

Entornos avanzados de medida

Cedric Prieels

7 de octubre 2016

Resumen

Esta práctica tenía por objetivos principales la adquisición y digitalización de una señal de cualquier tipo, y la construcción de una señal, usando el software Labview de National Instruments. Se han creado diferentes instrumentos virtuales que nos permiten generar señales, y que nos permiten controlar diferentes aparatos distintos (oscilador, osciloscopio), directamente desde el ordenador.

1. Introducción teórica

Cuando se adquiere una señal, se representan en general dos tipos de gráficas distintas : la frecuencia en función del tiempo de la medida, o bien la amplitud de la señal en función de la frecuencia ω . Para representar cualquiera señal, tenemos que distinguir dos parámetros relacionados con el tiempo importantes : el *tiempo de muestreo* t_m , que corresponde al intervalo de tiempo entre 2 medidas y que toma entonces valores discretas (en particular, usaremos también su inversa, la *frecuencia de muestreo* ν_m) y el *tiempo total de muestreo* t_t , que se obtiene fácilmente a partir del tiempo de muestro y del número de puntos medidos, mediante la ecuación (1).

$$t_t = N_p \cdot t_m \quad (1)$$

El parámetro t_m es importante porque es directamente relacionado con la frecuencia máxima de una señal que podemos medir, de la manera siguiente. Si esta relación (2) no se cumple, las frecuencias de la señal que van por encima del límite se plegan y un fenómeno de *aliasing* aparece. Es entonces importante tener este efecto en cuenta cuando se mide un espectro de una señal, usando un tiempo de muestro correcto o usando filtros *anti-aliasing*, que permiten quitar todas las frecuencias que van por encima del límite.

$$\nu_{max} = \frac{1}{2 \cdot t_m} = \frac{\nu_m}{2} \quad (2)$$

Para realizar esta práctica, se usa el software Labview, desarrollado por National Instruments, que nos ofrece un entorno de programación con un lenguaje visual gráfico, y que nos permite controlar a distancia y adquirir datos que provienen de diferentes tipos de instrumentos de medida. Los programas desarrollados por este software se llaman instrumentos virtuales, y cada instrumento tiene dos paneles distintos que juegan cada uno un papel importante en diferentes casos :

- Panel frontal : en este panel se introducen todos los controles y indicadores que necesitamos. Este lugar también nos sirve cuando queremos observar el resultado final devuelto por nuestro instrumento (en este caso, en forma de plots y gráficas).
- Diagrama de bloques : ahí se diseña lo que tiene el instrumento virtual, todas sus conexiones y implementando todo lo que tiene que hacer, con el uso de bucles y operadores matemáticos.

2. Desarrollo experimental

2.1. Primer día

El primer día de la práctica tenía por objetivo principal la creación con Labview de un instrumento virtual que nos permite generar una señal que equivale a la suma de dos cosenos del tipo (3). En este informe, llamaremos este aparato **Sum2Cos**.

$$V(t) = A_1 \cos(\omega_1 t + \phi_1) + A_2 \cos(\omega_2 t + \phi_2) \quad (3)$$

Esta señal es periódica solamente si el cociente de las dos frecuencias es racional, y tiene por valor de frecuencia total el máximo común divisor de las frecuencias individuales. Lo primero que hicimos en el laboratorio era entonces definir todos los valores de los parámetros que entran en (3) en el panel frontal de Labview (las dos amplitudes A_i , las dos frecuencias ω_i , los dos desfases ϕ_i , el tiempo de muestreo Δt y el número de puntos medidos N_p). El instrumento virtual que queremos crear tiene entonces 8 entradas, que corresponden a estos 8 parámetros, y una sola salida, un array de números de tipo float con los valores de la tensión de la señal generada.

