

UNIVERSIDAD CARLOS III DE MADRID



Ingeniería Técnica Industrial Electricidad

Proyecto Fin de Carrera

Instrumento virtual para la medida de descargas parciales

Autor:

José Manuel Fresno de Marcos

Director:

Dr. D. Guillermo Robles Muñoz

Escuela Politécnica Superior
Departamento de Ingeniería Eléctrica

Leganés, 2009

Índice general

Índice general	2
Índice de figuras	6
1 Objetivos.....	11
2 Introducción	12
3 Programa creado con <i>Labview</i>	15
3.1 Block diagram.....	15
3.1.1 Descripción de las principales acciones realizadas en la aplicación.	15
3.1.2 Diagrama de bloques con las principales acciones realizadas en la aplicación.....	17
3.1.3 Adquisición.....	17
3.1.4 Resta de la media en el canal 0.....	22
3.1.5 Resta de la media en el canal 1.....	24
3.1.6 Filtro canal 0 y canal 1.....	24
3.1.7 Ganancia en el canal 1.....	25
3.1.8 Integración tensión → intensidad.....	26
3.1.9 Recortar señal	27
3.1.10 Integración intensidad → carga	28
3.1.11 Calcula el primer máximo de la integral	29
3.1.12 Automatización de la calibración.....	30
3.1.13 Indicadores de carga máxima, mínima e instantánea	32
3.1.14 Parámetros del indicador de intensidad.....	33
3.1.15 Indicadores de carga promediada.....	35
3.1.16 Controla la escala vertical de los gráficos filtrada y entrada.....	36
3.1.17 Controla el valor de <i>trigger</i> y lo asocia a los cursores X e Y de los gráficos entrada y filtrada.....	37
3.1.18 Controla la escala de <i>trigger Y</i>	38
3.1.19 Calculo del valor máximo y mínimo absoluto de la integral hardware (canal 0) y la integral software (canal 1)	38

INDICE GENERAL

3.1.20	Ajuste de los límites de los indicadores de aguja y el grafico comparación integral hardware Vs software.....	40
3.1.21	Plantillas	41
3.1.22	Gráficos	41
3.2	Front panel	43
3.2.1	Ajuste Adquisición	43
3.2.2	Ajuste del filtro	43
3.2.3	Grafico señal de entrada	45
3.2.4	Grafico señal filtrada.....	46
3.2.5	Grafico comparación integral software Vs hardware	46
3.2.6	Calibración	48
3.2.7	Medidas en pico Culombios.....	48
3.2.8	Medidas en amperios	49
4	Resultados experimentales.....	51
4.1	Algoritmos de cálculo de carga mediante integral numérica	51
4.1.1	Integración numérica de toda la señal	51
4.1.2	Integración numérica de la señal recortada	52
4.2	Algoritmos de estimación de carga mediante la relación de un parámetro....	53
4.2.1	Estimación de la carga a partir de la primera área positiva con el tiempo base fijado por cursores.....	53
4.2.2	Estimación de la carga a partir de la primera área positiva con el tiempo base igual al doble del tiempo de cola	54
4.2.3	Estimación de la carga a partir de la primera área positiva estimando el tiempo de inicio.....	57
4.2.4	Estimación de la carga a partir del primer mínimo de la integral.	58
4.2.5	Estimación de la carga a partir del primer máximo de la integral.	60
5	Conclusiones y trabajos futuros.	67
5.1	Conclusiones.....	67
5.2	Trabajos futuros.....	67
6	Anexo 1-Elementos utilizados de labview	68

INDICE GENERAL

6.1	Estructuras:	68
6.1.1	<i>While loop</i>	68
6.1.2	<i>For loop</i>	68
6.1.3	<i>Case structure</i>	68
6.1.4	<i>Flat Sequence Structure</i>	69
6.2	Formato de datos:.....	70
6.3	Conversión entre tipos de datos	71
6.4	Bloques	72
6.4.1	Tarjeta de adquisición de datos niScope	72
6.4.2	Filtro	80
6.4.3	Integración.....	83
6.4.4	Media	84
6.4.5	Amplitud and level measurements.....	84
6.4.6	Fórmula	85
6.4.7	Creador de <i>array</i>	86
6.4.8	Localizador de máximo y mínimo en un <i>array</i>	86
6.4.9	Localizador de un elemento en un <i>array</i>	87
6.4.10	Extrae una porción de la señal.	87
6.4.11	Puerta lógica Or.....	88
6.4.12	Sumador.....	89
6.4.13	Comparador	90
6.4.14	Negador.....	90
6.4.15	Realimentación.....	91
6.4.16	Detector de picos.....	91
6.4.17	Plantilla de selección	92
6.4.18	<i>Property node</i>	93
6.4.19	Shift register	93
6.4.20	Primera ejecución.....	93

INDICE GENERAL

6.4.21	Identificador de estructuras.....	94
6.4.22	Agrupador de conductores.....	95
6.5	Controles:	95
6.5.1	Numeric control	95
6.5.2	Deslizador	96
6.5.3	Pulsador	96
6.5.1	Interruptor.....	97
6.6	Indicadores.....	97
6.6.1	Indicador grafico	97
6.6.2	Indicador numérico	98
6.6.3	Indicador de aguja	98
6.6.4	Indicador luminoso.....	98
7	Bibliografía.....	100

Índice de figuras

Figura 1.1 Representación de las imperfecciones en el aislante.....	13
Figura 1.2: Evolución de la tensión e intensidad en la vacuola.	13
Figura 1.3: Modelo eléctrico ABC y circuito eléctrico equivalente.....	13
Figura 1.4: Montaje del circuito de adquisición.....	14
Figura 2.1: Esquema completo del programa creado.....	16
Figura 2.2: Conexionado del diagrama de bloques para la adquisición.....	23
Figura 2.3: Conexionado del diagrama de bloques para la resta de la media en el canal 0.....	25
Figura 2.4: Conexionado del diagrama de bloques para la resta de la media en el canal 1.....	25
Figura 2.5: Conexionado del diagrama de bloques de los filtros.	25
Figura 2.6: Conexionado del diagrama de bloques para aplicar ganancia al canal 1.	26
Figura 2.7: Conexionado del diagrama de bloques para la integración tensión → intensidad.	27
Figura 2.8: Conexionado del diagrama de bloques para recortar la señal.....	28
Figura 2.9: Conexionado del diagrama de bloques para la integración intensidad → carga	29
Figura 2.10: Conexionado del diagrama de bloques para el cálculo del primer pico de la señal integrada.	29
Figura 2.11: Conexionado del diagrama de bloques para la calibración.....	31
Figura 2.12: Conexionado del diagrama de bloques para el cálculo de máximo, mínimo y valor actual de la carga en pC.	33
Figura 2.13: Conexionado del diagrama de bloques para el cálculo de máximo y mínimo pico de intensidad.	34
Figura 2.14: Conexionado del diagrama de bloques para el cálculo de la carga promediada.....	35
Figura 2.15: Conexionado del diagrama da bloques para controlar la escala vertical de los gráficos entrada y filtrada.	36
Figura 2.16: Conexionado del diagrama de bloques que controla el <i>trigger</i> y los cursores de los gráficos entrada y filtrada.....	37
Figura 2.17: Conexionado del diagrama de bloques para el control del <i>trigger</i>	39

INDICE DE FIGURAS

Figura 2.18: Conexionado del diagrama de bloques para el Cálculo de escala del grafico comparación integral software Vs hardware.....	39
Figura 2.19: Conexionado del diagrama de bloques para el ajuste de los límites del grafico comparación integral software Vs hardware y de los indicadores de aguja.	42
Figura 2.20: Representación grafica del diagrama de bloques para las plantillas	42
Figura 2.21: Conexionado del diagrama de bloques para los gráficos.	42
Figura 2.22: Visualización <i>Front panel</i> del programa creado.....	44
Figura 2.23: Representación grafica en <i>Front panel</i> de la plantilla ajuste de adquisición.....	45
Figura 2.24: Representación grafica en <i>Front panel</i> de la plantilla ajuste del filtro.	45
Figura 2.25: Representación grafica en <i>Front panel</i> del grafico señal de entrada.	46
Figura 2.26: Representación grafica en <i>Front panel</i> del grafico señal filtrada.	47
Figura 2.27: Representación grafica en <i>Front panel</i> del grafico comparación integral software Vs hardware.	47
Figura 2.28: Representación grafica en <i>Front panel</i> de la plantilla calibración.	49
Figura 2.29: Representación grafica en <i>Front panel</i> de la plantilla medir en pC.....	49
Figura 2.30: Representación grafica en <i>Front panel</i> de la plantilla medir en amperios.	50
Figura 3.1: Representación de la señal integrada mediante software sin restar la media 1.....	52
Figura 3.2: Representación de la señal integrada mediante software sin restar la media 2.....	52
Figura 3.3: Representación de la señal integrada mediante software y recortada sin restar la media 1.	53
Figura 3.4: Representación de la señal integrada mediante software y recortada sin restar la media 1.	53
Figura 3.5: Representación de la señal integrada mediante software restando la media y con dos cursores para calcular la primera área.	56
Figura 3.6: Representación de la señal dos veces integrada con todos sus valles señalizados en rojo.....	59
Figura 3.7: Representación de la señal dos veces integrada con todos sus picos señalizados en rojo.....	61

INDICE DE FIGURAS

Figura 3.8: Representación de la señal de entrada y de la segunda integral recortada después de integrar.	62
Figura 3.9: Representación del error que se comete por el ruido antes del disparo... 63	63
Figura 3.10: Representación de la primera integral recortada a partir del <i>trigger</i>	63
Figura 3.11: Representación de la segunda integral recortada a partir del <i>trigger</i>	63
Figura 3.12: Representación de la segunda integral y de la señal de entrada recortadas antes de integrar.	64
Figura 5.1: Representación grafica de la estructura <i>While loop</i>	69
Figura 5.2: Representación grafica de la estructura <i>For loop</i>	69
Figura 5.3: Representación grafica de la estructura <i>Case structure</i>	69
Figura 5.4: Representación grafica de la estructura <i>Flat Sequence Structure</i>	70
Figura 5.5: Conversor dynamic data.	72
Figura 5.6: Conversor de <i>array</i> a cluster y viceversa.	72
Figura 5.7: Extractor-Agrupador de las componentes <i>waveform</i>	72
Figura 5.8: Conversor números reales en números enteros (64bits de tamaño).	72
Figura 5.9: Bloque para la configuración horizontal de la tarjeta de adquisición.....	73
Figura 5.10: Indicador del error.	74
Figura 5.11: Bloque para la configuración de los canales de la tarjeta de adquisición.	74
Figura 5.12: Bloque para la configuración vertical de la tarjeta de adquisición.....	76
Figura 5.13: Bloque para la configuración del <i>trigger</i>	76
Figura 5.14: Bloque para iniciar la adquisición.	78
Figura 5.15: Bloque selector entre los distintos formatos de datos.....	78
Figura 5.16: Inicializar ni scope.	80
Figura 5.17: Cerrar ni scope.	80
Figura 5.18: Bloque del filtro.	81
Figura 5.19: Comportamiento de la magnitud y la fase en función de la frecuencia del filtro <i>butterworth</i>	81
Figura 5.20: Comportamiento de la magnitud y la fase en función de la frecuencia del filtro <i>chebyshev</i>	81

INDICE DE FIGURAS

Figura 5.21: Comportamiento de la magnitud y la fase en función de la frecuencia del filtro <i>inverse chebyshev</i>	82
Figura 5.22: Comportamiento de la magnitud y la fase en función de la frecuencia del filtro <i>elliptic</i>	82
Figura 5.23: Comportamiento de la magnitud y la fase en función de la frecuencia del filtro <i>bessel</i>	82
Figura 5.24: Bloque integrador.	84
Figura 5.25: Bloque que calcula la media del <i>array</i> de entrada.	84
Figura 5.26: Bloque devuelve los parámetros característicos de la señal.	85
Figura 5.27: Bloque fórmula.	86
Figura 5.28: Bloque creador de <i>array</i>	87
Figura 5.29: Bloque que devuelve el elemento mayor y el menor de un <i>array</i>	87
Figura 5.30: Bloque localizador de un elemento en un <i>array</i>	88
Figura 5.31: Bloque que recorta la señal.	88
Figura 5.32: Bloque puerta lógica or.	89
Figura 5.33: Bloque sumador.	90
Figura 5.34: Bloque comparador $X > Y$	90
Figura 5.35: Bloque negador.	90
Figura 5.36: Bloque para la realimentación.	92
Figura 5.37: Bloque que devuelve los picos o valles de la señal.	92
Figura 5.38: Plantillas	94
Figura 5.39: Enlaces con parámetros de los componentes de la aplicación.	94
Figura 5.40: Bloque transporta registro.	94
Figura 5.41: Bloque primera ejecución.	94
Figura 5.42: Bloque identificador de estructuras.	95
Figura 5.43: Bloque agrupador.	96
Figura 5.44: Representación grafica de los controles Numeric.	96
Figura 5.45: Representación grafica del control deslizador.	97
Figura 5.46: Representación grafica del pulsador reset.	97

INDICE DE FIGURAS

Figura 5.47: Representación grafica del interruptor.....	97
Figura 5.48: Representación grafica del indicador grafico.....	98
Figura 5.49: Representación grafica del indicador numérico.....	99
Figura 5.50: Representación grafica del Indicador de aguja.....	99
Figura 5.51: Representación grafica del indicador luminoso.	99

OBJETIVOS

1 Objetivos.

- Adquirir el pulso de alta frecuencia de la descarga parcial.
- Filtrar el pulso con un filtro paso bajo o paso banda.
- Integrar la señal dada por la sonda inductiva para obtener la intensidad del pulso.
- Representar la señal al estilo de un osciloscopio
- Cálculo de la carga desplazada por la descarga parcial

2 Introducción

Los aislantes de los componentes eléctricos no son perfectos, en su fabricación se cometen fallos inevitables que impiden mantener el mismo nivel de aislamiento a lo largo del aislante. Estas imperfecciones son huecos de aire o huecos rellenos de gas Figura 2.1.

Las descargas parciales son un fenómeno eléctrico que se produce en las oquedades de los aislantes de los componentes eléctricos. Consisten en rupturas locales que generan pulsos rápidos de corriente que propician un deterioro progresivo del aislante y la perforación del dieléctrico con el tiempo. Cuando el componente trabaja en condiciones nominales, el campo eléctrico dentro de las oquedades es mucho mayor que en el aislante debido a que la constante dieléctrica del aire o de los gases que puede haber dentro de los huecos es mucho menor que la del material aislante [1][2][3].

La tensión existente dentro de las vacuolas varía en función de la tensión V aplicada al componente eléctrico. Si la tensión de la vacuola V_c supera la tensión de ruptura del dieléctrico V_{rd} se produce una ruptura del dieléctrico y por tanto una descarga parcial que reduce la tensión en la vacuola hasta alcanzar la tensión de extinción V_e [4][5]. Cuando se extingue la descarga parcial vuelve a variar la tensión en la vacuola hasta alcanzar V_{rd} o $-V_{rd}$ porque este fenómeno también se produce en el semiciclo negativo. El fenómeno descrito en este párrafo se muestra en la Figura 2.2. La actividad las descargas parciales se reduce en la cresta de la tensión aplicada al componente eléctrico debido a que los incrementos de tensión son menores y por tanto se tarda más tiempo en alcanzar los valores V_{rd} o $-V_{rd}$. Finalmente conforme se van produciendo descargas la intensidad de las mismas disminuye siendo un claro indicador del estado del aislante [6].

La magnitud de las descargas parciales depende de las dimensiones, del contenido, de la temperatura y de la presión de la cavidad como de las condiciones y características del aislante lo que hace del fenómeno de las descargas parciales tenga un carácter estocástico [7].

Para el estudio del mecanismo de las descargas parciales, se utiliza el modelo eléctrico ABC propuesto por Whitehead [8], representa el aislante donde ocurre la descarga mediante una combinación de tres capacidades [9]. En la Figura 2.3 se puede observar el aislante sometido a tensión con la región para capacidad: a y b representan las capacidades para el material en buen estado y c representa la cavidad de aire o gas [10].

En este proyecto de fin de carrera se ha utilizado un sensor inductivo de bajo coste diseñado en el Departamento de Ingeniería Eléctrica. Este sensor se basa en la ley de inducción de Faraday, está formado por una espira rectangular capaz de medir pulsos de alta frecuencia. La espira está colocada junto al hilo que transporta la descarga parcial, cuando se produce una descarga circula un pulso de corriente por el hilo conductor, esta corriente genera un campo magnético que al atravesar el área de

INTRODUCCIÓN

la espira produce una diferencia de tensión en la misma. Esta tensión captada por la espira es la derivada de la descarga parcial reducida por el coeficiente de la

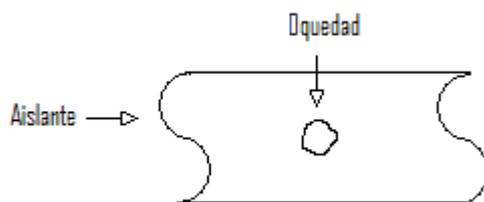


Figura 2.1 Representación de las imperfecciones en el aislante

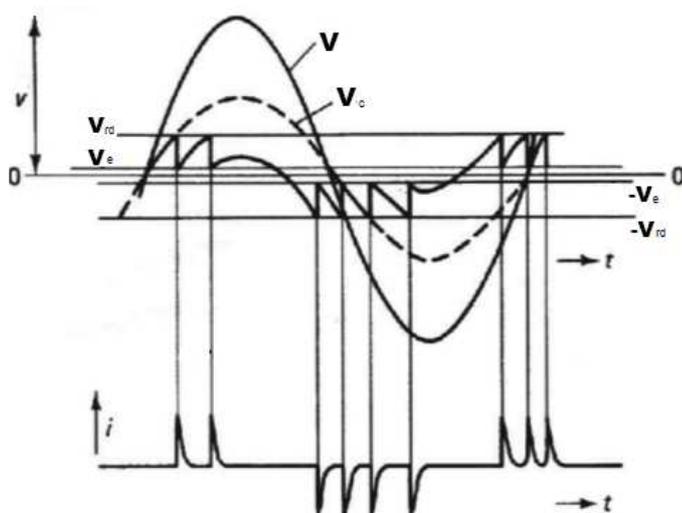


Figura 2.2: Evolución de la tensión e intensidad en la vacuola.

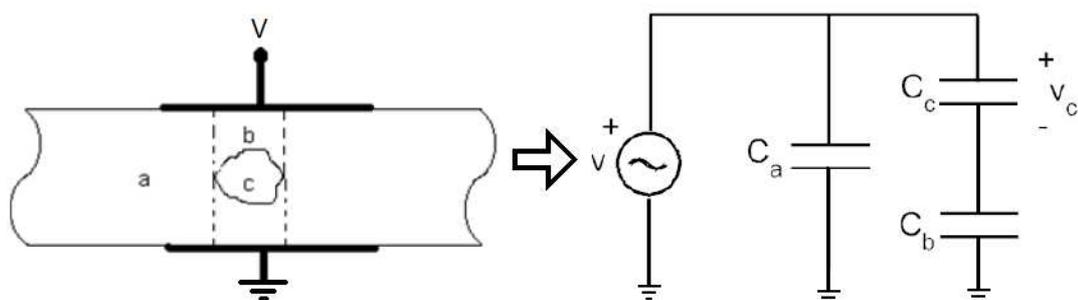


Figura 2.3: Modelo eléctrico ABC y circuito eléctrico equivalente.

inductancia mutua. Mediante la tarjeta de adquisición PCI-5152 de *National Instruments* conectada a un PC con el software *Labview* se procesan los datos que devuelven los sensores inductivos, un sensor simplemente adquiere la señal y el otro además de adquirir devuelve la señal integrada. El montaje para medir las descargas parciales con estos sensores es el mostrado en la Figura 2.4 y se basa en la norma IEC 60270.

INTRODUCCIÓN

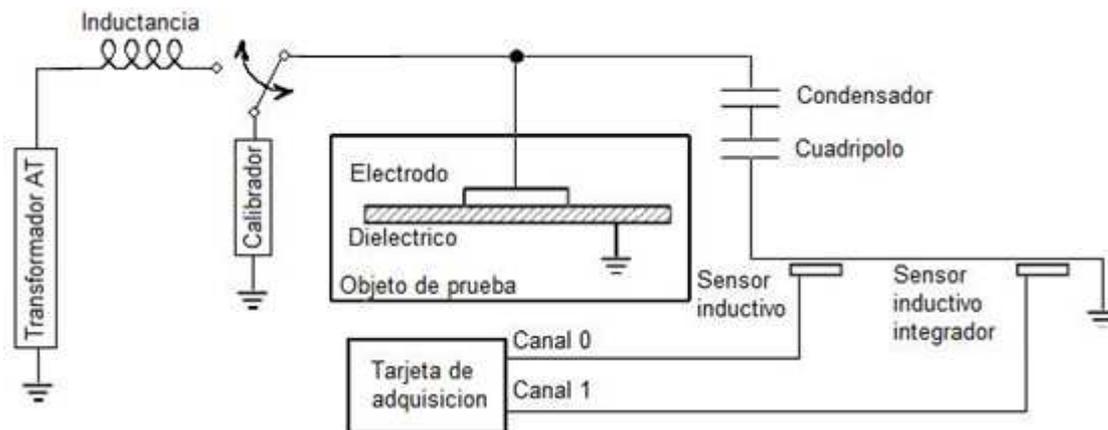


Figura 2.4: Montaje del circuito de adquisición

El principal motivo de realizar la adquisición con una tarjeta de *National Instruments* es porque es capaz de adquirir a una frecuencia de muestreo elevada ya que las descargas parciales son señales muy rápidas. La tarjeta utilizada en este proyecto tiene una frecuencia de muestreo máxima de 1GHz, que se puede doblar si se utiliza la función *interleaved sampling* y una resolución de 8bits.

Para obtener una señal que sea similar a la descarga parcial se utiliza *Labview* que permite adquirir, controlar y analizar datos mediante aplicaciones basadas en diagramas de bloques. Las aplicaciones creadas con *Labview* [11] se denominan VIs (Instrumentos virtuales), están compuestas por dos partes: panel frontal, que permite visualizar indicadores y modificar contrales, y diagrama de bloques que constituye el código fuente es donde se desarrolla la aplicación que procesa las entradas de la tarjeta y las salidas creadas en el panel frontal.

La aplicación que se ha desarrollado con *Labview* permite visualizar la señal original de la descarga parcial de forma clara y el cálculo de la carga de la descarga con un error inferior al 7% partiendo de la tensión inducida en la sonda que es una señal muy rápida, débil y con mucho ruido.

3 Programa creado con Labview.

Labview está formado por dos partes: *Front panel* que permite visualizar y trabajar con los datos adquiridos y *Block diagram* que constituye el código fuente, es donde se desarrolla la aplicación que procesa las entradas y salidas creadas en el *Front panel*.

3.1 *Block diagram*

En esta parte de *Labview* se programan las acciones que se tienen que realizar para obtener los objetivos propuestos, los dos objetivos principales de esta aplicación son:

- Adquirir y representar pulsos de descargas parciales.
- Obtener la carga de la descarga parcial en pC.

El esquema completo del programa generado *Labview* se muestra en la Figura 3.1.

3.1.1 Descripción de las principales acciones realizadas en la aplicación.

La sonda que se utiliza para la medición de las descargas parciales es una espira contigua al cable por el que discurren las descargas parciales y la señal que devuelve la sonda es la tensión que se induce en bornes de la espira que se corresponde a la derivada de la intensidad de la descarga parcial afectada por un coeficiente que es la inductancia entre el cable y la espira. Por tanto para tener una señal parecida a la señal original de la descarga parcial se debe integrar la señal adquirida. La señal a la salida de la sonda es $u(t) = \frac{di(t)}{dt} \cdot m$, se quiere conocer la magnitud real de la descarga parcial, para ello se integra la ecuación original y se divide entre la inductancia $i(t) = \frac{\int u(t) \cdot dt}{m}$. Cuando se realiza esta integral se pasa de tener una tensión a tener una intensidad.

Después de la integral se divide la señal de salida del bloque integral entre 54,9nH (valor de la inductancia mutua de la espira), de esta manera se obtiene una señal proporcional a la señal de la descarga parcial original.

A continuación se quiere calcular la carga de la descarga parcial a partir de la señal semejante a la descarga parcial original, para ello se parte de la siguiente ecuación $i(t) = \frac{dQ(t)}{dt}$, operando e integrando queda de la siguiente manera $Q(t) = \int i(t) \cdot dt$, que es valor de la carga de la descarga parcial. De esta manera realizando la integral discreta se obtiene la carga de la descarga parcial.

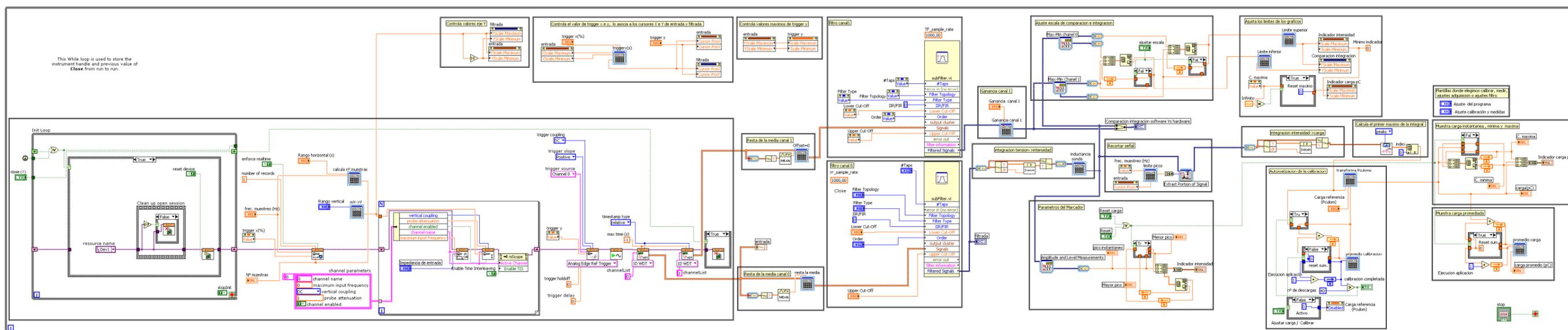


Figura 3.1: Esquema completo del programa creado.

PROGRAMA CREADO CON LABVIEW

La carga es un valor numérico no una señal, si la señal no tuviese ruido, la integral numérica daría un resultado con poco error pero no es así, existe mucho ruido que perturba la carga. Para evitar este error se utiliza la integral discreta en lugar de la numérica, se calcula el primer pico de esta segunda integral y se realiza una regla de tres. Para poder realizar la regla de tres se debe calibrar (memorizar el primer pico de la segunda integral para una carga de referencia) y posteriormente con el valor de pico que se obtenga a cada instante se puede estimar la carga de ese instante:

$$Q_{instantanea} = \frac{Q_{referencia}}{Primer\ pico_{calibracion}} \cdot Primer\ pico_{instantaneo}$$

En la aplicación se realizan mas acciones que describen detalladamente en apartado posteriores al diagrama de bloques con las principales acciones realizadas en la aplicación.

