Congreso Internacional en Ingeniería Electrónica. Mem. Electro 2015, Vol. 37, pp 13-18, Chihuahua, Chih. México
http://depi.itchihuahua.edu.mx/display/memorias_electro/MemoriaElectro2015
ISSN 1405-2172

PASSIVE INTERMODULATION TESTING SOFTWARE FOR AMPHENOL OPTIMIZE USING NI LABVIEW

Wirichaga Eduardo, Medina Luis, Espinoza Indelfonso, López Gabriel
Universidad Tecnológica de Nogales
Mecatrónica: Área Automatización
Av. Universidad 271 Col. Universitaria 84097. Nogales, Son.
Tel. 6311674304
ewirichaga@gmail.com, lmedinatnogales.edu.mx

RESUMEN.

La intermodulación pasiva, también conocida como PIM (Passive Intermodulation), es una distorsión que afecta diferentes componentes pasivos como antenas, cables u otros. La intermodulación pasiva tiene como consecuencia la degradación de la calidad de la comunicación inalámbrica [1]. El siguiente trabajo presenta el desarrollo de una aplicación de monitoreo y captura de datos de pruebas PIM para antenas. El desarrollo de la aplicación es utilizando LabVIEW de National Instruments, Microsoft Office Excel, Microsoft SQL Express, y equipo y software de Kaelus. La aplicación final es un ambiente de monitoreo completo diseñado para llevar registro del progreso de pruebas industriales, guardar datos, y generar reportes finales. Se llevaron a cabo pruebas para asegurar que la comunicación la funcionalidad entre el equipo de PIM y la interfaz de monitoreo, y también probar las diferentes características de la aplicación. El siguiente trabajo ha sido creado para el monitoreo y control de procesos.

Palabras Clave: interfaz, comunicación, intermodulación pasiva, monitoreo, control de procesos.

ABSTRACT.

Passive intermodulation (PIM) is a distortion that affects different passive components such as antennas, cables, among others. As a consequence, passive intermodulation can cause quality degradation and failure in wireless communication systems [1]. The following project presents the development of an application designed to monitor and capture data from antenna PIM testing. The development of the application includes the use of LabVIEW from National Instruments, Microsoft Office Excel, Microsoft SQL Express, and equipment and software from Kaelus. The final application is a complete monitoring environment designed to keep track of industrial testing progress, save data, and generate final reports. There were tests to make sure the communication between the PIM equipment and the application was functional, and also to test the different features of the application. The following project has been created for monitoring and control systems.

Keywords: interface, communication, passive intermodulation, monitoring, control systems.

1. INTRODUCTION

Passive intermodulation has become more necessary recently. Given the continuous development on communications, wireless communication systems have been getting more complex. The term PIM went from being a vaguely known issue to become a major concern for wireless service providers. The importance of PIM relies on its impact on long-term evolution (LTE) telecommunications. PIM interference can affect the wireless service quality substantially. Due to this, it is clear why PIM testing is becoming more important [2-3]. Human machine interfacing (HMI) refers to the development of applications to make possible then interaction between humans and machines. In this project, the communication between the computer and PIM equipment will allow an operator to interact with the PIM equipment so the testing process can be controlled [4].

The application communicates with the PIM testing equipment and takes measurements to determine if the tested antenna has an acceptable level of PIM interference. The application is designed to provide with live-time data so the tester can monitor the PIM interference level while he or she performs the test procedure set by the company's quality policy. The design and architecture of the software addresses the need for efficiency and speed.

It is worth to notice that creating a user interface implies smart design by the developer since the interface has to be simple and functional for anyone to understand.

2. DEVELOPMENT

2.1. Equipment

There are many providers for PIM equipment. This application is made to interface with equipment provided by Kaelus. The equipment is composed by different systems that are designed to test at different frequency bands in MHz. In this case, the application communicates with analyzers from the E-Series PIM Analyzers. Each system has a front-end (FE) and a transceiver (TR) module. The equipment connects to the computer via USB [5].