Después, se crea un programa principal en Labview, llamado **PlotSum2Cos**, en el cual vamos a usar el instrumento virtual que acabamos de crear, y que nos proporciona una señal que corresponde a la suma de dos cosenos. En este programa principal, se añaden tres gráficas distintas : la primera representa la evolución temporal de la señal, mientras que la segunda representa la fase en función de la frecuencia y que la tercera representa la amplitud del espectro con respecto a la frecuencia de la señal en Hz. Por supuesto, para poder representar estos espectros, hay que usar un dispositivo ya implementado por Labview que nos permite calcular la transformada de Fourier de nuestra señal, y se pintan todos estos plots con una escala logarítmica. Intentaremos con este dispositivo observar el fenómeno de *aliasing* cuando subimos el valor de Δt hasta llegar al límite introducido en la parte teórica.

2.2. Segundo día

El segundo día, estudiamos primero como usar Labview para controlar diferentes tipos de aparatos. Se usa en la práctica un protocolo de control muy estándar de tipo GPIB (un controlador permite controlar hasta 16 aparatos conectados a un bus, para que solo uno hable a la vez). Con estos tipos de sistemas, cada aparato tiene un número de identificación (una dirección) único. En este caso, tendremos solamente dos aparatos conectados : un oscilador (dirección 10) y un osciloscopio (dirección 6). En particular, usamos en la práctica un sistema un poco más general y sencillo que el GPIB : el VISA, que permite detectar solo que tipo de configuración tenemos.

Queremos construir un programa llamado **ControlWaveGenerator** que sea capaz de enviar al generador de onda señales de 5 tipos diferentes : cuadrada, sinusoidal, triangular, del usuario o de la memoria volátil (memoria del dispositivo que se borra cada vez que se reinicializa). El programa se construye siguiendo una serie de pasos que serán explicados en el apartado con los resultados, y tiene que ser capaz de controlar el generador de forma de onda.

El otro objetivo del segundo día era la construcción de un instrumento virtual **ConvertToString** que convierte un array de números en coma flotante a un string de enteros normalizados entre -2047 y 2047, que se puede dar como señal al oscilador (porque no es capaz de leer números flotantes directamente, solo podemos darle una señal de tipo string).

2.3. Tercer día

Por fin, el último día, teníamos que crear un instrumento virtual **CreateUserSignal** que envía al generador de forma de onda una señal que equivale a la suma de dos cosenos, usando el dispositivo creado durante el primer día.

Lo que queremos al final es tener instrumento que sea capaz de enseñarnos un plot con la función creada con los parámetros de entrada del usuario, y dos botones : uno que permite parar la ejecución del programa sin enviar la señal, y otro que permite enviar la señal y salir, pero solamente una vez la señal correctamente enviada. Esto se hace modificando el diseño del dispositivo **ControlWaveGenerator** que creamos durante el segundo día.

Por fin, el tercer objetivo del día consistía en modificar un instrumento virtual ya creado para poder controlar el osciloscopio también y ver su pantalla directamente en Labview. Queremos obtener el espectro de la señal que se genera con el oscilador, para poder observar y estudiarla con el osciloscopio directamente en Labview, sin tocar ningún instrumento.

3. Resultados obtenidos

3.1. Primer día

Lo primero que hicimos en el laboratorio consistía en añadir en el panel frontal del nuevo instrumento virtual **Suma2Cos** en Labview los 8 indicadores numéricos que nos permiten definir completamente la señal que queremos crear (igual a la suma de dos señales cosenoidales definidas cada una por su amplitud, se frecuencia y su fase). Se añade también un array que se usará como salida del dispositivo, y se cambian las etiquetas de cada elemento introducido para poner un nombre adecuado.

En el diagrama de bloques, se añade un bloque de tipo *for* y se eligen las operaciones que tenemos que hacer dentro o fuera de esta bucle (por ejemplo, el cambio de frecuencia f a frecuencia angular ω se hace mediante una multiplicación por 2π fuera del bucle, porque queremos hacer esta operación una sola vez).

Dentro del bucle así definido, se hacen todas las operaciones que nos permiten definir nuestra señal de salida :

- Se multiplica la frecuencia angular por el tiempo. Este tiempo se obtiene mediante una multiplicación entre el índice i del bucle y el tiempo de muestreo t_m .
- Sumamos al resultado de esta primera operación la fase ϕ de cada coseno, y calculamos el coseno del resultado de estas dos primeras operaciones.
- Se suman los dos cosenos resultantes y se sale del bucle con este resultado.

