

E.T.S. de Ingeniería Industrial,
Informática y de Telecomunicación

Desarrollo del software de control de los sistemas de posicionamiento de precisión para medida de antenas en milimétricas y submilimétricas



Grado en Ingeniería
en Tecnologías de Telecomunicación

Trabajo Fin de Grado

Autor: Mikel Goñi Esparza

Director: Juan Carlos Iriarte Galarregui

Codirector: Jorge Teniente Vallinas

Pamplona, 29 de Junio de 2015



Agradecimientos a...

*La empresa Anteral S.L. por darme la
oportunidad de realizar este proyecto
con su colaboración.*



Abstract

This project is focused on assessing the efficiency of a measuring and precision antenna positioning system, with the goal of improving a code programmed in LabView software that controls it. The current software has been studied and some improvements, like new functionalities, have been carried out enabling more accurate results, and also achieving a greater comfort in antenna radiation patterns measurements. For this purpose, many different versions of the software have been made according to the execution platform in which is going to be used, the kind of sweep and the number of the parameters which are desired.

The different programmed codes have been tested in various measurement setups using different frequency bands. The radiation patterns have been obtained via the processing of the results with a Matlab software tool, conclusions of relevance have been obtained analysing such radiation patterns. Besides, during this research, some problems arising from the programming structure and from the positioner features have appeared. This fact has caused some changes of initial thoughts related to what was needed to be done.

Resumen

Este proyecto se centra en evaluar el rendimiento de un sistema de medida y posicionamiento de precisión para antenas, con el objetivo de mejorar el software actual programado en LabVIEW que lo controla. Se ha estudiado el funcionamiento del software actual y se han llevado a cabo mejoras y nuevas funcionalidades que permiten obtener unos resultados más exactos y una mayor comodidad a la hora de medir los diagramas de radiación. Para ello, se han desarrollado distintas versiones del software en función de dónde se ejecute (PC o analizador), del tipo de barrido a realizar, así como del número de parámetros y de la precisión de los mismos que se desee obtener.

Se han realizado medidas de antenas en distintas bandas frecuenciales probando el código en diversas situaciones de medida. Los resultados obtenidos se han procesado con la herramienta software Matlab, y se han extraído conclusiones de gran relevancia gracias a las



representaciones de los diagramas. Además, durante el transcurso del trabajo han surgido ciertos problemas y obstáculos derivados tanto de la programación como de la propia estructura de posicionamiento. Este hecho ha provocado que algunos de los razonamientos iniciales hayan tenido que modificarse.



Palabras clave

Inglés:

- Near Field
- Antenna positioning system
- Sweep time
- Radiation pattern

Castellano:

- Campo cercano
- Posicionador
- Tiempo de barrido
- Diagrama de radiación



ÍNDICE

Abstract	1
Resumen	1
Palabras clave	3
CAPÍTULO 1. INTRODUCCIÓN	6
1.1. Motivación	6
1.2. Objetivos	6
CAPÍTULO 2. NEAR FIELD	8
2.1. Contexto tecnológico	8
2.2. Sistemas de medida en campo cercano.....	9
2.3. Bibliografía	12
CAPÍTULO 3. LABVIEW	13
3.1. Introducción.....	13
3.2. Conceptos generales (Programación gráfica)	13
3.3. Características y funciones.....	15
3.4. NI VISA.....	16
3.4.1. Introducción y primeros pasos.....	16
3.4.2. Explicación y uso	16
3.5. Bibliografía	18
CAPÍTULO 4. COMPONENTES Y SOFTWARE	19
4.1. Introducción.....	19
4.2. Componentes utilizados.....	19
4.3. Versiones previas	24
4.4. Versiones actuales y mejoras.....	26
4.4.1. Conceptos generales.....	26
4.4.2. Mejoras introducidas	27
4.4.2.1. Obtención de la posición (X, Y, Z).....	27



4.4.2.2.	Seleccionador de velocidad	29
4.4.2.3.	Tiempo automatizado	31
4.4.2.4.	Sweep time (tiempo de barrido).....	32
4.4.2.5.	Corrección en el barrido de puntos	34
4.4.2.6.	Sincronización y ejecución del programa.....	35
4.4.2.7.	Alineación de antenas.....	37
4.4.2.8.	Implementación en el PC	39
4.5.	Análisis y resultados.....	41
4.5.1.	Resultados conforme al tiempo de ejecución.....	41
4.6.	Problemas en el desarrollo	43
4.7.	Bibliografía	45
CAPÍTULO 5. MEDIDAS.....		46
5.1.	Introducción.....	46
5.2.	Metodología.....	46
5.3.	Análisis de los resultados	50
5.3.1.	Antena que escanea en frecuencia.....	51
5.3.2.	Bocina y lentes	56
5.3.3.	Array de 8 antenas.....	59
5.4.	Problema de entrelazado en la representación de los diagramas de radiación.....	62
CAPÍTULO 6. CONCLUSIONES Y LÍNEAS FUTURAS		67
6.1.	Conclusiones	67
6.2.	Líneas futuras.....	68
Anexos.....		70



CAPÍTULO 1. INTRODUCCIÓN

1.1. Motivación

La razón principal por la que he decidido realizar este proyecto ha sido la necesidad que presentaba la empresa Anteral S.L. de mejorar los sistemas de posicionamiento de antenas de los que disponía. Parte del funcionamiento de esta empresa reside en la medida de componentes de radiofrecuencia. Es por esto que se ha considerado una buena oportunidad estudiar el software actual y llevar a cabo nuevas mejoras que permitan que la empresa pueda desarrollar su labor de una manera más eficiente.

1.2. Objetivos

El objetivo de este proyecto consiste en desarrollar y evolucionar el software para el control y procesado de resultados de los sistemas de medida y posicionamiento 3D que dispone Anteral S.L., mediante los cuales se miden los correspondientes diagramas de radiación RF de antenas.

El programa encargado de realizar estas medidas se utiliza con mucha frecuencia y reducir el tiempo total de cada una de ellas es uno de los aspectos más importantes a mejorar durante este proyecto. Surge la necesidad de procesar los resultados con mayor exactitud y por este motivo una de las mejoras que se incluirá en el software es que este guarde la posición de cada uno de los puntos de los que se compone el diagrama de radiación, gracias a la conexión con un lector de posición.

Otro objetivo es independizar el software de toma de medidas del analizador, es decir, que el programa no esté instalado en él sino que esté en un PC externo que cada vez que se ejecute sea capaz de adquirir los datos del analizador y mover el posicionador.

Se pretende mejorar otros aspectos como es el requerimiento de intervención humana, haciendo el interfaz de usuario más sencillo. Por ejemplo, automatizando los tiempos de



sincronismo que usan las medidas o introduciendo funciones como la representación de gráficas que faciliten el alineamiento.

Como último objetivo se llevarán a cabo las medidas correspondientes para que la validación del software sea correcta.



CAPÍTULO 2. NEAR FIELD

2.1. Contexto tecnológico

Las medidas de diagramas de radiación, así como toda caracterización de antenas requieren de sistemas de posicionamiento y medida capaces de llevar a cabo estas funciones con gran exactitud. Uno de los métodos más baratos y más precisos en alta frecuencia son los sistemas de medida de antenas en campo cercano (Near Field) ya que el tamaño del setup se reduce considerablemente y además, en el caso que vamos a trabajar que es el campo cercano plano (planar near field) la complejidad de posicionamiento de medida se reduce puesto que los puntos de medida son una matriz de posiciones sobre un plano.

Hoy en día existen diferentes sistemas de medida para llevar a cabo diagramas de radiación de antenas. La gran ventaja de trabajar con el campo cercano respecto al campo lejano (Far Field) es en gran parte la comodidad dada por las dimensiones de los campos requeridos, aun utilizando antenas relativamente grandes. Además, la precisión que se obtiene en estas medidas es muy similar a la que se consigue en Far Field y el ambiente en el que se realizan está controlado, debido a que no dependen de las condiciones del tiempo y no afectan mucho las interferencias electromagnéticas.

La desventaja principal es que una medida en campo cercano de una antena normalmente lleva más tiempo que la misma medida en campo lejano, ya que debemos discretizar con al menos $\lambda/2$ densidad el campo cercano para poder hacer la transformación a campo lejano y extraer los campos radiados. Esta discretización suelen ser muchos puntos. También es destacable que en campo cercano y más concretamente en campo cercano plano tenemos otras fuentes de error como son los truncamientos, desalineamientos, etc.

2.2. Sistemas de medida en campo cercano

Cuando se analiza el diagrama de radiación de una antena es necesario tener en cuenta las tres distintas regiones en las que su radiación se puede dividir. En la Figura 2.1 aparecen representadas estas zonas: zona reactiva en campo cercano, radiación en campo cercano y campo lejano.

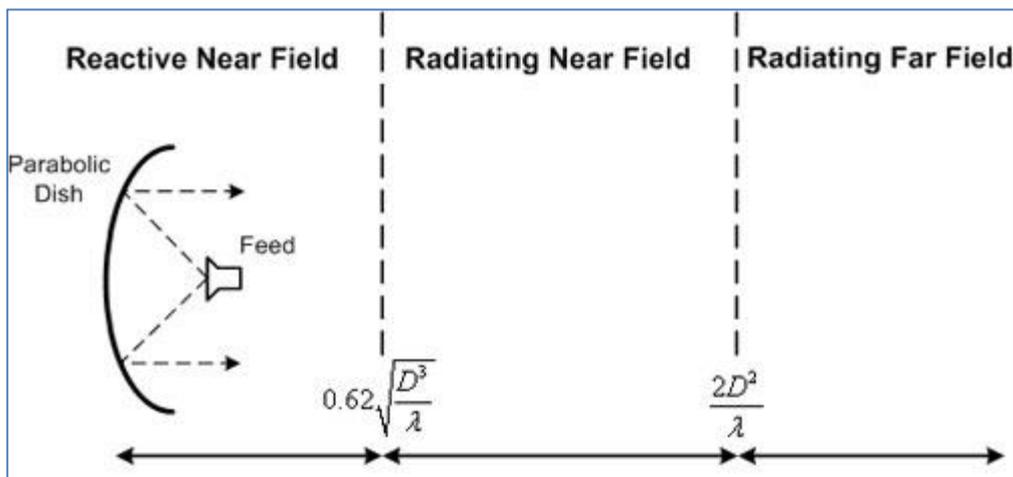


Figura 2.1: Regiones de radiación de una antena

La primera de las zonas, la reactiva, es una región donde la medida es prácticamente inviable debido a que todo objeto que se meta en dicha zona afecta al comportamiento de la antena alterando su funcionamiento, no obstante existen algunos sistemas de medida en campo cercano muy próximo que pueden dar resultados rápidos y que nos pueden dar una idea (sin ser muy exactos) del comportamiento de una antena. La segunda de las zonas corresponde al campo cercano (Región de Fresnel), es en este tramo es donde la energía en media es constante y por tanto se pueden realizar las medidas de los diagramas de radiación, como es el caso de este estudio. Por último, se encuentra el campo lejano donde la distribución de energía no cambia según la distancia. Si queremos medir una antena en campo lejano debemos estar más allá de la distancia de campo lejano en los puntos donde muestreemos el campo, como dicha distancia aumenta con la frecuencia, a más alta frecuencia estará más lejos y si la antena es de grandes dimensiones puede ser imposible hacerse en un entorno controlado (cámara anecoica).

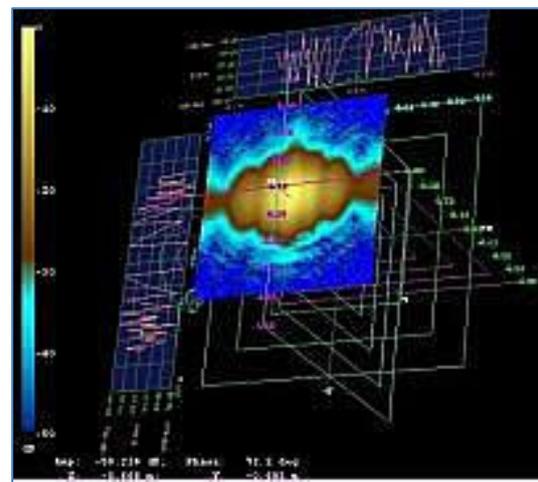
En este proyecto, como las medidas se han realizado en campo cercano, es necesaria su posterior transformación a campo lejano mediante la transformada de Fourier.

A continuación, se describen los tres sistemas de medida en campo cercano principales:

- Planar Near Field System: la antena a medir (AUT, antenna under test) no se mueve, y la sonda de medida se mueve en el eje de coordenadas cartesianas (en un plano X-Y) realizando un movimiento lineal en un plano perpendicular a la dirección de máxima radiación de la AUT. Los datos se obtienen acorde a un espaciado entre muestras, ya sea en el eje X o en el eje Y, como máximo de $\lambda/2$. Este es el tipo de sistema que se utiliza en este proyecto. En la Figura 2.2 se puede ver un ejemplo de la estructura y del diagrama de radiación que se obtiene.



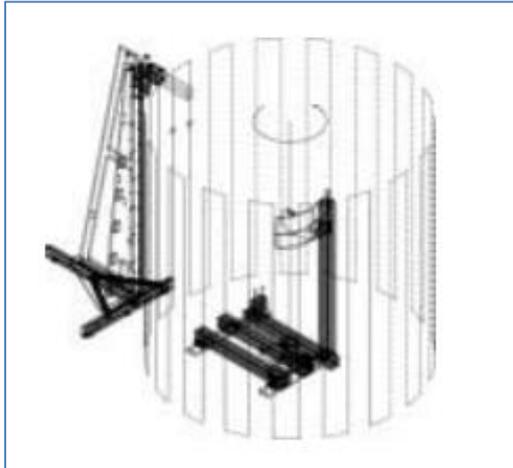
a)



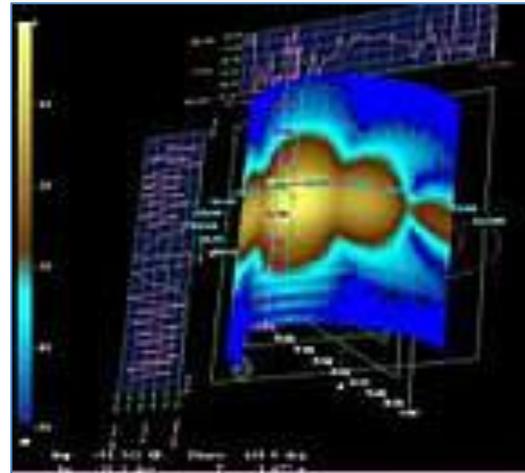
b)

Figura 2.2: a) Sistema de medida en campo cercano plano; b) Diagrama de radiación correspondiente

- Cylindrical Near Field System: este sistema describe un plano cilíndrico alrededor de la antena, como puede verse en la Figura 2.3. El funcionamiento es el siguiente, la AUT está situada en el centro y rota sobre su eje, la sonda de medida se sitúa en un eje paralelo a este y se mueve en vertical mientras la AUT rota. La combinación del movimiento de rotación en azimut de la primera de ellas y el movimiento vertical (en el eje Y) de la segunda conforman un cilindro.



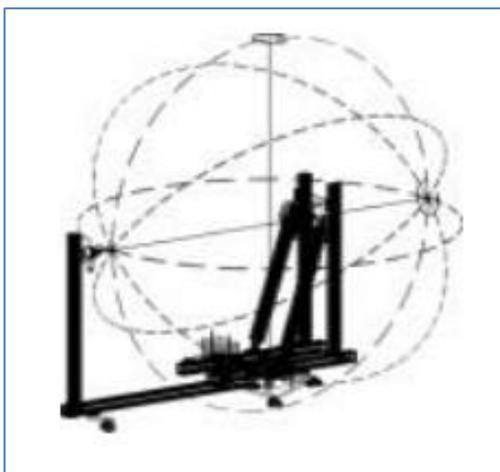
a)



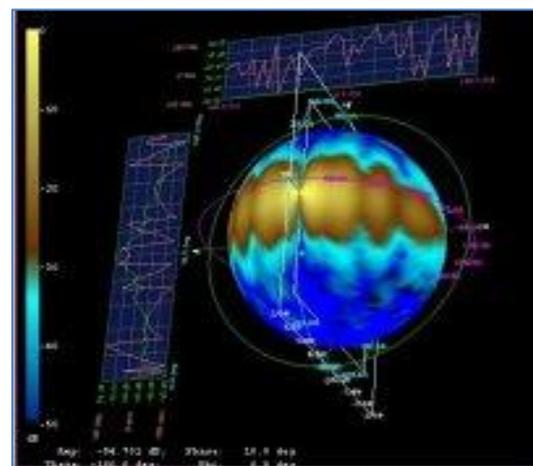
b)

Figura 2.3: Sistema de medida en campo cercano cilíndrico; b) Diagrama de radiación correspondiente

- Spherical Near Field System: se describe un plano totalmente esférico sobre la antena que se quiera medir. La AUT se monta sobre un eje que la rota en roll mientras que la sonda rota en acimut 360 grados alrededor de ella, por cada vuelta de acimut de la sonda la AUT rotara un cierto ángulo ϕ , como se ve en la Figura 2.4. Así, la antena va rotando recorriendo diferentes grados en ϕ y θ y dibujando un plano esférico.



a)



b)

Figura 2.4: Sistema de medida en campo cercano esférico; b) Diagrama de radiación correspondiente



2.3. Bibliografía

1. http://www.keysight.com/upload/cmc_upload/All/NSI-near-far.pdf?&cc=ES&lc=eng
2. www.emscan.com
3. <https://www.nearfield.com/products/NearFieldSystems.aspx>



CAPÍTULO 3. LABVIEW

3.1. Introducción

LabVIEW se define como un entorno de programación gráfica utilizado para desarrollar sistemas o aplicaciones dentro de ámbitos de innovación, investigación y obtención de resultados acelerados. De entre sus principales objetivos destacan la posibilidad de llevar a cabo tanto sistemas hardware como software de medida, pruebas, control y diseño, simulado o real y la disminución acorde al tiempo de desarrollo de las mismas.

En los últimos años LabVIEW se ha utilizado para desarrollar multitud de aplicaciones de medida y control (como se hace en este proyecto) y desarrollar test sofisticados. Esta herramienta de software contiene una amplia gama de características y herramientas con interfaces de usuario configurables, y se diferencia por ser un lenguaje de programación gráfico (conocido como G) con su compilador asociado y sus propias herramientas de depuración.

3.2. Conceptos generales (Programación gráfica)

El lenguaje de tipo gráfico (G) se basa en el uso de íconos gráficos unidos mediante cables. En LabVIEW se trabaja con elementos llamados VIs (Virtual Instruments), correspondientes a textos de código. Cada VI en se compone principalmente de dos partes: el panel frontal y el diagrama de bloques.

- Panel frontal: se corresponde con el interfaz de usuario. Es la parte en la que el usuario puede acceder cuando el programa se está ejecutando. Aquí se pueden crear tanto entradas como salidas del sistema, haciendo que la interacción a la hora de utilizar el programa sea lo más sencilla posible.

Estas entradas y salidas del algoritmo del diagrama de bloques se denominan controles e indicadores, respectivamente. Los controles son modificables por el usuario y pueden ser numéricos, cadenas de texto o incluso botones y pulsadores. El usuario puede elegir cual es el valor de entrada que quiere mandar en la ejecución del programa. Por el contrario los indicadores son valores de la propia ejecución que indican el resultado obtenido (Ver Figura 3.1).

