



Universidad de Sevilla
Escuela Politécnica Superior de Sevilla



Trabajo Fin de Grado en Ingeniería Electrónica Industrial e Ingeniería Eléctrica

Detección de movimiento y posición utilizando acelerómetros de 3 ejes

Autor

Francisco Prol Prol

Tutores

Carlos Jesús Jiménez Fernández

José Miguel Mora Gutiérrez

Septiembre 2022



Universidad de Sevilla
Escuela Politécnica Superior de Sevilla



RESUMEN

Detección de movimiento y posición utilizando acelerómetros de 3 ejes

Autor: Francisco Prol Prol

Tutores: Carlos Jesús Jiménez Fernández y José Miguel Mora Gutiérrez

El presente trabajo pretende estudiar y desarrollar un método para medir el desplazamiento de objetos y determinar su posición. El método propuesto consiste en medir la aceleración de un objeto, empleando como sensor un acelerómetro de 3 ejes, y aplicarles a estos datos un procesado que permita obtener la distancia recorrida. Una vez se obtiene el desplazamiento, conociendo la posición inicial, se puede determinar la posición final con relativa facilidad. Para realizar el sistema, se desarrolla un diseño hardware en VHDL que se implementa sobre una FPGA.

Se parte de la hipótesis de que los datos obtenidos por el acelerómetro tendrán la calidad suficiente para, una vez realizado el procesado, obtener resultados buenos. Por tanto, lo primero será comprobar la hipótesis.

Para ello, se realiza un estudio preliminar, desarrollando, en primer lugar, una primera versión hardware de la primera etapa del sistema que se encarga de la adquisición de datos, y en segundo, una primera versión software de la segunda etapa del sistema que se ocupa del procesado de datos. Tras analizar los datos de varias mediciones, a la salida de la primera etapa y a la salida de la segunda, se concluye que la hipótesis se cumple, aunque si se respetan ciertas condiciones de funcionamiento.

Una vez validada la hipótesis, se diseñan las versiones finales hardware de las dos primeras etapas. A continuación, se desarrolla la tercera etapa del sistema que permite visualizar la distancia recorrida empleando un visualizador 7 segmentos. Tras combinar las tres etapas, se consigue una versión final del sistema que funciona mientras se cumplan las restricciones. Por último, se desarrollan tres aplicaciones para el sistema y se comprueba su funcionamiento.

PALABRAS CLAVE

Acelerómetro, medidas de distancia, FPGA, VHDL.



Universidad de Sevilla
Escuela Politécnica Superior de Sevilla



ABSTRACT

Motion detection and position detection using 3-axis accelerometers

Author: Francisco Prol Prol

Tutor: Carlos Jesús Jiménez Fernández and José Miguel Mora Gutiérrez

The present work aims to study and develop a method to measure the displacement of objects and determine their position. The proposed method consists of measuring the acceleration of an object, using a 3-axis accelerometer as a sensor, and applying a process to these data that allows obtaining the distance travelled. Once the displacement is obtained, knowing the initial position, the final position can be determined with relative ease. To make the system, a hardware design is developed in VHDL that is implemented on an FPGA.

It is based on the hypothesis that the data obtained by the accelerometer will have sufficient quality to obtain good results once the processing has been carried out. Therefore, the first thing will be to test the hypothesis.

To do this, a preliminary study is carried out, developing, firstly, a first hardware version of the first stage of the system that is responsible for data acquisition, and secondly, a first software version of the second stage of the system that deals with data processing. After analysing the data of several measurements, at the output of the first stage and at the output of the second, it is concluded that the hypothesis is fulfilled, even if certain operating conditions are respected.

Once the hypothesis has been validated, the final hardware versions of the first two stages are designed. Next, the third stage of the system is developed that allows the distance travelled to be displayed using a 7-segment display. After combining the three stages, a final version of the system is achieved that works as long as the restrictions are met. Finally, three applications for the system are developed and their operation is checked.

KEYWORDS

Accelerometer, distance measurement, FPGA, VHDL.

ÍNDICE

1. INTRODUCCIÓN	1
1.1. Justificación	1
1.2. Objetivo	2
1.3. Hipótesis	4
1.4. Resumen	7
2. ESTADO DE LA CUESTIÓN	10
2.1. Métodos de medida de distancia y de detección de proximidad	10
2.1.1. Introducción	10
2.1.2. Clasificaciones de los métodos	10
2.1.3. Principios y técnicas de los métodos sin contacto	11
2.2. Sensores de distancia y de proximidad	14
2.2.1. Clasificación de los sensores	14
2.2.2. Sensores de distancia	20
2.2.3. Sensores de proximidad	20
3. CONCEPTOS TEÓRICOS, MATERIALES Y TÉCNICAS	21
3.1. Conceptos teóricos	21
3.1.1. Instrumentación y medidas	21
3.1.2. Cinemática de un movimiento rectilíneo uniformemente acelerado	28
3.2. Materiales	36
3.2.1. Hardware	36
3.2.2. Software	41
3.3. Técnicas	43
3.3.1. Lenguajes	43
3.3.2. Protocolos	44
4. ESTUDIO PRELIMINAR PARA COMPROBAR LA HIPÓTESIS	47
4.1. Primera versión hardware de la primera etapa “Adquisición de datos”	47
4.1.1. Introducción	47
4.1.2. Funcionamiento del diseño	47
4.1.3. Descripción de los componentes del diseño	48
4.1.4. Análisis de datos	60
4.2. Primera versión software de la segunda etapa “Procesado de datos”	62
4.2.1. Introducción	62
4.2.2. Base teórica del algoritmo	62

4.2.3.	Descripción del script de Matlab	63
4.2.4.	Análisis de datos	65
4.3.	Resultados.....	82
4.3.1.	Resultados para el desarrollo del sistema de medida.....	82
4.3.2.	Resultados del análisis de datos	83
4.4.	Conclusiones	84
5.	DESARROLLO HARDWARE DEL SISTEMA.....	85
5.1.	Introducción.....	85
5.2.	Versión final hardware de la primera etapa “Adquisición de datos”	86
5.2.1.	Funcionamiento del diseño	86
5.2.2.	Descripción de los componentes del diseño	86
5.3.	Versión final hardware de la segunda etapa “Procesado de datos”.....	90
5.3.1.	Funcionamiento del diseño	90
5.3.2.	Descripción de los componentes del diseño	92
5.4.	Versión final hardware de la tercera etapa “Presentación de los resultados”.....	98
5.4.1.	Funcionamiento del diseño	98
5.4.2.	Descripción de los componentes del diseño	99
5.5.	Primera versión del prototipo (1 eje).....	105
5.5.1.	Funcionamiento del diseño	105
5.5.2.	Descripción de los componentes del diseño	105
5.6.	Versión final del prototipo (2 ejes).....	108
5.6.1.	Funcionamiento del diseño	108
5.6.2.	Descripción de los componentes del diseño	108
6.	APLICACIONES.....	109
6.1.	Introducción.....	109
6.2.	Primera aplicación	110
6.2.1.	Funcionamiento del diseño	110
6.2.2.	Descripción de los componentes del diseño	110
6.2.3.	Resultados.....	113
6.3.	Segunda aplicación	114
6.3.1.	Funcionamiento del diseño	114
6.3.2.	Descripción de los componentes del diseño	114
6.3.3.	Resultados.....	115
6.4.	Tercera aplicación.....	116
6.4.1.	Funcionamiento del diseño	116
6.4.2.	Descripción de los componentes del diseño de la placa transmisora.....	117

6.4.3.	Descripción de los componentes del diseño de la placa receptora	117
6.4.4.	Resultados.....	117
7.	CONCLUSIONES.....	118
8.	BIBLIOGRAFÍA	120

ÍNDICE DE FIGURAS

Figura 1: Ecuaciones que describen la cinemática de un MRUA	2
Figura 2: Procesos principales del diseño hardware	2
Figura 3: Placa Nexys4 DDR de Digilent.....	3
Figura 4: Esquema de la primera versión hardware de la primera etapa “Adquisición de datos”	4
Figura 5: Representación gráfica de la aceleración en el eje X para un movimiento en la dirección del eje X en sentido positivo y negativo (en valores binarios) en función del tiempo (en valores muestreados, siendo cada unidad de valor muestreado equivalente al periodo de muestreo, es decir, a la inversa de la frecuencia de muestreo)	4
Figura 6: Representación gráfica de aceleración (rojo), velocidad (verde) y posición (azul) (en valores binarios) en función del tiempo (en valores muestreados, siendo cada unidad de valor muestreado equivalente al periodo de muestreo, es decir, a la inversa de la frecuencia de muestreo)	5
Figura 7: Esquema de la segunda etapa “Procesado de datos”	7
Figura 8: Esquema de la tercera etapa “Presentación de los resultados”.....	7
Figura 9: Superficie de la primera aplicación.....	8
Figura 10: Superficie de la segunda aplicación	8
Figura 11: Esquema del sistema empleando el módulo HC-12 para transmitir de forma inalámbrica	9
Figura 12: Técnica de la triangulación.....	12
Figura 13 (a): Circuito de un potenciómetro como un divisor de tensión.....	14
Figura 13 (b): Representación gráfica del voltaje a la salida del circuito en función del desplazamiento del elemento móvil.....	14
Figura 14: Circuito de un sensor inductivo	15
Figura 15: Diagrama esquemático de un sensor de desplazamiento mediante deflexión de haz óptico.....	17
Figura 16: Subsistema de un sensor de medida de distancia mediante ultrasonidos.....	18
Figura 17: Diagrama de bloques de un instrumento de medida	21
Figura 18: Función de transferencia lineal.....	23
Figura 19: Curva de calibración.....	24
Figura 20: Analogía de la precisión de la medición con los disparos a una diana	25

Figura 21: Distribución gaussiana	27
Figura 22: Desplazamiento en el eje Y	28
Figura 23: Posición de un objeto que se mueve en un eje en función del tiempo.....	29
Figura 24: Velocidad media	29
Figura 25: Velocidad instantánea	29
Figura 26: Aceleración media	30
Figura 27: Aceleración instantánea.....	30
Figura 28: Aceleración instantánea en un movimiento rectilíneo uniformemente acelerado	31
Figura 29: Aceleración instantánea en un movimiento rectilíneo uniformemente acelerado	31
Figura 30: Velocidad instantánea en un movimiento rectilíneo uniformemente acelerado...	31
Figura 31: Velocidad media independientemente de que la aceleración sea o no constante	32
Figura 32: Velocidad media si la aceleración es constante.....	32
Figura 33: Velocidad media si la aceleración es constante.....	32
Figura 34: Posición en función del tiempo en un movimiento rectilíneo uniformemente acelerado	32
Figura 35: Representación gráfica de la aceleración no constante frente al tiempo.....	33
Figura 36: Variación de velocidad instantánea en un intervalo Δt	34
Figura 37: Variación de la velocidad entre t_1 y t_2 a partir de la aceleración instantánea	34
Figura 38: Variación de posición en un intervalo Δt	34
Figura 39: Desplazamiento entre t_1 y t_2 a partir de la velocidad instantánea.....	34
Figura 40: Esquema de la arquitectura de una FPGA donde se señalan los bloques lógicos configurables, los bloques de entrada/salida y las interconexiones programables.....	36
Figura 41: Interfaz del acelerómetro ADXL362	37
Figura 42: Patrones correspondientes a los números decimales para un dígito de un visualizador 7 segmentos.....	38
Figura 43: Ánodos y cátodos de los siete LEDs que forman cada dígito y conjunto de los 8 dígitos	39
Figura 44: Módulo de comunicación de puerto serie inalámbrico HC-12	40
Figura 45: Principio de funcionamiento del módulo HC-12	40

Figura 46: Configuración del protocolo SPI con varios esclavos empleando en modo regular	44
Figura 47: Protocolo UART con bus de datos	46
Figura 48: Procesos principales que realiza el diseño hardware	47
Figura 49: Bloques principales que forman el diseño de la primera versión hardware de la primera etapa “Adquisición de datos”	47
Figura 50: Esquema del funcionamiento del diseño de la primera versión hardware de la primera etapa “Adquisición de datos”	48
Figura 51: Representación gráfica de la aceleración en el eje X para un movimiento en la dirección del eje X en sentido positivo y negativo (en valores binarios) en función del tiempo (en valores muestreados, siendo cada unidad de valor muestreado equivalente al periodo de muestreo, es decir, a la inversa de la frecuencia de muestreo)	60
Figura 52: Representación gráfica de la aceleración en el eje Y para un movimiento en la dirección del eje Y en sentido positivo y negativo (en valores binarios) en función del tiempo (en valores muestreados, siendo cada unidad de valor muestreado equivalente al periodo de muestreo, es decir, a la inversa de la frecuencia de muestreo)	61
Figura 53: Representación gráfica de la aceleración en el eje Z para un movimiento en la dirección del eje Z en sentido negativo y positivo (en valores binarios) en función del tiempo (en valores muestreados, siendo cada unidad de valor muestreado equivalente al periodo de muestreo, es decir, a la inversa de la frecuencia de muestreo)	61
Figura 54: Procesos principales que realiza el diseño hardware	62
Figura 55: Representación gráfica de las distancias recorridas calculadas en el procesado para movimientos lentos hacia la derecha de sólo ida (en valores binarios) en función de la medición realizada (adimensional)	71
Figura 56: Representación gráfica de las distancias recorridas calculadas en el procesado para movimientos a velocidad media hacia la derecha de sólo ida (en valores binarios) en función de la medición realizada (adimensional)	72
Figura 57: Representación gráfica de las distancias recorridas calculadas en el procesado para movimientos rápidos hacia la derecha de sólo ida (en valores binarios) en función de la medición realizada (adimensional)	73
Figura 58: Representación gráfica de las distancias recorridas calculadas en el procesado para movimientos a velocidad lenta, media y rápida hacia la derecha de sólo ida (en valores binarios) en función de la medición realizada (adimensional)	74
Figura 59: Representación gráfica de la aceleración (rojo), velocidad (verde) y posición (azul) para un movimiento de ida (en valores binarios) en función del tiempo (en valores muestreados, siendo cada unidad de valor muestreado equivalente al periodo de muestreo, es decir, a la inversa de la frecuencia de muestreo)	75
Figura 60: Representación gráfica de las distancias finales recorridas calculadas en el procesado para movimientos lentos hacia la derecha y hacia la izquierda (ida y vuelta) (en valores binarios) en función de la medición realizada (adimensional)	76

Figura 61: Representación gráfica de las distancias finales recorridas calculadas en el procesado para movimientos a velocidad media hacia la derecha y hacia la izquierda (ida y vuelta) (en valores binarios) en función de la medición realizada (adimensional)	77
Figura 62: Representación gráfica de las distancias finales recorridas calculadas en el procesado para movimientos rápidos hacia la derecha y hacia la izquierda (ida y vuelta) (en valores binarios) en función de la medición realizada (adimensional).....	78
Figura 63: Representación gráfica de las distancias finales recorridas calculadas en el procesado para movimientos a velocidad lenta, media y rápida hacia la derecha y hacia la izquierda (ida y vuelta) (en valores binarios) en función de la medición realizada (adimensional)	79
Figura 64: Representación gráfica de aceleración (rojo), velocidad (verde) y posición (azul) para un movimiento de ida y vuelta (en valores binarios) en función del tiempo (en valores muestreados, siendo cada unidad de valor muestreado equivalente al periodo de muestreo, es decir, a la inversa de la frecuencia de muestreo)	80
Figura 65: Procesos principales que realiza el diseño hardware	85
Figura 66: Esquema de la segunda etapa “Procesado de datos”	90
Figura 67: Esquema de la tercera etapa “Presentación de los resultados”.....	98
Figura 68: Superficie de la primera aplicación	110
Figura 69: Superficie de la segunda aplicación	114
Figura 70: Esquema del sistema empleando el módulo HC-12 para transmitir de forma inalámbrica	116

ÍNDICE DE TABLAS

Tabla 1: Características de los materiales que forman los sensores piezoeléctricos	19
Tabla 2: Descripción del módulo “general.vhd”	49
Tabla 3: Descripción del módulo “sensor.vhd”	50
Tabla 4: Descripción del módulo “ADXL362Ctrl.vhd”	51
Tabla 5: Descripción del módulo “SPI_if.vhd”	52
Tabla 6: Descripción del módulo “control.vhd”	53
Tabla 7: Descripción del módulo “mux4a1.vhd”	54
Tabla 8: Descripción del módulo “absoluto.vhd”	54
Tabla 9: Descripción del módulo “signo2bcd.vhd”	55
Tabla 10: Descripción del módulo “bin2bcd.vhd”	56
Tabla 11: Descripción del módulo “bcd2ascii.vhd”	57
Tabla 12: Descripción del módulo “ascii_selector.vhd”	58
Tabla 13: Descripción del módulo “uart.vhd”	59
Tabla 14: Análisis de datos correspondientes a la segunda medición	67
Tabla 15: Análisis de datos correspondientes a la tercera medición	70
Tabla 16: Descripción del módulo “adquisicion.vhd”	87
Tabla 17: Descripción del módulo “adquisicion_procesado.vhd”	88
Tabla 18: Descripción del módulo “procesado.vhd”	92
Tabla 19: Descripción del módulo “calibrado.vhd”	94
Tabla 20: Descripción del módulo “posicionado.vhd”	95
Tabla 21: Descripción del módulo “filtro.vhd”	96
Tabla 22: Descripción del módulo “integracion.vhd”	96
Tabla 23: Descripción del módulo “escalado.vhd”	97
Tabla 24: Descripción del módulo “presentacion.vhd”	99
Tabla 25: Descripción del módulo “binCa2_to_4bcd.vhd”	101
Tabla 26: Descripción del módulo “binary_to_bcd.vhd”	102

Tabla 27: Descripción del módulo “visu7hex8.vhd”	103
Tabla 28. Descripción del módulo “acel2pos.vhd”	106
Tabla 29: Descripción del módulo “demo01.vhd”	111
Tabla 30: Descripción del módulo “demo01_zona.vhd”	112
Tabla 31: Descripción del módulo “demo01_led.vhd”	113

GLOSARIO DE TÉRMINOS, SIGLAS Y ABREVIATURAS

ADC: (del inglés Analog-to-Digital Converter) conversor de analógico a digital.

ASIC: (del inglés Application Specific Integrated Circuit) circuito integrado de aplicación específica.

CLB: (del inglés Configurable Logic Blocks) bloques lógicos configurables.

CPLD: (del inglés Complex Programmable Logic Device) dispositivo lógico programable complejo.

DAC: (del inglés Digital-to-Analog Converter) conversor de digital a analógico.

DDR: (del inglés Double Data Rate) tasa de transferencia de datos doble. Un bus que funciona con una tasa de transferencia de datos doble transfiere datos en los flancos ascendentes y descendentes de la señal de reloj.

FFT: (del inglés Fast Fourier Transform) Transformada Rápida de Fourier.

FPGA: (del inglés Field Programmable Gate Array) matriz de puertas lógicas programables en campo.

HDL: (del inglés Hardware Description Language) lenguaje de descripción de hardware.

IEEE: (del inglés Institute of Electrical and Electronic Engineers) Instituto de Ingenieros Eléctricos y Electrónicos.

IOB: (del inglés Input/Output Blocks) bloques de entrada/salida.

LED: (del inglés Light-Emitting Diode) diodo emisor de luz.

LVDT: (del inglés Linear Variable Differential Transformer) transformador diferencial de variación lineal.

MCU: (del inglés Microcontroller Unit) microcontrolador.

MEMS: (del inglés Micro Electro Mechanical Systems) sistemas microelectromecánicos.

SPI: (del inglés Serial Peripheral Interface) Interfaz de Periféricos en Serie.

SRAM: (del inglés Static Random Access Memory) memoria estática de acceso aleatorio (o RAM estática).

UART: (del inglés Universal Asynchronous Receiver-Transmitter) Transmisor-Receptor Asíncrono Universal.

VHDL: (del inglés Very high speed integrated circuits Hardware Description Language) lenguaje de descripción de hardware de circuitos integrados de muy alta velocidad.

VHSIC: (del inglés Very High Speed Integrated Circuits) circuitos integrados de muy alta velocidad.

1. INTRODUCCIÓN

1.1. Justificación

La medición del desplazamiento y la determinación de la posición de objetos son algunos de los problemas más recurrentes a nivel industrial, ya que ambas son necesarias para realizar tareas básicas en diversos campos, como procesos de fabricación, automatización y logística, entre otros.

Actualmente, para dar solución a estas necesidades se emplean diversos métodos de medida que recurren a diferentes tipos de sensores. Según la manera en la que realizan la medida estos sensores se pueden clasificar en dos grandes grupos:

- Con contacto. Es necesario que se produzca el contacto que mueva un engranaje o vástago para generar la señal asociada al movimiento o presencia del objeto. Algunos transductores de este tipo son los potenciómetros, LVDT, reglas codificadas...
- Sin contacto. No existe contacto físico entre el transductor y el elemento medido. Se basan en ondas transmitidas entre ambos elementos. Algunos transductores de este tipo son los detectores de proximidad inductivos o capacitivos, ultrasonidos, láser, radar...

Todos estos transductores tienen características muy variadas independientemente del grupo al que pertenezcan, pero, en general, todos los sensores sin contacto tienen una vida útil mayor que los que necesitan contacto (debido a que no hay piezas que sufran desgaste por contacto durante el funcionamiento). Como contrapartida, los sistemas sin contacto tienden a ser más sofisticados y su coste suele ser también mayor.

A raíz de esta situación, este trabajo pretende estudiar la viabilidad de un método de medida alternativo a los anteriores. Un método que aúne ambos aspectos positivos:

- Una reducción significativa en el coste del sensor.
- Una mayor vida útil, al realizar la medida de forma indirecta evitando desgaste por contacto.

El sistema que se propone emplea como sensor un acelerómetro de tres ejes, por lo que el coste será mucho más bajo que el de la mayoría de los demás transductores empleados en otros métodos. Por otro lado, como se trata de un método que realiza la medida de forma indirecta, no se producirá desgaste en el equipo, siendo su vida útil independiente del número de ciclos de maniobra y por tanto mayor que la de un sensor con contacto.

Para llevar a cabo el sistema de medición se decide desarrollar un diseño hardware sobre una FPGA. Se ha optado por implementar las distintas versiones del prototipo sobre una FPGA ya que es una tecnología muy flexible que permite que se la re programe a medida que evoluciona el diseño, adaptándose a la aplicación deseada.

1.2. Objetivo

El método de medida que se propone consiste en medir la aceleración de un objeto y aplicarles a estos datos un algoritmo que realiza un proceso de integración para obtener la distancia recorrida. Este algoritmo está basado en las ecuaciones que describen la cinemática de un movimiento rectilíneo uniformemente acelerado (MRUA) (Figura 1).

$$\begin{aligned}
 \bar{r} & & a &= \text{constante} \\
 \bar{v} = \frac{d\bar{r}}{dt} & \rightarrow & v &= \int a \, dt = v_0 + at \\
 \bar{a} = \frac{d\bar{v}}{dt} & & r &= \int v \, dt = \int (v_0 + at) \, dt = r_0 + v_0t + \frac{1}{2}at^2
 \end{aligned}$$

Figura 1: Ecuaciones que describen la cinemática de un MRUA.
Fuente: Elaboración propia.

Una vez conocida la distancia recorrida por el objeto y siendo conocida la posición inicial previa, se puede determinar la posición final con facilidad.

Por tanto, el presente trabajo tiene como objetivo el estudio y desarrollo de un sistema con la capacidad de medir el desplazamiento y determinar la posición de un objeto a partir de su posición inicial y de su aceleración.

Para ello se implementará un diseño hardware que realice los siguientes procesos principales (Figura 2):

- 1) Adquisición de datos.
- 2) Procesado de datos.
- 3) Presentación de los resultados.

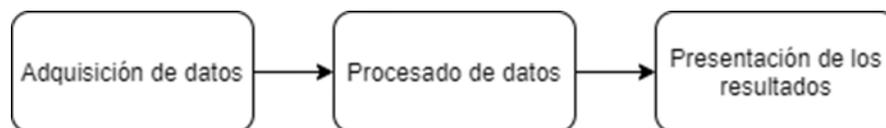


Figura 2: Procesos principales del diseño hardware.
Fuente: Elaboración propia.

El diseño hardware se llevará a cabo en VHDL empleando el entorno de diseño Xilinx ISE. Se implementará sobre una FPGA de la familia Artix-7 de Xilinx que está integrada en una placa Nexys4 DDR fabricada por la compañía Digilent. Además, esta placa contiene el acelerómetro ADXL362 de la compañía Analog Devices que se usará como sensor y el visualizador 7 segmentos en el que se mostrarán los resultados (Figura 3).

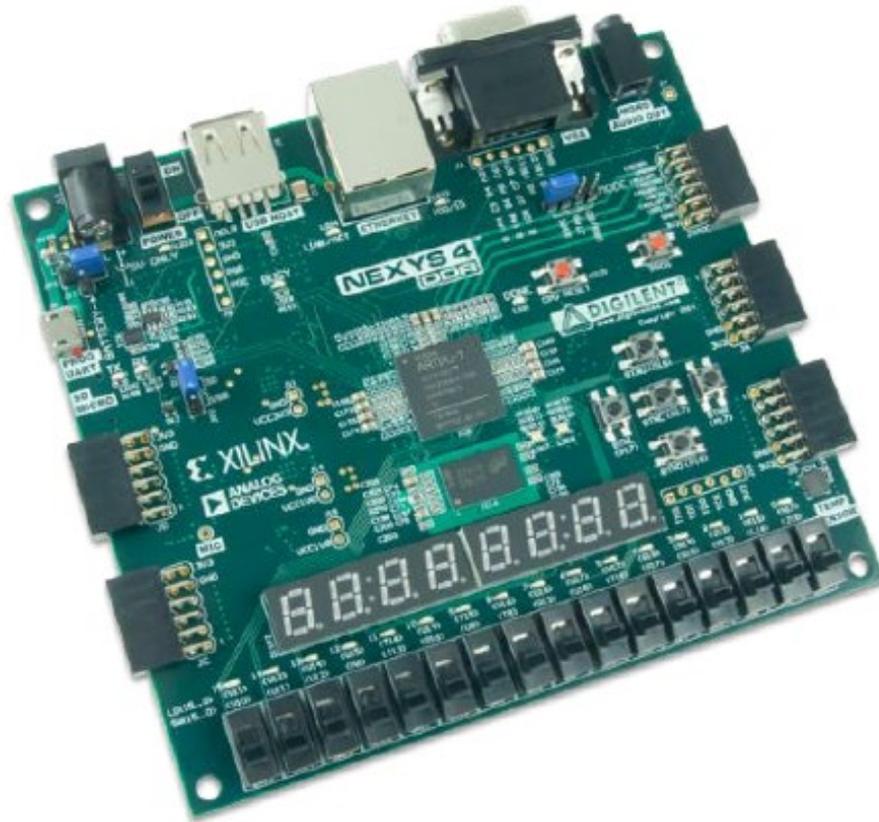


Figura 3: Placa Nexys4 DDR de Digilent.
Fuente: [4] Nexys4 DDR FPGA Board Reference Manual.