El instrumento virtual así creado se representa en la figura 1. Después de este trabajo, podemos volver al panel frontal para definir las entradas y la salida de nuestro instrumento virtual. Además, se crea un icono que nos permite identificar este aparato para poder usar y identificarlo después en otro sistema más complicado.

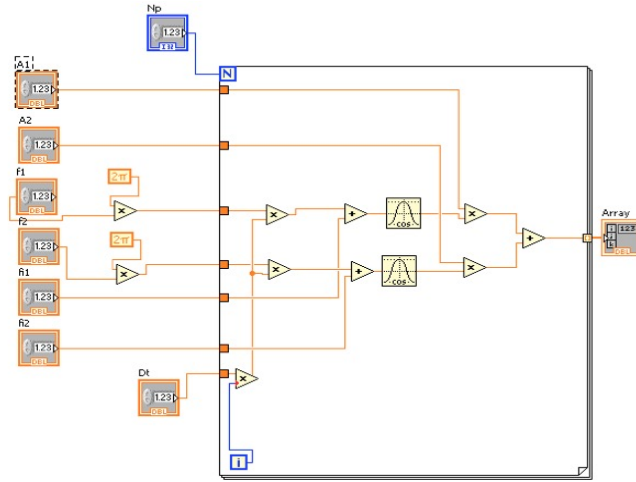


Figura 1: Diagrama de bloque del instrumento virtual **Suma2Cos** que proporciona una señal equivalente a la suma de dos cosenos.

Por fin, se añaden al panel frontal de un nuevo programa principal llamado **PlotSuma2Cos** las 3 gráficas que queremos representar (señal temporal, amplitud del espectro y fase en función de la frecuencia) y se cambian sus etiquetas. Añadimos también en el diagrama de bloques unos dispositivos que nos permiten calcular la transformada de Fourier de nuestra señal y multiplicamos la tensión V_{rms} que obtenemos de esta manera con $\sqrt{2}$, para tener un valor de tensión real que podemos usar. También añadimos un filtro en el caso del espectro de fase para quitar las amplitudes muy pequeñas que pueden dar cualquier resultado en la fase (y que dan entonces mucho ruido en el espectro). Solo se consideran entonces los valores mayores que 0,1 y se anula el resto.

En la figura 2 se representa el programa principal creado, que nos permite usar y pintar el array de salida del instrumento virtual que proporciona la suma de los dos cosenos.

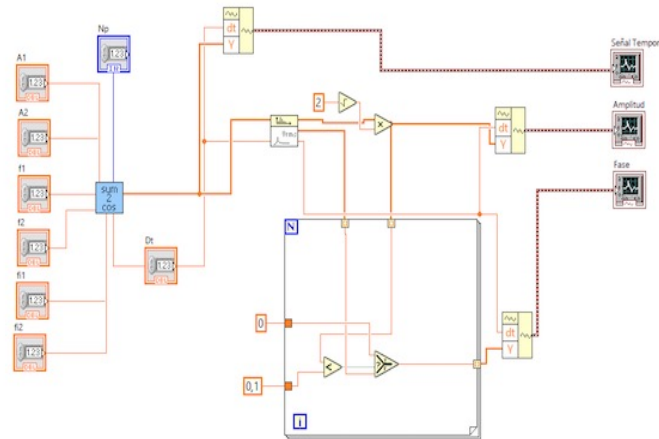


Figura 2: Diagrama de bloque del instrumento virtual **PlotSuma2Cos** que permite generar y representar una señal suma de 2 cosenos en función del tiempo, así que sus espectros de amplitud y de fase.

Con este dispositivo que hicimos, podemos intentar reproducir el fenómeno de *aliasing* explicado en la introducción teórica y que aparece cuando la relación (2) no se cumple. En este caso, subimos el valor de Δt hasta llegar a este limite. Por ejemplo en este caso, vemos que si ponemos una señal suma de dos cosenos de frecuencia de 3000 y 4000 Hz, si elegimos 1000 puntos y un valor de Δt superior a $1,3 \cdot 10^{-4}$, la frecuencia 2 se plega y aparece a unos 3000 Hz (en lugar de los 4000 Hz que debería tener). En la figura siguiente se representa el panel frontal del programa principal creado, con los parámetros elegidos de tal manera que se produzca este fenómeno de *aliasing*.

