



Informe Técnico

Sistema electrónico para un palpador LVDT

**Benjamín Valera Orozco
Gerardo Antonio Ruiz Botello
Sergio Padilla Olvera**

Mayo2022

Proyecto Interno

Financiamiento interno

Sistema electrónico para un palpador LVDT

Benjamín Valera Orozco, Gerardo Antonio Ruiz Botello y Sergio Padilla Olvera

Instituto de Ciencias Aplicadas y Tecnología
Universidad Nacional Autónoma de México, Circuito Exterior S/N, Ciudad Universitaria
AP 70-186, C.P. 04510, México D.F. México

Resumen

Un palpador LVDT es un transductor de posición utilizado en la medición de longitud de diversos procesos de medición en metrología dimensional. Su uso va desde las mediciones por comparación hasta la obtención de perfiles de rugosidad en donde se requieren resoluciones de fracciones de micrómetros sobre alcances de unos pocos milímetros. En este informe se describe el sistema electrónico desarrollado para la adecuación y procesamiento de las señales en un palpador LVDT que permiten la medición de longitud. Un LVDT, por sus siglas en inglés, Linear Variable Differential Transformer, es un transductor de posición compuesto por un transformador eléctrico que tiene como principio de operación la variación del voltaje inducido en su embobinado secundario cuando en el primario circula una corriente senoidal de amplitud y frecuencia constante. La variación del voltaje inducido es ocasionada por las variaciones de posición en el núcleo que se desplaza conforme a la posición del objeto a medir. En este sentido, el sistema electrónico descrito en el presente informe se encarga de generar una señal senoidal de amplitud y frecuencia constantes así como de procesar la señal del secundario con técnicas de procesamiento digital de señales. El enfoque digital adoptado supera y simplifica las capacidades de los enfoques analógicos tradicionalmente utilizados en los transductores LVDT ya que se pueden implementar algoritmos que incrementan la exactitud de la medición y la corrección de errores. Se describe el sistema electrónico desarrollado sobre la base de un microcontrolador Arduino de 32 bits y el algoritmo de Goertzel para la evaluación eficiente de la Transformada de Fourier, a partir de la cual se extrae la magnitud del voltaje en el secundario. Se presentan los resultados de la calibración del sistema de medición al ser comparado contra un instrumento patrón.

Índice	
Nomenclatura	3
Introducción	4
Definición del problema	4
Entorno actual	5
Descripción del problema a resolver y motivación	6
Relevancia y limitaciones	6
Relación con otras áreas	6
Método	6
Objetivos y resultados esperados	7
Organización del informe	8
1 Sistema electrónico	9
1.1 Descripción general	9
1.2 Descripción del hardware	10
1.2.1 Amplificador del lado primario	10
1.2.2 Amplificador del lado secundario	12
1.3 Descripción del software	14
1.3.1 Síntesis de la señal del primario	15
1.3.2 Algoritmo de Goertzel	17
1.3.3 Producto punto	19
1.4 Procedimiento de operación	20
2 Resultados y discusión	23
2.1 Resultados	23
2.1.1 Prototipo desarrollado	23
2.1.2 Síntesis de la señal en el primario	24
2.1.3 Evaluación en Matlab	26
2.1.4 Calibración	28
2.2 Discusión y trabajo a futuro	30
2.3 Especificaciones del sistema desarrollado	32
2.4 Guía para la detección de fallas	32
Conclusiones	33
Anexo A Diagrama esquemático	34
Anexo B Lista de partes	35
Anexo C Código fuente	36
Anexo D Código en Matlab	41
Referencias bibliográficas	42

Nomenclatura

Símbolo, término o abreviatura	Significado
DAC	Digital to Analog Converter, Convertidor digital a analógico.
ADC	Analog to Digital Converter, Convertidor analógico a digital.
ISA ARM	Instruction Set Architecture, Advanced RISC Machine. Es un modelo abstracto que define como el software controla un procesador.
LVDT	Linear Variable Differential Transformer. Transformador lineal variable diferencial. Es un transductor de posición.
IDE	Integrated Development Environment. Ambiente Integrado de Desarrollo. Son plataformas que permiten agrupar herramientas de desarrollo.
USB	Universal Serial Bus. Conector Universal Serie. Es un medio para conectar computadoras y periféricos.
TC	Timer Counter. Temporizador Contador. Dispositivo electrónico que registra eventos en el tiempo.
FFT	Fast Fourier Transform. Transformada Rápida de Fourier. Es una operación matemática.

Introducción

Definición del problema

Un palpador LVDT es un transductor de desplazamiento de gran resolución, aunque de alcance limitado. Se utiliza principalmente para realizar mediciones por comparación a partir de referencias de planitud. Con este método es posible determinar pequeñas variaciones longitudinales al comparar objetos bajo inspección contra objetos patrón. Comúnmente, los LVDT tienen resoluciones de unos pocos micrómetros y alcances de medición de unos pocos milímetros. La figura 1 muestra un LVDT y las conexiones eléctricas para su funcionamiento.

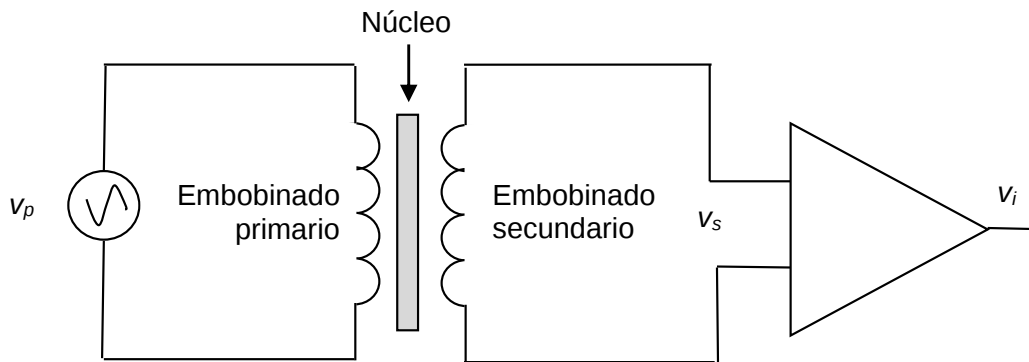


Figura 1. Transductor LVDT.

En la figura 1, el LVDT está formado por los dos embobinados, primario y secundario, y por el núcleo desplazable. El embobinado primario del LVDT se conecta a una fuente de voltaje alterno senoidal de amplitud, frecuencia y fase constantes, V_p . Este voltaje será inducido en el embobinado secundario y la magnitud inducida dependerá de la posición del núcleo. En este transductor la posición del núcleo está en contacto físico con la geometría del objeto a medir, de tal manera que la magnitud dimensional del objeto a medir está relacionada directamente con la magnitud del voltaje inducido V_s . Debido a que la magnitud del voltaje inducido, V_s , generalmente es pequeña, dicho voltaje, V_i , debe ser adecuado para desplegar una lectura apropiada para el usuario.

En la generación de la señal del primario y adecuación de la señal en el secundario existen dos enfoques: analógico y digital. El enfoque analógico es altamente vulnerable a errores y escasas posibilidades de ajustes (Poley, 2003). Por otra parte, el enfoque digital ha demostrado un buen desempeño en cuanto a la facilidad de implementar ajustes y correcciones por software, aunque sigue dependiendo de hardware analógico en algunas etapas de procesamiento de las señales.

En este informe se describe el hardware y software desarrollados para un palpador LVDT que adopta un esquema fuertemente digital de procesamiento, aunque algunas etapas siguen implementándose con el apoyo de electrónica analógica.

Entorno actual

En el Laboratorio de Metrología del Instituto de Ciencias Aplicadas y Tecnología de la UNAM, ICAT UNAM, se cuenta con diversos palpadores que son utilizados en los servicios de metrología dimensional que ofrece nuestra institución tanto a usuarios internos como externos. Entre estos palpadores se encuentran los palpadores electrónicos del tipo LVDT que, debido a su relevancia relacionada con su desempeño superior, son los que mayor atención requieren ya que sus interfaces o consolas de operación han sido rebasadas por la tecnología actual. En este sentido, notamos un caso de obsolescencia relacionado con las consolas de operación, pero se mantiene la exactitud y vigencia de los transductores LVDT.

En particular, el palpador de la máquina analizadora de superficie Federal 4000 de la figura 2, se encuentra en la situación mencionada anteriormente, ya que se ha comprobado su desempeño aceptable a pesar del deterioro de su consola de operación.



Figura 2. Palpador Federal.

Recientemente, en el año de 2021, se realizó un esfuerzo para rehabilitar la máquina analizadora de superficies Federal 4000 obteniendo un excelente desempeño (Valera y Ruiz, 2019). A pesar de los resultados obtenidos, la desventaja radica en la dependencia de tarjetas de adquisición comerciales de alto precio y de la complicada instalación de software de terceros. En nuestro concepto, el enfoque adoptado en el presente informe reduce esta dependencia innecesaria ya que la tecnología con base en sistemas abiertos lo permite.

Descripción del problema a resolver y motivación

El problema para resolver entonces consiste en desarrollar un sistema electrónico, hardware y software, para habilitar al palpador de la figura 2 en la medición de desplazamiento utilizando exclusivamente herramientas de desarrollo libres como las ofrecidas por la compañía Arduino (Arduino, 2019). Al utilizar este enfoque no solo se resuelve el problema de instrumentar un transductor LVDT, también se participa en la gran comunidad de desarrolladores que cada vez utilizan más la plataforma Arduino para implementar una enorme variedad de aplicaciones.

A esto habría que sumarle el interés profesional de los integrantes del Laboratorio de Metrología por desarrollar y experimentar con técnicas de procesamiento digital de señales que incrementan el desempeño de los sistemas de medición electrónicos.

Relevancia y limitaciones

En primer lugar, la principal relevancia del proyecto reportado en este informe consiste en fomentar la creación de infraestructura de desarrollo propio que puede ser ofrecida como una transferencia tecnológica y como ejemplo de aplicación a la comunidad de desarrolladores Arduino.

El alcance del presente proyecto es el desarrollo de un prototipo funcional que pueda ser validado al compararlo con patrones dimensionales.

La potencial limitación del prototipo radica en el probable deterioro del LVDT ocasionado por el desgaste en el transcurso del tiempo (30 años). Otra limitación importante es la falta de información confiable del propio palpador LVDT marca Federal.

Relación con otras áreas

Existe una gran relación con las áreas de procesamiento digital de señales, metrología dimensional y microcontroladores Arduino.

Método

En este informe se describe el desarrollo hardware y software para instrumentar el palpador LVDT del perfilómetro Federal 4000. El esquema general se presenta en la figura 3.

En la figura 3, en el ambiente de desarrollo integrado IDE Arduino se programan, en lenguaje C, los algoritmos de procesamiento digital de señales para la generación y medición que se decidieron ser los más apropiados para la presente aplicación. La selección de los algoritmos más apropiados se realizó con base en comparaciones numéricas y de tiempo de ejecución. En primer lugar, para el algoritmo de generación de la señal del primario, el microcontrolador Arduino sintetiza una señal senoidal de amplitud, frecuencia y fase constantes en su circuito DAC que después será amplificada para impulsar el embobinado primario del LVDT, v_p . En segundo lugar, para el algoritmo

de medición se registra en el circuito ADC el voltaje amplificado proveniente del embobinado secundario del LVDT, v_s , para correlacionarlo con la magnitud física del desplazamiento en el núcleo del LVDT. La correlación entre la amplitud de voltaje y el desplazamiento longitudinal se consigue al realizar un proceso de calibración en donde se obtiene la curva voltaje vs desplazamiento y la ecuación matemática que los relaciona.