Figure 1. FE and TR modules from E-Series PIM Analyzers.

2.2. Communication Protocol and Programming Environment

The Application Program Interface (API) protocol to communicate with the equipment is based on an exchange of commands and responses between a Web Engine application provided by Kaelus and an application capable of sending commands. The format of the commands and responses are Hypertext Transfer Protocol (HTTP), which is an application protocol used on data communication involving the World Wide Web. The commands are sent over from a standard HTTP connection established from a TCP port. The commands are sent to the Web Engine which gives you authorization to obtain data from the Kaelus website. The exchange of information is only possible using the Web Engine. The commands are already defined by the same equipment provider [6-7].

Figure 2. Communication diagram.

LabVIEW from National Instruments is a graphic programming environment specialized on industrial applications. One of the main reasons to use this type of language (besides its optimized and efficient performance) is that Kaelus provides an optimized library to communicate with the PIM analyzers. The library is designed to support high-speed communication which is critical given that the application has to be capable to display real-time data.

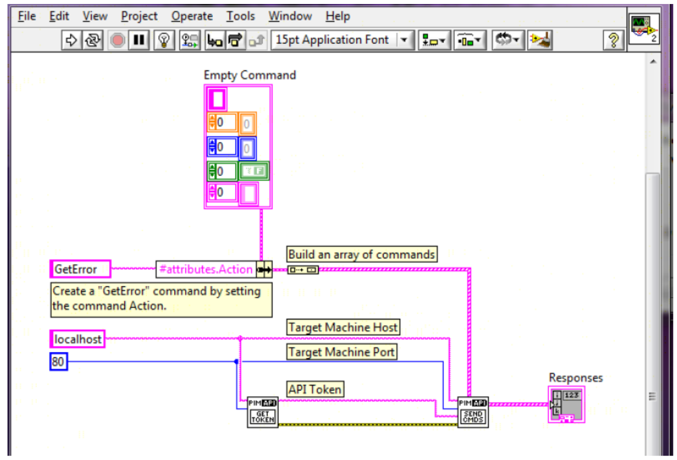

Figure 3. Example of basic communication between LabVIEW and PIM Equipment.

In the figure above, an example of communication using the library from Kaelus can be seen. The example asks for an authorization token which is given by the Web Engine. The token is necessary to be able to get data from the Kaelus website. The example shown is taken from the API Manual, after getting the token, the application in this example sends a GetError command and reads the responses from the Web Engine [7-8].

2.3. Antennas

An antenna is a structure designed for the transmission and reception of waves or electric energy. On transmission, the antenna accepts electric energy from a transmission line and sends radio waves to space. On reception, the antenna receives radio waves to create electric energy [9].

Given the growth of wireless services, it is easy to understand why antennas are a product on high demand. Antennas are a passive component that can be easily affected by passive intermodulation. It is important to take into account that wireless services operate at high frequency bands, and that is why any detail or mistake in the production line would increase the PIM level; an increased PIM level is enough to affect the quality of the product and provoke loss or repair as a consequence.

2.4. PIM Testing Characteristics

The testing application is designed to function with the following procedure:

1) Antenna goes to PIM testing room and PIM 1 test is performed by an operator/tester.
2) If the antenna passes the test, the antenna moves on to PIM 2. If the antenna does not pass PIM 1, then it is necessary to find the reason and repeat PIM 1 until the antenna passes. It is necessary to pass PIM 1 to go to PIM 2
3) Once in the PIM 2 stage, the antenna has to pass PIM 2 in order to be done with PIM testing at all.

It is important to mention that PIM 1 and PIM 2 are exactly the same test. The PIM test consists on getting measurements from the analyzers, if the intermodulation (IM) level is above a certain limit then it fails. But PIM 1 is a draft or pre-test before PIM 2. For the application, this means that the results for PIM 2 are going to be on the final report, even though data from PIM 1 and 2 are saved to the same database.