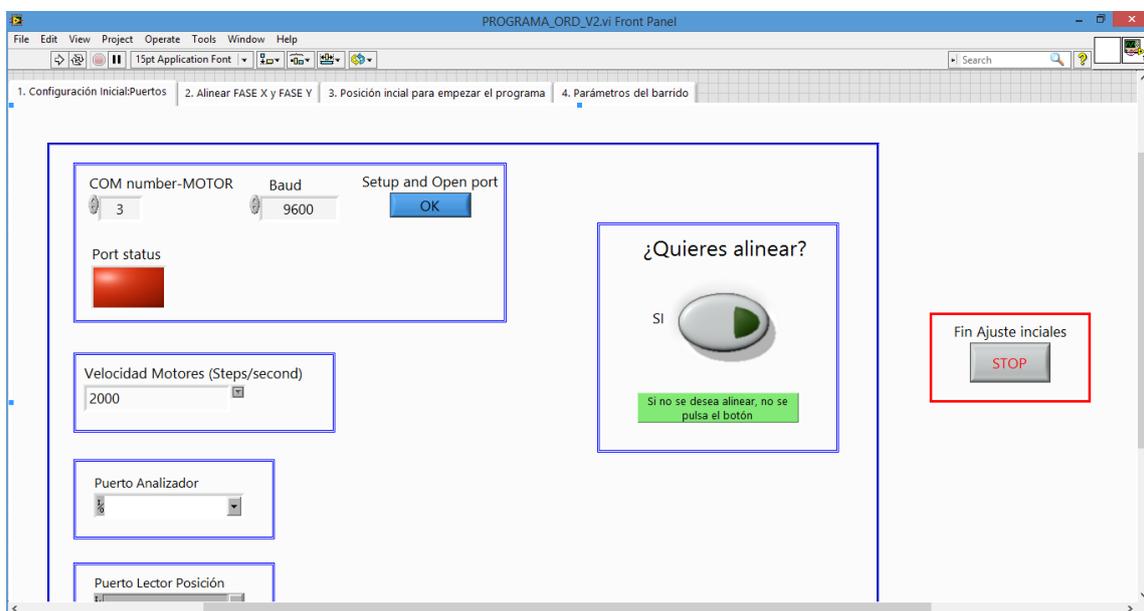


Figura 3.1: Panel frontal

- Diagrama de bloques: En el diagrama de bloques es donde se define la funcionalidad del programa que se está realizando (ver Figura 3.2). Cada uno de los iconos realiza una acción en LabVIEW, y su interconexión permite llevar a cabo el objetivo previsto. Tanto los indicadores como los controles tienen un icono en el diagrama de bloques que además se conecta con un icono que aparece en el panel frontal. De este modo el usuario podrá verlo en la ejecución del programa. Desde esta parte del software se pueden añadir funciones que no sean iconos propios de LabVIEW, como son los VIs. Se puede hacer referencia de unos a otros desde el programa siendo visualmente más fácil trabajar con ellos y dando mayor versatilidad.

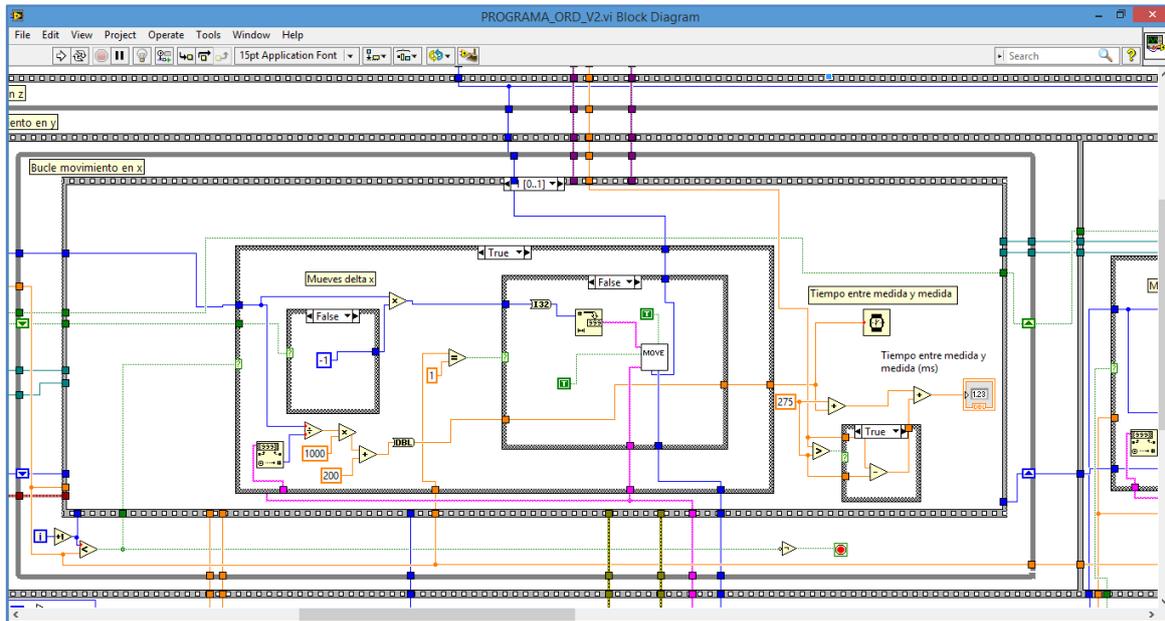


Figura 3.2: Diagrama de bloques

3.3. Características y funciones

Una de las principales características de LabVIEW es su facilidad de uso, siendo utilizado tanto por programadores profesionales como por programadores principiantes. Trabajando con el lenguaje de programación gráfica se puede llegar a realizar programas relativamente complejos pero muy completos. A la hora de desarrollar el código es rápido de implementar y cualquier tipo de usuario se puede beneficiar de ello.

Otra de las características más importantes es el método de programación. Los programas en LabVIEW se llaman instrumentos virtuales (VIs), como se ha explicado anteriormente. Se pueden llegar a crear miles de VIs que se corresponden a millones de páginas de código de texto. Su uso, por lo tanto, es más compacto.

Además LabVIEW facilita la conexión con los interfaces de comunicación de todo tipo: puerto serie, puerto paralelo, GPIB, PXI, VXI, TCP/IP, UDP, USB, OPC, etc... Esta herramienta de software puede interactuar con otros lenguajes de programación (DLL: librerías, .NET, ActiveX, Multisim, Matlab/Simulink, AutoCAD, SolidWorks, etc...), incluyéndolos directamente en su programación y haciendo que la reutilización de códigos ya existentes sea posible. Al permitir el



uso de drivers correspondientes a componentes de medida, analizadores, motores, etc... se facilita el manejo de dispositivos externos a través de un PC (como es el caso de este proyecto).

LabVIEW dispone de otras funciones como la sincronización de dispositivos, programación de FPGAs, tiempo real o control de movimiento.

3.4. NI VISA

3.4.1. Introducción y primeros pasos

El primer paso del proyecto ha consistido en realizar un programa que obtenga la posición marcada por un lector de posición colocado en un sistema de posicionamiento de tres ejes, X, Y y Z.

Para ello se necesita una interconexión entre un PC (donde se encuentra LabVIEW) y dicho componente. Se lleva a cabo mediante el software que proporciona LabVIEW para utilizar dispositivos externos, NI VISA.

3.4.2. Explicación y uso

NI VISA es un software proporcionado por National Instruments que permite la conexión con diferentes instrumentos que comprenden interfaces GPIB, VXI, PXI, serial (RS232/RS485), Ethernet/LXI y/o interfaces USB. Es un estándar para configurar, programar y depurar sistemas de instrumentación. Proporciona la interfaz de programación entre el hardware y los entornos de desarrollo de aplicaciones como NI LabVIEW, LabWindows™/CVI y Measurement Studio para Microsoft Visual Studio. Estos instrumentos pueden ser de todo tipo y fabricante: analizadores, instrumentos de posicionamiento, control...

En el desarrollo de este proyecto se utilizan la conexión Serial-USB para conectar el PC con los dispositivos externos, excepto en un caso, a la hora de conectar el analizador con el PC se ha configurado para tener una conexión vía USB.

Para su utilización es necesaria la instalación del programa NI-VISA que National Instruments permite utilizar si ya dispones de LabVIEW o alguno de sus programas. Este software sirve tanto para 32 como 64 bits.

En LabView, por tanto, se dispone de varios símbolos o íconos que permiten diferentes funciones, comunes para todo tipo de interfaces. Pero también existen algunas conexiones como GPIB que disponen de iconos propios. Dentro de estas las 4 funciones principales y más utilizadas en la ejecución de este proyecto son:

- VISA open: Abre la conexión del puerto especificado (VISA resource name, ver Figura 3.3) indicándole el nombre correspondiente. Devuelve un identificador de sesión para comprobar que el dispositivo está conectado.

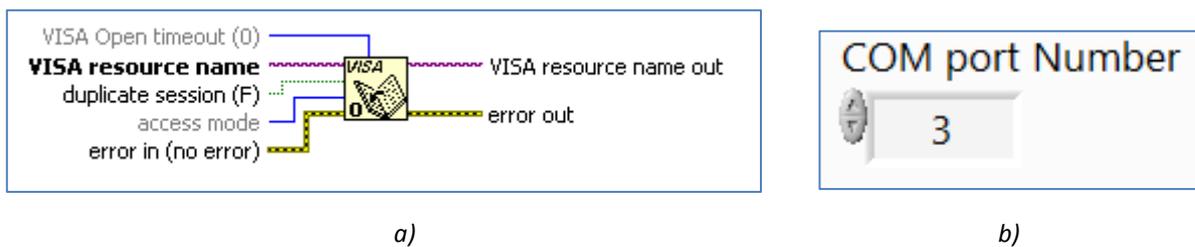


Figura 3.3: a) Visa open; b) Puerto COM

Para que el puerto sea seleccionado y ajustable por el usuario se crea un control en LabVIEW, que permita cambiar en este caso qué puerto COM es el que está conectado.

- VISA close: Cierra la sesión indicada en el VISA resource name. Si este se deja abierto la próxima vez que el programa se ejecute puede dar problemas en la conexión. El puerto puede llegar a no reconocer que algún instrumento está conectado. Se puede ver el icono a utilizar en la Figura 3.4.

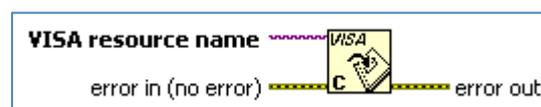


Figura 3.4: VISA close

- VISA write: permite mandar los comandos a través del puerto conectado, como se puede apreciar en la Figura 3.5. De esta manera el instrumento recibe la información necesaria para ejecutar una acción o en otro caso devolver el dato pedido al ordenador o analizador (en el caso de este proyecto).

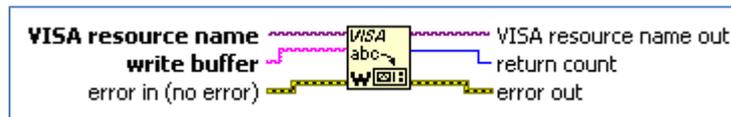


Figura 3.5: VISA write

- VISA read: Se encarga de leer y recibir los datos que está mandando el elemento conectado a un puerto (ver Figura 3.6). Suele utilizarse después del VISA write, para recibir la información que el instrumento está enviando.



Figura 3.6: VISA read

3.5. Bibliografía

1. <http://www.ni.com/labview/esa/>
2. <https://es.wikipedia.org/wiki/LabVIEW>
3. [https://www.ni.com/getting-started/labview-basics/esa/environment#Controls, Indicators, and Constants](https://www.ni.com/getting-started/labview-basics/esa/environment#Controls,Indicators,andConstants)
4. <http://www.ni.com/visa/>
5. http://zone.ni.com/reference/en-XX/help/371361J-01/lvinstio/visa_advanced_subpalette/
6. http://zone.ni.com/reference/en-XX/help/371361J-01/lvinstio/visa_lib_ref_func_descr/



CAPÍTULO 4. COMPONENTES Y SOFTWARE

4.1. Introducción

Este capítulo trata sobre la evolución del software de control del sistema de posicionamiento 3D, utilizado para las medidas de diagramas de radiación RF de antenas que se dan en este proyecto.

También se abordan las versiones previas y actuales de los programas utilizados, así como los resultados a nivel de tiempo de ejecución, interfaz de usuario y otros aspectos. Se explicará qué Virtual Instruments se han reutilizado y cuáles se han creado nuevos, donde se especificarán los cambios significantes.

Para que el proceso de evolución en el software y el funcionamiento del mismo sean totalmente entendibles, se describirán primero todos los componentes que son necesarios en este tipo de medidas.

4.2. Componentes utilizados

En todas las medidas se han utilizado una serie de componentes y dispositivos que se describen a continuación:

- ❖ Posicionador o Estructura motorizada: Se trata una estructura metálica encargada de mover el transmisor o receptor (en conjunto con la antena) dibujando un plano X-Y espacial cuadrado o rectangular. El posicionador actúa según las ordenes de unos motores, que mueven cada uno de los ejes. Este componente corresponde al proveedor *VELMEX*, al igual que los motores que lo mueven.

En la Figura 4.1 se puede ver la estructura utilizada para estas medidas, donde se representan los ejes principales y como se soportan entre ellos. Además, se puede observar la manera de colocar el transmisor o receptor (dibujado en negro) en el propio posicionador. La Figura 4.2 muestra la estructura real donde se realizan las medidas.

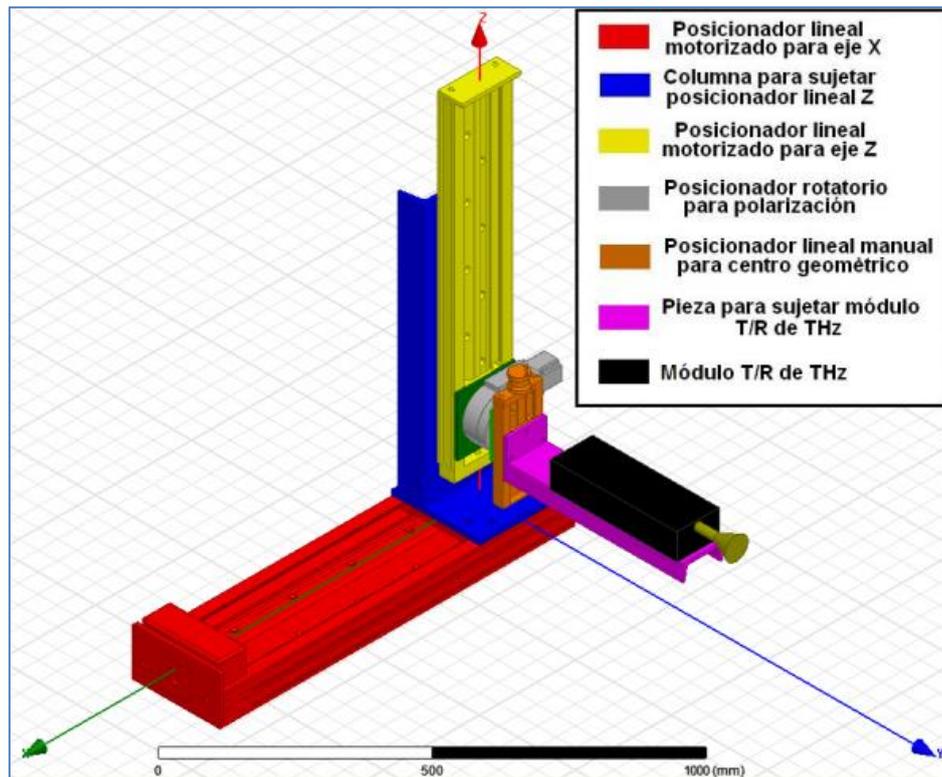


Figura 4.1: Ejes del posicionador



a)



b)

Figura 4.2: Estructura del posicionador

❖ Controladores: existen tres controladores para los diferentes motores del posicionador. Pertenecen al fabricante VELMEX (modelo VXM, ver Figura 4.3) y son los encargados de mover los ejes de la estructura. La comunicación entre el ordenador y estos controladores se realiza mediante una conexión RS232-USB, a través de la que se mandan los comandos necesarios para mover el posicionador **[ver Anexo 2]**.



Figura 4.3: Controlador del motor VXM para eje horizontal y vertical

❖ Transmisores y receptores (cabezales): para poder realizar la medida de diagramas de radiación u otros parámetros (ganancia, toma de imágenes...) de diferentes antenas son necesarios distintos tipos de cabezales, cada uno en una banda frecuencial concreta adecuado a la medida a realizar. Estos transmisores y receptores pertenecen al fabricante VDI. En la Figura 4.4 está representado el transmisor y receptor (realiza las dos funciones) en la banda WR 10, de 75-110 GHz. Mientras la Figura 4.5 corresponde a un receptor en la banda WR 2.2, 330-500 GHz. Sobre estos cabezales se sitúan las antenas transmisoras y receptoras.



a)



b)

Figura 4.4: a) Transmisor/Receptor en banda WR 10; b) Conexiones transceptor banda WR 10



Figura 4.5: Receptor en banda WR 2.2

❖ Lector de encoders: en la estructura del posicionador cada uno de los tres ejes principales tiene unido a él unos encoders que se encargan de recoger la posición exacta en

X, Y y Z. Estos encoders son de la marca *ACU-RITE* (ver Figura 4.6), y son capaces de dar una resolución de 0.5 a 5 μm .



Figura 4.6: Encoder ACU-RITE

La información de estos encoders llega al lector de posición, que se puede ver en la Figura 4.7. Este dispositivo externo, *ND 1100 QUADRA-CHEK (HEIDENHAIN)*, es el que se conecta al ordenador mediante un cable RS232 para obtener la posición en cada punto de medida, puede llegar a una resolución de 1 μm .



Figura 4.7: Lector de posición

❖ Analizador de redes PNA-X N5242A: Es uno de los componentes más importantes en el proceso de medida, es capaz de informar sobre todo tipo de datos correspondientes a la misma como son el sweep time, frecuencia central, datos de magnitud y fase, etc... El analizador utilizado en este proyecto pertenece a *KEYSIGHT (antigua Agilent Technologies)*, se puede ver en la Figura 4.8. El programa software que se ha desarrollado durante este proyecto puede ser ejecutado tanto desde el analizador como desde un PC. Cuando las

medidas se lanzan desde el ordenador se deberán interconectar los dispositivos mediante una configuración USB (también puede ser configurado mediante LAN).

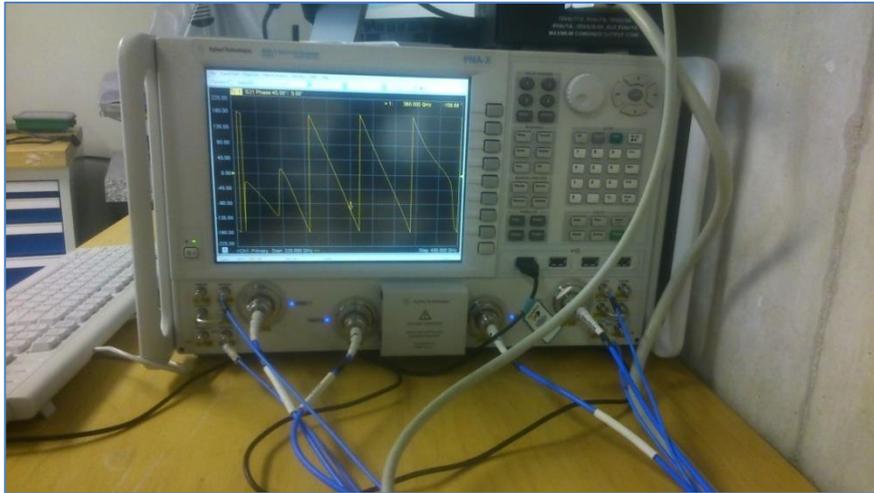


Figura 4.8: Analizador de redes PNA-X N5242A

4.3. Versiones previas

Se ha utilizado como punto de partida del software a desarrollar un código previo disponible. Se compone principalmente de un VI, donde el usuario puede ejecutar las instrucciones necesarias para medir un plano X-Y correctamente, de tal forma que el software se encarga de dar las instrucciones de movimiento al posicionador y las de toma de datos al analizador vectorial de redes.

