.

With the current technology for intermodulation interference analysis, which is characterized by low performance, it is almost impossible to address the current market needs and requirements.

Consequently, new methods and tools for interference detection, analysis, and resolution are necessary for the purpose of cost reduction, fast and reliable networks, and safe and secure environment. In contrast some other existing work like in [28] used clipping and filtering methods to minimize the effect of peak-to-average power ratio.

This paper introduces a new intermodulation product interference detection and analysis tool, Intermod60. The main power of this tool is the speed at which it can analyze and detect all possible intermodulation products that could cause interference. While the complexity of intermodulation grows in a non-linear fashion, the performance of Intermod60® exhibits linear behavior.

This paper is organized as follows: In Section 2, we discussed the motivation behind developing the presented tool and we further discuss the background and a brief overview of the intermodulation problem. Complexity analysis of intermodulation product detection and the novel proposed intermodulation interference detection algorithm will be provided in Section 3. Section 4 provides real examples, which demonstrates the functionality of the presented tool. Section 5. provides performance evaluation results of the Inermod60 (i.e., which applies the new presented methods here). Concluding remarks and future enhancements will be discussed in section 6.

2. Motivation and Background

The idea of developing Intermod60 was motivated by RF engineers working in network optimization for more than 10 years. An RF engineer would spend many hours and quite often days in order to diagnose an external interference related problem.

In some cases, the desktop computer would crash, due to resource exhaustion (CPU, RAM and disk space). The obvious conclusion was that the industry lacks a high-performance interference analysis tool. The inspection of currently existing tools in the market revealed that the implementation of the SW tools reflects the exact nature of intermodulation interference complexity.

Unfortunately, the complexity of intermodulation product analysis has a non-linear (exponential) growth pattern. As such, the SW tools inherited the non-linear behavior of the intermodulation product analysis complexity.

The idea was to look for a different way of representing the intermodulation paradigm. This is a real application software architecture problem. The SW architect, who is specialized in high performance and availability systems [25][26][27], investigated the architecture underlying the intermodulation problem. The two main objectives of the architecture of

Intermod60 were to reduce the time complexity, and to reduce the memory requirements of the SW system. The result of the investigation was the introduction of new algorithms for the detection of intermodulation products.

The algorithms explored some of the mathematical properties of intermodulation products formation. Consequently, the search algorithms were greatly amplified. The complexity of the 2nd order intermodulation product detection was reduced from $O(N^2)$ to O(N.Log(N). The complexity of the 3rd order intermodulation product election was reduced from $O(N^3)$ to O(N2.Log(N). The memory requirement of the algorithms during the analysis phase was reduced to O(N) for either 2nd or 3rd order. The core engine of Intermod60 is primarily based on these algorithms. Performance evaluation of Intermod60 will be presented in section (5).

Intermodulation distortion (IMD) is the term given to the phenomenon by which signals present in a non-linear device combine to create new signals [3][7][12][23]. The device is fed with two sinusoidal input signals (eqn.1) and the output is given by the device's transfer characteristic (eqn 1).

$$Vi = A_1 \cos(2\pi f_1 t) + A_2 \cos(2\pi f_2 t) \quad (1)$$

$$v_0(t) = a_0 + a_1 v_i(t) + a_2 v_i^2 + a_3 v_i^3(t) + \ldots \quad (2)$$

Where, v_o is output signal, v_i = input signal, a_0, a_1, a_2, a_3 = constants, and $v_i = v_1 + v_2$

The first two terms of the transfer characteristic represent the linear response while the third and higher terms represent the nonlinear response of the device. These terms contain the distortion products and are referred to as the nth-order intermodulation products. For example, the second-order products of the square-law term, a2vi2, consist of a dc term, second harmonics of the input frequencies and intermodulation products:

$$a_2 v_i^2(t) = a_2 [A_1 \cos(2\pi f_1 t) + A_2 \cos(2\pi f_2 t)]^2$$
$$= (\frac{a_2}{2})(A_1^2 + A_2^2) + (\frac{a_2}{2})[A_1^2 \cos(4\pi f_1 t) + A_2^2 \cos(4\pi f_2 t)]$$
$$+ a_2 A_1 A_2 \{\cos[2\pi(f_1 + f_2)t] + \cos[2\pi(f_1 - f_2)t]\} \quad (3)$$

Similarly, the cube-law term gives rise to the third-order products including third-order intermodulation (IM) products, and so on. Second and third order intermodulation products significantly increase when the number of input signals increases. Table 1 summarizes the second, third order, and second order harmonic intermodulation products.

Table 1. Intermodulation Products

2nd-Order IM Products	Frequency	Amplitude
	$f_1 + f_2$	$a_2 A_1 A_2$
	$f_1 - f_2$	$a_2 A_1 A_2$
3rd-Order Harmonics	$2f_1 + f_2$	$(0.75)a_3 A_1^2 A_2$
	$2f_1 - f_2$	$(0.75)a_3 A_1^2 A_2$
	$2f_2 + f_1$	$(0.75)a_3 A_1 A_2^2$
	$2f_2 - f_1$	$(0.75)a_3 A_1 A_2^2$
3^{rd} Order IM Products	f1+f2+f3	
	f1+f2-f3	$(0.75)a_3 A_1^2 A_2$
	f1-f2+f3	$(0.75)a_3 A_1^2 A_2$
	-f1+f2+f3	$(0.75)a_3 A_1^2 A_2$
	-f1-f2+f3	$(0.75)a_3 A_1^2 A_2$

The 2nd order harmonics and third-order IM products are the most severe in terms of interference and therefore the most analyzed. 4th order and higher intermodulation products have less impact in terms of practical interference, especially when the power of the formed intermodulation product is considered [5]. Assuming that the fundamental signal magnitudes are equal it can be shown that for each dB change in the fundamental signal level the third-order IM level will change by 3 dB. Intermodulation Rejection (IMR) is the difference between the fundamental and IM signal levels, whereas the intercept point (IP), is the point at which the IM level is equal to the fundamental level. It is a measure of the device's non-linearity. Actual device operation does not extend to the intercept point but is limited to the small-signal region below the 1 dB compression point and well below the intercept point.