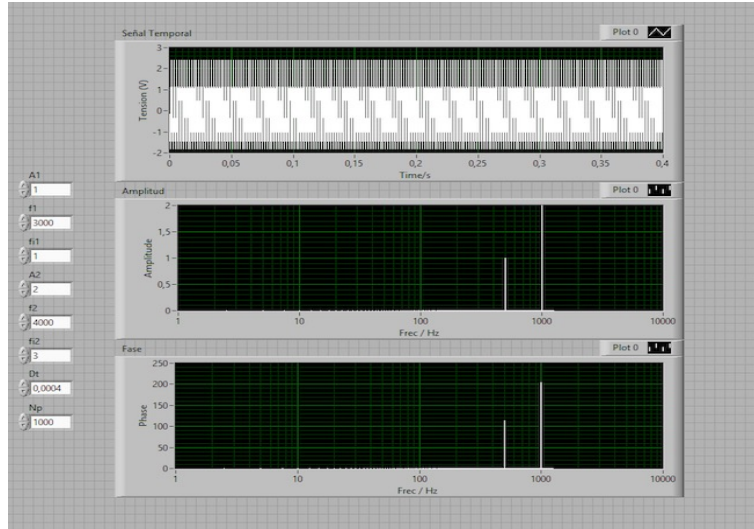


Figura 3: Panel frontal del programa principal **PlotSuma2Cos**, con unos parámetros de entrada y los plots de salida.

3.2. Segundo día

3.2.1. Primer dispositivo

Con este dispositivo llamado hasta ahora **ControlWaveGenerator**, queremos controlar un generador de forma de onda con el protocolo VISA, y en este caso, lo primero que hay que hacer, es abrir un VISA (y no olvidar cerrarlo al final del dispositivo que creamos). Se asocia además a este VISA la dirección del aparato que queremos controlar (el generador de funciones en este caso).

Dentro de un bucle de tipo *while* que definimos, añadimos una estructura de tipo *case* que nos permite elegir lo que hacer con cada uno de los 5 tipos de señales que estamos usando. Se conecta además un botón a esta estructura para poder enviar la señal construida.

Ahora que tenemos el esqueleto de nuestro dispositivo, tenemos que elegir lo que entra dentro de la estructura *case* que acabamos de definir. Queremos enviar señales de 5 tipos diferentes, que tienen amplitudes, frecuencias y offset diferentes. Entonces, definimos primero controles de tipo string (este tipo es el único que nos permite controlar el osciloscopio y el generador) y un *ring*, un control que da un número entero a cada tipo de señal. Asignamos de esta manera el número 1 a la señal senusoidal, el número 2 a la señal triangular, el número 3 a la señal rectangular, el número 4 a la señal del usuario y por fin el número 5 a la señal que viene de la memoria volátil.

Miramos después en el manual para encontrar un comando que permite enviar una señal al dispositivo que queremos controlar. Como el dispositivo se puede controlar solamente con un string de números separados por una coma, tenemos que concatenar los valores de amplitud, frecuencia y offset definidos antes con comas y con el comando de tipo *APPL*: que encontramos en el manual. El string final que obtenemos se conecta al dispositivo *write VISA*.

El último paso consiste en cerrar el VISA abierto (fuera del bucle que acabamos de crear, por supuesto) y en conectar a este dispositivo un aparato con un indicador que nos permite tratar los eventuales errores que podrían aparecer. El dispositivo completo se representa en la figura 4.