1.3. Hipótesis

El trabajo parte de la hipótesis de que las muestras tomadas por el acelerómetro ADXL362 en el proceso de “Adquisición de datos” formarán un conjunto de datos con la suficiente calidad para obtener un resultado óptimo tras el “Procesado de datos”.

Por este motivo, el primer paso es comprobar la validez de esta hipótesis. Para ello se realiza una primera versión hardware de la primera etapa “Adquisición de datos” (Figura 4) que envíe los datos captados por el acelerómetro a un ordenador, donde se realizará un estudio en Excel.

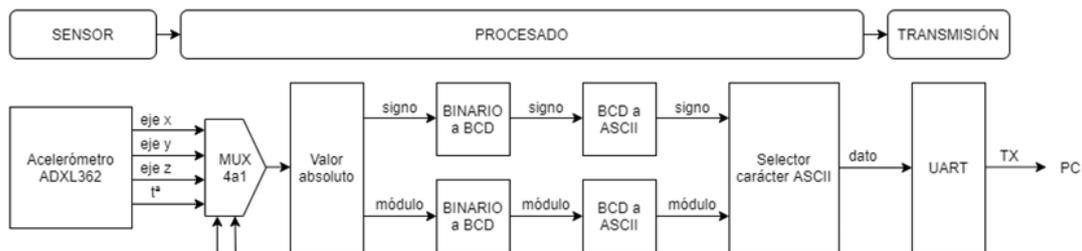


Figura 4: Esquema de la primera versión hardware de la primera etapa “Adquisición de datos”.
Fuente: Elaboración propia.

Los resultados de este primer estudio son positivos. El acelerómetro, al tomar datos de la aceleración de un movimiento en los tres ejes del espacio (eje X, eje Y y eje Z), permite estudiar el vector aceleración al completo, aunque el movimiento sólo se produzca en la dirección de uno de los ejes.

De esta forma, se puede comprobar que, cuando el movimiento se realiza en la dirección de un eje concreto, la componente del vector aceleración de ese mismo eje se comporta de la manera esperada (Figura 5), mientras que las otras dos componentes de los otros dos ejes, para ese mismo movimiento, presentan datos aleatorios y confusos.

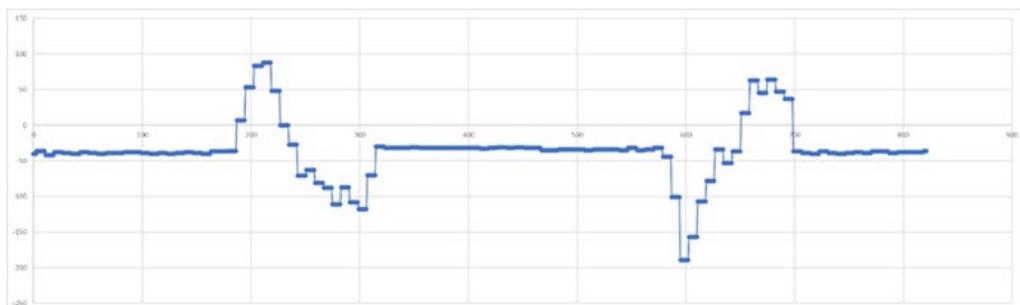


Figura 5: Representación gráfica de la aceleración en el eje X para un movimiento en la dirección del eje X en sentido positivo y negativo (en valores binarios) en función del tiempo (en valores muestreados, siendo cada unidad de valor muestreado equivalente al periodo de muestreo, es decir, a la inversa de la frecuencia de muestreo).

Fuente: Elaboración propia.

Siendo estos resultados muy interesantes, todavía no se puede concluir que la hipótesis sea válida. Es necesario comprobar que, tras aplicar el proceso de integración del “Procesado de datos”, los resultados obtenidos sean satisfactorios.

Para ello, se implementa en Matlab una primera versión software de la segunda etapa “Procesado de datos”. Se ha optado por realizarla primero en software, porque así resulta más sencillo y rápido implementar el algoritmo que realiza el proceso de integración. El script desarrollado, además de llevar a cabo el proceso de integración basado en las ecuaciones que describen la cinemática de un MRUA, representa de forma gráfica los resultados obtenidos tras aplicar a los datos de aceleración el algoritmo que realiza el proceso de integración (Figura 6).

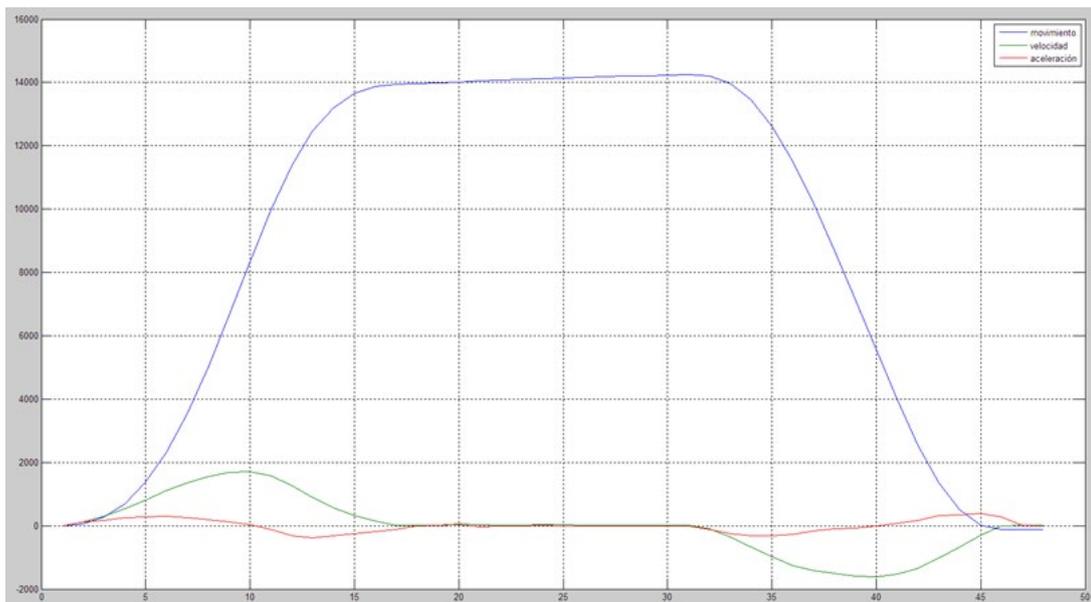


Figura 6: Representación gráfica de aceleración (rojo), velocidad (verde) y posición (azul) (en valores binarios) en función del tiempo (en valores muestrados, siendo cada unidad de valor muestrado equivalente al periodo de muestreo, es decir, a la inversa de la frecuencia de muestreo).

Fuente: Elaboración propia.

Tras aplicar el proceso de integración a nuevos datos obtenidos para distintas situaciones (distintas distancias, velocidades, superficies de desplazamiento...), se realiza un análisis mucho más riguroso a los resultados. Se determina que la hipótesis se cumple, aunque existen algunas restricciones que afectarán a las posibles aplicaciones.

Principalmente, cuando el objeto varía su velocidad ha de hacerlo con una aceleración suficientemente elevada para que estos datos no queden enmascarados por el ruido de la señal. Además, al cumplir esta condición es posible filtrar dicho ruido sin afectar a la información de la señal, evitando que afecte en el procesado.

La segunda gran restricción es que el sistema ha de trabajar sobre un plano horizontal, ya que de no hacerlo se introduce en los componentes X e Y del vector aceleración la proyección de la gravedad terrestre sobre estos ejes, lo que provoca que el sistema se vuelva inestable.

Finalmente, se prueban varias mejoras como la implementación de filtros para eliminar ruidos de la señal, tanto los introducidos al realizar la medición (errores aleatorios) como los propios del sistema (errores sistemáticos). Para eliminar los últimos se realiza un proceso previo de calibración que permita determinar el error de cero de la señal y así poder anulárselo a la misma.

1.4. Resumen

Una vez comprobada la validez de la hipótesis, el siguiente paso es trasladar el algoritmo desde el lenguaje de Matlab a VHDL, es decir, pasar de software a hardware el algoritmo y los filtros desarrollados. Una vez implementados estos diseños en VHDL se obtiene la segunda etapa “Procesado de datos” (Figura 7). Además, se modifica la primera versión hardware de la primera etapa “Adquisición de datos” para que en lugar de enviar los datos al ordenador a través de un cable USB, los envíe a la segunda etapa directamente.

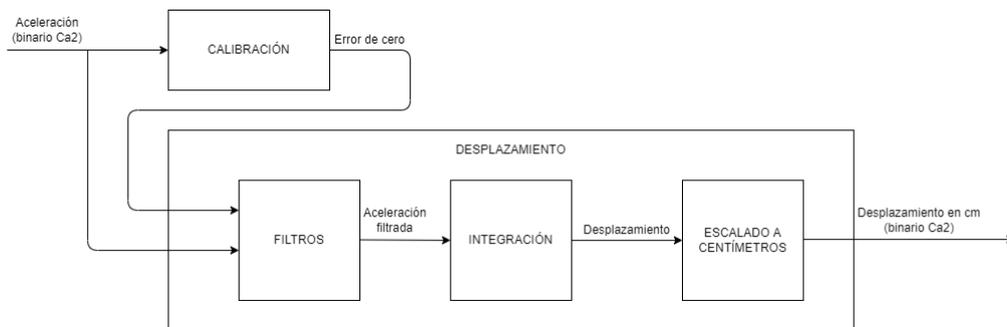


Figura 7: Esquema de la segunda etapa “Procesado de datos”.

Fuente: Elaboración propia.

Para poder visualizar los resultados obtenidos tras el procesamiento se ha optado por emplear el visualizador 7 segmentos de la propia placa Nexys4 DDR. La placa cuenta con dos módulos de 4 dígitos cada uno. Por tanto, se empleará uno de los módulos para representar el desplazamiento en el eje X y el otro para el desplazamiento en el eje Y. El dígito más significativo de cada módulo se empleará para representar el signo menos (“-”) cuando sea necesario. De esta forma, queda implementada la tercera etapa “Presentación de los resultados” (Figura 8).

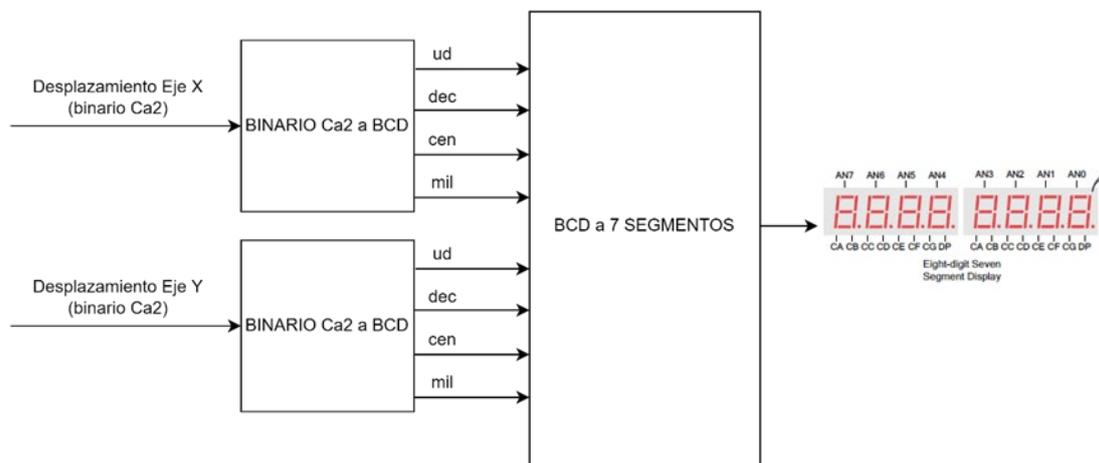


Figura 8: Esquema de la tercera etapa “Presentación de los resultados”.

Fuente: Elaboración propia.

Por último, se instancian los componentes que representan cada una de las tres etapas en la entidad principal y se realizan las conexiones pertinentes. También se crea el fichero de restricción de pines que asigna las entradas y salidas del sistema al hardware de la placa. Se genera el archivo final y se carga en la FPGA. Se comprueba que funciona correctamente y se implementan una serie de aplicaciones para poner a prueba el método de medición.

La primera de las aplicaciones consiste en que el sistema determine la posición en la que se halla la placa al moverse sobre una superficie con zonas delimitadas (Figura 9). La placa indicará en que zona se encuentra iluminando uno de sus LEDs, de tal forma que cada LED irá asociado a una zona distinta.

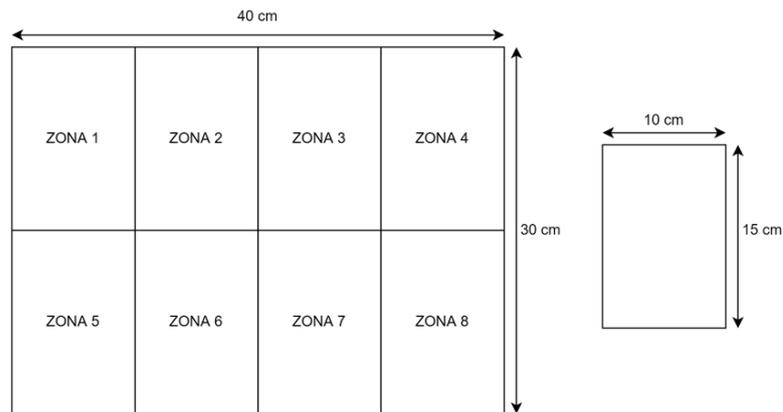


Figura 9: Superficie de la primera aplicación.
Fuente: Elaboración propia.

Dado que en la primera aplicación el sistema se comporta de forma satisfactoria, se aumenta la dificultad incrementando el número de zonas en la misma superficie (Figura 10). Los resultados también son positivos.

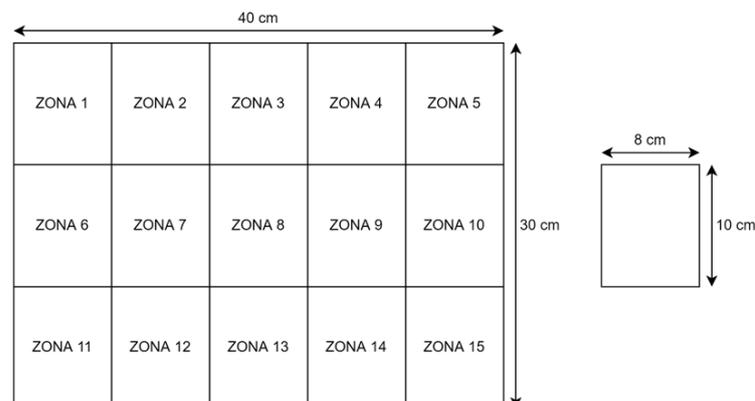


Figura 10: Superficie de la segunda aplicación.
Fuente: Elaboración propia.

Finalmente, para emular una aplicación real donde se evite la presencia de un operario in situ o la necesidad de cableado adicional para la transmisión de la información, el sistema transmite los datos de forma inalámbrica empleando el módulo HC-12 (mediante UART) a otra placa que recibe en que zona se encuentra el objeto (Figura 11).

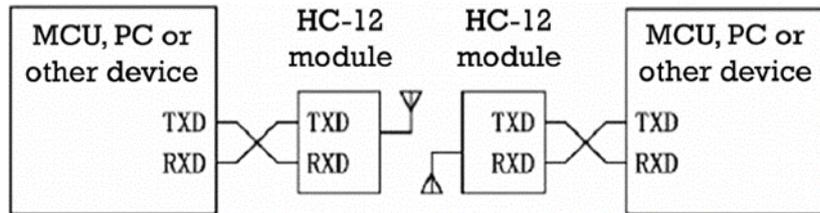


Figura 11: Esquema del sistema empleando el módulo HC-12 para transmitir de forma inalámbrica.
Fuente: HC-12 Wireless Serial Port Communication Module User Manual.

2. ESTADO DE LA CUESTIÓN

2.1. Métodos de medida de distancia y de detección de proximidad

2.1.1. Introducción

La medición de distancia se ocupa de determinar la longitud de una línea que une dos puntos en el espacio. Se suele tomar una serie de medidas de distancia para determinar la forma, la orientación o la posición de un objeto. Por ello, además de la distancia, es importante considerar la distribución espacial y temporal. Para llevar a cabo estas mediciones se emplean sensores de distancia.

Además de medir el desplazamiento, el prototipo del presente trabajo trata de determinar la posición final de un objeto. En la actualidad, la manera más común de determinar la posición de un objeto es a través de la detección de proximidad de éste. Para ello se emplean sensores de proximidad, los cuales no sólo permiten conocer la posición de un objeto, sino que también permiten determinar su orientación [6].

2.1.2. Clasificaciones de los métodos

Atendiendo a distintos aspectos se pueden realizar varias clasificaciones de los métodos de medida de distancia y de detección de proximidad. En este subapartado se mencionarán las clasificaciones más comunes.

Con contacto y sin contacto

Los métodos con contacto son aquellos que necesitan que se produzca el contacto que mueva un engranaje o vástago para generar la señal asociada a la detección de la presencia del objeto. De esta manera se puede medir la distancia entre una posición de referencia y un punto de un objeto empleando un dispositivo mecánico calibrado.

Los métodos mecánicos basados en contacto se utilizan ampliamente en la industria y pueden ser extremadamente precisos. La principal desventaja es que generalmente están restringidos a distancias y volúmenes de trabajo pequeños. Esto es debido a que, al aumentar las distancias a medir, el tamaño y la tolerancia mecánica requerida por la máquina hacen que los diseños no sean prácticos. Además, los métodos mecánicos son demasiado lentos para realizar múltiples mediciones en rápida sucesión, como se requiere en ciertas aplicaciones que demandan grandes conjuntos de datos distribuidos espacial o temporalmente.

Los métodos sin contacto para realizar medidas de distancia y determinar la posición son muy variados. En ellos no existe contacto físico entre el sensor y el elemento medido. Se basan en ondas transmitidas entre ambos elementos.

Activos o pasivos

Los métodos de medida de distancia y de detección de proximidad sin contacto pueden clasificarse en activos o pasivos.

Los métodos activos implican alguna forma de energía controlada (ondas o campos) que relaciona la ubicación de referencia conocida con la ubicación de destino desconocida. La fuente de energía suele estar asociada con la ubicación de referencia, aunque existen casos donde ocurre lo contrario, la fuente se asocia al objeto que contiene el punto de interés.

Los métodos pasivos se basan en una fuente de energía externa (por ejemplo, luz solar) para hacer que el objetivo sea detectable.

Los métodos activos suelen simplificar el problema de medición, ya que al permitir elegir la forma de energía y el nivel de potencia de la fuente logran un mayor grado de control. Un uso común de estos métodos es en mediciones a través de materiales que son mecánica u ópticamente impenetrables (por ejemplo, imágenes médicas, donde formas de energía como rayos X o ultrasonidos se utilizan para construir mapas superficiales o volumétricos de órganos y huesos).

Por su parte, los pasivos, aunque no ofrecen el mismo grado de control y flexibilidad que los activos, si tienen ciertas ventajas. Primero, su existencia no puede ser detectada por otro sistema de detección remota debido a que no emiten energía, lo que los hace interesantes en aplicaciones militares. En segundo lugar, pueden recopilar información más rápidamente porque no están limitados por la velocidad a la que pueden dirigir una fuente de energía hacia un objeto. Por último, la ausencia de una fuente de energía dirigida es una simplificación que puede reducir notablemente el tamaño, el costo y la complejidad del hardware de un dispositivo, aunque a expensas de una mayor complejidad del procesamiento de señales [6].

2.1.3. Principios y técnicas de los métodos sin contacto

Aunque existen muchos métodos sin contacto, prácticamente la totalidad se basan en alguno de estos tres principios básicos:

- La energía se propaga a una velocidad finita conocida (por ejemplo, la velocidad de la luz o la velocidad del sonido en el aire).
- La energía se propaga en línea recta a través de un medio homogéneo.
- Los campos de energía cambian de manera continua, monótonamente decreciente y predecible con la distancia de su fuente.

De estos tres principios básicos se derivan las tres técnicas que se emplean para desarrollar la mayoría de métodos sin contacto: el tiempo de vuelo, la triangulación y las técnicas basadas en campo.

Tiempo de vuelo

Los sistemas basados en el tiempo de vuelo pueden ser unidireccionales o bidireccionales, es decir, de “ida y vuelta” (por ejemplo, eco).

Los sistemas de “ida y vuelta” miden el tiempo que tarda una forma de energía emitida en viajar desde una fuente de referencia a un objeto parcialmente reflectante y viceversa. Como se conoce la velocidad de propagación de la energía, se obtiene la distancia en función del tiempo medido. Dependiendo de si se utilizan frecuencias de radio, de luz o de energía sonora, los dispositivos resultantes se conocen como RADAR, LIDAR y SONAR.

Los sistemas unidireccionales se basan en la misma idea, pero a diferencia de los bidireccionales, transmiten una señal desde el extremo de referencia y la reciben en el extremo objetivo, siendo necesaria una referencia de sincronización para establecer el tiempo de vuelo.

Una característica interesante de los sistemas basados en tiempo de vuelo es que su error es esencialmente constante en su rango operativo.

Triangulación

La técnica de la triangulación (Figura 12) se basa en la idea de que, si se conoce la longitud de un lado de un triángulo y dos de sus ángulos, se puede calcular la longitud de los otros dos lados.

El lado conocido se llama línea de base. Desde cada uno de los dos puntos extremos de la línea de base hasta el punto objetivo se forma una línea recta. A estas dos líneas se las conoce como líneas de detección. Si se pueden determinar los ángulos formados entre estas líneas y la línea de base, la distancia se calcula como:

$$R = \frac{b \sin \alpha \sin \beta}{\sin(\beta - \alpha)}$$

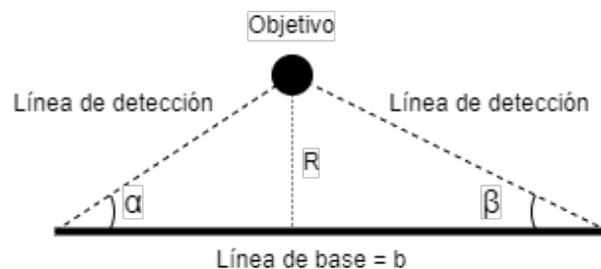


Figura 12: Técnica de la triangulación.

Fuente: Elaboración propia basado en [6] Measurement, Instrumentation, and Sensors Handbook.

Es importante tener en cuenta que la capacidad para estimar el rango de los sistemas basados en triangulación disminuye con el cuadrado del rango que se mide.

Técnicas basadas en campo

La intensidad de cualquier campo de energía varía en función de la distancia a su fuente. Además, los campos presentan características vectoriales. Por tanto, si se conoce la ubicación de la fuente que crea el campo y las características espaciales que produce la misma son predecibles, las mediciones de campo contienen información que puede usarse para inferir la distancia desde la fuente.

Una distinción interesante entre las técnicas basadas en campo y las basadas en ondas, es que las primeras no dependen de la propagación y conversión de energía, aunque emplean campos de energía. Es decir, pueden emplear campos estacionarios, como los generados por un imán o una carga estática. Dichos campos codifican la información de posición por su propia forma.

Las técnicas basadas en campo tienen algunos problemas básicos que limitan sus aplicaciones. Para empezar, las características de la mayoría de los campos suelen estar influenciadas por objetos o materiales cercanos y no siempre se puede garantizar que influyan de manera constante. En segundo lugar, la variación de campos a través del espacio es altamente no lineal (normalmente, la relación es inversamente proporcional al cuadrado o al cubo), lo que implica que la sensibilidad de una medición se ve muy afectada por la proximidad a la fuente [6].

2.2. Sensores de distancia y de proximidad

2.2.1. Clasificación de los sensores

Atendiendo a los fenómenos que posibilitan su funcionamiento, los sensores de distancia y de proximidad se clasifican en los siguientes tipos.

Sensores resistivos

Los sensores de desplazamiento resistivos, también conocidos como potenciómetros, son dispositivos electromecánicos que contienen un elemento resistivo fijo sobre el que se desliza un contacto móvil que conduce electricidad. El elemento resistivo queda dividido en el punto de contacto con el elemento móvil. Para medir el desplazamiento se conecta al potenciómetro como un divisor de tensión (Figura 13). De esta manera, la salida del circuito es un voltaje que varía en función de la posición del elemento móvil.

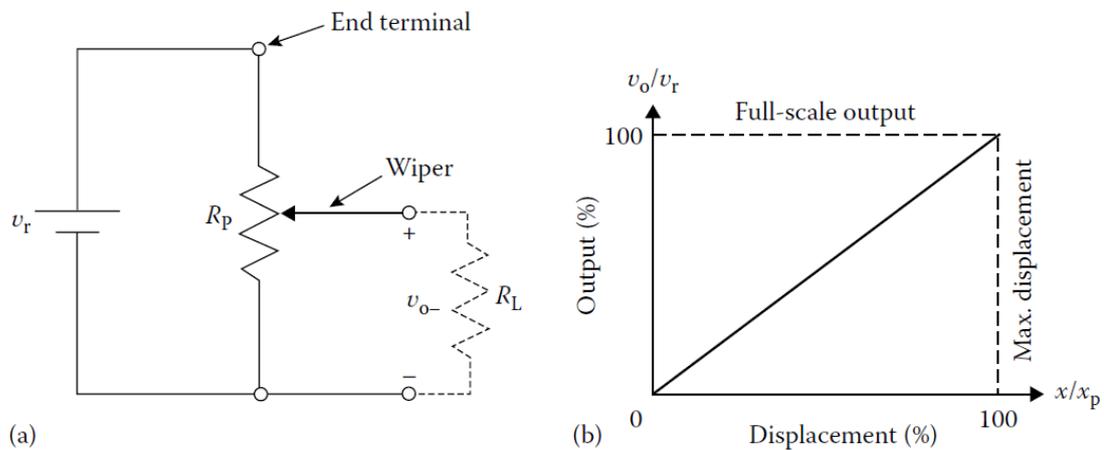


Figura 13: (a) Circuito de un potenciómetro como un divisor de tensión. (b) Representación gráfica del voltaje a la salida del circuito en función del desplazamiento del elemento móvil.