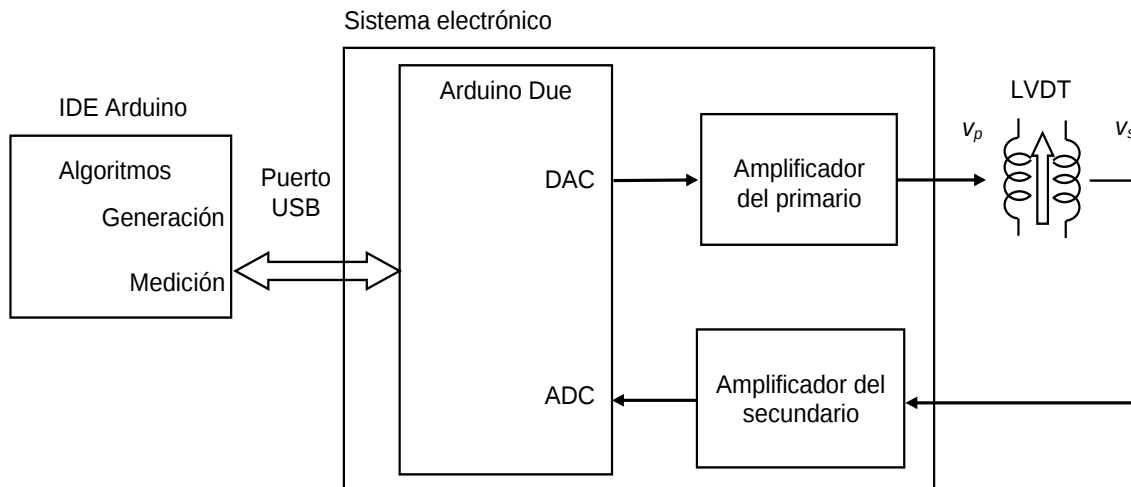


Figura 3. Método por emplear.

La conexión USB entre la computadora y el microcontrolador se utiliza para grabar los algoritmos en la memoria del microcontrolador y además permite el envío de las mediciones realizadas en el microcontrolador hacia la pantalla de la computadora.

Objetivos y resultados esperados

El objetivo del presente informe es mantener documentado el desarrollo tecnológico que aquí se describe de forma que sea de utilidad para su reproducibilidad o como referencia para desarrollos posteriores.

Por otra parte, el objetivo general del proyecto es desarrollar infraestructura para mantener actualizados los equipos de nuestro laboratorio.

Los siguientes son los objetivos particulares en el desarrollo del sistema electrónico:

- Desarrollar y construir un sistema electrónico con base en herramientas básicas de dominio libre para instrumentar un transductor LVDT.
- Obtener una resolución mínima de $0.1 \mu\text{m}$ la cual es común encontrar entre este tipo de transductores.
- Obtener un alcance de medición entre $-800 \mu\text{m}$ a $800 \mu\text{m}$ que también resulta ser un parámetro común para estos transductores.

Organización del informe

La sección de introducción plantea el problema, el método empleado para resolverlo y expone los resultados esperados. En el capítulo uno se describe el desarrollo del sistema electrónico en detalle que incluye el hardware y los algoritmos que ejecuta el microcontrolador, así como una descripción del procedimiento de medición propuesto para operar el sistema desarrollado. El capítulo dos describe los resultados obtenidos con el sistema desarrollado y su comparación contra un patrón dimensional. Finalmente, los anexos contienen información detallada para la reproducibilidad del presente desarrollo como lo son diagramas, lista de partes, código fuente y código auxiliar utilizado en la depuración de los algoritmos.

1 Sistema electrónico

El sistema electrónico que se describe en este informe está construido sobre la base del microcontrolador Arduino Due que emplea la arquitectura ISA ARM de 32 bits.

1.1 Descripción general

La figura 4 muestra el esquema general del sistema electrónico desarrollado y sus conexiones hacia una computadora y al transductor LVDT.

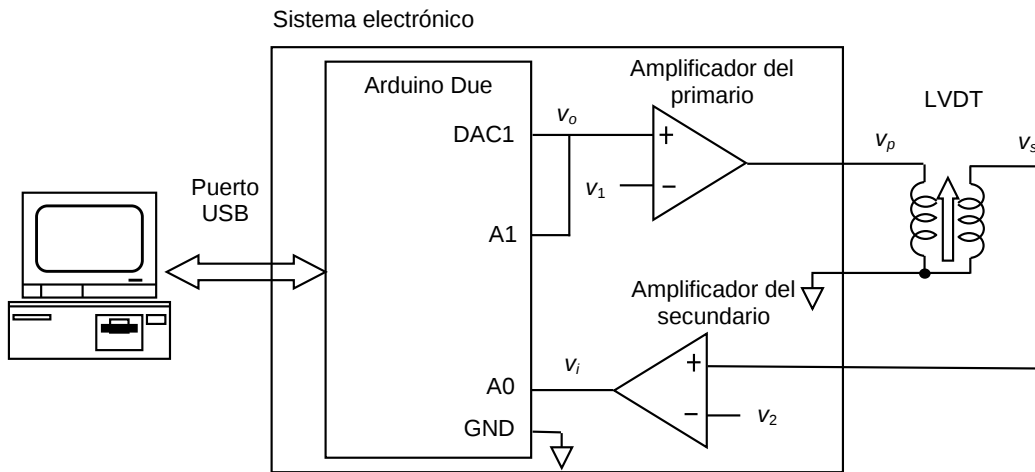


Figura 4. Esquema general.

Con referencia a la figura 4, el software en el procesador Arduino es el encargado de ejecutar los algoritmos para generar la señal del primario y registrar la señal del secundario del LVDT. En primer lugar, para el algoritmo de generación de la señal del primario, primero se pre-calculan 64 muestras por periodo de una señal senoidal para posteriormente enviar esta información al convertidor digital analógico DAC1 del microcontrolador Arduino Due. Estas muestras son salida por medio de un temporizador a una tasa de 1.6×10^6 muestras por segundo, lo cual resulta en una señal senoidal v_o de 25 kHz. En segundo lugar, para el algoritmo de medición se registran tanto la señal generada v_o en A1 como la señal amplificada del secundario v_i en A0. A partir de estas señales digitales se ejecuta el algoritmo de medición para la determinación de la magnitud y signo de esta.

El microcontrolador Arduino Due es la plataforma de procesamiento de señales seleccionada debido a su capacidad de tratamiento en 32 bits y convertidores analógico a digital y digital a analógico de 12 bits.

La adecuación analógica de las señales que manipula el microcontrolador se lleva a cabo con circuitos integrados analógicos del tipo amplificador operacional en las etapas de amplificación del primario y del secundario. El amplificador del primario adecua la

excursión analógica del convertidor digital a analógico en v_o (0 a 3 V) a los requerimientos de voltaje y corriente en el embobinado primario del LVDT en v_p (-1 V a 1 V) mediante un factor de amplificación y una componente de directa en v_1 . Por otra parte, el amplificador del secundario adecua la excursión analógica máxima del embobinado secundario en v_s (-0.5 V a 0.5 V) a los voltajes apropiados del convertidor analógico a digital del microcontrolador en v_i (0 a 3 V) mediante un factor de amplificación y una componente de directa en v_2 .

La conexión USB es utilizada en el ambiente integrado de desarrollo del Arduino para grabar los algoritmos de generación y medición, así como para recibir en la computadora los resultados de la medición.

1.2 Descripción del hardware

El sistema electrónico de la figura 4 cuenta con dos etapas de amplificación que además adecuan las señales analógicas a los niveles de excursión analógica y componentes de directa que manejan tanto el LVDT como los circuitos convertidores en el microcontrolador Arduino Due. Las dos etapas de amplificación están implementadas sobre la base del amplificador operacional en su configuración restadora, mostrada en la figura 5.

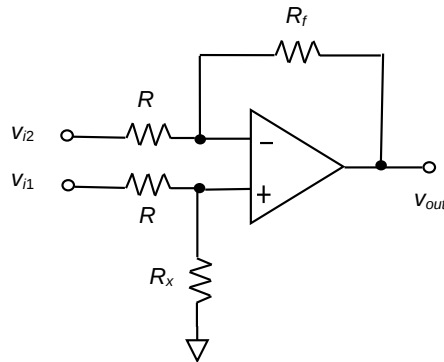


Figura 5. Amplificador operacional restador.

En donde (Coughlin y Driscoll, 2000):

$$v_{out} = \frac{R_x(R + R_f)}{R(R + R_x)} v_{i1} - \frac{R_f}{R} v_{i2} \quad (1)$$

1.2.1 Amplificador del lado primario

Para el amplificador del lado primario se tiene un amplificador restador con los siguientes datos de entrada:

$v_{out} = v_p = 1$ V el voltaje máximo para el primario del LVDT

$v_{i1} = v_o = 3$ V el voltaje máximo a la salida del DAC1

Proponiendo:

$$R = R_f = R_x = 10 \text{ k}\Omega$$

Y sustituyendo en la ecuación (1)

$$1 = \frac{10 \text{ k}(10 \text{ k} + 10 \text{ k})}{10 \text{ k}(10 \text{ k} + 10 \text{ k})} 3 - \frac{10 \text{ k}}{10 \text{ k}} v_{i2}$$

Despejando v_{i2}

$$v_{i2} = 2 \text{ V}$$

Este voltaje es la componente de directa en la entrada v_{i2} del amplificador restador, el cual corresponde al voltaje v_1 de la figura 4. Este voltaje de directa se genera con el divisor de voltaje de la figura 6.

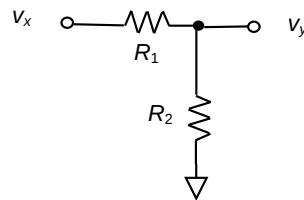


Figura 6. Divisor de voltaje.

La fórmula ampliamente conocida para el divisor de voltaje de la figura 6 es (Hayt, 2019):

$$v_y = \frac{R_2}{(R_1 + R_2)} v_x \quad (2)$$

En la ecuación (2) haciendo:

$$v_y = v_{i2} = v_1 = 2 \text{ V}$$

$$v_x = 5 \text{ V Voltaje de polarización}$$

$$R_1 = 1 \text{ k}\Omega$$

Se obtiene:

$$2 = \frac{R_2}{(1 \text{ k} + R_2)} 5$$

$$R_2 = 666 \Omega$$

Por lo tanto, el circuito para la etapa de amplificación en el primario está formado por un amplificador restador y un divisor de voltaje como se muestra en la figura 7.

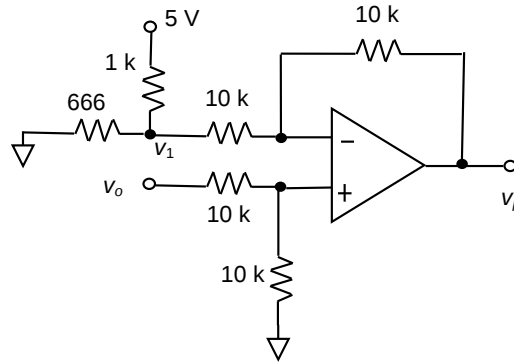


Figura 7. Amplificador del primario.

El análisis realizado en esta sección es una primera aproximación a la implementación real. Los anexos A y B muestran el diagrama esquemático y la lista de partes con los valores ajustados de las resistencias que proporcionaron un mejor desempeño.

1.2.2 Amplificador del lado secundario

El amplificador del lado secundario esta sobre la base del amplificador restador de la figura 5. En esta configuración se propone:

$$R = R_x = 10 \text{ k}\Omega$$

Para este amplificador se requiere que la excursión analógica máxima del LVDT (0.5 V a -0.5 V) en v_s cubra el rango dinámico del ADC del Arduino Due (0 a 3 V) en v_i . Sustituyendo:

$$v_{i1} = v_s = (0.5 \text{ V}, -0.5 \text{ V})$$

$$v_{out} = v_i = (0, 3 \text{ V})$$

En la ecuación (1) se forma el siguiente sistema de dos ecuaciones con dos incógnitas, R_f y v_{i2} .

$$\begin{cases} 3 = \frac{10 \text{ k}(10 \text{ k} + R_f)}{10 \text{ k}(10 \text{ k} + 10 \text{ k})} (0.5) - \frac{R_f}{10 \text{ k}} v_{i2} \\ 0 = \frac{10 \text{ k}(10 \text{ k} + R_f)}{10 \text{ k}(10 \text{ k} + 10 \text{ k})} (-0.5) - \frac{R_f}{10 \text{ k}} v_{i2} \end{cases}$$

Simplificando

$$\begin{cases} 110 \text{ k} = R_f - 4R_f v_{i2} \\ 10 \text{ k} = -R_f - 4R_f v_{i2} \end{cases} \quad (3)$$

Despejando de la primera ecuación de (3) a R_f

$$R_f = \frac{110 \text{ k}}{1 - 4v_{i2}} \quad (4)$$

Sustituyendo la ecuación (4) en la segunda ecuación de (3)

$$10 \text{ k} = -\frac{110 \text{ k}}{1 - 4v_{i2}} - 4\frac{110 \text{ k}}{1 - 4v_{i2}} v_{i2}$$

Despejando v_{i2}

$$v_{i2} = -0.3 \text{ V} \quad (5)$$

La ecuación (5) resulta ser la componente de directa en el amplificador del secundario v_2 , como se muestra en la figura 4

$$v_2 = v_{i2} = -0.3 \text{ V}$$

Este voltaje se genera con el divisor de voltaje de la figura 6, en donde, sustituyendo en su ecuación (2) lo siguiente:

$$v_y = v_2 = v_{i2} = -0.3 \text{ V}$$

$$v_x = -5 \text{ V (Voltaje de polarización)}$$

$$R_1 = 5 \text{ k}\Omega$$

Se obtiene:

$$-0.3 = \frac{R_2}{(5 \text{ k} + R_2)} (-5)$$

$$R_2 = 319 \Omega$$

Sustituyendo la ecuación (5) en (4)

$$R_f = \frac{110 \text{ k}}{1 - 4(-0.3)}$$

$$R_f = 50 \text{ k}\Omega$$

Entonces el circuito para la etapa de amplificación en el secundario está formado por un amplificador restador y un divisor de voltaje como se muestra en la figura 8.