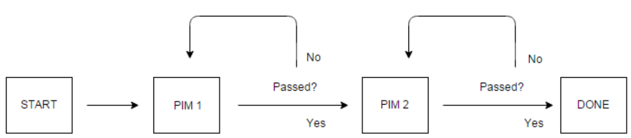

Figure 4. PIM testing diagram

The application needs to display the data in a graph. The graph has to be able to create a swept effect and be overwritten by new data. The equipment will perform power measurements (dBc) over a defined frequency band. There are two carriers to be measured. For example, if the analyzer to use is set to measure and test the antenna at a 100-150 MHz frequency band; it will test one carrier from 100 to 150 MHz, then it will test the other carrier at the same frequency band and repeat the process over and over again until the tester ends the test. That means the graph has to show the IM Level (power, dBc) vs Frequency (MHz) and be updated point by point.

Each type of antenna has a different set of ports which are mostly named after colors. Each color/port has two terminals +45 and -45. There is just one case where the port is not named after a color, in this case the name is PORT with terminals 1 and 2.

2.5. Application Architecture

The most important part involving the development of this application is to create a software full of features without losing performance since the real-time data acquisition had to be part of the application. The library from Kaelus improved the development of the project significantly. The VIs (LabVIEW files) provided by the library are designed to send a cluster (or package) of data that will be later formatted into HTTP. The commands used to control the equipment were:

1) GetAnalyzers: returns a list of analyzers connected to the PC.
2) SetInit: Initializes communication with connected hardware.
3) SetMeasBand: sets the measurement band depending on the band index (band index is provided by GetAnalyzers).
4) SetSweepModeTx: turns on sweep mode.
5) SetPreset: sets default parameters for testing.
6) SetPxx: sets port and type of test.
7) SetTxOn (True,True): turns on both carriers.
8) SetTrigger: forces measurement.
9) GetSamples: read samples obtained from SetTrigger.
10) SetTxOn (False, False): turns off both carriers.
11) SetExit: closes connection with connected hardware.

To avoid losing speed, it is important to be efficient with the architecture. There has to be efficiency with the way the commands are sent and how the responses are processed. To make it faster, that commands are sent in arrays so the application can send the commands and get responses as fast as possible. The architecture of the application is based mainly on two basic methods well known by LabVIEW developers: state machine and producer-consumer. State machine refers to an architecture that lets the application perform different tasks depending on a certain state. In LabVIEW, the most basic state machine is composed of a case structure inside of a while loop or for loop. A producer-consumer structure is a method used to pass data between different processes; local variables are the most basic components that can accomplish a basic producer-consumer structure. Generally, the producer is a loop constantly acquiring data while the consumer is another loop reading and/or manipulating the data from the producer loop [7, 10, 11].

2.6. Application Guide and Features

The application is simple and easy to understand by any user. The interface has to be user friendly and functional [11]. One of the most important features of the application is its constant communication with different databases. All the databases are from Microsoft SQL Express; the application writes its results on a table from a database, but it also reads data from different tables located in different databases (including that database in which the application writes on). It is recommendable to follow the next steps when using the PIM testing software:

1) When open the application will search for available analyzers. If not found, the square indicator besides the PIM Equipment field will be white and a Get Analyzers button will be shown; if found, the square

indicator will be blue is there is equipment detected. The first step is to enter a serial number. This application is designed to automate the PIM testing process. Once the serial number is entered, the application will then read from a database to get a part number and check the database to determine the stage of the test (whether the antenna is at PIM 1, PIM2, or DONE). The status check is accomplished by reading from the database to check past results and determine if the antenna has been through any of the stages. If the status shows PIM 1, then the results from the test will be for PIM 1, and the same applies for PIM 2. If the status shows DONE, then there won't be any test because the antenna already passed the necessary stages. The stages progress can be seen by checking the PIM 1 and PIM 2 indicators; the indicators will be red if failed, green if passed, and blue if the stage has not been tested yet.