Fuente: [6] Measurement, Instrumentation, and Sensors Handbook.

Son sensores de bajo coste, fácil uso y no requieren de electrónica adicional. En contrapartida, su ancho de banda es limitado, tienen carga por fricción, carga inercial y al necesitar contacto para funcionar sufren desgaste por uso.

Los potenciómetros pueden ser lineales o angulares.

Sensores inductivos

Los sensores inductivos (Figura 14) son sensores que se basan en los principios del electromagnetismo. Constan de un circuito magnético hecho de un núcleo ferromagnético con una bobina enrollada en él. La bobina actúa como fuente magnetomotriz que conduce el flujo magnético a través del núcleo y el entrehierro. En el entrehierro se provoca un aumento de la reluctancia y la correspondiente disminución del flujo magnético. De esta manera, pequeñas variaciones en el entrehierro provocadas por el desplazamiento o la presencia de objetos metálicos generan variaciones medibles en la inductancia.

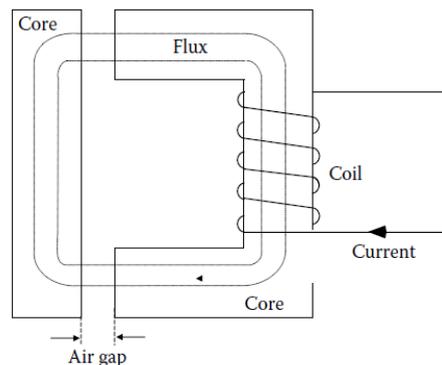


Figura 14: Circuito de un sensor inductivo.

Fuente: [6] Measurement, Instrumentation, and Sensors Handbook.

Los sensores inductivos detectan la presencia de objetos metálicos siendo su rango de detección variable en función del tipo de metal.

Su funcionamiento se basa en que el objeto, al acercarse al campo magnético de alta frecuencia generado por la bobina en el circuito de oscilación, tiene una inducción o corriente de Foucault inducida en él creando un campo magnético opuesto que reduce efectivamente la inductancia del sensor inductivo. Los sensores inductivos funcionan mediante dos técnicas.

La primera técnica consiste en que a medida que el objeto se acerca al sensor, el flujo de corriente de inducción aumenta, lo que incrementa la carga en el circuito de oscilación. Esto hace que su oscilación se atenúe o se detenga. El sensor detecta este cambio en el estado de oscilación con un circuito de detección de amplitud y emite una señal de detección.

La segunda técnica emplea el cambio en la frecuencia de la oscilación resultante provocada por la presencia de un objeto, en lugar del cambio en la amplitud. Un objeto de metal no ferroso (como el aluminio o el cobre) que se acerca al sensor hace que la frecuencia de oscilación aumente, mientras que un objeto de metal ferroso (como el hierro o el acero) hace que la frecuencia de oscilación disminuya. El cambio de la frecuencia de oscilación con respecto a una frecuencia de referencia hace que el estado de salida del sensor cambie.

Los sensores inductivos se utilizan ampliamente en la industria en un gran número de aplicaciones. Son robustos y compactos. Además, se ven menos afectados por factores ambientales (por ejemplo, humedad, polvo) que los sensores capacitivos.

Sensores capacitivos

Los sensores capacitivos son sensores que trabajan detectando la posición o cambio de posición de objetos metálicos y no metálicos, en forma de polvo, granulado, líquido y sólido, con mediciones de alta resolución.

Se basan en cambios en la capacitancia en respuesta a variaciones físicas. El elemento sensor básico de estos sensores consta de dos electrodos paralelos con una capacitancia que varía en función de la distancia entre los electrodos, el área de superficie de los mismos y la permitividad del dieléctrico que se encuentra entre ambos. Los sensores de desplazamiento capacitivos se clasifican en tres clases dependiendo cuál de las anteriores características es la que varía, habiendo sensores de distancia variable, de área variable y de dieléctrico variable.

Los dispositivos son generalmente similares a los sensores inductivos, excepto que las bobinas de detección del sensor inductivo se sustituyen por una placa de detección capacitiva. La placa forma un condensador con el objeto y la capacitancia varía con la distancia al objeto. La capacitancia de detección determina la frecuencia del oscilador, que se controla para conmutar el estado de salida cuando se cruza el umbral de frecuencia.

Los sensores capacitivos se emplean como solución a múltiples problemas industriales, científicos y de ingeniería. Satisfacen los requisitos de una amplia variedad de aplicaciones en las que se necesitan una alta linealidad y amplios rangos (desde unos pocos centímetros hasta un par de nanómetros).

Sensores magnéticos

Los sensores magnéticos se emplean para medir la posición y la velocidad de componentes metálicos en movimiento.

Existe una gran variedad de sensores magnéticos. Pueden ser activos (necesitan una fuente de alimentación), como un sensor de efecto Hall, o pasivos (no necesitan una fuente de alimentación), como un sensor de reluctancia variable.

El sensor de reluctancia variable mide los cambios en la reluctancia magnética (que es análoga a la resistencia eléctrica en un circuito eléctrico). Consta de un imán permanente, una pieza polar y una bobina de detección encerrada en una caja. Cuando un objeto ferromagnético pasa cerca de la pieza polar provoca una variación del campo magnético. Esta variación genera a su vez una tensión de señal en la bobina de señal. La magnitud de la tensión de la

señal depende del tamaño del objeto, de su velocidad y del tamaño del hueco entre la pieza polar y el objeto. El objeto debe estar en movimiento para ser detectado por el sensor.

Otro tipo de sensor magnético es el sensor basado en el efecto Hall. Estos sensores emplean el efecto Hall para detectar la presencia de un campo magnético. El efecto Hall describe la interacción de un conductor portador de corriente y un campo magnético perpendicular al plano del conductor. Cuando el conductor portador de corriente se coloca en un campo magnético, se genera una tensión (tensión Hall) perpendicular a la corriente y al campo. La tensión Hall es proporcional a la densidad de flujo del campo magnético y requiere un blanco que esté magnetizado. Este sensor puede funcionar a velocidades de conmutación de hasta 10 kHz y puede detectar campos magnéticos tanto dinámicos como estáticos, lo que le permite detectar un objeto en una posición fija.

Sensores ópticos

Los sensores ópticos (Figura 15) utilizan la luz infrarroja o visible para detectar objetos. Tienen la ventaja de que no es necesario que el objeto sea magnético o metálico, únicamente tiene que obstruir o reflejar la luz.

Los sensores ópticos basan su funcionamiento en la detección de luz emitida o reflejada por un objeto. La luz está asociada a la presencia o al movimiento de un objeto mediante distintas técnicas dependiendo del método empleado (método mediante deflexión de haz óptico o método mediante fibra óptica, entre otros). De esta manera, la detección de luz se traduce en una señal eléctrica proporcional en la salida del sensor óptico.

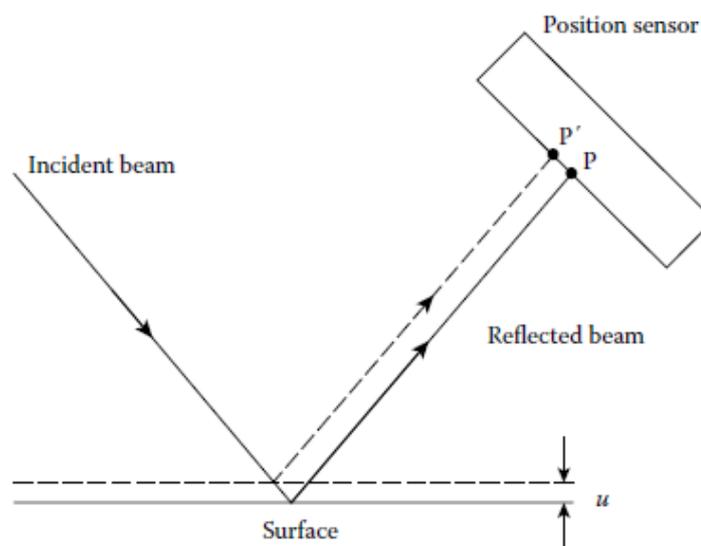


Figura 15: Diagrama esquemático de un sensor de desplazamiento mediante deflexión de haz óptico.

Fuente: [6] Measurement, Instrumentation, and Sensors Handbook.

Sensores ultrasónicos

El ultrasonido es una onda acústica con una frecuencia superior al rango audible del oído humano, que es de 20 kHz. El uso de ultrasonidos para la medición de distancias se basa en la técnica del tiempo de vuelo. Para ello, se transmite a través de un medio un pulso de ultrasonido y cuando este llega a otro medio se refleja total o parcialmente. Se mide el tiempo transcurrido desde la emisión hasta la detección del pulso reflejado. Una vez conocido el tiempo transcurrido y la velocidad a la que viaja el pulso, se calcula la distancia.

Los sensores ultrasónicos (Figura 16) están formados por subsistemas. En primer lugar, un oscilador genera una señal eléctrica con una frecuencia típica de 40 kHz. Esta señal eléctrica se transforma en vibraciones mecánicas de la misma frecuencia en el transmisor. Estas vibraciones generan ondas de sonido que son reflejadas por el objeto. El eco sonoro reflejado es captado por el receptor, lo que provoca una señal eléctrica. El transmisor y el receptor pueden ser el mismo dispositivo. El tiempo entre la transmisión y la recepción se puede obtener mediante distintos métodos y la velocidad de propagación del sonido es previamente conocida, por lo que se puede calcular la distancia. Si la distancia indica que el objeto está en la zona de proximidad, se conmuta la salida.

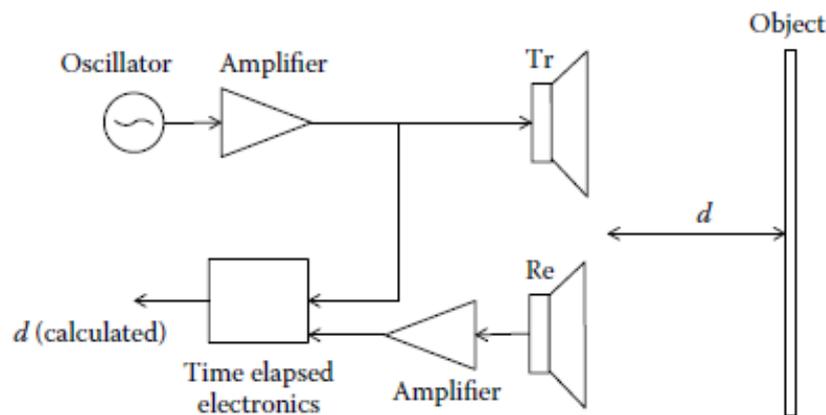


Figura 16: Subsistemas de un sensor de medida de distancia mediante ultrasonidos.

Fuente: [6] Measurement, Instrumentation, and Sensors Handbook.

Los sensores de proximidad ultrasónicos detectan objetos de cualquier tipo a distancias de hasta varios metros. Para mediciones precisas, la velocidad del sonido es un parámetro crucial que está influenciado por la presión del aire, la temperatura del aire y la composición química del aire (agua, CO₂, etc.). Medir distancias en un ambiente con grandes gradientes de temperatura puede resultar en mediciones erróneas. Otro problema de las mediciones de distancia por ultrasonidos es el ruido acústico industrial. Este puede afectar a las señales recibidas, aunque como no está correlacionado con las señales enviadas puede ser eliminado por el uso de métodos de correlación. A su favor, las ondas de ultrasonido tienen la ventaja de que son robustas frente a otras perturbaciones como la luz, el humo y la interferencia electromagnética.

Sensores piezoeléctricos

La piezoelectricidad es un fenómeno físico mediante el cual ciertos materiales desarrollan una carga eléctrica a partir de una tensión mecánica aplicada sobre los mismos, de forma directamente proporcional. Estos materiales pueden mostrar el efecto contrario, es decir, se deformarán mecánicamente de forma proporcional si se les aplica un campo eléctrico. El primero de estos casos se conoce como efecto piezoeléctrico directo y el segundo, como efecto piezoeléctrico inverso.

Los efectos piezoeléctricos en varios materiales han llevado a su uso en transductores electromecánicos. Los sensores piezoeléctricos empleados para medir desplazamientos se basan en el efecto piezoeléctrico directo, esto es, se obtiene un voltaje de salida que está relacionado con la deformación mecánica del sensor provocada por el movimiento del objeto a estudiar.

Los transductores piezoeléctricos están fabricados por tres tipos de materiales: cerámicas piezoeléctricas, polímeros piezoeléctricos y compuestos de cerámicas piezoeléctricas con polímeros inactivos. Las características de cada uno de estos tres tipos de materiales se recogen en la tabla siguiente (Tabla 1) [5] [6] [10].

Parámetro	Cerámicas piezoeléctricas	Polímeros piezoeléctricos	Compuestos de cerámicas piezoeléctricas con polímeros inactivos
Impedancia acústica	Alto	Bajo	Bajo
Factor de acoplamiento	Alto	Bajo	Alto
Modos espurios	Muchos	Pocos	Pocos
Constante dieléctrica	Alto	Bajo	Medio
Flexibilidad	Rígido	Flexible	Flexible
Costo	Barato	Caro	Medio

Tabla 1: Características de los materiales que forman los sensores piezoeléctricos.

Fuente: [6] Measurement, Instrumentation, and Sensors Handbook.

2.2.2. Sensores de distancia

Tanto la ciencia como la ingeniería tienen interés en medir un enorme rango de distancias, desde distancias interatómicas hasta distancias intergalácticas. Por este motivo existen una gran cantidad de sensores de distancia distintos. A continuación, se enumeran algunos de ellos:

- Sensores de distancia resistivos (potenciómetros).
- Sensores de distancia capacitivos.
- Sensores de distancia inductivos.
 - LVDT.
- Sensores de distancia magnéticos.
 - Sensores magnetostrictivos.
 - Sensores de impedancia distribuida.
 - Sensores magnetorresistivos.
 - Sensores de efecto Hall.
 - Codificadores magnéticos.
 - Synchros y resolvers.
- Sensores piezoeléctricos.
- Sensores de distancia ópticos.
 - Codificadores ópticos (optical encoders).
 - Mediante fibra óptica.
 - Mediante deflexión de haz óptico.
 - Mediante interferómetro láser.
- Sensores de distancia ultrasónicos [5] [6] [10].

2.2.3. Sensores de proximidad

Los sensores de proximidad son sensores que detectan si un objeto se encuentra o no en una determinada posición. Por este motivo se clasifican como sensores binarios, ya que solo tienen dos estados.

Los sensores de proximidad tienen la capacidad de detectar la presencia o la posición de objetos sin establecer contacto físico, por lo que se incluyen en el grupo de métodos sin contacto. Esta utilidad tiene múltiples aplicaciones y cada una de ellas requiere de unas características específicas del sensor.

Los sensores de proximidad se enmarcan en cinco grandes grupos, a saber:

- Sensores de proximidad inductivos.
- Sensores de proximidad capacitivos.
- Sensores de proximidad magnéticos.
- Sensores de proximidad ópticos.
- Sensores de proximidad ultrasónicos [5] [6] [10].

3. CONCEPTOS TEÓRICOS, MATERIALES Y TÉCNICAS

3.1. Conceptos teóricos

3.1.1. Instrumentación y medidas

Instrumentación

Los datos son la entidad básica que se necesita para desarrollar el conocimiento. Se obtienen mediante sensores, conjuntos físicos que se utilizan para observar y detectar variables del sistema. Los datos de uno o más sensores se procesan para brindar una comprensión más significativa. Al resultado de este proceso se le denomina información. A la combinación entre los sensores y los medios para procesar los datos y convertirlos en información se la denomina instrumentación.

Los instrumentos de medida, o instrumentación, son dispositivos artificiales que permiten determinar el valor de una variable física. Los instrumentos detectan variaciones en una variable física y responden generando señales proporcionales. La respuesta total de los instrumentos es una combinación de respuestas dinámicas y estáticas. Si las señales generadas cambian rápidamente, las propiedades dinámicas del instrumento serán importantes. Pero, si varían de forma lenta, las propiedades dinámicas serán despreciables.

Un instrumento se puede dividir en distintos bloques o etapas (Figura 17). Normalmente, los instrumentos tienen una estructura de bloques común. Primero, se emplea una etapa de adquisición de datos que contendrá al elemento sensor. A esta le siguen otras de acondicionamiento y procesado de la señal. Finalmente, hay una etapa de salida que se ocupa de la transmisión o visualización de la información.

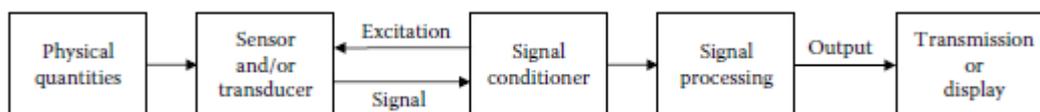


Figura 17: Diagrama de bloques de un instrumento de medida.
Fuente: [6] Measurement, Instrumentation, and Sensors Handbook.

Los instrumentos pueden ser analógicos, digitales o una combinación de ambos. A día de hoy la mayoría de los instrumentos son producidos para ser digitales por sus ventajas (por ejemplo, realizar operaciones matemáticas o almacenar y transmitir datos), aunque la parte delantera suele seguir siendo analógica. Es decir, la mayoría de los sensores y transductores de la etapa de adquisición generan señales analógicas que posteriormente se convierten a formato digital para su procesamiento.

Los instrumentos analógicos se caracterizan por trabajar con señales continuas. Un sistema analógico puro mide, transmite, muestra y almacena datos en forma analógica. El acondicionamiento de la señal se realiza integrando muchos bloques funcionales como puentes, amplificadores, filtros, osciladores, moduladores, compensadores y convertidores de nivel. La información útil se transmite mediante variaciones de amplitud, fase, frecuencia o una combinación de las tres.

Los instrumentos digitales, como ya se ha comentado anteriormente, suelen adquirir los datos de las variables físicas empleando sensores y transductores analógicos. Estas señales analógicas se convierten a digitales antes de pasar a las otras partes del sistema. Para ello se emplean convertidores analógico/digital (ADC) junto con dispositivos de muestreo y multiplexión. La conversión A/D implica tres etapas: muestreo, cuantificación y codificación.

El teorema de muestreo de Nyquist debe cumplirse durante el muestreo, es decir, “el número de muestras por segundo debe ser al menos el doble de la frecuencia más alta presente en la señal continua”. Como regla general, dependiendo de la importancia de las altas frecuencias, el muestreo debe ser de 5 a 10 veces la frecuencia más alta de la señal.

La siguiente etapa es la cuantificación, que determina la resolución de las señales muestreadas. El error de cuantificación disminuye a medida que aumenta el número de bits.

En la etapa de codificación, los valores cuantificados se convierten en números binarios para ser procesados digitalmente. Una vez en formato digital, los datos pueden procesarse aún más empleando diversas técnicas, como análisis FFT, filtrado digital, toma de decisiones secuencial o lógica, métodos de correlación, análisis de espectro y más.

En los últimos años, el rápido crecimiento de la electrónica de circuitos integrados y la disponibilidad de procesadores rentables han llevado a un progreso impresionante en instrumentación y mediciones. Esto, junto con la mejora de los métodos matemáticos y las técnicas digitales en nuevas áreas de aplicación, han permitido a la instrumentación moderna sobresalir hasta nuevas cotas.

Medidas

Una medida es un proceso de recopilación de información del mundo físico y comparación de dicha información con estándares acordados. Si se conoce el comportamiento de una variable física, se puede monitorear y evaluar su desempeño mediante métodos adecuados de detección y acondicionamiento de la señal. Para ello, se utiliza instrumentación que emplea a los sensores como los elementos primarios de captación de información.

Los sensores generan una respuesta ante la variación de la variable física. Las señales de salida del sensor de la mayoría de instrumentos modernos tienen

forma eléctrica. Esto se debe a que las señales eléctricas son fáciles de procesar, visualizar, almacenar y transmitir. Una vez se convierte la señal de salida del sensor a forma eléctrica, se puede expresar mediante una función de transferencia la relación entre las variaciones físicas y la señal de salida del sensor.

La función de transferencia es un modelo matemático que relaciona la variable física estudiada y la señal de salida del sensor. En un sistema continuo, la función de transferencia puede ser lineal o no lineal. Una relación lineal (Figura 18) puede expresarse mediante la siguiente ecuación:

$$y = a + bx \quad [3.1.1]$$

Donde

y es la señal eléctrica

x es el estímulo físico

a es la intersección en el eje y que da la señal de salida para una entrada cero.

b es la pendiente que también se conoce como sensibilidad

Figura 18: Función de transferencia lineal.

Fuente: Elaboración propia.

Una vez realizadas las mediciones, se pueden utilizar varias formas de análisis de datos para extraer la información útil de las medidas y sacar conclusiones. El análisis de datos puede incluir métodos estadísticos, ajuste de curvas, selección o descarte de subconjuntos de datos o muchas otras técnicas.

Calibración

En la sección anterior, se ha visto la función de transferencia que relaciona la variable física de entrada con la señal de salida obtenida por un sensor específico. La función de transferencia permite calibrar el sistema de medición.

Por lo general, un sensor o un sistema completo de instrumentación se calibra proporcionando una entrada física conocida al sistema y registrando la salida. Los datos se trazan en una curva de calibración (Figura 19).

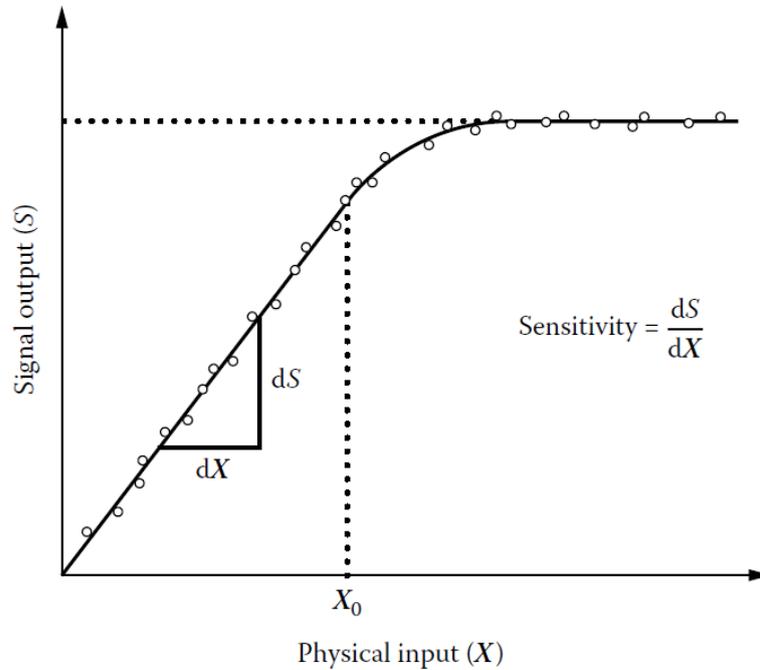


Figura 19: Curva de calibración.

Fuente: [6] Measurement, Instrumentation, and Sensors Handbook.

En la figura anterior (Figura 19), el sensor tiene una respuesta lineal para valores de entrada menores que X_0 . La sensibilidad del dispositivo es determinada por la pendiente de la curva de calibración. Para valores de entrada superiores a X_0 , la curva de calibración se vuelve menos sensible hasta alcanzar un valor límite de la señal de salida. Este comportamiento se conoce como saturación.

Por tanto, el sensor no se puede utilizar para mediciones mayores a su valor de saturación (X_0). Además, algunos sensores tampoco servirán para valores muy pequeños de entrada porque tampoco cumplirán la relación de linealidad. La diferencia entre las entradas físicas más pequeñas y las más grandes que se pueden medir por el instrumento de forma confiable se denomina rango dinámico del dispositivo.

Errores de medición

Hasta ahora se han descrito los procesos de medición como si fuesen procesos ideales, sin presencia de errores, alinealidad o desviaciones. Sin embargo, en las mediciones reales existen muchas causas que introducen error.

En las mediciones se realiza un proceso de mapeo de un valor real en un valor equivalente (medido). Se denomina error a la diferencia que existe entre el valor medido y el valor real. En una medida se permite cierta cantidad de error siempre que esté por debajo del nivel de incertidumbre que se acepte en esa situación.

Antes de pasar a exponer las causas que introducen error en una medición, se profundizará en el significado de varios términos básicos utilizados para describir los resultados de una actividad de medición: discriminación, precisión y exactitud.

La discriminación (o también, resolución) se define como el incremento más pequeño que se puede discernir al realizar una medición. La discriminación de una medición es importante porque permite saber si el proceso de detección es capaz de detectar cambios suficientemente pequeños en el mensurando. Incluso si la discriminación es satisfactoria para realizar una medida, el valor obtenido de una medición a otra rara vez es igual. Esto se debe a los errores de medición.

La precisión se define como la dispersión de los valores obtenidos en el conjunto de las mediciones. Dos términos importantes son la repetibilidad y la reproducibilidad. La repetibilidad indica la cercanía entre los valores de un conjunto de medidas realizadas en el corto plazo. La reproducibilidad indica la cercanía entre los valores de un conjunto de medidas realizadas durante un período prolongado en condiciones similares.

La exactitud se define como la cercanía entre el valor medido y el valor real. Cuando se realiza un conjunto de mediciones, se obtiene una variedad de valores medidos. Por este motivo, para determinar la exactitud en este caso, es necesario conseguir un solo valor que represente al conjunto de valores obtenidos en las mediciones. Esto se hace utilizando métodos estadísticos que prevén el cálculo del valor medio del conjunto y la dispersión de valores asociada, llamada varianza.