3.1.2 Diagrama de bloques con las principales acciones realizadas en la aplicación.

Estados	Acciones
Descarga parcial	
↓	Adquisición
Descarga parcial adquirida [Voltios]	
↓	Integración discreta
Descarga parcial adquirida [Amperios]	→ Indicadores de intensidad
↓	Inductancia Mutua
Descarga parcial (Proporcional a la real)	
↓	Integración discreta
Carga de la descarga parcial [pC]	
↓	Calibración
Valores representativos de carga [pC]	→ Indicadores de carga pC

SUBCIRCUITOS DE LA APLICACIÓN.

3.1.3 Adquisición.

El esquema eléctrico del subcircuito *bloque de adquisición* se muestra en la Figura 3.2. En este subcircuito se configura la adquisición para devolver la señal adquirida en el formato deseado.

Este subcircuito comienza comprobando que la tarjeta de adquisición utilizada esté lista para la adquisición después el bloque *Ni Scope initialize* crea una nueva sesión de adquisición, este bloque mediante el cable *instrument handle* indica al resto de bloques involucrados en la adquisición la tarjeta de adquisición que se utiliza. Tras crear una nueva sesión de adquisición se ejecutan los bloques que configuran los parámetros de la adquisición:

3.1.3.1 Bloque *Ni Scope configure horizontal timing*.

3.1.3.1.1 Descripción: Este bloque configura los parámetros horizontales de la adquisición.

PROGRAMA CREADO CON LABVIEW

3.1.3.1.2 Terminales de entrada:

- Controles: *Frecuencia de muestreo* y *Rango horizontal* que se usan para calcular el número de muestras para la adquisición mediante la fórmula *Calcula nº muestras* con la ecuación:

$$N^{\circ} \text{ de muestras} = \text{Frecuencia de muestreo} \cdot \text{Rango horizontal}$$

A la salida de esta fórmula está conectado el indicador *Nº muestras*.

- Constantes: *Enforce realtime* y *Number of record* son dos constantes, no se han utilizado controles para estos parámetros porque no resulta interesante modificarlos durante la adquisición.
- *Property node*: El parámetro valor de *trigger x* que es el valor de un control ubicado en otro subcircuito de la aplicación.
- Salidas de otros bloques: *Instrument handle* es la entrada que asigna a los parámetros horizontales configurados a la tarjeta de adquisición utilizada, este terminal esta cableado desde el bloque *Ni Scope initialize*.

3.1.3.1.3 Terminales de salida:

- Entradas para otros bloques: *Instrument handle out*, devuelve el hardware al que se han ajustado los parámetros horizontales para continuar realizando mas configuraciones en otros bloques, este terminal está conectado al bloque *Ni Scope configure chan characteristics*.

3.1.3.2 Bloque *Ni Scope configure chan characteristics*.

3.1.3.2.1 Descripción: Este bloque permite la configuración de los canales de la tarjeta de adquisición.

3.1.3.2.2 Terminales de entrada:

- Controles: *Impedancia de entrada*
- Constantes: *Chanel name* y *Max input frequency* (ancho de banda).
- Salidas de otros bloques: *Instrument handle* es la entrada que asigna a los parámetros de los canales configurados a la tarjeta de adquisición utilizada, este terminal esta cableado desde el bloque *Ni Scope configure horizontal timing*.

PROGRAMA CREADO CON LABVIEW

3.1.3.2.3 Terminales de salida:

- Entradas para otros bloques: *Instrument handle out*, devuelve el hardware al que se han ajustado los parámetros de los canales para continuar realizando mas configuraciones en otros bloques, este terminal está conectado al bloque *Ni Scope configure vertical*.

3.1.3.3 Bloque *Ni Scope configure vertical*.

3.1.3.3.1 Descripción: Este bloque permite configurar los parámetros verticales de la adquisición.

3.1.3.3.2 Terminales de entrada:

- Controles: *Rango vertical* .
- Constantes: *Vertical coupling, Probe attenuation, Channel enabled y Cannel name*. El control *rango vertical* devuelve el valor de rango vertical en mV y el bloque *Configuración vertical* lo necesita en V para realizar este cambio de unidades se utiliza la fórmula *mV->V*, con la ecuación:

$$Valor_{Bloque\ configuracion\ vertical} = \frac{Valor_{Control\ Rango\ vertical}}{1000}$$

- Salidas de otros bloques: *Instrument handle* es la entrada que asigna a los parámetros verticales configurados a la tarjeta de adquisición utilizada, este terminal esta cableado desde el bloque *Ni Scope configure chan characteristics*.

3.1.3.3.3 Terminales de salida:

- Entradas para otros bloques: *Instrument handle out*, devuelve el hardware al que se han ajustado los parámetros verticales para continuar realizando mas configuraciones en otros bloques, este terminal está conectado al *Property node Ni Scope*.

3.1.3.4 *Property node Ni Scope*

3.1.3.4.1 Descripción: Este bloque activa para la adquisición los canales que se han configurado en el bloque *Ni Scope configure chan characteristics*. También permite doblar la frecuencia de muestreo activando el terminal *Time interleaving sampling*, con esta forma de adquisición solo se puede adquirir por un canal (canal 0) porque se utiliza todo el hardware del otro canal (canal 1). Se conecta la señal que se quiere adquirir al canal 0, internamente esta señal se puentea para disponer de ella en los dos canales, el canal 0 adquiere de manera normal y el canal 1 adquiere a la vez que el canal 0 pero con un desfase; $Desfase_{canal\ 1} = \frac{0,5}{Frecuencia_{muestreo}}$ de

PROGRAMA CREADO CON LABVIEW

esta manera se adquiere a una frecuencia de muestreo dos veces mayor que la frecuencia máxima para un canal solo.

3.1.3.4.2 Terminales de entrada:

- Constantes: *Time interleaving sampling*, activa o desactiva la adquisición a una frecuencia dos veces mayor que la frecuencia máxima para un canal solo.
- Salidas de otros bloques: *Active channel*, este terminal esta cableado desde el bloque *Ni Scope configure vertical*. *Instrument handle* es la entrada que asigna a los parámetros ajustados en este *Property node* a la tarjeta de adquisición utilizada, este terminal esta cableado desde el bloque *Ni Scope configure vertical*.

3.1.3.4.3 Terminales de salida:

- Entradas para otros bloques: *Instrument handle out*, devuelve el hardware al que se han ajustado los parámetros de este *Property node* para continuar realizando mas configuraciones en otros bloques, este terminal está conectado al bloque *Ni Scope configure trigger*.

3.1.3.5 Bloque *Ni Scope configure trigger*.

3.1.3.5.1 Descripción: Este bloque permite configurar el disparo de la adquisición. En esta aplicación se utiliza el disparo por flanco (*edge*) debido a los buenos resultados obtenidos durante su utilización.

3.1.3.5.2 Terminales de entrada:

- Constantes: *trigger coupling*, *trigger slope*, *trigger source*, *trigger holdoff* y *trigger delay*.
- *Property node*: El parámetro valor de *trigger* y que es el valor de un control ubicado en otro subcircuito de la aplicación.
- Salidas de otros bloques: *Instrument handle* es la entrada que asigna los parámetros ajustados en este bloque a la tarjeta de adquisición utilizada, este terminal esta cableado desde el *Property node Ni Scope*.

3.1.3.5.3 Terminales de salida:

- Entradas para otros bloques: *Instrument handle out*, devuelve el hardware al que se han ajustado los parámetros del disparo para continuar realizando mas configuraciones en otros bloques, este terminal está conectado al bloque *Ni Scope initiate acquisition*.

PROGRAMA CREADO CON LABVIEW

3.1.3.6 Bloque *Ni Scope initiate acquisition*.

3.1.3.6.1 Descripción: Este bloque inicia la adquisición para el hardware seleccionado mediante el terminal *Instrument handle*.

3.1.3.6.2 Terminales de entrada:

- Salidas de otros bloques: *Instrument handle* es la entrada que asigna la tarjeta de adquisición utilizada para iniciar la adquisición con ella, este terminal está cableado desde el bloque *Ni Scope configure trigger*.

3.1.3.6.3 Terminales de salida:

- Entradas para otros bloques: *Instrument handle out*, devuelve el hardware que ha inicializado la adquisición para continuar realizando más configuraciones en otros bloques, este terminal está conectado al bloque *Ni Scope fetch 1D WDT*.

3.1.3.7 Bloque *Ni Scope fetch 1D WDT*.

3.1.3.7.1 Descripción: Este bloque devuelve las señales adquiridas por cada uno de los dos canales de la tarjeta de adquisición en formato *waveform 1D*.

3.1.3.7.2 Terminales de entrada:

- Constantes: *timestamp type*, *max time* y *channel list*.
- Salidas de otros bloques: *Instrument handle* es la entrada que asigna el hardware desde el cual se van a adquirir los datos, este terminal está cableado desde el bloque *Ni Scope initiate acquisition*.

3.1.3.7.3 Terminales de salida:

- Entradas para otros bloques: *Instrument handle out*, devuelve el hardware con el que se está adquiriendo datos para continuar realizando más operaciones en otros bloques, este terminal está conectado al bloque *Ni Scope close*

3.1.3.8 Bloque *Ni Scope close*

3.1.3.8.1 Descripción: Este subcircuito cierra la sesión de adquisición creada y deja de adquirir datos mediante el bloque *Ni Scope close*, este bloque se ejecuta mediante una estructura del tipo *Case structure* que se activa mediante el control *close t*.

3.1.3.8.2 Terminales de entrada:

- Salidas de otros bloques: *Instrument handle*, asigna el hardware con el que se quiere dejar de adquirir.

3.1.4 Resta de la media en el canal 0

El esquema eléctrico del subcircuito *resta de la media en el canal 0* se muestra en la Figura 3.3.

3.1.4.1 Descripción: Se convierte la señal adquirida en un *array* porque el bloque *media* solo admite datos en su entrada con el formato *array*, una vez se calcula la media de la señal se introduce este parámetro en la fórmula, que devuelve la señal sin valor medio. Ecuación ejecutada en el bloque *fórmula*:

$$\text{Resultado} = \text{Entrada} - \text{Media}$$

El motivo de restar la media se debe a que en la integral posterior se cometía mucho error, partiendo de que la señal que se quiere integrar tiene un valor medio a se puede concluir que la señal de entrada era de la forma $F(x) + a$, al integrar esta señal quedaría una señal de la forma $G(x) + (ax)$, es decir, que la señal de salida que nos interesa esta sumada a una rampa de valor a .

3.1.4.2 Terminales de entrada:

- Salidas de otros bloques: *Entrada*, es el terminal donde se conecta la salida del bloque *Ni Scope fetch 1D WDT* (Canal 0) que devuelve la señal adquirida por el canal 0.

3.1.4.3 Terminales de salida:

- Entradas para otros bloques: *Resultado*, este terminal devuelve el resultado de la ecuación una vez se ha ejecutado la fórmula, este terminal se conecta con la entrada del subcircuito *filtro canal 0*.

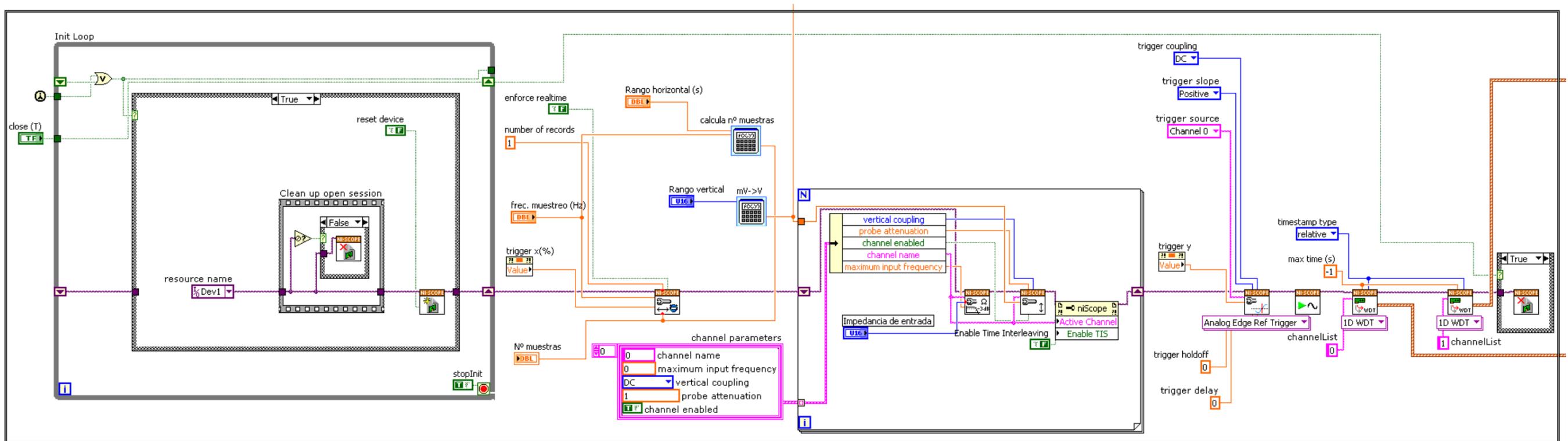


Figura 3.2: Conexión del diagrama de bloques para la adquisición.

3.1.5 Resta de la media en el canal 1

El esquema eléctrico del subcircuito *resta de la media en el canal 1* se muestra en la Figura 3.4.

3.1.5.1 Descripción: Utiliza el mismo método de cálculo que en el subcircuito *resta de la media en el canal 0*. En este caso el motivo de restar la media es que en este canal se conectan sondas que integran vía hardware y tienen *offset*, interesa eliminarlo para poder comparar este canal, canal 1, con el canal 0, que adquiere una señal integrada vía software.

3.1.5.2 Terminales de entrada:

- Salidas de otros bloques: *Entrada*, es el terminal donde se conecta la salida del bloque *Ni Scope fetch 1D WDT (Canal 1)* que devuelve la señal adquirida por el canal 1.

3.1.5.3 Terminales de salida:

- Entradas para otros bloques: *Resultado*, este terminal devuelve el resultado de la ecuación una vez se ha ejecutado la fórmula, este terminal se conecta con la entrada del subcircuito *filtro canal 1*.

3.1.6 Filtro canal 0 y canal 1

El esquema eléctrico del subcircuito *Filtro canal 0 y canal 1* se muestra en la Figura 3.5.

3.1.6.1 Descripción: En estos subcircuitos se eliminan las frecuencias seleccionadas por el usuario utilizando el bloque *filter*. En esta aplicación se pretende eliminar el ruido quitando las altas frecuencias. Los dos filtros están conectados en paralelo, el filtro del canal 0 tiene conectados los controles y el filtro del canal 1 tiene conectados los *Property node* asociados a los valores de los controles del filtro del canal 0, se optó por esta forma de conexión por cuestión de limpieza y claridad que suponía no cablear un número tan grande de terminales.

3.1.6.2 Terminales de entrada:

- Controles: *Taps, filter topology, filter type, lower cut-off, order y upper cut-off*.
- Constantes: *IIR/FIR y Output cluster*.
- Salidas de otros bloques: *Entrada*, es el terminal donde se conecta la salida del bloque *fórmula Resta la media* para el canal 0 y del bloque *fórmula offset=0* para el canal 1.

PROGRAMA CREADO CON LABVIEW

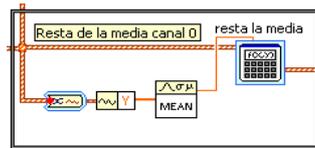


Figura 3.3: Conexionado del diagrama de bloques para la resta de la media en el canal 0.

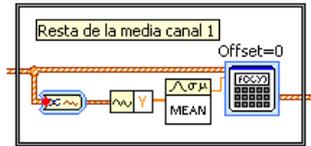


Figura 3.4: Conexionado del diagrama de bloques para la resta de la media en el canal 1.

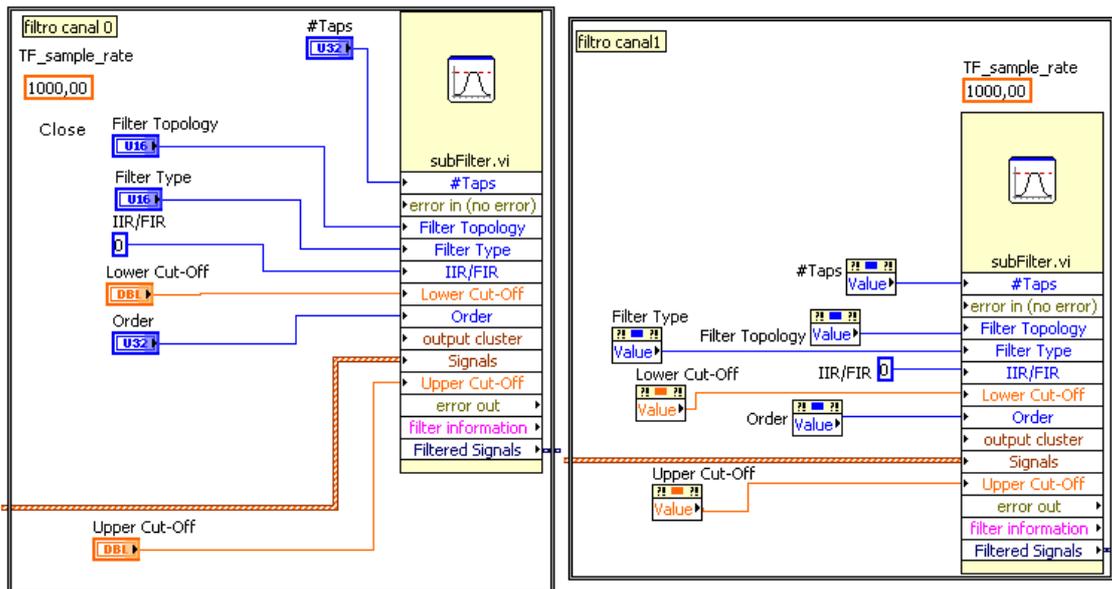


Figura 3.5: Conexionado del diagrama de bloques de los filtros.

3.1.6.3 Terminales de salida:

- Entradas para otros bloques: *Resultado*, este terminal devuelve el resultado del filtro y se conecta con la entrada del subcircuito *Integración tensión* → *intensidad* para el canal 0 y *Ganancia canal 1* para canal 1.

3.1.7 Ganancia en el canal 1

El esquema eléctrico del subcircuito *Ganancia en el canal 1* se muestra en la Figura 3.6.

3.1.7.1 Descripción: Este subcircuito multiplica el valor de la *Ganancia canal 1* por la señal de entrada mediante una fórmula, la ecuación del bloque *fórmula* es:

$$\text{Resultado} = \text{Entrada} \times \text{Ganancia canal 1}$$

PROGRAMA CREADO CON LABVIEW

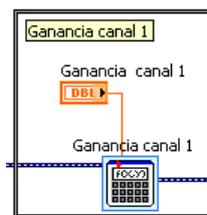


Figura 3.6: Conexionado del diagrama de bloques para aplicar ganancia al canal 1.

El motivo de poner este subcircuito es poder comparar los dos canales. Tomando por referencia la magnitud del canal 0, se multiplica al canal 1 por la ganancia que se controla desde la parte superior derecha del grafico *comparación integral software Vs integral hardware con el control ganancia canal 1*.

3.1.7.2 Terminales de entrada:

- Controles: *Ganancia canal 1*.
- Salidas de otros bloques: *Entrada*, es el terminal donde se conecta la salida del subcircuito *filtro canal 1*.

3.1.7.3 Terminales de salida:

- Entradas para otros bloques: *Resultado*, este terminal devuelve el resultado de la fórmula *Ganancia canal 1*, este terminal está conectado al grafico *comparación integral software Vs integral hardware*.

3.1.8 Integración tensión → intensidad

El esquema eléctrico del subcircuito *Integración tensión → intensidad* se muestra en la Figura 3.7.

3.1.8.1 Descripción: En este subcircuito se transforma la señal del formato *waveform* al formato *array* porque el bloque *integrador* solo admite los datos de entrada en este formato, integra la señal de entrada y se divide entre la inductancia mutua que existe entre la espira y el conductor por el que discurren las descargas parciales. Tanto la integración como la fórmula que hay colocada posteriormente sirven para contrarrestar los efectos de la sonda de medida.

La señal a la salida de la sonda es $u(t) = \frac{di(t)}{dt} \cdot m$, se quiere conocer la magnitud real de la descarga parcial, para ello se integra la ecuación original y se divide entre la inductancia $i(t) = \frac{\int u(t) \cdot dt}{m}$.

PROGRAMA CREADO CON LABVIEW

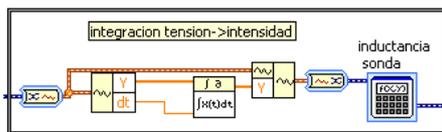


Figura 3.7: Conexionado del diagrama de bloques para la integración tensión → intensidad.

El motivo de integrar la señal de entrada es porque la sonda que se utiliza para la medición de las descargas parciales es una espira contigua al cable por el que discurren las descargas parciales y la señal que devuelve la sonda es la tensión que se induce en bornes de la espira que se corresponde a la derivada de la intensidad de la descarga parcial afectada por un coeficiente que es la inductancia entre el cable y la espira. Por tanto para tener una señal parecida a la señal original de la descarga parcial se debe integrar la señal adquirida. Cuando se realiza esta integral se pasa de tener una tensión a tener una intensidad.

La fórmula que hay colocada después de la integral divide la señal de salida del bloque *integrador* entre 54,9nH (valor de la inductancia mutua de la espira), de esta manera se obtiene una señal igual a la señal de la descarga parcial. Ecuación de la fórmula:

$$\text{Resultado} = \text{Entrada} / 54,9^{-9}$$

3.1.8.2 Terminales de entrada:

- Salidas de otros bloques: *Entrada*, es el terminal donde se conecta la salida del subcircuito *filtro canal 0*.

3.1.8.3 Terminales de salida:

- Entradas para otros bloques: *Resultado*, este terminal devuelve el resultado de la fórmula *inductancia sonda*, este terminal está conectado al subcircuito *recortar señal*.

3.1.9 Recortar señal

El esquema eléctrico del subcircuito *recortar señal* se muestra en la Figura 3.8.

- 3.1.9.1 Descripción: Este subcircuito recorta el principio de la señal. Interesa recortar el ruido que existe antes de producirse el disparo porque a la hora de calcular la carga realizando una nueva integral se genera un gran error, sobre todo cuando el disparo horizontal es elevado.

PROGRAMA CREADO CON LABVIEW

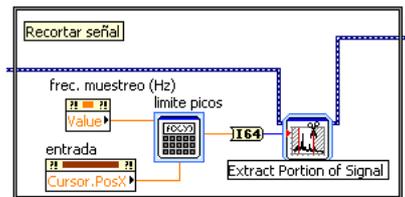


Figura 3.8: Conexionado del diagrama de bloques para recortar la señal.

Para poder eliminar el ruido existente antes de la señal se utiliza el bloque *extract portion of signal*. Este bloque se configura para recortar solo el inicio de la señal, el bloque pide la muestra hasta donde se quiere eliminar la señal, a partir de esa muestra deja pasar la señal. Con el fin de automatizar la aplicación se calcula la muestra en que se produce el disparo horizontal y ese valor se introduce en el bloque *recortador*. En el momento que se produce una modificación en el disparo horizontal automáticamente se modifica la porción que se recorta a la señal. Para calcular la muestra donde se produce el disparo horizontal se utilizan dos *Property node*, *frecuencia de muestreo* [Hz-muestra] y *value trigger x* [s], realizando el producto entre ellos se obtiene la muestra donde se produce el disparo horizontal. Se introducen estos dos parámetros en la fórmula *límite de picos*, la ecuación del bloque fórmula es:

$$Muestra_{disparo\ horizontal} = Frecuencia\ de\ muestreo \times Value\ trigger\ x$$

3.1.9.2 Terminales de entrada:

- *Property node: Frecuencia de muestreo y value trigger x.*
- Salidas de otros bloques: *Entrada*, es el terminal *signal* del bloque *extract portion of signal*, se conecta con la salida del subcircuito *Integración tensión* → *intensidad*.

3.1.9.3 Terminales de salida:

- Entradas para otros bloques: *Resultado*, este terminal devuelve la señal recortada y está conectado al subcircuito *Integración intensidad* → *carga*.

3.1.10 Integración intensidad → carga

El esquema eléctrico del subcircuito *Integración intensidad* → *carga* se muestra en la Figura 3.9.

3.1.10.1 Descripción: En este subcircuito se integra la señal recortada para obtener la carga, para ello previamente se cambia al formato *array* a la señal, después de integrar se vuelve cambiar el formato de la señal al formato *Dinamic data*, para tener unidos los datos con el instante en que se adquieren.

PROGRAMA CREADO CON LABVIEW

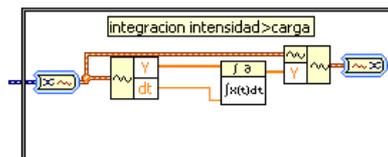


Figura 3.9: Conexionado del diagrama de bloques para la integración intensidad → carga

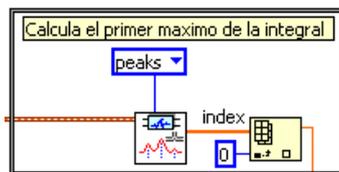


Figura 3.10: Conexionado del diagrama de bloques para el cálculo del primer pico de la señal integrada.

Anteriormente se trató la señal adquirida para obtener una señal igual que las descargas parciales, ahora se quiere calcular a partir de esta última señal la carga de la descarga parcial, para ello se parte de la siguiente ecuación $i(t) = \frac{dQ(t)}{dt}$, operando e integrando queda de la siguiente manera $Q(t) = \int i(t) \cdot dt$, que es el valor de la carga de la descarga parcial.

La carga es un valor numérico no una señal, si la señal fuera perfecta (no tuviera ruido), la integral numérica daría un resultado con poco error pero no es así, existe mucho ruido que perturba la carga. Para evitar este error se utiliza la integral discreta en lugar de la numérica, utilizando el primer pico de esta señal como referencia se realiza una regla de tres, para ello se debe calibrar (memorizar un valor para una carga conocida) y posteriormente con el valor de pico que se obtenga se puede estimar la carga:

$$Q_{instantanea} = \frac{Q_{referencia}}{Area_{calibracion}} \cdot Area_{instantanea}$$

3.1.10.2 Terminales de entrada:

- Salidas de otros bloques: *Entrada*, es el terminal del *convertor dinamic data to waveform* y se conecta la salida del subcircuito *recortar señal*.

3.1.10.3 Terminales de salida:

- Entradas para otros bloques: *Resultado*, este terminal devuelve la señal *carga* y está conectado al subcircuito *calcula el primer máximo*.

3.1.11 Calcula el primer máximo de la integral

El esquema eléctrico del subcircuito *calcula el primer máximo de la integral* se muestra en la Figura 3.10.

PROGRAMA CREADO CON LABVIEW

3.1.11.1 Descripción: En este subcircuito se obtiene el valor que se va a utilizar para calibrar y posteriormente para calcular la carga, para ello se introduce la integral discreta de la intensidad (la carga) en el bloque *detector de picos*. El detector de picos devuelve un *array* con todos los picos que ha detectado colocados en orden según se producen, como se quiere obtener el primer máximo de la carga, se ha colocado el bloque *localizador de elementos en un array* que se configura para que devuelva el valor del elemento que ocupa la posición 0.