For this paper's purposes, it suffices that IM signals are generated due to mixing two or more signals in a non-linear operation. The multiple transmit signals at a site can leak into each other's paths and mix in the base station hardware. The frequencies and power levels of the IM products are a function of the mixing signals and the transfer characteristics' hardware. IM generation can occur at any of several points along the transmit and receive signal paths. However, keeping in mind the scope of this study, the primary sources of IM are the transmit amplifiers, receiver front ends, and antennas. IM frequencies that fall within a collocated receiver's in-band can severely affect the receiver's sensitivity. Reducing IM interference requires controlling the level of the mixing signals and their frequencies. Evidently, to reduce the IM level, the IM with different orders (up to 4th order) should be detected and properly identified.

3. Intermod60: Intermodulation Tool

3.1 Structure

The structure of Intermod60 is shown in Figure 1. The core engine of Intermod60 implements the intermodulation algorithms. The engine is designed to be independent of the input or output format, for maximum flexibility. The user interface is designed in a modular manner to allow for all types of interference sources to be analyzed. Currently, the SW tool implements four types of interference sources, namely single tower, multiple towers, rooftop6, and measured field data.

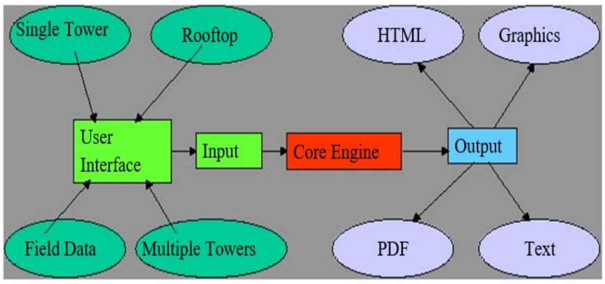

Figure 1. Intermod60 Structure Overview

The single tower, multiple towers, and rooftop cases are used to carry exhaustive search for intermodulation products, used mostly for pre-planning and collocation analysis. The measured field data input allows for data gathered by frequency analyzers to be analyzed for intermodulation hits. Note that in the case of multiple towers and rooftops, the number of collocated carriers can be very large. Consequently, the number of all possible transmitting frequency channels can be very large. Currently, the only tool in the market capable of processing all four input variations is Intermod60. The main reason for this is the time complexity involved in executing rooftop and multiple tower data.

The structure of Intermod60 allows for multiple input sources to be analyzed without having to change or modify the core engine. Similarly, the output format is independent of either the input or the core engine. Currently, Intermod60 generates output in PDF, HTML, graphics and/or text formats.

The Intermod engine carries all the necessary computation and generates a generic output to the output subsystem, which in turns generates the output reports.

Intermod60 engine analyzes both transmitter and receiver generated IM products. Specifically, it analyzes worse case scenarios by calculating the IM products at the receiver antenna, examining each channel licensed to the carriers in the area. In addition to IM, AIM Intermod60 analyzes transmitter noise, receiver desensitization noise, spurious noise and transmitter harmonics noise.

The modular design of Intermod60 provides the user with flexibility and ease of use. It also provides the software engineers flexibility in the modification, maintenance, and upgrade of the system. It is worthwhile noting that Intermod60 is web-enabled tool. In other words, interference analysis can be conducted off the website. This feature is made possible because of the high speed of the tool and the low memory requirements. Multiple studies can be conducted simultaneously on the same server.

3.2 Algorithms and Analysis

One of the most serious challenges in the area of intermodulation product interference analysis is the time and space complexity required to perform intermodulation analysis. To avoid massive computational complexity, several studies suggested the use of analytical techniques [5] or massively parallel processing machines [21]. The first approach is not efficient when the number of IM products is relatively large, which is typical in the case of tower farms. The use of massive parallel process systems is an expensive solution. This study presents a new method used for reducing the complexity of IM analysis.

Next, we present the complexity analysis for intermodulation product interference. The complexity of typical intermodulation detection algorithms, commonly used in current software technology, exhibits nonlinear complexity behavior. The complexity analysis of a typical IM detection algorithm is given below.

For the purpose of generality, let the number of frequency transmitters be (G), and each transmitter be marked as Txi. Let each Txi generate Fi different frequency channels. The total number of frequency channels is given by:

$$N = \sum_{i=1}^{G} F_i$$

Assume that the N distinct frequencies (f_i) are sorted in an ascending order in an array like data structure (A). Assume further that there are (K) different receiving frequency spectrum ranges (Rx). Each Rx is characterized by a minimum (R_min) and a maximum (R_max).

Next, we define the problem and complexity for second order and third order intermodulation schemes.

3.2.1 Second Order Frequency Modulation

Assumptions:

The array size is N. A(1 .. N)

The values in A are sorted in ascending order, i.e., A(I+1) > A (I)

Requirement:

Given a range of values [R1, R2], where R2>R1, find all double combinations {A(i), A(j)} in array A, such that

R1<={A(i) + A(j)}<=R2

R1 <= {-A(i)+A(j)} <= R2

Solution for Requirement:

Find the set of values S1: {A(i), A(j)} such that [A(i) + A(j)] ≥ R1;

Find the set of values S2: {A(k), A(L)} such that [A(k) + A(L)] ≤ R2;

Find the set of values S = S1 ∩ S2

Mark all entries in the set S.

ALGORITHM-1:

Algorithm-1 is used to find the set S1.

Complexity Analysis:

The maximum number of steps required to locate all combinations {A(i), A(j)} such that A(i)+A(j) ≥ R1 is computed as follows:

Log (n-1) + log (n-2) + log(n-k) … + log (2)

ALGORITHM-2

Algorithm-2 is used to find the set S2.

Complexity Analysis:

The maximum number of steps required to locate all combinations {A(i), A(j)} such that A(i)+A(j) ≤ R2 is computed as follows:

Log (n-1) + log (n-2) + log(n-k) … + log (2)

Discussion:

We will illustrate the use of Algorithms-1 and 2 through an example.

Let the array size N=10.

For the add operation {A(i) + A(j)} there are a total of (n!)/(n-2)!*2! = 45 combinations.