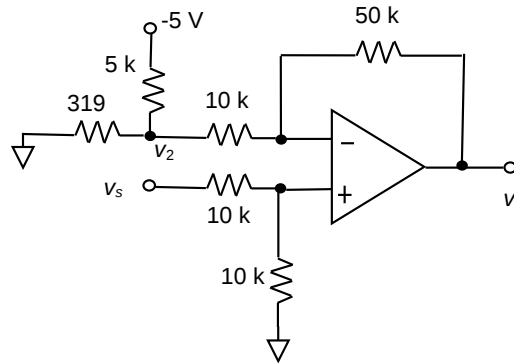


Figura 8. Amplificador del secundario.

El análisis realizado en esta sección es una primera aproximación a la implementación real. Los anexos A y B muestran el diagrama esquemático y la lista de partes con los valores ajustados de las resistencias que proporcionaron un mejor desempeño.

1.3 Descripción del software

El software descrito en la presente sección está conformado por los algoritmos para generar la señal senoidal en el primario del LVDT y para procesar la señal del secundario de este. El procesamiento de la señal en el secundario incluye el algoritmo de Goertzel para la determinación de la magnitud de la medición y el algoritmo del producto punto para determinar el signo de la medición. Estos algoritmos están escritos en lenguaje C de programación y son ejecutados por el microcontrolador Arduino Due, como se muestra en la figura 9.

En la figura 9 los tres algoritmos principales del software son:

1. Síntesis de la señal del primario.
2. Algoritmo de Goertzel.
3. Producto punto.

La síntesis de la señal del primario se ejecuta en el Arduino Due sobre la base de un esquema de interrupciones del temporizador y valores pre-calculados de un periodo de una señal senoidal. Con el algoritmo de Goertzel se calculan las componentes real e imaginaria del espectro de frecuencias a 25 kHz, “ykr” e “yki” de las cuales se obtiene la magnitud en la variable “acumuladorValor”. El producto punto es la operación matemática que permite obtener el signo de la medición en la variable “acumuladorSigno” a través de la interpretación del ángulo entre las señales v_0 y v_i . Los resultados de la medición del LVDT son los promedios de 150 repeticiones del proceso que se almacenan en las variables “promedioValor” y “promedioSigno” que son enviadas por puerto serie a la computadora huésped para su despliegue.

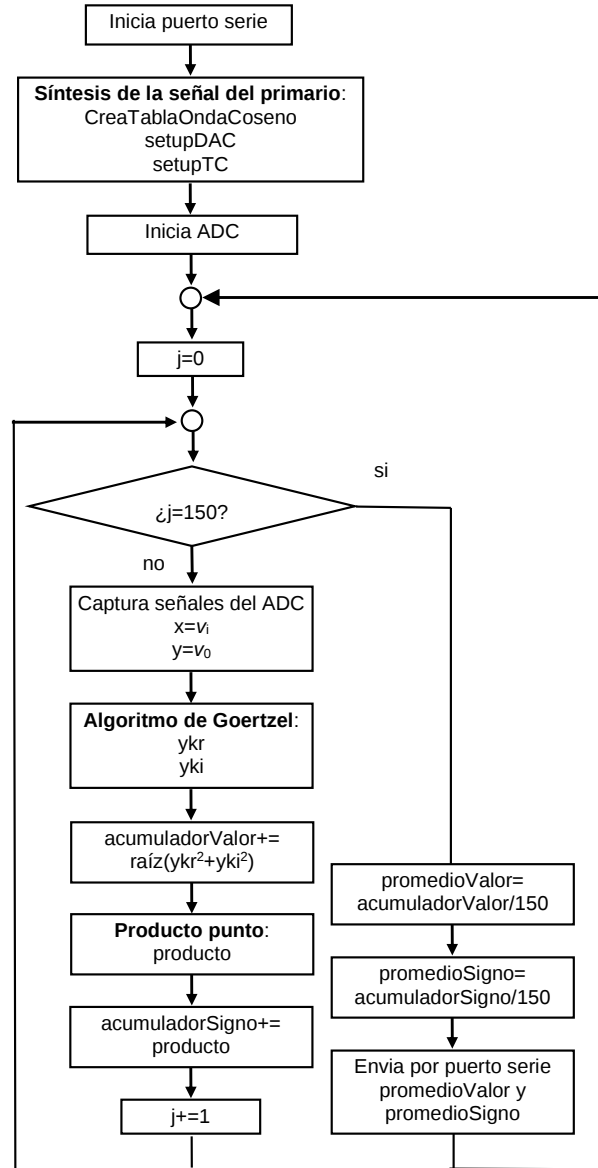


Figura 9. Diagrama de flujo del software en el Arduino Due.

1.3.1 Síntesis de la señal del primario

Para sintetizar una señal analógica a partir de datos digitales primero se pre-calculan 64 valores de un periodo de una señal coseno mediante la fórmula de la ecuación (6).

$$x(k) = 2^{n-1} \cos\left(\frac{2\pi}{k/N}\right) + 2^{n-1} \quad (6)$$

En donde:

$$k = 0, 1, 2, \dots, N-1$$

$N = 64$ muestras en un periodo
 $n = 12$ bits de resolución del DAC del Arduino Due

La figura 10 muestra los valores calculados y graficados a partir de la ecuación (6) en Excel.

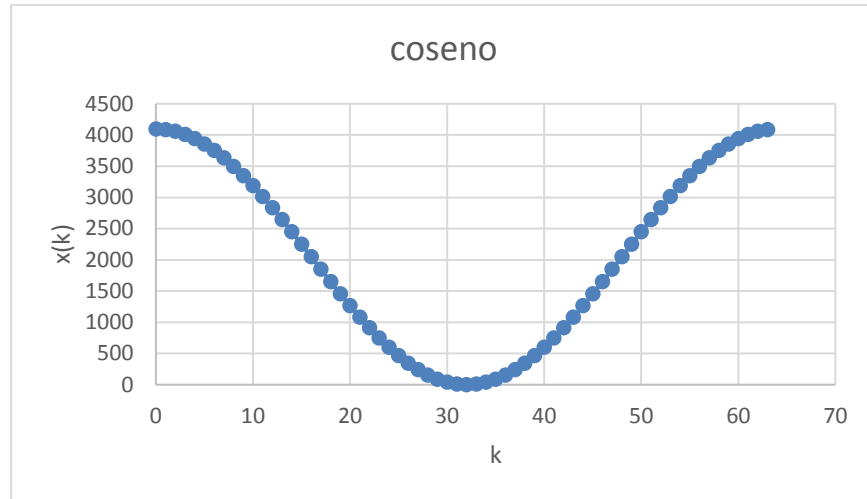


Figura 10. Valores pre-calculados para un periodo de la señal coseno.

Los datos de la figura 10 cubren el rango dinámico del convertidor digital analógico (0 a 3 V) en 12 bits (0 a 4095). La función que crea por única ocasión, al inicio del programa, la tabla de valores de la función coseno es “CreaTablaOndaCoseno”, del anexo C. Los 64 valores de la figura 9 son almacenados en el código fuente mediante un arreglo de enteros de 16 bits cada uno en la variable “wave[64]”.

Posteriormente se configura, en el Arduino Due, la inicialización estándar del convertidor digital a analógico mediante la función “SetupDAC”. Si bien no existe una información clara de cómo llevar a cabo este proceso, la función “SetupDAC” se obtuvo del foro de desarrolladores Arduino (Kyle, 2019). La función se muestra en el anexo C.

Después se configura la interrupción por comparación del temporizador TC0 en el canal 0, “SetupTC”, para establecer una frecuencia de 25 kHz en la señal periódica v_o a partir del contador de pulsos necesarios para generar la interrupción, en la variable “pasos” del código fuente.

$$\text{pasos} = \frac{\frac{\text{Frecuencia del reloj}}{\text{Frecuencia de la señal periodica}}}{\text{Muestras por periodo}}$$

En donde:

$$\text{Frecuencia del reloj} = 42 \text{ MHz} = 42 \times 10^6 \frac{\text{pulsos}}{\text{s}}$$

Frecuencia de la señal periódica = 25 kHz = $25 \times 10^3 \frac{\text{periodos}}{\text{s}}$

Muestras por periodo = $64 \frac{\text{muestras}}{\text{periodo}} = 64 \frac{\text{interrupciones}}{\text{periodo}}$

Sustituyendo

$$\text{pasos} = \frac{\frac{42 \times 10^6 \frac{\text{pulsos}}{\text{s}}}{25 \times 10^3 \frac{\text{periodos}}{\text{s}}}}{64 \frac{\text{interrupciones}}{\text{periodo}}} = 26 \frac{\text{pulsos}}{\text{interrupcion}}$$

De esta manera, el contador TCO del canal 0 generará una interrupción cada 26 pulsos del reloj maestro, es decir cada $26 \times (1/42 \text{ MHz}) = 619 \text{ ns}$, en donde el servicio de atención a la interrupción “DACC_Handler” enviará una muestra “wave[i]” al DAC1. En total, 64 muestras cada 619 ns completan una señal periódica de $1/(64 \times 619 \text{ ns}) = 25 \text{ kHz}$.

1.3.2 Algoritmo de Goertzel

El algoritmo de Goertzel es un filtro digital que permite el cálculo eficiente de la Transformada de Fourier para una frecuencia específica, sin la necesidad de realizar cálculos para todo el contenido espectral, con lo cual se optimiza el tiempo de procesamiento. El algoritmo de Goertzel se puede resumir en el diagrama de bloques del filtro digital de la figura 11 y en su respectiva ecuación (7) (Cortes et al, 2010).

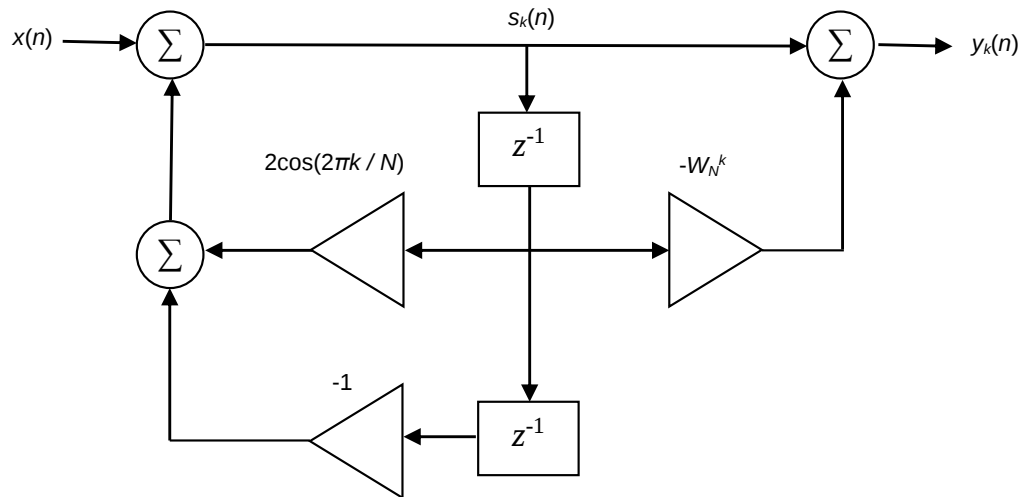


Figura 11. Filtro digital para el algoritmo de Goertzel.