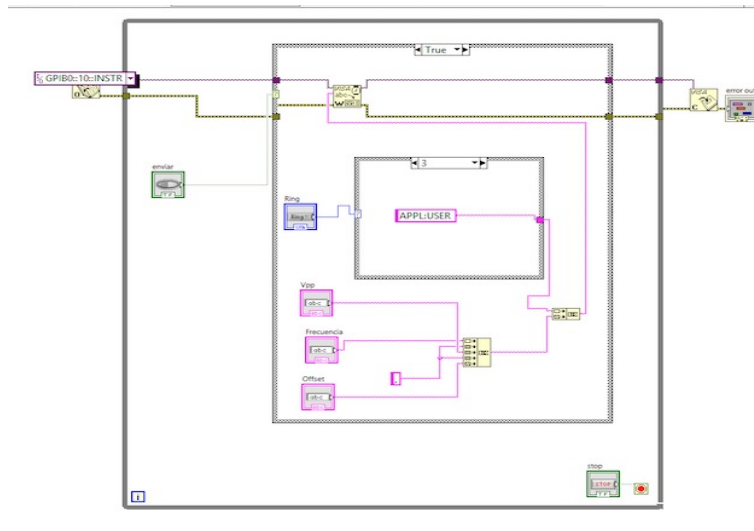


Figura 4: Instrumento virtual **ControlWaveGenerator** creado para controlar el generador de forma de onda y enviarle señales de diferentes tipos.

3.2.2. Segundo dispositivo

Queríamos también este día crear un otro instrumento virtual (conocido como **ConvertToString**) que convierte un array de números en coma flotante a un string de enteros normalizados entre -2047 y 2047 (para dar como señal al oscilador).

Este dispositivo se crea con un array de entrada, del cual queremos sacar el valor máximo para después dividir cada elemento del array por el valor del ratio $\frac{2047}{\text{maximo}}$. Este se hace con un bucle *for* para deshacer el array. Dentro de este bucle, convertimos cada número entero así creado a un string y separamos cada número por una coma para crear la forma de onda que queremos (los strings se adjuntan usando los registros de desplazamientos). Este dispositivo se guarda como un nuevo instrumento virtual y se representa en la figura 5.

En el laboratorio, queríamos también verificar si el bloque que acabamos de crear funciona correctamente. Para esto, creamos un nuevo dispositivo muy sencillo. Queríamos verificar que el valor máximo de entrada corresponde bien al número 2047 (o -2047) en la salida, y que cada número de entrada se convierte correctamente a un número entero cuya valor absoluto tiene valores entre 0 y 2047.

3.3. Tercer día

Durante el último día, se hizo primero el instrumento virtual **CreateUserSignal**, que tiene que ser capaz de enviar al generador de forma de onda la señal suma de dos cosenos. Para crear este dispositivo, cargamos entonces el instrumento virtual **Suma2Cos** creado el primer día y le conectamos a un plot, que nos permitirá ver la pinta que tiene la función que estamos creando. Después, creamos un botón *mandar* que conectamos a un enlace de tipo *case*. En el caso *true*, cuando activamos este último botón, queremos llamar al instrumento virtual **ConvertToString** que hicimos durante el segundo día de práctica, para convertir los números reales que salen de **Suma2Cos** a un string de números enteros normalizados, que el generador es capaz de leer.

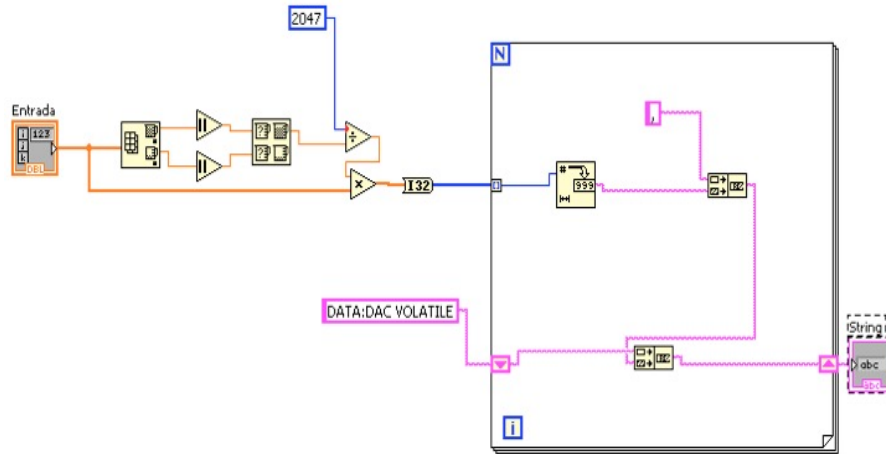


Figura 5: Bloque **ConvertToString** que permite convertir un array de números en coma flotante a un string de números enteros normalizados entre -2047 y 2047.