Este VI principal no es el único, puesto que dentro de él existen varios subVIs, cada uno encargado de una función concreta. Un ejemplo de ellos es el funcionamiento de los motores y el analizador. En cada uno de los pasos que el motor está realizando se cogen los datos del analizador, tanto para la magnitud como para la fase, y esto se realiza con VIs creados específicamente para cada una de estas acciones. El código por tanto es más compacto y se puede reutilizar el código de manera sencilla.

Esta versión inicial estaba preparada para recorrer el eje X en una de las dos direcciones. El motor avanza en el eje recorriendo todos los pasos de la primera línea y cuando ésta acaba

vuelve a su posición inicial antes de pasar a la siguiente línea. De la misma forma lleva a cabo todas las líneas del plano que se esté midiendo.

El tiempo entre medida y medida (de un punto del plano a otro) no es automático, lo introduce el usuario.

La versión principal de este software era utilizada desde el propio analizador (PNA-X N5242A), lo que dificulta la interacción con dispositivos externos, a la vez de recargar en exceso el analizador.

Visualmente el diseño original estaba dividido en dos pestañas. Una, la encargada de la configuración del puerto COM en el que se conecta el motor y posición de partida de la medida (ver Figura 4.9) y la otra, la correspondiente a la longitud de la medida en cada uno de los ejes (“parámetros del barrido”) y la visualización de las gráficas tanto de magnitud como de fase (ver Figura 4.10).

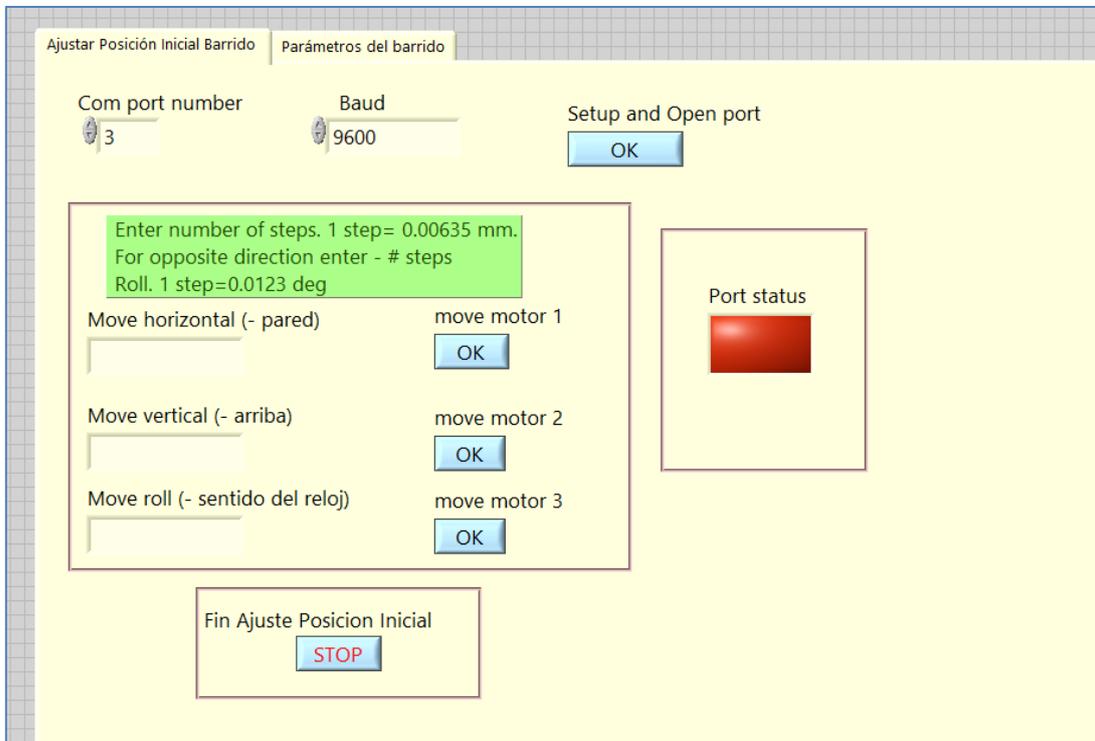


Figura 4.9: Ajuste Posición inicial de barrido

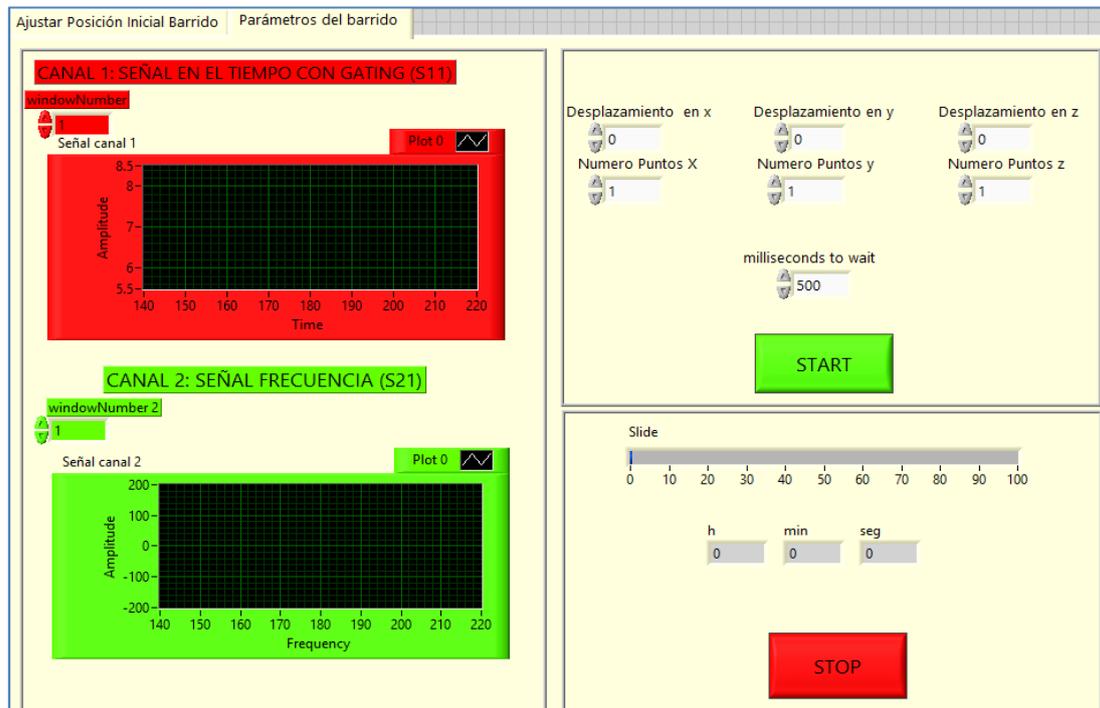


Figura 4.10: Parámetros de barrido

4.4. Versiones actuales y mejoras

4.4.1. Conceptos generales

Una vez descritas las versiones de las que se comenzó, en este apartado se van a abordar las versiones actuales junto con las mejoras que se han hecho en el código y los nuevos elementos introducidos.

Respecto a las versiones se pueden diferenciar dos grupos principales. Uno formado por las versiones que se pueden utilizar desde el analizador y un segundo con las que se utilizan desde el PC. Aunque gran parte del código es el mismo, varias partes de la toma de datos y otros aspectos referentes al analizador ha habido que cambiarlos al introducir un PC que ejecuta el programa externamente.

El código utilizado en el analizador dispone de 1 programa principal. Es una versión que incluye todas las mejoras que posteriormente se explican, excepto una, la adquisición de posición del lector de encoders. Hay una versión, también para el analizador, que sí que realiza

esta acción pero al ser ejecutada desde el analizador no funciona correctamente (error explicado en el apartado 4.6).

En el caso del PC existen hasta tres versiones. Al igual que se ha explicado anteriormente existe una versión cogiendo la posición del lector de encoders y otra sin abarcar esta funcionalidad. La tercera es muy similar, pero en el barrido no realiza un ZIG-ZAG (explicación detallada en el apartado 4.3.2.5) a la hora de recorrer los puntos en la medida del plano seleccionado. La necesidad de utilizar esta versión surge por un problema en las medidas que posteriormente será explicado.

4.4.2. Mejoras introducidas

4.4.2.1. Obtención de la posición (X, Y, Z)

La adquisición de la posición exacta en cada uno de los pasos de medida es el primer objetivo del proyecto, ya que es una de las funciones que posteriormente permite representar los diagramas de radiación con una mayor exactitud.

Uno de los primeros pasos a realizar es la conexión del lector de encoders con el PC o el analizador. Dicha conexión se realiza mediante un cable serial RS232-USB. El lector de posición dispone de un puerto RS232, mientras en el ordenador dispone de un interfaz USB (puerto COM).

El modo por el que el PC detecta que tiene un dispositivo externo conectado es mediante una configuración con NI-VISA, que se ha explicado en el apartado 3.4.

Para la obtención de la posición se ha realizado un VI específico (ver Figura 4.11).

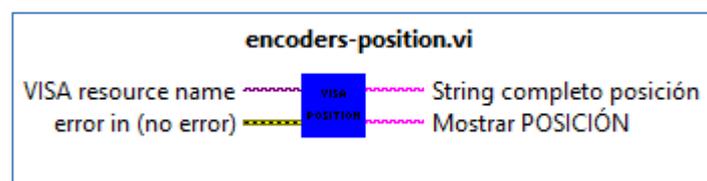


Figura 4.11: "encoders-position.vi"



Este Virtual Instrument se compone de dos entradas principales. El VISA resource name corresponde al nombre del puerto donde está conectado el componente y la segunda entrada pertenece al cable que informa de error, en caso de existir.

Como salidas tiene un string con la posición, que posteriormente se mete en un archivo de texto para ir guardando todos los valores. Por el contrario, la salida “Mostrar POSICIÓN” es un string que se convertirá en un indicador para que el usuario pueda ver en todo momento cual es la posición exacta en la que se encuentra.

El método para obtener la posición de todos los ejes es el siguiente:

1. Mediante la función VISA-write (apartado 3.4) se escribe el comando necesario haciéndolo llegar al dispositivo conectado. En este caso se manda el comando “KEY 16 +intro” **[ver Anexo 1]**, que simula la pulsación del botón que posee el lector de encoders para devolver esta posición.
2. El siguiente paso es leer lo que el lector nos está mandando al recibir ese comando, se hace con el VISA-read. Entre las dos funciones hay un tiempo de espera, es el tiempo que le cuesta al lector recibir el comando y procesarlo. Para una mayor optimización se ha ido ajustando este valor hasta hacerlo lo mínimo posible. Fijando finalmente el valor de 175 ms (ver Figura 4.12).

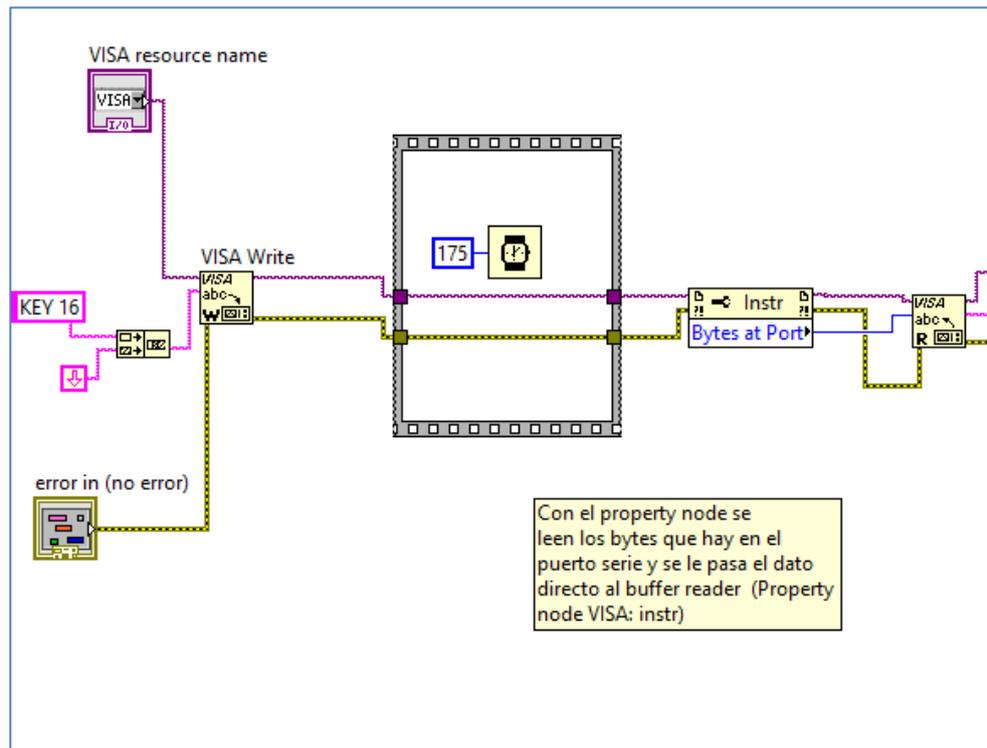


Figura 4.12: Adquisición de posición

3. Para recibir las tres posiciones es necesario ejecutar el VISA read tres veces, uno por cada uno de los tres valores a recibir. Exceptuando el primero, el tiempo de espera antes de ejecutar los otros dos es de 50ms.

En total el tiempo de ejecución de este VI es de unos 275ms aproximadamente. Cada vez que el motor mueva una posición este programa se ejecutará para saber exactamente en qué posición ha parado. Esto supondrá que se pierdan 275ms más en cada punto del plano de medida, pero gracias al sincronismo con otros elementos del programa, como el sweep time, se ha podido suplir en gran parte esa pérdida de tiempo.

4.4.2.2. *Seleccionador de velocidad*

Los motores que mueven el posicionador pueden alcanzar una velocidad de hasta 6000 steps/second (pasos/segundo). Por este motivo se ha incluido un control en LabVIEW que

permita al usuario seleccionar la velocidad deseada para la medida. Las velocidades de los motores de las que se puede disponer son las siguientes: 500, 1000, 2000, 3000, 4000 steps/second.

A la hora de seleccionar una velocidad es necesario saber si ésta puede no ser la adecuada. Cuando se selecciona una velocidad alta, el tiempo que tarda el motor en recorrer los pasos mandados es bastante menor, pero como inconveniente, el posicionador puede incrementar el error. Esto se debe a que la aceleración y deceleración de la estructura es más brusca, vibra moderadamente, y la posición puede llegar a no ser tan exacta.

Este es un problema que no se puede permitir en medidas en las que se necesita mucha precisión, como son las pruebas realizadas en muy alta frecuencia (banda WR-2.2, 330-500 GHz). En estas bandas un error milimétrico por muy pequeño que sea afecta de manera drástica en cada uno de los pasos y puntos medidos.

Al contrario sucede dejando una velocidad baja, como es 2000 steps/second (velocidad por defecto). De esta manera la medida se puede realizar correctamente sin que el error se llegue a producir.

Para incluirlo en el programa ha sido necesario modificar el VI de la versión anterior llamado "MOVER_3motores.vi", donde se ha incluido una entrada nueva que recoge la velocidad seleccionada por el usuario. Este VI, mostrado en la Figura 4.13, es el encargado de mover los pasos seleccionados mediante los motores.

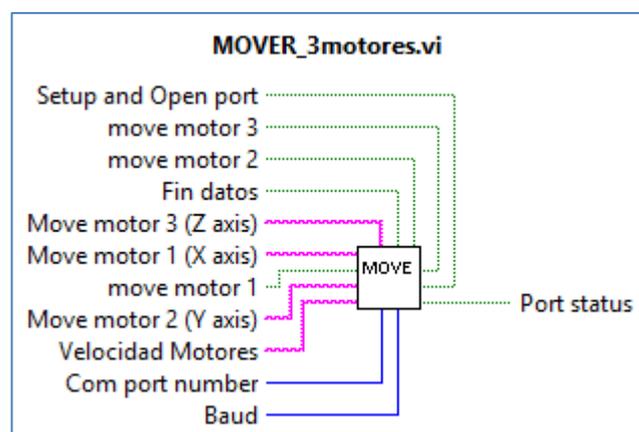


Figura 4.13: "Mover_3motores.vi"



En la mayoría de las medidas realizadas en este proyecto se utiliza la velocidad: 2000 steps/second.

4.4.2.3. Tiempo automatizado

La versión anterior del programa no disponía de esta posibilidad. En ella el tiempo de espera entre la medida de un punto del barrido y el siguiente no era automático, sino que el usuario debía introducirlo según su propio criterio. De este modo no se sabía cuál era exactamente el mejor tiempo para la medida y se podía basar únicamente en el sweep time seleccionado desde el analizador. El tiempo utilizado giraba en torno a los 500-600ms, para medidas con un sweep time menor a 200ms.

En la versión actual el tiempo que espera el programa entre cada punto de la medida está controlado automáticamente, es decir, no se permite cambiar nada. De esta forma el usuario no tiene que llegar a pensar cual será el tiempo correcto para cada medida y el interfaz llega es más sencillo.

El cálculo se ha realizado de forma generalizada para todo tipo de medidas. Estos son los tiempos de referencia en los que se basa:

- 200ms por defecto: Pertenecen a la ejecución del programa y aceleración y deceleración de los motores en cada uno de los puntos de medida.
- El tiempo correspondiente a un tiempo de barrido (sweep time). El valor será el que este seleccionado en el analizador (Explicación en el apartado 4.4.2.4.).
- A esto se le suma el tiempo que le cuesta a los motores realizar el movimiento de un punto al siguiente.

mm (paso)/ velocidad (steps/second)

Según el caso en el que se esté trabajando estos tiempos pueden variar o no. Utilizando el programa que no coge la posición la suma de estos tres valores dará el resultado total.



En cambio, ejecutando el programa con posición, el valor total de este tiempo es diferente. El proceso a seguir depende del tiempo que se tarda en adquirir la posición en comparación con el sweep time.

- Si el tiempo de barrido del analizador es menor que el tiempo en coger la posición el cálculo será:

$$200\text{ms} + \text{mm/velocidad} + 275\text{ms (coger posición)}$$

- Si el tiempo de barrido es mayor entonces se sumará al diferencia de tiempos al resultado total:

$$200\text{ms} + \text{mm/velocidad} + 275\text{ms (coger posición)} + (\text{SweepTime}-275\text{ms})$$

Estos serán los tiempos calculados de forma exacta para cada una de las opciones disponibles.

4.4.2.4. *Sweep time (tiempo de barrido)*

Como se ha visto en el subapartado anterior conocer cuál es el tiempo de barrido es un aspecto muy importante a la hora de elaborar el código.

Para ello se ha creado un subVI nuevo llamado "Sweep_time.vi" mediante el que se obtiene el tiempo correspondiente a un barrido de frecuencias en el analizador de redes.

Este barrido de frecuencias depende de dos parámetros principales, el nº de puntos seleccionados y el IF Bandwidth. Estos dos valores se seleccionan también desde el analizador. Hay que tener en cuenta que a mayor nº de puntos mayor será el sweep time y por tanto las medidas tendrán un tiempo total bastante superior. Se puede compensar cambiando el ancho de banda IF, ya que a mayor valor da un menor sweep time.