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1 - W_N^k z^{-1}}{1 - 2 \cos\left(\frac{2\pi k}{N}\right) z^{-1} + z^{-2}} \quad (7)$$

De la figura 10 se obtienen las ecuaciones en diferencias (8) y (9)

$$s_k(n) = x(n) + 2\cos\left(\frac{2\pi k}{N}\right)s_k(n-1) - s_k(n-2) \quad (8)$$

$$y_k(n) = s_k(n) - W_N^k s_k(n-1) \quad (9)$$

En donde:

N = Número de muestras = 256

n = Índice de la n -ésima muestra = 0, 1, ..., $N-1$

k = Índice de la frecuencia fundamental = $\frac{Nf_0}{f_s}$

f_0 = Frecuencia de la señal periódica v_0 , $f_0 = 25$ kHz

f_s = Frecuencia de muestreo en el ADC del Arduino Due = 500 kmuestras/s

W_N^k = Factor de rotación, $W_N^k = \exp\left(-j\frac{2\pi k}{N}\right) = \cos\left(\frac{2\pi k}{N}\right) - j\sin\left(\frac{2\pi k}{N}\right)$

Sustituyendo el factor de rotación en la ecuación (9)

$$y_k(n) = s_k(n) - \left[\cos\left(\frac{2\pi k}{N}\right) - j\sin\left(\frac{2\pi k}{N}\right)\right]s_k(n-1)$$

Separando las componentes real e imaginaria de $y_k(n)$ en $y_{kR}(n)$ e $y_{kI}(n)$

$$y_{kR}(n) = s_k(n) - \cos\left(\frac{2\pi k}{N}\right)s_k(n-1) \quad (10)$$

$$y_{kI}(n) = \sin\left(\frac{2\pi k}{N}\right)s_k(n-1) \quad (11)$$

Las ecuaciones (8) (10) y (11) constituyen el algoritmo de Goertzel. Definiendo lo siguiente:

$$C_1 = 2\cos\left(\frac{2\pi k}{N}\right)$$

$$C_2 = \cos\left(\frac{2\pi k}{N}\right)$$

$$C_3 = \sin\left(\frac{2\pi k}{N}\right)$$

Las ecuaciones (8), (10) y (11) se pueden reescribir como:

$$\begin{aligned} s_k(n) &= x(n) + C_1 s_k(n-1) - s_k(n-2) \\ y_{kR}(n) &= s_k(n) - C_2 s_k(n-1) \\ y_{kI}(n) &= C_3 s_k(n-1) \end{aligned} \quad (12)$$

Para $n = 0, 1, \dots, 256$.

El grupo de ecuaciones en (12) se calculan en un ciclo iterativo en el software del Arduino Due como se muestra en el anexo C.

1.3.3 Producto punto

El producto punto entre dos arreglos, $x[n]$ y $y[n]$ se define como

$$x[n] \cdot y[n] = \sum_{i=0}^{N-1} x[i]y[i] = |x||y|\cos(\theta) \quad (13)$$

En donde θ es el ángulo entre los dos arreglos $x[n]$ y $y[n]$, $|x|$ y $|y|$ son las magnitudes de los arreglos y N es el número de muestras.

En un LVDT el ángulo θ entre las señales senoidales del primario, $v_p(t)$, y del secundario, $v_s(t)$, solo puede tomar dos valores, como se muestra en la figura 12 (Poley, 2003).

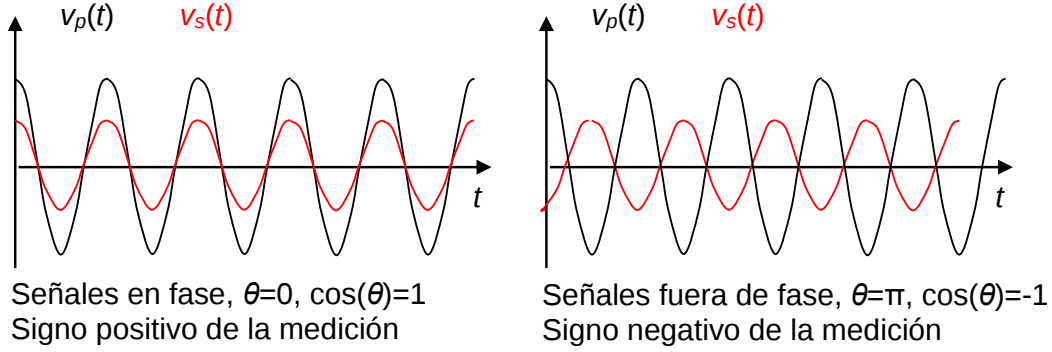


Figura 12. Relación entre las señales senoidales del primario y del secundario en un LVDT.

Sustituyendo la situación de la figura 12 en la ecuación (13)

$$\cos(\theta) = \frac{\sum_{i=0}^{N-1} v_p[i]v_s[i]}{|v_p||v_s|}$$

El equivalente de la ecuación anterior para las variables digitalizadas de la figura 4

$$\cos(\theta) = \frac{\sum_{i=0}^{N-1} v_o[i]v_i[i]}{|v_o||v_i|}$$

Dado que $|v_o|$ $|v_i|$ resulta ser constante y solamente se requiere saber el signo de la medición, entonces

$$\text{Signo} = \begin{cases} \text{positivo si } \sum_{i=0}^{N-1} v_o[i]v_i[i] > 0 \\ \text{negativo si } \sum_{i=0}^{N-1} v_o[i]v_i[i] < 0 \end{cases}$$

Definiendo la variable “producto”

$$\text{producto} = \sum_{i=0}^{N-1} v_o[i]v_i[i]$$

La relación que define el signo de la medición resulta ser

$$\text{Signo} = \begin{cases} \text{positivo si } \text{producto} > 0 \\ \text{negativo si } \text{producto} < 0 \end{cases} \quad (14)$$

La ecuación (14) se calcula en un ciclo iterativo en el software del Arduino Due como se muestra en el anexo C.

1.4 Procedimiento de operación

La figura 13 muestra el diagrama de conexiones para operar el sistema electrónico para el palpador LVDT.

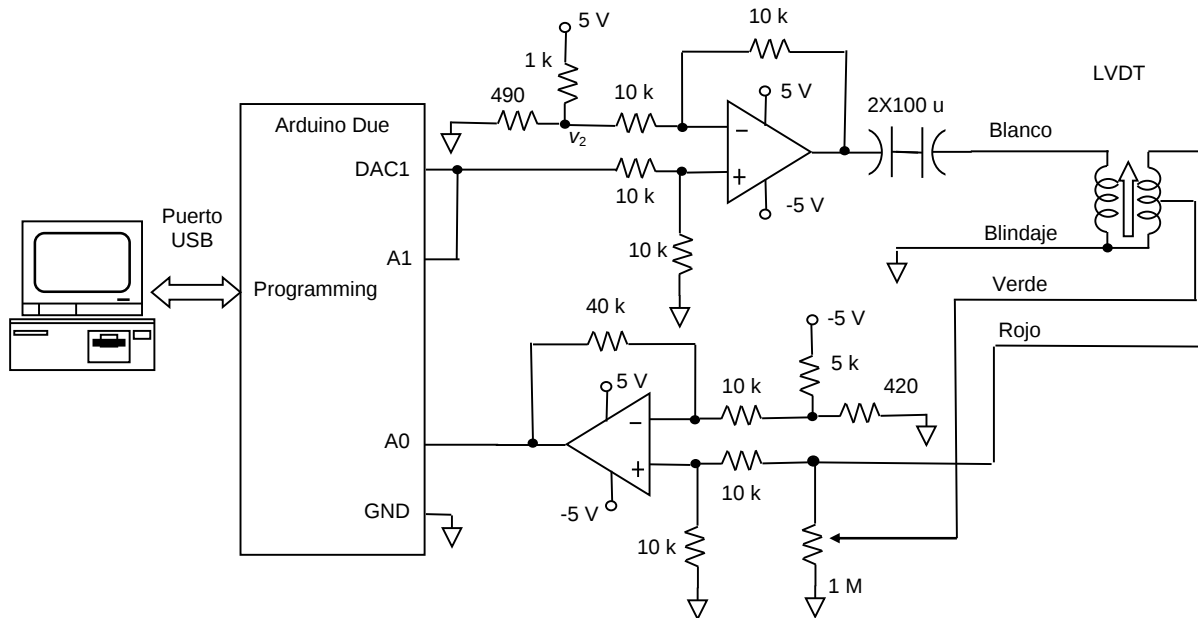


Figura 13. Diagrama de conexiones.

La siguiente es la lista de componentes necesarios en el diagrama de la figura 13:

- Sistema electrónico de amplificación.
- Tarjeta de desarrollo Arduino Due.
- Dos fuentes de voltaje de directa de 5 V, 1 A.
- Computadora personal con puerto USB.
- Palpador LVDT del perfilómetro Federal.

La computadora personal requiere de la instalación de algún programa de interconexión con el puerto serie como puede ser la “Hyper Terminal” de Windows o la consola “Monitor Serie” del IDE de Arduino. En este caso, se utilizó el programa “Tera Term” disponible como recurso gratuito en la red de internet (Tera Term, 2022). Esta interfaz de conexión serie se debe configurar con los siguientes parámetros:

- Port: COM8: Arduino Due Programming port.
- Baud rate: 57600.
- Data: 8.
- Parity: None.
- Stop: 1.
- Flow control: None.

El parámetro “Port” debe ser el registrado por el “Administrador de dispositivos” de Windows, como se muestra en la figura 14.

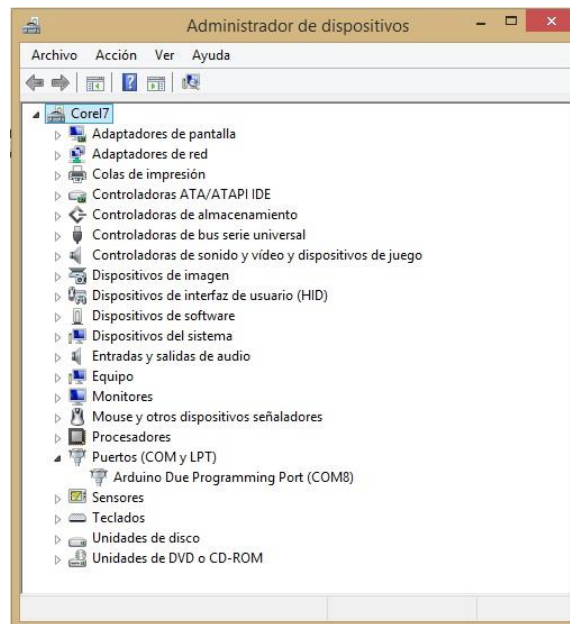


Figura 14. Administrador de dispositivos.

Una vez realizadas las conexiones y energizado el sistema, al ejecutar el programa terminal del puerto serie se debe observar un despliegue como el de la figura 15.

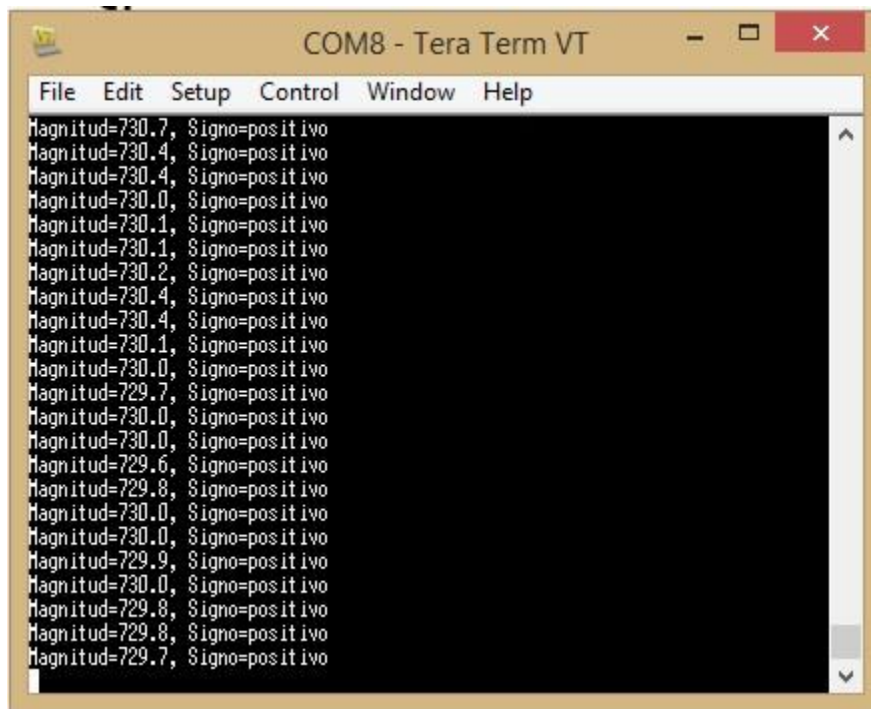


Figura 15. Programa de terminal para el puerto serie.