Una vez descrita la terminología anterior, a continuación, se expondrán las principales fuentes de error en una medición: fuentes de error sistemático (sesgo) y fuentes de error aleatorio (ruido).

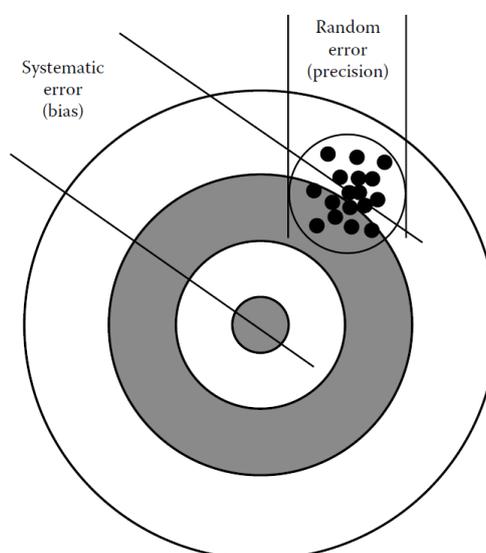


Figura 20: Analogía de la precisión de la medición con los disparos a una diana.
Fuente: [6] Measurement, Instrumentation, and Sensors Handbook.

La combinación de error sistemático y aleatorio se puede visualizar en figura anterior (Figura 20). El error sistemático da como resultado que la agrupación de disparos se desplace del centro de la diana (por ejemplo, debido a una desalineación de la mira o al viento). El tamaño de la agrupación de disparos está determinado por fuentes de error aleatorio y determina la precisión del instrumento.

Los errores sistemáticos (sesgo) son aquellos errores que se pueden eliminar de una medición. Se pueden producir por una gran variedad de factores.

Una clase de factores son aquellos que cambian la respuesta de entrada-salida del sensor, lo que resulta en una mala calibración. Las entradas de modificación o las de interferencia pueden resultar en una mala calibración. Por ejemplo, la temperatura es una entrada de modificación, ya que se puede realizar la medición en unas condiciones de temperatura diferentes a las que había cuando se realizó la calibración, lo que afecta en la medición por una mala calibración.

Hay otros factores que pueden afectar, como el envejecimiento de los componentes o el daño del sensor, que pueden cambiar la calibración. Para evitar estos errores sistemáticos, los sensores deben recalibrarse periódicamente.

También se pueden introducir errores sistemáticos en el procesado de la señal. Si la señal se modifica de alguna manera, la medida indicada será diferente del valor detectado. Por ejemplo, en los circuitos eléctricos, la resistencia o la atenuación también pueden modificar la señal, dando como resultado un error sistemático.

Finalmente, los observadores humanos pueden introducir errores o sesgos sistemáticos al leer la medición. Un ejemplo común de error de sesgo del observador es el de paralaje, que se produce cuando un observador lee un dial desde un ángulo no normal y la aguja indicadora se aprecia desplazada del valor correcto.

Los errores aleatorios (ruido) son aquellos errores permanentes que no se pueden eliminar. Definen la precisión de una medición. El error aleatorio a veces se denomina ruido, que se define como una señal que no lleva información útil. Si una medición con verdadero error aleatorio se repite una gran cantidad de veces exhibirá una distribución gaussiana (Figura 21), en la que se grafica la cantidad de veces que se miden valores dentro de rangos específicos.

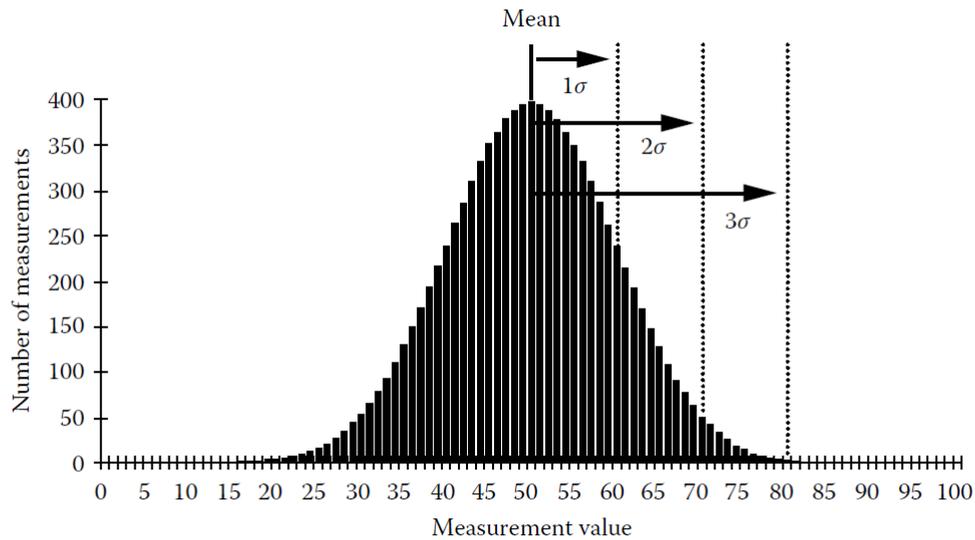


Figura 21: Distribución gaussiana.

Fuente: [6] Measurement, Instrumentation, and Sensors Handbook.

Suponiendo que no haya errores sistemáticos, la distribución gaussiana se centra en el valor real, por lo que la media o el promedio de todas las mediciones darán una buena estimación del valor real.

La precisión de la medida normalmente se cuantifica por la desviación estándar (σ) que indica el ancho de la distribución gaussiana. Una desviación estándar más pequeña conduce a una medición más precisa.

Hay una variedad de fuentes de aleatoriedad que pueden degradar la precisión de la medición, comenzando con la repetibilidad del propio mensurando. Por ejemplo, si se realizan medidas repetidas todas deben ser sobre la misma ubicación exacta (no variando lo que se mide).

El ruido de generación de errores aleatorios también se puede introducir en cada etapa del proceso de medición. Las entradas de interferencia aleatorias darán como resultado ruido del entorno de medición (se introduce antes del sensor). El ruido del propio sensor también se puede introducir dentro del sensor. También se puede introducir error en la ruta de transmisión entre el transductor y el amplificador.

Es importante tener en cuenta que el ruido se amplificará junto con la señal a medida que pasa por el amplificador. Como consecuencia, la cifra que se emplea para analizar el ruido no es el nivel de las fuentes de ruido combinadas, sino el nivel de la relación señal-ruido (SNR), definida como la relación entre la potencia de la señal y la potencia del ruido combinado de las fuentes. Es común informar SNR en unidades de decibelios.

Al determinar el desempeño de un instrumento es esencial apreciar como surgen los errores, ya sean por fuentes de error sistemático o aleatorio. Es importante identificar estas fuentes y realizar un presupuesto de error [6].

3.1.2. Cinemática de un movimiento rectilíneo uniformemente acelerado

Introducción

La cinemática, en física, es la parte de la mecánica que describe el movimiento. Para ello, se emplean tres cantidades físicas: la posición, la velocidad y la aceleración.

En este subapartado se desarrollará la teoría que describe el movimiento rectilíneo en el caso en que la aceleración es constante. También se considerará el caso en que la aceleración varíe durante el movimiento.

Posición y desplazamiento

Para estudiar el movimiento rectilíneo de un objeto se partirá de dos simplificaciones. La primera consiste en establecer que el objeto se comportará como si fuese una partícula. La segunda es que se emplee un sistema de coordenadas.

La posición del objeto viene determinada por el valor de la coordenada y . A medida que el objeto se mueve, la posición cambia en el tiempo. La variación de la posición en el tiempo se denomina desplazamiento. El desplazamiento (Figura 22) es un vector que tiene la dirección del movimiento que lo origina.

$$\Delta y = y_2 - y_1 \quad [3.1.2]$$

Figura 22: Desplazamiento en el eje y .
Fuente: Elaboración propia.

Velocidad media y velocidad instantánea

Inicialmente, se parte de un objeto que se encuentra en movimiento. En la siguiente gráfica (Figura 23) se representa la posición de un objeto que se mueve en un eje en función del tiempo.

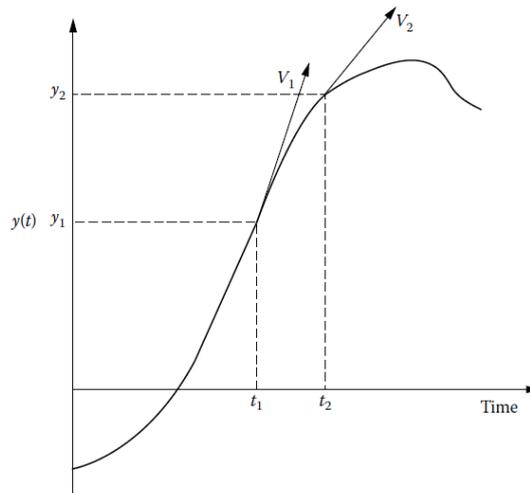


Figura 23: Posición de un objeto que se mueve en un eje en función del tiempo.
Fuente: [6] Measurement, Instrumentation, and Sensors Handbook.

La velocidad media de un objeto (Figura 24) entre dos puntos se define como la tasa de cambio de la posición del objeto, es decir, el desplazamiento realizado, entre el intervalo de tiempo en el que se ha realizado dicho desplazamiento.

$$v_{med} = \frac{y_2 - y_1}{t_2 - t_1} = \frac{\Delta y}{\Delta t} \quad [3.1.3]$$

Figura 24: Velocidad media.
Fuente: Elaboración propia.

Para describir el movimiento con mayor detalle, es necesario definir la velocidad en cualquier instante específico o punto específico de la trayectoria. Esta es la velocidad instantánea.

A medida que el intervalo de tiempo se vuelve más pequeño, la velocidad media se convierte en velocidad instantánea (Figura 25).

$$v_y = \lim_{\Delta t \rightarrow 0} \frac{\Delta y}{\Delta t} = \frac{dy}{dt} \quad [3.1.4]$$

Figura 25: Velocidad instantánea.
Fuente: Elaboración propia.

El término de la velocidad instantánea v_y en un punto se corresponde además con la pendiente de la función posición-tiempo en ese punto.

Aceleración media y aceleración instantánea

Así como la velocidad describe la tasa de cambio de posición con el tiempo, la aceleración describe la tasa de cambio de la velocidad con el tiempo. Tanto la velocidad como la aceleración son vectores.

El objeto en movimiento sobre un eje de la Figura 23 se encuentra en la posición y_1 en el tiempo t_1 con una velocidad instantánea v_{y1} . Tras desplazarse, se encuentra en la posición y_2 en el instante t_2 con una velocidad instantánea v_{y2} .

La aceleración media de un objeto (Figura 26) entre dos puntos se define como la tasa de cambio de la velocidad instantánea en cada uno de los puntos entre el intervalo de tiempo en el que se realiza el desplazamiento desde un punto al otro.

$$a_{med} = \frac{v_{y2} - v_{y1}}{t_2 - t_1} = \frac{\Delta v_y}{\Delta t} \quad [3.1.5]$$

Figura 26: Aceleración media.
Fuente: Elaboración propia.

Al igual que ocurría con la velocidad, ocurre con la aceleración. Es decir, para describir el movimiento con mayor detalle, se requiere conocer la aceleración en cualquier instante de la trayectoria. Esto se logra conociendo la aceleración instantánea.

A medida que el intervalo de tiempo se vuelve más pequeño, la aceleración media se convierte en aceleración instantánea (Figura 27).

$$a_y = \lim_{\Delta t \rightarrow 0} \frac{\Delta v_y}{\Delta t} = \frac{dv_y}{dt} \quad [3.1.6]$$

Figura 27: Aceleración instantánea.
Fuente: Elaboración propia.

El término de la aceleración instantánea a_y en un punto se corresponde además con la pendiente de la función velocidad-tiempo en ese punto.

Movimiento rectilíneo uniformemente acelerado

El movimiento acelerado más sencillo es el rectilíneo con aceleración constante. En este caso, la velocidad cambia al mismo ritmo todo el tiempo. Este tipo de movimiento aparece a menudo en la naturaleza. Desde un cuerpo que

cae con aceleración constante, si los efectos del aire no son importantes, hasta un cuerpo que se desliza sobre una superficie horizontal áspera.

Cuando la aceleración instantánea a_y es constante, la aceleración media a_{med} para cualquier intervalo es igual a la instantánea (Figura 28).

$$a_y = a_{med} = \frac{v_{y2} - v_{y1}}{t_2 - t_1} \quad [3.1.7]$$

Figura 28: Aceleración instantánea en un movimiento rectilíneo uniformemente acelerado.
Fuente: Elaboración propia.

Siendo ahora $t_1=0$ y t_2 cualquier instante posterior t , la velocidad instantánea inicial será v_{y0} y la velocidad instantánea en el instante posterior t , v_y (Figura 29).

$$a_y = \frac{v_y - v_{y0}}{t - 0} \quad [3.1.8]$$

Figura 29: Aceleración instantánea en un movimiento rectilíneo uniformemente acelerado.
Fuente: Elaboración propia.

A partir de la expresión anterior, se obtiene la velocidad instantánea en un movimiento rectilíneo uniformemente acelerado (Figura 30).

$$v_y = v_{y0} + a_y \cdot t \quad [3.1.9]$$

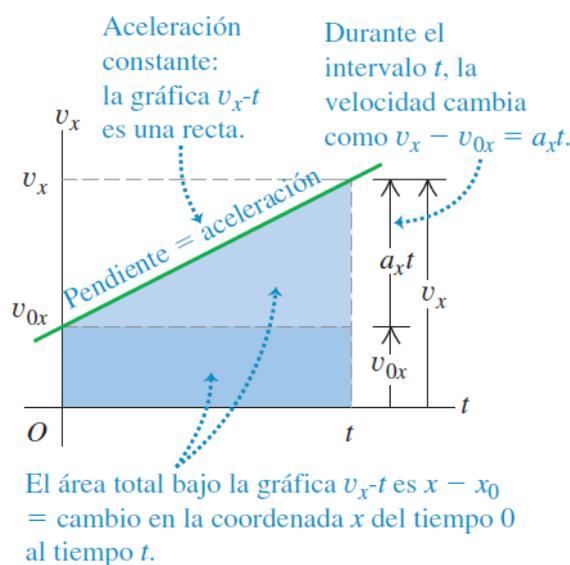


Figura 30: Velocidad instantánea en un movimiento rectilíneo uniformemente acelerado.
Fuente: [16] SEARS ZEMANSKY Física Universitaria.

A continuación, se deducirá una ecuación que determine la posición en función del tiempo cuando la aceleración es constante.

Para ello se parte de la expresión [3.1.3] de la velocidad media, siendo ahora $t_1=0$ y t_2 cualquier instante posterior t . La posición inicial será y_0 y la posición en el instante posterior t , y (Figura 31).

$$v_{med} = \frac{y-y_0}{t-0} = \frac{y-y_0}{t} \quad [3.1.10]$$

Figura 31: Velocidad media independientemente de que la aceleración sea o no constante.
Fuente: Elaboración propia.

Además, si la aceleración es constante, se puede obtener otra expresión de la velocidad media. Al ser la aceleración constante, la velocidad se representa gráficamente como una línea recta (Figura 30). En este caso, la velocidad media en cualquier intervalo es sólo el promedio de las velocidades instantáneas al principio y al final del intervalo (Figura 32).

$$v_{med} = \frac{v_{y0}+v_y}{2} \quad [3.1.11]$$

Figura 32: Velocidad media si la aceleración es constante.
Fuente: Elaboración propia.

Sustituyendo la expresión [3.1.9] en la expresión anterior [3.1.11] se obtiene la siguiente ecuación (Figura 33).

$$v_{med} = \frac{1}{2}(v_{y0} + v_{y0} + a_y \cdot t) \quad [3.1.12]$$

Figura 33: Velocidad media si la aceleración es constante.
Fuente: Elaboración propia.

Igualando la ecuación resultante anterior [3.1.12] con la expresión [3.1.10], se obtiene, tras simplificar, una ecuación que describe la posición en función del tiempo (Figura 34).

$$y = y_0 + v_{y0} \cdot t + \frac{1}{2} \cdot a_y \cdot t^2 \quad [3.1.13]$$

Figura 34: Posición en función del tiempo en un movimiento rectilíneo uniformemente acelerado.
Fuente: Elaboración propia.

Esta ecuación [3.1.13] indica que si, en el instante $t=0$, una partícula está en la posición inicial y_0 y tiene velocidad inicial v_{y0} , su nueva posición y en un t posterior es la suma de tres términos:

- La posición inicial y_0 .
- La distancia $v_{y0} \cdot t$ que recorrería si su velocidad fuese constante.
- Una distancia adicional $\frac{1}{2} \cdot a_y \cdot t^2$ causada por el cambio de velocidad.

La gráfica de esta expresión anterior [3.1.13] que describe la posición de un objeto en función del tiempo representará una parábola.

Velocidad y posición por integración en movimientos rectilíneos con aceleración no constante

Hasta ahora se ha analizado el movimiento rectilíneo con aceleración constante. En esta sección, se tratará el caso en el que la aceleración varía en el tiempo.

Si la aceleración no es constante, no se podrán emplear las expresiones vistas hasta ahora. Pero, aun así, se puede usar la relación $v_y = dx/dt$ para obtener la velocidad v_y en función del tiempo si la posición y es una función conocida de t , y se puede usar $a_y = dv_y/dt$ para obtener la aceleración a_y en función del tiempo si v_y es una función conocida de t . Sin embargo, en muchas ocasiones no se conocen la posición ni la velocidad en función del tiempo, pero sí la aceleración.

La siguiente gráfica (Figura 35) representa la aceleración frente al tiempo de un objeto cuya aceleración no es constante. Se divide el intervalo entre los tiempos t_1 y t_2 en intervalos más pequeños Δt .

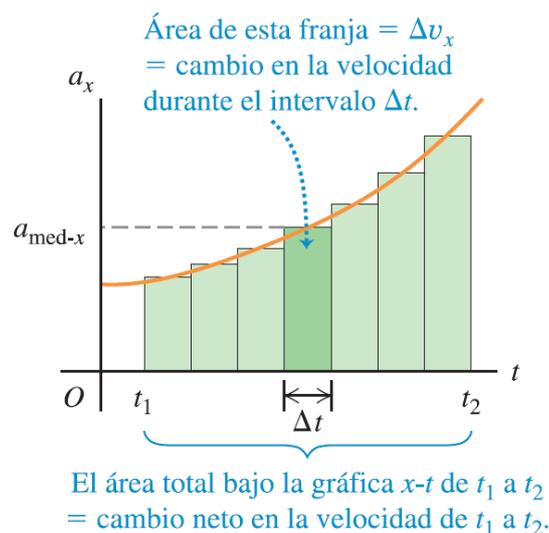


Figura 35: Representación gráfica de la aceleración no constante frente al tiempo.
Fuente: [16] SEARS ZEMANSKY Física Universitaria.

Se considera que la aceleración en Δt es constante, al ser Δt un valor lo suficientemente pequeño. Por este motivo, se emplea la expresión [3.1.5] de la aceleración media aplicada a un intervalo Δt (Figura 36).

$$\Delta v_y = a_{med} \cdot \Delta t \quad [3.1.14]$$

Figura 36: Variación de velocidad instantánea en un intervalo Δt .
Fuente: Elaboración propia.

Gráficamente, Δv_y se corresponde con el área de la tira sombreada (Figura 35) con altura a_{med} y anchura Δt , es decir, el área bajo la curva entre Δt . El cambio de velocidad total será entonces el área total bajo la curva de la aceleración instantánea frente al tiempo. Para calcular dicha área se emplea un proceso de integración (Figura 37).

$$v_{y_2} - v_{y_1} = \int_{v_{y_1}}^{v_{y_2}} dv_y = \int_{t_1}^{t_2} a_y dt \quad [3.1.15]$$

Figura 37: Variación de la velocidad entre t_1 y t_2 a partir de la aceleración instantánea.
Fuente: Elaboración propia.

Se sigue el mismo procedimiento para obtener la variación de la posición o desplazamiento, Δy . Siendo y_1 la posición del objeto en t_1 e y_2 la posición en t_2 , a partir de la expresión [3.1.3] de la velocidad media aplicada a un intervalo Δt , se obtiene la siguiente ecuación (Figura 38).

$$\Delta y = v_{med} \cdot \Delta t \quad [3.1.16]$$

Figura 38: Variación de posición en un intervalo Δt .
Fuente: Elaboración propia.

Gráficamente, el desplazamiento Δy entre t_1 y t_2 es el área bajo la curva de la velocidad instantánea frente al tiempo, entre esos dos instantes. Realizando un proceso de integración se obtiene dicha área (Figura 39).

$$y_2 - y_1 = \int_{y_1}^{y_2} dy = \int_{t_1}^{t_2} v_y dt \quad [3.1.17]$$

Figura 39: Desplazamiento entre t_1 y t_2 a partir de la velocidad instantánea.
Fuente: Elaboración propia.

Las ecuaciones anteriores se pueden utilizar como base para realizar una medición de desplazamiento. Debe tenerse en cuenta que para el movimiento en más de una dimensión habría más de un componente, y por tanto, habría ecuaciones correspondientes para las otras dimensiones [6] [16].

3.2. Materiales

3.2.1. Hardware

FPGA

Las FPGA o matrices de puertas lógicas programables en campo son dispositivos semiconductores basados en una matriz de bloques lógicos configurables (CLBs) conectados mediante interconexiones programables. Las entradas y salidas del circuito se realizan mediante bloques de entrada/salida (IOBs) (Figura 40).

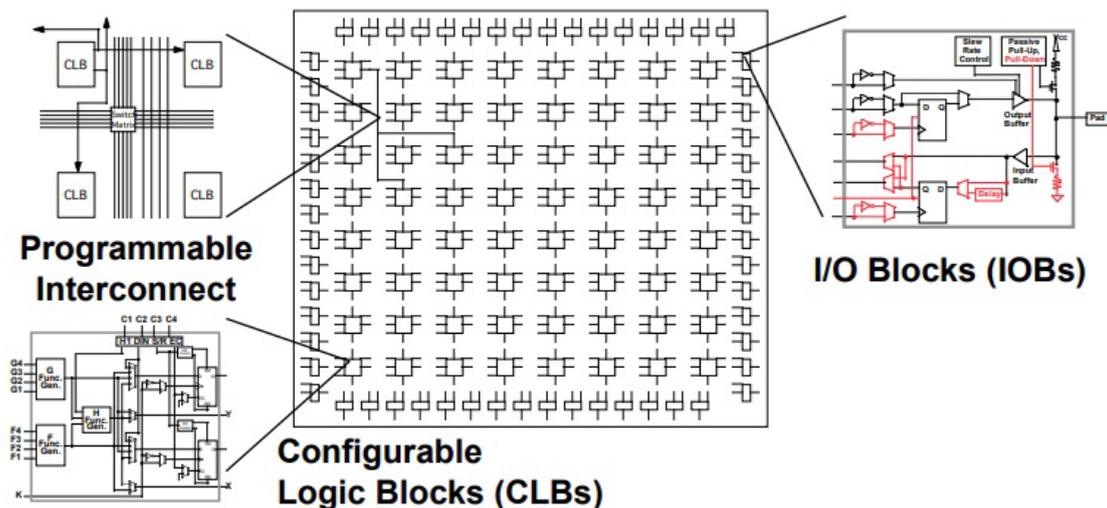


Figura 40: Esquema de la arquitectura de una FPGA donde se señalan los bloques lógicos configurables, los bloques de entrada/salida y las interconexiones programables.

Fuente: <https://www.researchgate.net/profile/Victor-Martinez-Hernando/publication/39562537/figure/fig18/AS:552661919047689@1508776277911/Figura-410-Esquema-basico-de-la-arquitectura-de-una-FPGA-Xilinx-XC4000-Xilinx-web.png> .

La forma en que se determinan las funciones de los bloques lógicos y el enrutamiento de la interconexión es mediante celdas de configuración, que pueden visualizarse como interruptores 0/1 (apagado/encendido). Dependiendo de la FPGA, las celdas de configuración pueden implementarse usando una de estas tres tecnologías:

- Antifusible: Estas celdas de configuración son programables una sola vez (OTP), lo que significa que una vez que el dispositivo ha sido programado no hay vuelta atrás. Estos dispositivos tienden a estar limitados a aplicaciones espaciales y de alta seguridad. Como se venden en pequeñas cantidades, su precio es alto y son una opción costosa de diseño.
- Flash: Al igual que las celdas de configuración basadas en antifusibles, las celdas basadas en flash son no volátiles. A diferencia de las celdas de antifusible, las celdas de flash pueden reprogramarse según sea

necesario. Las celdas de configuración flash son tolerantes a la radiación, lo que hace que estos dispositivos sean adecuados para aplicaciones espaciales (aunque con modificaciones en sus capas y paquetes de metalización superiores).

- **SRAM:** En este caso, los datos de configuración se almacenan en una memoria externa desde donde se cargan cada vez que se activa el FPGA (o según se recomiende en el caso de escenarios de configuración dinámica). Las FPGA con celdas de configuración basadas en SRAM ofrecen un mayor rendimiento que las basadas en antifusible o flash, pero consumen más energía.

Aunque existen FPGA que se pueden programar una sola vez, la mayoría se pueden reprogramar a medida que evoluciona el diseño, adaptándose a la aplicación deseada o los requisitos de funcionalidad después de la fabricación. Esta característica distingue a las FPGA de los ASIC, que se fabrican a medida para tareas de diseño específicas.

Las primeras FPGA se desarrollan en 1984 por la compañía tecnológica estadounidense Xilinx, fundada por Ross Freeman, Bernie Von der Schmitt y Jim Barnett. Desde su aparición, las FPGA han crecido en capacidad y rendimiento y han disminuido en costo y energía por operación.

En la actualidad cuentan con procesadores integrados, bloques DSP y relojes entre otras características. Los precios son cada vez más bajos, siendo una propuesta atractiva para casi cualquier tipo de diseño.

Los campos de aplicación de las FPGA son múltiples: aeroespacial, automotriz, centros de datos, comunicaciones, creación de prototipos, defensa, electrónica de consumo, industria, medicina [8] [12] [15].