Let the range of values [Ri, Rj] = [9,13].

It is required to find all combinations {A(i),A(j)} such that {9≤ A(i)+A(j) ≤ 13}. I will call this set the Hit Set (HS)

The Hit Set (HS) is found in two steps.

First: Find all combinations A(i),A(j) such that A(i)+A(j) ≥ 9. Call this set S1

Second: Find all combinations A(k),A(L) such that A(k)+A(L) ≤ 13. Call this set S2

The hit set HS is given as HS = S1 ∩ S2.

For simplicity, assume that the values stored in A are 1, 2, 3, 4, 5, 6, 7, 8, 9 and 10.

The combinations (A(i),A(j)) and their corresponding values are shown in Table 2.

Note that we sorted the combinations such that the combinations form a lower left part of a 2-dimensional array. Also, note that each array is sorted in an ascending order. This is expected because initially the elements of array A are sorted in an ascending order. Hence, for a given I, A(i)+A(j) < A(i)+A(j+k) for k ≥ 1 (follows from the fact that A(j+k) > A(j)

Table 2. combinations (A(i),A(j)) and their corresponding values

A(I,J)	I=1	I=2	I=3	I=4	I=5	I=6	I=7	I=8	I=9
J=1	A(I)+ A(j)								
J=2	(1,2)= 3								
J=3	(1,3)= 4	(2,3)= 5							
J=4	(1,4)= 5	(2,4)= 6	(3,4)= 7						
J=5	(1,5)= 6	(2,5)= 7	(3,5)= 8	(4,5)= 9					
J=6	(1,6)= 7	(2,6)= 8	(3,6)= 9	(4,6)= 10	(5,6)= 11				
J=7	(1,7)= 8	(2,7)= 9	(3,7)= 10	(4,7)= 11	(5,7)= 12	(6,7)= 13			
J=8	(1,8)= 9	(2,8)= 10	(3,8)= 11	(4,8)= 12	(5,8)= 13	(6,8)= 14	(7,8)= 15		
J=9	(1,9)= 10	(2,9)= 11	(3,9)= 12	(4,9)= 13	(5,9)= 14	(6,9)= 15	(7,9)= 16	(8,9)= 17	
J=10	(1,10) =11	(2,10) =12	(3,10) =13	(4,10) =14	(5,10) =15	(6,10) =16	(7,10) =17	(8,10) =18	(9,10) =19

Find Set S1:

Begin with I=1

The smallest value in the column is 3 < 9. The largest value in the column is 11 > 9. Therefore, column I=1 contains values that are larger than 9.

Use binary search to find the smallest value in column I=1, such that the value is ≥ 9.

By inspecting column I=1, we can see that the 8th element A(1)+A(8) = 9.

The binary search steps are (given that A(1), A(n) are not equal to 9):

Find the middle value: (10+2)/2 = 6; A(1)+A(6) = 7;

The 6th element value is less than 9 (7 < 9).

Go to the upper half of the array.

Find element (6+10)/2 = 8; A(1)+A(8) = 9

Stop the search; the lower range (9) is located in column 1;

Mark all values in the column with index A(1) and A(j) where j= 8, 9, and 10.

S1 = {A(1,8), A(1,9), A(1,10).

Increment I=I+1 = 2 (move to column 2)

Repeat same steps

Locate A(2), A(7) where 2+7 = 9;

Mark A(2), A(7), A(8), A(9), A(10)

S1 = { A(1,8), A(1,9), A(1,10), A(2,7), A(2,8), A(2,9), A(2,10)}

Increment I=I+1 = 3 (move to column 3)

Repeat same steps

Locate A(3), A(6) where 3+6 = 9;

Mark A(3), A(6), A(7),A(8), A(9), A(10)

S1 = { A(1,8), A(1,9), A(1,10), A(2,7), A(2,8), A(2,9), A(2,10), A(3,6), A(3,7), A(3,8), A(3,9), A(3,10)}

Increment I=I+1 = 4 (move to column 4)

Locate A(4), A(5) where 4+5 = 9;

Note that A(4), A(5) is the first element in the column

Also note that all elements in column 4, column 5, 6, 7, 8, and 9 are larger than 9

Mark A(4,5), A(4,6), … A(4,10), A(5,j : 6 .. 10), A(6,j: 7 .. 10), A(7,j: 8 .. 10), A(8,j: 9..10), A(9,j: 10)

S1 = { A(1,8), A(1,9), A(1,10), A(2,7), A(2,8), A(2,9), A(2,10), A(4,5), A(4,6), … A(4,10), A(5,j : 6 .. 10), A(6,j: 7 .. 10), A(7,j: 8 .. 10), A(8,j: 9..10), A(9,j: 10)}

Find Set S2:

Begin with I=1

A(1)+A(2) = 3 < 13

A(1)+A(10) = 11 < 13

Mark all entries in column I=1 {A(1,2), … A(1,10)

S2 = {A(1,2), … A(1,10)}

Increment I=I+1 = 2 (move to column I=2)

A(2)+A(2) = 5 < 13

A(2)+A(10) = 12 < 13

Mark all entries in column I=2 {A(2,3), … A(2,10)

S2 = {A(1,2), … A(1,10), A(2,3), … A(2,10)}

Increment I=I+1 = 3 (move to column I=3)

A(3)+A(4) = 7< 13

A(3)+A(10) = 13 < 13

Mark all entries in column I=3 {A(3,4), … A(3,10)

S2 = {A(1,2), … A(1,10), A(2,3), … A(2,10), A(3,4), … A(3,10)}

Increment I=I+1 = 4 (move to column I=4)

A(4)+A(5) = 9< 13

A(4)+A(10) = 14 > 13

Use binary search to located element A(4),A(9) (4+9=13)

Element A(4), A(9) is found in a maximum of 3 steps.

Mark the elements in column I=4 {A(4,5), … A(4,9)}

S2 = {A(1,2), … A(1,10), A(2,3), … A(2,10), A(3,4), … A(3,10), A(4,5) … A(4,9)}

Increment I=I+1 = 5(move to column I=5)

A(5)+A(6) = 11< 13

A(5)+A(10) = 15 > 13

Use binary search to located element A(5),A(8) (5+8=13)

Element A(5), A(8) is found in a maximum of 3 steps.