Lo único que falta en este instrumento virtual es un VISA que nos permite controlar el aparato directamente. Dentro del caso *true* del enlace, añadimos entonces un primero *VISA write* que conectamos a la salida del bloque **ConvertToString**, y un segundo *VISA write* que conectamos a la instrucción *FUNC:USER VOLATILE*. Esto nos permite guardar la señal de usuario en la memoria volátil. Todo este dispositivo se representa en la figura 6.

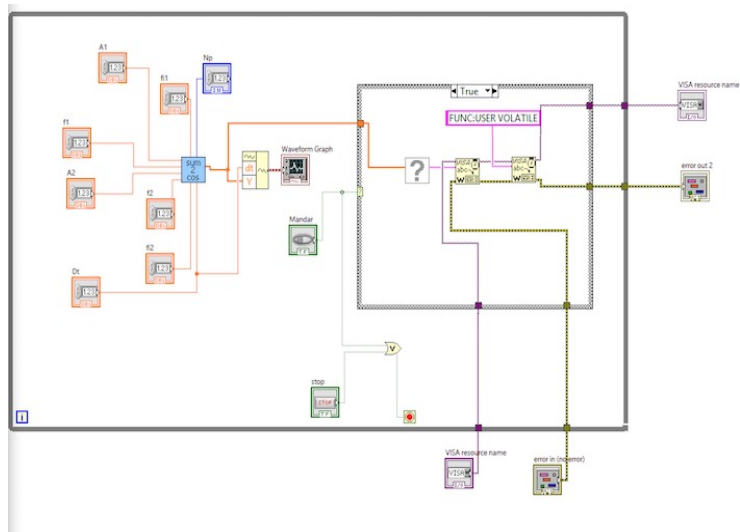


Figura 6: Instrumento virtual **CreateUserSignal** que permite guardar en la memoria volátil la señal de tipo usuario que corresponde a la suma de dos cosenos.

El paso siguiente consistía en modificar el diseño del dispositivo **ControlWaveGenerator** que creamos el segundo día para poder cumplir el objetivo que tenemos. Queremos modificar este instrumento para poder enviar una señal de tipo usuario también, que para nosotros será la suma de los dos cosenos. Creamos entonces un nuevo botón, *user signal*, que conectamos a una nueva estructura de tipo case. Si activamos este botón, queremos que el programa llame al instrumento virtual **CreateUserSignal** y que aparezca su panel frontal en el cual podemos modificar los parámetros de la señal (amplitud, frecuencia y fase de los dos cosenos) y donde podemos ver un plot con la señal que estamos generando. Después, podemos elegir si enviar la señal o si salir sin enviarla. El dispositivo así creado se representa en la figura 7.