Acelerómetro ADXL362

La placa Nexys4 DDR incluye un acelerómetro ADXL362 de Analog Device. El acelerómetro ADXL362 es un acelerómetro MEMS de 3 ejes de potencia ultrabaja. La FPGA se comunica con el ADXL362 a través de la interfaz SPI (Figura 41).

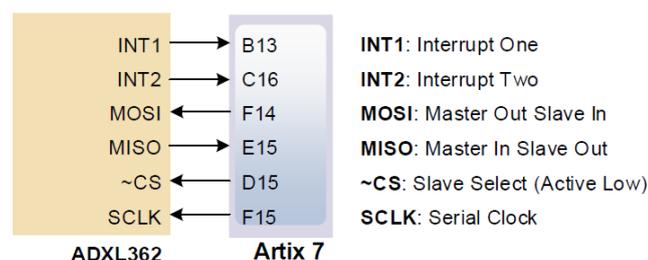


Figura 41: Interfaz del acelerómetro ADXL362.
Fuente: [4] Nexys4 DDR FPGA Board Reference Manual.

Mientras el acelerómetro está en el modo de medición, mide y almacena continuamente datos de aceleración correspondientes a los ejes X, Y y Z. El acelerómetro también incorpora un sensor de temperatura, por lo que a la salida se obtienen cuatro señales: las tres de aceleración y la de temperatura.

Proporciona una resolución de salida de 12 bits, aunque si fuese suficiente una resolución más baja permite proporcionar datos formateados para un solo byte. Además de este cambio, el acelerómetro permite configurar otros parámetros, lo que hace que sea atractivo para múltiples aplicaciones que necesitan flexibilidad [1] [4].

Visualizador 7 segmentos

La placa Nexys4 DDR contiene dos módulos de visualizadores 7 segmentos de cuatro dígitos cada uno. Los dos módulos están compuestos para comportarse como uno solo de ocho dígitos.

Los LEDs de cada uno de los segmentos se puede iluminar individualmente, por lo que cualquiera de los 128 patrones se puede mostrar en un dígito iluminando ciertos segmentos LED y dejando los demás oscuros (Figura 42). De estos 128 patrones posibles, los más útiles son los que se corresponden con los números decimales.

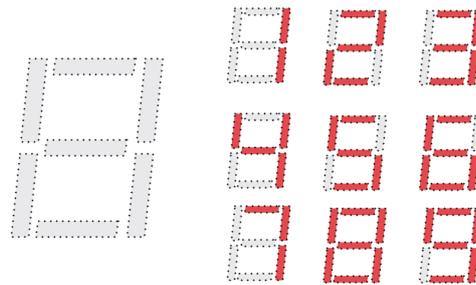


Figura 42: Patrones correspondientes a los números decimales para un dígito de un visualizador 7 segmentos.

Fuente: [4] Nexys4 DDR FPGA Board Reference Manual.

Los ánodos de los siete LEDs que forman cada dígito están unidos en un nodo de circuito habiendo un “ánodo común”. De esta manera, existen ocho señales de ánodo común, una para cada dígito, que permiten o no la “habilitación de dígitos” en el visualizador. Estas se nombran desde AN0 hasta AN7.

Por otro lado, los cátodos de los LEDs pertenecientes a un mismo dígito permanecen separados entre sí, pero se unen con sus equivalentes de los demás dígitos. De esta forma, los cátodos de los segmentos iguales en los ocho dígitos están conectados dando lugar a siete nodos, uno por segmento, que se nombran desde CA hasta CG (Figura 43).

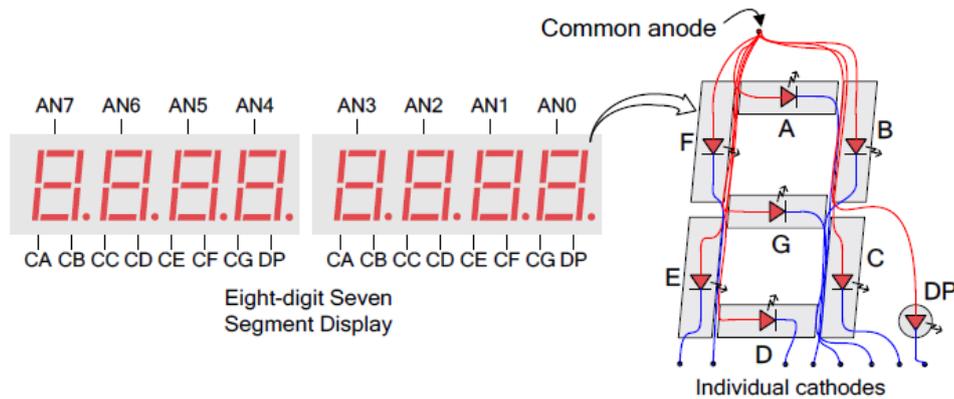


Figura 43: Ánodos y cátodos de los siete LEDs que forman cada dígito y conjunto de los 8 dígitos.
Fuente: [4] Nexys4 DDR FPGA Board Reference Manual.

Este esquema de conexión de señales crea una pantalla multiplexada, donde las señales de cátodo son comunes a todos los dígitos, pero solo pueden iluminar los segmentos del dígito cuya señal de ánodo esté activada. Para iluminar un segmento, el ánodo debe estar alto mientras que el cátodo debe estar bajo.

Se puede usar un circuito controlador del visualizador para mostrar un número de ocho dígitos. Este circuito configura las señales de ánodo y cátodo correspondientes a cada dígito en una sucesión continua repetitiva dígito a dígito. Al ser la tasa de actualización más rápida que el ojo humano, el ojo no puede percibir el oscurecimiento de un dígito antes de que se ilumine nuevamente, por lo que causa la impresión de que el dígito está continuamente iluminado. Si la tasa de actualización o de refresco se reduce alrededor de 45 Hz, se puede notar un parpadeo en el monitor. Para que cada uno de los dígitos parezca iluminado continuamente, los ocho dígitos deben activarse una vez cada 1 a 16 ms, es decir, con una frecuencia de refresco entre 1 kHz y 60 Hz aproximadamente. Para cada uno de los ocho dígitos, la tasa de refresco tendrá que ser la octava parte del total [4].

Módulo de comunicación de puerto serie inalámbrico HC-12

El módulo de comunicación de puerto serie inalámbrico HC-12 (Figura 44) es un módulo de transmisión de datos inalámbrico integrado multicanal. Tiene las siguientes características:

- Transmisión inalámbrica de larga distancia:
 - FU3: 1000 m en espacio abierto con una tasa de baudios de 5000 bps.
 - FU4: 1800 m en espacio abierto con una tasa de baudios de 500 bps.
- Banda de frecuencia de trabajo es 433.4-473.0 MHz, con 100 canales de comunicación.

- Potencia de transmisión máxima de 100 mW (20 dBm) (se pueden configurar 8 niveles de potencia).
- El número de bytes transmitidos continuamente es ilimitado.
- Tiene un MCU integrado para realizar la comunicación con el dispositivo externo a través de puerto serie, por lo que no requiere programación o configuración para uso básico.

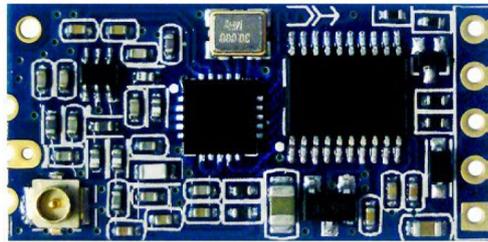


Figura 44: Módulo de comunicación de puerto serie inalámbrico HC-12.
Fuente: [11] HC-12 Wireless Serial Port Communication Module.

Como se muestra en el siguiente diagrama (Figura 45), se pueden emplear dos módulos HC-12 en lugar del cableado físico de un enlace de comunicaciones en serie semidúplex.

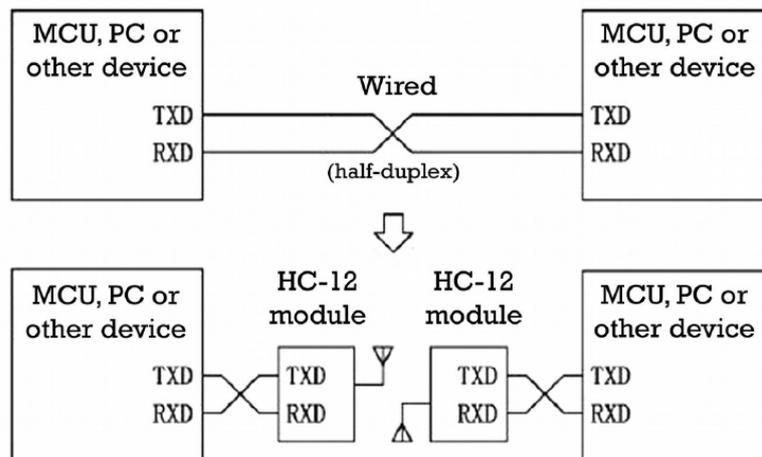


Figura 45: Principio de funcionamiento del módulo HC-12.
Fuente: [11] HC-12 Wireless Serial Port Communication Module.

El módulo HC-12 se puede emplear en diversas aplicaciones, siendo algunas de ellas:

- Redes inalámbricas de PC.
- Adquisición inalámbrica de datos.
- Control remoto industrial y telemetría [11].

3.2.2. Software

Xilinx ISE Design Suite

Xilinx ISE Design Suite es una herramienta de software que permite la síntesis y el análisis de diseños hardware empleando HDL, permitiendo el diseño FPGA y CPLD.

ISE permite al diseñador realizar síntesis HDL, simulación, implementación, ajuste de dispositivos y programación JTAG, ofreciendo así un flujo de diseño completo.

El entorno de diseño proporciona las herramientas fundamentales, las tecnologías y un flujo de trabajo familiar para lograr resultados de diseño óptimos. Esto permite una mayor productividad y flexibilidad del sistema, aumentando el rendimiento y reduciendo costes.

El software es descargable para escritorio de manera gratuita y está disponible para Microsoft Windows y Linux.

ISE fue desarrollado por la compañía tecnológica estadounidense Xilinx quien lanzó la última actualización del software (Versión 14.7) en octubre de 2013. La compañía recomienda su nuevo software actualizado para los nuevos dispositivos Vivado Design Suite [13] [14].

Matlab

Matlab es un software de programación y cálculo numérico que se emplea para analizar datos, desarrollar algoritmos y crear modelos.

Matlab combina un entorno de escritorio con un lenguaje de programación propio que permite describir expresiones matemáticas matriciales y vectoriales de forma directa. Cuenta con herramientas que ofrecen:

- Análisis de datos: exploración, modelado y visualización de datos.
- Cálculo en la nube: ejecución en entornos en la nube desde MathWorks Cloud hasta nubes públicas como AWS y Azure.
- Cálculo paralelo: cálculos a gran escala mediante equipos multinúcleo, GPU, clusters, mallas y nubes.
- Creación de apps: tanto web como de escritorio.
- Desarrollo de algoritmos: diseño de algoritmos para aplicaciones de escritorio y embebidas.
- Hardware: conexión de Matlab con hardware.
- Uso de Matlab con otros lenguajes: utilización de Matlab con Python, C/C+, Fortran, Java y otros lenguajes.

Algunos de los múltiples campos de aplicación de Matlab son:

- Comunicaciones inalámbricas.
- Deep Learning: preparación de datos, diseño, simulación y despliegue para redes neuronales profundas.
- Machine learning: entrenamiento de modelos, ajuste de parámetros y despliegue en producción o en el perímetro.
- Mantenimiento predictivo: desarrollo e implementación software de supervisión de condiciones y mantenimiento predictivo.
- Procesamiento de imágenes y visión artificial: adquisición, procesamiento y análisis de imágenes y vídeos para el desarrollo de algoritmos y el diseño de sistemas.
- Procesamiento de señales: análisis de señales y datos en serie. Modelado, diseño y simulación de sistemas de procesamiento de señales.
- Prueba y medición: adquisición, análisis y exploración de datos y automatización de pruebas.
- Robótica: conversión de ideas y conceptos de robótica en sistemas autónomos que funcionan a la perfección en entornos reales.
- Sistemas de control.

El programa es software de pago, aunque también ofrece licencias gratuitas para organizaciones educativas. Está disponible para Microsoft Windows, Linux y macOS.

Matlab fue desarrollado por la compañía estadounidense MathWorks. La versión más reciente (Versión R2020a) fue publicada en marzo de 2020 [7].

3.3. Técnicas

3.3.1. Lenguajes

VHDL

VHDL es un lenguaje empleado en la descripción de sistemas electrónicos digitales. VHDL surgió del programa de Circuitos Integrados de Muy Alta Velocidad del Departamento de Defensa de los Estados Unidos. En el transcurso del programa se pone de manifiesto la necesidad de un lenguaje estándar para describir la estructura y las funciones de los circuitos integrados. Con este fin se creó el Lenguaje de Descripción de Hardware de circuitos integrados de muy alta Velocidad (VHDL, del inglés Very high speed integrated circuits Hardware Description Language).

Posteriormente fue desarrollado por el Instituto de Ingenieros Eléctricos y Electrónicos (IEEE). En 1987 se adopta el "IEEE Standard 1076, Standard VHDL Language Reference Manual". Esta primera versión normalizada del lenguaje se denomina VHDL-87.

Cada cierto tiempo, como todos los estándares IEEE, el estándar VHDL se revisa y actualiza teniendo en cuenta los comentarios y sugerencias de los usuarios. De este modo, en 1992 se propone una nueva versión que finalmente se adopta en 1993, dando como resultado el VHDL-93. Una segunda revisión se inició en 1998 dando lugar al VHDL-2002. Después de esta modificación, el IEEE y Acellera Systems Initiative, una organización cuya finalidad era promover estándares para el diseño de electrónica, llevan a cabo un mayor desarrollo del lenguaje que tuvo como consecuencia el VHDL-2008. En la actualidad la última revisión del estándar es la del año 2019.

VHDL está diseñado para satisfacer una serie de necesidades en el proceso de diseño:

- 1) Permite la descripción de la estructura de un sistema, es decir, como se descompone en subsistemas y como estos subsistemas están interconectados.
- 2) Permite especificar la función de un sistema empleando formas similares a un lenguaje de programación.
- 3) Posibilita que el diseño de un sistema pueda ser simulado antes de ser fabricado, de modo que los diseñadores puedan comparar rápidamente las alternativas y probar correcciones sin la demora y el gasto de prototipos de hardware.
- 4) Permite que la estructura detallada de un diseño sea sintetizada a partir de especificaciones más abstractas, permitiendo a los diseñadores enfocarse en decisiones de diseño más estratégicas reduciendo los tiempos de comercialización.

En la actualidad, VHDL es uno de los lenguajes de descripción de hardware más empleados junto a Verilog [2].

3.3.2. Protocolos

Protocolo SPI

SPI (Figura 46) es un protocolo de comunicaciones síncrono con una configuración full-duplex que utiliza cuatro señales para las comunicaciones entre un nodo maestro y uno o varios esclavos. Estas cuatro señales son:

- Chip Select (CS).
- Clock (SCK).
- Master Out/Slave In (MOSI).
- Master In/Slave Out (MISO).

Los datos del nodo maestro (o principal) y los del nodo esclavo (o secundario) se sincronizan en el flanco ascendente o descendente del reloj, cuya señal es generada por el nodo principal. Tanto el maestro como el esclavo pueden transmitir datos al mismo tiempo. Los dispositivos SPI admiten frecuencia de reloj muy elevadas en comparación con otras interfaces, como I2C.

La señal de selección de chip del maestro se usa para seleccionar al esclavo. Cuando se emplean múltiples nodos secundarios, se requiere una señal de selección de chip individual para cada esclavo. Hay dos posibles configuraciones en el caso de utilizar múltiples esclavos, el modo SPI regular y el modo de conexión en cadena.

MOSI y MISO son las líneas de datos. MOSI transmite datos del maestro a los esclavos y MISO transmite datos desde los esclavos al maestro.

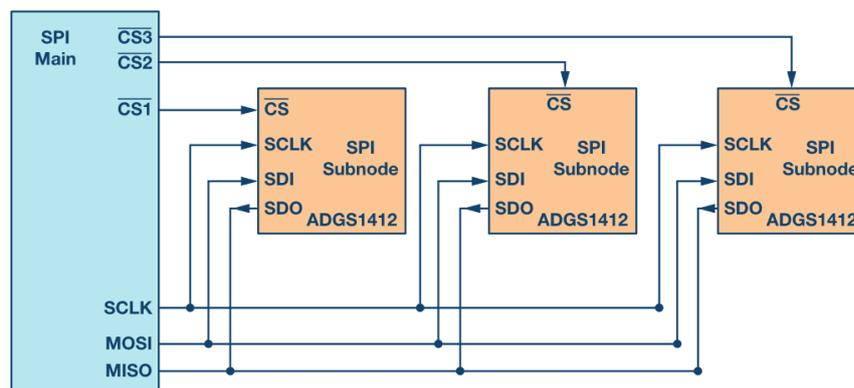


Figura 46: Configuración del protocolo SPI con varios esclavos empleando en modo regular.
Fuente: [3] Introduction to SPI Interface.

Es una de las interfaces más empleadas entre microcontroladores y circuitos integrados periféricos, como sensores, ADC, DAC y SRAM, entre otros [3] [4].

Protocolo UART

UART (Figura 47) es un protocolo de comunicación de hardware que utiliza comunicación serial asíncrona con velocidad configurable. Es uno de los protocolos de comunicación dispositivo a dispositivo más utilizados. Puede funcionar con muchos tipos diferentes de protocolos en serie que implican la transmisión y recepción de datos en serie.

En la comunicación en serie, los datos se transfieren bit a bit utilizando una sola línea o cable. Al ser una comunicación bidireccional, UART emplea dos cables, transmisión y recepción, para una transferencia de datos en serie exitosa. Las comunicaciones en serie necesitan menos circuitos y cables, lo que reduce el costo de implementación.

Las dos señales de cada dispositivo UART se conocen como:

- Transmisor (TX).
- Receptor (RX).

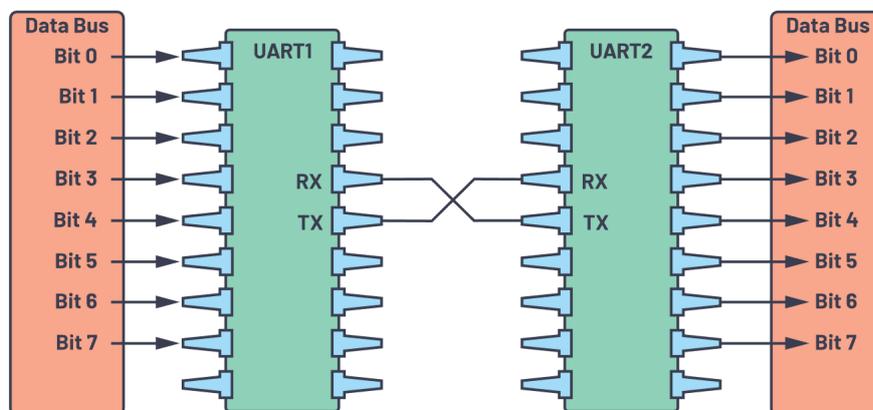


Figura 47: Protocolo UART con bus de datos.

Fuente: [9] UART: A Hardware Communication Protocol Understanding Universal Asynchronous Receiver/Transmitter

En el protocolo UART, primero, un bus de datos de control envía datos en forma paralela. Estos datos son recibidos por el UART transmisor que los transmite en serie a través de su línea de transmisión (TX), bit a bit, al UART receptor. Este segundo UART, a su vez, convierte los datos en serie en paralelo para enviárselos a un segundo bus de datos.

Para UART y la mayoría de las comunicaciones en serie, la velocidad en baudios debe configurarse de la misma manera tanto en el dispositivo de transmisión como en el de recepción. La velocidad en baudios es la velocidad a la que se transfiere la información a un canal de comunicación. En el contexto del puerto serie, la tasa de baudios establecida se traduce como el número máximo de bits por segundo que se transferirán.

La interfaz UART no utiliza una señal de reloj para sincronizar los dispositivos transmisor y receptor; transmite datos de forma asíncrona. En lugar de una señal de reloj, el transmisor genera un flujo de bits basado en su señal de reloj mientras que el receptor usa su señal de reloj interna para muestrear los datos entrantes. El punto de sincronización se gestiona teniendo la misma tasa de baudios en ambos dispositivos. El no hacerlo puede afectar el tiempo de envío y recepción de datos, lo que puede causar discrepancias durante el manejo de datos. La diferencia permisible de velocidad en baudios es de hasta un 10% antes de que la temporización de los bits se aleje demasiado.

El modo que emplea el protocolo UART para transmitir información es a través de paquetes. Un paquete consta de un bit de inicio, una trama de datos, un bit de paridad y bits de parada.

El protocolo UART se emplea en muchas aplicaciones, como en depuración de errores durante el desarrollo de un sistema o la verificación y testeo de productos antes de pasar al proceso de fabricación [4] [9].

4. ESTUDIO PRELIMINAR PARA COMPROBAR LA HIPÓTESIS

4.1. Primera versión hardware de la primera etapa “Adquisición de datos”

4.1.1. Introducción

En primer lugar, se debe comprobar el cumplimiento de la hipótesis, es decir, que los datos obtenidos por el sensor (acelerómetro ADXL362) en el proceso de “Adquisición de datos” tengan la calidad suficiente para que en el proceso posterior de “Procesado de datos” los resultados sean óptimos (Figura 48).

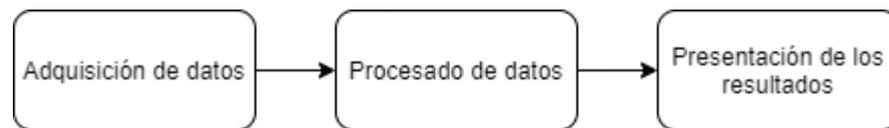


Figura 48: Procesos principales que realiza el diseño hardware.
Fuente: Elaboración propia.

A tal efecto, se diseña una versión preliminar de la primera etapa “Adquisición de datos” para obtener datos del sensor y posteriormente, poder analizarlos. En los siguientes subapartados se desarrolla con más detalle la descripción de este diseño y el análisis de los datos obtenidos.

4.1.2. Funcionamiento del diseño

Esta primera versión hardware de la primera etapa “Adquisición de datos” está formada por tres bloques principales (Figura 49).



Figura 49: Bloques principales que forman el diseño de la primera versión hardware de la primera etapa “Adquisición de datos”.

Fuente: Elaboración propia.

El primer bloque “Sensor” contiene la entidad que controla el acelerómetro ADXL362. Este bloque permite obtener a su salida las señales de la aceleración en el eje X, eje Y y eje Z, así como una señal de temperatura que también es obtenida por el acelerómetro ADXL362.

El segundo bloque “Procesado” permite seleccionar una de las cuatro señales anteriores y adaptarla a código ASCII para su posterior transmisión.

El tercer bloque “Transmisión” está formado por una UART que se encarga de enviar los datos en código ASCII a un ordenador. Los datos se muestran en pantalla empleando el programa “terminal.exe”. La tasa de baudios empleada en esta comunicación está configurada a 9600 bits por segundo.

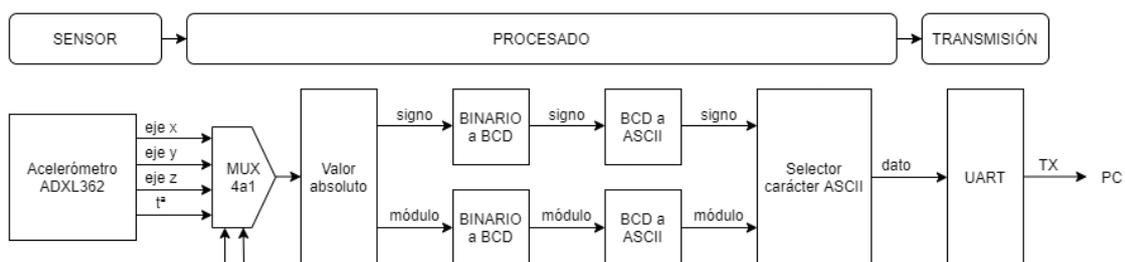


Figura 50: Esquema del funcionamiento del diseño de la primera versión hardware de la primera etapa “Adquisición de datos”.

Fuente: Elaboración propia.

4.1.3. Descripción de los componentes del diseño

En este subapartado se realiza una descripción más detallada de los componentes empleados en el diseño de la primera versión hardware de la primera etapa. Para facilitar la comprensión del lector se ha decidido organizar la información en tablas, de tal forma que cada componente tiene su propia tabla.

Los diseños en VHDL de los componentes de esta primera versión de la primera etapa se pueden encontrar en la siguiente ruta, en el CD adjunto: Otros/Anexos/1 – Etapa 1 (Adquisición de datos) – Primera versión hardware/Diseño hardware. El archivo que contiene el diseño de la primera versión hardware de la primera etapa “Adquisición de datos” se llama “v1.1.xise”.

Módulo principal

El módulo principal del archivo “v1.1.xise” se llama “general.vhd” (Tabla 2). Contiene los tres bloques funcionales:

- Bloque “Sensor” con su módulo “sensor.vhd”.
- Bloque “Procesado” con su módulo “control.vhd”.
- Bloque “Transmisión” con su módulo “uart.vhd”.

general.vhd

El módulo “general.vhd” gestiona la adquisición de datos del sensor, su procesado y su transmisión al ordenador a través de cable USB.

SEÑAL	BITS	I/O	DESCRIPCIÓN
clk	1	input	Señal de reloj.
reset	1	input	Señal de inicialización asíncrono.
start	1	input	Señal de inicio del procesado y de la transmisión.
sel	2	input	Señal selectora para escoger entre la aceleración del eje x, y, z o la temperatura.
miso	1	input	Señal MISO del protocolo SPI.
sclk	1	output	Señal de reloj del protocolo SPI.
mosi	1	output	Señal MOSI del protocolo SPI.
ss	1	output	Señal de habilitación maestro/esclavo del protocolo SPI.
tx	1	output	Señal transmitida al ordenador mediante cable USB a través del protocolo UART.

Tabla 2: Descripción del módulo “general.vhd”.
Fuente: Elaboración propia.