Mark the elements in column I=4 {A(5,6), … A(5,8)}

S2 = {A(1,2), … A(1,10), A(2,3), … A(2,10), A(3,4), … A(3,10), A(4,5) … A(4,9), A(5,6), … A(5,8)}

Increment I=I+1 = 6(move to column I=6)

A(6)+A(7) = 11< 13

Note that all elements in column 6 are larger than A(6), A(7)

Also, all elements in columns I=7, 8, 9, and 10 are larger than A(6), A(7).

Mark element A(6), A(7)

S2 = {A(1,2), … A(1,10), A(2,3), … A(2,10), A(3,4), … A(3,10), A(4,5) … A(4,9), A(5,6), … A(5,8), A(6,7)}

Stop

The Hit Set HS = S1 ∩ S2 = {(1,8), (1,9), (1,10), (2,7), (2,8), (2,9), (2,10), (3,6), (3,7), (3,8), (3,9), (3,10), (4,5), (4,6), (4,7), (4,8), (4,9), (5,6), (5,7), (5,8), (6,7)}

Implementation Issues:

The table of the combinations need not be generated physically. The code only must deal with the original array A. The set S1 and S2 can be very large; therefore, it is recommended to describe the sets in terms of indexes. It is more difficult this way (requires more processing) but should save memory.

The reason we want to avoid excessive use of memory is because we want to maximize the cache-hit ratio.

Solution for Requirement 2: {-A(i) + A(j)}. Same as for Requirement 2. The only difference is that the values of the array elements will be different. But the algorithms and the procedures are the same. The complexity of the algorithms for requirement 3 is similar to the algorithms for requirement 2.

3.2.2 Third Order Intermodulation

Third order interference modulation computes the superposition of three different frequency values provided by one or more carriers.

Assumptions:

Each carrier provides a range of frequency values using a minimum , a maximum and an increment.

All frequency values are stored in a one-dimensional array A, with size N.
All values in array A are distinct and shall be sorted in ascending order.
The superposition of the values in array A allow addition and subtraction to the values in the following manner:
Ai+Aj+Ak
Ai+Aj -Ak
Ai -Aj+AK
Ai-Aj- Ak (not used since this is a negative value)
–Ai +AJ +Ak
–Ai+ Aj-Ak (not used since this is a negative value)
–Ai -Aj+Ak
–Ai-AJ-Ak (not used since this is a negative value)

Corollary:
For each superposition formula, the number of combination (Ai, Aj, AK) {i≠j≠k} is given by (N!)/(N-3)!*3!. For N = 10, the number of combinations is 120
The combinations (Ai, Aj, Ak) are generated in the following order. All combinations with the same index (i) and the same index (j) are assumed to be stored sequentially in one dimensional array {N>j>i) and k varies from j+1 to N
Example: Let N=5; the combinations order is shown in Tables 3. The values in each column of tables 3 are sorted as follows:
Ai+Aj+Ak: ascending order
Ai+Aj-Ak: descending order
Ai-Aj+Ak: ascending order
-Ai+Aj+Ak: ascending order
-Aj-Aj+Ak: ascending order

Table 3. 3rd Order Combinations for an Array

I=1	J=2	J=3	J=4
	A(I,J,K)		
	1,2,3		
	1,2,4	1,3,4	
	1,2,5	1,3,5	1,4,5

A: I=1

I=2	J=3	J=4
	A(I,J,K)	
	2,3,4	
	2,3,5	2,4,5
	1,2,5	1,3,5

B: I=2

I=3	J=4
	A(I,J,K)
	3,4,5

C: I=3

Table 4. The values for the case of A(I)+A(J)+A(K)

I=1	J=2	J=3	J=4
	A(I)+A(J)+A(K)		
	1+2+3=6		
	1+2+4=7	1+3+4=8	
	1+2+5=8	1+3+5=9	1+4+5=10

A: I=1

I=2	J=3	J=4
	A(I)+A(J)+A(K)	
	2+3+4=9	
	2+3+5=10	2+4+5=11

B: I=2

I=3	J=4
	A(I)+A(J)+A(K)
	3+4+5=12

C: I=3

Note that the values in each of the columns in the tables 3 are stored in ascending order for cases 1, 3, 4, and 5. For case 2, the values are stored in descending order. For the sake of the example, assume that the values in array A are the same as the index, e.g., A(1)=1, A(2)=2, … A(5) = 5. For this case, the values of Tables (3) are shown in Tables 4 for case 1 {A(I)+A(J)+A(K)}.

Table 5. The values for the case of A(I)-A(J)+A(K)

I=1	J=2	J=3	J=4
	A(I)-A(J)+A(K)		
	1-2+3=2		
	1-2+4=3	1-3+4=2	
	1-2+5=4	1-3+5=3	1-4+5=2

A: I=1

I=2	J=3	J=4
	A(I)-A(J)+A(K)	
	2-3+4=3	
	2-3+5=4	2-4+5=3

B: I=2

I=3	J=4
	A(I)-A(J)+A(K)
	3-4+5=4

C: I=3

Each table in the set of Tables 3 will be referred to as a combination table. Each combination table has a unique i index. Each combination table is a diagonal table, i.e., only the elements along and below the diagonal have values.
The values for case 3 {A(I)-A(J)+A(K)} are shown in Tables 5.

Requirement:
Given a range of values [R1, R2], where R2>R1, find all triple combinations {A(i), A(j), A(K); i≠j≠k , N>j>I; k>j; } in array A, such that:
R1<=Ai + Aj + Ak <=R2
R1 <= Ai+Aj-Ak <= R2
R1 <= Ai-Aj+Ak <= R2
R1 <= -Ai+Aj+Ak<= R2
R1 <=-Aj-Aj+Ak<= R2

Solution for Requirement:
Find the set of values S1: {Ai,Aj,Ak} such that [Ai + Aj+Ak] ≥ R1;
Find the set of values S2: {Ak, Am,An} such that [Ak+Am+An] ≤ R2;
Find the set of values S = S1 ∩ S2
Mark all entries in the set S.