La figura 15 muestra el despliegue de la magnitud y signo de la medición calculadas por el sistema electrónico desarrollado.

2 Resultados y discusión

En esta sección se exponen los resultados obtenidos al desarrollar el sistema electrónico para la medición dimensional usando el palpador LVDT del perfilómetro Federal.

2.1 Resultados

Se desarrolló un prototipo funcional para la medición longitudinal y se comprobó su desempeño mediante la calibración con un instrumento patrón. Además, se implementaron programas de apoyo en Matlab que permitieron la depuración y verificación de los algoritmos implementados en el software del sistema electrónico.

2.1.1 Prototipo desarrollado

La figura 16 muestra el sistema electrónico desarrollado y su uso en la medición dimensional.



Figura 16. Sistema electrónico para el palpador LVDT.

La figura 16 muestra la importancia de contar con superficies de referencia robustas en el momento de realizar mediciones dimensionales de gran resolución. En este caso se emplea una mesa de granito con un gran nivel de planitud y estabilidad que proporciona tanto un nivel de referencia confiable como aislamiento de las vibraciones ambientales.

La figura 17 muestra la tarjeta electrónica desarrollada en una tablilla de laboratorio del tipo “protoboard”.

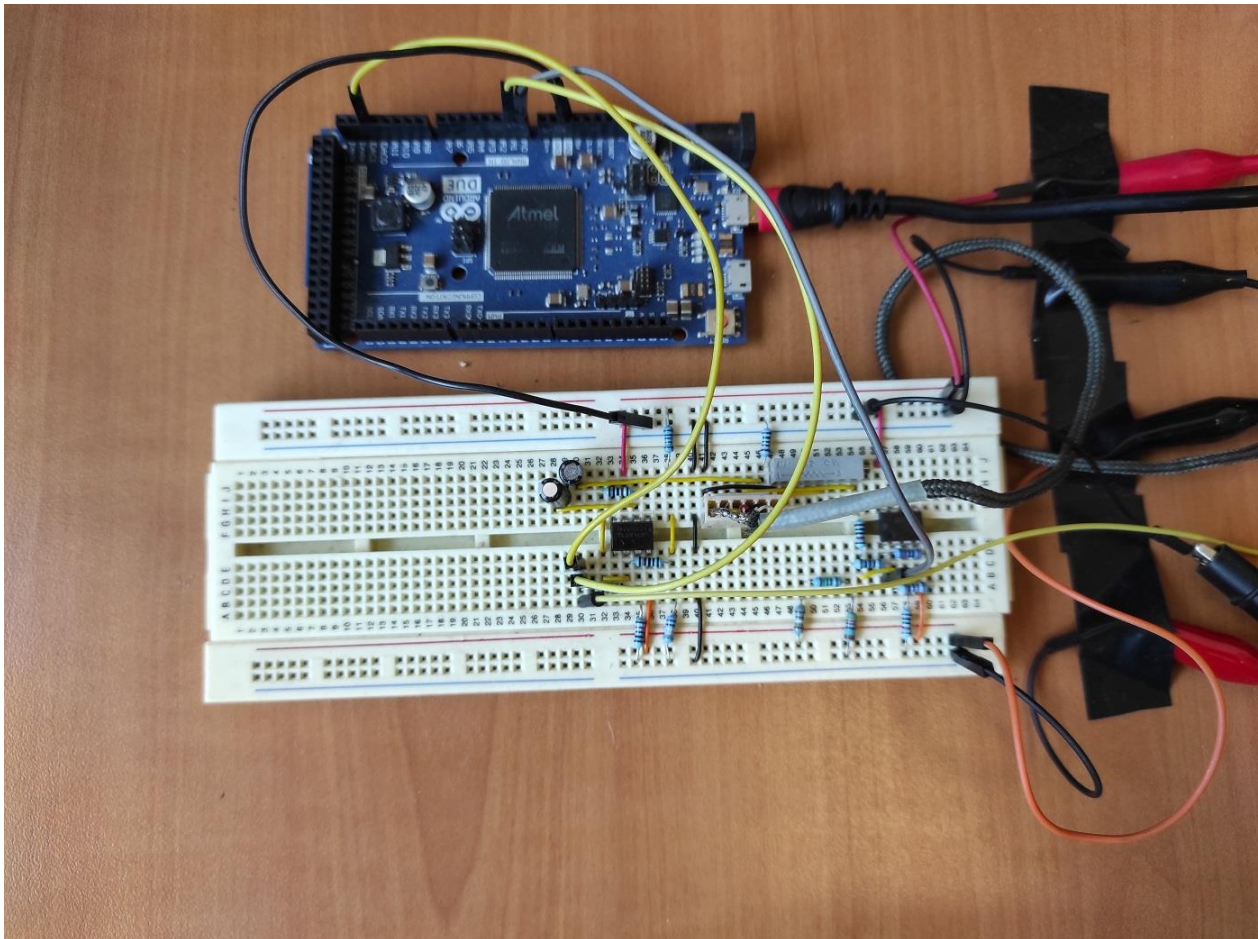


Figura 17. Tarjeta electrónica.

2.1.2 Síntesis de la señal en el primario

La figura 18 muestra la señal sintetizada en la salida DAC1 del Arduino Due, v_o de la figura 4. La excursión analógica mostrada en la figura 18 es de 2.8 V a 0.48 V, es decir, 2.32 V pico a pico, muy cercana a la excursión deseada de 3 V a 0 V o 3 V pico a pico. La diferencia entre los valores obtenidos y deseados se debe al error propio del convertidor digital a analógico en el Arduino Due. Por otra parte, la frecuencia fundamental de la señal senoidal de la figura 18 es de 25.2402 kHz valor muy cercano a la frecuencia deseada de 25kHz. A pesar de los errores mencionados, estos tienen

poca influencia en la calidad de la medición, ya que en la etapa de amplificación del primario se puede realizar la compensación por el error en la excursión analógica y el algoritmo de Goertzel se puede ajustar para obtener exactamente la magnitud de la componente fundamental de la onda senoidal.

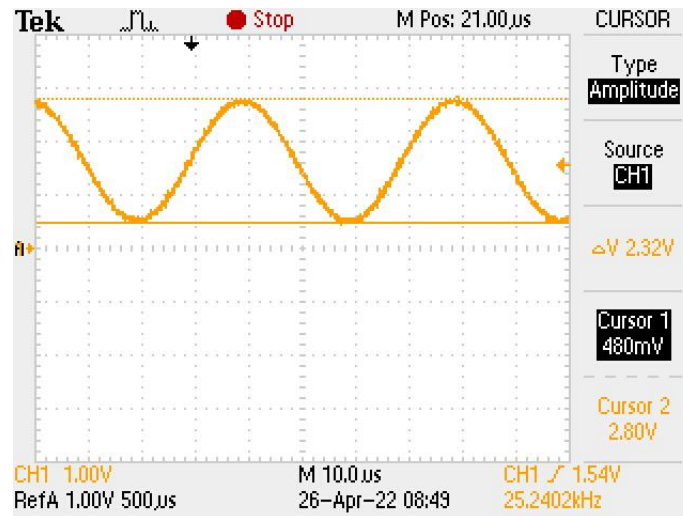


Figura 18. Señal sintetizada v_o .

La figura 19 muestra la señal acondicionada v_p de la figura 4. Como se puede apreciar, la excursión analógica de esta señal aplicada al primario del LVDT es de ± 1.1 V, es decir, 2.2 V de pico a pico, sin componente de directa, requisitos necesarios para el mejor desempeño del LVDT. Por otra parte, la frecuencia fundamental es de 25.2401 kHz, como era de esperarse. De esta forma se comprueba que el acondicionamiento analógico de la señal compensa el error generado por el convertidor digital a analógico del Arduino Due.

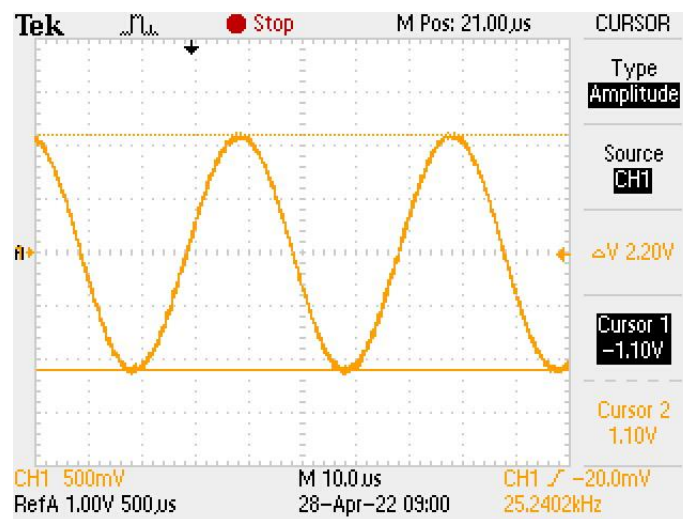


Figura 19. Señal acondicionada v_p .

2.1.3 Evaluación en Matlab

La consistencia y confiabilidad numérica del algoritmo de Goertzel, es evaluada al comparar el desempeño del algoritmo al utilizar el grupo de ecuaciones en (12) ejecutándose en el Arduino Due contra el cálculo estándar que proporciona Matlab como parte de su función “fft()”, Fast Fourier Transform (Transformada Rápida de Fourier). En este sentido se considera que la función “fft” de Matlab es la referencia patrón para afinar los detalles que existan en el momento de implementar el grupo de ecuaciones en (12).

La figura 20 muestra las señales en el tiempo del primario y secundario capturadas para la extracción de la medición mediante los algoritmos mencionados para el análisis espectral.

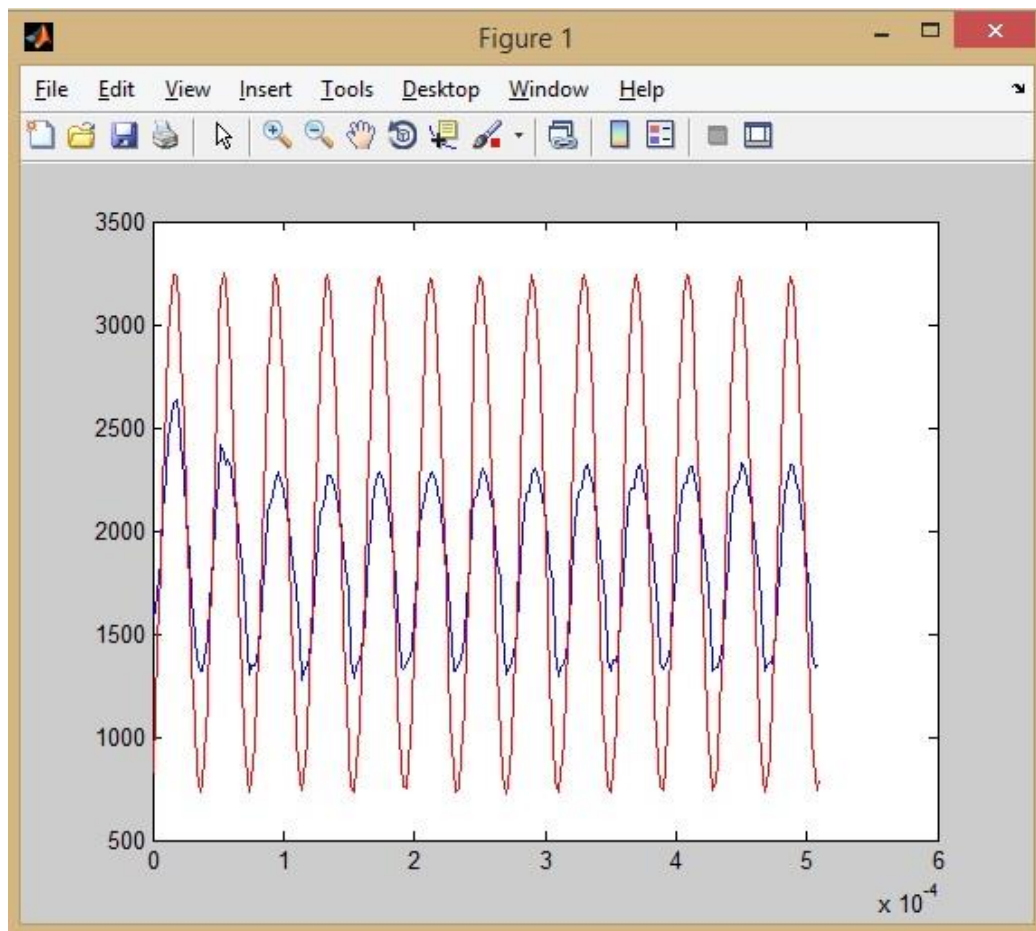


Figura 20. Señales del primario v_o “rojo” y del secundario v_i “azul”.