3.1.11.2 Terminales de entrada:

- Constantes: *Index*.
- Salidas de otros bloques: Entrada, es el terminal del bloque *detector de picos* y se conecta la salida del subcircuito *Integración intensidad* → *carga*.

3.1.11.3 Terminales de salida:

- Entradas para otros bloques: *Resultado*, este terminal devuelve el valor del primer pico de la señal carga y está conectado al subcircuito *Automatización de la calibración*.

3.1.12 Automatización de la calibración

El esquema eléctrico del subcircuito *Automatización de la calibración* se muestra en la Figura 3.11.

3.1.12.1 Descripción: En este subcircuito se calibra la aplicación y se transforma el primer valor de pico de la señal carga en un valor de carga en pC.

Para calibrar se generan impulsos con un calibrador conectado a la sonda. El calibrador tiene una carga regulable. En el control *carga referencia* se debe colocar el valor de la carga que tienen los impulsos que genera el calibrador, este paso solo se puede ejecutar mientras el interruptor *ajustar carga / calibrar* → (*false / true*) está en la posición *ajustar carga (false)*, en la otra posición (*true*) el control *carga referencia* queda bloqueado y sombreado para impedir modificaciones en *la carga de referencia* durante la calibración, estas modificaciones o la variación de la carga del calibrador durante la calibración anularían la calibración y por tanto se debería repetir. Una vez ajustada la *carga de referencia* se conmuta el interruptor a la posición *calibrar*, entonces se comienza a ejecutar un bucle en el que hay un sumatorio, es decir, que se comienza a sumar los valores del primer pico de la señal carga, cuando la aplicación se ha ejecutado cuarenta veces la calibración se da por concluida. Se sabe que se ha ejecutado cuarenta veces porque paralelamente existe otro bucle que suma la unidad cada vez que se ejecuta la aplicación, por tanto este sumatorio indica el número de ejecuciones de la aplicación.

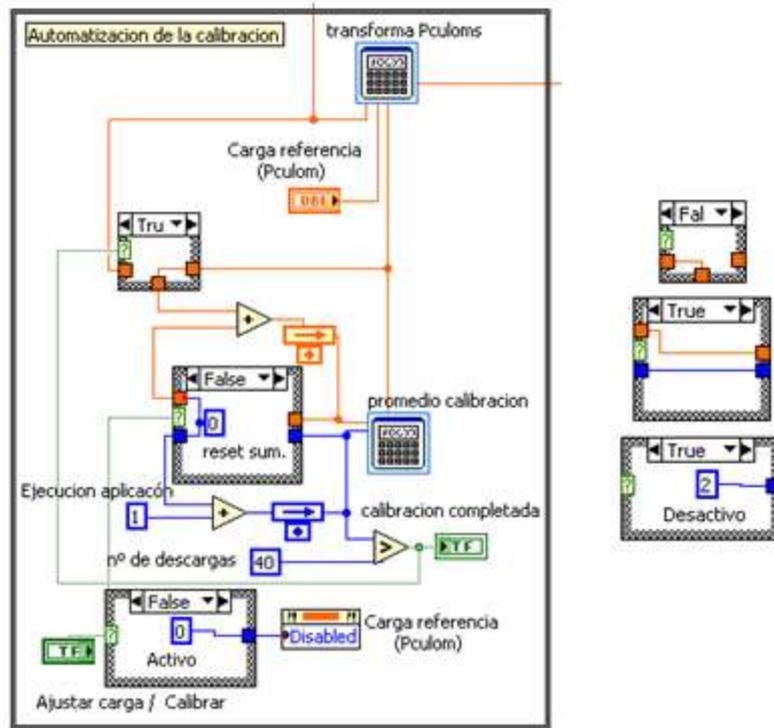


Figura 3.11: Conexionado del diagrama de bloques para la calibración.

Con estos dos sumatorios se calcula el promedio del valor del primer pico de la señal carga, mediante la fórmula *promedio calibración* que divide el sumatorio del valor del primer pico de la señal carga entre el sumatorio de ejecuciones. Este promedio se usara como proporción para calcular la *carga instantánea* en pC. Ecuación de la fórmula *promedio calibración*:

$$Valor\ de\ pico_{promediado} = \frac{\sum Valor\ del\ primer\ pico}{\sum Ejecuciones}$$

Hay un comparador que devuelve *true* cuando la aplicación se ha ejecutado más de cuarenta veces, hasta entonces el valor de este comparador es *false*. Una vez concluida la calibración se enciende el *led* calibración completada que indica que la calibración se ha completado. Paralelamente se cierra otro bucle, el sumatorio de los valores del primer pico de la señal carga deja de adquirir nuevos datos y comienza a sumar el valor promediado, de esta manera se suma *n* veces el mismo valor mientras que el sumatorio de las ejecuciones marca *n* ejecuciones.

Para *resetear* el valor promedio calibración se debe colocar el interruptor en la posición *ajustar carga*, de esta manera se abren los bucles y los sumatorios se hacen cero, por tanto en la siguiente ejecución el primer sumatorio es el valor instantáneo y el sumatorio de ejecuciones es la unidad que implica que el valor promedio es simplemente el valor instantáneo.

La fórmula *transforma pC* se encarga de devolver el valor de la carga de la descarga parcial en pC.

PROGRAMA CREADO CON LABVIEW

$$Q_{instantanea} [pC] = \frac{Q_{referencia} [pC]}{Valor\ de\ pico_{promediado}} \cdot Valor\ de\ pico_{instantaneo}$$

3.1.12.2 Terminales de entrada:

- Control: *Carga de referencia* (pC).
- Constantes: *nº de descargas, ejecución aplicación, activo, desactivo y reset sum.*
- Interruptor: *Ajustar carga / Calibrar*
- Salidas de otros bloques: *Entrada*, es el terminal que devuelve el valor del primer máximo de la señal *carga* y se conecta la salida del subcircuito *Calcula el primer máximo de la integral.*

3.1.12.3 Terminales de salida:

- *Indicador luminoso: Calibración completada.*
- *Property node: Carga de referencia (Disabled).*
- Entradas para otros bloques: *Resultado*, este terminal devuelve el valor de la carga transformado a pC y está conectado a los subcircuitos *Muestra carga instantánea, mínima y máxima y Muestra carga promediada.*

3.1.13 Indicadores de carga máxima, mínima e instantánea

El esquema eléctrico del subcircuito *indicadores de carga máxima, mínima e instantánea* se muestra en la Figura 3.12.

3.1.13.1 Descripción: En este subcircuito se calcula el valor máximo y mínimo mediante bucles para mostrarlos en indicadores junto al valor instantáneo.

Partiendo del valor instantáneo de carga en pC se realizan dos bucles, uno para el valor de carga máxima y el otro para el valor de carga mínima. Se genera con el bloque *creador de array* un *array* con de tres elementos *carga máxima, mínima e instantánea*. Mediante el bloque *buscador de máximo y mínimo* se obtiene el máximo y el mínimo del *array* de manera que si el valor instantáneo es superior que el valor máximo o inferior al valor mínimo estos valores se modifican por los nuevos valores detectados. Desde las salidas del bloque *buscador de máximo y mínimo* se cablea hasta el bloque *creador de array* en los terminales correspondientes a carga máxima y carga mínima, en medio de esta conexión hay colocado una estructura del tipo *case structure* donde se *resetean* estos valores si se pulsa el pulsador *reset* o el pulsador *reset carga*. Cuando se pulsa uno de estos pulsadores lo que se hace es igualar a cero los valores máximo y mínimo.

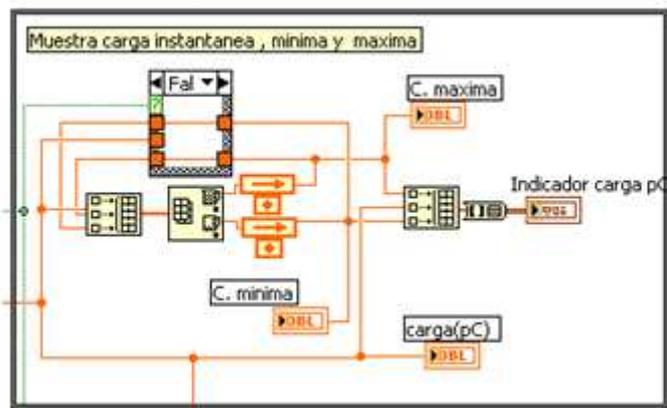


Figura 3.12: Conexionado del diagrama de bloques para el cálculo de máximo, mínimo y valor actual de la carga en pC.

Desde las salidas del bloque *buscador de máximo y mínimo* también salen otras conexiones una va hacia el indicador *C. máxima* y otra hacia *C. mínima*, el indicador *Carga (pC)* se conecta con el terminal de carga instantánea que hay en el bloque *creador de array*. Con estos tres valores se genera un *array* y a continuación un *cluster* para poder graficar estos tres elementos en el indicador de aguja *Indicador carga pC*. En este indicador hay tres agujas, la azul marca el mínimo, la negra el valor instantáneo y la roja el valor máximo.

3.1.13.2 Terminales de entrada:

- Salidas de otros bloques: *Entrada*, es el *resultado* de la fórmula transformada en pC del subcircuito *Automatización de la calibración* que devuelve el valor instantáneo de la carga en pC. *Condición case structure*, es el valor booleano que devuelve la puerta lógica *or* que tiene conectada a sus entradas los pulsadores *reset* y el *reset carga*.

3.1.13.3 Terminales de salida:

- *Indicadores numéricos: C. máxima, C. mínima y carga (pC).*
- *Indicador aguja: Indicador carga pC*
- Entradas para otros bloques: La entrada a este subcircuito es a su vez la entrada del subcircuito *Indicadores de carga promediada*.

3.1.14 Parámetros del indicador de intensidad

El esquema eléctrico del subcircuito *parámetros del indicador de intensidad* se muestra en la Figura 3.13.

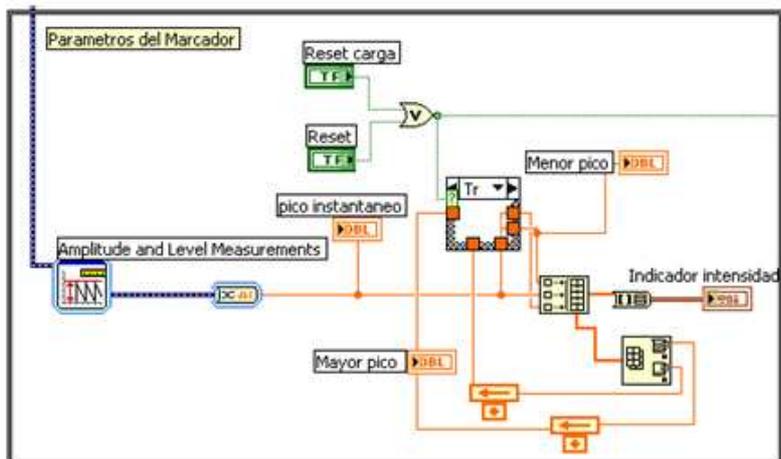


Figura 3.13: Conexionado del diagrama de bloques para el cálculo de máximo y mínimo pico de intensidad.

3.1.14.1 Descripción: En este subcircuito se calcula el mayor pico de intensidad, el menor pico de intensidad y el pico instantáneo. Estos tres valores se muestran en indicadores numéricos por separado y juntos en un indicador de aguja. En este subcircuito se ubican los pulsadores *reset* y *reset carga*.

Este subcircuito realiza las mismas funciones que el subcircuito *indicadores de carga máxima, mínima e instantánea* que se muestra en la Figura 3.12. La entrada a este subcircuito es la señal del canal 1 después de integrarla y aplicarla la inductancia mutua. Se calcula el pico positivo de esta señal y realiza el mismo proceso que en el subcircuito *indicadores de carga máxima, mínima e instantánea*. En este apartado los indicadores se llaman *menor pico*, *mayor pico* y *pico instantáneo*. El indicador de aguja se llama *indicador de intensidad*.

3.1.14.2 Terminales de entrada:

- Pulsadores: *Reset* y *carga reset*.
- Salidas de otros bloques: *Entrada*, es el terminal *signal* del bloque *amplitude and level measurements*, se conecta con la salida del subcircuito *Integración tensión* → *intensidad*.

3.1.14.3 Terminales de salida:

- *Indicadores numéricos*: *Menor pico*, *Mayor pico* y *Pico instantáneo*.
- *Indicador aguja*: *Indicador intensidad*
- Entradas para otros bloques: Desde este subcircuito sale un cable en formato booleano que devuelve la puerta lógica *or*, esta puerta tiene conectada a sus entradas los pulsadores *reset* y el *reset carga*.

PROGRAMA CREADO CON LABVIEW

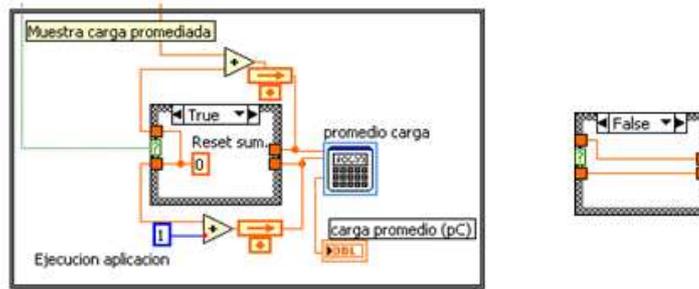


Figura 3.14: Conexionado del diagrama de bloques para el cálculo de la carga promediada.

3.1.15 Indicadores de carga promediada

El esquema eléctrico del subcircuito *indicadores de carga promediada* se muestra en la Figura 3.14.

3.1.15.1 Descripción: En este subcircuito se calcula el valor de carga promediado mediante bucles para mostrarlo en el indicador *carga promedio (pC)*.

La carga promediada es un valor orientativo del orden de magnitud de las descargas parciales que se están produciendo en el material de ensayo.

En este subcircuito se calcula el valor promedio de la carga en pC de la misma manera que se calcula el promedio de la calibración, se hacen dos sumatorios (como en el apartado Automatización de la calibración) uno con el valor de carga instantánea y otro con el número de veces que se ha ejecutado la aplicación. En el sumatorio de carga instantánea se suman tantos valores de carga instantánea como veces se ejecuta la aplicación, por tanto cuando se introducen estos dos sumatorios en la fórmula *promedio carga* y se divide el sumatorio de carga instantánea entre el sumatorio de ejecuciones se obtiene el valor promedio de la carga. Ecuación de la fórmula *promedio carga*:

$$Carga_{promedio} = \frac{\sum Carga\ instantanea}{\sum Ejecuciones}$$

Para *resetear* el valor promedio de la carga en pC se debe pulsar el pulsador *reset* o *reset carga*. Los bucles se abren mediante una estructura del tipo *case structure* que tiene conectada a su terminal selector un bloque *puerta lógica or* cuyas entradas tienen conectadas el pulsador *reset* y el *reset carga*. Cuando se abren los bucles los sumatorios se hacen cero de manera que en la siguiente ejecución el primer sumatorio es el valor instantáneo y el sumatorio de ejecuciones es la unidad y por tanto el valor promedio es simplemente el valor instantáneo.

El resultado de la fórmula que es un valor numérico se muestra en el indicador *carga promedio (pC)*.

PROGRAMA CREADO CON LABVIEW

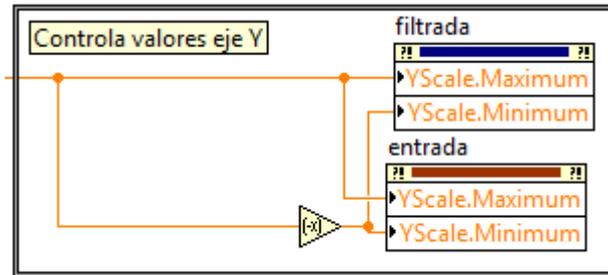


Figura 3.15: Conexionado del diagrama da bloques para controlar la escala vertical de los gráficos entrada y filtrada.

3.1.15.2 Terminales de entrada:

- Constantes: *ejecución aplicación* y *reset sum*.
- Salidas de otros bloques: *Entrada*, es el *resultado* de la fórmula transforma en pC del subcircuito *Automatización de la calibración* que devuelve el valor instantáneo de la carga en pC. *Condición case structure*, es el valor booleano que devuelve la puerta lógica *or* que tiene conectada a sus entradas los pulsadores *reset* y el *reset carga*.

3.1.15.3 Terminales de salida:

- *Indicador numérico: Carga promedio (pC)*.

3.1.16 Controla la escala vertical de los gráficos filtrada y entrada

El esquema eléctrico del subcircuito *Controla la escala vertical de los gráficos filtrada y entrada* se muestra en la Figura 3.15.

3.1.16.1 Descripción: En este subcircuito se automatiza el control del valor de la escala de los gráficos entrada y filtrada asociándolos al control *rango vertical*.

La entrada a este subcircuito es la salida de la fórmula $mv \rightarrow V$, que transforma de mili voltios a voltios la salida del control *rango vertical*, se toma este valor como escala de los gráficos, este valor limita la adquisición y no se pueden adquirir señales con una tensión superior al valor del control *rango vertical* porque se satura la tarjeta y deforma la señal adquirida. Se desea tener una escala que grafique entre valores opuestos, para ello se utiliza el bloque *negador*. El valor de escala máxima se obtiene de la salida de la fórmula $mv \rightarrow V$, la salida del bloque *negador* nos devolverá el valor de la escala mínima.

Una vez se tienen los valores máximos y mínimos de la escala se utilizan los cuatro *Property node*. El valor de la escala máxima se cablea a los *Property node Yscale maximun* de los *graficos entrada y filtrada*, el valor de la escala mínima se cablea a los *Property node Yscale minimun* de los *graficos entrada y filtrada*.

PROGRAMA CREADO CON LABVIEW

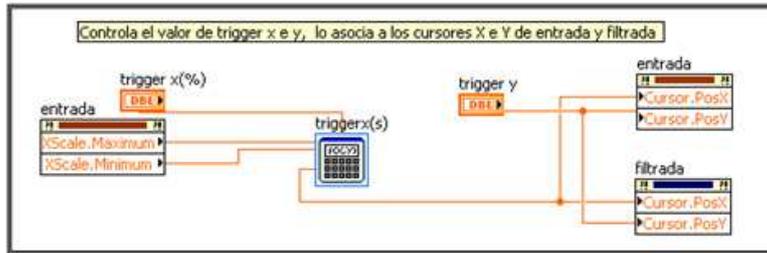


Figura 3.16: Conexionado del diagrama de bloques que controla el *trigger* y los cursores de los gráficos entrada y filtrada.

3.1.16.2 Terminales de entrada:

- Salidas de otros bloques: *Entrada*, es el resultado de la fórmula $mv \rightarrow V$ del subcircuito *Adquisicion*.

3.1.16.3 Terminales de salida:

- *Property node*: *Yscale maximun*, *Yscale minimun* de los gráficos *entrada* y *filtrada*.

3.1.17 Controla el valor de *trigger* y lo asocia a los cursores X e Y de los gráficos entrada y filtrada

El esquema eléctrico del subcircuito *Controla el valor de trigger y lo asocia a los cursores X e Y de los gráficos entrada y filtrada* se muestra en la Figura 3.16.

3.1.17.1 Descripción: En este subcircuito se automatiza la posición de los cursores de los gráficos *entrada* y *filtrada* asociándolos a los valores de los controles del disparo horizontal y vertical.

Para controlar la posición de los cursores X e Y se utiliza dos *Property node* que devuelven el valor máximo y mínimo de la escala horizontal del grafico *entrada* en segundos, el valor mínimo es siempre cero y el máximo depende del control *rango horizontal (s)*. El control *trigger x (%)* está configurado para poder variar entre 0% y 100% con un incremento de la unidad. El valor del cursor x se debe introducir en segundos, para hacer la conversión de porcentaje a segundos se utiliza la fórmula *Trigger x (s)* tiene tres entradas los dos *Property node* descritos en este párrafo y el control *trigger x (%)*, la ecuación que se ejecuta en esta fórmula es :

$$Trigger\ x\ [s] = \frac{Trigger\ x[\%] \cdot (X\ scale\ max - X\ scale\ min)}{100}$$

Para controlar la posición de los cursores Y se utiliza el control *trigger y*.

Una vez se tienen los valores de *trigger* en segundos se relacionan con los *Property node* correspondientes. El resultado de la fórmula se cablea directamente con los *Property node cursor pos. X* de los gráficos *entrada* y *filtrada*. La salida del control

PROGRAMA CREADO CON LABVIEW

trigger y se cablea directamente con los *Property node cursor pos*. Y de los gráficos *entrada* y *filtrada*.

3.1.17.2 Terminales de entrada:

- Controles deslizador: *trigger x (%)* y *trigger*.
- *Property node: X scale maximum* y *X scale minimum* del gráfico *entrada*.

3.1.17.3 Terminales de salida:

- *Property node: Cursor pos X* y *Cursor pos Y* de los gráficos *entrada* y *filtrada*.

3.1.18 Controla la escala de *trigger Y*

El esquema eléctrico del subcircuito *Controla la escala de trigger Y* se muestra en la Figura 3.17.

3.1.18.1 Descripción: En este subcircuito se define la escala del control deslizador *trigger y*. Para ello se utilizan cuatro *Property node*, dos para llamar a la escala máxima y mínima del gráfico *entrada* y otros dos para introducir el valor máximo y mínimo del control deslizador *trigger y*, se cablean correspondientemente la escala máxima y mínima del gráfico con la escala máxima y mínima del control *trigger y*.

De esta manera como el control y el gráfico tienen la misma dimensión, si se varía el valor del control *trigger y* se varía el valor del cursor en la misma proporción coincidiendo en la misma posición el indicador del control con el cursor.

3.1.18.2 Terminales de entrada:

- *Property node: Y scale maximum* y *Y scale minimum* del gráfico *entrada*.

3.1.18.3 Terminales de salida:

- *Property node: Scale maximum* y *Scale minimum* del control *trigger y*.

3.1.19 Cálculo del valor máximo y mínimo absoluto de la integral hardware (canal 0) y la integral software (canal 1)

El esquema eléctrico del subcircuito *Cálculo del valor máximo y mínimo absoluto de la integral hardware (canal 0) y la integral software (canal 1)* se muestra en la Figura 3.18.

PROGRAMA CREADO CON LABVIEW

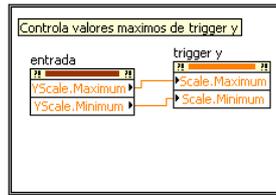


Figura 3.17: Conexionado del diagrama de bloques para el control del *trigger*.

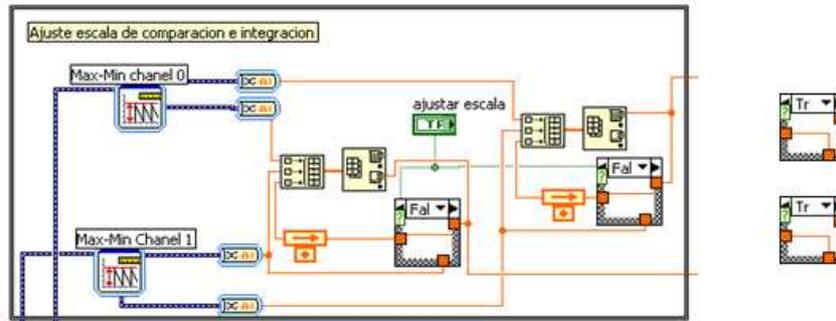


Figura 3.18: Conexionado del diagrama de bloques para el Cálculo de escala del gráfico comparación integral software Vs hardware.

3.1.19.1 Descripción: En este subcircuito se utilizan dos bloques que nos permiten localizar el máximo y mínimo de una señal, el bloque *Max-min channel 0* extrae el máximo y el mínimo de la integral hardware (canal 0) y el bloque *Max-min channel 1* extrae el máximo y el mínimo de la integral software (canal 1). El bloque devuelve estos valores en formato *dynamic data type*, se transforman al formato. Posteriormente el subcircuito se divide en dos partes cableadas de la misma forma pero que operan con distintos valores, una que trabaja con los máximos y otra que trabaja con los mínimos. Se explicara una de las dos partes, la de los valores máximos, la de los mínimos es análoga.

Se genera un *array* mediante el bloque *creador de array* con tres elementos, uno será el máximo de la integral hardware otro máximo de la integral software y el ultimo será el máximo absoluto. Una vez que se dispone del *array* se pasa por el bloque *localizador del máximo y el mínimo*, en este caso el máximo. Se cablea la salida *max value* con el terminal *máximo absoluto* del bloque *creador de array*, de esta manera se consigue memorizar siempre el máximo absoluto. Si se desea *resetear* este valor, existe una estructura del tipo *case structure* controlada por el pulsador *ajustar escala*. El bucle atraviesa esta estructura, cuando el pulsador *ajustar escala* esta en reposo (*false*) el bucle queda cerrado, cuando el pulsador *ajustar escala* esta excitado (*true*) el bucle se abre y asigna al valor máximo absoluto el valor máximo instantáneo del canal 1.

El terminal *max value* del bloque *localizador del máximo y el mínimo* de la señal de entrada más adelante se utilizara en el subcircuito *Ajuste de los límites de los indicadores de aguja y el gráfico comparación integral hardware Vs software*.

PROGRAMA CREADO CON LABVIEW

3.1.19.2 Terminales de entrada:

- Pulsador: *Ajustar escala.*
- Salidas de otros bloques: Al bloque *Max-min canal 0* se conecta la salida de la fórmula *inductancia sonda* del subcircuito *Integral Tensión* → *Intensidad* y Al bloque *Max-min canal 1* se conecta la salida de la fórmula *Ganancia canal 1* del subcircuito *Ganancia canal 1*.

3.1.19.3 Terminales de salida:

- Entradas para otros bloques: Las salidas de este subcircuito son el máximo y mínimo absoluto alcanzado en las señales de entrada, estos dos valores se cablean con el subcircuito *Ajuste de los límites de los indicadores de aguja y el grafico comparación integral hardware Vs software.*

3.1.20 Ajuste de los límites de los indicadores de aguja y el grafico comparación integral hardware Vs software

El esquema eléctrico del subcircuito *Ajuste de los límites de los indicadores de aguja y el grafico comparación integral hardware Vs software* se muestra en la Figura 3.19.

3.1.20.1 Descripción: En este subcircuito se automatiza la escala de los indicadores de aguja y del grafico *comparación integral hardware Vs software.*

Se utiliza el máximo absoluto cableado desde el subcircuito *Calculo del valor máximo y mínimo absoluto de la integral hardware (canal 0) y la integral software (canal 1)* para ajustar la escala máxima del indicador de aguja *indicador intensidad* y del grafico *comparación integral software Vs integral hardware*, para que las señales no queden fuera de escala, se multiplica el máximo absoluto por 1,1 en la fórmula *limite intensidad*. Posteriormente se introduce ese valor en el *Property node scale maximun* del *indicador de intensidad*, de esta manera la escala máxima siempre estará un 10% por encima del valor máximo. El *Property node scale minimun* del *indicador de intensidad*, se cablea un cero.