En el Anexo 1 se puede consultar el esquema de este módulo principal.

Bloque “Sensor”

El módulo del bloque “Sensor” se llama “sensor.vhd” (Tabla 3). En las siguientes tablas se describen los componentes del bloque “Sensor”.

sensor.vhd

El módulo “sensor.vhd” gestiona la adquisición de datos del sensor de manera que tiene 4 señales de salida (aceleración en el eje X, Y, Z y la temperatura). Estas 4 señales cambian de forma síncrona y cuando lo hacen se activa durante un pulso la señal “done”.

SEÑAL	BITS	I/O	DESCRIPCIÓN
clk	1	input	Señal de reloj.
reset	1	input	Señal de inicialización asíncrono.
miso	1	input	Señal MISO del protocolo SPI.
sclk	1	output	Señal de reloj del protocolo SPI.
mosi	1	output	Señal MOSI del protocolo SPI.
ss	1	output	Señal de habilitación maestro/esclavo del protocolo SPI.
done	1	output	Señal de finalización.
x	12	output	Señal de aceleración en el eje X en Ca2.
y	12	output	Señal de aceleración en el eje Y en Ca2.
z	12	output	Señal de aceleración en el eje Z en Ca2.
tmp	12	output	Señal de temperatura en Ca2.

Tabla 3: Descripción del módulo “sensor.vhd”.
Fuente: Elaboración propia.

El módulo “sensor.vhd” tiene instanciado un módulo llamado “ADXL362Ctrl.vhd” (Tabla 4) que controla el acelerómetro ADXL362. A su vez, éste último requiere de otro módulo llamado “SPI_If.vhd” (Tabla 5) para realizar la comunicación con el acelerómetro mediante el protocolo SPI.

ADXL362Ctrl.vhd

El módulo “ADXL362Ctrl.vhd” representa el controlador del acelerómetro ADXL362 de la Nexys4. El módulo emplea el protocolo SPI para comunicarse con el acelerómetro.

En primer lugar, el módulo configura los registros del acelerómetro ADXL362. A continuación, lee de forma continua las aceleraciones de los ejes X, Y, Z y la temperatura, y les aplica un algoritmo. Por último, el resultado de cada variable se envía empleando señales de salida de 12 en Ca2.

SEÑAL	BITS	I/O	DESCRIPCIÓN
SYSCLK	1	input	Señal de reloj.
RESET	1	input	Señal de inicialización asíncrono.
MISO	1	input	Señal MISO del protocolo SPI.
SCLK	1	output	Señal de reloj del protocolo SPI.
MOSI	1	output	Señal MOSI del protocolo SPI.
SS	1	output	Señal de habilitación maestro/esclavo del protocolo SPI.
Data_Ready	1	output	Señal de dato recibido.
ACCEL_X	12	output	Señal de aceleración en el eje X en Ca2.
ACCEL_Y	12	output	Señal de aceleración en el eje Y en Ca2.
ACCEL_Z	12	output	Señal de aceleración en el eje Z en Ca2.
ACCEL_TMP	12	output	Señal de temperatura en Ca2.

Tabla 4: Descripción del módulo “ADXL362Ctrl.vhd”.
Fuente: Elaboración propia.

SPI_If.vhd

El módulo “SPI_If.vhd” representa el controlador del protocolo SPI. El controlador transfiere 8 bits de datos (MSB primero).

SEÑAL	BITS	I/O	DESCRIPCIÓN
SYCLK	1	input	Señal de reloj.
RESET	1	input	Señal de inicialización asíncrono.
MISO	1	input	Señal MISO del protocolo SPI.
SCLK	1	output	Señal de reloj del protocolo SPI.
MOSI	1	output	Señal MOSI del protocolo SPI.
SS	1	output	Señal de habilitación maestro/esclavo del protocolo SPI.
Din	8	input	Datos a transmitir.
Dout	8	output	Datos recibidos.
Start	1	input	Señal de inicio de la transmisión.
Done	1	output	Señal de finalización de la transmisión.
HOLD_SS	1	input	Señal que fuerza en bajo a SS en caso de modo transmisión/recepción de múltiples bytes.

Tabla 5: Descripción del módulo “SPI_If.vhd”.
Fuente: Elaboración propia.

En el Anexo 2 se puede consultar el esquema de este bloque “Sensor”.

Bloque “Procesado”

El módulo del bloque “Procesado” se llama “control.vhd” (Tabla 6). En las siguientes tablas se describen los componentes del bloque “Procesado”.

control.vhd			
<p>El módulo “control.vhd” controla el procesamiento de datos de la primera versión hardware de la primera etapa “Adquisición de datos”. Permite seleccionar una de las señales adquiridas por el sensor (aceleración del eje X, Y, Z o la temperatura) y la adapta a código ASCII para su posterior transmisión.</p>			
SEÑAL	BITS	I/O	DESCRIPCIÓN
clk	1	input	Señal de reloj.
reset	1	input	Señal de inicialización asíncrono.
start	1	input	Señal de inicio del procesamiento.
tx_busy	1	input	Señal de transmisión ocupada.
sel	2	input	Señal selectora para escoger entre la aceleración del eje X, Y, Z o la temperatura.
x	12	input	Señal de aceleración en el eje X en Ca2.
y	12	input	Señal de aceleración en el eje Y en Ca2.
z	12	input	Señal de aceleración en el eje Z en Ca2.
t	12	input	Señal de temperatura en Ca2.
tx_start	1	output	Señal de inicio de la transmisión.
tx_data	8	output	Datos a transmitir.

Tabla 6: Descripción del módulo “control.vhd”.
Fuente: Elaboración propia.

mux4a1.vhd

El módulo “mux4a1.vhd” contiene el diseño de un multiplexor 4 a 1. Los datos de entrada “x”, “y”, “z”, “t” (12 bits en Ca2) entran a este multiplexor 4 a 1 y mediante su entrada “sel” (2 bits) se elige el dato a procesar que pasa a la señal de salida “sal” (12 bits en Ca2).

SEÑAL	BITS	I/O	DESCRIPCIÓN
sel	2	input	Señal selectora para escoger entre la aceleración del eje X, Y, Z o la temperatura.
x	12	input	Señal de aceleración en el eje X en Ca2.
y	12	input	Señal de aceleración en el eje Y en Ca2.
z	12	input	Señal de aceleración en el eje Z en Ca2.
t	12	input	Señal de temperatura en Ca2.
sal	12	output	Señal de salida del MUX en Ca2.

Tabla 7: Descripción del módulo “mux4a1.vhd”.

Fuente: Elaboración propia.

absoluto.vhd

El módulo “absoluto.vhd” realiza el estudio del signo de un número y calcula su valor absoluto. Al dato contenido en la señal “data” se le realiza un estudio del signo para ver si es positivo o negativo. En caso de ser negativo, se calcula su valor absoluto. A la salida de este módulo habrá las siguientes señales: “signo” (1 bit), el valor absoluto en la señal “bin” (13 bits) y “done” (1 bit).

SEÑAL	BITS	I/O	DESCRIPCIÓN
clk	1	input	Señal de reloj
reset	1	input	Señal de inicialización asíncrono.

start	1	input	Señal de inicio.
data	12	input	Señal que contiene el dato a procesar en Ca2.
done	1	output	Señal de finalización.
signo	1	output	Señal con información del signo del dato ("0" si es positivo, "1" si es negativo).
bin	13	output	Señal que contiene el valor absoluto del dato.

Tabla 8: Descripción del módulo "absoluto.vhd".
Fuente: Elaboración propia.

signo2bcd.vhd

El módulo "signo2bcd.vhd" asigna un valor BCD (4 bits) en función del valor de la entrada "signo", para que posteriormente, en el módulo "bcd2ascii.vhd", la salida "bcd" pueda convertirse según corresponda a código ASCII para un signo menos "-" o un espacio " ".

SEÑAL	BITS	I/O	DESCRIPCIÓN
clk	1	input	Señal de reloj.
reset	1	input	Señal de inicialización asíncrono.
signo	1	input	Señal con información del signo del dato ("0" si es positivo, "1" si es negativo).
bcd	4	output	Señal con valor BCD conocido en función de la entrada para posterior conversión a ASCII.

Tabla 9: Descripción del módulo "signo2bcd.vhd".
Fuente: Elaboración propia.

bin2bcd.vhd

El módulo “bin2bcd.vhd” convierte un número binario absoluto (13 bits) a un número BCD de 4 cifras. A la salida de este módulo habrá las siguientes señales: “ud” (4 bits), “dec” (4 bits), “cen” (4 bits) y “mil” (4 bits).

SEÑAL	BITS	I/O	DESCRIPCIÓN
clk	1	input	Señal de reloj.
reset	1	input	Señal de inicialización asíncrono.
start	1	input	Señal de inicio.
done	1	output	Señal de finalización.
num_bin	13	input	Señal con el número binario absoluto.
ud	4	output	Señal con las unidades del número BCD.
dec	4	output	Señal con las decenas del número BCD.
cen	4	output	Señal con las centenas del número BCD.
mil	4	output	Señal con los millares del número BCD.

Tabla 10: Descripción del módulo “bin2bcd.vhd”.
Fuente: Elaboración propia.

bcd2ascii.vhd

El módulo “bcd2ascii.vhd” convierte los valores BCD (4 bits) que contienen información del signo, unidades, centenas y millares a código ASCII (8 bits). Por este motivo se instanciarán 5 componentes “bcd2ascii.vhd”.

SEÑAL	BITS	I/O	DESCRIPCIÓN
clk	1	input	Señal de reloj.
reset	1	input	Señal de inicialización asíncrono.
bcd	4	input	Señal con el valor BCD a convertir.
ascii	8	output	Señal con el carácter ASCII asociado correspondiente.

Tabla 11: Descripción del módulo “bcd2ascii.vhd”.
Fuente: Elaboración propia.

ascii_selector.vhd

El módulo “ascii_selector.vhd” realiza el proceso de selección de los datos ASCII (“signo”, “mil”, “cen”, “dec”, “ud” e intro), de manera que pone en la salida “tx” (8 bits) un dato distinto para que posteriormente la UART realice la transmisión.

SEÑAL	BITS	I/O	DESCRIPCIÓN
clk	1	input	Señal de reloj.
reset	1	input	Señal de inicialización asíncrono.
start	1	input	Señal de inicio.
tx_busy	1	input	Señal de transmisión ocupada.
signo	8	input	Señal con el código ASCII del signo.
ud	8	input	Señal con el código ASCII de las unidades.
dec	8	input	Señal con el código ASCII de las decenas.
cen	8	input	Señal con el código ASCII de las centenas.
mil	8	input	Señal con el código ASCII de los millares.
ciclo_busy	1	output	Señal de proceso de selección de los datos ASCII ocupado.
tx_start	1	output	Señal de inicio de la transmisión por la UART.
tx_data	8	output	Señal con los datos a transmitir por la UART.

Tabla 12: Descripción del módulo “ascii_selector.vhd”.
Fuente: Elaboración propia.

En el Anexo 3 se puede consultar el esquema de este bloque “Procesado”.

Bloque “Transmisión”

El módulo del bloque “Transmisión” se llama “uart.vhd” (Tabla 13). Este bloque solamente emplea este módulo.

uart.vhd

El módulo “uart.vhd” representa el controlador del protocolo UART.

El controlador recibe en la entrada una señal con los datos a transmitir (8 bits) y un pulso de la señal “tx_ena”. En la salida transmite la trama, bit a bit, mediante “tx” y mientras lo está haciendo tiene activo en alto la señal “tx_busy”. Transmite a una tasa de baudios de 9600 bits por segundo.

SEÑAL	BITS	I/O	DESCRIPCIÓN
clk	1	input	Señal de reloj.
reset_n	1	input	Señal de inicialización asíncrono activo en baja.
tx_ena	1	input	Señal de inicio de la transmisión.
tx_data	8	input	Datos a transmitir.
rx	1	input	Datos a recibir.
rx_busy	1	output	Señal del proceso de recepción ocupado.
rx_error	1	output	Señal de error de la recepción.
rx_data	8	output	Datos recibidos.
tx_busy	1	output	Señal del proceso de transmisión ocupado.
tx	1	output	Datos transmitidos.

Tabla 13: Descripción del módulo “uart.vhd”.
Fuente: Elaboración propia.

En el Anexo 4 se puede consultar el esquema de este bloque “Transmisión”.

4.1.4. Análisis de datos

Una vez diseñada la primera versión hardware de la primera etapa “Adquisición de datos” se realizan varios movimientos y se muestrean las señales de los tres ejes y la temperatura para cada uno de ellos, dando lugar a una primera medición.

Primera medición

Respecto a la temperatura, se han tomado valores con el sensor en movimiento (Fichero “Temperatura-Movimiento.txt”) y parado (Fichero “Temperatura-Parado.txt”). En ambos casos, los valores oscilan entre 500 y 504 unidades. Como cabía esperar, la temperatura no presenta variación en función de si hay o no movimiento. Para el desarrollo de este trabajo, el estudio de la temperatura es irrelevante por lo que no se realizarán más comentarios acerca de esta magnitud de aquí en adelante.

Para los distintos ejes, se toman medidas realizando los siguientes movimientos:

- Parado.
- Desplazamiento hacia arriba y hacia abajo.
- Desplazamiento hacia la derecha y hacia la izquierda.
- Desplazamiento hacia adelante y hacia atrás.

Todos los datos obtenidos en esta primera medición están disponibles en ficheros ubicados en la siguiente ruta, en el CD adjunto: Otros/Anexos/1 – Etapa 1 (Adquisición de datos) – Primera versión hardware/Análisis de datos/Primera medición.

En las mediciones en las que el movimiento se realiza en el eje de la señal medida se aprecia con claridad, en la representación gráfica, el comportamiento de la aceleración a lo largo del desplazamiento (Figuras 51, 52 y 53). En las que no coinciden, no es posible interpretar gráficamente el movimiento.

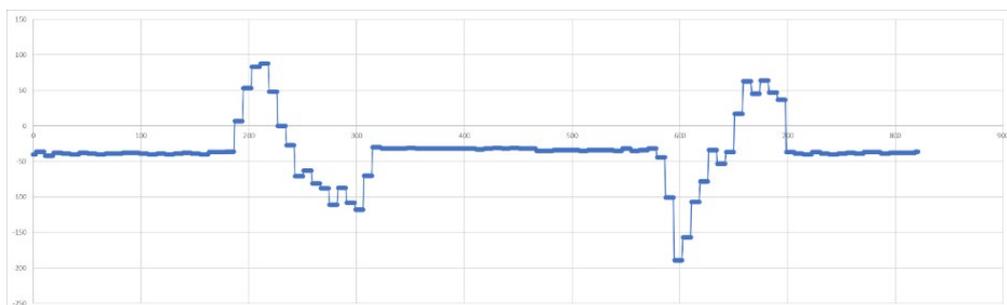


Figura 51: Representación gráfica de la aceleración en el eje X para un movimiento en la dirección del eje X en sentido positivo y negativo (en valores binarios) en función del tiempo (en valores muestreados, siendo cada unidad de valor muestreado equivalente al periodo de muestreo, es decir, a la inversa de la frecuencia de muestreo).

Fuente: Elaboración propia.

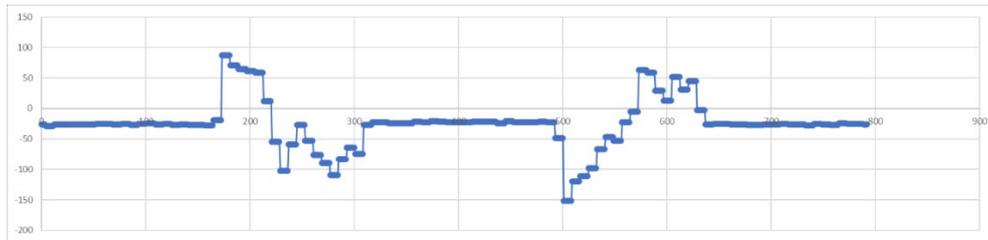


Figura 52: Representación gráfica de la aceleración en el eje Y para un movimiento en la dirección del eje Y en sentido positivo y negativo (en valores binarios) en función del tiempo (en valores muestrados, siendo cada unidad de valor muestrado equivalente al periodo de muestreo, es decir, a la inversa de la frecuencia de muestreo).

Fuente: Elaboración propia.

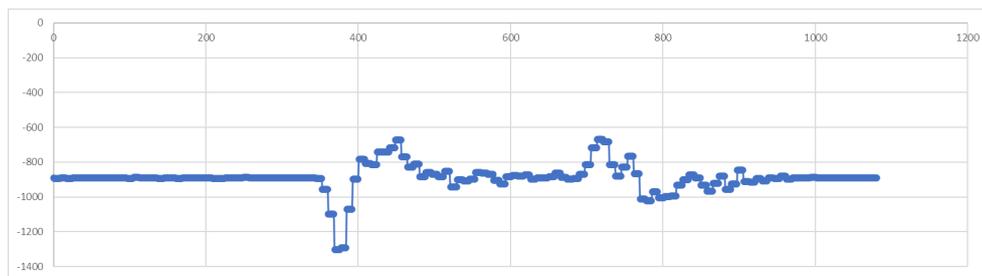


Figura 53: Representación gráfica de la aceleración en el eje Z para un movimiento en la dirección del eje Z en sentido negativo y positivo (en valores binarios) en función del tiempo (en valores muestrados, siendo cada unidad de valor muestrado equivalente al periodo de muestreo, es decir, a la inversa de la frecuencia de muestreo).

Fuente: Elaboración propia.

En las tres representaciones gráficas se distinguen las dos direcciones de desplazamiento, la ida y la vuelta a la posición inicial.

Inicialmente, la aceleración aumenta hasta casi llegar al final del recorrido, donde pasa a ser una aceleración de frenado, en sentido contrario. Durante un breve periodo, se detiene. A continuación, se realiza la vuelta, produciéndose una alternancia de signos en la aceleración en comparación con el movimiento de ida.

Se aprecia error de cero en las señales de todos los ejes. Además, dado que un movimiento en el eje Z no se realiza sobre una superficie estable, cuando está parado entre la ida y la vuelta, el valor de aceleración no se estabiliza, lo que afectará en el cálculo de la distancia recorrida. Para simplificar el trabajo, se medirá la distancia recorrida sólo en los ejes X e Y, es decir, en un plano horizontal. Además, en caso de no hacerlo, se introduce en los componentes X e Y del vector aceleración, la proyección de la gravedad terrestre sobre estos ejes, lo que provoca que, aunque la placa esté parada tenga una aceleración, lo que provoca que el sistema se vuelva inestable.

Aunque los resultados son prometedores, no se puede garantizar todavía la validez de la hipótesis. Es necesario comprobar que, tras aplicar el proceso de integración de la segunda etapa "Procesado de datos", los resultados obtenidos sean satisfactorios.

4.2. Primera versión software de la segunda etapa “Procesado de datos”

4.2.1. Introducción

Llegados a este punto, para poder comprobar que la hipótesis se cumple, es necesario aplicar el proceso de integración de la segunda etapa a los datos obtenidos por la primera etapa para poder analizar si los resultados del procesado son aceptables.

Para ello, se desarrolla una primera versión software de la segunda etapa correspondiente al “Procesado de datos” (Figura 54).

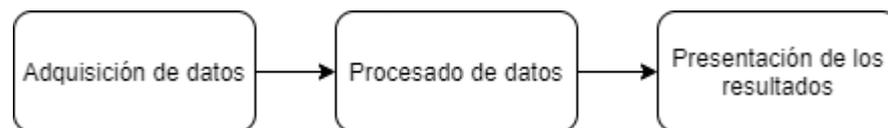


Figura 54: Procesos principales que realiza el diseño hardware.

Fuente: Elaboración propia.

En primer lugar, la motivación que ha llevado a realizar una versión preliminar por software de la etapa de “Procesado de datos” es porque así resulta más sencillo y rápido implementar el algoritmo que realiza el proceso de integración antes de pasarlo a hardware. Esto permite, además, comprobar si la hipótesis se cumple antes de realizar el diseño hardware de la segunda etapa, ahorrando tiempo.

En los siguientes subapartados, se expone la base teórica del algoritmo empleado y la descripción del mismo.

4.2.2. Base teórica del algoritmo

El presente trabajo busca desarrollar una tecnología con la capacidad de determinar la posición final de un objeto a partir de su posición inicial y de su aceleración.

Mediante la primera versión hardware de la primera etapa “Adquisición de datos” ya se ha conseguido obtener la aceleración de forma discreta durante el movimiento. Por lo tanto, siendo conocida la posición inicial, ya se conocen ambos datos de partida.

La base teórica que permite obtener la posición final de un objeto a partir de su posición inicial y de su aceleración se encuentra en las ecuaciones que describen el Movimiento Rectilíneo Uniformemente Acelerado (MRUA).

A simple vista, dado que durante el movimiento en el presente trabajo la aceleración varía de forma no uniforme, no parece viable seguir por esta vía. Sin

embargo, si se considera a cada uno de los valores de aceleración obtenidos en la primera etapa, como el valor uniforme de aceleración de un MRUA independiente, las ecuaciones son válidas.

Por tanto, podemos considerar nuestro Movimiento Rectilíneo No Uniformemente Acelerado como un conjunto de Movimientos Rectilíneos Uniformemente Acelerados, donde los valores finales en las ecuaciones de un MRUA, son los valores iniciales en las ecuaciones del MRUA siguiente.

De esta forma, se pueden aplicar las ecuaciones del MRUA de forma iterativa tantas veces como valores de aceleración haya, obteniendo los valores finales de velocidad y posición.

En el siguiente subapartado, se describe el algoritmo que realiza este proceso iterativo.

4.2.3. Descripción del script de Matlab

Para desarrollar en software el algoritmo correspondiente al proceso iterativo se ha decidido emplear Matlab, por su idoneidad para trabajar con grandes volúmenes de datos, hacer cálculos matemáticos y realizar representaciones gráficas.

El script realizado se adjunta en la siguiente ruta, en el CD adjunto: Otros/Anexos/2 – Etapa 2 (Procesado de datos) – Primera versión software/Script de Matlab.

El script de Matlab creado (el código se puede consultar en el Anexo 5) lleva a cabo el proceso iterativo basado en las ecuaciones que describen la cinemática de un MRUA y al finalizar, representa gráficamente tanto los datos iniciales, como los resultados, obteniéndose así la representación de las tres magnitudes (aceleración, velocidad y aceleración) en cada momento de la iteración. A continuación, se describe el script.

Antes de realizar el cálculo de la distancia recorrida, en las tres primeras líneas del script se eliminan variables previas y se cierran otras gráficas.

```
1 clc
2 close all
3 clear all
```

Para calcular la distancia recorrida, lo primero es leer los datos contenidos en una hoja de Excel “datos.xlsx” y guardarlos en un vector llamado “aceleracion”. A partir de la longitud del vector “aceleracion” se obtiene el número de valores de aceleración que se corresponde con las veces que habrá que realizar la iteración.

```

8 [aceleracion]=xlsread('Datos.xlsx');
9 tamaño=size(aceleracion);

```

A continuación, se inicializan los valores iniciales de velocidad, posición y tiempo:

```

12 t=1;           %Independiente de 200Hz/400Hz
13 v0=0;         %Inicialmente sin velocidad (parado)
14 y0=0;         %Inicialmente sin distancia

```

El proceso de iteración propiamente dicho se ejecuta mediante un bucle “for” que varía el número de veces que se ejecuta en función de la longitud del vector “aceleracion”. En el bucle se calculan los nuevos valores finales de la velocidad y la posición, se actualizan los valores iniciales para el próximo ciclo con los nuevos valores finales y se almacenan los resultados para graficarlos al terminar el bucle.

```

17 for i=1:tamaño[1]
18
19     vf = v0 + (aceleracion(i)*t);
20     yf = y0 + (v0*t) + (0.5*aceleracion(i)*t*t);
21
22     % Actualización de valores
23     v0 = vf;
24     y0 = yf;
25
26     % Almacena los resultados para graficarlos
27     x(i)=i;
28     velocidad(i)=vf;
29     movimiento(i)=yf;
30
31 end

```

Finalmente, se grafican los resultados:

```

34 plot(x,movimiento,x,velocidad,x,aceleracion)
35 legend('movimiento','velocidad','aceleración')
36 grid on
37 hold

```

4.2.4. Análisis de datos

Una vez desarrollada la primera versión software de la segunda etapa “Procesado de datos”, es posible aplicar el proceso de integración a los datos obtenidos por la primera versión hardware de la primera etapa “Adquisición de datos”. Esto permite calcular el valor de la distancia recorrida de una medida concreta.

Para comprobar si finalmente la hipótesis se cumple o no, se evaluarán los resultados obtenidos después de aplicar el procesado. Dicha evaluación consistirá en comparar las distancias recorridas calculadas con el procesado entre sí y en función de la distancia recorrida real. De esta forma, para que la hipótesis sea válida tendrán que darse dos requisitos:

- Repetibilidad. Los valores de distancia calculados de distintas muestras de la misma distancia real tendrán que coincidir o aproximarse razonablemente.
- Linealidad. Deberá poder reconocerse proporcionalidad en los valores, de tal forma que, si el valor real de una medida es el doble que el de una medida anterior, el valor calculado también debe serlo.

Para poder realizar un análisis más riguroso, se toman más datos con la primera etapa “Adquisición de datos”, dando lugar a una segunda medición.

Segunda medición

Las medidas de esta segunda medición se encuentran disponibles en un directorio ubicado en la siguiente ruta, en el CD adjunto: Otros/Anexos/1 – Etapa 1 (Adquisición de datos) – Primera versión hardware/Análisis de datos/Segunda medición.

El diseño de la primera versión hardware de la etapa “Adquisición de datos” contiene una entidad que controla el acelerómetro ADXL362. Esta entidad permite modificar una serie de parámetros que afectan a la manera en la que el sensor realiza el muestreo de datos. Estos parámetros son:

- La frecuencia de muestreo.
- El número de lecturas para la realización de una media aritmética.