Con la señal v_i de la figura 20 se ejecutan los algoritmos para la Transformada Rápida de Fourier en Matlab, figura 21, y para el cálculo de las ecuaciones en (12) en el Arduino Due, figura 22.

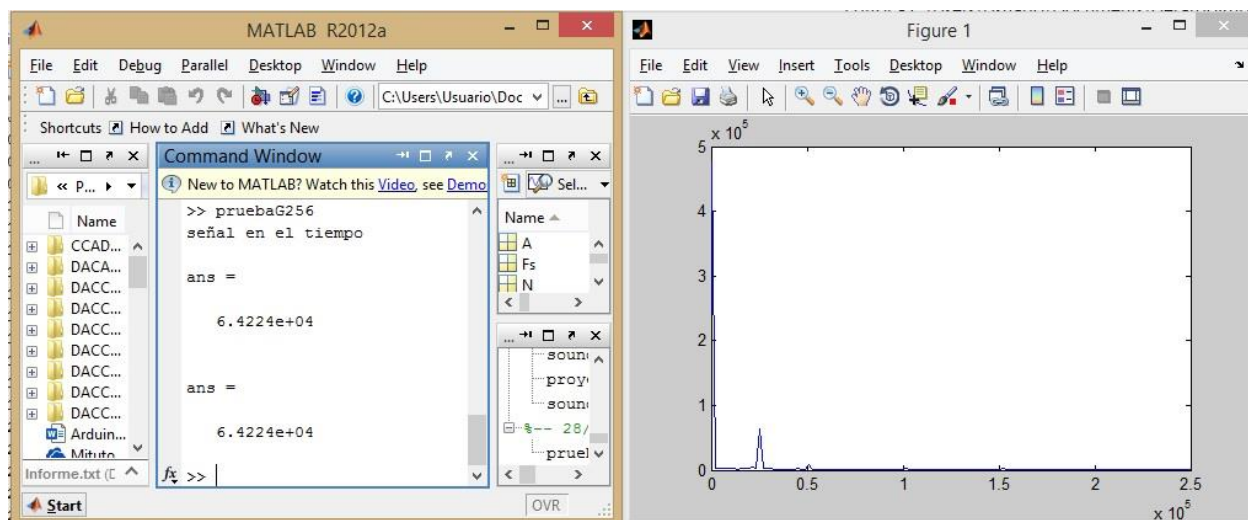


Figura 21. Transformada Rápida de Fourier en Matlab.

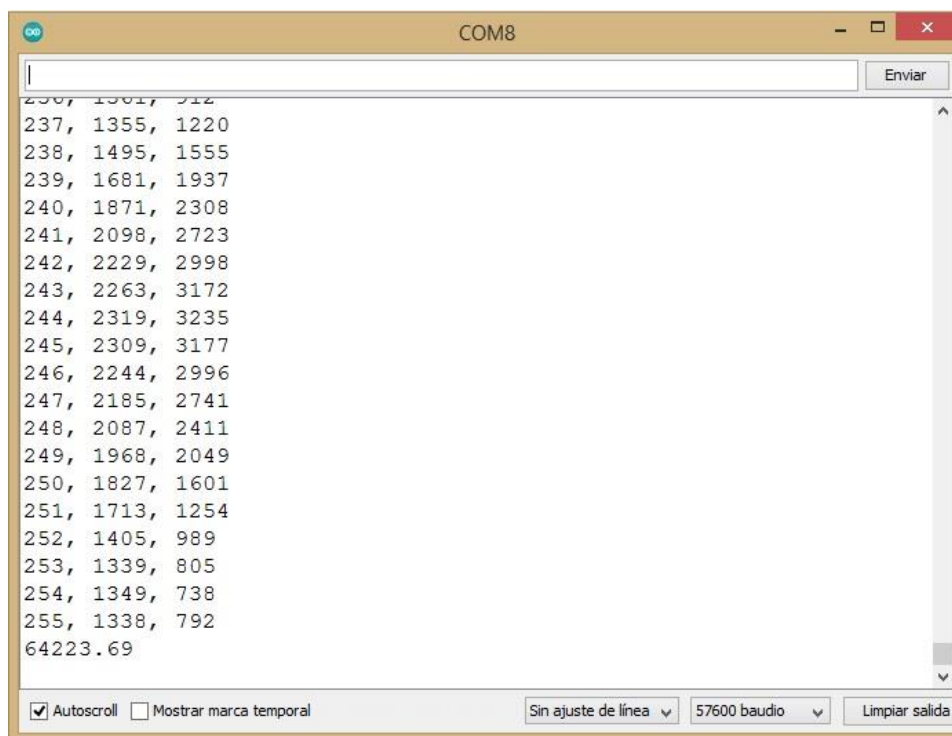


Figura 22. Algoritmo de Goertzel en Arduino Due.

Resumiendo los resultados de las figuras 21 y 22 en la tabla 1.

	MatLab	Arduino Due
Magnitud de la componente fundamental	64224	64223.69

Tabla 1. Comparación de resultados para el análisis espectral de la señal del LVDT.

La tabla 1 muestra un error insignificante al implementar el algoritmo de Goertzel en la aritmética de punto flotante del Arduino Due, con lo cual se asegura la estabilidad numérica y confiabilidad de la medición.

El anexo D contiene el código fuente en Matlab que proporciona el resultado de la figura 21 mientras que el anexo C contiene el código fuente utilizado para obtener el resultado de la figura 22. En el código del anexo D aparecen comentarios acerca de que líneas se deben comentar o des comentar para obtener el resultado de la figura 21.

2.1.4 Calibración

En la calibración del sistema desarrollado se utilizó el probador de calibración Mitutoyo 521-105, mostrado en la figura 23. Con este calibrador es posible generar desplazamientos con resolución de $0.2 \mu\text{m}$ en un alcance de 0 a 5 mm al girar el tambor superior que desplaza micrométricamente la platina de contacto entre el probador, y en este caso, el palpador LVDT.



Figura 23. Mitutoyo 521-105.

El experimento de calibración realizado consistió en la generación de desplazamientos patrón mediante el Mitutoyo 521-105 y su registro correspondiente por el sistema electrónico desarrollado. La figura 24 muestra la curva de calibración obtenida para un alcance burdo ($\pm 800 \mu\text{m}$). La figura 25 muestra la curva de calibración en un alcance fino ($\pm 40 \mu\text{m}$).

La figura 24 muestra la generación de desplazamientos en el Mitutoyo 521-105 a partir de $800 \mu\text{m}$ en pasos de $25 \mu\text{m}$ hasta $-800 \mu\text{m}$ pasando por el cero electrónico del sistema desarrollado y regresando nuevamente a la posición inicial de la medición, es decir $800 \mu\text{m}$. Con el regreso a $800 \mu\text{m}$ se verifica la repetibilidad del sistema. Cabe destacar la incoherencia en el cruce por cero entre ambos equipos ya que la señal residual del punto nulo en el palpador del LVDT es considerablemente mayor a la uniformidad en el avance por pasos de $25 \mu\text{m}$ del Mitutoyo 521-105. Para observar de mejor manera este último fenómeno, la figura 25 muestra un detalle de medición fina en la zona del cruce por cero.

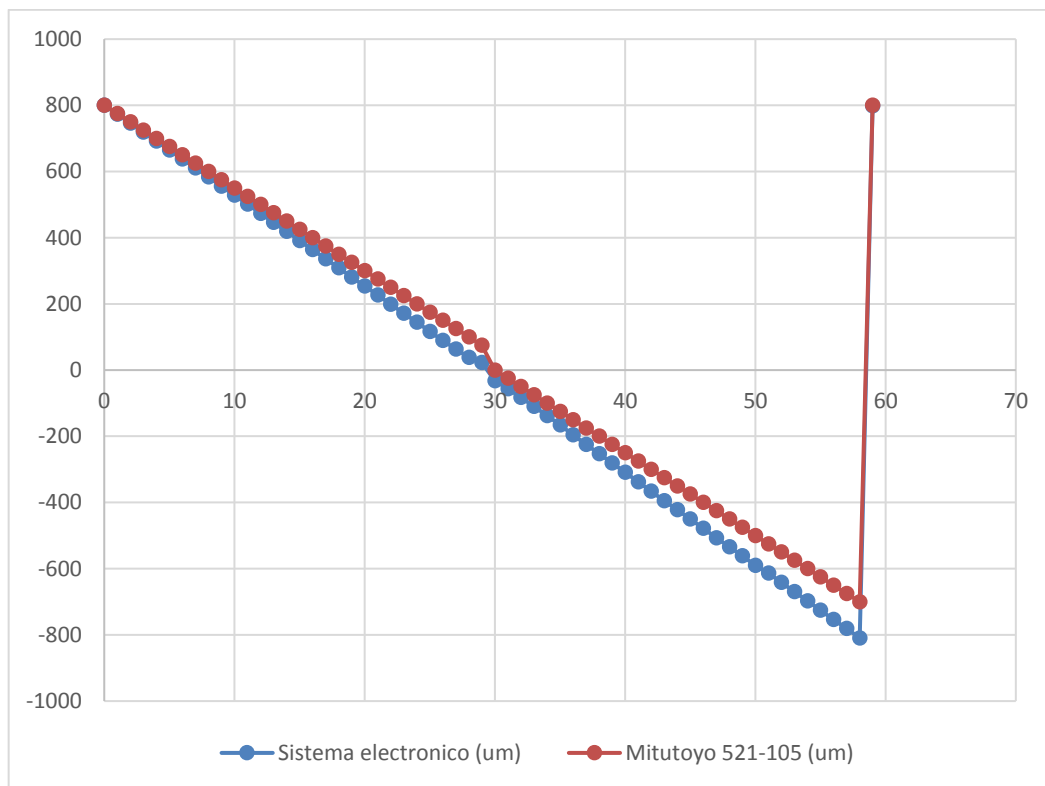


Figura 24. Curva de calibración para el alcance burdo ($\pm 800 \mu\text{m}$).

La curva de la figura 25 se obtuvo al generar desplazamientos patrón del Mitutoyo 521-105 a partir de $-38 \mu\text{m}$ hasta $-20 \mu\text{m}$ en pasos de $1 \mu\text{m}$ pasando por el cero del palpador LVDT. En la figura 25 se aprecia notablemente la no linealidad del palpador LVDT en las cercanías del punto nulo, pero más notorio es la falta de sensibilidad del palpador alrededor de $\pm 20 \mu\text{m}$ del punto nulo. La señal residual en el punto nulo del LVDT es un defecto de construcción propio del dispositivo y se debe al desbalance que existió en el

momento de enrollar los dos embobinados del secundario. Lo anterior indica que la zona cercana al cero electrónico del palpador no es apropiada para ser usada en el transcurso de una medición real.

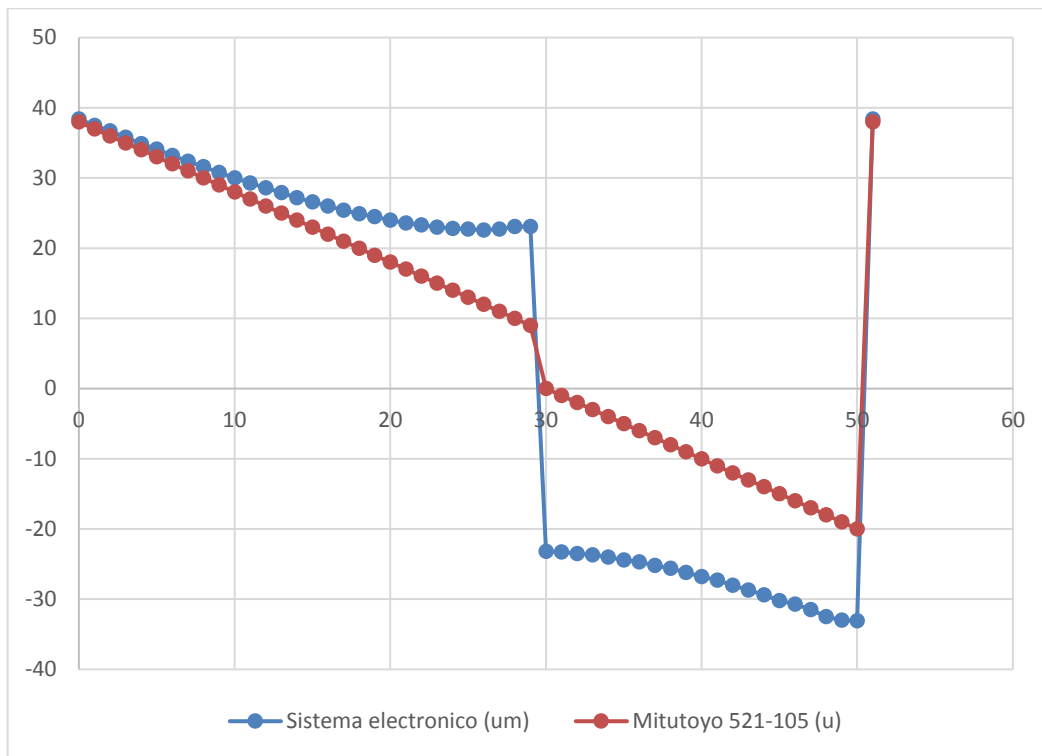


Figura 25. Curva de calibración para el alcance fino ($\pm 40 \mu\text{m}$).