ALGORITHM-3:

Algorithm-3 is used to find the set S1.
Algorithm-3 is shown in Figure-3A and B in a flow chart format.

Complexity Analysis:
The maximum number of steps required to locate all combinations {A(i), A(j)} such that A(i)+A(j) ≥ R1 is computed as follows:
Log (n-1) + log (n-2) + log(n-k) … + log (2)

ALGORITHM-4:

Algorithm-4 is used to find the set S2.

Complexity Analysis:

The maximum number of steps required to locate all combinations {A(i), A(j)} such that A(i)+A(j) ≤ R2 is computed as follows:

Log (n-1) + log (n-2) + log(n-k) … + log (2)

Discussion on Algorithm-3:

Algorithm-3 searches for the following values.

The smallest value Vmin ≥ R1; All values larger than Vmin must be larger than R1 and marked as part of S1.

The largest value Vmax ≤ R1; All values less than Vmax must be less than R1 as well. These values are not part of S1 and will be ignored.

Once these two values (Vmin and Vmax) are located, Algorithm-3 will continue to search the combinations that exist between Vmin and Vmax in order to locate all values that fall within the range [R1-R2]

Algorithm-3: Vmin Location

In order to locate Vmin, Algorithm-3 makes the following observations.

The first element of each combination table is (Ai, Aj,Ak), where j=i+1 and k=j+1

The first element of each combination table has the smallest value in the table

The value of the first element in each table is smaller than the value of the first element in the next subsequent table, i.e. {A(i)+A(j)+A(k)} < {A(i+1)+A(j+1)+A(k+1)} [j=i+1; k=j+1]

Algorithm-3 searches for Vmin among all the values that constitute the first element in each of the tables. These values are given by i=1 .. N-2; j=i+1 and k=i+2. The total number of these values is N-2

Algorithm-3 uses Binary search over N-2 elements to locate Vmin

Algorithm-3: Vmax Location

In order to locate Vmax, Algorithm-3 makes the following observations.

The last element of each combination table is (Ai,Aj,Ak), where j=N-1 and k=N

The last element of each combination table has the largest value in the table.

The value of the last element in each table is smaller than the value of the last element in the next subsequent table, i.e. {A(i)+A(N1)+A(N)}<{A(i+1)+A(N-1)+A(N)}

Algorithm-3 searches for Vmax among all the values that constitute the last element in each of the tables. These values are given by i=1 .. N-2; j=N-1 and k=N. The total number of these values is N-2. Algorithm-3 uses Binary search over N-2 elements to locate Vmax.

Algorithm-3: Constructing S1 {(Ai+Aj+Ak) ≥ R1}

Find Vmin

Find Vmax

Ignore all combinations (Ai, Aj, Ak) < Vmax

Mark all combinations (Ar,Am,An) ≥ Vmin

Search the combinations between Vmin and Vmax

All the tables with index (I) that falls between (i+1) and (r-1) must be searched. Note that r>i; If r-i <2 stop the search

Each table with index I must be searched using the algorithm described in Algorithm-1.

Algorithm-3 is illustrated using the flow chart in Figure 2 (i.e., Figure 2-A locate V-min while Figure 2-B locates V-max).

Algorithm-4: Constructing S2 {(Ai+Aj+Ak) ≤ R2}

Find Vmin

Find Vmax

Ignore all combinations (Ai, Aj, Ak) > Vmin

Mark all combinations (Ar,Am,An) ≤ Vmax

Search the combinations between Vmin and Vmax

All the tables with index (J) that falls between (i-1) and (r+1) must be searched. Note that r<i; If i-r <2 stop the search

Each table with index (J) must be searched using the algorithm described in Algorithm-2.

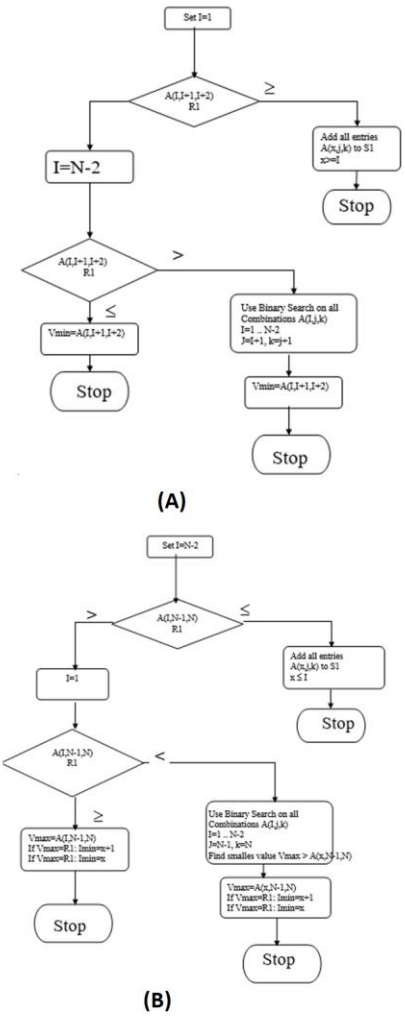

Figure 2. Algorithm-3 for (A) Locating V-min and (B) Locating V-max

4. Practical Implementation Use Cases

In this section, we provide real use studies, which demonstrate the functionality and the performance of Intermod60.

4.1 Single Tower Case

Intermod60 interface allows the user to select between single and multiple towers. Most of the required information for conducting the study is available online from several web servers, e.g., the FCC web server, and web servers of various antenna makers. Once the user enters the location of the tower/site in terms of longitude and latitude, Intermod60 will pull most of the required information, such as the carriers' names, the antenna and filter configurations. The user can modify or add any other relevant information. This process simplifies the task of the user a great deal. In addition, the SW interface verifies the validity of the user-supplied information. The interface automatically generates an input text file to the core engine.