Para fijar la escala máxima del grafico *comparación integral software Vs integral hardware* se utiliza el resultado de la fórmula *Limite de intensidad*, se conecta su salida con el *Property node scale maximun* del grafico. La escala mínima del grafico se obtiene de la fórmula *limite inferior*, esta fórmula multiplica por 1,1 a la *entrada que es el mínimo absoluto de la integral hardware (canal 0) y la integral software (canal 1)*, la salida se introduce en el *Property node scale minimun* del grafico.

La escala del indicador *Indicador carga pC* se configura a partir del *Property node value* del indicador *C.maxima*. Cuando no se detecta señal no se detecta el primer pico de la señal carga, esto implica que no existe un valor de pico para poder calibrar y el promedio del valor de pico es cero, por tanto la carga instantánea es

PROGRAMA CREADO CON LABVIEW

infinito de manera que el valor de carga máxima pasa a ser infinito. El indicador numérico si puede graficar infinito pero la escala máxima de un grafico no puede tomar este valor. Para corregir este error se colocó una estructura del tipo *case structure* gobernada por la condición *C. máxima=infinito*. Si esta condición se cumple, el valor la escala máxima pasa a ser cero. Si esta condición no se cumple, es decir, que todo funciona bien la estructura simplemente ejecuta la fórmula *límite carga* que multiplica por 1,1 a su entrada que es *C. máxima*. La salida de esta fórmula se introduce en el *Property node scale maximun* del indicador *Indicador carga pC*. Al *Property node scale minimun* del *Indicador carga pC*, se cablea un cero.

Las ecuaciones de las tres fórmulas anteriores *límite intensidad*, *límite inferior* y *límite carga es la misma*:

$$\text{Resultado} = \text{Entrada} \times 1,1$$

3.1.20.2 Terminales de entrada:

- Constantes: *Mínimo indicadores, infinito y reset máxima*.
- *Property node: C. maxima*.
- Salidas de otros bloques: Hay dos entradas provenientes del subcircuito *Calculo del valor máximo y mínimo absoluto de la integral hardware (canal 0)* y *la integral software (canal 1)*, una es el valor máximo absoluto alcanzado entre la integral hardware (canal 0) y la integral software (canal 1) y la otra es el valor mínimo absoluto entre esas dos señales.

3.1.20.3 Terminales de salida:

- *Property node: Scale maximun, scale minimun* de los indicadores de aguja *Indicador intensidad e Indicador carga pC*. Y *scale maximun, Y scale minimun del grafico comparación integral software Vs integral hardware*.

3.1.21 Plantillas

El esquema eléctrico del subcircuito *Plantillas* se muestra en la Figura 3.20.

3.1.21.1 Descripción: Este subcircuito no dispone de ningún tipo de cableado, simplemente se han generado dos plantillas donde hay distintas páginas. La configuración de estas páginas se realiza en el *Front panel*.

3.1.22 Gráficos

El esquema eléctrico del subcircuito *Gráficos* se muestra en la Figura 3.21.

3.1.22.1 Descripción: En estos subcircuitos se cablea directamente la señal que se quiere graficar al bloque *indicador grafico*.

PROGRAMA CREADO CON LABVIEW

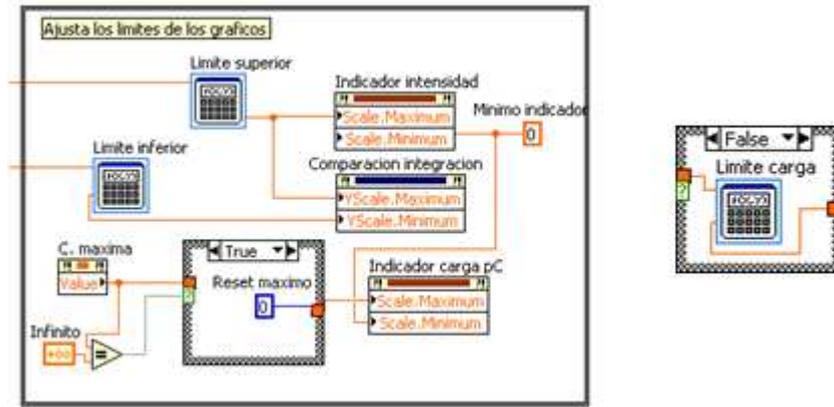


Figura 3.19: Conexionado del diagrama de bloques para el ajuste de los límites del grafico comparación integral software Vs hardware y de los indicadores de aguja.

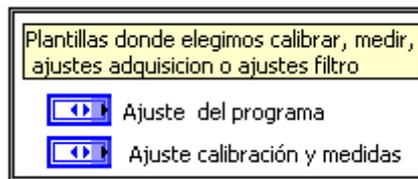


Figura 3.20: Representación grafica del diagrama de bloques para las plantillas

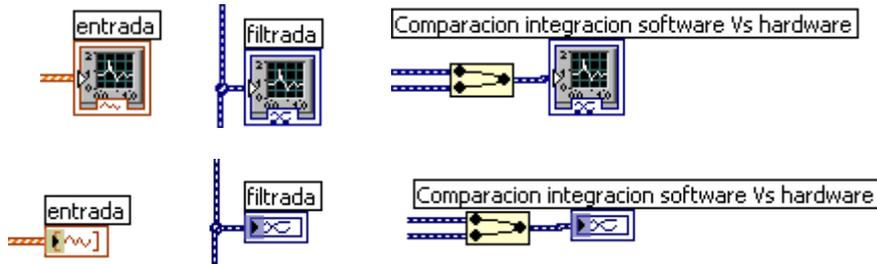


Figura 3.21: Conexionado del diagrama de bloques para los gráficos.

3.1.22.2 Terminales de entrada: Al indicador grafico *entrada* se conecta la señal de salida del subcircuito *Adquisición*. Al indicador grafico *filtrada* se conecta la señal de salida del subcircuito *filtro canal 0*. Al indicador grafico *comparación integral software Vs integral hardware* se conecta la señal de salida del subcircuito *Ganancia del canal 1* y la señal de salida del subcircuito *integración tensión* → *intensidad*.

3.2 **Front panel**

Front panel es la parte de *Labview* creada para visualizar y controlar los indicadores y controles generados en *Block diagram*, en la aplicación creada tiene la forma que se muestra en la Figura 3.22, las variantes de esta imagen se describen detalladamente en los apartados correspondientes.

3.2.1 **Ajuste Adquisición**

La plantilla *ajuste programa* en la página *ajuste adquisición* se muestra en la Figura 3.23.

3.2.1.1.1 Descripción: La plantilla ajuste programa permite ajustar los parámetros de la adquisición y los parámetros del filtro seleccionando una de las pestañas que hay en la parte superior. Para poder modificar unos parámetros u otros solo se tiene que pinchar sobre los iconos correspondientes.

Controles e indicadores: La página *ajuste adquisición* dispone de cuatro controles y un indicador numérico. *Rango vertical* es un control que tiene unos valores predeterminados, y se puede pasar de uno a otro pulsando sobre los pulsadores que hay a la izquierda del indicador si se desea introducir un valor que no está entre los datos ofrecidos se debe hacer un clic con el botón izquierdo del ratón sobre el indicador, se desplegará una lista con todos los valores configurados y en último lugar se muestra la palabra <other>. Pinchando sobre esta palabra se puede introducir el valor deseado teniendo en cuenta que el control esta en mV. Los controles *frecuencia de muestreo* y *rango horizontal* pueden adoptar cualquier valor dentro de los límites establecidos por la tarjeta de adquisición. *Nº de muestras* es un indicador. *Impedancia de entrada* es un control que despliega un menú con dos opciones disponibles.

3.2.2 **Ajuste del filtro**

La plantilla *ajuste programa* en la página *ajuste filtro* se muestra en la Figura 3.24.

3.2.2.1.1 Descripción: Para ajustar los parámetros del filtro selecciona la pestaña que hay en la parte superior derecha de la plantilla *ajuste programa*. Para poder modificar los parámetros de la página *ajuste del filtro* solo se tiene que pinchar sobre los iconos correspondientes.

3.2.2.1.2 Controle e indicadores: La pagina *ajuste filtro* de la plantilla ajuste filtro dispone de seis controles. *Filter topology* y *filter type* permite elegir ente los tipos y topologías de filtro que tiene configuradas el bloque *filtro*. Los otros cuatro controles *#Taps*, *order*, *lower cut-off*, *upper cut-off* pueden adoptar cualquier valor positivo con la restricción que *lower cut-off* debe ser menor que *upper cut-off*.

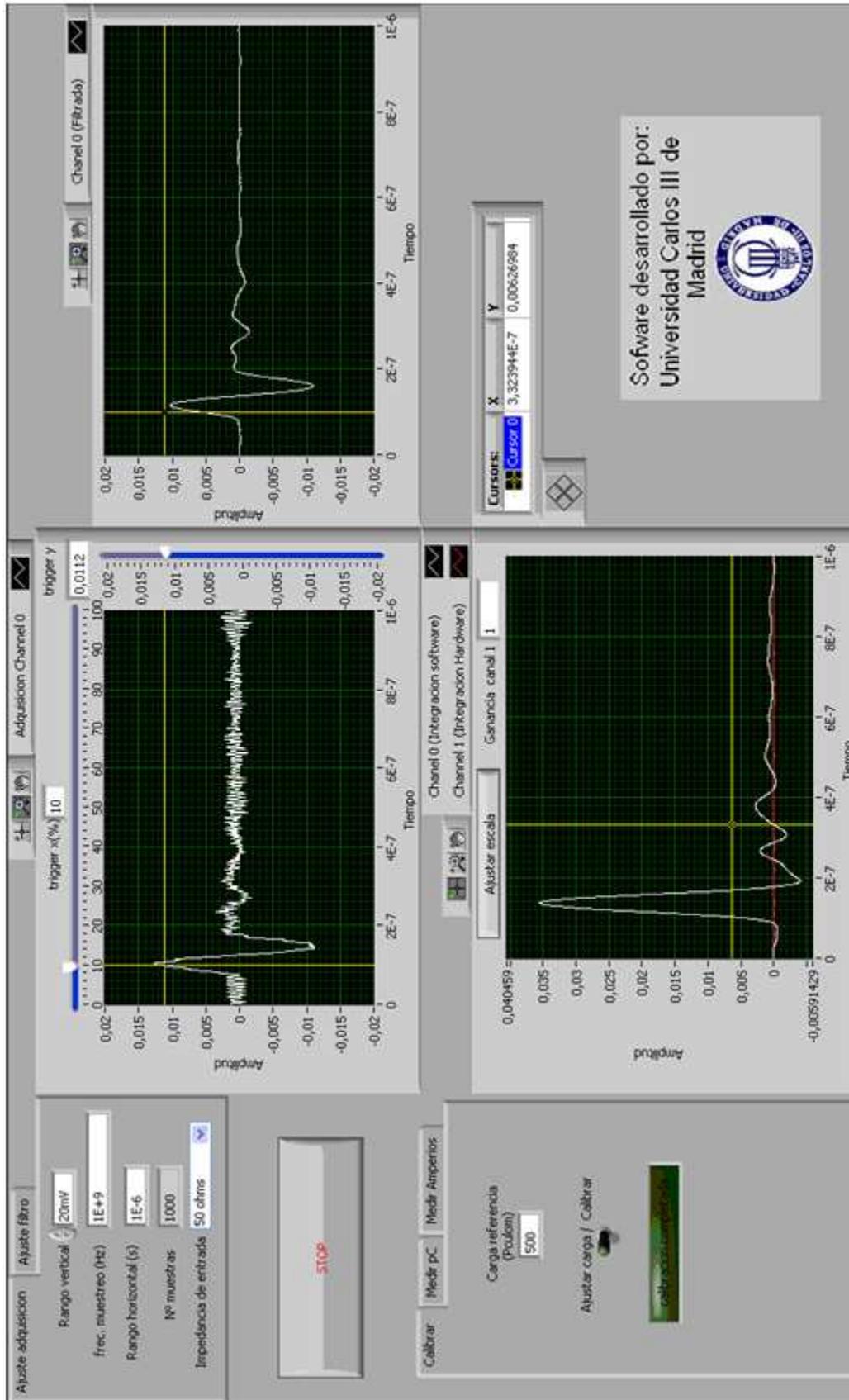


Figura 3.22: Visualización *Front panel* del programa creado



Figura 3.23: Representación grafica en *Front panel* de la plantilla ajuste de adquisición.

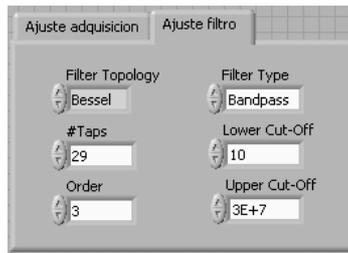


Figura 3.24: Representación grafica en *Front panel* de la plantilla ajuste del filtro.

3.2.3 Grafico señal de entrada

El grafico de la *señal de entrada* se muestra en la Figura 3.25.

3.2.3.1.1 Descripción: Grafica la señal adquirida por la tarjeta de adquisición, es decir, grafica la señal de salida del subcircuito *Adquisición*.

3.2.3.1.2 Controles e indicadores: Tiene dos controles deslizadores que controlan el disparo de la señal, el disparo horizontal se controla de manera porcentual de 0% a 100% y el disparo vertical se regula en voltios pudiendo poner cualquier valor. La regulación de estos controles se puede realizar moviendo el deslizador o ajustando un valor concreto en el indicador numérico que hay junto a los nombres de los controles. Los cursores del grafico están unidos a los valores de los controles como se ha explicado anteriormente, por tanto el valor de la posición de los cursores es la que indica el control correspondiente.

Abajo en la izquierda se encuentran las herramientas del grafico. El botón de la izquierda permite mover el cursor. El botón central permiten hacer distintos tipos de zoom: verticales, horizontales, ventanas... El botón de la derecha sirve para arrastrar la imagen una vez se ha realizado algún tipo de zoom.

En este grafico no se puede controlar ni el cursor ni el zoom durante la ejecución de la aplicación porque la posición del cursor, la escala máxima y mínima del grafico están controlados por controles manuales de manera que cuando utilizas estas herramientas modificando el valor del cursor, la escala máxima o mínima del grafico se mantiene el valor durante un ciclo de la aplicación, en el siguiente se vuelven a retomar los valores de los controles y se muestra la señal completa. Cuando la aplicación esta en reposo si mantienen los zooms que se hagan, porque no se refrescan los valores de los controles.

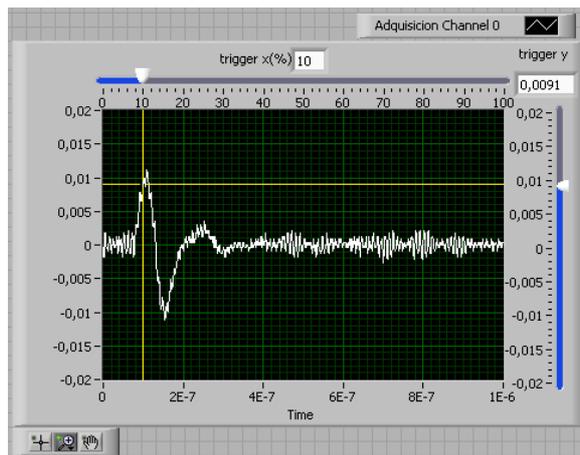


Figura 3.25: Representación grafica en *Front panel* del grafico señal de entrada.

3.2.4 Grafico señal filtrada

El grafico de la *señal filtrada* se muestra en la Figura 3.26.

3.2.4.1.1 Descripción: Grafica la señal adquirida una vez se la ha eliminado la media y ha sido filtrada, es decir, grafica la salida del subcircuito *filtro canal 1*.

3.2.4.1.2 Controles e indicadores: En este grafico tampoco se pueden utilizar las herramientas del grafico de manera fructífera hasta que la aplicación este en reposo, por los motivos expuestos en el apartado *grafica señal de entrada*.

3.2.5 Grafico comparación integral software Vs hardware

El grafico *comparación integral software Vs hardware* se muestra en la Figura 3.27.

3.2.5.1.1 Descripción: Grafica las señales del canal 0 después de haber sido integrada y del canal 1 después de aplicarla la ganancia, es decir, la señal de salida del subcircuito *Ganancia del canal 1* y la señal de salida del subcircuito *integración tensión* → *intensidad*.

3.2.5.1.2 Controles e indicadores: Tiene un pulsador *ajustar escala* para ajustar la escala vertical máxima y mínima del grafico, este ajuste se describe en el apartado *Ajuste de los límites de los indicadores de aguja y el grafico comparación integral hardware Vs software*. También hay un control *ganancia canal 1* para ajustar la ganancia del canal 1.

PROGRAMA CREADO CON LABVIEW

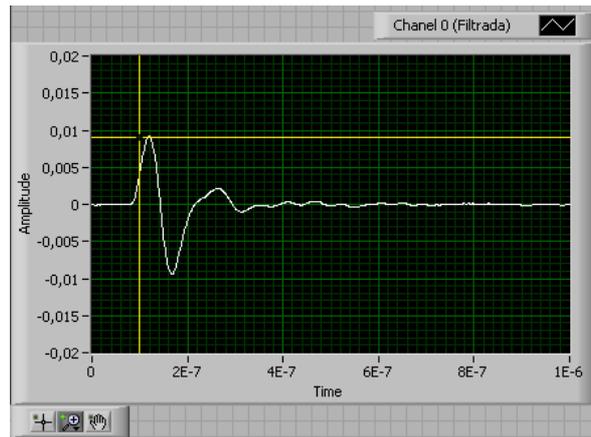


Figura 3.26: Representación grafica en *Front panel* del grafico señal filtrada.

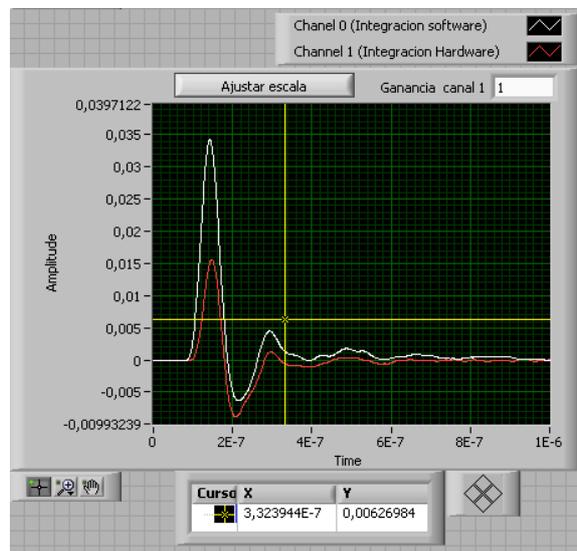


Figura 3.27: Representación grafica en *Front panel* del grafico comparación integral software Vs hardware.

En este grafico el cursor y el zoom horizontal se puede controlar desde las herramientas que hay abajo en la izquierda, se han colocado dos indicadores que muestran el valor en el eje x y en el eje y, a la derecha de estos indicadores se encuentra unos pulsadores en forma de rombo que sirven para desplazar el cursor. El zoom vertical no se puede controlar desde las herramientas del grafico porque se controlan mediante un algoritmo que se describe en los apartados: *Calculo del valor máximo y mínimo absoluto de la integral hardware (canal 0) y la integral software (canal 1)* y *Ajuste de los límites de los indicadores de aguja y el grafico comparación integral hardware Vs software.*

3.2.6 Calibración

La plantilla *Ajuste calibración y medidas* en la página *calibración* se muestra en la Figura 3.28.

- 3.2.6.1.1 Descripción: La página *calibración* permite realizar los ajustes para la calibración. Esta página puede estar en tres estados diferentes, según la Figura 3.28, el estado inicial (imagen de la izquierda), con el interruptor en la posición *ajustar carga* se introduce un valor en pC de la carga a partir de la cual se va a calibrar. Después de conmutar el interruptor se adopta el segundo estado (imagen central), que es un estado de transición hasta que se alcanzan las cuarenta ejecuciones de la aplicación y la calibración se da por concluida, en este estado ya no se puede ajustar la carga de referencia para evitar calibraciones erróneas. El tercer estado (imagen de la derecha) se alcanza cuando la calibración está completada, instante en el que se enciende el *led calibración completada*. A partir de este instante se pueden realizar medidas obteniendo unos datos fiables en pC.
- 3.2.6.1.2 Controles e indicadores: Dispone de un control *carga de referencia (pC)*, interruptor *ajustar carga / Calibrar* y un indicador luminoso *led calibración completada*.

3.2.7 Medidas en pico Culombios

La plantilla *Ajuste calibración y medidas* en la página *medir pC* se muestra en la Figura 3.29.

- 3.2.7.1.1 Descripción: La página *medir pC* permite medir la carga máxima, mínima, promediada e instantánea. Para ver la carga instantánea hay que fijarse en el indicador *carga pC*, para ver la carga mínima hay que fijarse en el indicador *C. mínima*, para ver la carga máxima hay que fijarse en el indicador *C. máxima* y para ver la carga promediada hay que fijarse en el indicador *Carga promedio*. El indicador de aguja *indicador carga pC* tiene tres agujas, la azul marca la carga mínima, la roja la carga máxima y la negra la carga instantánea. Si se quiere refrescar los valores máximo, mínimo y promedio tanto de los indicadores numéricos como los del indicador de aguja se debe excitar el pulsador *reset*. Al pulsar este pulsador también se refrescan los valores del apartado *Medidas en amperios*, de la misma manera si se excita el pulsador *reset carga* (Su nombre en *Front panel* es *reset*) que está ubicado en la página *medir en amperios* se refrescan los valores de este apartado.
- 3.2.7.1.2 Controles e indicadores: Para ello dispone de cuatro indicadores numéricos *carga pC*, *C. máxima*, *C. mínima* y *carga promedio (pC)*, el indicador de aguja *indicador carga numérica* y el pulsador *reset*.

PROGRAMA CREADO CON LABVIEW



Figura 3.28: Representación grafica en *Front panel* de la plantilla calibración.

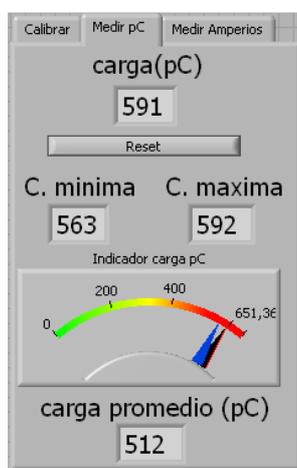


Figura 3.29: Representación grafica en *Front panel* de la plantilla medir en pC.

3.2.8 Medidas en amperios

La plantilla *Ajuste calibracion* y medidas en la pagina *medir amperios* se muestra en la Figura 3.30.

- 3.2.8.1.1 Descripción: La pagina *medir amperios* permite visualizar el valor de *pico máximo*, *mínimo e instantáneo* de la intensidad de las descargas parciales. Para ver el pico de intensidad instantáneo hay que fijarse en el indicador *pico instantáneo*, para ver el pico de intensidad mínimo hay que fijarse en el indicador *menor pico* y para ver el pico de intensidad máximo hay que fijarse en el indicador *mayor pico*. El indicador de aguja *indicador intensidad* tiene tres agujas, la azul marca el menor pico, la roja el mayor pico y la negra el pico instantáneo. La forma de refrescar estos valores de describe en el apartado *Medidas en pico Culombios*.
- 3.2.8.1.2 Controles e indicadores: Dispone de tres indicadores numéricos *pico instantáneo*, *menor pico* y *mayor pico*, el indicador de aguja *indicador de intensidad* y el pulsador *reset carga* (Su nombre en *Front panel* es *reset*).

PROGRAMA CREADO CON *LABVIEW*



Figura 3.30: Representación grafica en *Front panel* de la plantilla medir en amperios.

4 Resultados experimentales.

4.1 Algoritmos de cálculo de carga mediante integral numérica

4.1.1 Integración numérica de toda la señal

El primer algoritmo de cálculo de la carga fue colocar una integral numérica, una vez realizado el tratamiento a la señal adquirida para transformarla en la señal de las descargas que se producen este punto se encuentra a la salida de la fórmula *Inductancia sonda*. De esta manera la salida de la integral devuelve el valor numérico de la integral, valor que es la carga de la descarga parcial en Culombios.

Al realizar la integral tensión \rightarrow intensidad se obtiene un error debido a la componente continua que tiene la señal adquirida, esta componente tiene un valor aleatorio que puede ser de mayor o menor valor, positivo o negativo, estas variaciones en el valor medio hacían fluctuar mucho el valor de la carga como se muestra en las siguientes figuras: Figura 4.1y Figura 4.2.

Como se puede comprobar la variación entre las señales es considerable e introduce un error a tener en cuenta.

Para eliminar el valor medio de la señal se colocó una función calculadora que restaba a la señal adquirida un *offset* regulable mediante una ruleta, era muy difícil eliminar la oscilación de la señal integrada porque el *offset* no era constante y variaba en cada ciclo del programa.

Tras desechar la opción de regular un *offset* se optó por restar el valor medio de la señal. El *offset* de una señal es la media de la misma, se colocó el bloque *mean* que calcula la media de la señal adquirida cada ciclo. Esto permite restar a cada adquisición su valor medio de manera que la señal integrada tiene un valor final igual a cero.

Se comprobó que aunque la señal convergiera a cero eliminando la media, las oscilaciones de la señal eran inevitables, la señal se desplazaba en el eje vertical. Cuando la señal se desplazaba hacia arriba el área positiva se hacía más grande y el área negativa más pequeña, cuando la señal se desplazaba hacia abajo ocurría lo contrario. Gran parte del error se cometía al considerar que la cola de la señal era descarga parcial cuando en realidad es un remanente provocado por el ruido.

RESULTADOS EXPERIMENTALES

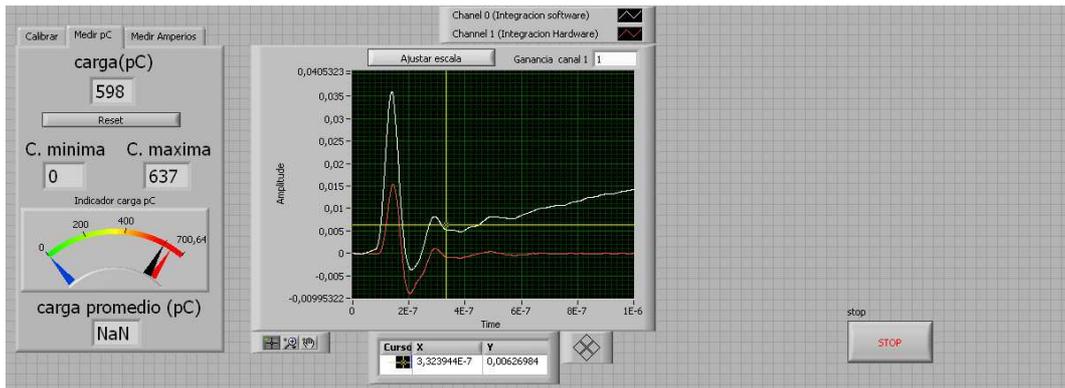


Figura 4.1: Representación de la señal integrada mediante software sin restar la media 1.