2) Once the application retrieves the part number (antenna), it will then communicate with another database to retrieve information about the ports. It will retrieve what ports does the antenna have and the frequency bands those ports should be tested for.
3) As the information about the ports is retrieved, the application will be refreshed to display and enable only the available ports for the part number corresponding the serial number.
4) Once the ports are available and equipment has been detected, then the user can select a port. Once the port is selected, it is important to notice the two buttons on top of the ports (LB for Low Band and HB for High Band). If the port is to be tested only on low frequency band, then the HB button will be disabled and grayed out, and vice versa. If the port is to be tested on both bands, both buttons will be enabled.
5) From now on the user can fill the rest of the information in any order, but has to make sure to fill all the information since the operator won't be able to perform the test if any information is missing. There is a CLEAR button to erase information if needed.
6) The operator can easily fill the Operator camp with his/her id number. However, the IM Limit and the DUT port selection (analyzer port to perform the test) have to be set by an authorized user. To set those parameters, click on Setup and the user will be prompted with a login window. The authorized user needs to login so he/she can change the IM Limit and set the DUT port. The login information will be compared to a database to determine if the user has access. Once the parameters are set, click on Setup again to go back to the main window.
7) The START and DONE button will be enabled. The REPORT button will only be enable when PIM 2 has been passed.
8) Press START and the graph will initialize and it will show cursors that follow that peak value for each curve. That value is important because it is the maximum value, and if any value is above the limit line test fails. There is a big PASS/FAIL indication at the right side showing the current result, and a gray slider displaying the highest value recorded.
9) Once the operator is ready to finish the test, he or she can press DONE. When pressing DONE, a pop-up window will appear asking if the user is ready to write the results on the database or cancel. If the user pressed "Ok", the data will be written on a database, if not, the test will be cancelled and the application will be ready to test another port.
10) The writing of data on the database sets the end of the test for the port. If the user sent data from a port to the database, a small pass/fail indicator will appear to keep track of the progress. The indicators work like a matrix, if the user tested BLUE +45 at HB, the indicator will be located at the interception between the port button and HB button. The indicator ends that particular test at all so the same test won't be repeated. The user can restart and start over the entire antenna test by clicking EXIT.
11) Once PIM 1 and PIM 2 are completed. The status will change to DONE and the REPORT button will be available. By pressing REPORT, the application creates a report based on data from PIM 2 on Excel and sends the report to print.

Other features include a timer to keep track of time, an error bar that indicates if there is an error involving the measurements, DUT port indicators, date and time, calibration information, and peak values above the graph.

Here are pictures showing the composing panels of the application:

Congreso Internacional en Ingeniería Electrónica. Mem. Electro 2015, Vol. 37, pp 13-18, Chihuahua, Chih. México
http://depi.itchihuahua.edu.mx/display/memorias_electro/MemoriaElectro2015
ISSN 1405-2172

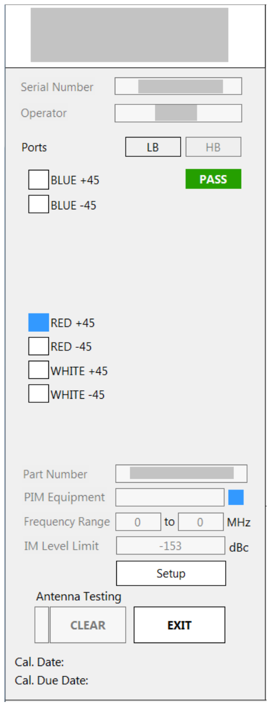

Figure 5. Left panel.

The left panel is where the user sets parameters and can keep track of the testing progress. The user can access to a login window and set certain parameters by clicking on Setup.