Table 6 shows an example of an input file. The data in Table 6 is necessary to calculate the intermodulation hits. The frequencies transmitted by a carrier antenna are determined by the minimum (TX Min), the maximum (TX Max) and the channel increment bandwidth (the last column). The RX_Min and RX_max give the receiving frequency range. For example, carrier 1 transmits at frequencies 850, 850.025, 850.05, 850.075, 850.1, … 865 (a total of 600 channels). Carrier 1 receives signals in the range 806-821 MHz. The intermodulation hits are further evaluated based on the power of the interference generated by the intermodulation products.

Table 6. User Provided Data for the Intermodulation Calculation Phase

Carrier Namen	TX Min (MHz)	TX Max (MHz)	RX Min (MHz)	RX Max (MHz)	TXChannel Bandwidith (KhZ)
C-1	850	865	806	821	25
C-2	870	880	824	835	30
C-3	880	890	835	845	30
C-4	890	891.5	845	846.5	30
C-5	891.5	894	846.5	849	30
C-6	902	928	902	928	1200

The data required for this part of the calculation is shown in Table 7.

Table 7. User Provided Data for the Power Calculation Phase

Carrier Name	Height ft	Power Out(W)	Total Coax Loss	Equipment Receive Sensitivity (dBm)	Antenna Length (ft)
C-1	170	30	-6	-100	6
C-2	105	30	-6	-105	4
C-3	150	30	-6	-105	6
C-4	105	30	-6	-105	4
C-5	150	30	-6	-105	6
C-6	125	2.5	-6	-110	8

Intermod60 calculates all the 2nd and 3rd order intermodulation products. Fourth and higher order intermodulation products were found to have too little power to cause harmful interference (C/I ratio is not significantly reduced). Consequently, Intermod60 reports only 2nd and 3rd order intermodulation products with sufficient power to cause a harmful interference. The power level of interference is a user defined optimization parameter. Typical intermodulation products results are shown in Table 8.

The results in Table 8 show the channel impacted by the intermodulation product (victim carrier). The OP field indicates the intermodulation operation, which results in the IM product (1 means add, -1 means subtract, and 2 means two adds for the same frequency used for the third order harmonics). For example, the IM product 851+851-881 interferes with 821, which belongs to C-1 (first row). The last column presents the interference power generated by the intermodulation product (-73.937 dB for the first row).

Table 8. Sample Intermodulation Output Results

Victim Carrier		1st Signal Interfering			2nd Signal Interfering			3rd Signal Interfering			Interference Power
Name	Frequency	OP	Name	Frequency	Op	Name	Frequency	OP	Name	Frequency	
C-1	821	2	C-1	851	-1	C-2	881	0	-	0	-73.937
C-1	820.9	2	C-1	851	-1	C-2	881.1	0	-	0	-73.939
C-3	812.43	1	C-1	851	1	C-1	853.8	-1	C-2	892.37	-75.133
C-3	812.53	1	C-1	851	1	C-1	853.9	-1	C-2	892.37	-75.135

The number of intermodulation products can be extremely large, depending on the number of collocated carriers, and the spectrum utilized by each carrier. Intermod60 provides a summary report, indicating the name of carriers participating in a harmful interference. An example of a summary report is shown in Table 9. The first column shows which carrier is being impacted by interference. The next three columns show the carriers that generate intermodulation hit against the victim carrier. The last column shows the amount of power loss (X) needed in order to avoid interference; X would be reduced to zero once interference is resolved.

Table 9. Example of a Summary Report

Victim Carrier	First Tx Component	Second Tx Component	Third Tx Component	Additional Loss Needed
C-v	C-i	C-j	C-k	X dB

Table 10 shows the final result of all the interference tests performed by Intermod60.

Table 10. Summary Interference Results

Type of interference	Status	Noise Above Threshold
Intermodulation	Failed	14.463
Transmission Noise	Failed	82.57
Receiver Desensitization	Failed	81.16
Transmitter Harmonics	Failed	16.415
Transmitter Spurious	Pass	0

4.2 Multiple Tower Case

Due to increased demand for service coverage in certain areas, the number of towers located in close proximity to one another has increased. It is typical in the industry to refer to towers located in close proximity as a “tower farm”. Each tower hosts one or several service provider carriers. Because the distance between the towers is relatively small, it is possible for intermodulation products to form among carriers on different towers. The intermodulation calculation process is similar to the single tower case, except that the scale of the calculation complexity is far greater than that for single towers.

However, the relative distance between the towers has a direct impact on the power level of the intermodulation product. The mathematics for the power calculation includes the distance between various towers. The user specifies the latitude and longitude of each tower in the tower farm, and the interface will calculate the relative distance matrix for all the towers involved in the study.

Intermod60 is capable of processing up to 26 towers, with 12 carriers per tower. The limitation is a soft limitation and can be adjusted based on users' needs. However, it is our experience that the current limit is more than sufficient for all practical cases.

4.3 Rooftop Case

Building towers in highly populated urban areas is difficult, expensive, and faces serious zoning challenges. Consequently, service providers use rooftops of high-rise buildings to place their antennas. Each carrier uses several (typically 3) sectors for signals transmit and receive. From interference analysis perspective, each sector is treated as an independent source of frequencies, as well as a potential recipient of frequencies. The rooftop case is similar to the multiple tower case, except that the exact address of each sector/carrier cannot be given in longitude and latitude. There is only one address for the rooftop. Hence, the calculation of the distances between the various sectors takes a different approach.
Intermod60 interface provides the user with a graphical view of a grid, representing the surface of a rooftop. The user can drag and drop the sectors to the proper locations on the grid. The size of the grid represents the size of the rooftop. Once the user allocates all the sectors on the rooftop grid, the interface will calculate the relative distances between all the sectors. The calculation engine generates all the intermodulation products, whose power level exceeds the receiver sensitivity power level. Intermod60 is capable of processing up to 50 sectors per rooftop.