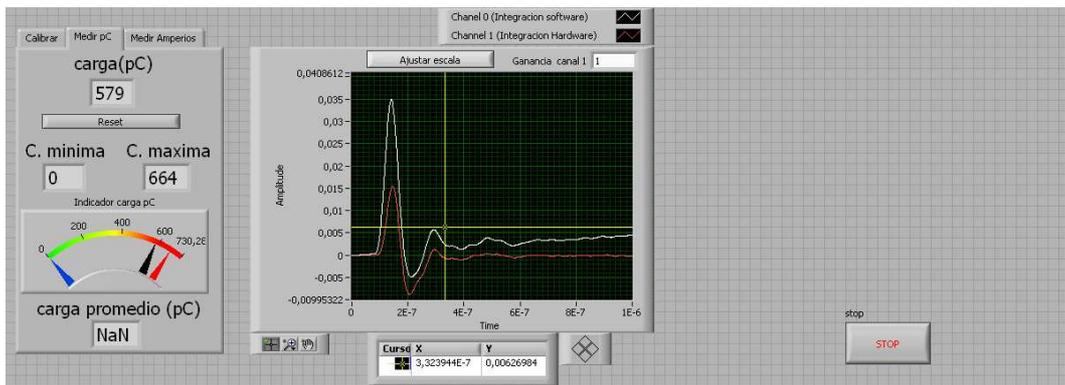


Figura 4.2: Representación de la señal integrada mediante software sin restar la media 2.

4.1.2 Integración numérica de la señal recortada

En la siguiente versión de la aplicación se diseñó un algoritmo capaz de recortar la señal hasta donde el usuario deseara, la carga se calcularía integrando numéricamente la señal recortada. Con este método se elimina el ruido posterior a la descarga parcial, el problema se presenta en saber hasta dónde llega la descarga parcial y donde empieza el ruido. Por tanto con este algoritmo queda a criterio del usuario la duración de la descarga parcial, si el usuario recorta demasiado la señal obtendrá una carga menor de la real, por el contrario si deja pasar ruido obtendrá una carga mayor que la real. La carga de la descarga parcial se continuaba mostrando en Culombios.

Aparte del nuevo problema mostrado en el párrafo anterior, la señal seguía oscilando sobre el eje vertical, cuando la señal se desplazaba hacia arriba el área positiva se hacía más grande y el área negativa más pequeña, cuando la señal se desplazaba hacia abajo ocurría lo contrario. La oscilación era tal que la carga oscilaba entre valores cercanos al cincuenta por ciento, como se muestra en las figuras: Figura 4.3 y Figura 4.4.

RESULTADOS EXPERIMENTALES

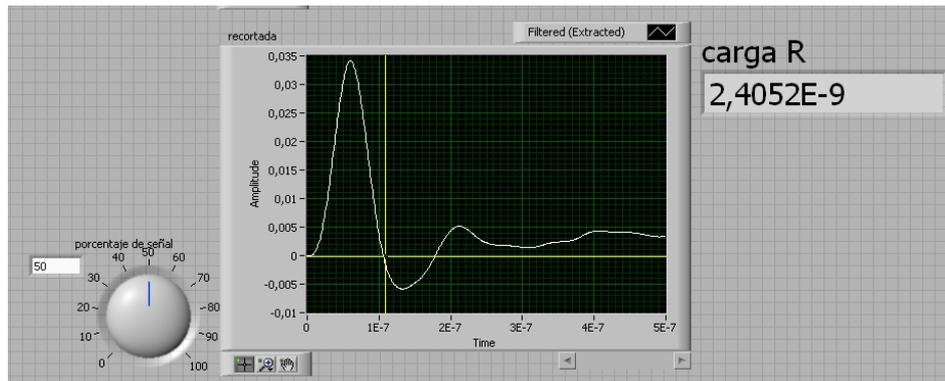


Figura 4.3: Representación de la señal integrada mediante software y recortada sin restar la media 1.

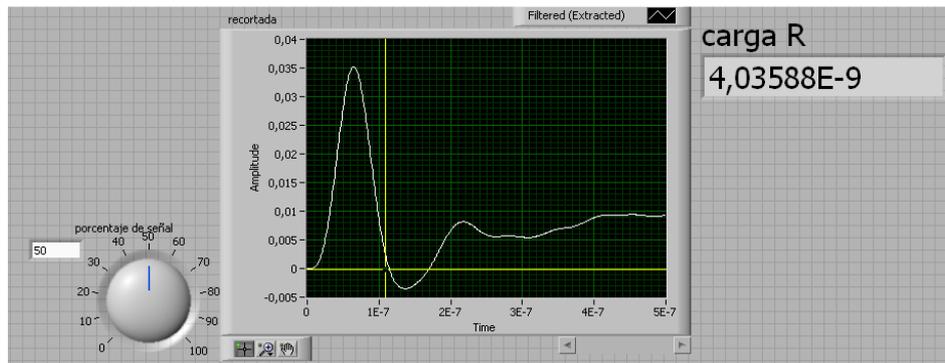


Figura 4.4: Representación de la señal integrada mediante software y recortada sin restar la media 1.

Tras estos resultados se desechó la opción de calcular la carga a partir de la integral numérica por la complejidad de eliminar el ruido que provoca el error de cuantificación de la tarjeta de adquisición. Desde este momento se comenzó a buscar una alternativa al método de cálculo de la carga. La primera idea que se planteó fue buscar una proporción entre un parámetro que se mantuviera estable en la señal integrada (intensidad).

4.2 Algoritmos de estimación de carga mediante la relación de un parámetro.

4.2.1 Estimación de la carga a partir de la primera área positiva con el tiempo base fijado por cursores

Observando que el error de cuantificación de la tarjeta de adquisición se incrementaba conforme avanzaba la señal se optó por buscar un parámetro que estuviera en la parte inicial de la señal. Este parámetro fue la primera área positiva de la señal. Las variaciones que sufría eran mínimas. Esta primera área se asemeja a un triángulo, para calcular su área se debe conocer el tiempo de la base y la amplitud de pico de la señal, de esta manera el área será:

$$\text{Área} = \frac{\text{Base} \times \text{Altura}}{2} = \frac{\text{Tiempo}_{\text{Base}} \times \text{Aplitud}_{\text{de pico}}}{2}$$

RESULTADOS EXPERIMENTALES

El primer algoritmo planteado para realizar el cálculo del área del primer triángulo de la señal mostrado en la Figura 4.5. Este algoritmo obtiene la amplitud del pico de la señal mediante el bloque *Detector de picos* que se utiliza en el subcircuito *Calcula el primer máximo de la integral* (Este parámetro se calcula de manera análoga para todos algoritmos que requieran el valor del primer pico de la señal). El tiempo de la base del triángulo se calcula mediante la diferencia de dos cursores colocados uno al inicio y otro al final del triángulo.

El área que se utiliza como referencia se calcula mediante una fórmula con tres entradas Amplitud de pico, cursor 1 y cursor 2. La ecuación del bloque *fórmula* es:

$$\text{Área} = \frac{(\text{Cursor}_2 - \text{Cursor}_1) \times \text{Amplitud de pico}}{2}$$

Como este método de cálculo requiere establecer una relación entre una carga conocida y el área del primer triángulo, se debe conseguir un valor fiable del área del primer triángulo para establecer dicha relación. Para ello se generó un nuevo subcircuito Automatización de la calibración.

Al realizar cálculos experimentales se comprobó que no era un algoritmo robusto porque al variar la carga del calibrador se debían reajustar los cursores debido a que el tiempo base variaba, lo mismo ocurría pero a mayor escala cuando se medían descargas parciales. Cuando se cambiaba el valor del *trigger* horizontal la señal se desplazaba en el eje horizontal ello conllevaba que los cursores quedaran fuera del triángulo, desconociendo si estos mostraban el tiempo base real. Todas estas variaciones que se debían realizar en los cursores introducían grandes errores porque es imposible guardar siempre la misma proporción en la colocación de los cursores.

Una vez que se descartó este algoritmo se buscó otro que automatizara el cálculo del tiempo base.

4.2.2 Estimación de la carga a partir de la primera área positiva con el tiempo base igual al doble del tiempo de cola

El primer algoritmo que calcula automáticamente el tiempo base del triángulo se describe en este apartado. Se hace una suposición, el frente de subida es igual que el de bajada. No es difícil calcular el tiempo transcurrido entre el máximo y el siguiente paso por cero. El bloque *Detector de picos* aparte del valor del primer pico devuelve el tiempo donde se produce. Colocando un bloque *Localizador de un elemento en un array* devuelve la muestra donde se encuentra el primer valor igual a cero, para encontrar este valor se eliminó toda la parte negativa de la señal de otra manera el bloque no podía encontrar el valor cero debido a que los valores de la señal tenían seis decimales y era improbable que esos seis decimales fueran cero. Cuando se tenía la muestra donde se alcanzaba el valor cero se dividía entre la frecuencia de muestreo y así se obtenía el tiempo donde se alcanzaba el valor cero.

Una vez obtenido estos tres parámetros se calcula el área mediante un bloque *fórmula* con la siguiente ecuación:

RESULTADOS EXPERIMENTALES

$$\begin{aligned} \text{Área} &= \frac{2 \times (\text{Tiempo}_{\text{paso por cero}} - \text{Tiempo}_{\text{de pico}}) \times \text{Aplitud}_{\text{de pico}}}{2} \\ &= (\text{Tiempo}_{\text{paso por cero}} - \text{Tiempo}_{\text{de pico}}) \times \text{Aplitud}_{\text{de pico}} \end{aligned}$$

Esta área será el parámetro utilizado para realizar la calibración.

Se realizaron tres medidas para cinco tiempos distintos, en las siguientes condiciones: onda completa, periodo de muestreo= 10^{-6} s, *trigger* horizontal al 10%, *trigger* vertical a 1V *Labview* y 2,9V *Techimp* no estaba sincronizado, una tensión de 2,4kV.

El *trigger* de nuestra aplicación y el de *Techimp* tienen valores distintos. Esto se debe a que la ganancia de la sonda que extrae la señal a *Labview* y la ferrita que proporciona la señal a *Techimp* tiene distinta ganancia. A la hora de calibrar, los impulsos medidos tienen distinta amplitud, por tanto se debía obtener un valor que indicara la relación entre los dos *trigger*. Este valor fue la amplitud de la carga máxima, cuando en *Techimp* había una amplitud de 0,104V en *Labview* la amplitud era de 0,036V. Por tanto la relación a utilizar durante las medidas de descargas parciales es $\text{Trigger}_{\text{Labview}} = 0,343 \cdot \text{Trigger}_{\text{Techimp}}$

Los valores que hay a la derecha de los recuadros *Techimp* son los valores de carga obtenidos con el software desarrollado por la empresa *Techimp* y los valores que hay a la derecha de los recuadros *Labview* son los valores de carga obtenidos con la aplicación que se ha desarrollado. Los resultados obtenidos con este algoritmo se muestran en la Tabla 1 y Grafica 1.

La diferencia (p.u.) se calcula mediante la siguiente fórmula:

$$\text{Diferencia [p. u.]} = \frac{(\text{Techimp} - \text{Labview})}{\frac{(\text{Techimp} + \text{Labview})}{2}}$$

El error promediado se calcula mediante la siguiente fórmula:

$$\text{Error promediado [\%]} = \frac{\sum \text{Diferencia (p. u.)}_{\text{Tiempo } t}}{\text{N}^{\circ} \text{ de sumandos}} \times 100$$

El error medio se calcula mediante la siguiente fórmula:

$$\text{Error medio [\%]} = \frac{\sum \text{Error promediado}}{\text{N}^{\circ} \text{ de sumandos}}$$

De estos resultados se concluye que el error disminuye conforme el tiempo aumenta a excepción de las medidas en dos minutos y que el error medio de todas las medidas es del 12,9%.

Con la intención de reducir el error se desarrollaron diferentes algoritmos de cálculo.

RESULTADOS EXPERIMENTALES



Figura 4.5: Representación de la señal integrada mediante software restando la media y con dos cursores para calcular la primera área.

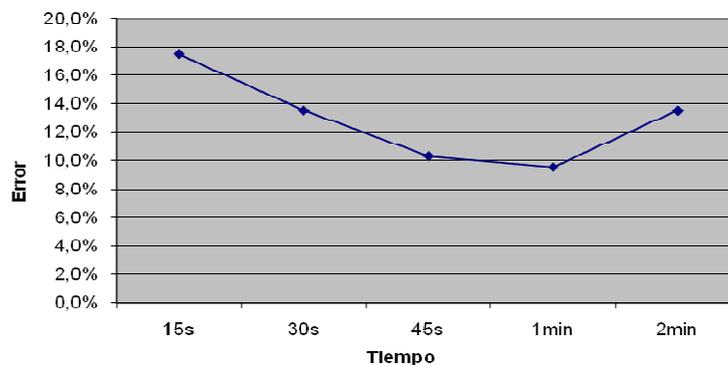
	tiempo	15s	30s	45s	1min	2min
1medida	<i>Techimp</i>	17927	16592	17447	17453	17292
	<i>Labview</i>	20761	20593	19546	21005	20203
	Diferencia (p.u.)	0,147	0,215	0,113	0,185	0,155

2medida	<i>Techimp</i>	17343	17994	17000	16740	17752
	<i>Labview</i>	20700	20539	18491	17196,9	21128
	Diferencia (p.u.)	0,176	0,132	0,084	0,027	0,174

3medida	<i>Techimp</i>	16518	17460	16456	16388	16667
	<i>Labview</i>	20224	18522	18423	17674	18002
	Diferencia (p.u.)	0,202	0,059	0,113	0,076	0,077

						Error medio
Error promediado (%)	17,5%	13,5%	10,3%	9,6%	13,5%	12,9%

Tabla 1: Resultados del algoritmo triangulo con el doble del tiempo de cola.



Grafica 1: Resultados del error promediado en algoritmo triangulo con el doble del tiempo de cola.

RESULTADOS EXPERIMENTALES

4.2.3 Estimación de la carga a partir de la primera área positiva estimando el tiempo de inicio.

Con este algoritmo se pretende tener en cuenta el frente de subida que se ignoraba en el apartado *Estimación de la carga a partir de la primera área positiva con el tiempo base igual al doble del tiempo de cola*.

Para tener en cuenta el frente de subida se necesita el tiempo donde empieza el impulso, es imposible saber el instante en que se produce el impulso, por ello se debe estimar un valor de inicio. Para estimar este valor se calcula el tiempo en que la señal supera en amplitud el 10% de la amplitud de pico (valor superior al ruido que impide conocer el inicio de la señal). Por otro lado, se obtiene el tiempo en que se produce la amplitud de pico. Teniendo estos dos valores se puede hacer una estimación del tiempo donde empieza el impulso, para ello se utiliza una fórmula con la siguiente ecuación:

$$t_{\text{inicio triangulo}} = t_{\text{pico}} - \frac{10}{9}(t_{\text{pico}} - t_{10\%})$$

Una vez calculado el tiempo donde comienza el impulso, se tiene que calcular el tiempo donde termina la primera área positiva de la señal. El tiempo que termina se calcula de la misma manera que en el apartado *Estimación de la carga a partir de la primera área positiva con el tiempo base igual al doble del tiempo de cola*. Para determinar la base del triangulo se utiliza una fórmula con la siguiente ecuación:

$$Tiempo_{\text{base}} = Tiempo_{\text{fin triangulo}} - Tiempo_{\text{inicio triangulo}}$$

El pico de la amplitud se calcula de la misma manera que en el apartado *Estimación de la carga a partir de la primera área positiva con el tiempo base igual al doble del tiempo de cola*.

Una vez obtenido el tiempo base y el pico de amplitud se calcula el área mediante un bloque *fórmula* con la siguiente ecuación:

$$\text{Área} = \frac{Tiempo_{\text{base}} \times \text{Aplitud}_{\text{de pico}}}{2}$$

Esta área será el parámetro utilizado para realizar la calibración.

Se realizaron tres medidas para cinco tiempos distintos, en las siguientes condiciones: onda completa, periodo de muestreo= 10^{-6} s, *trigger* horizontal al 10%, *trigger* vertical a 0,4V *Labview* y 1,2V *Techimp* no estaba sincronizado, una tensión de 2,4kV. Los resultados obtenidos con este algoritmo se muestran en la Tabla 2 y Grafica 2.

RESULTADOS EXPERIMENTALES

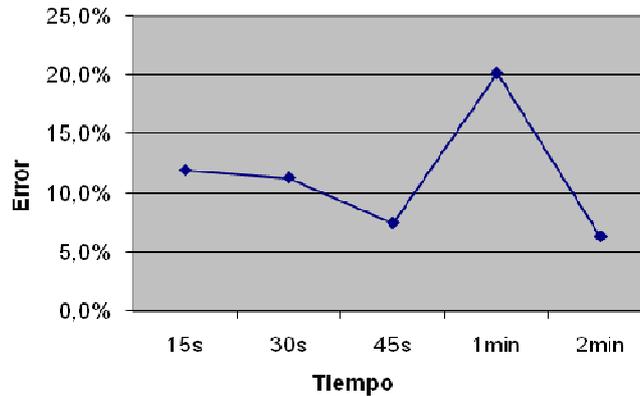
	tiempo	15s	30s	45s	1min	2min
1medida	<i>Techimp</i>	8546	9146	10025	10106	11087
	<i>Labview</i>	8772	11234	10971	10603	11793
	Diferencia (p.u.)	0,026	0,205	0,090	0,048	0,062

2medida	<i>Techimp</i>	9719	8936	10543	10021	10276
	<i>Labview</i>	11834	8969	12034	13570	10739
	Diferencia (p.u.)	0,196	0,004	0,132	0,301	0,044

3medida	<i>Techimp</i>	11314	10084	10444	11223	11270
	<i>Labview</i>	9872	8842	10465	14548	12254
	Diferencia (p.u.)	0,136	0,131	0,002	0,258	0,084

						Error medio
Error promediado (%)	11,9%	11,3%	7,5%	20,2%	6,3%	11,5%

Tabla 2: Resultados del algoritmo triangulo estimando el inicio.



Gráfica 2: Grafica del error promediado en algoritmo triangulo estimando el inicio.

De estos resultados se concluye que el error disminuye conforme el tiempo aumenta a excepción de las medidas en un minuto y que el error medio de todas las medidas es del 11,5%.

Con este algoritmo se reduce un poco el error pero requiere mucho cálculo, esto conlleva a la utilización de memoria del ordenador y no interesa porque ralentiza el proceso de adquisición que es realmente lo importante.

4.2.4 Estimación de la carga a partir del primer mínimo de la integral.

Este algoritmo realiza la integral discreta de la señal intensidad, obtiene el primer mínimo (valle) de esta señal y devuelve su valor, este valor es el utilizado como referencia para hallar la carga.

El mínimo de una integral se corresponde con un paso por cero de la señal primitiva, por tanto para este caso el valor del primer mínimo se corresponde con el área comprendida entre el inicio de la señal y el segundo paso por cero a partir del inicio de la señal. Con este algoritmo se tiene en cuenta las dos primeras áreas.

RESULTADOS EXPERIMENTALES

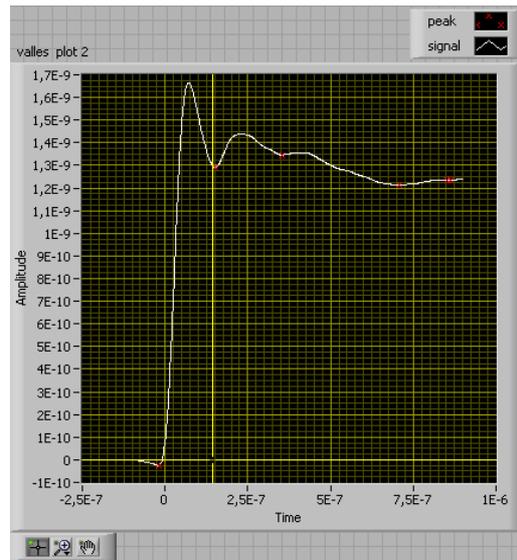


Figura 4.6: Representación de la señal dos veces integrada con todos sus valles señalizados en rojo.

En la Figura 4.6 se muestra la integral de la intensidad que es la carga en función del tiempo. Se calculan los mínimos de esta señal y se saca el valor del primer mínimo, este valor será el parámetro utilizado para realizar la calibración.

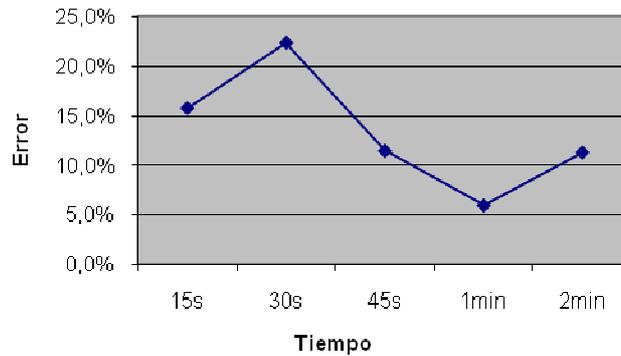
Se realizaron tres medidas para cinco tiempos distintos, en las siguientes condiciones: onda completa, periodo de muestreo= 10^{-6} s, *trigger* horizontal al 10%, *trigger* vertical a 0,4V *Labview* y 1,2V *Techimp* no estaba sincronizado, una tensión de 2,4kV. Los resultados obtenidos con este algoritmo se muestran en la Tabla 3 y Grafica 3.

De estos resultados se concluye que el error no tiene mucha correlación con una recta pero disminuye conforme el tiempo aumenta y que el error medio de todas las medidas es del 13,4%. Un error mayor que en los apartados que se calculaba el área aproximándola a un triángulo. Como el cálculo del área es más exacto con este algoritmo se concluyo que el punto débil de este algoritmo es tener en cuenta las dos primeras áreas porque la señal oscila sobre el eje vertical. Cuando la señal se desplaza hacia arriba el área positiva se hacía más grande y el área negativa más pequeña esto conlleva que el mínimo calculado sea mayor, cuando la señal se desplazaba hacia abajo ocurría lo contrario. Para intentar obtener mejores resultados se desarrollo un algoritmo con el mismo diseño que este pero trabajando con el primer máximo que equivale a la primera área.

RESULTADOS EXPERIMENTALES

	tiempo	15s	30s	45s	1min	2min	
1medida	<i>Techimp</i>	3461	3866	3507	3128	3410	
	<i>Labview</i>	3151	3700	2800	3261	3802	
	Diferencia (p.u.)	0,094	0,044	0,224	0,042	0,109	
2medida	<i>Techimp</i>	2955	3966	3478	3286	3935	
	<i>Labview</i>	2951	2729	2968	2726	3802	
	Diferencia (p.u.)	0,001	0,370	0,158	0,186	0,034	
3medida	<i>Techimp</i>	3282	3537	2845	2906	3596	
	<i>Labview</i>	1894	2163	2635	2877	2633	
	Diferencia (p.u.)	0,536	0,482	0,077	0,010	0,309	
						Error medio	
Error promediado (%)		15,8%	22,4%	11,5%	5,9%	11,3%	13,4%

Tabla 3: Resultados del algoritmo estimación de la carga a partir del primer mínimo de la integral.



Grafica 3: Grafica del error promediado en algoritmo estimación de la carga a partir del primer mínimo de la integral.

4.2.5 Estimación de la carga a partir del primer máximo de la integral.

Este algoritmo realiza la integral discreta de la señal intensidad, obtiene el primer máximo (pico) de esta señal y devuelve su valor, este valor es el utilizado como referencia para hallar la carga.

El máximo de una integral se corresponde con un paso por cero de la señal primitiva, por tanto para este caso el valor del primer máximo se corresponde con el área comprendida entre el inicio de la señal y el primer paso por cero a partir del máximo medido. Es un cálculo más exacto de la primera área positiva que en los casos que se supone que esta primera área es un triángulo.

En la Figura 4.7 se muestra la integral de la intensidad que es la carga en función del tiempo. Se calculan los máximos de esta señal y se saca el valor del primer máximo, este valor es el utilizado como referencia para hallar la carga.

RESULTADOS EXPERIMENTALES

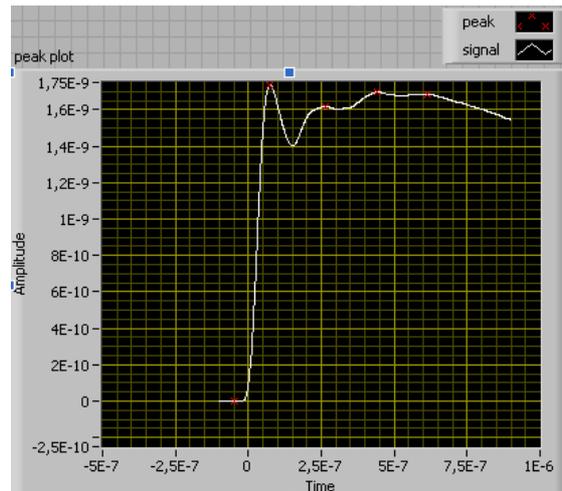


Figura 4.7: Representación de la señal dos veces integrada con todos sus picos señalizados en rojo.

Se detectó que en este algoritmo se cometía un error. Este error consistía en que el primer máximo que detecta el bloque *Detector de picos* se encuentra en la zona de ruido y ese no es el máximo deseado (Figura 4.7. Los puntos rojos son los máximos detectados). Este error también lo cometía el algoritmo Estimación de la carga a partir del primer mínimo de la integral. (Figura 4.6. Los puntos rojos son los máximos detectados). Este error no afectaba al cálculo de la carga máxima pero sí al cálculo la carga mínima y carga promediada, para solucionar este problema se utilizó el subcircuito Recortar señal configurado para eliminar todos los datos adquiridos antes del disparo del impulso.

El subcircuito Recortar señal, inicialmente se colocó antes del bloque *Detector de picos* y eliminaba la porción de la señal que se encontraba antes del disparo. Con esta configuración se integraba antes de recortar por tanto el ruido anterior al impulso se integraba, esto implica que la señal integrada iba arrastrando un error que desplazaba a la integral de manera que el máximo también era desplazado hacia arriba y hacia abajo, cuando el *trigger* horizontal es pequeño el error es pequeño pero cuando el *trigger* horizontal es grande el error es considerable (Estas variaciones se muestran en la Figura 4.8 y Figura 4.9). Para subsanar este error únicamente se recortó la señal antes de integrar. Con esta nueva configuración se obtienen unos resultados aceptables Figura 4.10, Figura 4.11 y Figura 4.12.