Las entradas y salidas para la realización del subVI correspondiente al sweep time se muestran en las Figuras 4.14 y 4.15.

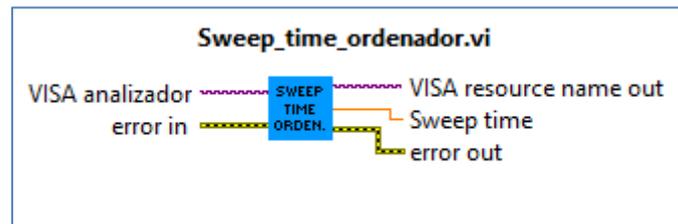


Figura 4.14: Entradas y salidas del VI que obtiene el sweep time desde el ordenador

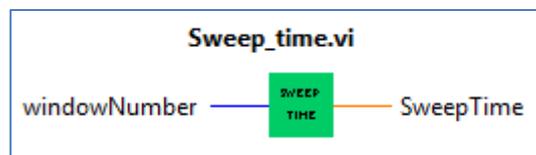


Figura 4.15: Entradas y salidas del VI que obtiene el sweep time desde el analizador

La gran diferencia de estas dos figuras es dónde se llevan a cabo los VI. En el Figura 4.14 se muestran las entradas necesarias cuando el programa es lanzado desde un PC externo (apartado 4.4.2.8), por ello se necesita la entrada de una referencia (VISA analizador) que indique el puerto utilizado. Así es posible conectarse con el analizador y adquirir la salida llamada sweep time.

Por el contrario en la Figura 4.15 no se necesita conexión con el PNA debido a que el programa se ejecuta desde el mismo, sólo hay que pasar como entrada la ventana (canal) donde se está realizando la medida para que devuelve el tiempo de barrido.

Como se ha explicado en el apartado anterior esta versión del software permite utilizar este valor para varias funciones, de hecho toma un papel importante en el sincronismo de la ejecución del programa (apartado 4.4.2.6).

Además, existe un indicador (ver Figura 4.16) en el panel frontal del código, donde se puede visualizar el valor del sweep time. El usuario puede conocer en todo momento este tiempo, y se evita el problema de que la medida sea lanzada con un tiempo de barrido erróneo, por ejemplo.

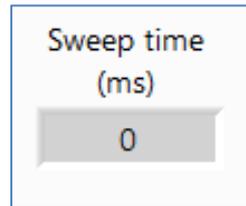


Figura 4.16: Indicador sweep time

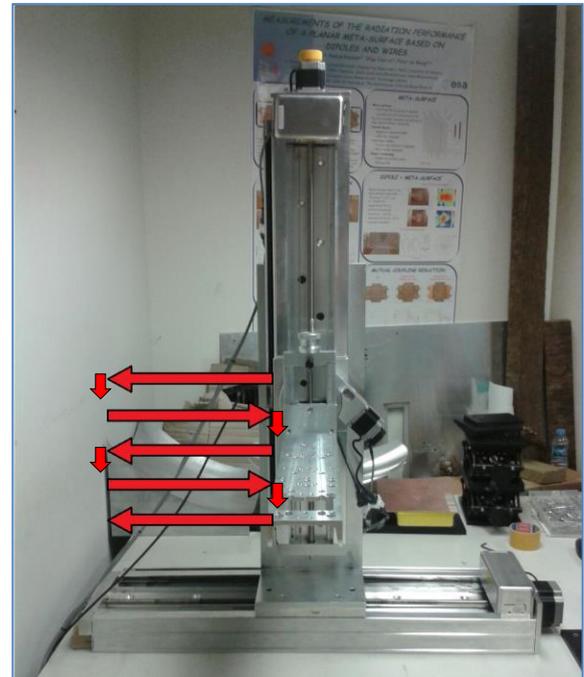
4.4.2.5. Corrección en el barrido de puntos

La corrección en el barrido de puntos es uno de los aspectos más destacables de las versiones nuevas. En la versión anterior la manera de recorrer los puntos de un plano de medida era en una sola dirección, es decir, de izquierda a derecha o al revés. Cuando en el eje X (horizontal) acaba una línea el posicionador vuelve hasta la posición inicial, en la que ha comenzado la medida. Al llegar a esta posición se mueve un punto en el eje Y (vertical) y vuelve a recorrer una línea en horizontal.

Lo que se introduce con esta mejora es la posibilidad de hacer un barrido en ZIG-ZAG cuando se realiza la medida. De esta forma, al recorrer una línea en el eje horizontal no hace falta que los motores muevan el posicionador hasta la posición inicial, sino que bajan una posición en el eje vertical directamente. Para explicar con mayor claridad este proceso la Figura 4.17 muestra una comparación de cómo se recorre el barrido de puntos en las dos situaciones.



a)



b)

Figura 4.17: a) Barrido versión previa; b) Barrido versión actual

Durante el proceso de este proyecto, y en gran medida al realizar los diferentes diagramas de radiación, ha surgido un problema de vital importancia debido principalmente a la introducción de esta mejora. Al dibujar los diagramas correspondientes a las medidas que se habían realizado surgía un entrelazado en el eje horizontal en las imágenes representadas, tanto para la magnitud como para la fase. Este aspecto será comentado de manera detallada y con sus posibles soluciones en el apartado 5.4.

Como conclusión, cabe destacar el ahorro considerable de tiempo en cada una de las medidas debido al no retroceso a la posición inicial en cada una de las líneas.

4.4.2.6. Sincronización y ejecución del programa

Se pueden diferenciar dos grupos de programas principales en torno a sincronización y ejecución del programa, uno referente al código que incluye detectar la posición (X, Y, Z) del lector de encoders y otro que no implementa esta función.



Respecto al primero, se sigue el orden detallado a continuación para llevar a cabo las acciones y funciones del programa:

- Primero, se coge la posición del lector de posición (lector de encoders), esta acción tarda 275ms. En este paso del proceso hay que asegurarse de que los motores que mueven el posicionador estén parados en una posición exacta (no se puede realizar si todavía están en movimiento, porque la posición no sería correcta).
- Segundo, se adquieren los datos del analizador, tanto para la magnitud como para la fase. En este punto hay que tener en cuenta si el tiempo de 275 ms, correspondiente a la lectura de la posición, es mayor al sweep time o menor (tal y como se ha explicado en apartados anteriores).

En caso de ser mayor, no se añade un tiempo adicional. Si sucede al contrario, se restará: sweep time – 275ms. Y esta diferencia es el tiempo que hay que esperar en el programa para poder adquirir los datos.

- Tercero, se mueve el motor: Se moverá la cantidad de mm que sean seleccionados por el usuario. Tras mover el motor espera un tiempo:

200ms + steps/velocidad (steps/second)

Este tiempo se ha introducido en la ejecución para poder sincronizar adecuadamente el momento en el que el motor está parado. Para ello, se suman los 200 ms (tiempo que le cuesta al motor acelerar y decelerar) más lo que le cuesta al motor realizar el paso seleccionado. Esperando este tiempo te aseguras de que cuando se va a adquirir la posición y los datos del analizador, el motor esté completamente inmóvil.

Pueden surgir ciertos problemas de sincronismo debido a la mecánica de los motores que mueven el posicionador. En ciertas ocasiones, si las órdenes de movimiento son demasiado rápidas, parece que al posicionador le cuesta reaccionar para dar los siguientes pasos y es por esto que no es recomendable poner tiempos entre medida y medida menores a 400ms aproximadamente.

El segundo caso tiene un orden de acciones muy similares, el único aspecto que cambia es la posición. En esta versión no hay recogida de posición y por ello el primer paso se elimina, pasando a ser la primera acción la de esperar un tiempo de barrido. Al esperar un sweep time asegurándose que el motor está parado (paso 3), los datos cogidos del analizador serán totalmente correctos.

4.4.2.7. *Alineación de antenas*

A la hora de medir un plano X-Y de una antena uno de los aspectos más importantes es la alineación de la misma.

Esta mejora da la posibilidad al usuario de poder decidir si quiere alinear antes de mandar la medida o no (ver Figura 4.18).

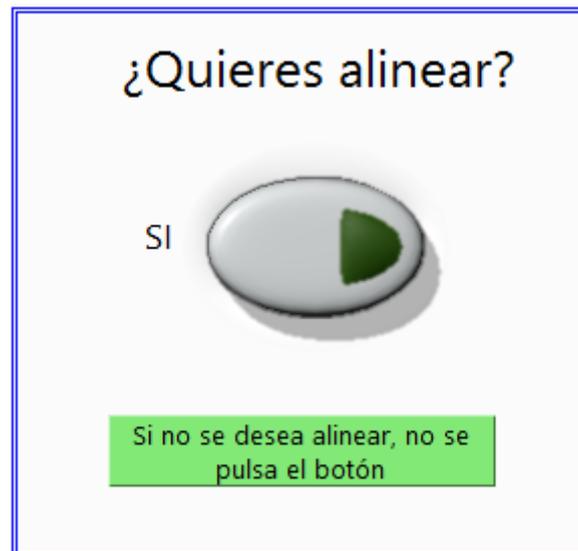


Figura 4.18: Interfaz de usuario con posibilidad de alineación

Si se selecciona este pulsador, el usuario deberá dirigirse al panel frontal donde puede mandar una medida de una línea completa, indicando que paso entre punto y punto desea y la cantidad total de puntos a obtener (ver Figura 4.19).

Aunque la alineación de las antenas se realice manualmente, en el software se han incluido dos gráficas nuevas que representan la fase en el eje X y en el eje Y. Estas dos gráficas dibujan una línea de puntos en uno de estos dos ejes. En cada punto de la medida se hace un barrido de frecuencias, de las que nos quedamos con la frecuencia central. Por supuesto estos datos son recogidos para la fase, que es la medida más influyente en la alineación de antenas.

El objetivo de estas dos gráficas es obtener una representación de la fase donde se aprecie si ésta es simétrica respecto al punto central, es decir, que la fase actúe de igual forma hacia puntos positivos y negativos. Como se puede observar en la Figura 4.19, los ejes horizontales de las gráficas muestran los mm recorridos simétricamente de números negativos a positivos. El eje vertical indica la amplitud de fase en grados en cada una de estas posiciones.



Figura 4.19: Gráficas de Fase para la alineación de antenas

Para la realización de esta función se han añadido dos subVI ("Grafica_PNA_FASE_X.vi" y "Grafica_PNA_FASE_Y.vi") al programa inicial. Son los encargados de coger todas las frecuencias para cada punto y separarlas hasta quedarse únicamente con la frecuencia deseada (central).



Cada uno de estos valores se va introduciendo en un array de forma que los puntos se vayan representando en el mm correcto en el mismo instante en el que se están midiendo.

4.4.2.8. Implementación en el PC

En los siete apartados anteriores se han explicado todas las mejoras introducidas en las nuevas versiones de software realizadas. Estas mejoras se implementan tanto en las versiones utilizadas desde el analizador como en las ejecutadas desde un PC.

Sabiendo que LabVIEW suponía una carga excesiva para el analizador se decidió instalar un ordenador donde ejecutar los programas de manera externa. Sin embargo, algunas de las funciones que llevan a cabo los programas de este proyecto tenían un resultado negativo si eran ejecutadas desde el propio analizador. Por ejemplo, el primer objetivo (recogida de posición del lector de encoders), no se podía conseguir desde el analizador debido a un error al ejecutar el código y sin embargo, la instalación del ordenador ha permitido realizar esta función.

Al instalar el ordenador, el principal inconveniente viene dado por la conexión con el analizador. Los códigos que se habían realizado ya no son válidos para ser usados desde el PC. Por ello las versiones de los VI: “sweep_time.vi”, “Grafica_PNA_FASE_X.vi”, “Grafica_PNA_FASE_Y.vi”, “toma_datos_PNA.vi” se han sido realizado de nuevo para un uso externo. Al no haber acceso directo al analizador se requiere una conexión de interfaz (en este caso configurada con USB) para poder obtener todos los datos relacionados con él.

La conexión entre el ordenador y el analizador de redes se lleva a cabo mediante un interfaz USB, siendo el nombre del puerto correspondiente “USB_analizador” (ha sido cambiado para no dar confusión a la hora de seleccionar el puerto). En todos los programas del PC hay un control realizado con LabVIEW que deja elegir este puerto para conectarse con el analizador, la transferencia de datos en ambos sentidos se ha hecho mediante una sesión con NI VISA. De manera similar a casos comentados anteriormente, se utilizan las funciones VISA_write y VISA_read para mandar los comandos correctos según cada objetivo propuesto.

Como ejemplo y para ver una diferencia entre el código que se necesita en el analizador y el que se necesita en el PC, se representa en la Figura 4.20 la programación que se ha llevado a

cabo en el ordenador y en la Figura 4.21 la que se hace desde el analizador. Las dos indican la forma en la que se coge el valor del sweep time. Se puede observar que en la primera se conecta el ordenador con el analizador mediante NI VISA y se le manda un comando ("SENSe1:SWEEp:TIME?") para que el analizador entienda que se está pidiendo. En el segundo caso esto es más sencillo, te deja seleccionar directamente que quieres obtener.

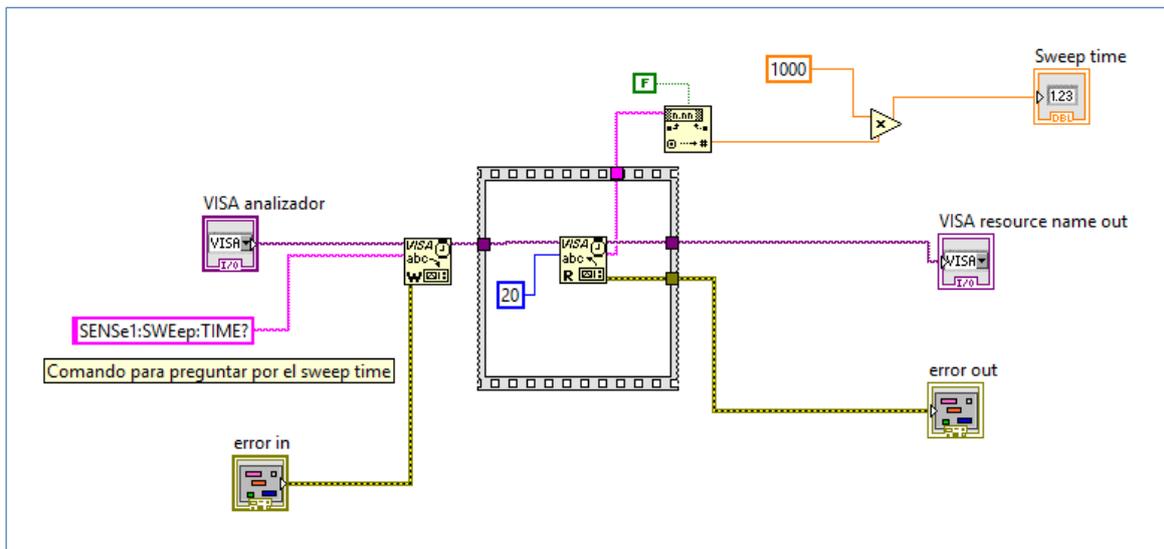


Figura 4.20: Código de LabVIEW (PC) para obtener el valor del sweep time

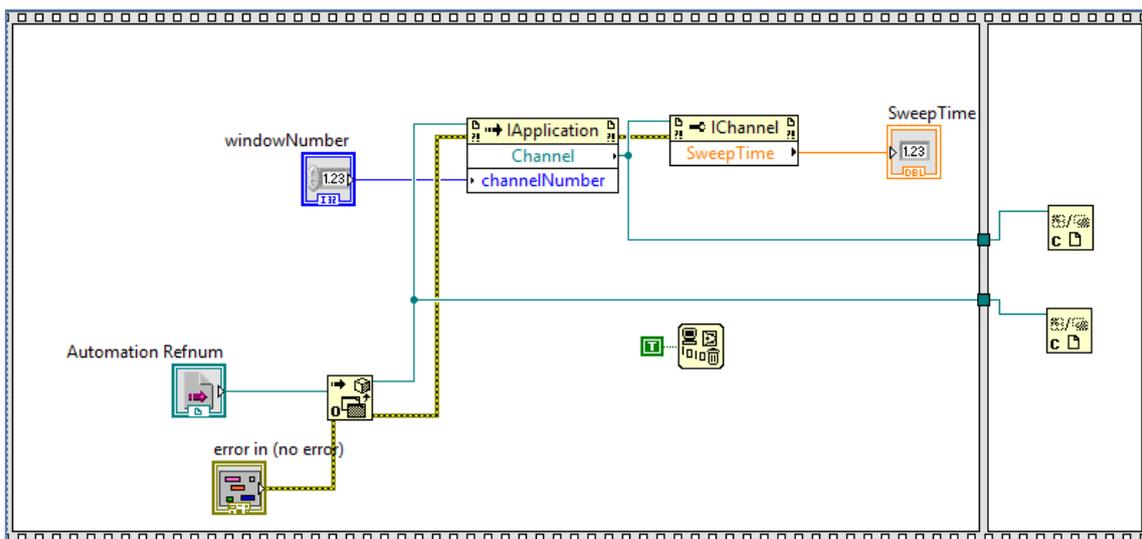


Figura 4.21: Código de LabVIEW (Analizador PNA) para obtener el valor del sweep time



4.5. Análisis y resultados

Las mejoras descritas en apartados anteriores se han llevado a cabo por diversos motivos. A parte de obtener la posición, se quiere conseguir un ahorro significativo de tiempo en el del valor total de la medida realizada. Se ha mejorado también el interfaz de usuario, haciéndolo más accesible e incluyendo nuevas funcionalidades.

4.5.1. Resultados conforme al tiempo de ejecución

Para hacer una comprobación correcta de los cálculos, se ha utilizado una medida de una antena en banda WR 2.2, donde las medidas suelen tardar más tiempo debido al pequeño valor de λ . Se ha llevado a cabo dicho plano espacial con las dos versiones, antigua y mejorada. En este caso, recorre 200mm en el eje X y 200mm en el eje Y. Esto se corresponde a 571 puntos en cada una de las líneas de los dos ejes. El paso de medida teniendo estos valores será de 0.35mm.

Utilizando la versión anterior se usa un tiempo entre medida y medida del analizador, es decir, entre punto y punto del plano seleccionado de **500 ms** aproximadamente. El analizador en esta medida tiene configurado un tiempo de barrido de 185.523ms por lo que el tiempo automático con la versión nueva (sin adquisición de la posición) es el siguiente:

$$200ms + 185.523ms(\text{sweep time}) + 27.56 ms = 413ms$$

Los 27.56ms corresponden al tiempo que le cuesta al posicionador mover esos 0.35mm:

$$0.35mm * \frac{1 \text{ step}}{0.00635mm} * \frac{1 \text{ second}}{2000 \text{ steps}} = 27.56ms$$

Como se puede observar hay un gran ahorro de tiempo, $500-413ms=87ms$, en cada recogida de datos del analizador. Por supuesto esta gran mejora se debe también a que estos datos son para una medida que no opta por coger la posición exacta del lector de encoders.