Tal y como se describe en el datasheet del ADXL362 (Anexo 6), la frecuencia de muestreo se configura en la dirección 0x2C del Filter Control Register. Puede ser 12.5 Hz, 25 Hz, 50 Hz, 100 Hz, 200 Hz y 400 Hz. Por defecto, el diseño tiene 200 Hz. Ya que interesa tener la mayor precisión posible, se trabajará con una frecuencia de muestreo de 200 Hz (0x14) y 400 Hz (0x17). Para modificar entre estos valores basta con cambiar los parámetros en la línea 201 del módulo “ADXL362Ctrl.vhd” del proyecto “v1.1.xise”.

```

198 constant Cmd_Reg_Data : rom_type := (X"1F", X"52",
199                                     X"1F", X"00",
200                                     X"2D", X"02",
201                                     X"2C", X"14"
202                                     );

```

El otro parámetro que se puede modificar es el número de lecturas para la realización de una media aritmética (NUM_READS_AVG). Cuando el sensor funciona, lee continuamente valores y hace una media aritmética cuando realiza un número de lecturas determinado por NUM_READS_AVG. Este número debe ser potencia de dos. Por defecto, NUM_READS_AVG es 16. Se trabajará con NUM_READS_AVG de 8 y 16. Para modificar entre estos valores basta con cambiar los valores en la línea 94 del módulo "ADXL362Ctrl.vhd" del proyecto "v1.1.xise".

```

90 generic
91 (
92     SYSCLK_FREQUENCY_HZ : integer := 100000000;
93     SCLK_FREQUENCY_HZ   : integer := 1000000;
94     NUM_READS_AVG       : integer := 16;
95     UPDATE_FREQUENCY_HZ : integer := 1000
96 );

```

De esta manera, hay cuatro posibles combinaciones de parámetros para realizar medidas:

- Frecuencia de muestreo de 200 Hz y NUM_READS_AVG de 8.
- Frecuencia de muestreo de 200 Hz y NUM_READS_AVG de 16.
- Frecuencia de muestreo de 400 Hz y NUM_READS_AVG de 8.
- Frecuencia de muestreo de 400 Hz y NUM_READS_AVG de 16.

En esta segunda medición se realizan cinco medidas para un movimiento hacia la derecha de 30 cm de longitud, para cada una de las cuatro combinaciones de parámetros.

Para evaluar la calidad de las muestras obtenidas correspondientes a cada configuración de parámetros, así como, para ver cuál es la mejor de las configuraciones, se aplica el procesado de datos de la segunda etapa a cada medida y se analizan los resultados. Los resultados obtenidos se recogen en la siguiente tabla (Tabla 14).

Desplazamiento hacia la derecha (30 cm)				
Medida	16 muestras 200 Hz	8 muestras 200 Hz	16 muestras 400 Hz	8 muestras 400 Hz
Medida 1	0.14190	0.46220	0.31700	1.71400
Medida 2	0.14330	0.56090	0.20470	1.07100
Medida 3	0.15700	0.59230	0.12570	1.94800
Medida 4	0.12040	0.59020	0.33120	1.86300
Medida 5	0.08355	0.54350	0.21540	1.04400
Media	0.12923	0.54982	0.23880	1.52800
Máximo	0.15700	0.59230	0.33120	1.94800
Mínimo	0.08355	0.46220	0.12570	1.04400
Rango	0.07345	0.13010	0.20550	0.90400
Varianza	0.00066	0.00226	0.00583	0.15327
(Rango/Mínimo) · 100	87.91%	28.15%	163.48%	86.59%

Tabla 14: Análisis de datos correspondientes a la segunda medición.
Fuente: Elaboración propia.

Al igual que ocurre en los datos de la primera medición, se observa que en los datos obtenidos tras el proceso de adquisición de esta segunda medición también aparece error de cero, es decir, que el sensor no muestrea aceleración nula estando la placa parada, sino un valor pequeño. Esto afecta de manera notable al aplicar el procesado, ya que, al tratarse de un proceso iterativo, los errores, aunque pequeños, se van acumulando ciclo a ciclo. Para evitar este problema y mejorar los resultados del análisis, se opta por llevar a cabo los siguientes procedimientos.

En primer lugar, estando la placa parada, se toman muestras. A estos valores se les calcula la media que se corresponde con el error de cero aproximado. Este valor se resta a todos los elementos de las muestras obtenidas antes de aplicarles el procesado por software, mejorando así los resultados. Este proceso se corresponde con una calibración del instrumento. Si la hipótesis finalmente se valida y se continúa con el trabajo, habrá que implementar en hardware este proceso de calibración para no tener que hacerlo de forma manual.

Se ha comprobado que el error de cero es un error sistemático, es decir, propio de cada sistema. Varía dependiendo la placa que se emplee, ya que el error de cero en esta segunda medición es distinto del que hay en la primera debido a que se emplean placas distintas. Otro factor que podría afectar es el de realizar la medida en un entorno diferente (otra superficie, otras condiciones ambientales...), lo que haría variar el error de cero. Al calibrar el instrumento antes de su uso, esto se tiene en cuenta y no supone un problema.

Los valores obtenidos con la placa parada, además de ser pequeños, varían ligeramente. Por este motivo, al restarles el error de cero aproximado, sigue habiendo valores distintos de cero. Estos errores aleatorios en la señal siguen afectando en el proceso de integración iterativo. Para eliminarlos de manera rápida, se deciden suprimir manualmente de forma selectiva estos valores de la función aceleración antes de aplicar el script de Matlab, lo que permite obtener mejores resultados tras el procesado. Esta mejora también será implementada en hardware a través de un filtro, en caso de que se apruebe la hipótesis.

Además de estos problemas, se ha detectado otro provocado por errores aleatorios. De forma aleatoria, en la muestra aparecen valores excesivamente grandes o pequeños por fallos en la transmisión de información entre la UART y el ordenador. En el proceso iterativo estos valores perturban el procesado, obteniéndose malos resultados. Estos valores se quitan de forma manual, suavizando la gráfica de la aceleración, lo que permite obtener mejores resultados. No será necesario desarrollar un proceso que filtre estos datos aleatorios en la versión hardware, porque no se transmitirá desde la UART al ordenador.

Tras evaluar los resultados obtenidos tras las mejoras implementadas (Tabla 14), se concluye que la mejor combinación de parámetros es la que tiene una frecuencia de muestreo de 200 Hz y NUM_READS_AVG de 8. Esto se debe a que tiene, con diferencia, la variación porcentual de rango más baja y también, una varianza aceptable. Hay que comentar que, siendo la distancia real recorrida de 30 cm, cabría esperar que la distancia calcula también lo fuese o se aproximase al menos. Esto no es así, porque los datos de aceleración no están en unidades derivadas del Sistema Internacional (m/s^2), sino en valor binario.

Dado que cinco medidas podrían no ser suficientes, se decide realizar una tercera medición para más movimientos y longitudes. Para evitar cometer un error eligiendo la configuración de parámetros más adecuada, este tercer análisis se va a realizar también para la configuración de parámetros opuesta, es decir, frecuencia de muestreo de 400 Hz y NUM_READS_AVG de 16. De esta forma, se verá si hay diferencias en función de la configuración de parámetros elegida.

Tercera medición

En esta tercera medición se realiza el estudio final para determinar cuál de las dos combinaciones de parámetros seleccionada resulta más interesante y si ésta permite validar la hipótesis.

Las mediciones se realizarán con dos configuraciones de parámetros:

- Opción 1: frecuencia de muestreo de 200 Hz y NUM_READS_AVG de 8 muestras.
- Opción 2: frecuencia de muestreo de 400 Hz y NUM_READS_AVG de 16 muestras.

Los datos obtenidos pertenecen a la señal “Eje Y”, con 10 medidas para cada uno de los siguientes movimientos:

- Desplazamiento hacia la derecha de 10 cm (lento, medio, rápido).
- Desplazamiento hacia la derecha de 20 cm (lento, medio, rápido).
- Desplazamiento hacia la derecha de 30 cm (lento, medio, rápido).
- Desplazamiento hacia la derecha de 40 cm (lento, medio, rápido).
- Desplazamiento hacia la derecha y hacia la izquierda de 10 cm (lento, medio, rápido).
- Desplazamiento hacia la derecha y hacia la izquierda de 20 cm (lento, medio, rápido).
- Desplazamiento hacia la derecha y hacia la izquierda de 30 cm (lento, medio, rápido).
- Desplazamiento hacia la derecha y hacia la izquierda de 40 cm (lento, medio, rápido).

Las medidas de esta tercera medición se encuentran disponibles en una carpeta ubicada en la siguiente ruta, en el CD adjunto: Otros/Anexos/1 – Etapa 1 (Adquisición de datos) – Primera versión hardware/Análisis de datos/Tercera medición.

Tras implementar las mejoras comentadas en la segunda medición también a estas medidas, se aplica el Script de Matlab, obteniendo los resultados contenidos en el fichero “Datos procesados y estudio.xlsx” adjunto en la siguiente ruta, en el CD adjunto: Otros/Anexos/1 – Etapa 1 (Adquisición de datos) – Primera versión hardware/Análisis de datos/Tercera medición.

A estos resultados, correspondientes a las distancias recorridas calculadas con el procesado de datos, se les realiza un estudio (Tabla 15 y Figuras 55 a 64) para poder extraer información relevante.

		Estudio					
		lento		medio		rápido	
		8 / 200	16 / 400	8 / 200	16 / 400	8 / 200	16 / 400
10 cm	Máximo	17850	9013	8273	7986	7716	7900
	Mínimo	7808	7302	6906	7295	6971	6947
	Rango	10042	1711	1367	691	745	953
	(Rango/Mínimo):100	129%	23%	20%	9%	11%	14%
	Media	10322,8	8274,5	7653,1	7680,7	7303,3	7399,4
	Varianza	7618046,96	372421,05	197882,49	42923,81	44984,21	79590,84
20 cm	Máximo	15910	18660	15440	15210	14920	16350
	Mínimo	8838	15380	13360	14690	13650	14720
	Rango	7072	3280	2080	520	1270	1630
	(Rango/Mínimo):100	80%	21%	16%	4%	9%	11%
	Media	12112,9	16844	14678	14967	14234	15348
	Varianza	5583850,09	1149284	395696	18641	177504	316336
30 cm	Máximo	37640	41810	23140	23300	22450	23230
	Mínimo	18380	3675	20550	21990	21680	21210
	Rango	19260	38135	2590	1310	770	2020
	(Rango/Mínimo):100	105%	1038%	13%	6%	4%	10%
	Media	28086	28830,9	21968	22692	22015	21825
	Varianza	32377104	169894327	610816	125576	60825	317565
40 cm	Máximo	46090	50820	30580	30510	29030	30440
	Mínimo	30600	31790	28940	29230	27870	29120
	Rango	15490	19030	1640	1280	1160	1320
	(Rango/Mínimo):100	51%	60%	6%	4%	4%	5%
	Media	36538	37449	29741	29929	28524	29554
	Varianza	19779336	24932989	180549	145389	123444	138884
10 cm (ida y vuelta)	Máximo	15150	18180	6015	2617	777	669
	Mínimo	5355	440	2183	372	-264	-393
	Rango	9795	17740	3832	2245	1041	1062
	(Rango/Mínimo):100	183%	4032%	176%	603%	-394%	-270%
	Media	11577,9	9208,4	4204,5	1618,8	359,9	142,7
	Varianza	11657755,69	25157769,6	1309305,25	404469,36	103730,89	102700,81
20 cm (ida y vuelta)	Máximo	46760	33610	4405	2475	1279	1794
	Mínimo	16520	14310	-282	211	-113	-777
	Rango	30240	19300	4687	2264	1392	2571
	(Rango/Mínimo):100	183%	135%	-1662%	1073%	-1232%	-331%
	Media	28097	21857	2369,6	1761,6	627,1	411
	Varianza	73282741	34833461	2056816,64	329601,24	132534,49	730223,2
30 cm (ida y vuelta)	Máximo	95930	61220	19120	5725	426	1830
	Mínimo	36060	41280	4807	298	-1471	-468
	Rango	59870	19940	14313	5427	1897	2298
	(Rango/Mínimo):100	166%	48%	298%	1821%	-129%	-491%
	Media	63148	50994	10545,8	2436,4	-259	671
	Varianza	385419636	32460424	16657708,8	2874481,04	319829,6	676231,2
40 cm (ida y vuelta)	Máximo	59580	46670	12510	7575	1415	1542
	Mínimo	18790	15670	3692	242	-145	-3950
	Rango	40790	31000	8818	7333	1560	5492
	(Rango/Mínimo):100	217%	198%	239%	3030%	-1076%	-139%
	Media	39298	29576	5949,1	4066,1	648,4	4,4
	Varianza	183770296	61121444	6632901,49	5865067,69	318906,84	3478707,24

Tabla 15: Análisis de datos correspondientes a la tercera medición.

Fuente: Elaboración propia.

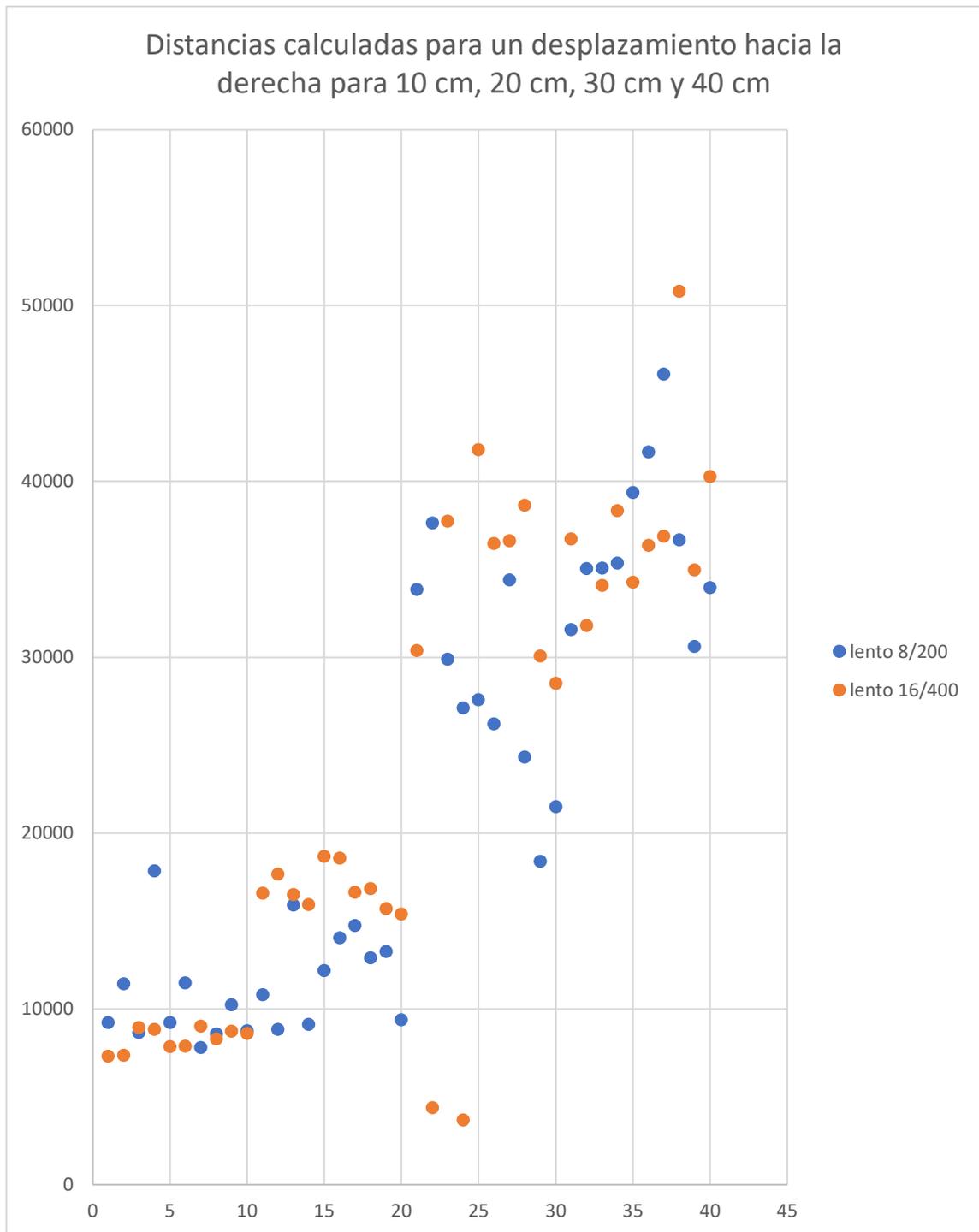


Figura 55: Representación gráfica de las distancias recorridas calculadas en el procesado para movimientos lentos hacia la derecha de sólo ida (en valores binarios) en función de la medida realizada (adimensional).

Fuente: Elaboración propia.

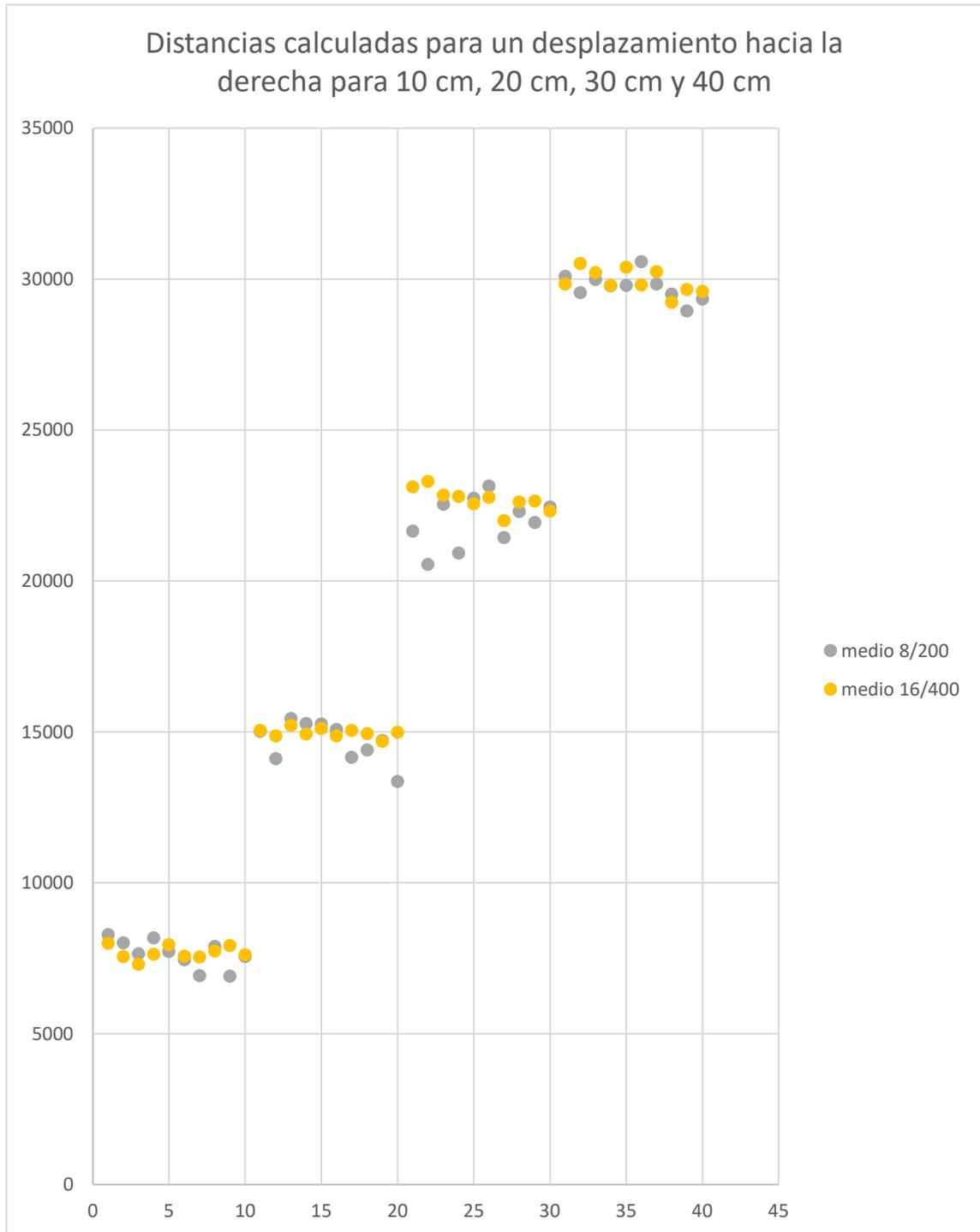


Figura 56: Representación gráfica de las distancias recorridas calculadas en el procesado para movimientos a velocidad media hacia la derecha de sólo ida (en valores binarios) en función de la medida realizada (adimensional).

Fuente: Elaboración propia.

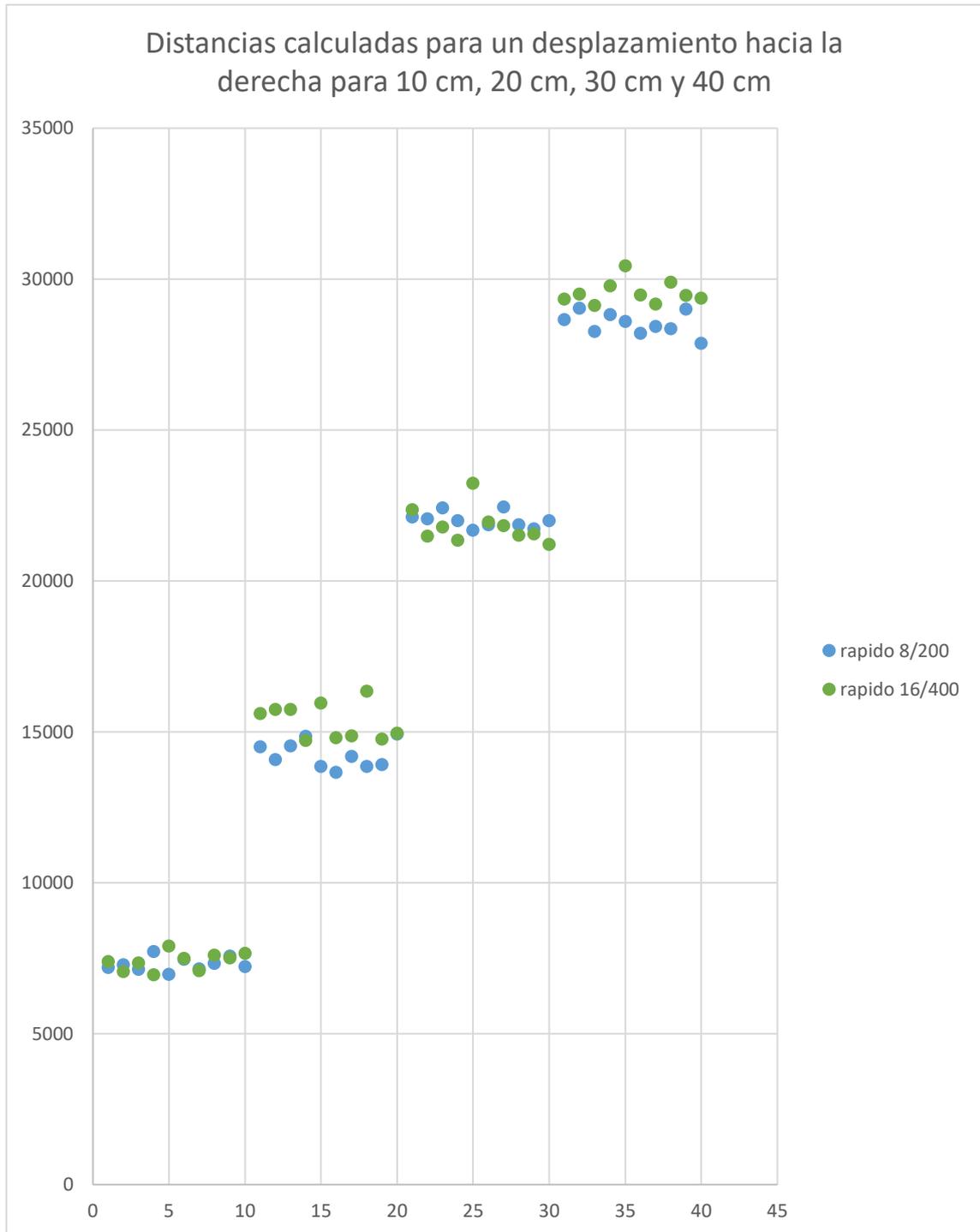


Figura 57: Representación gráfica de las distancias recorridas calculadas en el procesado para movimientos rápidos hacia la derecha de sólo ida (en valores binarios) en función de la medida realizada (adimensional).

Fuente: Elaboración propia.