RESULTADOS EXPERIMENTALES

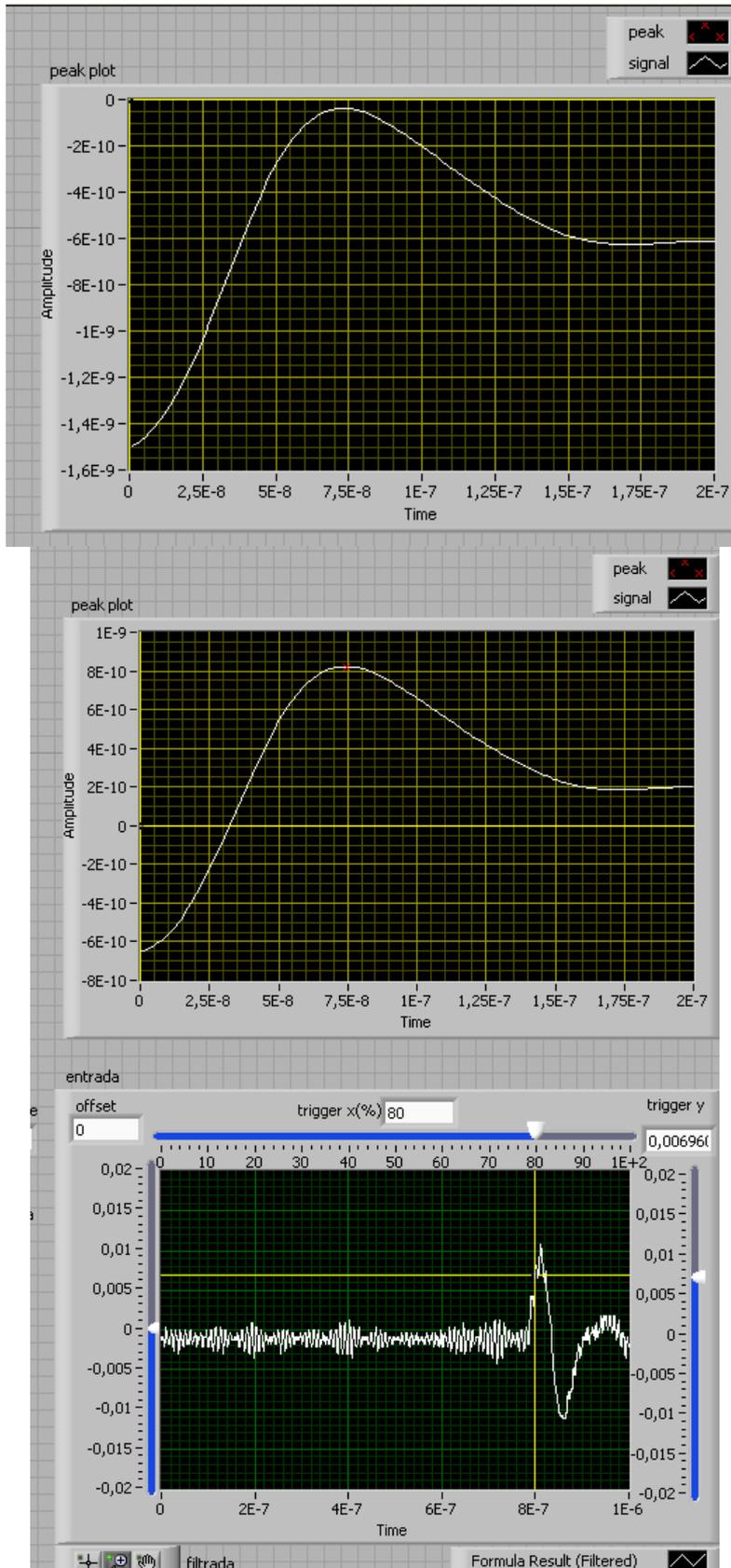


Figura 4.8: Representación de la señal de entrada y de la segunda integral recortada después de integrar.

RESULTADOS EXPERIMENTALES

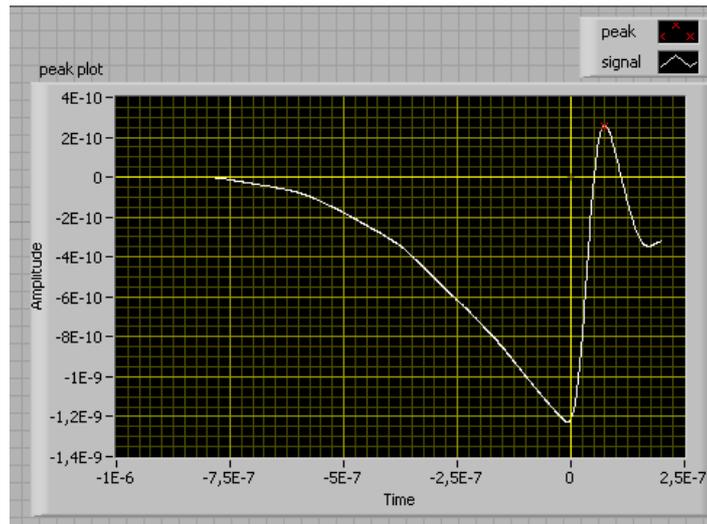


Figura 4.9: Representación del error que se comete por el ruido antes del disparo.

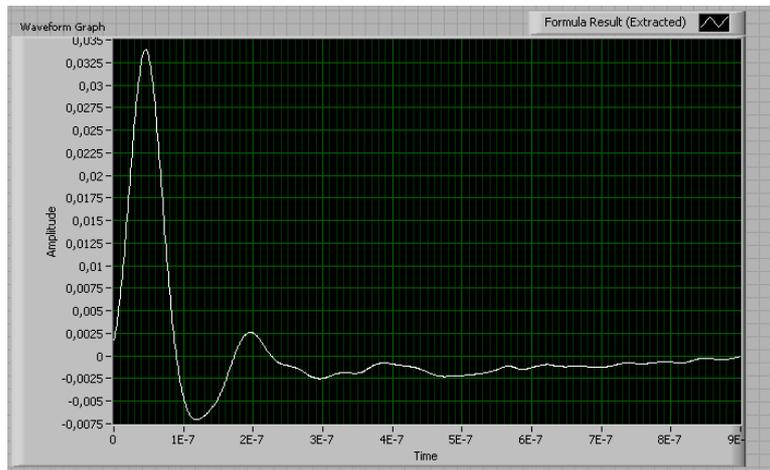


Figura 4.10: Representación de la primera integral recortada a partir del *trigger*.

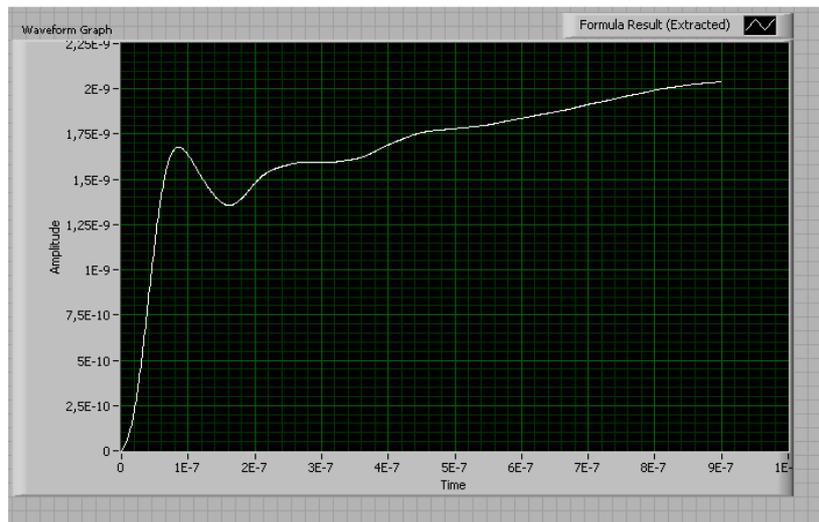


Figura 4.11: Representación de la segunda integral recortada a partir del *trigger*.

RESULTADOS EXPERIMENTALES

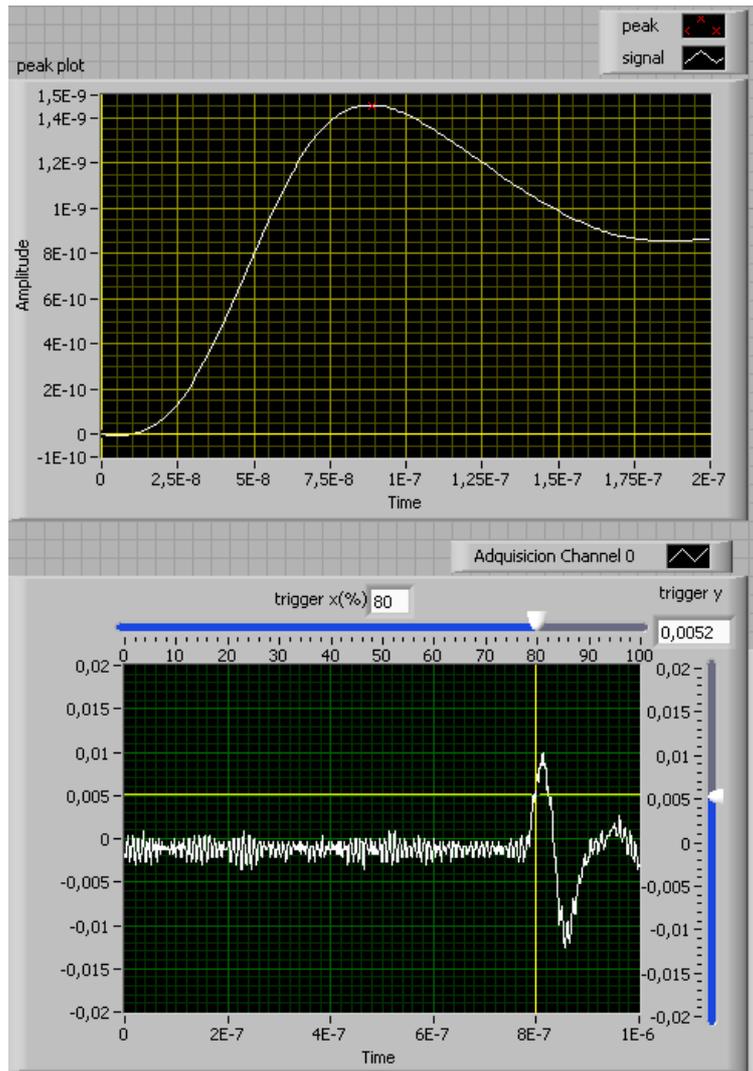


Figura 4.12: Representación de la segunda integral y de la señal de entrada recortadas antes de integrar.

Una vez corregidos todos los errores observados se realizaron medidas para conocer la calidad de este algoritmo.

Inicialmente Se realizaron tres medidas para cinco tiempos distintos pero se observaron buenos resultados, para contrastar mejor estos resultados se realizaron diez nuevas medidas para siete tiempos distintos, en las siguientes condiciones: onda completa, periodo de muestreo= 10^{-6} s, *trigger* horizontal al 10%, *trigger* vertical a 0,1V *Labview* y 0,35V *Techimp* no estaba sincronizado, una tensión de 2,4kV. Los resultados obtenidos con este algoritmo se muestran en la Tabla 4 y Grafica 4.

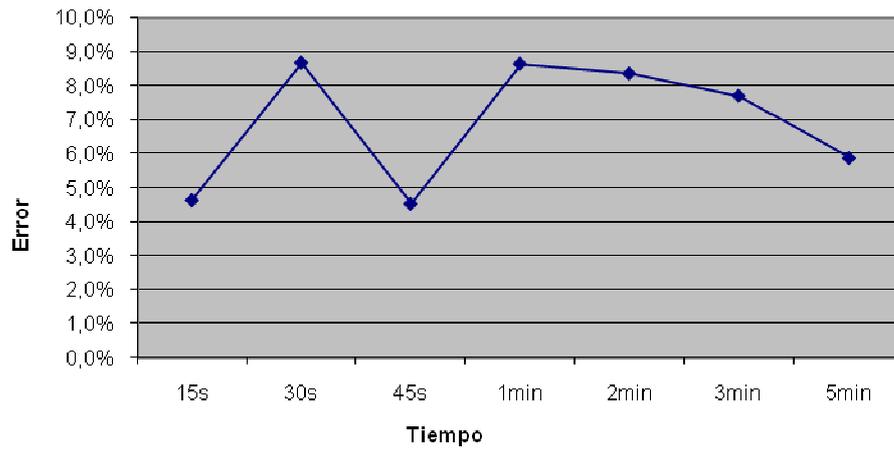
De estos resultados se concluye que el error disminuye conforme el tiempo aumenta a excepción de las medidas en un 15 y 45 s y que el error medio de todas las medidas es del 6,9%. Resultados aceptables e inferiores a los otro algoritmos desarrollados.

RESULTADOS EXPERIMENTALES

	tiempo	15s	30s	45s	1min	2min	3min	5min	
1medida	<i>Techimp</i>	1879	2283	2137	2209	2858	2499	2956	
	<i>Labview</i>	1826	2106	2355	2198	2395	2725	2634	
	Diferencia (p.u.)	0,029	0,081	0,097	0,005	0,176	0,087	0,115	
2medida	<i>Techimp</i>	1956	2475	2691	2027	2533	2749	2541	
	<i>Labview</i>	1855	2384	2433	2315	2919	3222	2719	
	Diferencia (p.u.)	0,053	0,037	0,101	0,133	0,142	0,158	0,068	
3medida	<i>Techimp</i>	1846	1976	2124	1921	2655	2410	2426	
	<i>Labview</i>	1695	1851	2109	1929	2677	2278	2521	
	Diferencia (p.u.)	0,085	0,065	0,007	0,004	0,008	0,056	0,038	
4medida	<i>Techimp</i>	2034	2079	2167	1935	2569	2669	2407	
	<i>Labview</i>	1876	1793	2120	2090	2470	2360	2357	
	Diferencia (p.u.)	0,081	0,148	0,022	0,077	0,039	0,123	0,021	
5medida	<i>Techimp</i>	2255	2196	2486	2075	2553	2671	2685	
	<i>Labview</i>	2307	2162	2266	2069	2908	2817	2561	
	Diferencia (p.u.)	0,023	0,016	0,093	0,003	0,130	0,053	0,047	
6medida	<i>Techimp</i>	1956	2690	2400	1925	2490	2475	2734	
	<i>Labview</i>	1999	1994	2296	1766	2414	2534	2580	
	Diferencia (p.u.)	0,022	0,297	0,044	0,086	0,031	0,024	0,058	
7medida	<i>Techimp</i>	1841	2395	2116	2364	2432	2765	2372	
	<i>Labview</i>	1778	2413	2124	2327	2622	2558	2855	
	Diferencia (p.u.)	0,035	0,007	0,004	0,016	0,075	0,078	0,185	
8medida	<i>Techimp</i>	1864	2200	2111	2523	2490	2427	2534	
	<i>Labview</i>	1981	2191	2115	2420	2511	2351	2616	
	Diferencia (p.u.)	0,061	0,004	0,002	0,042	0,008	0,032	0,032	
9medida	<i>Techimp</i>	2082	1885	2438	2385	2286	2305	2268	
	<i>Labview</i>	2045	2009	2415	3102	2297	2486	2288	
	Diferencia (p.u.)	0,018	0,064	0,009	0,261	0,005	0,076	0,009	
10medida	<i>Techimp</i>	1919	2287	2371	2080	2285	2356	2503	
	<i>Labview</i>	2036	1969	2557	2644	2856	2163	2464	
	Diferencia (p.u.)	0,059	0,149	0,075	0,239	0,222	0,085	0,016	
									Error medio
error promediado (%)		4,7%	8,7%	4,5%	8,7%	8,4%	7,7%	5,9%	6,9%

Tabla 4: Resultados del algoritmo estimación de la carga a partir del primer máximo de la integral.

RESULTADOS EXPERIMENTALES



Grafica 4: Grafica del error promediado en algoritmo estimación de la carga a partir del primer máximo de la integral.

5 Conclusiones y trabajos futuros.

5.1 Conclusiones

Tras realizar este proyecto se puede concluir:

La integral realizada por el software *Labview* a partir del sensor sencillo (utilizado en el canal 0) es igual a la señal dada por el transformador de corriente de alta frecuencia de marca Bergoz (utilizado en el canal 1) Estos dos elementos tienen el mismo ancho de banda (400 MHz) porque cuando se ha usado la ferrita de *Techimp* o el sensor integrador el resultado ha sido distinto obteniendo ligeras diferencias entre las formas de onda resultantes.

El cálculo de la carga mediante el algoritmo Estimación de la carga a partir del primer máximo de la integral. resulta ser fiable porque en la comparación con el cálculo de la carga mediante *Techimp* se obtiene una diferencia media del 6,9%, diferencia muy pequeña. *Techimp* no certifica que las cargas obtenidas por su programa sean exactas ni que sean la verdad absoluta, por tanto las cargas calculadas con *Techimp* nos sirven como referencia orientativa no como un objetivo exacto.

Durante la medición de descargas parciales se producen muchas más de las que se miden, esto implica que se pueden producir descargas máximas o mínimas en los instantes posteriores a la adquisición de otra descarga. Es imposible adquirir las mismas descargas en los dos programas (*Techimp* y *Labview*) a la vez por ello una diferencia del 6,9% es una diferencia aceptable.

5.2 Trabajos futuros

- Clasificar descargas parciales
- Representar descargas parciales en fase

6 Anexo 1-Elementos utilizados de labview

6.1 **Estructuras:**

Las estructuras contienen bloques que se ejecutan de distinto modo dependiendo del tipo de estructura que los envuelva.

6.1.1 **While loop**

La estructura *While loop* se muestra en la Figura 6.1.

6.1.1.1 Descripción: Esta estructura es equivalente al bucle *while* en los lenguajes de programación convencionales. La estructura se ejecuta mientras el terminal condicional no este activo, en el momento que se active el proceso termina.

6.1.1.2 Terminales de entrada: Terminal condicional, cuando este terminal se activa, la estructura deja de ejecutarse.

6.1.2 **For loop**

La estructura *For loop* se muestra en la Figura 6.2.

6.1.2.1 Descripción: Esta estructura es equivalente al bucle *for* en los lenguajes de programación convencionales. La estructura se ejecuta tantas veces como se determine, el número de ejecuciones se ajusta en la parte superior izquierda de la estructura en el terminal *loop count*.

6.1.2.2 Terminales de entrada: *Loop count*, número de veces que se va a ejecutar la estructura.

6.1.2.3 Terminales de salida: *Loop iteration*, número de veces que se ha ejecutado la estructura.

6.1.3 **Case structure**

La estructura *Case structure* se muestra en la Figura 6.3.

6.1.3.1 Descripción: Esta estructura está formada por varios subdiagramas o casos, tantos como se configure la estructura (como mínimo dos, verdadero o falso). A cada caso se le asocia un valor, el selector de subdiagramas lee el valor que le entra entonces activa y coloca en la parte visible el subdiagrama que hay asociado a ese valor, mientras los demás subdiagramas permanecen inhabilitados. Junto con el indicador de subdiagrama hay unas flechas y un menú desplegable que permite elegir entre los distintos subdiagramas.

6.1.3.2 Terminales de entrada: *Caseselector*, selecciona entre los distintos subdiagramas generados.

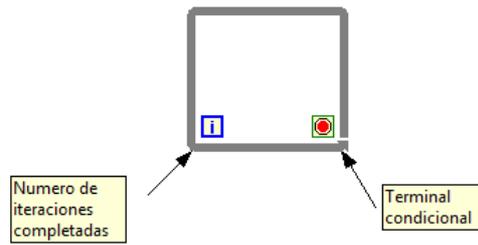


Figura 6.1: Representación grafica de la estructura *While loop*.

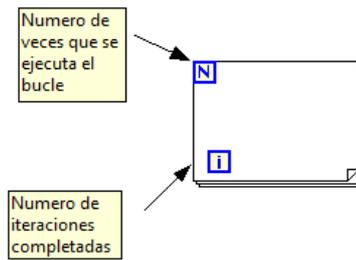


Figura 6.2: Representación grafica de la estructura *For loop*.

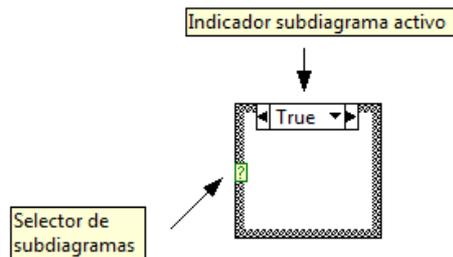


Figura 6.3: Representación grafica de la estructura *Case structure*.

6.1.4 *Flat Sequence Structure*

La estructura *Flat Sequence Structure* se muestra en la Figura 6.4.

- 6.1.4.1 Descripción: Esta estructura está formada por varios subdiagramas o casos, tantos como se configure. Con esta estructura se asegura que todos los subdiagramas se ejecutan en el orden que se desee, la estructura cambia de subdiagrama y empieza a ejecutar el siguiente cuando el subdiagrama en ejecución ha finalizado.

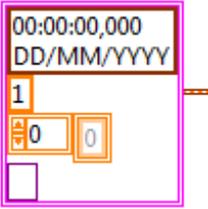
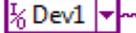


Figura 6.4: Representación grafica de la estructura *Flat Sequence Structure*.

6.2 *Formato de datos:*

Control	Indicador	Constante	Conexión	Tipo de dato	Descripción
				<i>Double-precision, Floating-point numeric</i>	Transporta y almacena números reales (64bits).
				<i>Unsigned integer numeric</i>	Transporta y almacena datos del tipo entero (16 bits) y doble entero (32 bits) sin signo.
				<i>Boolean</i>	Transporta y almacena datos del tipo booleano (0/1, falso/verdadero)
				<i>String</i>	Transporta cadenas de caracteres de longitud variable, que además de datos numéricos pueden ser palabras.
				<i>Array</i>	Está diseñado para transportar datos en forma de <i>arrays</i> .
				<i>Cluster</i>	Transporta varios datos instantáneamente por una única conexión. Puede contener varios tipos de datos, será marrón si todos los elementos en el <i>cluster</i> son <i>numeric</i> . Será rosa si los elementos del <i>cluster</i> son de distinto tipo. Será negro y amarillo para <i>Error code cluster</i> , proporciona información del error, si existe error o no, el código del error, en qué función se ha producido y cuál es la causa que lo ha generado.

ANEXO 1 – ELEMENTOS UTILIZADOS DE LABVIEW

		Por defecto coloca un numero real		<i>Dynamic data</i>	Incluye datos asociados a la señal y atributos que dan información sobre la señal, como por ejemplo el nombre de la señal, el instante en que la señal fue adquirida y el tiempo transcurrido entre muestras (dt).
				<i>Waveform</i>	Transporta los datos de la señal, el instante en que se inicio la señal y el tiempo transcurrido entre muestras (dt).
				<i>I/O Name</i>	Te permite configurar y seleccionar las entradas y salidas que dispongas para poder trabajar con ellas dentro del VI's.

6.3 Conversión entre tipos de datos

Es necesario poder cambiar de formato a los datos porque depende de que bloque se este usando necesitara los datos de un tipo u otro.

Los conversores *dynamic data* se muestran en la Figura 6.5.

6.3.1.1 Descripción: El convertor de la izquierda permite la conversion del formato *dynamic data type* a los formatos *numeric*, *boolean*, *waveform* y *array*. El convertor de la derecha realiza las conversiones opuestas.

El convertor *array to cluster* se muestra en la Figura 6.6.

6.3.1.2 Descripción: Transforma datos que están en formato *array* en datos que viajan en paralelo, es decir, en formato *cluster* y viceversa. Es importante configurar el tamaño del *cluster* para evitar que labview de error, el tamaño del *cluster* debe ser igual que el numero de columnas del *array*.

El bloque *get waveform components* se muestra en la Figura 6.7.

6.3.1.3 Descripción: Extrae de una señal *waveform* sus distintas componentes, el instante en que se ha producido el disparo de la señal (t_0), el tiempo en segundos entre los datos (dt), el valor de los datos de la señal *waveform* (Y) y los atributos que hay asociados a los datos.

El convertor *to quad integer* se muestra en la Figura 6.8.

6.3.1.4 Descripción: Transforma números reales en números enteros.

ANEXO 1 – ELEMENTOS UTILIZADOS DE LABVIEW



Figura 6.5: Conversor de datos dinámicos.



Figura 6.6: Conversor de array a cluster y viceversa.



Figura 6.7: Extractor-Agrupador de las componentes waveform.

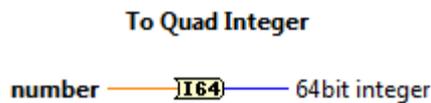


Figura 6.8: Conversor números reales en números enteros (64bits de tamaño).

6.4 Bloques

6.4.1 Tarjeta de adquisición de datos niScope

6.4.1.1 Configuración horizontal de la tarjeta de adquisición.

El bloque *ni scope configure horizontal timing* se muestra en la Figura 6.9.

6.4.1.1.1 Descripción: Configura las propiedades más comunes del subsistema horizontal.

6.4.1.1.2 Terminales de entrada:

- *Enforce realtime*, indica si la adquisición se fuerza a tiempo real o permite adquirir en un tiempo equivalente.
- *Number of records*, especifica el número de registros que se quieren adquirir.
- *Instrument handle*, identifica el hardware al que se quiere ajustar los parámetros horizontales.
- *Min sample rate*, en el esquema el control que ajusta este terminal se llama *Frec. muestreo* (Hz) y controla la frecuencia con la que se quiere adquirir muestras.

ANEXO 1 – ELEMENTOS UTILIZADOS DE LABVIEW

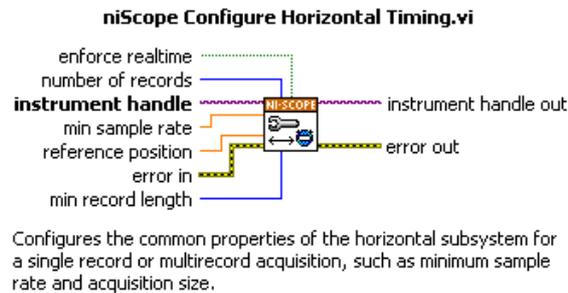


Figura 6.9: Bloque para la configuración horizontal de la tarjeta de adquisición.

- *Reference position*, en el esquema el control que ajusta este terminal se llama *trigger x (%)* y controla el inicio de la adquisición en tanto por ciento del tiempo de muestreo.
- *Min record length*, en el esquema se controla este valor a partir de dos controles, *rango horizontal* y *frecuencia de muestreo*, se realiza el producto entre estos parámetros y se obtiene el número de muestras de la adquisición, que es el valor que se asigna a este terminal.
- *Error in*, este terminal se cablea mediante *clusters* Figura 6.10 que transporta información de los errores que se producen en el sistema. Tienen el mismo formato en todos los bloques de la aplicación. Su principal función es conectar en serie todos los bloques de la aplicación y cuando se produce un fallo, para la ejecución de la aplicación. En nuestro caso se ha optado por colocar un pulsador *stop* para detener la ejecución de la aplicación, obviando el cableado de este terminal.

Componentes del *cluster error*:

- o *Status*, indica si se ha producido o no se ha producido un error en la aplicación.
- o *Code*, si se ha producido un error indica el numero *código* que tiene asignado.
- o *Source*, si se ha producido un error indica donde se ha producido el error.

6.4.1.1.3 Terminales de salida:

- *Instrument handle out*, asigna el hardware al que se quiere ajustar los parámetros horizontales.
- *Error out*, transporta información de los errores que se producen en el sistema.

ANEXO 1 – ELEMENTOS UTILIZADOS DE LABVIEW

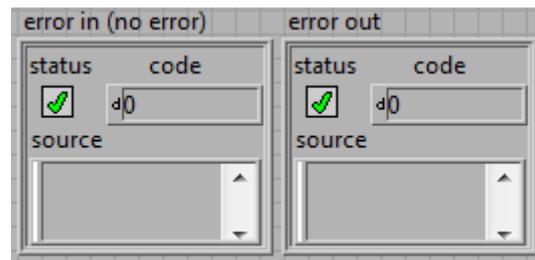


Figura 6.10: Indicador del error.