Sabiendo que son $571 \cdot 571 = 326.041$ puntos totales a medir, una diferencia tan mínima como 87ms puede llegar a ser de vital importancia.

$$87ms * 326041 \text{ puntos} = 28365567ms = 7.88 \text{ horas menos de medida}$$

Otro aspecto a destacar es la introducción del ZIG-ZAG (explicado en el apartado 4.4.2.5). Después de medir una línea en horizontal el programa se ahorra la cantidad de tiempo que le costaba al posicionador volver hasta ese punto inicial de medida.

$$200mm(1 \text{ línea}) * \frac{1 \text{ step}}{0.00635mm} * \frac{1 \text{ second}}{2000 \text{ steps}} = 15.74 \text{ s}$$

Esta medida al tener 571 líneas supone un ahorro total de:

$$571 * 15.74s = 8987.54s = 2.5 \text{ horas}$$

Como tiempo total de ahorro en la medida se tienen $7.88 + 2.5 = 10.38$ horas aproximadamente. Al haber medido este plano X-Y con la versión antigua y los 500ms seleccionados por el usuario se conoce que el tiempo total de la ejecución del programa ronda en torno a las 60 horas totales.

$$\frac{10.38h}{60h} = 0.173 * 100 = 17.3\% \text{ de ahorro en tiempo}$$

Este ahorro (4.16% correspondiente a la corrección en el barrido de puntos y 13.14% correspondiente al menor tiempo entre punto y punto) de tiempo es significativo, ya que ahorrar 2 horas de 10 o 10 horas de 50, es bastante cuando se necesita realizar más de una medida.

También se puede hacer este cálculo con las otras versiones del código. En el caso de la versión que incluye todas las mejoras, es decir, utiliza ZIG-ZAG y adquisición de posición, el paso entre punto de medida y medida sería:

$$200ms + 275ms(\text{sweep time}) + 27.56ms = 502.56ms$$

Con esta versión tardará 2.56ms más en cada uno de los puntos del plano.

$$2.56ms * 326041 \text{ puntos} = 834664.96ms = 0.23 \text{ horas más de medida}$$



Pero al igual que antes, con el ZIG-ZAG se tienen 2.5 horas de ahorro lo que hace un total de 2.27 horas de ahorro respecto al tiempo total de medida.

$$\frac{2.27h}{60h} = 0.0378 * 100 = \mathbf{3.78\% \textit{ de ahorro en tiempo}}$$

Y por último, en la versión que no realiza ZIG-ZAG (pero si coge posición) sucede lo mismo que en el caso anterior, pero sin tener el ahorro de 2.5 horas que aporta la corrección en el barrido de puntos. En este caso, se puede decir que el tiempo total de medida no cambia prácticamente nada al que ya existía, pero está realizado a propósito por un problema que ha surgido durante el proyecto y que posteriormente se describe en el apartado 5.4.

4.6. Problemas en el desarrollo

Durante la creación del código correspondiente a las mejoras descritas anteriormente se han ido presentando una serie de problemas que no permitían avanzar en su desarrollo.

SOFTWARE NI VISA:

Uno de los más habituales tiene relación con el software NI VISA. En muchas de las ocasiones, para conectarse con dispositivos externos a través de este software, surgía el inconveniente de no poder hacerlo y no comprender cuál era la manera correcta de llevarlo a cabo. Utilizando esta herramienta desde el propio analizador, aun estando todo correctamente instalado al igual que en el PC, la función de recogida de posición no se podía usar debido a unos problemas relacionados con actualizaciones de firmware del ordenador o falta de librerías. Además, si las conexiones no se realizan en el orden correcto o por algún casual la conexión en ese puerto no estaba cerrada completamente, no se podrá iniciar el programa y este llevará a error nada más ejecutarlo. También pueden surgir estos problemas con la versión del software NI VISA o drivers.



TIEMPOS DEL PROGRAMA:

Otro gran problema ha sido sincronizar el tiempo de ejecución del programa, en gran parte debido a la recogida de posición del lector de encoders. Al mandar un comando a un dispositivo conectado externamente hay cierto retardo antes de recoger los datos, y por este motivo ha sido un inconveniente sincronizar el subVI encargado de esta acción. En ciertas ejecuciones del programa la recogida de los datos era incorrecta, es decir, en la posición de X se cogía la posición de Z, en la posición Y aparecía la de X y así continuamente. Los datos eran guardados de manera incorrecta en el archivo y su posterior utilización para una buena representación era imposible. Se probó con diferentes tiempos de espera en cada una de las posiciones que había que recoger, hasta que se comprobó que en la primera recogida de posición (posición X) se necesitaba un mayor tiempo para funcionar correctamente, como se ha comentado en apartados anteriores.

ERRORES E INCONVENIENTES EN LA UNIÓN ANALIZADOR-PC:

Al conectar el analizador con el ordenador para ejecutar LabVIEW externamente, han surgido ciertos inconvenientes derivados de los datos que había que adquirir en la medida. Los comandos necesarios para obtenerlos y el propio proceso según cuál sea el objetivo (sweep time, frecuencias, datos de magnitud, fase, etc...), son diferentes.

En el propio driver, que hay que incluir en los directorios de LabVIEW, vienen cerca de 100 iconos para realizar todas las funciones, y no aparecen del todo explicadas las propiedades de cada uno de ellos. Ha sido difícil encontrar exactamente cuáles utilizar para la recogida de datos de la medida.

Lo más problemático viene dado por la obtención de datos de magnitud y fase. El analizador puede contener diferentes parámetros a medir, de modo que cuando accedes a él tienes que enviarle el nombre de qué medida vas a coger esos datos. Esto se ha ido descubriendo poco a poco debido a los errores (en este caso errores -420, -114 y 103) que daba el analizador, que indicaban que es lo que estaba pasando.

Gracias a estos errores se concluyó que el primer paso a llevar a cabo consiste en seleccionar la medida mediante este comando: "CH1_S11_1". Se trata del nombre por defecto



que utiliza el analizador sea cual sea el tipo de dato a obtener. El segundo paso es seleccionar que tipo de datos se necesitan, si magnitud o fase, y el último es incluir un ícono encargado de coger los datos de un buffer (este icono funcionaba correctamente, suponiendo que siempre es el mismo). Este último paso es el que más dudas generaba, puesto que había diferentes iconos a utilizar para la adquisición de datos y no se encontraba claramente cuál realizaba esta función.

4.7. Bibliografía

1. <http://www.velmex.com/index.asp>
2. <http://www.acu-rite.com/senc150.html>
3. <http://www.danielmunoz.com.ar/blog/2009/04/30/ejemplos-labview/>
4. http://www.keysight.com/owc_discussions/message.jspa?messageID=51258
5. <http://www.ni.com/tutorial/9860/en/>
6. <http://www.ni.com/white-paper/2804/en/>
7. <http://www.ni.com/webcast/1434/en/>
8. <http://www.keysight.com/main/software.jsp?cc=ES&lc=spa&nid=-11143.0.00&id=1857238>
9. <http://www.keysight.com/main/editorial.jsp?ckey=1483532&id=1483532&nid=-11143.0.00&lc=fr&cc=CA>
10. http://www.keysight.com/owc_discussions/thread.jspa?messageID=100256
11. http://na.support.keysight.com/pxi/help/latest/Support/SCPI_Errors.htm



CAPÍTULO 5. MEDIDAS

5.1. Introducción

Este capítulo va a abarcar todos los aspectos relacionados con las medidas que se han llevado a cabo durante el proyecto. Estas medidas han servido de validación del software y de las mejoras realizadas.

Se va a analizar la metodología a seguir para realizar una medida totalmente válida. Además, se comentarán todos los resultados obtenidos y algunos problemas e inconvenientes que han ido surgiendo en el proceso.

5.2. Metodología

Antes de realizar una medida de un plano espacial X-Y cualquiera es necesario llevar a cabo una serie de pasos en un orden concreto:

1. Configuración del set-up: El primer paso es colocar la estructura del posicionador así como los transmisores y receptores (o transceptores) de la manera adecuada para realizar la medida. Para que el diagrama de radiación sea totalmente válido, la alineación de las antenas debe ser casi perfecta. Tanto la estructura metálica del posicionador, que sujetará al transmisor (por ejemplo), como la placa metálica donde se sitúe el receptor deben estar situadas en paralelo con un mínimo margen de error.

Los brazos de la estructura que sujetan al transmisor también deben estar nivelados con la horizontal, es decir, que el brazo que sujeta este cabezal no se doble en gran medida. Se puede ver en la Figura 5.1 un ejemplo sobre cómo se pueden situar los componentes en una medida.



Figura 5.1: Ejemplo de sistema de medida

En la colocación de estos componentes, la distancia entre el transmisor y el receptor debe ser la máxima distancia posible para minimizar el acoplo entre sonda de medida y AUT (al menos más allá de la frontera con la región de Rayleigh o de campos reactivos) y por otro lado se ha de tener en cuenta que el tamaño del campo plano a medir ha de ser lo suficientemente grande como para no truncar significativamente los campos. A más distancia de medida más grande será el tamaño de la medida y más puntos serán necesarios a la hora de muestrearlo, ya que el muestreo tiene un valor límite de paso máximo que es $\lambda/2$. Por lo general, el plano de medida ha de extenderse hasta que el campo en los extremos caiga al menos 30 dB respecto al del centro, aunque 40 dB de truncamiento máximo dará mejores resultados.

2. Seleccionar parámetro a medir: Una vez que el sistema se ha situado correctamente, se eligen ciertos parámetros desde el analizador. Por ejemplo se decide si la medida va a ser del S21, S21, S11, etc... según cual sea la necesidad. Habitualmente en antenas mediremos el S21 o el S12 y entre ambos, donde normalmente la AUT radia y la sonda recibe ya que es más fácil cubrir de material absorbente la sonda porque siempre será la misma, mientras que la AUT es diferente cada vez.

También se suele aplicar un enventanado en el dominio del tiempo “time gating”, para eliminar las reflexiones de los objetos que estén lejos de la sonda y tener que preocuparnos menos de cubrirlos de material absorbente, para ello se transforma la señal en el dominio de la frecuencia mediante una FFT al dominio del tiempo, se enventana la respuesta principal y se transforma de nuevo al dominio de la frecuencia.

3. Alinear: para saber si hay que alinear las antenas primero hay que buscar la posición del máximo de radiación, posteriormente se mira la magnitud y fase moviendo la sonda en el eje horizontal y vertical. Para que el resultado sea bueno y el plano de medida esté bien alineado y absolutamente perpendicular a la dirección de propagación, es conveniente analizar con detalle lo que ocurre en la fase. Se puede medir una línea del eje X viendo si la fase se comporta igual de $x=0\text{mm}$ a mm positivos y de $x=0\text{mm}$ a mm negativos (por ello se ha introducido la mejora 4.3.2.7, mayor comodidad). De esta manera se va ajustando la fase modificando angularmente el plano de medida tanto en horizontal como en vertical hasta que se comporte de la misma forma hacia los dos lados del máximo o mínimo, es decir, siendo simétrica (ver ejemplo en la Figura 5.2). Se suelen realizar estos pasos conforme a la frecuencia central de la medida. Para el eje vertical (Y), el procedimiento de alineación será el mismo.

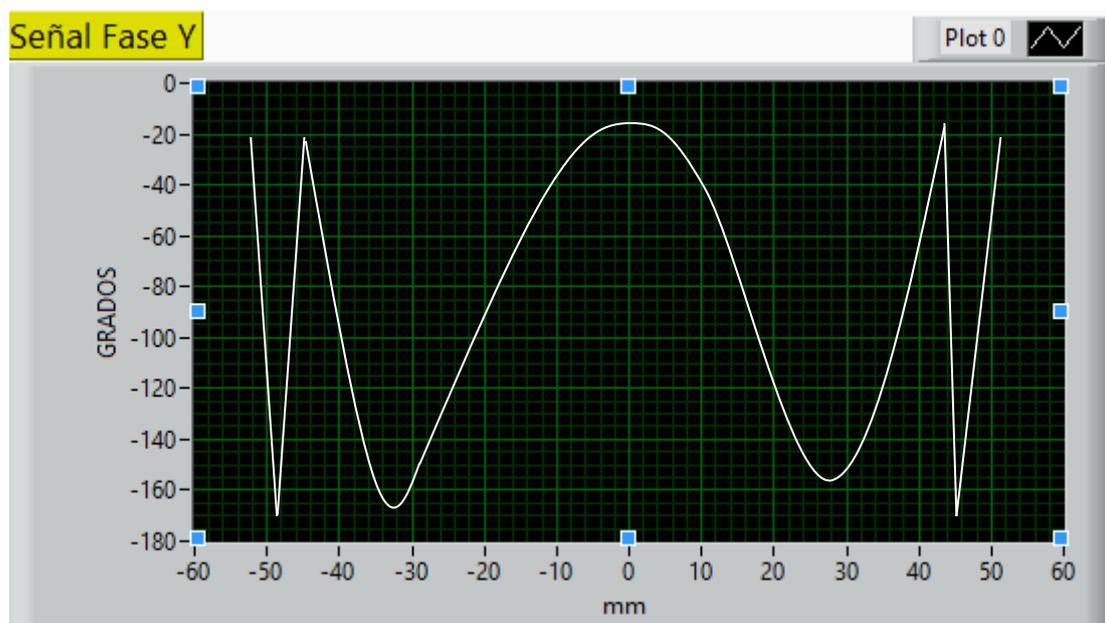


Figura 5.2: Gráfica de fase para el eje X

4. Ajustar últimos parámetros: se ajusta el sweep time, que depende principalmente del IF Bandwidth y del número de puntos que se quieran adquirir en un barrido de frecuencias. Un sweep time adecuado puede estar en torno a los 200-250ms para que la medida sea lo más rápida posible. Cuanto más bajo sea el IF bandwidth más precisa será la medida, pero mayor será el sweep time.

Previamente a lanzar la medida es interesante no olvidarse de poner el máximo del diagrama de radiación (punto central del plano X-Y) en 0dB para que la posterior representación sea normalizada. Se implementa utilizando la función del analizador Data-Memory.

5. Lanzar la medida: Una vez todos los parámetros están seleccionados se ejecuta uno de los programas disponibles en LabVIEW. Se sitúa el cabezal (unido al posicionador) en la posición inicial de medida, que suele ser en un punto extremo de la parte superior, como se puede ver en la Figura 5.3.



Figura 5.3: Punto inicial de barrido de un plano de medida



6. Representación con Matlab: se trata de hacer una representación del diagrama de radiación para la magnitud y para la fase en las diferentes frecuencias correspondientes a la banda de la medida. Se utiliza un programa que calcula el diagrama de radiación en campo lejano a partir de la medida en campo cercano. Para ello hay que introducir los archivos que llevan los datos de magnitud y fase e indicar cuál es el tamaño del plano que se ha medido y el número de puntos escogido. En este proyecto se ha intentado desarrollar un programa en Matlab que fuese capaz de dibujar el diagrama de radiación en función de la posición de cada uno de los puntos de medida (que se recoge con la mejora 4.4.2.1) para hacer los resultados más exactos.

5.3. Análisis de los resultados

En el transcurso de este proyecto se han realizado diferentes medidas, utilizando tanto las versiones antiguas como nuevas del software. A continuación, se van a describir los resultados obtenidos para tres tipos de medidas en diferentes bandas de frecuencias.

Se han llevado a cabo tres medidas principales:

- Medida del campo radiado por una antena que escanea en frecuencia a la distancia donde se toman las imágenes. La antena consta de un array de bocinas que escanea en frecuencia y un reflector parabólico offset para concentrar la energía. Funcionaba en la banda WR-10 (75-110GHz) y se midió el diagrama de radiación en campo cercano entre 75 y 85 GHz.
- Bocina y Lentes: banda WR-10 (75-110GHz), medida del diagrama de radiación que producen las antenas de bocina estándar de Flann Microwave con y sin lente dieléctrica en su apertura. Las lentes fueron fabricadas con impresora 3D.
- Array 8 antenas: banda WR-2.2 (330-500GHz), medida de diagrama de radiación para las antenas 1 y 4 del array.

5.3.1. Antena que escanea en frecuencia

El sistema que se va a describir a continuación está preparado para escanear una cadena de producción agroalimentaria tomando imágenes de los diferentes productos que son transportados por la cadena y en la que se van analizando los mismos en busca de partículas extrañas o no deseadas que no se puedan detectar a simple vista.

En esta medida se trabaja en la banda frecuencial WR-10, pero utilizando como $F_{min}=75$ GHz y como $F_{max}=85$ GHz. La frecuencia central utilizada será 80.2 GHz, donde el lóbulo principal sale perfectamente centrado en la representación.

Inicialmente se sitúa toda la estructura, con la distancia adecuada entre el transmisor y receptor, que en este caso es 50 cm del centro de reflector, como se puede ver en las Figuras 5.4 y 5.5.



Figura 5.4: Estructura del sistema de medida con el reflector parabólico



Figura 5.5: Sujeción de la estructura nivelada

En este sistema la antena transmisora que es realmente un frequency scanning array, emite una señal que se refleja en el reflector parabólico, haciendo así que llegue al receptor. En este proceso se ha medido el S21.

Se calcula λ para la máxima frecuencia, 85 GHz en este caso. El valor es $\lambda=3.53\text{mm}$, y la separación entre punto y punto de medida tiene que ser al menos $\lambda/2$, es decir, 1.764mm. Por este motivo, el paso en esta medida es de 1.5mm, suficiente para cumplir este criterio.

El plano espacial seleccionado recorre 400mm (de -200 a 200) en el eje X y 200mm (de -100 a 100) en el eje Y. De este modo se asegura que los lóbulos secundarios se aprecien perfectamente y que el nivel en los extremos baje hasta 40dB.

En esta medida comenzaron a surgir ciertos problemas en la representación de los datos adquiridos con el analizador de redes, ya que estos salían con un cierto entrelazado en las líneas pares e impares. Este error es comentado en el apartado 5.4 con mayor detalle.

Los resultados obtenidos (de magnitud y fase) quedan representados de las Figuras 5.6 a 5.11, donde se puede ver que dependiendo de la frecuencia el máximo del diagrama apunta en una dirección o en otra.