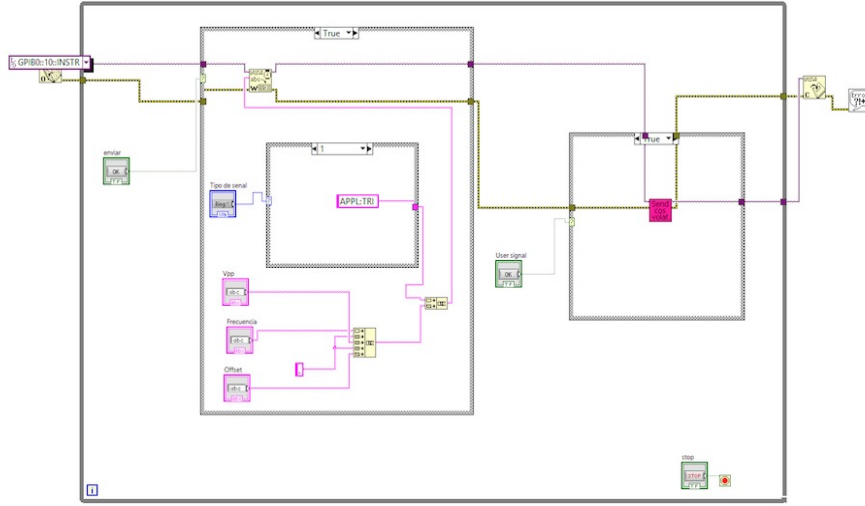


Figura 7: Última modificación del instrumento virtual **ControlWaveGenerator** para que sea capaz de enviar una señal de tipo suma de dos cosenos al generador de forma de onda.

El último paso consistía en modificar un poco un instrumento virtual ya creado por otra persona, que permite ver y controlar el osciloscopio directamente con Labview. Lo modificamos un poco para poder pintar la amplitud de salida además del plot que representa la pantalla del osciloscopio que está por defecto. Añadimos también un botón que permite actualizar las informaciones que aparecen en los plots, y un botón plot para salir de la ejecución del programa. El dispositivo final se representa en la figura 8, con su panel frontal y los plots obtenidos en las figuras 9 y 10.

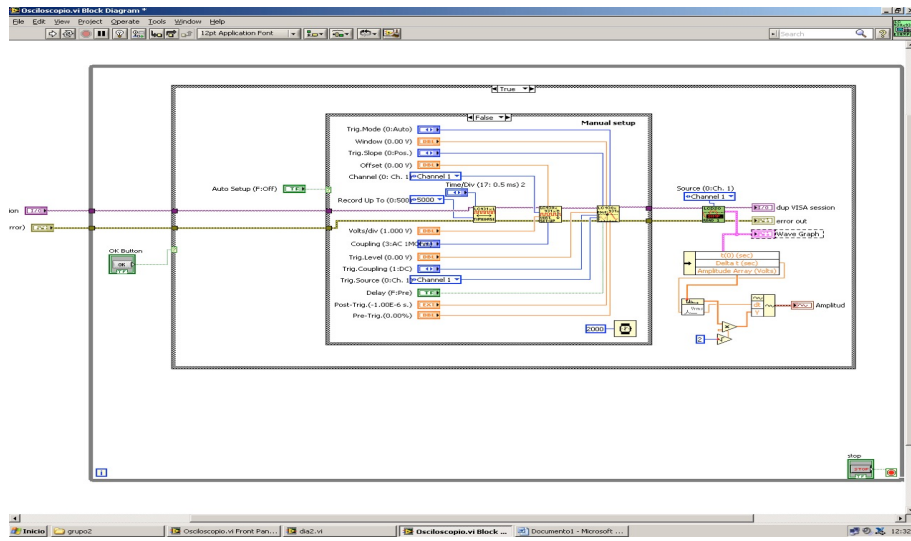


Figura 8: Diagrama de bloque del intrumento **ControlOscilloscope** que permite controlar y ver la pantalla del osciloscopio directamente en Labview.

En los dos casos de salidas, enviamos como señal la suma de dos cosenos de frecuencias de 1000 y 4000 Hz, de amplitud 1 y 3V, con un tiempo de muestreo de $10^{-6}s$. Entonces, para tener un periodo complete, hay que enviar 1000 puntos, según la relación (4).