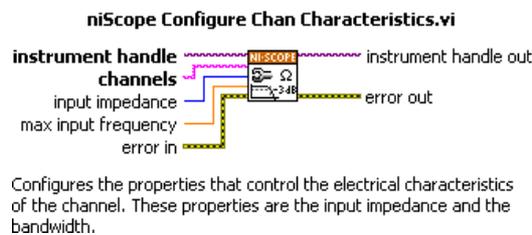


Figura 6.11: Bloque para la configuración de los canales de la tarjeta de adquisición.

6.4.1.2 Configuración de parámetros eléctricos de los canales de la tarjeta de adquisición

El bloque *ni scope configure chan characteristics* se muestra en la Figura 6.11.

6.4.1.2.1 Descripción: Configura los parámetros eléctricos del canal. Estos parámetros son impedancia de entrada del canal y ancho de banda.

6.4.1.2.2 Terminales de entrada:

- *Channels*, especifica el canal desde el cual se están adquiriendo datos.
- *Input impedance*, es la impedancia de entrada del canal.
- *Maximun input frecuency*, especifica el ancho de banda del canal que es atenuada 3dB. El valor por defecto es 0, con este valor se utiliza el ancho de banda que tenga la tarjeta de adquisición asignado. Para el valor -1 consigues todo el ancho de banda.
- *Instrument handle*, identifica el hardware al que se quiere ajustar los parámetros de los canales.
- *Error in*, este terminal tiene las mismas funciones en todos los bloques, su descripción se puede ver en la descripción de los terminales del bloque configuración horizontal de la tarjeta de adquisición.

6.4.1.2.3 Terminales de salida:

- *Instrument handle out*, asigna el hardware al que se quiere ajustar los parámetros de los canales.
- *Error out*, este terminal tiene las mismas funciones en todos los bloques, su descripción se puede ver en la descripción de los terminales del bloque configuración horizontal de la tarjeta de adquisición.

ANEXO 1 – ELEMENTOS UTILIZADOS DE LABVIEW

6.4.1.3 Configuración vertical de la tarjeta de adquisición

El bloque *ni scope configure vertical* se muestra en la Figura 6.12.

6.4.1.3.1 Descripción: Configura las propiedades más comunes del subsistema vertical.

6.4.1.3.2 Terminales de entrada:

- *Channels*, especifica el canal desde el cual se están adquiriendo datos.
- *Vertical range*, especifica el valor absoluto del rango de entrada del canal. Por ejemplo para adquirir una señal comprendida entre 5V y -5V se debe poner de rango vertical 10V. Los límites establecidos para la tarjeta de adquisición PCI-5152 usada en esta aplicación son 100mV y 10V.
- *Vertical offset*, ajusta la localización del centro del rango vertical respecto de tierra.
- *Probe attenuation*, especifica la atenuación del canal de entrada.
- *Vertical coupling*, especifica como adquirir la señal de entrada.
- *Channel enabled*, especifica si el canal está disponible para la adquisición.
- *Instrument handle*, identifica el hardware al que se quiere ajustar los parámetros verticales.
- *Error in*, este terminal tiene las mismas funciones en todos los bloques, su descripción se puede ver en la descripción de los terminales del bloque *configuración horizontal* de la tarjeta de adquisición.

6.4.1.3.3 Terminales de salida:

- *Instrument handle out*, asigna el hardware al que se quiere ajustar los parámetros verticales.
- *Error out*, este terminal tiene las mismas funciones en todos los bloques, su descripción se puede ver en la descripción de los terminales del bloque *configuración horizontal* de la tarjeta de adquisición.

6.4.1.4 Configuración de los parámetros del *trigger*

El bloque *ni scope configure trigger edge* se muestra en la Figura 6.13.

6.4.1.4.1 Descripción: Se puede elegir entre varios tipos de disparo *analógico edge*, *Analógico Histeresis*, *analógico video*, *analógico window*, *digital edge*, *inmediato* y *software*.

En nuestra aplicación se elige el *trigger analógico edge* este tipo de trigger inicializa la adquisición de datos en el momento que se alcanza el valor configurado en *trigger level*.

ANEXO 1 – ELEMENTOS UTILIZADOS DE LABVIEW

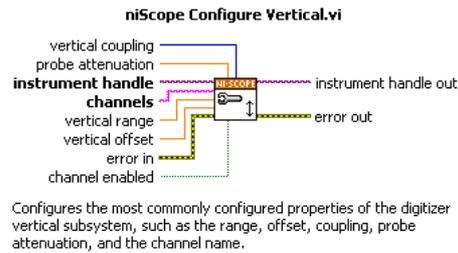


Figura 6.12: Bloque para la configuración vertical de la tarjeta de adquisición.

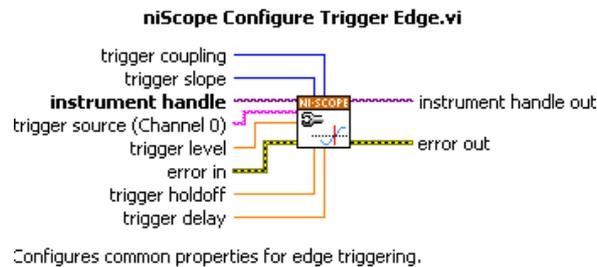


Figura 6.13: Bloque para la configuración del *trigger*.

6.4.1.4.2 Terminales de entrada:

- *Trigger source*, permite elegir en que canal se quiere que se detecte el disparo.
- *Trigger level*, especifica el umbral en voltios a partir del cual se quieren detectar señales. En el esquema, el control que ajusta este terminal se llama *trigger y*.
- *Trigger slope*, selecciona uno de los dos semiplanos (positivo o negativo) donde se quieren detectar los disparos.
- *Trigger coupling*, especifica como acoplar la señal de disparo, tiene varias opciones:
 - o *AC*, el osciloscopio acoplar la señal de disparo a una señal alterna.
 - o *DC*, el osciloscopio acopla la señal de disparo a una señal continua.
 - o *LF reject*, el osciloscopio filtra las bajas frecuencias de la señal a la hora de ejecutar el disparo.
 - o *HF reject*, el osciloscopio filtra las altas frecuencias de la señal a la hora de ejecutar el disparo.
 - o *Noise reject*, el osciloscopio filtra el ruido de la señal a la hora de ejecutar el disparo.
- *Trigger holdoff*, especifica el tiempo que espera el bloque en detectar un disparo después de haber detectado otro.

ANEXO 1 – ELEMENTOS UTILIZADOS DE LABVIEW

- *Trigger delay*, especifica el tiempo de retraso que introduce el bloque desde que se detecta el disparo hasta que se comienza a adquirir datos.
- *Instrument handle*, identifica el hardware al que se quiere ajustar los parámetros del *trigger*.
- *Error in*, este terminal tiene las mismas funciones en todos los bloques, su descripción se puede ver en la descripción de los terminales del bloque *configuración horizontal* de la tarjeta de adquisición.

6.4.1.4.3 Terminales de salida:

- *Instrument handle out*, asigna el hardware al que se quiere ajustar los parámetros del *trigger*.
- *Error out*, este terminal tiene las mismas funciones en todos los bloques, su descripción se puede ver en la descripción de los terminales del bloque *configuración horizontal* de la tarjeta de adquisición.

6.4.1.5 Inicio de la adquisición

El bloque *ni scope initialite acquisition* se muestra en la Figura 6.14.

6.4.1.5.1 Descripción: Da la orden para la adquisición de datos, cuando se ejecuta el instrumento virtual que es nuestra aplicación, este bloque deja de estar inactivo para pasar a un estado de espera hasta que el *trigger* detecte disparo, cuando esto ocurre da la orden para el inicio de la adquisición.

6.4.1.5.2 Terminales de entrada:

- *Instrument handle*, identifica el hardware con el que se quiere iniciar la adquisición.
- *Error in*, este terminal tiene las mismas funciones en todos los bloques, su descripción se puede ver en la descripción de los terminales del bloque *configuración horizontal* de la tarjeta de adquisición.

6.4.1.5.3 Terminales de salida:

- *Instrument handle out*, asigna el hardware con el que se quiere iniciar la adquisición.
- *Error out*, este terminal tiene las mismas funciones en todos los bloques, su descripción se puede ver en la descripción de los terminales del bloque *configuración horizontal* de la tarjeta de adquisición.

ANEXO 1 – ELEMENTOS UTILIZADOS DE LABVIEW



Figura 6.14: Bloque para iniciar la adquisición.

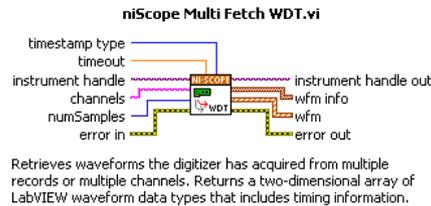


Figura 6.15: Bloque selector entre los distintos formatos de datos.

6.4.1.6 Formato de los datos adquiridos

El bloque *ni scope multi fetch WDT* se muestra en la Figura 6.15.

6.4.1.6.1 Descripción: Adquiere los datos y nos da a elegir el formato de los datos adquiridos, en nuestro caso se ha elegido 1D *waveform*.

6.4.1.6.2 Terminales de entrada:

- *Channels*, especifica el canal del que se quiere obtener los datos.
- *Num samples*, es el número máximo de muestras que se adquirirán de la señal.
- *Timeout*, es el tiempo en segundos que espera el bloque a que los datos sean adquiridos. Colocando -1 timeout es infinito.
- *Timestamp type*, especifica la base de tiempos de los datos adquiridos.
- *Wfm info*, da toda la información de tiempos y escala sobre la señal adquirida.
- *Wfm*, devuelve los datos adquiridos en el formato 1D *waveform*.
- *Instrument handle*, identifica el hardware al que se quiere configurar el formato de los datos adquiridos.
- *Error in*, este terminal tiene las mismas funciones en todos los bloques, su descripción se puede ver en la descripción de los terminales del bloque *configuración horizontal* de la tarjeta de adquisición.

6.4.1.6.3 Terminales de salida:

- *Instrument handle out*, asigna el hardware al que se quiere configurar el formato de los datos adquiridos.

ANEXO 1 – ELEMENTOS UTILIZADOS DE LABVIEW

- *Error out*, este terminal tiene las mismas funciones en todos los bloques, su descripción se puede ver en la descripción de los terminales del bloque *configuración horizontal* de la tarjeta de adquisición.

6.4.1.7 Inicializar niScope

El bloque *ni scope initialize* se muestra en la Figura 6.16.

6.4.1.7.1 Descripción: Crea una sesión de *ni scope* que permite la adquisición de datos con la tarjeta de adquisición seleccionada en *resource name*.

6.4.1.7.2 Terminales de entrada:

- *Resource name*, introduce al bloque el nombre asignado a la tarjeta de adquisición que se haya seleccionado en el control *resource name*.
- *Id query*, verifica que la sesión que se ha iniciado en *ni scope* es compatible con la tarjeta de adquisición que se quiere trabajar.
- *Reset device*, especifica si *resetea* la herramienta durante el proceso de iniciación.
- *Error in*, este terminal tiene las mismas funciones en todos los bloques, su descripción se puede ver en la descripción de los terminales del bloque *configuración horizontal* de la tarjeta de adquisición.

6.4.1.7.3 Terminales de salida:

- *Instrument handle*, genera una denominación al hardware con el que se quiere trabajar durante la sesión.
- *Error out*, este terminal tiene las mismas funciones en todos los bloques, su descripción se puede ver en la descripción de los terminales del bloque *configuración horizontal* de la tarjeta de adquisición.

6.4.1.8 Cerrar niScope

El bloque *ni scope close* se muestra en la Figura 6.17.

6.4.1.8.1 Descripción: Cierra la sesión de adquisición.

6.4.1.8.2 Terminales de entrada:

- *Instrument handle*, identifica el hardware con el que se quiere dejar de adquirir.
- *Error in*, este terminal tiene las mismas funciones en todos los bloques, su descripción se puede ver en la descripción de los terminales del bloque *configuración horizontal* de la tarjeta de adquisición.

ANEXO 1 – ELEMENTOS UTILIZADOS DE LABVIEW

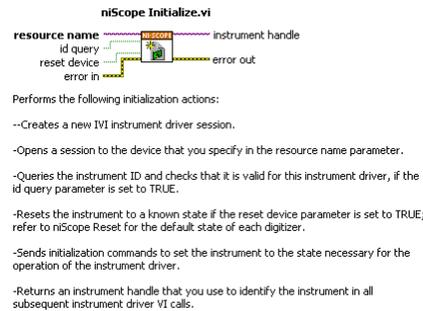


Figura 6.16: Inicializar ni scope.

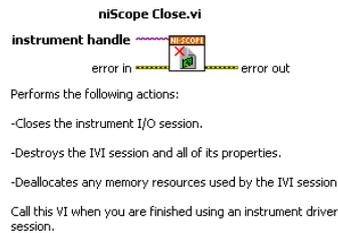


Figura 6.17: Cerrar ni scope.

6.4.1.8.3 Terminales de salida:

- *Error out*, este terminal tiene las mismas funciones en todos los bloques, su descripción se puede ver en la descripción de los terminales del bloque *configuración horizontal* de la tarjeta de adquisición.

6.4.2 Filtro

El bloque *subFilter* se muestra en la Figura 6.18.

6.4.2.1 Descripción: Permite eliminar ciertas frecuencias del espectro eléctrico de la señal de entrada.

6.4.2.1.1 Terminales de entrada:

- *Taps*, especifica el número total de coeficientes de FIR (*finite impulse response*), en nuestro caso no se tendrá que ajustar este parámetro porque nuestra señal es IIR (*infite impulse response*).
- *Filter topology*, designa la topología que se usa el filtro, labview nos da a elegir entre los siguientes tipos: *Butterworth* Figura 6.19, *Chebyshev* Figura 6.20, *Inverse Chebyshev* Figura 6.21, *Elliptic* Figura 6.22y *Bessel* Figura 6.23. En nuestra aplicación se ha usado en los dos filtros la misma tipología, *Bessel*. El usuario podrá seleccionar cualquiera de las opciones.

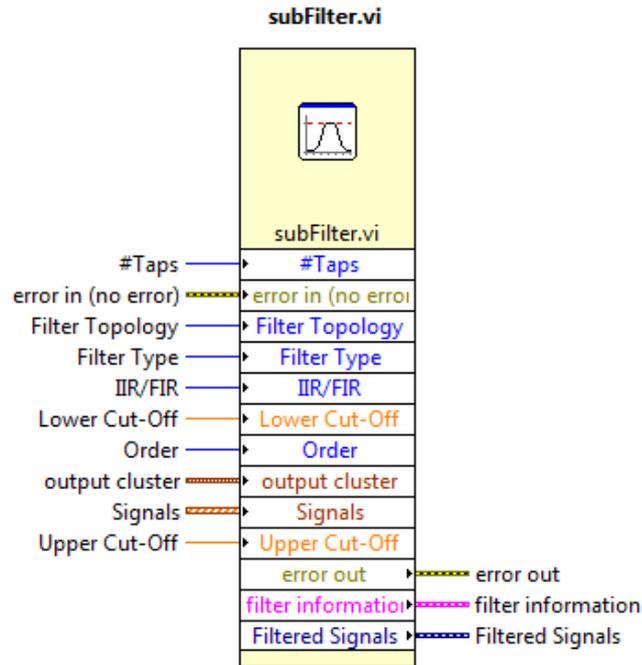


Figura 6.18: Bloque del filtro.

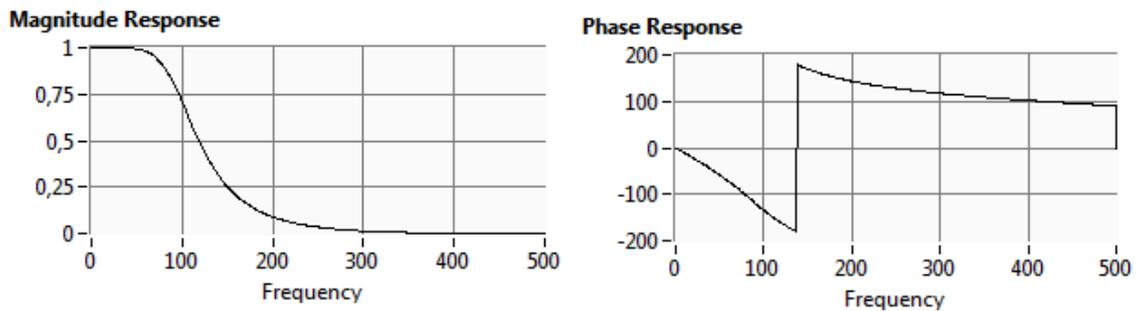


Figura 6.19: Comportamiento de la magnitud y la fase en función de la frecuencia del filtro *butterworth*.

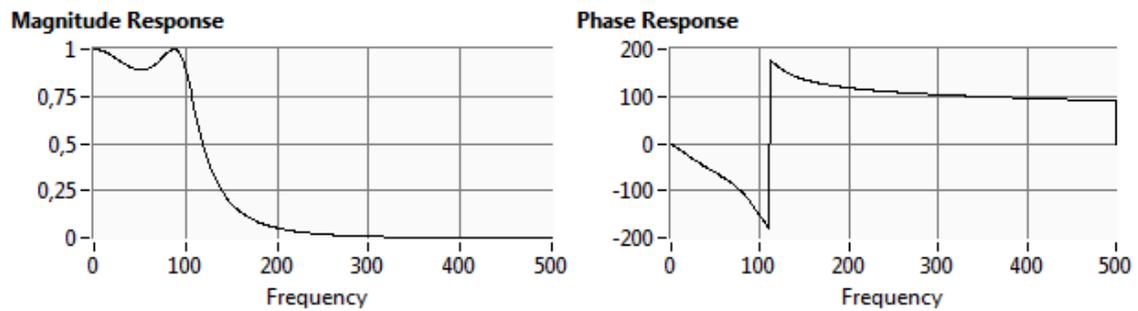


Figura 6.20: Comportamiento de la magnitud y la fase en función de la frecuencia del filtro *chebyshev*.

ANEXO 1 – ELEMENTOS UTILIZADOS DE LABVIEW

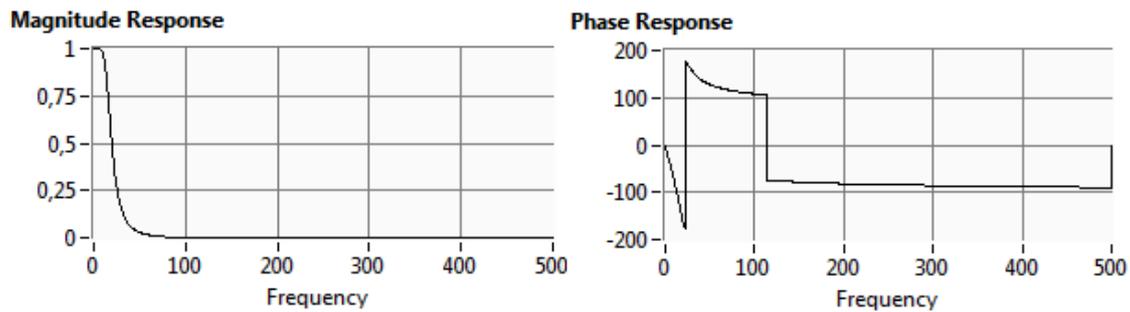


Figura 6.21: Comportamiento de la magnitud y la fase en función de la frecuencia del filtro *inverse chebyshev*.

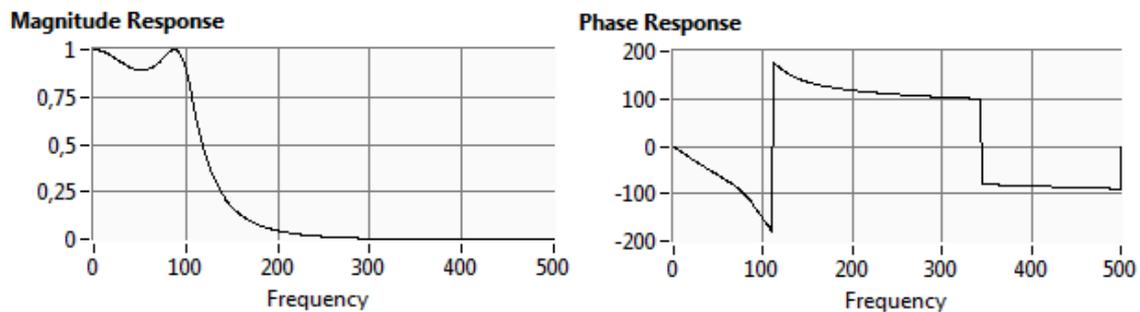


Figura 6.22: Comportamiento de la magnitud y la fase en función de la frecuencia del filtro *elliptic*.

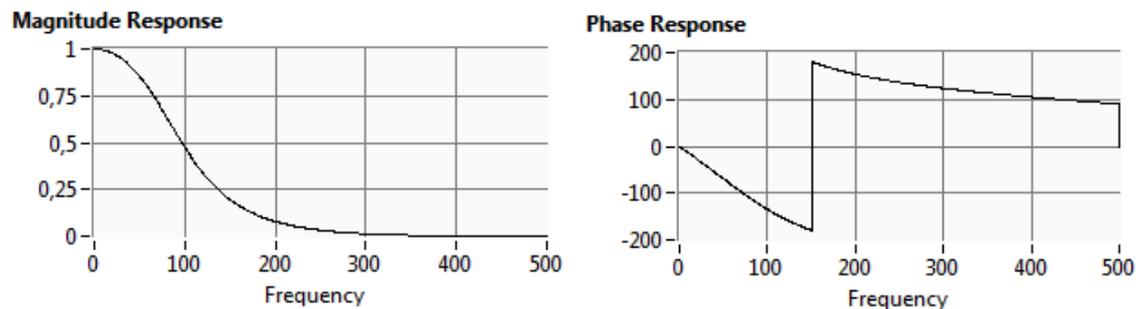


Figura 6.23: Comportamiento de la magnitud y la fase en función de la frecuencia del filtro *bessel*.

- *Filter type*, especifica el tipo de filtro que se quiere utilizar. Labview nos da a elegir entre los siguientes tipos:
 - *Lowpass*: Filtro paso bajo, deja pasar todas las frecuencias que están por debajo de la frecuencia seleccionada.
 - *Highpass*: Filtro paso alto, deja pasar todas las frecuencias que están por encima de la frecuencia seleccionada.
 - *Bandpass*: Filtro paso banda, deja pasar todas las frecuencias que están entre las dos frecuencias seleccionadas.
 - *Bandstop*: Filtro elimina banda, elimina las frecuencias que están entre las dos frecuencias seleccionadas.

ANEXO 1 – ELEMENTOS UTILIZADOS DE LABVIEW

- *Smooth*: Este filtro hace la media entre varias señales para eliminar el ruido que trae la señal, este tipo de filtro no sirve para nuestra aplicación porque cada descarga parcial tiene amplitud distinta y ello implicaría hacer la media entre descargas de distinta amplitud, este hecho no nos permitiría ver las descargas mas grandes ni las más pequeñas.
- *IIR/FIR*, permite elegir entre *finite impulse response* o *infinite impulse response*. En nuestra aplicación se usa *IIR*.
- *Lower cut-off*, cuando se trabaja con un filtro paso bajo o paso alto este terminal especifica la frecuencia de corte, cuando se trabaja con un filtro paso banda o elimina banda este terminal especifica la frecuencia de corte inferior.
- *Order*, especifica el orden del filtro *IIR*.
- *Output cluster*, este terminal solo se usa para el filtro *smooth*.
- *Signals*, este terminal introduce la señal que se quiere filtrar.
- *Upper cut-off*, cuando se trabaja con un filtro paso bajo o paso alto este terminal esta deshabilitado, cuando se trabaja con un filtro paso banda o elimina banda este terminal especifica la frecuencia de corte superior.
- *Error in*, este terminal tiene las mismas funciones en todos los bloques, su descripción se puede ver en la descripción de los terminales del bloque *configuración horizontal* de la tarjeta de adquisición.

6.4.2.1.2 Terminales de salida:

- *Error out*, este terminal tiene las mismas funciones en todos los bloques, su descripción se puede ver en la descripción de los terminales del bloque *configuración horizontal* de la tarjeta de adquisición.
- *Filter information*, devuelve datos sobre la magnitud y la fase del filtro. En nuestra aplicación no se ha cableado este terminal.
- *Filtred signal*, devuelve la señal filtrada.

6.4.3 Integración

El bloque *NI_AALPro.lvlib: integral x (t)* se muestra en la Figura 6.24.

6.4.3.1 Descripción: Realiza la integral discreta de las muestras de la señal x.

6.4.3.2 Terminales de entrada:

- *X*, terminal donde se conecta la señal que se quiere integrar.
- *Initial condition*, Valor inicial, te da la opción de reducir el error en la integral si la muestra anterior a la inicial tiene un valor distinto de 0.

ANEXO 1 – ELEMENTOS UTILIZADOS DE LABVIEW

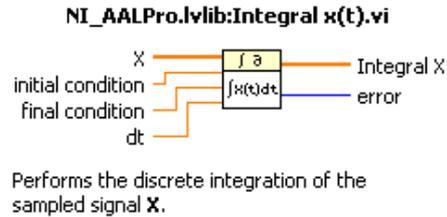


Figura 6.24: Bloque integrador.

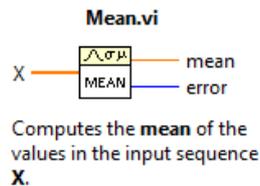


Figura 6.25: Bloque que calcula la media del array de entrada.

- *Final condition*, Valor final, permite cambiar el valor de la muestra n+1 si tiene un valor distinto de 0, de esta manera se reduce el error en la integral
- *dt*, especifica el tiempo que existe entre dos muestras consecutivas.

6.4.3.3 Terminales de salida:

- *Integral x*, devuelve la integral de la señal x.
- *Error*, si se ha producido un error indica el numero *código* que tiene asignado ese error.

6.4.4 Media

El bloque *mean* se muestra en la Figura 6.25.

6.4.4.1 Descripción: Devuelve el valor medio de todos los elementos del *array* de entrada

6.4.4.2 Terminales de entrada:

- *X*, terminal donde se conecta la señal que se quiere calcular.

6.4.4.3 Terminales de salida:

- *Mean*, devuelve la media de la señal x.
- *Error*, si se ha producido un error indica el numero *código* que tiene asignado ese error.

6.4.5 Amplitud and level measurements

El bloque *amplitude and level measurements* se muestra en la Figura 6.26.

ANEXO 1 – ELEMENTOS UTILIZADOS DE LABVIEW

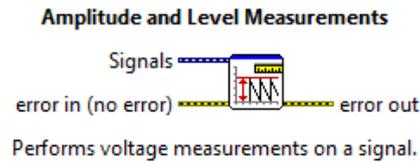


Figura 6.26: Bloque devuelve los parámetros característicos de la señal.

6.4.5.1 Descripción: Extrae de la señal de entrada los parámetros característicos que son: Pico máximo, pico mínimo, valor pico a pico, la media de la señal y el valor eficaz de la señal.

6.4.5.2 Terminales de entrada:

- *Signals*, terminal donde se conecta la señal de la cual se quiere conocer algún parámetro de los arriba indicados.
- *Error in*, este terminal tiene las mismas funciones en todos los bloques, su descripción se puede ver en la descripción de los terminales del bloque *configuración horizontal* de la tarjeta de adquisición.