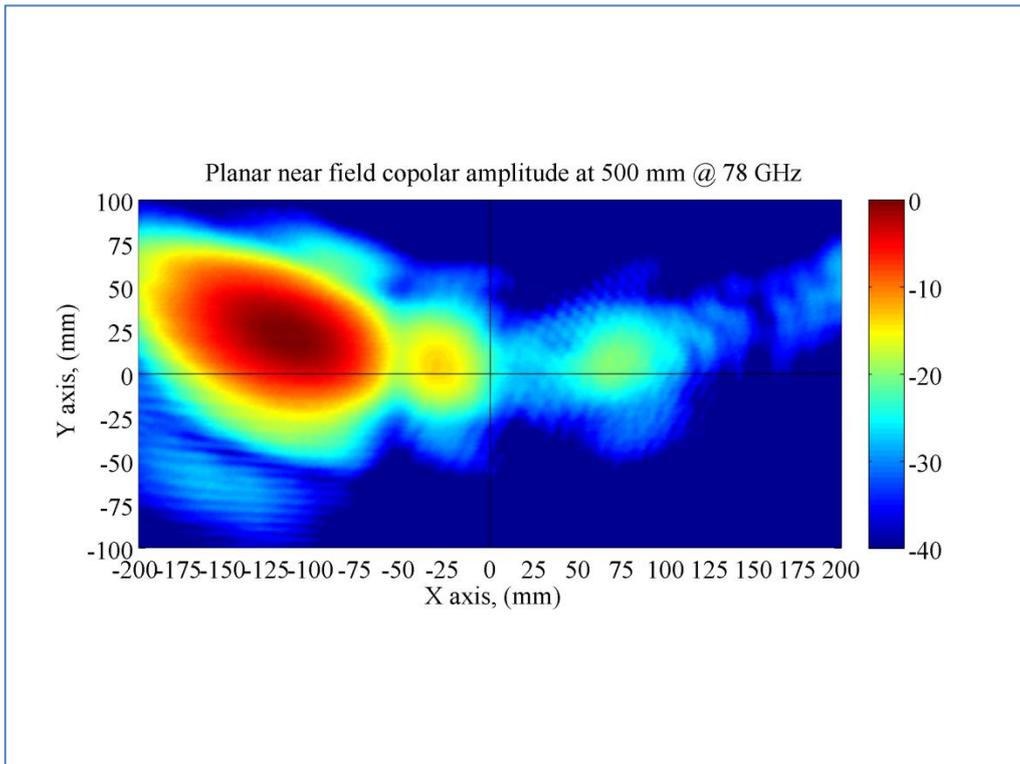


Figura 5.6: Magnitud en copolar para 78GHz

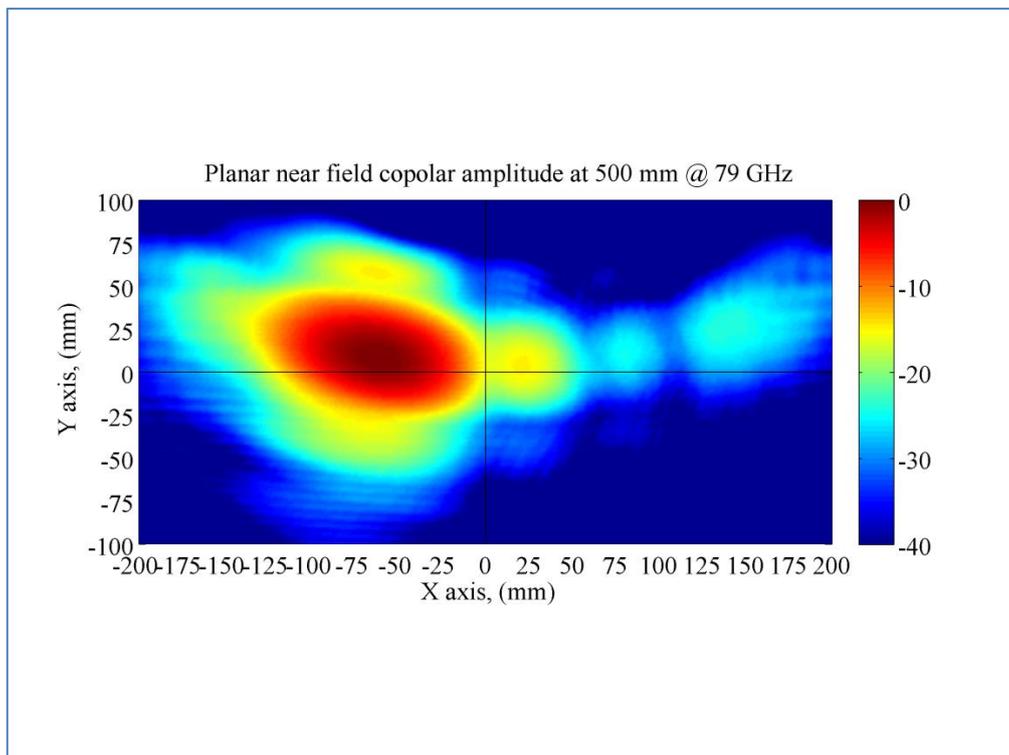


Figura 5.7: Magnitud en copolar para 79GHz

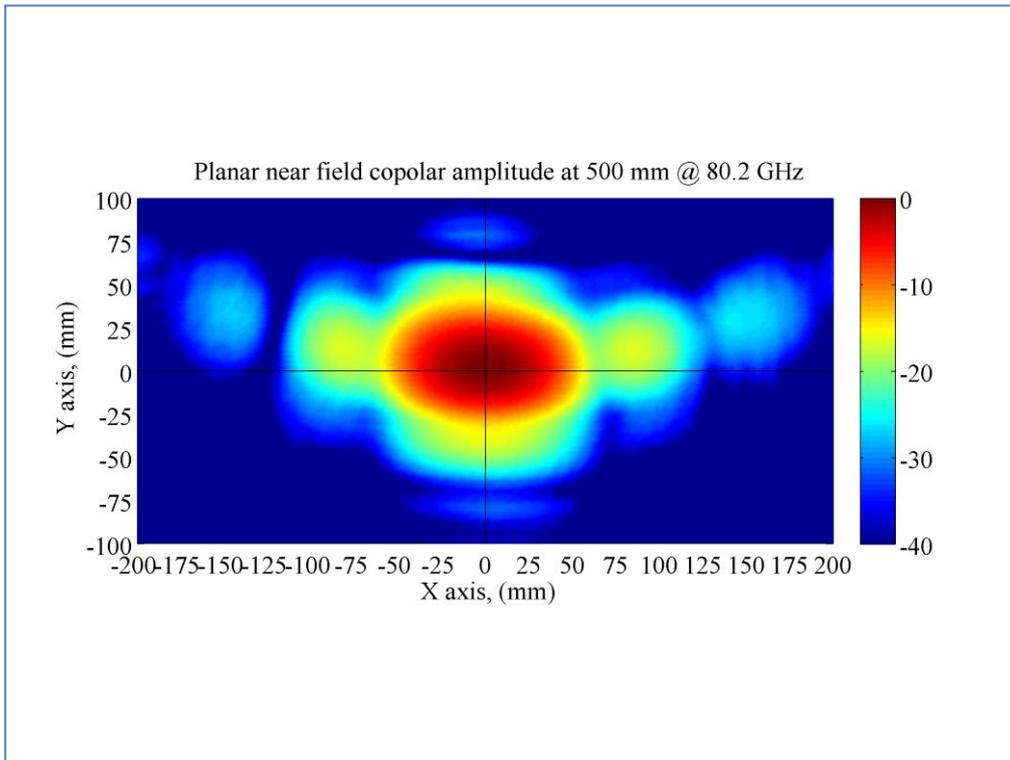


Figura 5.8: Magnitud en copolar para 80.2GHz

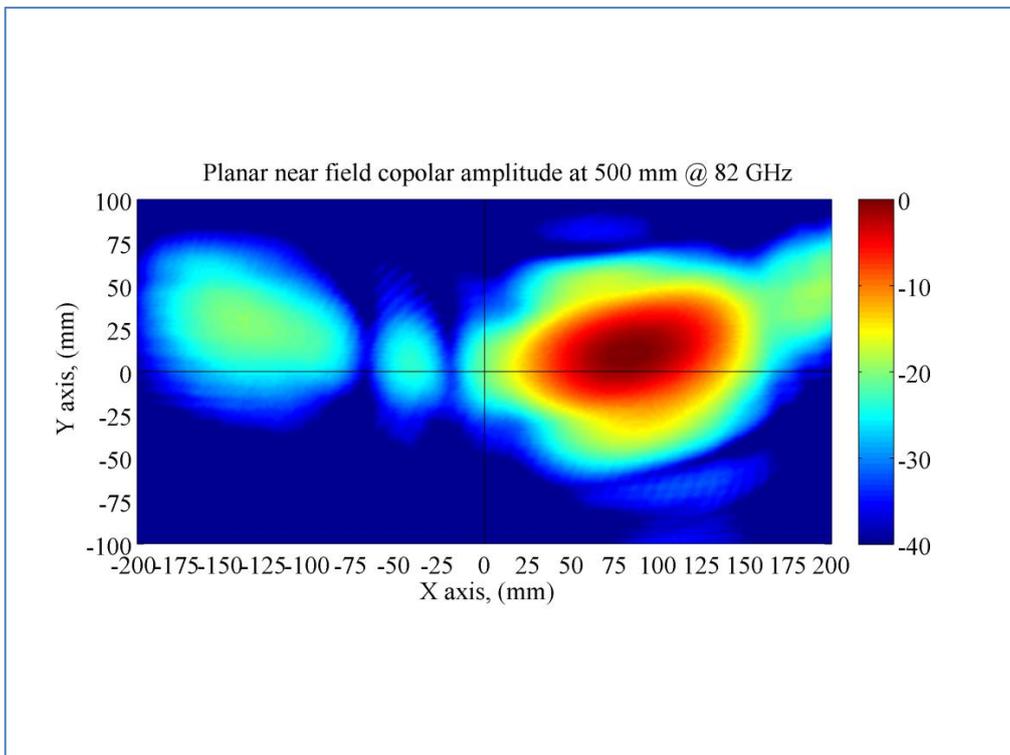


Figura 5.9: Magnitud en copolar para 82GHz

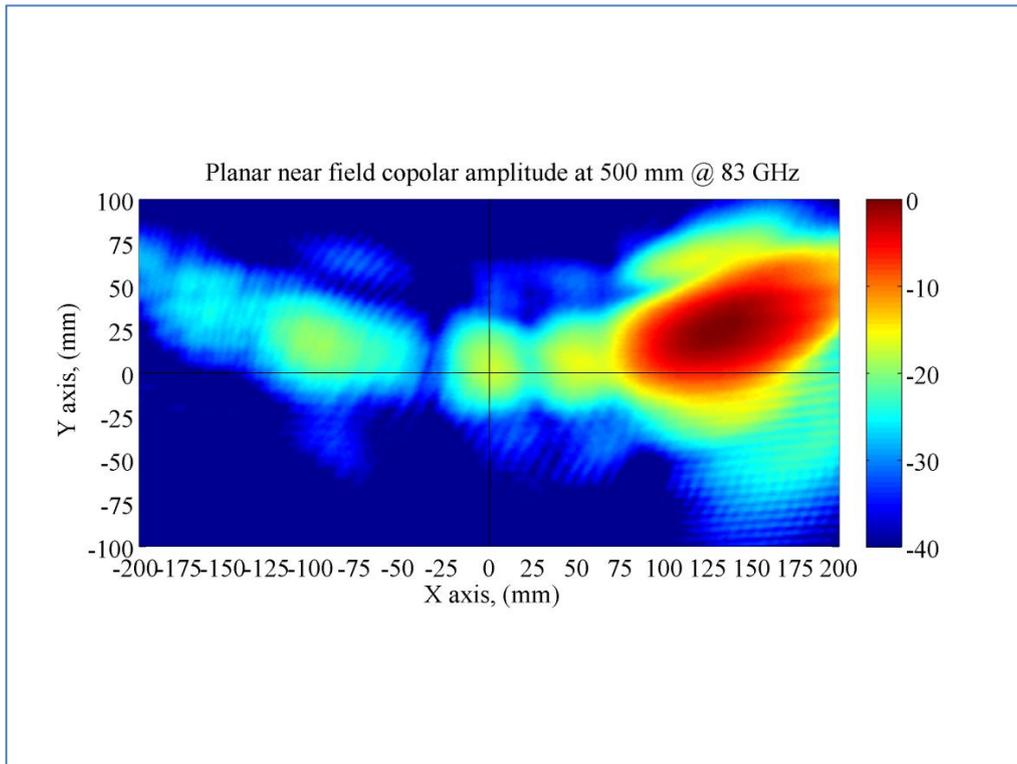


Figura 5.10: Magnitud en copolar para 83GHz

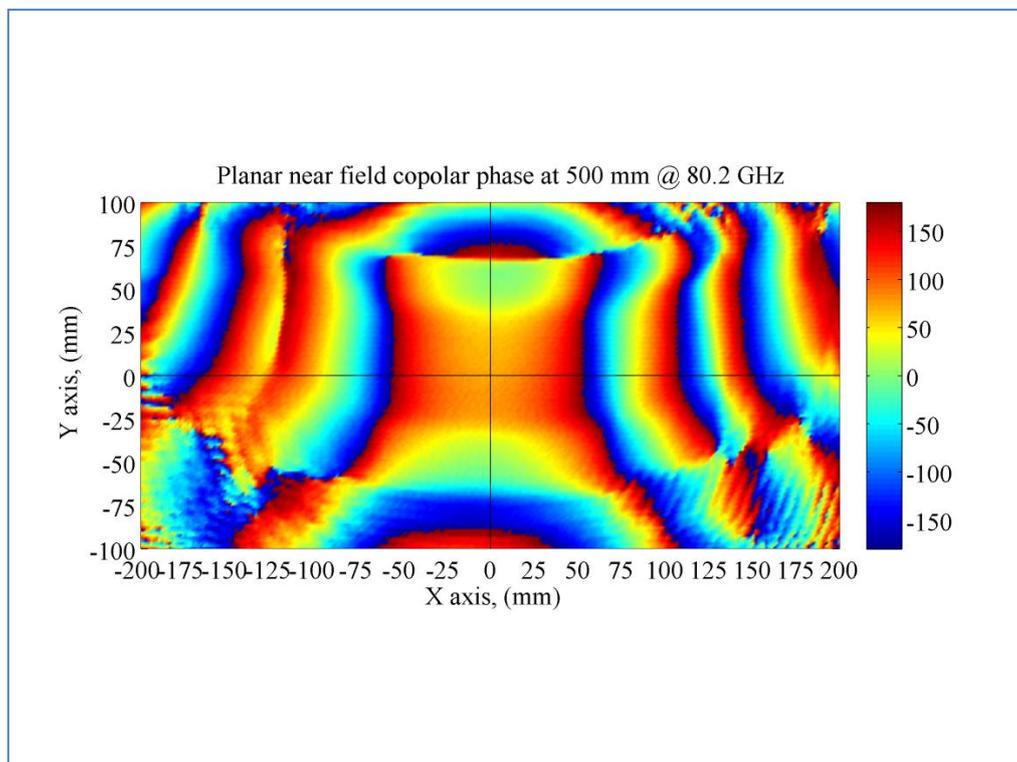


Figura 5.11: Fase en copolar para 80.2GHz

5.3.2. Bocina y lentes

El objetivo de esta medida es conseguir una mayor ganancia mediante el uso de una lente dieléctrica situada sobre la boca de una bocina piramidal en la banda frecuencial WR-10.

La situación de los cabezales de 75-110GHz y demás componentes del sistema se pueden ver en la Figura 5.12. Para medir el diagrama de radiación, como antena transmisora se utiliza la bocina con lente y como receptora una sonda de campo cercano (OEWG, open ended waveguide).

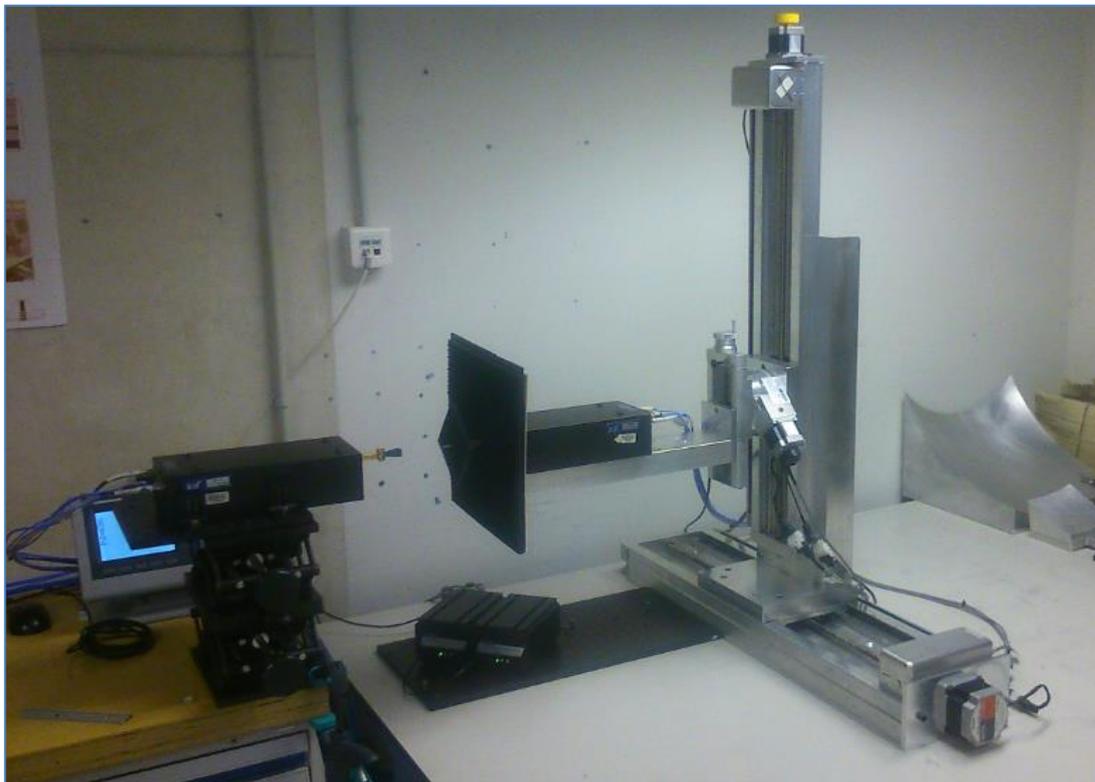


Figura 5.12: Sistema de posicionamiento para una medida de bocina con lentes

El primer paso ha consistido en realizar una medida de ganancia, para comparar si la lente (ver Figura 5.13) funcionaba de forma correcta. Para ello se utiliza el método de las tres antenas: una antena receptora, una bocina calibrada (a 93GHz - 20dB de ganancia) y la bocina con la lente que se quiere medir. Comparando las dos bocinas se observó que la bocina con lente perdía 2 dB de ganancia. Estas pérdidas vienen dadas por la propagación de radiofrecuencia a

través de la lente ya que el material con el que está construida tiene una $\tan\delta$ de 0.02 a 100 GHz aproximadamente y por el cambio de medio entre el aire y la lente ya que esta tiene una constante dieléctrica de 2.7 a 100 GHz aproximadamente.



Figura 5.13: Lente

También se ha realizado la medida del diagrama de radiación (medida del S21) para este sistema, con y sin lente, para ver la diferencia entre ambas. Los planos espaciales medidos son de 120mm en el eje X y 180mm en el eje Y. Las frecuencias utilizadas son las siguientes: $F_{min}=70\text{GHz}$, $F_{max}=115\text{GHz}$ y $F_{central}=93\text{GHz}$. En este caso el valor de $\lambda/2$ es igual a 1.6mm, por lo que para cumplir la condición se han utilizado pasos de 1mm. Las medidas realizadas han sido cuatro: sistema de bocina con lente copolar y crosspolar y bocina sin lente copolar y crosspolar.

En las siguientes figuras de la medida se puede apreciar un problema de entrelazado, puesto que no se ha encontrado una solución a este inconveniente en el momento de la medida. En las Figuras 5.14 y 5.15 se ve una comparación en términos de magnitud para una medida con lente y otra sin ella.