2.2 Discusión y trabajo a futuro

Un aspecto importante para discutir es el incremento en la resolución que proporciona la manipulación numérica del algoritmo de medición implementado en el Arduino Due. Para empezar, la resolución del convertidor digital a analógico del Arduino Due es de 12 bits, es decir $2^{12} = 4096$ niveles discretos, que en teoría este sería la resolución de la medición. Si, por ejemplo, se tiene un alcance de medición de $\pm 800 \mu\text{m}$ entonces la resolución teórica en la medición sería de $1600/4096 = 0.39 \mu\text{m}$. No obstante, esta limitación teórica, en la práctica y con el ejercicio de calibración, se pudo comprobar que la resolución realmente alcanzada fue de $0.1 \mu\text{m}$ en el mismo alcance de medición de $\pm 800 \mu\text{m}$. La mejora obtenida es resultado del abundante tratamiento numérico en punto flotante que desempeña el algoritmo de medición en el Arduino Due. Como se observa en la figura 26, y tomando el intervalo entre dos marcas de tiempo, cada aproximadamente 0.3 s el algoritmo de medición ejecuta 150 transformadas de Fourier mediante el algoritmo de Goertzel y 150 productos punto de 256 muestras respectivamente, lo cual implica una gran cantidad de multiplicaciones y sumas en punto flotante para después sopesar con divisiones y raíz cuadrada. De aquí la gran cantidad de datos y el incremento en la resolución de la medición.

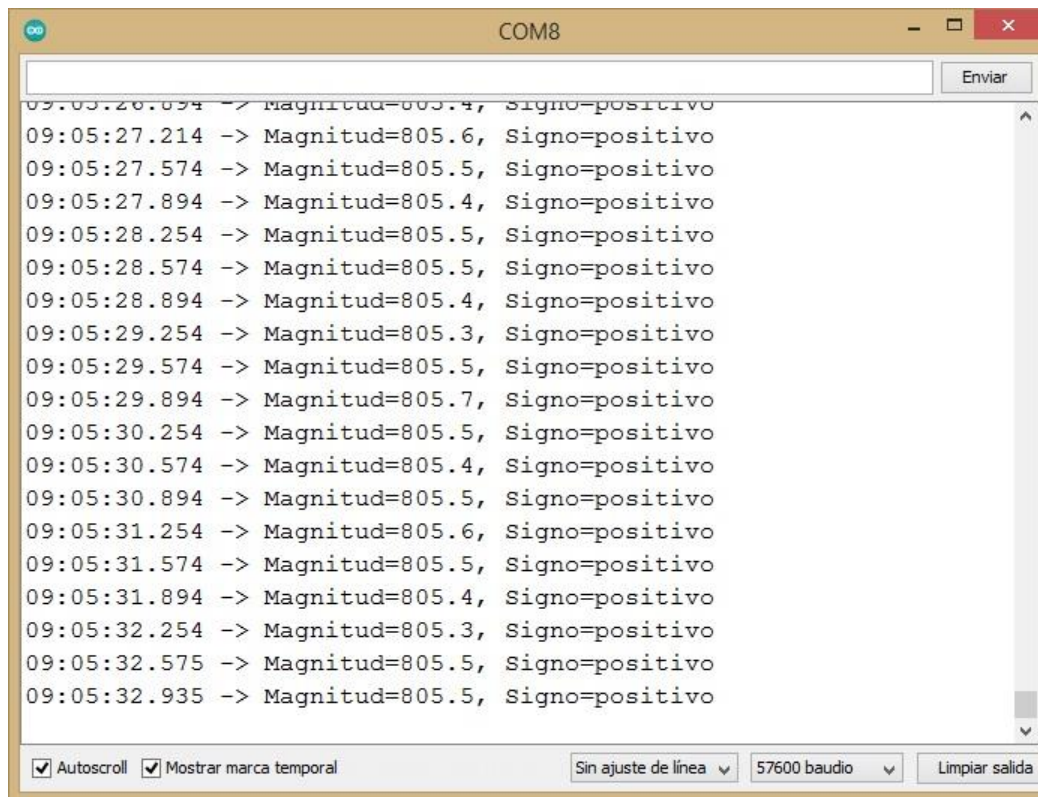


Figura 26. Incremento en la resolución.

Otro aspecto para discutir es la corrección del error inherente al punto nulo del LVDT. Como se muestra en la figura 25, una estrategia sería evitar esta zona con rangos deshabilitados definidos por software, con el inconveniente de que en una medición real podría ser que el objeto a medir obligue a utilizar esta zona en cuyo caso se debería repetir la medición e introducir un desplazamiento mecánico. Otra estrategia sería diseñar amplificadores que por hardware desplacen la zona útil de medición hacia uno de los extremos, ya sea positivo o negativo. El inconveniente en este último caso sería el desperdicio de media zona útil del LVDT. Otro enfoque sería ampliar el alcance analógico del convertidor analógico a digital de 0 a 3 V hasta ± 10 V con el objeto de incrementar la ganancia en las etapas de amplificación y con esto incrementar a su vez la relación señal a ruido. Con el incremento de la relación señal a ruido se debería reducir la zona no lineal cercana al punto nulo. El inconveniente de esta última estrategia es el inevitable remplazo del Arduino DUE por otro dispositivo con convertidor analógico a digital de mayor excursión analógica.

Otro aspecto para discutir sería el incremento de la resolución en la medición. Si bien lo deseable era obtener resolución de $0.01 \mu\text{m}$ en este trabajo solamente se obtuvo $0.1 \mu\text{m}$.

Como trabajo a futuro se piensa trabajar en estrategias que permitan alcanzar resoluciones de $0.01 \mu\text{m}$ con estabilidades en la medición de $\pm 0.01 \mu\text{m}$ a tasas de hasta 10 mediciones por segundo.

2.3 Especificaciones del sistema desarrollado

Las especificaciones que podemos obtener para el sistema electrónico desarrollado y que se pudieron comprobar como parte del ejercicio de calibración son las siguientes:

- Alcance de medición: $\pm 800 \mu\text{m}$.
- División mínima: $0.1 \mu\text{m}$.
- Estabilidad de la medición: $\pm 0.1 \mu\text{m}$.

2.4 Guía para la detección de fallas

La tabla 2 presenta una lista de posibles fallas y como solucionarlas.

Situación anómala del sistema de medición	Solución
La pantalla de terminal despliega caracteres extraños que no concuerdan con una medición.	Ajuste los parámetros de comunicación en el programa de terminal a 57600 bits por segundo, 8 bits de datos, sin paridad, 1 bit de parada y sin control de flujo.
La pantalla de terminal no presenta actividad alguna de medición.	Verifique que el driver de puerto serie esta correctamente identificado por el sistema operativo y que éste corresponde con el número de puerto en el programa de terminal. Presione el botón "RESET" en el Arduino DUE.
La medición en el programa de terminal es errónea.	Verifique que el sistema electrónico esta correctamente polarizado con fuentes externas de $\pm 5 \text{ V}$.

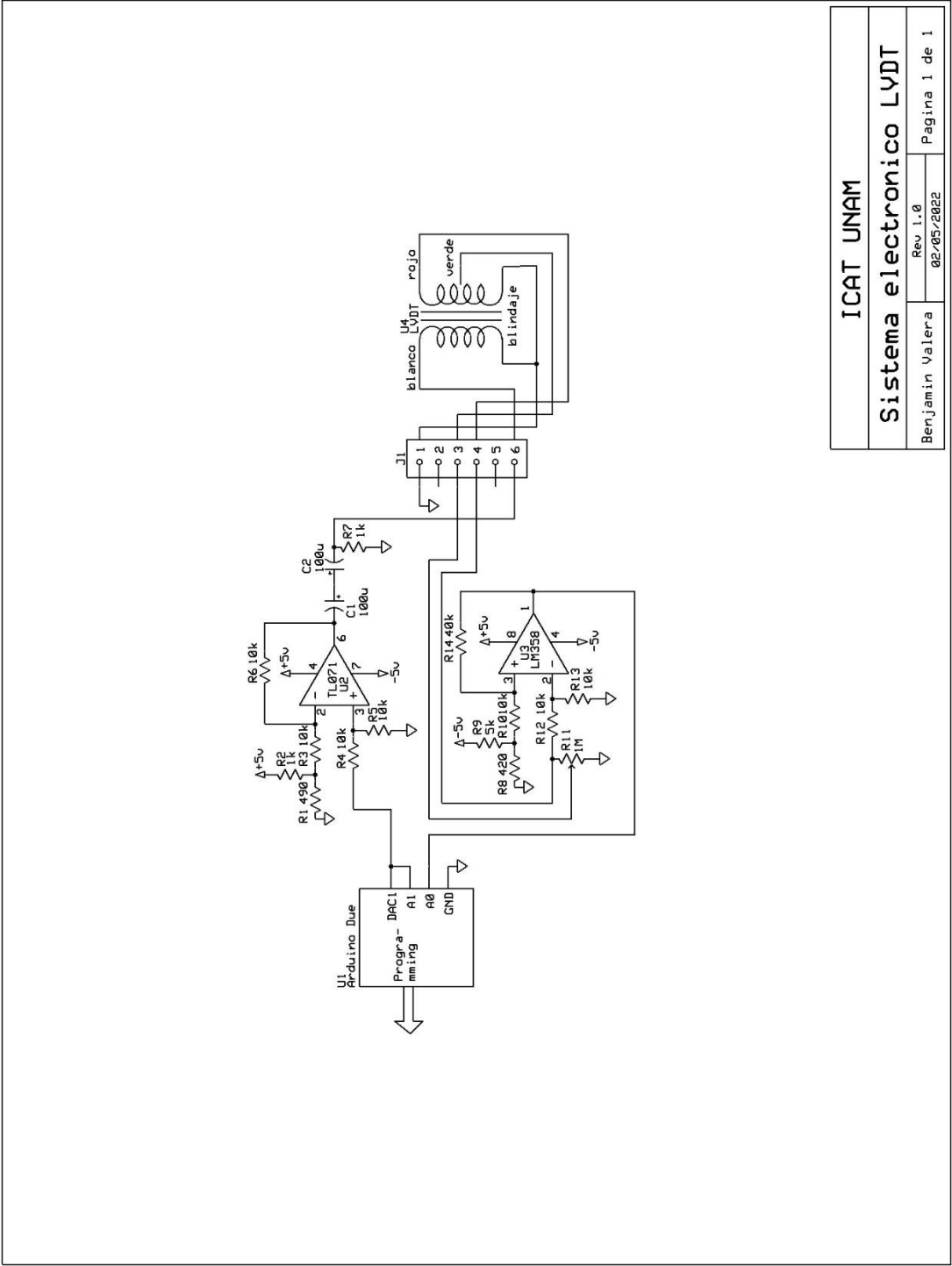
Tabla 2. Condiciones anormales de operación para el sistema de medición.