6.4.5.3 Terminales de salida:

- *Las salidas*, no muestran terminal hasta que se selecciona en las propiedades del bloque algún parámetro.
- *Error out*, este terminal tiene las mismas funciones en todos los bloques, su descripción se puede ver en la descripción de los terminales del bloque *configuración horizontal* de la tarjeta de adquisición.

6.4.6 Fórmula

El bloque *fórmula* se muestra en la Figura 6.27.

6.4.6.1 Descripción: Se introduce una ecuación entrando en sus propiedades, genera tantos terminales de entrada como variables tenga la ecuación. Para generar la ecuación se pueden utilizar casi todas las funciones que se disponen en una básica calculadora científica. Este bloque permite operar con cualquier tipo de datos menos con *string* que son palabras.

6.4.6.2 Terminales de entrada:

- *X1*, terminal donde se conecta la variable X1 (el nombre de las variables se puede cambiar). En el ejemplo de la Figura 6.27: Bloque fórmula. solo hay una variable pero a lo largo de la aplicación hay fórmulas que tienen hasta tres variables, el bloque te permite introducir ocho variables.
- *Error in*, este terminal tiene las mismas funciones en todos los bloques, su descripción se puede ver en la descripción de los terminales del bloque *configuración horizontal* de la tarjeta de adquisición.

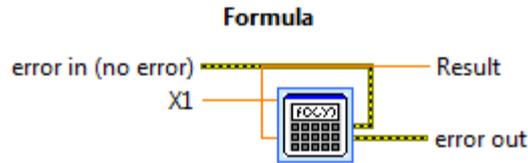


Figura 6.27: Bloque fórmula.

6.4.6.3 Terminales de salida:

- *Result*, devuelve el resultado de la ecuación.
- *Error out*, este terminal tiene las mismas funciones en todos los bloques, su descripción se puede ver en la descripción de los terminales del bloque *configuración horizontal* de la tarjeta de adquisición.

6.4.7 **Creador de array**

El bloque *build array* se muestra en la Figura 6.28.

6.4.7.1 Descripción: Permite crear un *array* a partir de valores numéricos, a partir de otros *arrays* o con la combinación de valores numéricos y *arrays*.

6.4.7.2 Terminales de entrada:

- *Array ó element*, son las entradas del bloque, como se ha indicado en el párrafo anterior las entradas pueden ser *array (array)* o números (*element*).

6.4.7.3 Terminales de salida:

- *Appended array*, es el resultado de la combinación de todas las entradas.

6.4.8 **Localizador de máximo y mínimo en un array**

El bloque *array max & min* se muestra en la Figura 6.29.

6.4.8.1 Descripción: Devuelve el máximo y el mínimo valor encontrado en el *array* deseado, también devuelve índice de la ubicación de estos dos elementos.

6.4.8.2 Terminales de entrada:

- *Array*, es la entrada al bloque, en este terminal se conecta el *array* del que se quiere conocer su máximo o su mínimo.

6.4.8.3 Terminales de salida:

- *Max value*, es el valor del mayor elemento del *array*.
- *Max index*, devuelve la posición del mayor elemento del *array*.
- *Min value*, es el valor del menor elemento del *array*.
- *Min index*, devuelve la posición del menor elemento del *array*.

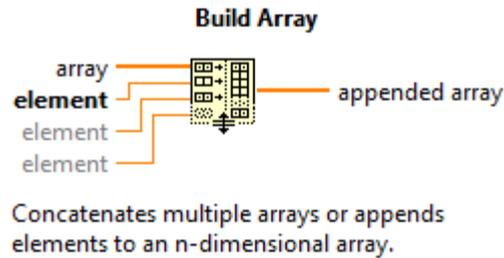


Figura 6.28: Bloque creador de *array*.

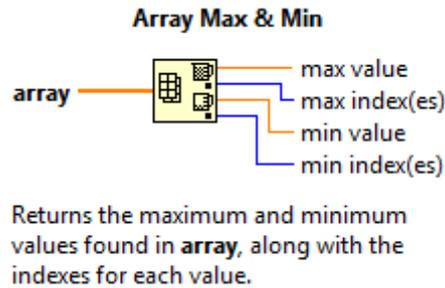


Figura 6.29: Bloque que devuelve el elemento mayor y el menor de un *array*.

6.4.9 Localizador de un elemento en un *array*

El bloque *index array* se muestra en la Figura 6.30.

6.4.9.1 Descripción: Permite extraer uno o varios elementos de un *array* indicando la posición del elemento o elementos que se quiere extraer.

6.4.9.2 Terminales de entrada:

- *N-dimensión array*, a este terminal se cablea el *array* del que se quiere sacar uno o varios elementos.
- *Index*, especifica la posición del elemento que se quiere extraer, se generan tantas entradas de este tipo como números se desee extraer.

6.4.9.3 Terminales de salida:

- *Element of subarray*, devuelve el valor del elemento que ocupa la posición especificada en *index*, se generaran tantas salidas de este tipo como entradas de índice se hayan generado.

6.4.10 Extrae una porción de la señal.

El bloque *extract portion of signal* se muestra en la Figura 6.31.

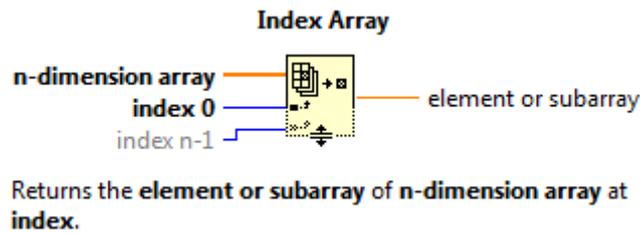


Figura 6.30: Bloque localizador de un elemento en un array.

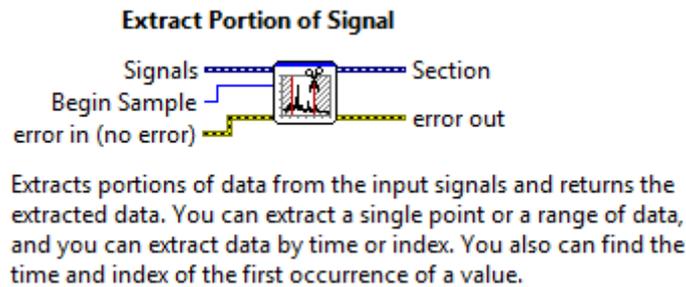


Figura 6.31: Bloque que recorta la señal.

6.4.10.1 Descripción: Extrae una porción de datos de la señal de entrada y devuelve la porción extraída, se puede extraer un único punto o un rango de datos. Los datos se pueden extraer en función del tiempo o de los índices que comprende la porción extraída.

6.4.10.2 Terminales de entrada:

- *Signals*, terminal donde se conecta la señal que se quiere recortar.
- *Begin simple*, especifica el índice a partir del cual se quiere recortar la señal.
- *Error in*, este terminal tiene las mismas funciones en todos los bloques, su descripción se puede ver en la descripción de los terminales del bloque *configuración horizontal* de la tarjeta de adquisición.

6.4.10.3 Terminales de salida:

- *Section*, devuelve la señal recortada.
- *Error out*, este terminal tiene las mismas funciones en todos los bloques, su descripción se puede ver en la descripción de los terminales del bloque *configuración horizontal* de la tarjeta de adquisición.

6.4.11 Puerta lógica Or

El bloque *or* se muestra en la Figura 6.32.

6.4.11.1 Descripción: Calcula la lógica *or* de las dos entradas. Las dos entradas deben ser booleanas, la tabla de la verdad de esta puerta lógica es la Tabla 5.

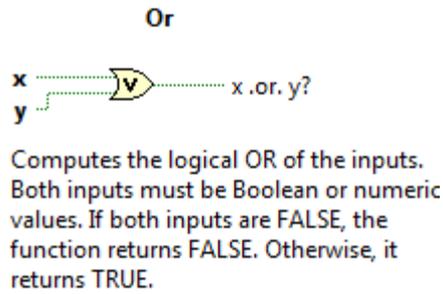


Figura 6.32: Bloque puerta lógica or.

x	y	x.or.y
Falso	Falso	Falso
Falso	Verdadero	Verdadero
Verdadero	Falso	Verdadero
Verdadero	Verdadero	Verdadero

Tabla 5: Tabla de la verdad de la puerta lógica or.

6.4.11.2 Terminales de entrada:

- X, primera entrada de la puerta lógica.
- Y, segunda entrada de la puerta lógica.

6.4.11.3 Terminales de salida:

- X.or.Y, resultado de la puerta lógica.

6.4.12 Sumador

El bloque *add* se muestra en la Figura 6.33.

6.4.12.1 Descripción: Calcula la suma de las dos entradas.

6.4.12.2 Terminales de entrada:

- X, primera entrada del bloque *sumador*.
- Y, segunda entrada del bloque *sumador*.

6.4.12.3 Terminales de salida:

- X.+ Y, resultado del bloque *sumador*.

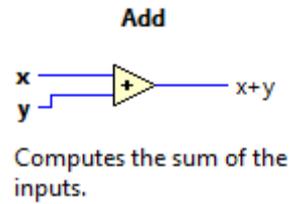


Figura 6.33: Bloque sumador.

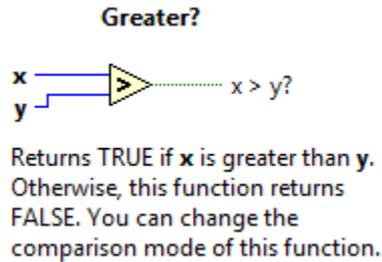


Figura 6.34: Bloque comparador X > Y.

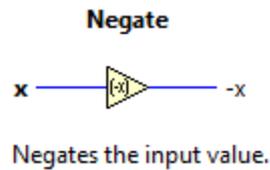


Figura 6.35: Bloque negador.

6.4.13 Comparador

El bloque *greater* se muestra en la Figura 6.34.

6.4.13.1 Descripción: Compara las dos entradas y si *x* es mayor que *y* devuelve *true*, en caso que *x* sea igual o menor que *y* la salida toma el valor *false*.

6.4.13.2 Terminales de entrada:

- *X*, primera entrada del bloque *comparador*.
- *Y*, segunda entrada del bloque *comparador*.

6.4.13.3 Terminales de salida:

- *X > Y*, resultado del bloque *comparador*.

6.4.14 Negador

El bloque *negate* se muestra en la Figura 6.35.

6.4.14.1 Descripción: Este bloque devuelve en su salida el valor opuesto que tiene en su entrada.

6.4.14.2 Terminales de entrada:

- *X*, entrada del bloque *negador*.

6.4.14.3 Terminales de salida:

- X , salida del bloque *negador* con el valor opuesto a la entrada.

6.4.15 **Realimentación**

El bloque *Feedback Node* se muestra en la Figura 6.36.

6.4.15.1 Descripción: Durante el desarrollo de la aplicación fue necesario generar bucles, cuando se realizaba el cableado de los mismos automáticamente labview colocaba este bloque. Su función es memorizar un valor en una sesión, para disponer del en la siguiente.

6.4.15.2 Terminales de salida:

- *In*, terminal que se cablea desde el inicio del bucle.
- *Initial value*, permite asignar un valor inicial para disponer en la primera ejecución, por defecto el valor inicial es 0.

6.4.15.3 Terminales de salida:

- *Out*, terminal que se cablea hacia el final del bucle.

6.4.16 **Detector de picos**

El bloque *peak detector* se muestra en la Figura 6.37.

6.4.16.1.1 Descripción: Este bloque da la opción de detectar picos o valles pero no las dos cosas a la vez, tiene un umbral ajustable para discriminar los picos que estén por debajo de ese valor o para discriminar los valles que estén por encima. Devuelve un *array* con todos los picos o valles detectados y otro *array* con los tiempos donde se han producido los picos o valles.

6.4.16.2 Terminales de entrada:

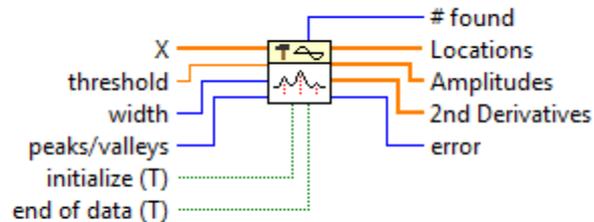
- X , terminal donde se cablea la señal que se quiere analizar.
- *Threshold*, umbral por debajo del cual se discriminan los picos, en caso de usar el bloque en el método valles umbral por encima del cual se discriminan los valles.
- *Width*, determina el numero consecutivo de puntos para usar en la detección de picos o valles con el objetivo de discriminar el ruido.
- *Peaks/valles*, especifica si se quiere detectar picos o valles.
- *Initialize (T)*, especifica si se desea controlar el inicio de la detección.
- *End of data (T)*, si solo se quiere procesar un bloque de datos, se debe dejar sin cablear o con el valor *true*, si se desea procesar sucesivos bloques de datos hay que introducir el valor *false*.

Feedback Node



Figura 6.36: Bloque para la realimentación.

NI_AALPro.lvlib:Peak Detector.vi



Finds the location, amplitude, and second derivative of peaks or valleys in the input signal.

Figura 6.37: Bloque que devuelve los picos o valles de la señal.

6.4.16.3 Terminales de salida:

- *# found*, numero de picos o valles detectados.
- *Locations*, devuelve un *array* con el tiempo donde se han detectado los picos o valles.
- *Amplitudes*, devuelve un *array* con la amplitud de los picos o valles detectados.
- *2nd derivates*, da la segunda derivada de cada pico o cada valle encontrado en cada bloque de datos.
- *Error*, si se ha producido un error indica el numero código que tiene asignado ese error.

6.4.17 **Plantilla de selección**

El bloque *Ajuste calibración y medidas* se muestra en la Figura 6.38.

6.4.17.1 Descripción: Da la opción de superponer indicadores y controles en un área reducida. Su funcionamiento se basa en generar un área en *Front panel* donde esta función genera distintas páginas superpuestas. En la parte superior del área destinada a esta función, se generan distintas pestañas selectoras que te permiten elegir entre las distintas páginas generadas. De esta manera se multiplica el área disponible en *Front panel*.

6.4.17.2 Terminales: Esta función no tiene terminales para conectar, simplemente se deben colocar encima de sus páginas los controles o indicadores que se deseen.

6.4.18 **Property node**

El bloque *Property node* se muestra en la Figura 6.39.

- 6.4.18.1 Descripción: Esta función permite extraer parámetros de los distintos bloques utilizados a lo largo de la aplicación para controlarlos durante la ejecución de la aplicación, de otra manera se debería parar la ejecución del programa y modificar el parámetro deseado desde las propiedades del bloque.

En la parte interior de los recuadros se indica el parámetro que se quiere leer o modificar, en la mayoría de los parámetros te permite conmutar entre lectura y escritura. En la parte exterior al recuadro se indica el bloque al que pertenece el parámetro que se quiere manipular.

- 6.4.18.2 Terminales: Los terminales de este bloque son el parámetro que se quiere modificar que en cada caso será distinto.

6.4.19 **Shift register**

El bloque *shift register* se muestra en la Figura 6.40.

- 6.4.19.1 Descripción: Esta función se utiliza cuando se quiere pasar un valor directamente de una iteración a la siguiente iteración.

6.4.19.2 Terminales de entrada:

- *Entrada*, terminal donde se cablea el valor que se quiere transportar.

6.4.19.3 Terminales de salida:

- *Salida*, devuelve el valor introducido en la entrada.

6.4.20 **Primera ejecución**

El bloque *first call* se muestra en la Figura 6.41.

- 6.4.20.1 Descripción: Indica si la aplicación o una parte de la aplicación se está ejecutando por primera vez. Esta función devuelve *true* solo la primera vez que se la consulte después de apretar el pulsador *run*.

6.4.20.2 Terminales de salida:

- *Salida*, devuelve *true* si es la primera vez que se está ejecutando la aplicación, por el contrario, si no es la primera vez devuelve *false*.

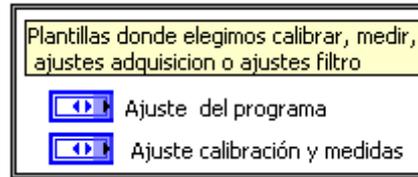


Figura 6.38: Plantillas

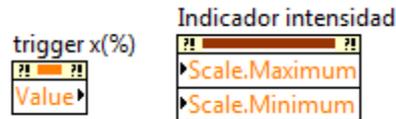


Figura 6.39: Enlaces con parámetros de los componentes de la aplicación.



Figura 6.40: Bloque transporta registro.

First Call?



Indicates that a subVI or section of a block diagram is running for the first time. The First Call? function returns TRUE only the first time you call it after you click the **Run** button.

Figura 6.41: Bloque primera ejecución.

6.4.21 Identificador de estructuras.

El bloque *not a number/path/refnum* se muestra en la Figura 6.42.

6.4.21.1 Descripción: Este bloque analiza la entrada y si detecta que el formato de la entrada no es *number* (Numero), *path* (camino) o *refnum* (referencia numerica) devuelve en la salida *true*, por el contrario si detecta que el formato de la entrada es *number*, *path* o *refnum* devuelve en la salida *false*.

6.4.21.2 Terminales de entrada:

- *Number/path/refnum*, entrada del bloque puede ser un escalar *Number/path/refnum*, o también puede ser un *cluster* o un *array* de *Number/path/refnum*.

6.4.21.3 Terminales de salida:

- *NaN/Path/Refnum*, salida del bloque es un valor booleano de la misma estructura que la entrada.

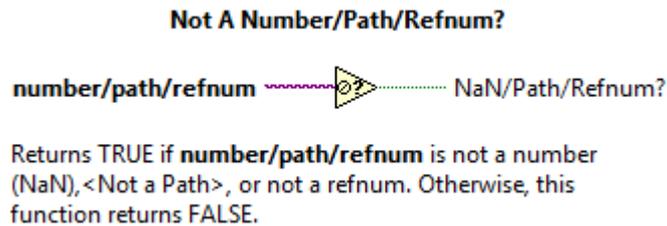


Figura 6.42: Bloque identificador de estructuras.

6.4.22 Agrupador de conductores.

El bloque *merge signals* se muestra en la Figura 6.43.

- 6.4.22.1 Descripción: Este bloque agrupa o separa señales dependiendo de la forma que este cableado.
- 6.4.22.2 Terminales de entrada: *Signal 1, 2, 3...* son las señales que se quiere mezclar. Si trabaja en el sentido contrario estos terminales serian las señales descompuestas de la entrada.
- 6.4.22.3 Terminales de salida: *Combined signal*, devuelve la mezcla de las señales de entrada. Si trabaja en el sentido contrario este terminal seria donde se conecta el cable que conduce las señales mezcladas.

6.5 **Controles:**

6.5.1 **Numeric control**

Los controles *numeric* se muestran en la Figura 6.44.

- 6.5.1.1 Descripción: Estos son dos tipos de controles usados en nuestra aplicación, el primero trabaja con números enteros y te permite seleccionar distintos valores dentro de un rango previamente configurado, el incremento de estos valores también es configurable, el segundo trabaja con números reales asociados a un texto, desde *Front panel* se selecciona en un panel desplegable una de las opciones que te permite elegir, una vez seleccionada una opción la salida del control es el valor asociado a esa opción.

Dentro de la gama *numeric control* existen distintos formatos de controles, a continuación se muestran los utilizados en la aplicación.

- 6.5.1.2 Terminales de salida: *Salida*, devuelve el valor numérico del control.

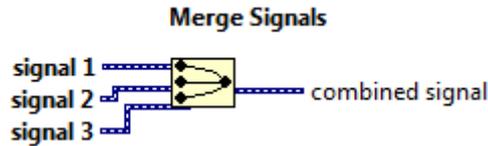


Figura 6.43: Bloque agrupador.

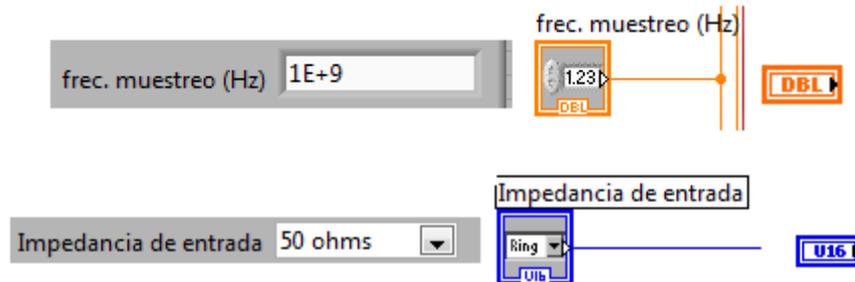


Figura 6.44: Representación grafica de los controles Numeric.

Pointer slider

6.5.2 Deslizador

El control *pointer slider* se muestra en la Figura 6.45.

6.5.2.1 Descripción: Este control trabaja con números enteros y te permite deslizar el selector entre distintos valores dentro de un rango previamente configurado también se puede modificar su valor escribiéndolo directamente en el indicador externo a la barra deslizador, el incremento se configura en las propiedades del controlador.

6.5.2.2 Terminales de salida: *Salida*, devuelve el valor numérico del control.

6.5.3 Pulsador

El control *reset button* se muestra en la Figura 6.46.

6.5.3.1 Descripción: Este control trabaja con valores booleanos, en reposo está en la posición *false* y cuando se presiona cambia a la posición *true*, en el momento que se deja de presionar vuelve a la posición de reposo. En la aplicación está conectado a una estructura condicional que permite realizar acciones distintas en función de la posición donde este. En nuestra aplicación cuando se activa pone a cero los indicadores a los que está asociado.

6.5.3.2 Terminales de salida: *Salida*, devuelve el valor *true/false* dependiendo de la configuración del pulsador y del estado.

ANEXO 1 – ELEMENTOS UTILIZADOS DE LABVIEW

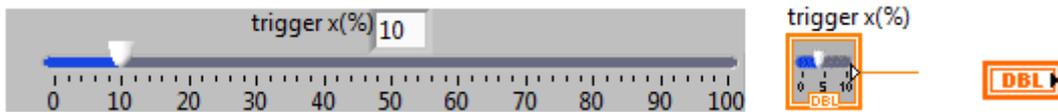


Figura 6.45: Representación grafica del control deslizador.



Figura 6.46: Representación grafica del pulsador reset.



Figura 6.47: Representación grafica del interruptor.

6.5.1 Interruptor

El control *toggle switch* se muestra en la Figura 6.47.

6.5.1.1 Descripción: Este control trabaja con valores booleanos, cuando se presiona sobre el cambia de estado, conmuta entre los valores *true* y *false*. En la aplicación se utiliza para controlar estructuras condicionales.

6.5.1.1.1 Terminales de salida: *Salida*, devuelve el valor *true/false* dependiendo de la configuración del pulsador y del estado.

6.6 Indicadores

6.6.1 Indicador grafico

El *indicador grafico* se muestra en la Figura 6.48.

6.6.1.1 Descripción: Este indicador grafica los datos que son del tipo *Dinamic data*.

6.6.1.2 Terminales de entrada: *Entrada*, terminal donde se conecta la señal que se quiere graficar.

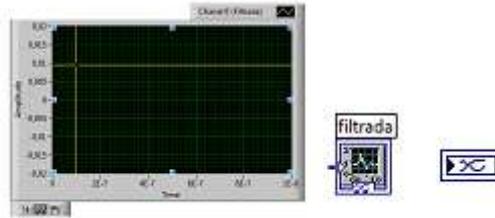


Figura 6.48: Representación grafica del indicador grafico.

6.6.2 Indicador numérico

El *indicador numérico* se muestra en la Figura 6.49.

6.6.2.1 Descripción: Este indicador grafica los datos que son del tipo *numeric*.

6.6.2.2 Terminales de entrada: *Entrada*, terminal donde se conecta el cable *numeric* que se quiere visualizar.

6.6.3 Indicador de aguja

El *indicador de aguja* se muestra en la Figura 6.50.

6.6.3.1 Descripción: Este indicador dispone una aguja que indica la magnitud del dato del tipo *numeric* que recibe. En la aplicación los dos indicadores de este tipo tienen tres agujas, para poder introducir tres valores distintos el indicador modifica su entrada del tipo *scalar numeric* a una entrada tipo *cluster* de tamaño 3, formada por tres *scalar numeric*.

6.6.3.2 Terminales de entrada: *Entrada*, terminal donde se conecta el cable *cluster* con componentes del tipo *numeric*, que se quiere visualizar.

6.6.4 Indicador luminoso

El *indicador luminoso (led)* se muestra en la Figura 6.51.

6.6.4.1 Descripción: Este indicador trabaja con señales booleanas, cuando recibe el valor *true* se enciende y cuando recibe el valor *false* se queda en reposo.

6.6.4.2 Terminales de entrada: *Entrada*, terminal donde se conecta el cable booleano del que se quiere conocer su estado, encendido->*true* y apagado->*false*.

ANEXO 1 – ELEMENTOS UTILIZADOS DE LABVIEW



Figura 6.49: Representación grafica del indicador numérico.

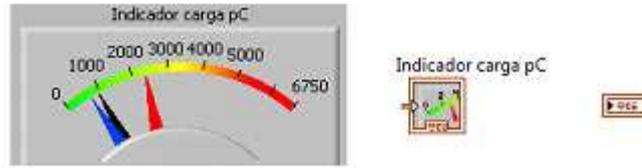


Figura 6.50: Representación grafica del Indicador de aguja.



Figura 6.51: Representación grafica del indicador luminoso.

7 Bibliografía

- [1] F. H. Kreuger. Partial Discharge Detection in High-Voltage Equipment. Butterworths, Londres, 1989.
- [2] Bartnikas y McMahon. Engineering Dielectrics, Volume I. Corona Measurements and Interpretation. ASTM, Philadelphia, 1979.
- [3] R. Tallam, S. B. Lee, G. Stone, G. Kliman, J. Yoo, T. Habetler y R. Harley. A Survey of Methods for Detection of Stator-Related Faults in Induction Machines. Industry Applications, IEEE Transactions on, tomo 43(4): 920_ 933, 2007.
- [4] P. Morshuis. Partial Discharge Mechanisms. Delft University Press, Netherlands, 1993.
- [5] J. Lopez-Roldan, T. Tang y M. Gaskin. Optimisation of a sensor for onsite detection of partial discharges in power transformers by the UHF method. Dielectrics and Electrical Insulation, IEEE Transactions on, tomo 15(6): 1634 1639, 2008.
- [6] G. Paoletti y A. Golubev. Partial discharge theory and technologies related to medium-voltage electrical equipment. Industry Applications, IEEE Transactions on, tomo 37(1): 90 103, 2001.
- [7] H. Okubo, N. Hayakawa y G. Montanari. Technical Development on Partial Discharge Measurement and Electrical Insulation Techniques for Low Voltage Motors Driven by Voltage Inverters. Dielectrics and Electrical Insulation, IEEE Transactions on, tomo 14(6): 1516 1530, 2007.
- [8] S. Whitehead. Dielectric breakdown of solids. Clarendon Press, 1951.
- [9] F. A. C. Martinez. Diagnóstico en línea de motores de gran capacidad la detección de descargas parciales utilizando tecnicas de banda ultra ancha. Proyecto Fin de Carrera, Instituto Politécnico Nacional, 2003.
- [10] I. M. Vásquez y C. V. Baltra. Descargas Parciales en transformadores y máquinas rotatorias. Seminario.
- [11] National Instruments Corporation. Empezando con LabVIEW, Edición de abril 2003