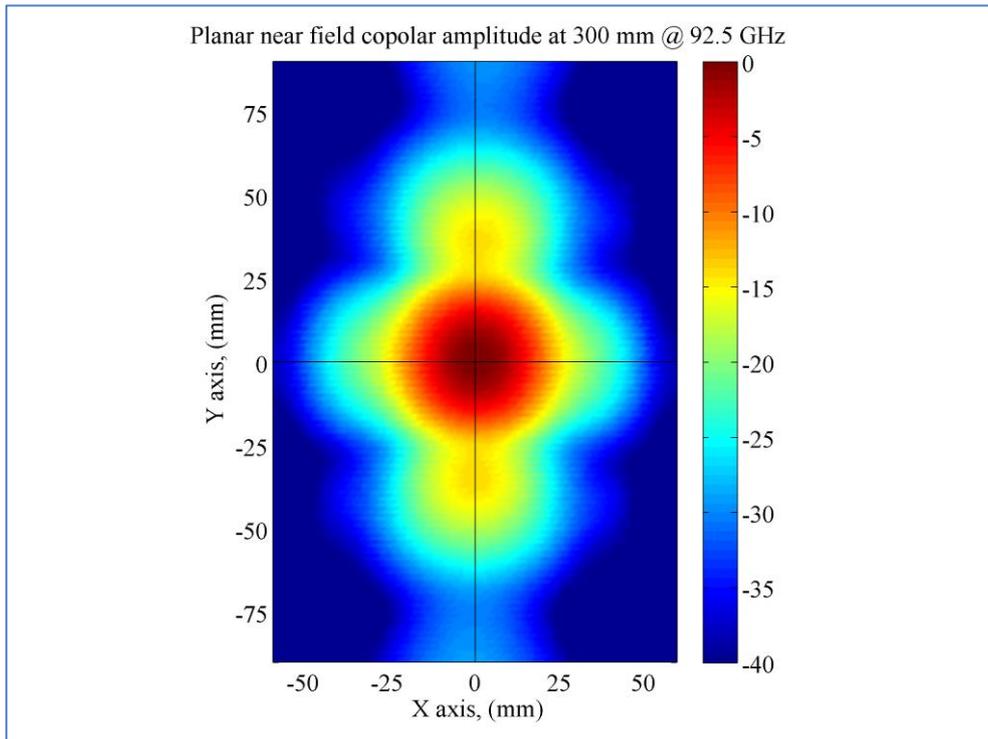


Figura 5.14: Magnitud copolar para una bocina sin lente a 92.5GHz

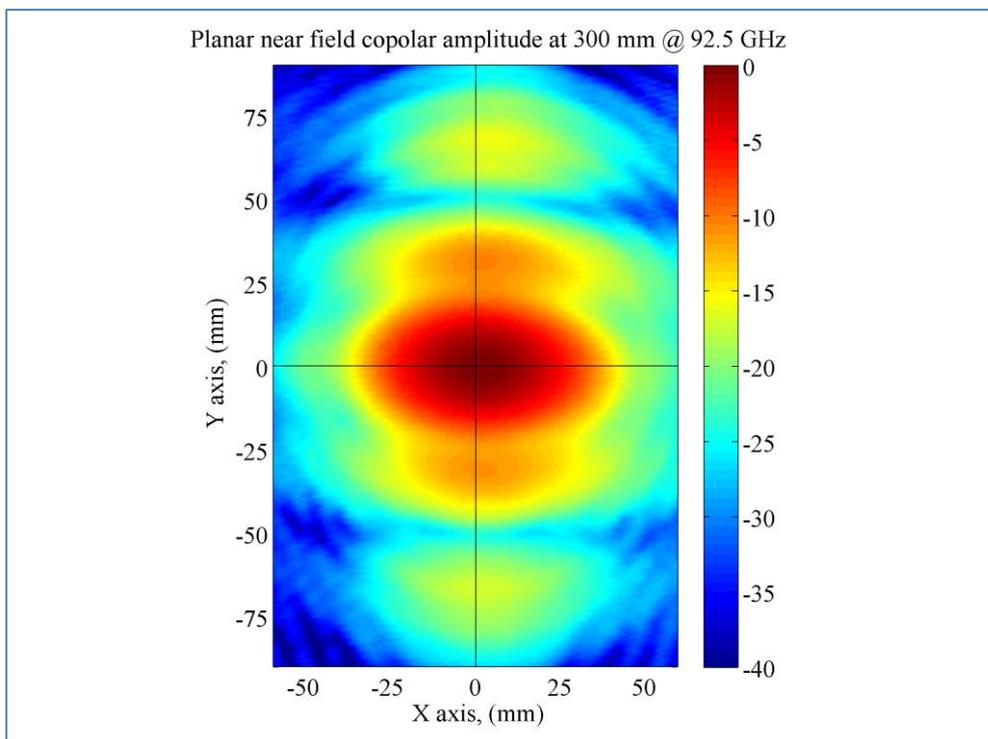


Figura 5.15: Magnitud copolar para una bocina con lente a 92.5GHz

Después de hacer la transformación a campo lejano e integrar el diagrama de radiación, los resultados, en valor de directividad, que dan las antenas de bocina con lente y sin lente respecto a los datos de calibración se ven la Figura 5.16.

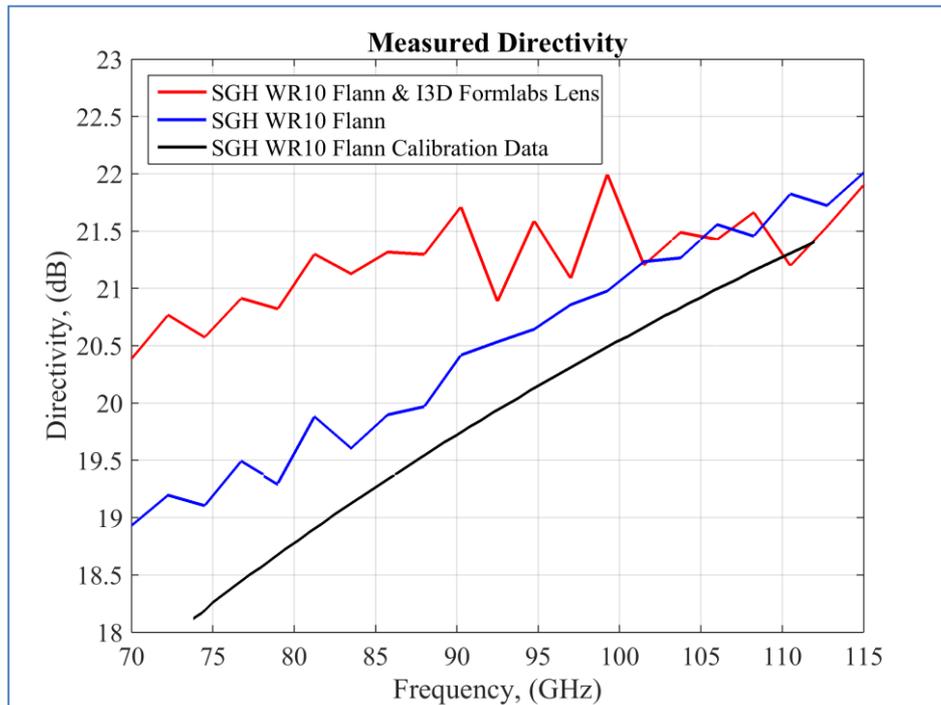


Figura 5.16: Comparación de la directividad en bocina con lente, sin lente y calibrada

El resultado de la Figura 5.16 indica que la bocina con lente funciona, ya que aumenta la directividad que era el objetivo, sin embargo la ganancia disminuye por las pérdidas asociadas a la propagación dentro de la lente. Para solucionar este problema se ha diseñado y construido una lente de Fresnel para que la distancia de propagación sea mucho menor.

5.3.3. Array de 8 antenas

En esta medida se trataba de obtener los diagramas de radiación correspondientes a un array de 8 antenas. La antena utilizada en este proceso se puede ver en la Figura 5.17.



Figura 5.17: Array de 8 antenas

Esta medida ha sido realizada con los cabezales (transmisores y receptores) de la banda WR-2.2, que comprende las frecuencias entre 330-500GHz. En estas frecuencias el problema de representación de diagramas con entrelazado se ha notado considerablemente debido a que este afecta a frecuencias más altas. Por este motivo, esta medida fue realizada con la versión anterior del programa donde el ZIG-ZAG no estaba presente. En la Figura 5.18 se puede ver como se montó la estructura para esta medida.



Figura 5.18: Sistema de medida Array de 8 antenas en banda WR 2.2

Se han medido los diagramas de radiación tanto para la antena 1 (comenzando desde la de arriba) como para la antena 4. El plano espacial de medida es de 49.7 mm en el eje horizontal y de 79.8 mm en el vertical, siendo la distancia entre antenas de 4.5cm. Exactamente la banda frecuencial es 320-400GHz ($F_{central}=360GHz$), y por ello los pasos seleccionados son de 0.35mm

cumpliendo con el requisito $\text{paso} < \lambda/2$. El número de puntos en cada barrido tiene un valor de 401 y dura 193.603ms. Los resultados obtenidos para la antena 1 se pueden ver en las Figuras 5.19 y 5.20.

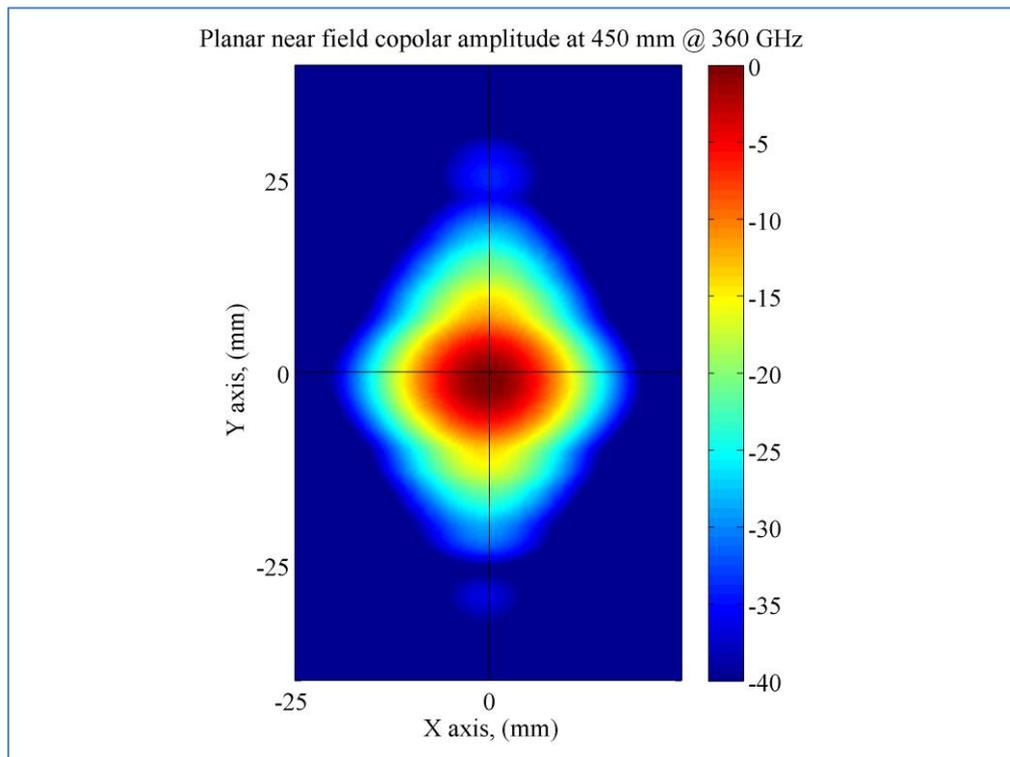


Figura 5.19: Magnitud de antena 1 en copolar para 360GHz

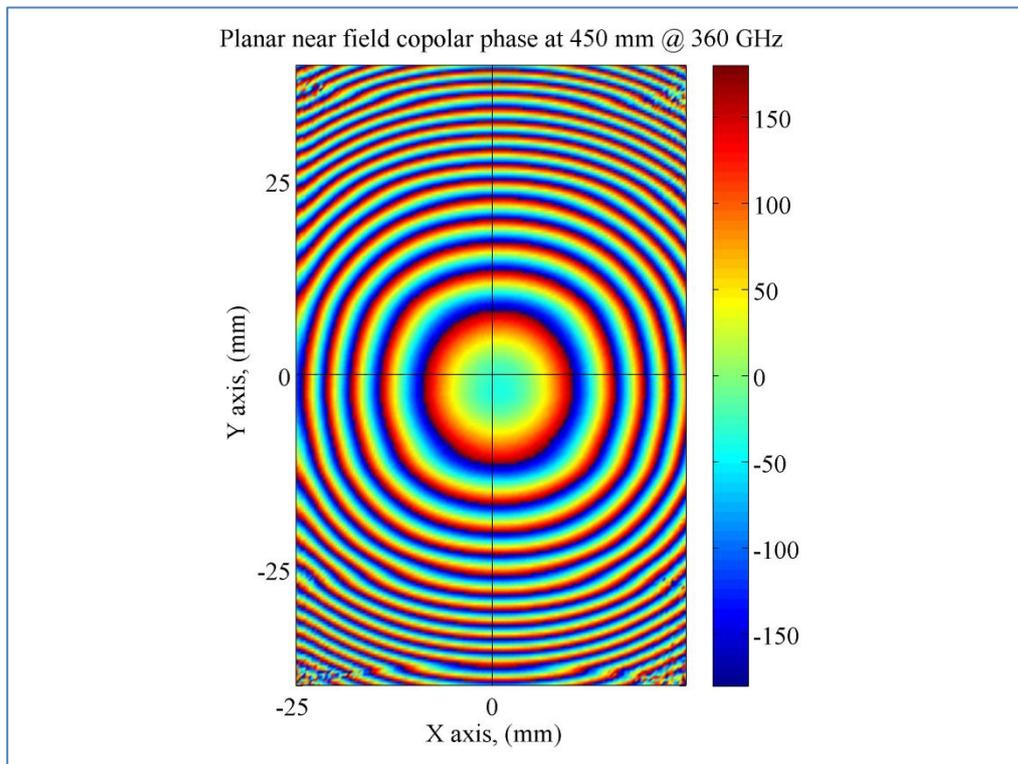


Figura 5.20: Fase de antena 1 en copolar para 360GHz

Para la antena 4 los resultados son muy similares y por eso no son presentados en la memoria del trabajo.

5.4. Problema de entrelazado en la representación de los diagramas de radiación

Este es el principal problema que se ha dado durante el transcurso de este trabajo. Una vez se lleva a cabo la recogida de datos del analizador de redes, se procede a la representación de los mismos mediante una herramienta software “MATLAB”. En algunas de las medidas se aprecia que los datos representados no eran del todo correctos, ya que el dibujo del diagrama de radiación presenta un entrelazado en las líneas horizontales, como se puede observar en las Figuras 5.21 y 5.22, que representan un diagrama de radiación de una medida de un array de 8 antenas a una frecuencia de 360 GHz.

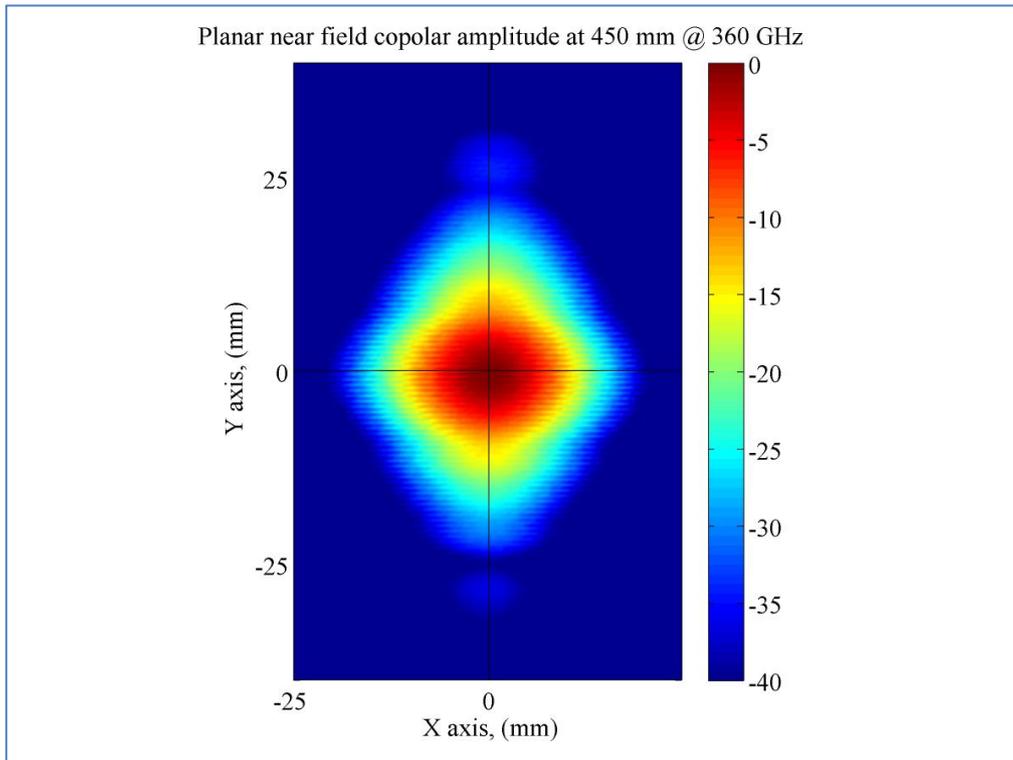


Figura 5.21: Diagrama de magnitud entrelazado a 360GHz

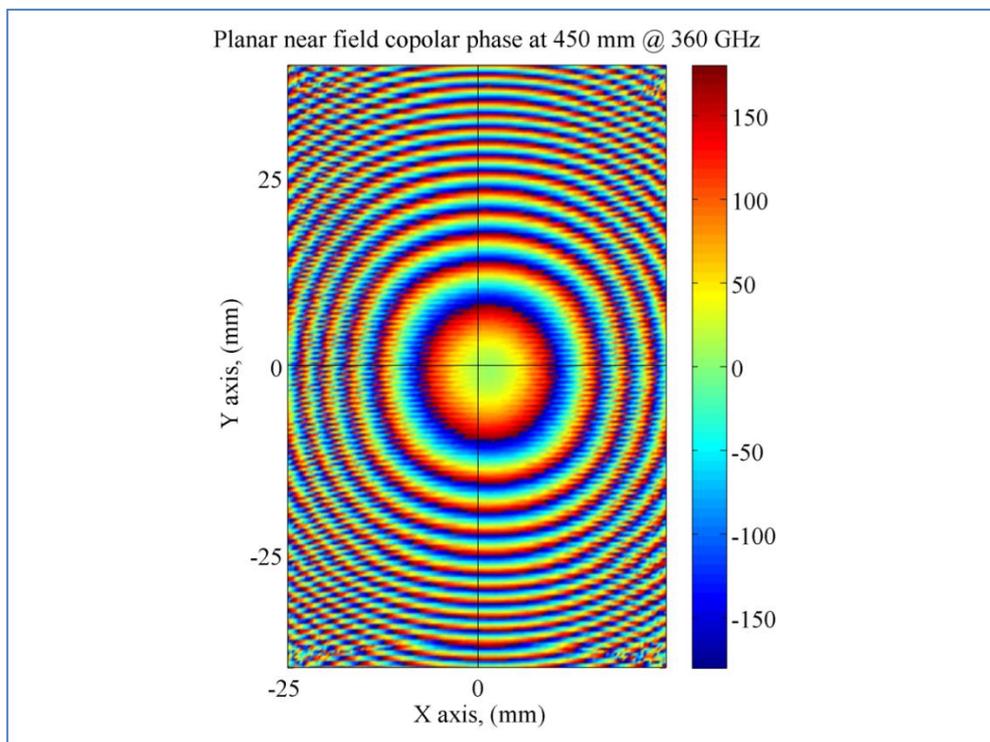


Figura 5.22: Diagrama de Fase entrelazado a 360 GHz



En la primera figura, donde está representada la magnitud, el lóbulo principal no está bien definido y se aprecia perfectamente que las líneas horizontales pares e impares no coinciden en su totalidad.

La mayor dificultad para encontrar una solución a este problema ha sido precisamente hallar el origen del mismo.