Conclusiones

Se diseñó y construyó un sistema electrónico para realizar mediciones dimensionales con el palpador LVDT del perfilómetro Federal 4000. El sistema electrónico está construido sobre la base del microcontrolador de 32 bits Arduino Due. Se implementaron etapas de acondicionamiento analógico que permiten adecuar las señales que manipula tanto el Arduino Due como el palpador LVDT. Se implementó un algoritmo de medición en lenguaje C para Arduino Due que realiza el análisis espectral de la señal del secundario del LVDT mediante el algoritmo de Goertzel que permite obtener la magnitud de la medición y que, por otra parte, obtiene el signo de la medición mediante el producto punto entre los dos arreglos de voltaje del primario y del secundario. Para comprobar el funcionamiento del sistema, se realizó un experimento de calibración en donde se caracterizó el palpador LVDT y se obtuvieron las especificaciones del sistema. Además, se realizaron diversas pruebas de depuración tanto en los circuitos analógicos como en los algoritmos empleados.

Los resultados alcanzados con el sistema de medición comprueban su buen desempeño y viabilidad para usar esta tecnología en casos prácticos.

Anexo A Diagrama esquemático



ICAT UNAM		
Sistema electronico LYDT		
Benjamin Valera	Rev 1.0 02/05/2022	Pagina 1 de 1

Figura 27. Dibujo esquemático.

Anexo B Lista de partes

Identificación	Descripción
C1	Capacitor electrolítico de 100u a 10V
C2	Capacitor electrolítico de 100u a 10V
J1	Conector Molex
R1	Resistencia de precisión 490 a 1/4W
R2	Resistencia de precisión 1k a 1/4W
R3	Resistencia de precisión 10k a 1/4W
R4	Resistencia de precisión 10k a 1/4W
R5	Resistencia de precisión 10k a 1/4W
R6	Resistencia de precisión 10k a 1/4W
R7	Resistencia de precisión 1k a 1/4W
R8	Resistencia de precisión 420 a 1/4W
R9	Resistencia de precisión 5k a 1/4W
R10	Resistencia de precisión 10k a 1/4W
R11	Potenciómetro de 10 vueltas1M
R12	Resistencia de precisión 10k a 1/4W
R13	Resistencia de precisión 10k a 1/4W
R14	Resistencia de precisión 40k a 1/4W
U1	Microcontrolador Arduino Due
U2	Amplificador operacional TL071
U3	Amplificador operacional LM358
U4	Palpador LVDT del perfilómetro Federal 4000

Tabla 3. Lista de partes.

Anexo C Código fuente

```
// DAC
// Frecuencia = 25KHz
// ADC
// Muestreo = 500KMuestras/s = 20 muestras por periodo
// Algoritmo de Goertzel

#define SAMPLES 64 // Muestras del DAC
int x[256]; // Primario tiempo
int y[256]; // Secundario tiempo
const float c2=0.94952818059; // Constantes del algoritmo
const float c1=1.89905636119;
const float c3=0.3136817404;
float sk0=0; // Variables del algoritmo
float sk1=0;
float sk2=0;
float ykr;
float yki;
float producto; // Variables estadísticas
double acumuladorValor;
double promedioValor;
double acumuladorSigno;
double promedioSigno;
uint16_t wave[SAMPLES]; // Senoidal sintetizada

void CreaTablaOndaCoseno()
{
    wave[0]=4095;
    wave[1]=4086;
    wave[2]=4056;
    wave[3]=4007;
    wave[4]=3940;
    wave[5]=3854;
    wave[6]=3750;
    wave[7]=3631;
    wave[8]=3496;
    wave[9]=3347;
    wave[10]=3185;
    wave[11]=3013;
    wave[12]=2831;
    wave[13]=2642;
    wave[14]=2447;
    wave[15]=2248;
    wave[16]=2048;
    wave[17]=1847;
    wave[18]=1648;
    wave[19]=1453;
    wave[20]=1264;
    wave[21]=1082;
    wave[22]=910;
    wave[23]=748;
    wave[24]=599;
    wave[25]=464;
    wave[26]=345;
    wave[27]=241;
```

```

    wave[28]=155;
    wave[29]=88;
    wave[30]=39;
    wave[31]=9;
    wave[32]=0;
    wave[33]=9;
    wave[34]=39;
    wave[35]=88;
    wave[36]=155;
    wave[37]=241;
    wave[38]=345;
    wave[39]=464;
    wave[40]=599;
    wave[41]=748;
    wave[42]=910;
    wave[43]=1082;
    wave[44]=1264;
    wave[45]=1453;
    wave[46]=1648;
    wave[47]=1847;
    wave[48]=2047;
    wave[49]=2248;
    wave[50]=2447;
    wave[51]=2642;
    wave[52]=2831;
    wave[53]=3013;
    wave[54]=3185;
    wave[55]=3347;
    wave[56]=3496;
    wave[57]=3631;
    wave[58]=3750;
    wave[59]=3854;
    wave[60]=3940;
    wave[61]=4007;
    wave[62]=4056;
    wave[63]=4086;
}

void setupDAC()
{
    pmc_enable_periph_clk (DAC_INTERFACE_ID) ;    // Start clocking DAC.
    dacc_reset(DACC);
    dacc_set_transfer_mode(DACC, 0);
    dacc_set_power_save(DACC, 0, 1);               // sleep = 0, fast wakeup = 1
    dacc_set_analog_control(DACC, DACC_ACR_IBCTLCH0(0x02) |
DAC_ACR_IBCTLCH1(0x02) | DACC_ACR_IBCTLDACCORE(0x01));
    dacc_set_trigger(DACC, 1);
    dacc_set_channel_selection(DACC, 1);
    dacc_enable_channel(DACC, 1);
    NVIC_DisableIRQ(DACC_IRQn);
    NVIC_ClearPendingIRQ(DACC_IRQn);
    NVIC_EnableIRQ(DACC_IRQn);
    dacc_enable_interrupt(DACC, DACC_IER_ENDTX);
    DACC->DACC_PTCR = 0x00000100;
}

void DACC_Handler(void)

```

```

{
    DACC->DACC_TNPR = (uint32_t) wave;
    DACC->DACC_TNCR = SAMPLES; // Number of counts until Handler
re-triggered
}

// System timer clock set-up for DAC wave.
void setupTC (float freq_hz)
{
    int pasos = (42000000UL / freq_hz) / (SAMPLES);
    pmc_enable_periph_clk(TC_INTERFACE_ID);
    TcChannel * t = &(TC0->TC_CHANNEL)[0];
    t->TC_CCR = TC_CCR_CLKDIS; // Disable TC clock.
    t->TC_IDR = 0xFFFFFFFF;
    t->TC_SR; // Clear status register.
    t->TC_CMR = // Capture mode.
        TC_CMR_TCCLKS_TIMER_CLOCK1 | // Set the timer clock to TCLK1
(MCK/2 = 84MHz/2 = 48MHz).
        TC_CMR_WAVE | // Waveform mode.
        TC_CMR_WAVSEL_UP_RC; // Count up with automatic trigger
on RC compare.
    t->TC_RC = pasos; // Frequency.
    t->TC_RA = pasos /2; // Duty cycle (btwn 1 and RC).
    t->TC_CMR = (t->TC_CMR & 0xFFF0FFFF) |
        TC_CMR_ACPA_CLEAR | // Clear TIOA on counter match
with RA0.
        TC_CMR_ACPC_SET; // Set TIOA on counter match with
RC0.
    t->TC_CCR = TC_CCR_CLKEN | TC_CCR_SWTRG; // Enables the clock if CLKDIS is
not 1.
}

void setup()
{
    Serial.begin(57600); // initialize the serial
port:
// DAC
    analogWriteResolution(12);
    CreaTablaOndaCoseno();
    setupDAC();
    float freq_hz = 25000; // Target: 25kHz
    setupTC(freq_hz);
    NVIC_EnableIRQ(DACC_IRQn);

// ADC
    REG_ADC_MR = 0x10380180; // change from 10380200 to
10380180, 1 is the PREESCALER and 8 means FREERUN
    ADC -> ADC_CHER = 0xC0; // enable ADC on pin A0 y A1
}

void loop()
{
// ADC
    acumuladorValor=0;
    acumuladorSigno=0;
    for(int j=0; j<150; j++) // enmascarar para depurar
    { // enmascarar para depurar

```

```

    for(int i=0; i<256; i++)
    {
        while((ADC->ADC_ISR & 0xC0)==0); // wait for conversion
        x[i]= ADC->ADC_CDR[7]; // read value A0
        y[i]= ADC->ADC_CDR[6]; // read value A1
        while((ADC->ADC_ISR & 0xC0)==0); // wait for conversion
        x[i]= ADC->ADC_CDR[7]; // read value A0
        y[i]= ADC->ADC_CDR[6]; // read value A1
    }
// Algoritmo de Goertzel
    sk1=0;
    sk2=0;
    for(int i=0; i<256; i++)
    {
        sk0=x[i]+(c1*sk1)-sk2;
        sk2=sk1;
        sk1=sk0;
    }
    ykr = (c2*sk1) - sk2;
    yki = c3*sk1;
    acumuladorValor=acumuladorValor+sqrt((ykr*ykr)+(yki*yki));

// Determina el signo segun el defasamiento entre x y y y mediante su producto
punto
    producto=0;
    for(int i=0; i<256; i++)
    {
        producto=producto+(float)((x[i]-2048)*(y[i]-2048));
    }
    acumuladorSigno=acumuladorSigno+producto;
} // enmascarar para depurar

promedioValor=acumuladorValor/150; // Promedio de 150 lecturas
promedioValor=promedioValor*0.0105; // Calibracion

promedioSigno=acumuladorSigno/150;

// Imprime

Serial.print("Magnitud="); // enmascarar para depurar
Serial.print(promedioValor,1);
Serial.print(", Signo=");
if(promedioSigno>0) Serial.println("positivo");
else Serial.println("negativo");
/* // des enmascarar para depurar
for(int i=0; i<256; i++)
{
    Serial.print(i);
    Serial.print(", ");
    Serial.print(x[i]);
    Serial.print(", ");
    Serial.println(y[i]);
}
Serial.println(acumuladorValor);

do
{

```

```
    }while(true);  
*/  
}
```

Anexo D Código en Matlab

```
clear all
close all

Fs = 500000;           % Frecuencia de muestreo
T = 1/Fs;              % Periodo de muestreo
N = 256;               % Longitud de la señal
t = (0:N-1)*T;         % Vector tiempo
f = Fs*(0:(N/2))/N;     % Vector frecuencia

% Lectura de la señal en x
fprintf(1,'señales en el tiempo\n');
fileID = fopen('informe.txt','r');
formatSpec = '%d, %d, %d';
sizeA = [3 Inf];
A = fscanf(fileID,formatSpec,sizeA);
fclose(fileID);
x=A(2,(1:end));
y=A(3,(1:end));
plot(t, x, 'b', t, y, 'r');
pause

% fft
Y=fft(x);
abs(Y(14))
plot(f,abs(Y(1:129)));

% Algoritmo de goertzel
c2=0.94952818059;      % c2=cos(2pi(14-1)/N)
c1=1.89905636119;      % c1=2cos(2pi(14-1)/N)
c3=0.3136817404;       % c3=sin(2pi(14-1)/N)
sk1=0; sk2=0;
for n = 1 : N
    sk0 = x(n) + (c1*sk1) - sk2;
    sk2 = sk1;
    sk1 = sk0;
end
ykr = (c2*sk1)-sk2;
yki = c3*sk1;
sqrt((ykr*ykr)+(yki*yki))
```

Referencias bibliográficas

Arduino (2019) Arduino, [Internet], Disponible desde <<https://www.arduino.cc/>> [Acceso 4 de mayo de 2022].

Cortés J. A., Mendoza J. A., Muriel J. A. (2010) Alternativa al análisis en frecuencia de la fft mediante el algoritmo Goertzel. Scientia et Technica Año XVI, No 44 Abril, p. 217-222.

Coughlin R. F. y Driscoll F. F. (2000) Amplificadores operacionales y circuitos integrados, Prentice Hall 5a edición, 1 Mayo, p. 547.

Hayt W. (2019) Análisis de circuitos en ingeniería, McGraw-Hill Interamericana de España S.L.; 9a edición 30 de Mayo, p. 888.

Kyle (2019), Waveform Generator, [Internet], Disponible desde <<http://www.kylescholz.com/wp/waveform-generator>> [Acceso 4 de mayo de 2022].

Poley R. (2003) Signal Conditioning an LVDT Using a TMS320F2812 DSP, Texas Instruments, Agosto Application Report, p. 27.

Tera Term (2022) Tera Term Home Page, [Internet], Disponible desde <<https://ttssh2.osdn.jp/index.html.en>> [Acceso 4 de mayo de 2022].

Valera B. y Ruiz G. (2019) Sistema electrónico para un perfilómetro, ICAT UNAM, Informe Técnico, p. 41.