4.4 Measured Field Data Case

The interference analysis based on tower or rooftop collocation provides a base for pre-collocation optimization as well as post-collocation diagnosis and analysis. This process takes into consideration the entire spectrum allocated to each of the carriers. Although this process is thorough and complete, there is always a need to conduct an interference study based on the actual utilized frequencies in a specific area. For example, when a carrier experiences high call drop rate at one of its sites, it is often required to conduct an interference analysis for all the frequency signals in the neighborhood of the site. In this case, the user would use a spectrum analyzer to scan the frequency signals during busy hours. The scanned frequencies are then analyzed for interference.
Intermod60 interface accepts raw text data generated by spectrum analyzers, which measures the signal frequency, strength and source. This data is then fed to the Intermod60 engine for intermodulation product analysis. This capability enables Intermod60 to be a comprehensive tool for external interference analysis. The interface SW filters the data file and eliminates the repeated frequencies and the frequencies with weak signals (below a threshold set by the user). The size of the file and the number of frequency channels scanned are not limited in the software tool.

5. Performance Evaluation

For IM interference to be thoroughly analyzed, the time and space complexity of the algorithms used in IM tools must be reduced by at least an order of magnitude. In the absence of such performance improvement, the engineers must perform different types of manual reductions, assumptions, and eliminations to keep the complexity within practical limits. This results in a loss of accuracy and higher engineering costs.
Most of the interference analysis tools currently existing in the industry employ one or more variations of the typical algorithms (1-T and 2-T) discussed above. As a result, the computation time required for interference analysis has been prohibitive, significantly when the number of frequency channels exceeds 500 (complexity is 1012) for 3rd order IM products. To appreciate this type of complexity, assume that each computer operation (sort, search, generate permutation) requires 10 ns (a supercomputer performance). The processing time for detecting the IM products produced by 500 frequency channels is 3 hours.

We have evaluated the performance of three intermodulation product interference analysis tools, namely TAP/SoftWright® [22], Intermod60 and Comsite Pro® [23]. TAP/SoftWright® and Comsite Pro® are two of the most widely used tools in the industry, and Intermod60®. Both TAP and Comsite tools use analysis algorithms (1-T and 3-T). Intermod60 uses a rather linear algorithm, which utilizes the concept of trending search algorithm. Trending search first detects a particular trend in the behavior of intermodulation permutations; then the trend is used to detect harmful permutations and ignore benign ones. Table 11 summarizes the performance of the three tools.

Table 11. Performance Evaluation of Interference Analysis Tools

Channels	Comsite Pro (RCC)	SoftRight	Intermod60
	Time(hh:mm:ss)	Time(hh:mm:ss)	Time(hh:mm:ss)
50	00:02:00	00:02:00	00:00:10
100	00:45:00	00:20:00	00:00:30
200	01:54:00	01:20:00	00:00:40
400	08:20:00	05:20:00	00:00:55
800	18:10:00	15:10:00	00:01:20
1000	26:30:00	24:30:00	00:04:30

The performance of Intermod60 shows a linear behavior (as shown in Figure 3), while the other two tools (SoftRight® and Comsite Pro®) exhibit exponential behavior. Intermod60 completed the calculation for 1000 channels (both 2nd and 3rd order intermodulation) in less than 5 minutes. The other two tools took more than 24 hours. The exponential behavior of SoftRight® and Comsite Pro® suggests that some variation of the typical algorithms is used in their implementation. The linear behavior exhibited by Intermod60 performance is due to the restructuring of the intermodulation product detection algorithms utilized by the tool.

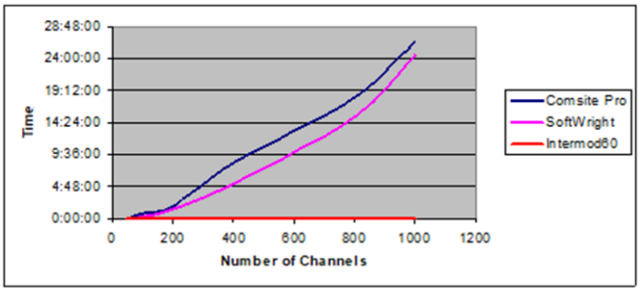

Figure 3. Performance Evaluation of 3 intermodulation SW

We also plot the number of intermodulation products generated (in millions of hits). The reason we show the number of hits is because the execution time includes the time needed to write the hits to the hard disk, in the format shown in Table 11. Note that Intermod60 took less than 40 minutes to generate 50.9 million intermodulation products resulting from 2000 channels. It took less than 10 minutes to complete a study with 1000 channels and more than 14 million intermodulation products. Tools such as Comsite Pro®8 [http://www.rcc.com/] and TAP/SoftWright®9 [http://www.softwright.com/], take more than 24 hours to complete the 1000 channel study. Both tools have repeatedly failed to complete the 2000 channels study.

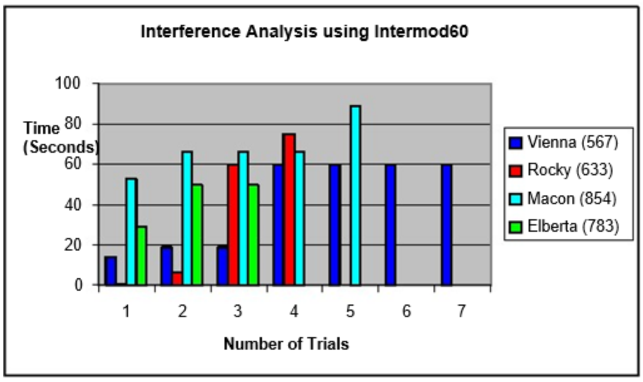

Figure 4. Interference Analysis Using Inermod60

Table 12. Site Interference Characteristics

Site Name	Number of carriers	Number of channels	Number of receiving Ranges	Maximum number of Hits	Max. Time (sec)
Vienna	3	567	6	12776	60
Rocky	4	633	6	31719	75
Macon	3	854	5	52870	89
Elberta	4	783	5	320	50

Figure 4 demonstrates how the high performance of Intermod60 simplifies and speeds up the process of interference analysis. The figure shows the results of performing intermodulation analysis on four sites with the generic names Vienna, Rocky, Macon, and Elberta. The number in parenthesis is the number of distinct frequency channels used at the corresponding site. For each site we conducted a number of interference analysis, while changing some of the parameters in the system in order to achieve zero interference. For Vienna site (with 567 channels) we conducted 7 studies, each lasting no more than 60 seconds. For Macon site (with 854 channels) we conducted 5 studies, each taking on the average 60 seconds, with a maximum of 85 seconds. These studies are real field studies. The characteristics of the sites are given in Table 12. Typically, it would take several weeks to complete these studies if other tools are used to make the analysis.