En un principio, cuando los diagramas representados mostraban el error, se pensaba que podía ser debido a la recogida de datos del analizador. Al haber automatizado el tiempo entre paso y paso (punto y punto del plano espacial seleccionado) el posicionador tenía que moverse demasiado rápido, en algunas ocasiones los pasos eran cercanos a 400 ms, y esto ocasionaba un retardo en los datos. Lo primero que se pensó fue que la estructura del posicionador no estaba del todo parada cuando el programa de LabVIEW se disponía a coger los datos del analizador de redes, es decir, que no había esperado un sweep time entero antes de coger los mismos. En una de las primeras medidas realizadas, concretamente en el array de escaneo de frecuencia para toma de imágenes, el entrelazado surgía de manera evidente y el resultado fue bastante negativo. Por ello, se pasó a dejar más tiempo entre punto y punto del plano, haciendo que el tiempo de espera antes de coger los valores de magnitud y fase fuese mayor y asegurándose así de que los motores estaban completamente parados y había transcurrido un tiempo de barrido. El resultado fue bastante mejor porque el entrelazado prácticamente había desaparecido casi por completo, aunque en algunas zonas del diagrama se podía apreciar, pero en menor medida.

Aunque en este caso se había podido solventar el problema, no era la solución adecuada de forma generalizada. En los siguientes diagramas realizados, para otro tipo de medidas diferentes, este suceso seguía presente y utilizando el método anterior no se conseguía resolver el problema. Incluso se probó a cambiar el método del trigger en el analizador, poniéndolo en "single", de esta forma se podía asegurar que no se estaba realizando un sweep time continuamente, sino únicamente cuando el motor estaba parado.

Se investigaron otras alternativas, como la comprobación de que los puntos en el eje X coincidían para todas las líneas que se recorrían en el plano, es decir, que las posiciones por las que el posicionador pasa en una línea en el eje X fuesen iguales a las de la siguiente línea. En algunos casos, las posiciones reales no eran las que se estimaban.



Aquí es donde se ha encontrado el origen del problema, los puntos por los que pasa el posicionador durante una línea en el eje X no son siempre los mismos y eso hace que los datos que se cogen del analizador no sean exactamente los que debería. Esto no es debido a un error en la programación del software, sino a una confusión mecánica de los motores del posicionador y a la introducción de la mejora relacionada con el ZIG-ZAG en el barrido de puntos.

Cuando la estructura recorre la primera línea normalmente no hay error, pero al bajar una posición en el eje vertical (Y) y cambiar de dirección para recorrer la siguiente (ZIG-ZAG), por la propia inercia que llevaban los motores el primer paso (de cada una de las líneas) que hace es menor al que realmente debería hacer. Por ejemplo, en una de las medidas en la banda WR-2.2, el paso a hacer después de bajar en el eje Y sería 0.35mm, pero realmente el motor está moviendo 0.25mm.

Dependiendo de las medidas, este error es bastante considerable, aunque parezca que es mínimo. Se puede decir entonces que el error, debido a la propia mecánica de la estructura, será mayor o menor según la banda frecuencial en la que se esté trabajando. Los mm que se desajusta están entre 0.09 y 0.15mm. Al descubrir cuál era el problema principal cuadran los resultados obtenidos en el primer plano espacial medido. Como los datos recogidos eran en banda WR-10 (75-110GHz), el paso entre punto y punto tiene un valor de 1.5mm, y comparando 1.5mm con un error de 0.1mm se puede decir que afecta en menor medida. En la segunda medida, bocinas y lentes, a pesar de ser en la misma banda frecuencial, se utiliza un paso de 1mm y por ello el entrelazado se aprecia de manera significativa.

Pero el gran inconveniente ocurre en medidas en bandas de muy alta frecuencia, entre 330-500GHz, donde los pasos (menores que $\lambda/2$) son de 0.35mm aproximadamente. Al comparar 0.1mm con 0.35mm la diferencia es muy pequeña y el error llega a ser enorme, y prácticamente incompensable.

Se intentó solucionar estos problemas mediante el envío de ciertos comandos a los motores, que permitiesen resetear de alguna manera los mismos cada vez que una línea era totalmente completada. De esta forma, mandando comandos a los motores como C "Clear all



commands from currently selected program” o *K “Kill operation/program in progress and reset user outputs”* se comprobó que el problema no era de software sino de mecánica.

Por este motivo, en las versiones disponibles hay una que no realiza ZIG-ZAG (donde los puntos coinciden siempre), única forma de solucionar el error que introduce la inercia de los motores y conseguir que los resultados obtenidos en los diagramas de radicación sean totalmente válidos. No obstante, como guardamos la posición de todos los puntos de medida y estos son recogidos en el punto exacto donde para el motor, consideramos que siempre se podría interpolar para conseguir los puntos sobre una matriz regular y solventar el problema del entrelazado sin tener que recurrir a no usar el Zigzag que ahorra tiempo de medida, pero no ha sido posible implementar esta solución por falta de tiempo antes de finalizar el Trabajo Final de Grado.



CAPÍTULO 6. CONCLUSIONES Y LÍNEAS FUTURAS

6.1. Conclusiones

Durante el transcurso de este proyecto se han afrontado dos tipos de trabajos, como son el diseño y optimización del software de procesado del sistema de posicionamiento para realizar medidas en milimétricas y THz, así como por otra parte la toma de medidas con el software optimizado de distintos tipos de antenas.

Se han realizado 3 versiones de este software para ser utilizadas desde un PC externo y una muy similar que pueda ser lanzada desde el analizador de redes. Las que se encuentran en el ordenador se diferencian en las funciones que realizan, una de ellas contiene todas las mejoras realizadas, otra no coge la posición del lector de encoders y la última no realiza un ZIG-ZAG en el barrido de puntos del plano espacial que se mida. Sin embargo, la versión que se puede utilizar desde el analizador no dispone únicamente de la función de recogida de posición y es debido a que este aparato no permitía realizar dicha acción.

Cabe destacar la complejidad de todo tipo de medidas que se realizan con estos sistemas de posicionamiento, sobre todo, en muy alta frecuencia. Llevar a cabo una medida totalmente válida en banda WR 2.2 resulta muy complicado debido al pequeño valor de λ . Esto junto al problema de entrelazado han hecho que ciertos resultados no hayan sido del todo satisfactorios durante el proyecto.

Por este motivo, conviene saber cuándo es necesario utilizar cada una de las diferentes versiones de software disponibles. Si las medidas no llegan a los 80 GHz como frecuencia máxima ($\lambda=3.75\text{mm}$, $\lambda/2=1.875\text{mm}$), se puede utilizar la versión que incorpora todas las mejoras, pero si se superan estas frecuencias es recomendable mandar las medidas con la versión que no realiza un ZIG-ZAG en el barrido de puntos.



Debido a este problema, surgido a partir de una mejora que se introdujo en el software (“corrección en el barrido de puntos”), se ha perdido mucho tiempo en su posible solución y no ha dado tiempo a incluir otras mejoras en el programa, que posteriormente se explican.

Otro aspecto a resaltar es la mejora que se ha podido hacer respecto al tiempo total de medida. Este sistema de posicionamiento es muy utilizado, y las mejoras como el tiempo automatizado y la corrección en el barrido han hecho que sea posible una mejora de casi un 20% del valor de tiempo anterior. Por supuesto, si no es utilizado con la versión de ZIG-ZAG este ahorro se disminuye, pero se sigue notando. Se puede decir, que este tiempo ha sido totalmente ajustado para abarcar tanto el movimiento del posicionador en cada paso en suma al sweep time que este seleccionado. No existiría ninguna otra forma de hacer que este sistema fuese más rápido, excepto si se selecciona mayor velocidad (que ya está incluido) y mayor aceleración, aunque estos tienen un inconveniente, hacen vibrar demasiado la estructura metálica y en ciertas medidas no es adecuado utilizarlos.

Uno de los primeros objetivos, como se ha comentado en apartados anteriores, era coger la posición exacta donde se encontraba el posicionador mediante un lector de encoders. El objetivo se ha conseguido llevar a cabo, aunque la acción tarda 275ms en completarse y las medidas duraran más tiempo (no mucho más). Se podría investigar en un futuro reducir este tiempo lo máximo posible, aunque reduciéndolo más provoca la aparición de problemas de sincronización.

6.2. Líneas futuras

Durante el desarrollo del proyecto se ha visto que se podían incluir todavía más mejoras en el programa, aunque por falta de tiempo no se han podido implementar y se quedan como líneas futuras a tener en cuenta.

Por ejemplo, uno de los aspectos a evaluar es el tamaño de los archivos donde se guardan los datos. En ciertas ocasiones, si las medidas son muy largas, estos archivos son enormes. En este proyecto se pensó en guardar los documentos en formato binario para ahorrar una gran cantidad de espacio, pero en todas las pruebas que se realizaron surgían problemas a la hora de



abrir esos datos con otro tipo de herramientas software. A parte de este motivo, el ahorro que se había conseguido era solo de un 12%, no todo lo esperado. En este ámbito se puede trabajar para conseguir que estos archivos no llenen la memoria del PC o analizador, y que su procesamiento sea mucho más rápido.

Otra alternativa es que el propio software, una vez acabada la ejecución del programa, te dé a elegir con que puntos de barrido (frecuencias) te quieres quedar y así eliminar el resto de datos del documento haciendo que este ocupe menos.

También se pueden realizar nuevos diagramas o gráficas que vayan dibujando el valor de magnitud y fase en cada uno de los puntos de medida para la frecuencia central, así se puede ir viendo el diagrama en 2D en tiempo real. De esta forma, se facilita el ver cómo está saliendo la medida.

Por último, en el software se puede incluir una opción que permita hacer un barrido “on fly”. Esto significa que no habrá un paso entre punto y punto de medida, sino que las líneas horizontales se recorrerán sin parar de principio a fin. Este es un método de barrido muy rápido que generaría resultados no del todo exactos pero sí que servirían como indicación.

Anexos



Anexo 1. “Comandos Lector HEIDENHAIN QUADRA-CHEK”

KEYCODES for Metronics QC100

(each command below is case sensitive and MUST be terminated with a carriage return)

COMMAND QC100 Front Panel Key Name

KEY1 Left Wide Key
KEY2 Right Wide Key

KEY3 First Axis
KEY4 Second Axis
KEY5 Third Axis
KEY6 Fourth Axis

KEY7 Left Arrow
KEY8 Up Arrow
KEY9 Right Arrow
KEY10 Down Arrow

KEY11 Soft Key 1
KEY12 Soft Key 2
KEY13 Soft Key 3
KEY14 Soft Key 4
KEY15 Soft Key 5

KEY16 Print

KEY21 0
KEY22 1
KEY23 2
KEY24 3
KEY25 4
KEY26 5
KEY27 6
KEY28 7
KEY29 8
KEY30 9

KEY31 . (decimal point)
KEY32 +/-

KEY33 Enter
KEY34 Finish
KEY35 Cancel
KEY36 Quit
KEY44 Power

Anexo 2. "Comandos motores Velmex VXM"

VXM Program Stored Commands

Motor commands:

ImMx Set steps to incremental Index motor CW (positive), m= motor# (1,2,3,4), x=1 to 16,777,215
ImM-x Set steps to incremental Index motor CCW (negative), m= motor# (1,2,3,4), x=1 to 16,777,215
IAmMx Set Absolute Index distance, m= motor# (1,2,3,4), x= ±1 to ±16,777,215 steps
IAMM0 Index motor to Absolute zero position, m= motor# (1,2,3,4)
IaMm-0 Zero motor position for motor# m, m= 1,2,3,4
ImM0 Index motor until positive limit is encountered, m= motor# (1,2,3,4)
ImM-0 Index motor until negative limit is encountered, m= motor# (1,2,3,4)
(i3,j1...) Combine Index commands to run simultaneously on two VXM controllers connected by VXM bus
SmMx Set Speed of motor (70% power), m= motor# (1,2,3,4), x=1 to 6000 steps/sec.
SAMMx Set Speed of motor (100% power), m= motor# (1,2,3,4), x=1 to 6000 steps/sec.
SmM-x Read and assign analog input value to motor m speed (70% power), x= range
SAMM-x Read and assign analog input value to motor m speed (100% power), x= range
AmMx Acceleration/deceleration, m= motor# (1,2,3,4), x=1 to 127.

Looping/branching commands:

L0 Loop continually from the beginning or Loop-to-marker of the current program
LM0 Sets the Loop-to-marker at the current location in the program
LM-0 Resets the Loop-to-marker to the beginning of the current program
Lx Loop from beginning or Loop-to-marker x-1 times (x=2 to 65,535), when the loop reaches its last count the non-loop command directly preceding will be ignored
L-x Loop from beginning or Loop-to-marker x-1 times, alternating direction of motor 1, when the loop reaches its last count the non-loop command directly preceding will be ignored
LAX Loop Always from beginning or Loop-to-marker x-1 times (x=2 to 65,535)
LA-x Loop Always from beginning or Loop-to-marker x-1 times, alternating direction of motor 1
LM-2 Loop once from beginning or Loop-to-marker reversing index direction of motor 2
LM-3 Loop once from beginning or Loop-to-marker reversing index direction of motor 1 and motor 2
Jx Jump to the beginning of program number x, x= 0 to 4
JMx Jump to the beginning of program number x and come back for More after program x ends, x= 0 to 4
JM-x Similar to JMx except automatically moves back from absolute indexes after program x ends: For pick-and-place within matrix looping patterns

Pausing commands:

Px Pause x tenths of a second, (x=0 to 65,535)
P-x Pause x tenths of a millisecond, (x=1 to 65,535)
PAx Pause x tenths of a second (x=0 to 65,535, 10 µsec pause when x=0) Altering output 1 high for duration of the pause, tenths of a millisecond when x is negative
PA-x Pause x tenths of a millisecond (x=1 to 65,535) Altering output 1 high for duration of the pause

Input/output commands:

U0 Wait for a "low" on user input 1
U1 Wait for a low on user input 1, holding user output 1 high while waiting
U2 Enable Jog mode while waiting for an input
U3 Disable Jog mode while waiting for an input
U4 User output 1 "low" (reset state)
U5 User output 1 high
U6 Send "W" to host and wait for a "G" to continue
U7 Start of Continuous Index with pulse on output 2
U77 Start of Continuous Index with no output
U8 Start of Continuous Index sending "@" to the host
U9 End of Continuous Index with auto-decel to stop
U91 End of Continuous Index with auto-generate a deceleration Index as next command
U92 End of Continuous Index using next Index for deceleration to stop
U99 End of Continuous Index with instantaneous stop
U11 Skip next command if input 1 is high
U21 Skip next command if input 1 is low
U12 Skip next command if input 2 is high
U22 Skip next command if input 2 is low
U13 Wait for a front panel button to jump to a program or continue: "Motor 1 Jog -" button to jump to program #1, "Motor 1 Jog +" button to jump to program #2, "Run" button to proceed in current program.
U14 User output 2 low (reset state)
U15 User output 2 high
U16 Optional User output 3 low (reset state)
U17 Optional User output 3 high
U18 Optional User output 4 low (reset state)
U19 Optional User output 4 high
U23 Wait for a front panel button to jump to a program and come back, or continue: "Motor 1 Jog -" button to jump and return to program #1, "Motor 1 Jog +" button to jump and return to program #2, "Run" button to proceed in current program
U30 Wait for a low to high transition on user input 1
U31 Wait for a low to high transition on user input 1, holding user output 1 high while waiting
U32 Wait for "Motor 1 Jog -" button to be pressed on front panel with debouncing
U33 Wait for "Motor 1 Jog +" button to be pressed on front panel with debouncing
U50 Wait for a low and high on user input 1 with debouncing for a mechanical push-button switch
U51 Wait for a low and high on user input 1 with debouncing for a mechanical push-button switch, holding user output 1 high while waiting
U90 Wait for a low to high on the Run button or connection I/O,4 with debouncing for a mechanical push-button switch

VXM Set Commands

setMmMx Set axis m for motor type/size x (see table below for value for x)
setDMx Set VXM/VP9000 or NF90 emulation modes, and other operating parameters
setDax Set Analog Joystick Deadband value
setjmm Set first Jog Speed setting for motor m
setjAmM Set first Analog Joystick range setting for motor m
setJmM Set second Jog Speed setting for motor m
setJAmM Set second Analog Joystick range setting for motor m
setLmMx Set limit switch mode for axis m
setPmMx Set "Pulse Every x # Steps" on output 2 for axis m
setPAx Set Pulse width used by setPmMx and U7, x=1 to 255 (10 µsec increments)
setIx Set operating mode of inputs
setBx Set RS-232 Baud rate (9=9600, 19=19200, 38=38400)
Bx Backlash compensation, x=0 to 255, 0= off (default), 1= 20 steps
Ox Indicate limit switch Over-travel to host, off when x=0, VXM sends "O" when x=1 and hit limit, x=3 program stops too
PMAx Program Associate program x in Master to program x in Slave (Linked VXMs start the same time) (255= default/disabled)

VXM Immediate Commands

Operation commands:

C	Clear all commands from currently selected program
D	Decelerate to a stop (interrupts current index/ program in progress)
E	Enable On-Line mode with echo "on"
F	Enable On-Line mode with echo "off"
G	Enable On-Line mode with echo off Grouping a <cr> with "^", ":", "W", "O" responses; Also Go after waiting or holding
H	Put Controller on Hold (stop after each command and wait for go)
K	Kill operation/program in progress and reset user outputs
N	Null (zero) motors 1,2,3,4 absolute position registers
Q	Quit On-Line mode (return to Local mode)
R	Run currently selected program
!	Record motor positions for later recall with "x","y" commands
rsm	Run save memory (saves setup/ program values to nonvolatile memory)
res	Software reset controller
del	Delete last command
[i1,i2...]	Send data to Slave through Master (two VXM controllers connected by VXM bus)
setD0	Set VXM back to factory defaults (All programs, settings, motor selections will be erased)
PMx	Select Program number x, x= 0 to 4
PM-x	Select and clear all commands from Program number x, x= 0 to 4

Status request commands:

V	Verify Controller's status, VXM sends "B" to host if busy, "R" if ready, "J" if in the Jog/slew mode, or "b" if Jog/slewing
X	Send current position of motor 1 to host (Motor can be in motion)
Y	Send current position of motor 2 to host (Motor can be in motion)
Z	Send current position of motor 3 to host (Motor must be stationary)
T	Send current position of motor 4 to host (Motor must be stationary)
M	Request Memory available for currently selected program
!st	List current program to host (ASCII text)
x	Send last 4 positions of motor 1 to host that were captured by the "!" command or Input 4 trigger
y	Send last 4 positions of motor 2 to host that were captured by the "!" command or Input 4 trigger
#	Request the number of the currently selected motor
*	Request the position when the last motor started decelerating (shows position when "D" command or Stop/User input 4 used)
?	Read state of limit switch inputs for motor 1 and 2 (8 bit binary value)
~	Read state of User Inputs, Motor 1 and 2 Jog Inputs (8 bit binary value)
\$	Read state of User Outputs (8 bit binary value)
@	Read user analog input value
B	Read Backlash compensation setting
O	Read Indicate limit switch setting
D	Read/Digitize motor position (Jog Mode)
PM	Request the number of the current Program
PMA	Request the current program associate number (255= default/disabled)
getMmM	Read motor type/size selected for axis m
getDM	Read operating mode of VXM
getD0	Gets the VXM's firmware version in the format X.XX
getD1	Gets the VXM's firmware date code in the format XX-XX-XX (month,day,year)
getD2	Returns 2 if system is a single VXM, returns 4 if VXM is a Master
getDA	Read Analog Joystick Deadband setting
getjmM	Read first Jog Speed setting for motor m
getJAmM	Read first Analog Joystick range setting for motor m
getJmM	Read second Jog Speed setting for motor m
getJAmM	Read second Analog Joystick range setting for motor m
getLmM	Read mode of limits for motor m
getPmM	Read "Pulse Every x # Steps" value for axis m
getPA	Read Pulse width used by setPmMx and U7
getI	Read operating mode of user inputs