$$N_p = \frac{1}{\Delta t \cdot MCD(1000, 4000)} = 1000 \quad (4)$$

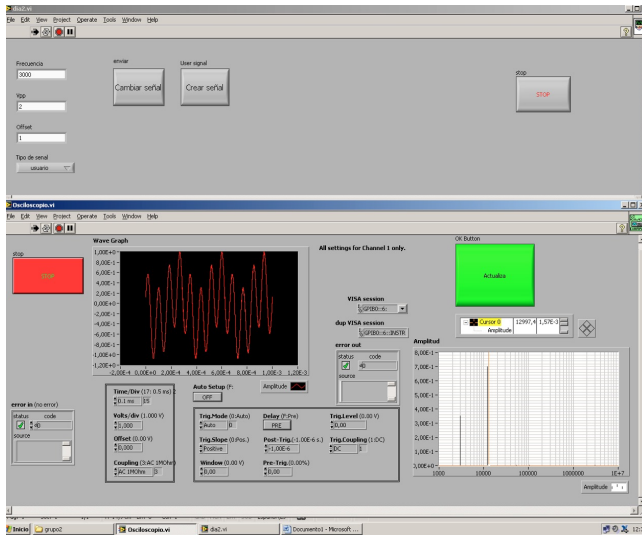


Figura 9: Salida obtenida si enviamos exactamente un periodo al generador.

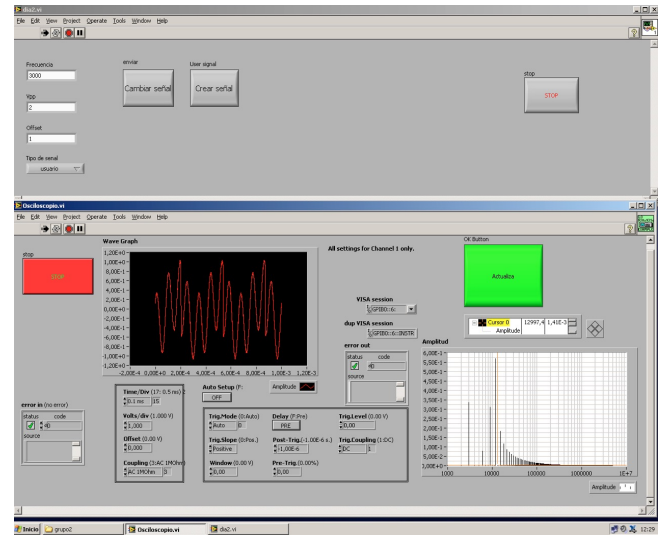


Figura 10: Salida obtenida si enviamos menos de un periodo al generador.

Si enviamos 1000 puntos, estamos en el caso de la figura 10 mientras que si enviamos solamente 900 puntos al generador de forma de onda, estamos en el caso de la figura 11 donde cuando las señales se juntan, saltos aparecen y aparecen muchas frecuencias más que los dos picos a 3000 y 12000Hz que teníamos antes (podemos observar estos dos valores de frecuencia porque son dos números que tienen un máximo común divisor igual a 3000 Hz, la *frecuencia por modulación* que enviamos, y un ratio entre las frecuencias exactamente igual a 4, el valor del ratio entre las frecuencias de la señal suma de dos cosenos que enviamos).

4. Conclusión

En conclusión, esta práctica era una primera introducción al uso del software Labview para crear, observar y analizar diferentes tipos de señales (senusoidal, triangular, cuadrada, suma de dos cosenos y de la memoria volátil). Creamos 6 diferentes instrumentos virtuales para lograr este objetivo :

- **Suma2Cos**, que nos permite crear una señal suma de dos cosenos cuyos parámetros principales (amplitudes, frecuencias, fases, tiempo de muestreo y número de puntos) son elegidos por el usuario.
- **PlotSuma2Cos**, que permite representar la señal de salida del primer dispositivo de diferentes maneras (representar la señal en función del tiempo, así que sus espectros de amplitud y de fase).
- **ControlWaveGenerator**, que nos permite controlar el generador de forma de onda usando un dispositivo VISA para generar le señal del tipo que nos interesa.
- **ConvertToString**, que permite convertir un array de números a coma flotante a un string de números enteros normalizados entre -2047 y 2047 que podemos usar para controlar los diferentes dispositivos.
- **CreateUserSignal**, capaz de enviar una señal del tipo suma de dos cosenos al generador de forma de onda.
- **ControlOscilloscope**, que controla el osciloscopio y que permite ver su pantalla directamente en el ordenador.

Pudimos además observar con este montaje el fenómeno de *aliasing* que aparece cuando la frecuencia de la señal supera el valor limite descrito en la introducción teórica.