6. Conclusions

Intermodulation products continue to be one of the significant sources of external interference. The detection and analysis of this type of interference require a relatively large number of computations. This paper presented Intermod60, a new and high-performance intermodulation interference analysis tool. Intermod60 utilizes high speed search algorithms designed to detect intermodulation products. The performance of Intermod60 significantly enhances the process of interference analysis. Using Intermod60, it is now possible to conduct series of intermodulation interference analysis aimed at interference optimization.

Currently, the performance of Intermod60 is only hindered by the report generation, which is directly proportional to the number of intermodulation hits. The authors are considering new approaches for improving this part of the tool. Furthermore, we are looking at means of enhancing the tool by automating the interference optimization process. On the other hand, the current tools, which are commonly used in the market, exhibit rather non-linear behavior.

References

[1] IS54, IS95, and IS136

[2] Timiri, Shailender. "RF interference analysis for collocated systems." Microwave Journal 40, no. 1 (1997): 80-90.

[3] "Intermodulation Interference in Radio Systems," Technical Reference, AT&T Bell System, Nov 1972

[4] R. Barden, "Testing on the Edge Part I", Feb 2002. https://www.evaluationengineering.com/home/article/1300199 0/testing-on-the-edge-part-i . Accessed on Nov 17, 2020.

[5] K. Chamberlin, "Quantitative Analysis of Intermodulation Product Interference". *IEEE Trans*, vol. 31, no. 3, pp.316-319, 1989.

[6] N. B. de Carvalho and R. C. Madureira, "Intermodulation Interference in the GSM/UMTS Bands", in *Proc.* European Conference on Wireless Technologies, Paris, France, 2000.

[7] R. Nagel and K. Audenaerde "Passive Intermodulation in Duplexed Communication Systems", Semint97, Parana, Brazil, Oct. 1997.

[8] J. A. Betts and D. R. Ebenezer "Generation of Intermodulation Interference Due to Nonlinear Effects in the Near- Field Regions of Multiple-Transmission Communication Systems". in *Proc.* AGARD Conf. on Aerospace Telecommunication Systems. May 1972, pp. 25.1-25.12

[9] J. A. Betts and D. R. Ebenezer "Intermodulation Interference in Mobile Multiple-Transmission Communication Systems Operating at High Frequencies (3-30 MHz)". in *Proc.* IEE, 1973, 120, (11), pp. 1337- 1344

[10] P.S.W. Ho, et al. "Intermodulation Interference in Radio Systems". in *Proc.* AGARD (NATO) Conf. on Effects of Electromagnetic Noise and Interference on Performance of Military Communication Systems. Portugal, 1987, AGARD-CP-420, pp. 30/1-30/8

[11] J. C. Lee "Intermodulation Measurement and Analysis of Some Conducting Materials Commonly Used in Aerospace". in *Proc.* IEEE Int. Conf. on Communications, Vol. 2, June 1980, pp. 25.6.1-25.6.6

[12] A. W. D. Watson "The Measurement, Detection, Location and Suppression of External Non-linearities which Affect Radio Systems". in *Proc.* IERE Conf. on Electromagnetic Compatibility, 1980, pp. 1-10

[13] J. A. Woody and T. G. Shands "Investigation of Intermodulation Products Generated in Coaxial Cables and Connectors". RADC-TR-82-240. Georgia Institute of Technology, USA. September 1982

[14] A. K. Brown, "Passive intermodulation products in antennas-an overview," *IEE Colloquium on Passive Intermodulation Products in Antennas and Related Structures*, London, UK, 1989, pp. 1/1-1/3.

[15] Stauss, Bailey, Bond, Carosella and Ehrlich: "Studies on the Reduction of Internodulation Generation in Communications Systems". Naval Research Laboratory, Washington, D.C. (Ref: ADA086995)

[16] D. Molta "Whose Problem is WLAN Interference?", *Network Computing*, July 2002

[17] J. Silberberg "Performance Degradation of Electronic Medical Devices Due to Electromagnetic Interference." *J. Compliance Engineering*, vol. 10, no. 5, 1993.

[18] U. Bostrum "Interference from Mobile Telephones-a Challenge for Clinical Engineers" Clinical Engineering Update, No. 10, November 1991

[19] "Paperman, W. David, Y. David, and K. A. McKee. "Electromagnetic interference: causes and concerns in the health care environment." Healthcare facilities management series (1994): 1-15.

[20] "Position Paper on WLAN Radio Frequency Interference Issues", Wireless Campus Networking Working Group, http://www.educause.edu/ir/library/pdf/NET0014.pdf . Accessed Nov 18, 2020.

[21] D. Costas et. al. "Time Domain Modeling for Large Scale Cosite Interference Problems Utilizing Parallel Computing and Wavelets" in *Proc.* HPCMP Users Group Conference 2001.

[22] Softwright . http://www.softwright.com/ . Accessed Nov 18, 2020.

[23] http://www.rcc.int/. Accessed Nov 18, 2020.

[24] J.L Smith "A Method to Predict the Level of Intermodulation Products in Broadband Power Amplifiers", *Microwave Journal*, February, 2003

[25] Greg Ignatius, Brian Moore, Mohammad Malkawi, Larry Votta "Availability Work Products – A Strategic Approach", IEEE Signal Processing Society 5thWSES International Conference, Crete, July 2001.

[26] Bashir Haswarey, Mohammad Malkawi, Larry Votta "Overload Management in a High Availability Platform", Software Engineering Symposium Journal, 2000, Vol.3

[27] Mohammad Malkawi, Larry Votta, and Steve Matz "Analysis of Failure and Recovery Rates in a Wireless elecommunications System", International Performance and Dependability Symposium, June 23-27, 2002,Bethesda, Md, pp.570-581

[28] Haque, Tanvir Ul, and Sharif Hossen. "PAPR reduction in OFDM system using clipping and filtering methods based on CCDF." International Journal of Computer Network and Information Security 11, no. 10 (2019): 12.