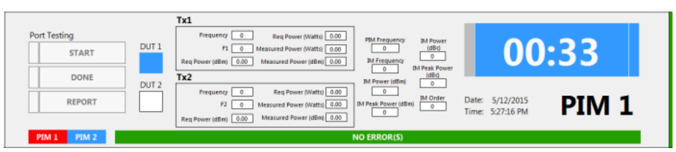

Figure 6. Center panel (control and data section).

This section of the center panel shoes the test stage and the timer. In this section, the user can see the progress of PIM 1 and 2, the error bar, control the test, and see data in a numerical way.

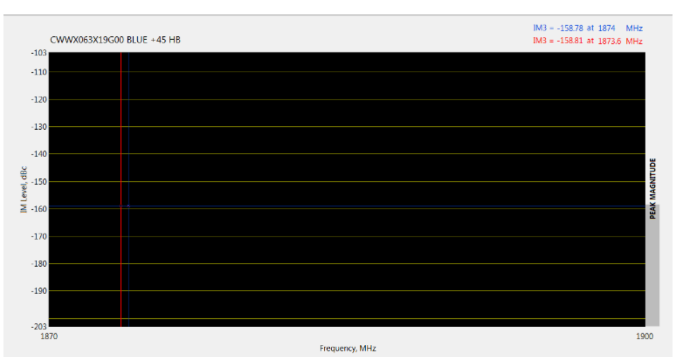

Figure 7. Center panel (graph section).

The graph section along with the right panel (PASS/FAIL indication) is the most important part of the interface for the user. The operator has to be able to see real time data in a graphical way from the testing site.

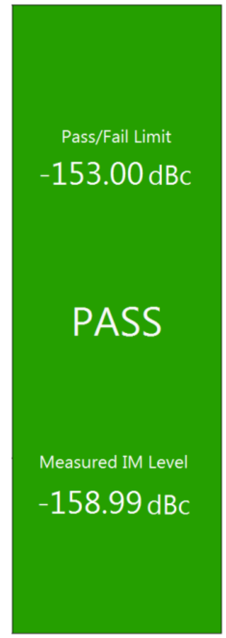

Figure 8. Right panel.

The right panel is a big indicator so the operator can monitor the current outcome status from the testing site without mucho effort.

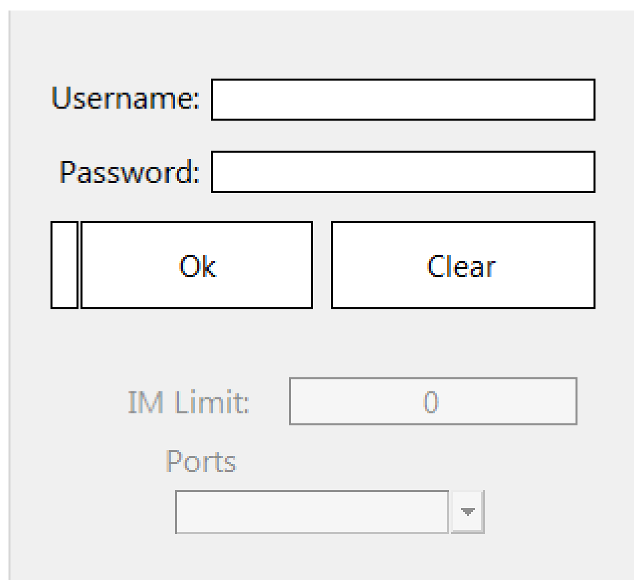

Figure 9. Login window.

Some parameters for the test have to be authorized by certain users. This windows allows and authorized user to login and set the IM Limit and the DUT port to use.

3. TESTING

The application went through tests to make sure it was working properly. The application successfully communicated with the equipment and performed as expected. A picture of one the tests is shown below.

The figure below shows the application during its interaction with the equipment. The current results shows PASS because all the points on both curves (carrier 1 and 2) are below the limit line.

Figure 10. Interface during test.

The testing stage cannot be skipped in the development of software involving interaction with instrumentation. It is critical to know how the equipment will react to the algorithms written on the application code so the developer can apply changes if needed. The development of the PIM testing application involved a lot of testing to study the behavior of the equipment. A lot of changes and adjustments were made based on how the equipment reacted to the code. The testing stage permitted to make sure the application used the right commands to accomplish the desired behavior.

4. RECOMENDATIONS

It is of extreme importance to follow the application guide. And it is also useful to understand that if for any reason the equipment is not detected or working properly; the user should try to disconnect and reconnect the USB cable that connects the equipment to the computer. Given that the connection between the PC and the analyzers is via USB, the user should be aware that some issues can be solved by disconnecting and reconnecting the equipment. There is a connection issue with the Web Engine when changing equipment to test different frequencies; the change between analyzers may be too sudden for the Web Engine causing the communication to be unsuccessful. However, the application was created to avoid that issue by closing connection with the hardware when a test is done, and reinitializing the connection when starting another test.

5. CONCLUSION

When developing HMI applications, the developer needs to be aware that the communication with the equipment is not the hardest part but just the beginning of a long process. The most challenging part of the development is to write algorithms to successfully create the desired behavior of the application. The user interface has to be simple and functional. The developer has to remember that he/she is not designing and application for himself'/herself but for another user that may not be as technical and informed about the process as him/her.

This application is important because PIM testing is becoming more and more necessary in a world where wireless services are starting to become a human need and not just a luxury. Wireless service providers need reliable alternatives to keep PIM interference under control and make sure they produce quality equipment for their services. There is no better alternative to reduce error than the development of systems capable of monitoring and controlling processes.

6. ACKNOWLEDGMENT

Amphenol Optimize proposed the project and provided all the resources for its development. Amphenol Optimize must be recognized for their support, investment, and willingness to develop the PIM testing software.

7. REFERENCES

[1] Keysight Technologies. *Passive Intermodulation (PIM) Measurement*. (n.d) retrieved May 13 2015, from Keysight Technologies Web Site: http://www.keysight.com/main/application.jspx?cc=MX&lc=spa&ckey=2127060&nid=-33157.0.00&id=2127060

[2] Hill, K (2014, October, 15). *PIM explained: passive intermodulation*. Retrieved May 13 2015, from RCR Wireless News Web Site: http://www.rcrwireless.com/20141015/test-and-measurement/pim-explained-passive-intermodulation-tag6

[3] Butler, R (2014). *PIM Testing: Advanced wireless services emphasize the need for better PIM control.* Commscope.

[4] Fiset, J., (2008). *Human-Machine Interface Design for Process Control Applications*. 1st ed. United States: ISA.

[5] Kaelus. *E-Series PIM Analyzers*. (n.d) retrieved May 13 2015, from Kaelus Web Site: http://www.kaelus.com/en/test-measurement-solutions/bench-pim/e-series-pim-analyzers

[6] Gourley, D., (2002). *HTTP: The Definitive Guide (Definitive Guides)*. 1st ed. United States: O'Relly Media.

[7] Kaelus (2013). *Kaelus Passive IM Analyzer System Advanced Programming Interface: Programming and Usage Guide.* Centennial, Colorado: Kaelus, Inc.

[8] Travis, J & Kring, J (2006). *LabVIEW for Everyone: Graphical Programming Made Easy and Fun, 3rd Edition.* Upper Saddle River, New Jersey: Prentice Hall.

[9] Johnson, R & Jasik, H (1993). *Antenna Engineering Handbook, 3rd Edition.* Atlanta, Georgia: McGraw Hill.

[10] National Instruments, (2015, April, 1). *Tutorial: State Machines*. Retrieved May 13 2015, from National Instruments Web Site: http://www.ni.com/tutorial/7595/en/

[11] National Instruments, (2012, September, 19). *Application Design Patterns: Producer/Consumer*. Retrieved May 13 2015, from National Instruments Web Site: http://www.ni.com/white-paper/3023/en/