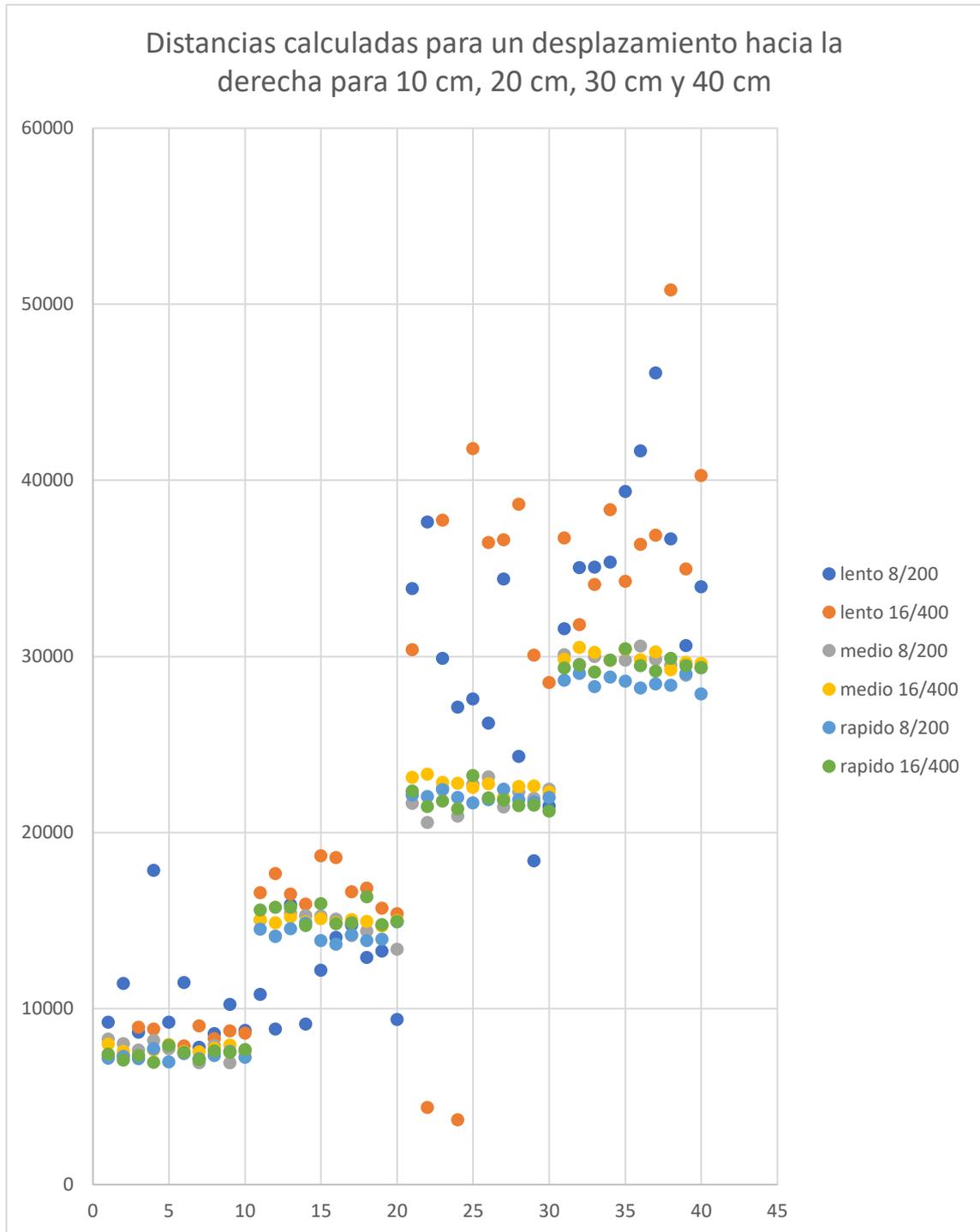


Figura 58: Representación gráfica de las distancias recorridas calculadas en el procesado para movimientos a velocidad lenta, media y rápida hacia la derecha de sólo ida (en valores binarios) en función de la medida realizada (adimensional).

Fuente: Elaboración propia.

Los movimientos lentos hacia la derecha de sólo ida tienen una varianza y una variación porcentual de rango mucho mayor que los que se realizan con una velocidad media o rápida, lo que indica que son peores. En la representación gráfica de estos valores (Figura 55) se hace visible esto. No se aprecia repetibilidad ni linealidad clara para ninguna de las dos configuraciones de parámetros.

Si es cierto que la combinación de 16 muestras y 400 Hz, para las primeras distancias de 10 y 20 cm, parece tener linealidad y repetibilidad de forma leve, ya que sus valores parecen agruparse en torno 10000, los primeros, y 20000, los segundos. Esto no es suficiente para validar la hipótesis para este tipo de velocidad.

En los movimientos a velocidad media y rápida, los resultados mejoran de forma evidente independientemente de la configuración de parámetros analizada. De hecho, no se identifica de forma clara por los datos de varianza o variación porcentual de rango de la Tabla 15 que una velocidad o combinación de parámetros sea mejor.

En las representaciones gráficas para movimientos a velocidad media (Figura 56) y rápida (Figura 57) se aprecia claramente linealidad. Los datos correspondientes a una distancia real recorrida de 10 cm tienen un valor calculado de aproximadamente 7500. Para 20 cm, el doble, 15000. Y lo mismo sucede para 30 cm y 40 cm, donde se obtienen valores cercanos a 22500 y 30000. Esta proporcionalidad permite obtener la relación que existe entre la distancia real recorrida y la distancia calculada, siendo 1 cm real equivalente a 750.

Cabe destacar que en la gráfica a velocidad media (Figura 56) se aprecia mayor repetibilidad en los valores con la configuración de 16 muestras y 400 Hz.

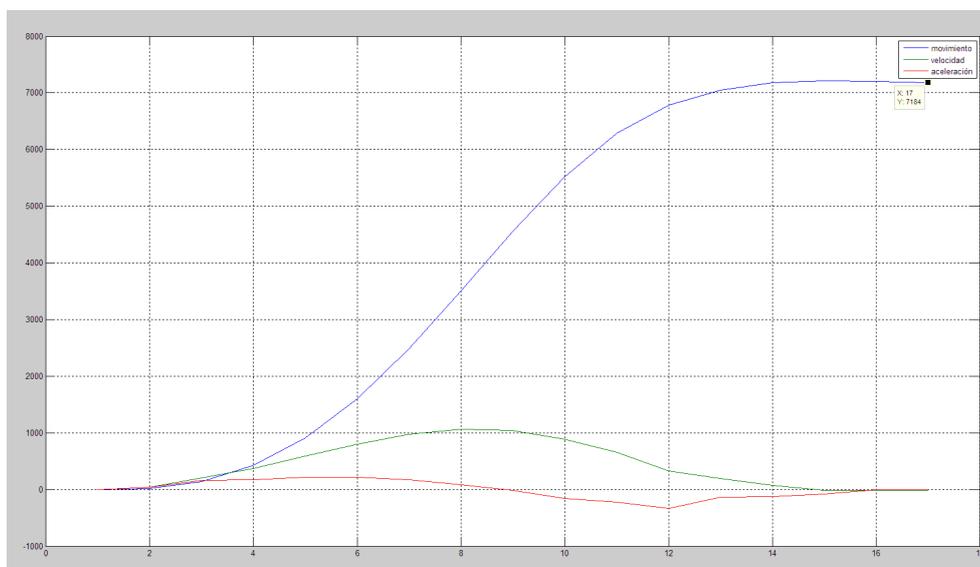


Figura 59: Representación gráfica de la aceleración (rojo), velocidad (verde) y posición (azul) para un movimiento de ida (en valores binarios) en función del tiempo (en valores muestrados, siendo cada unidad de valor muestreado equivalente al periodo de muestreo, es decir, a la inversa de la frecuencia de muestreo).

Fuente: Elaboración propia.

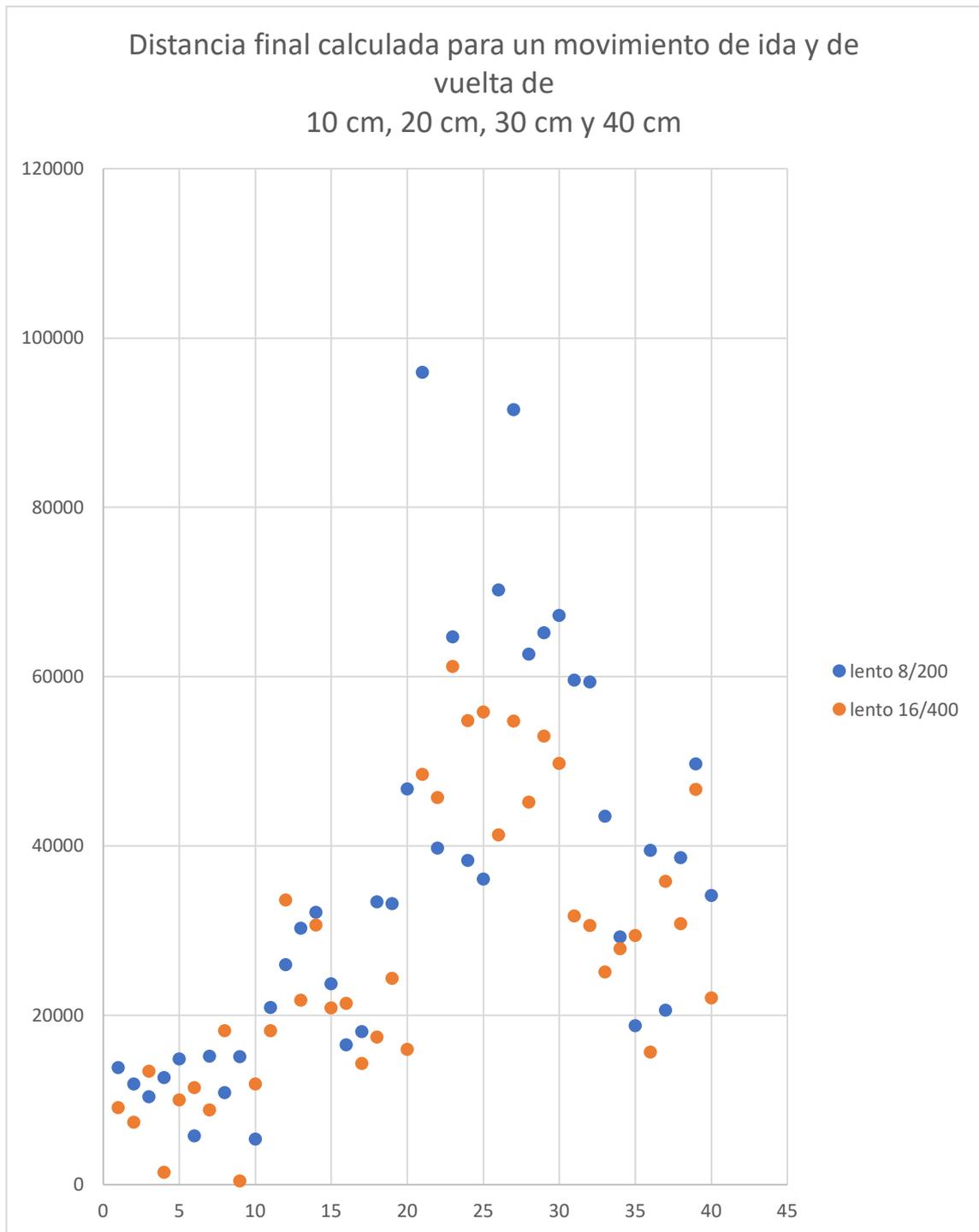


Figura 60: Representación gráfica de las distancias finales recorridas calculadas en el procesado para movimientos lentos hacia la derecha y hacia la izquierda (ida y vuelta) (en valores binarios) en función de la medida realizada (adimensional).

Fuente: Elaboración propia.

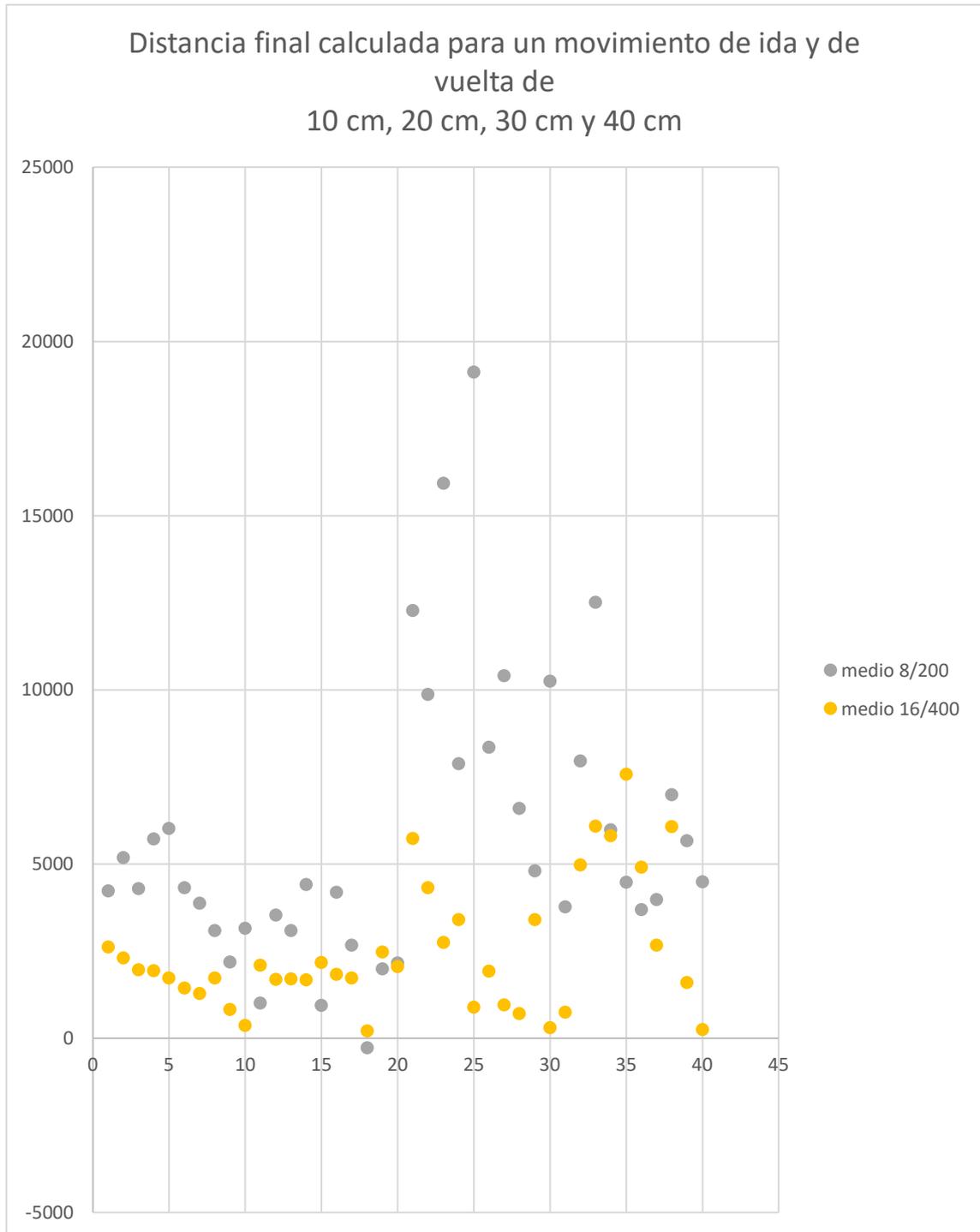


Figura 61: Representación gráfica de las distancias finales recorridas calculadas en el procesado para movimientos a velocidad media hacia la derecha y hacia la izquierda (ida y vuelta) (en valores binarios) en función de la medida realizada (adimensional).

Fuente: Elaboración propia.

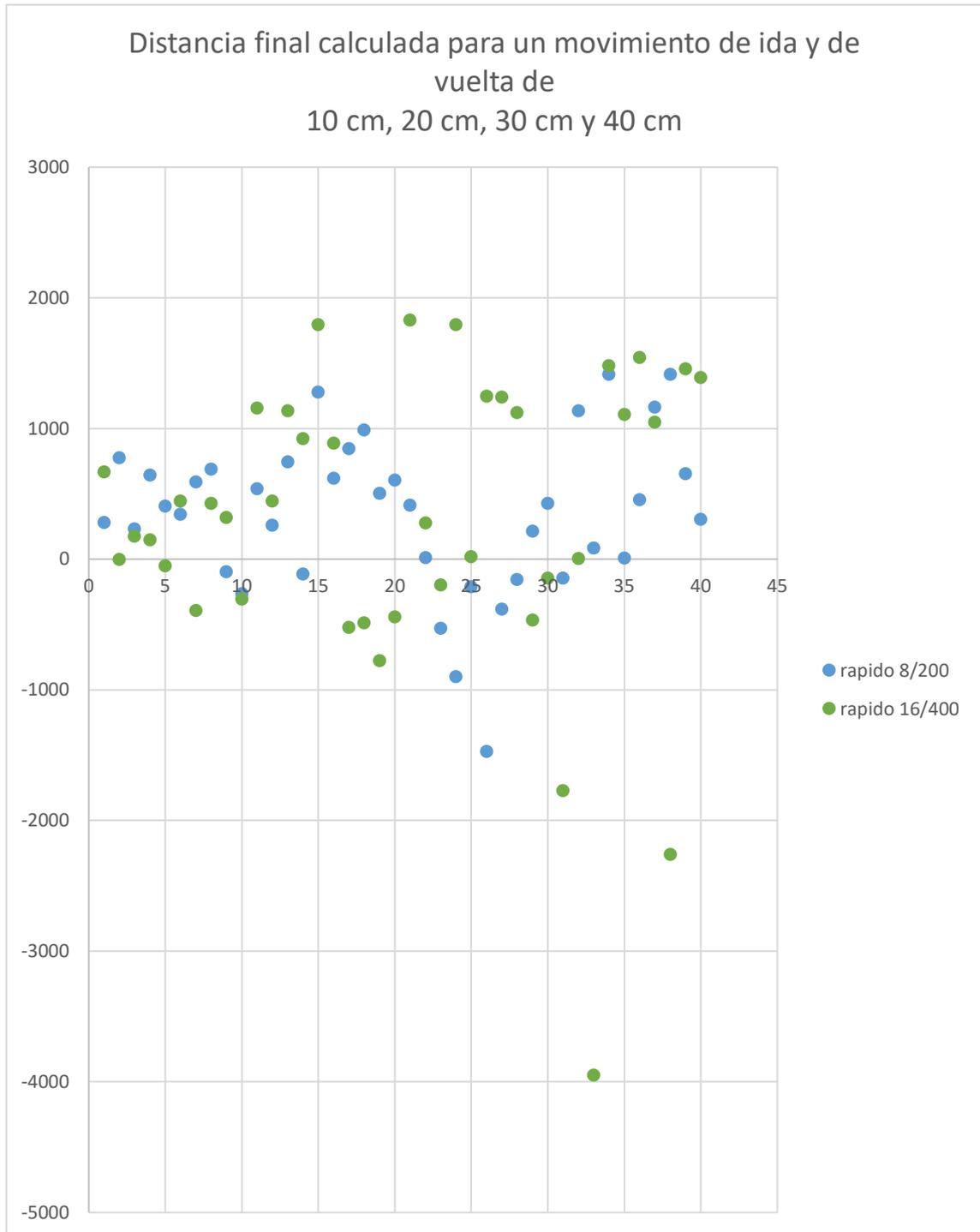


Figura 62: Representación gráfica de las distancias finales recorridas calculadas en el procesado para movimientos rápidos hacia la derecha y hacia la izquierda (ida y vuelta) (en valores binarios) en función de la medida realizada (adimensional).

Fuente: Elaboración propia.

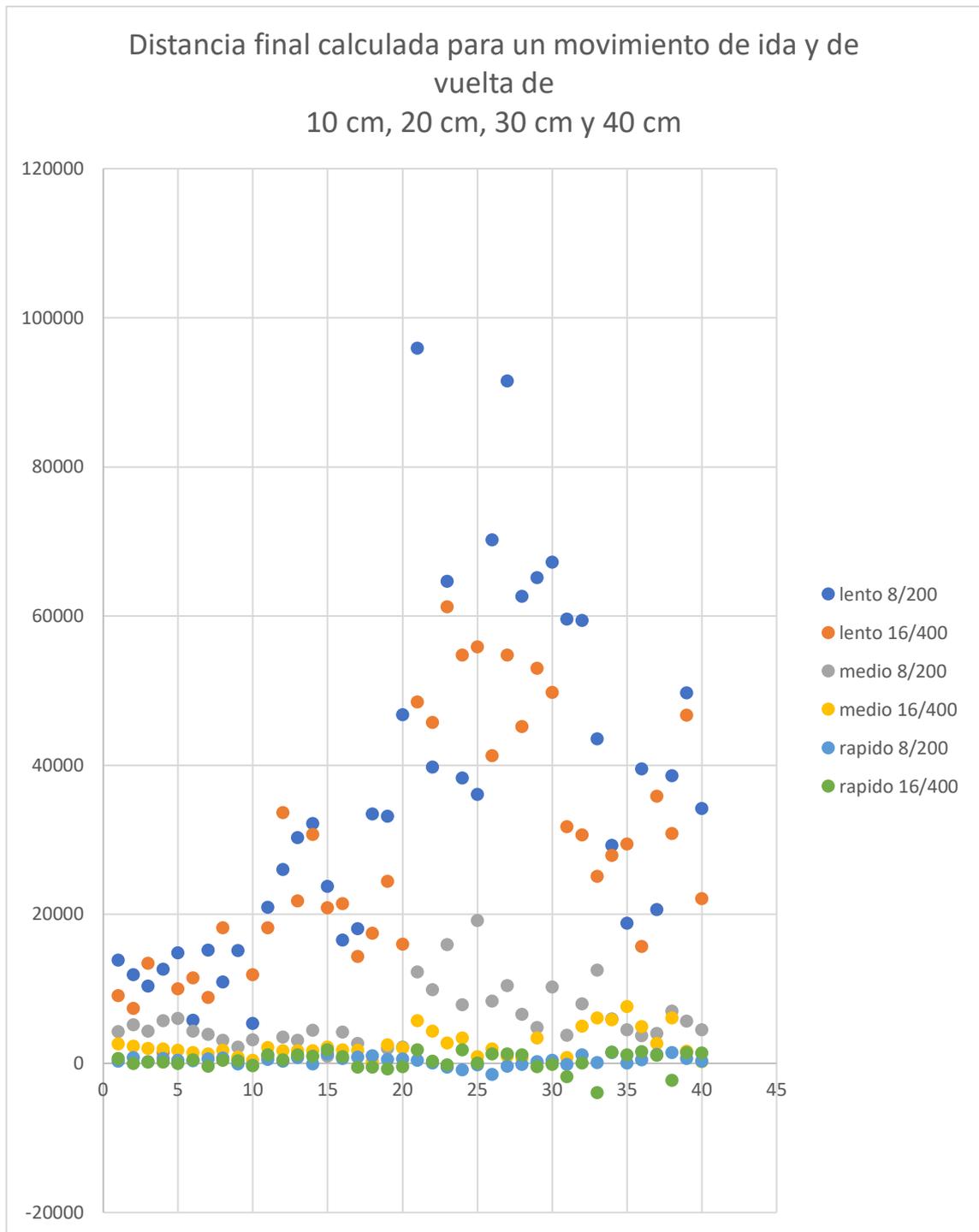


Figura 63: Representación gráfica de las distancias finales recorridas calculadas en el procesado para movimientos a velocidad lenta, media y rápida hacia la derecha y hacia la izquierda (ida y vuelta) (en valores binarios) en función de la medida realizada (adimensional).

Fuente: Elaboración propia.

A continuación, se evalúan los resultados obtenidos para los movimientos hacia la derecha y hacia la izquierda, es decir, de ida y de vuelta. Esta parte del estudio es muy importante para validar la hipótesis, porque pone a prueba al sistema de la manera en la que se comportará realmente, es decir, realizando movimientos continuos, uno tras otro.

En este caso, realiza dos movimientos seguidos, uno de ida y otro de vuelta. Dado que ambos tienen la misma longitud, pero en sentidos contrarios, el desplazamiento total final debería ser nulo. En la Figura 64 se puede ver que los resultados obtenidos tras el procesado son prometedores, ya que se aprecia perfectamente como la aceleración (en rojo), la velocidad (en verde) y el desplazamiento (en azul) siguen el comportamiento esperado. Cuando la aceleración es positiva, la velocidad aumenta y cuando la velocidad es positiva, el desplazamiento aumenta (ida). Cuando la aceleración es negativa, la velocidad disminuye y cuando la velocidad es negativa, el desplazamiento se produce en el sentido contrario (vuelta) por lo que disminuye.

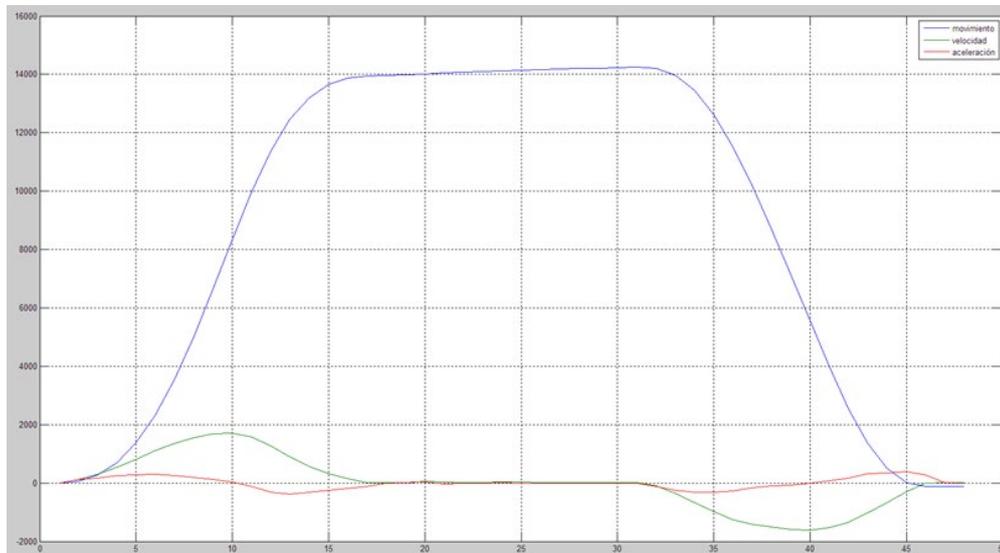


Figura 64: Representación gráfica de aceleración (rojo), velocidad (verde) y posición (azul) para un movimiento de ida y vuelta (en valores binarios) en función del tiempo (en valores muestreados, siendo cada unidad de valor muestreado equivalente al periodo de muestreo, es decir, a la inversa de la frecuencia de muestreo).

Fuente: Elaboración propia.

En cuanto a la información que se encuentra en la Tabla 15, se vuelve a ver que tanto los rangos como las varianzas correspondientes a movimientos lentos indican que los valores de dichos movimientos no cumplen la hipótesis. Esto da lugar a una restricción de estos movimientos de forma definitiva.

En cuanto a los movimientos a velocidad media y rápida, de sus gráficas (Figura 61 y 62) se extrae que ambos son válidos, ya que sus valores están cercanos a cero, independientemente de la distancia recorrida. Es digno de mención que los movimientos rápidos presentan valores más próximos a cero,

lo que lleva a pensar que cuanto más rápido se mueva la placa, mejores resultados se obtendrán.

Por último, la configuración de parámetros que se considera mejor es la de 16 muestras y 400 Hz. Además de tener mayor repetibilidad en la Figura 56, en la Figura 61 sus valores están más próximos a cero que los de la combinación de 8 muestras y 200 Hz. Por tanto, dado que a velocidades medias esta configuración muestra un mejor comportamiento, ésta será la seleccionada.

4.3. Resultados

Los resultados obtenidos una vez realizado el estudio preliminar se pueden agrupar en dos grandes bloques:

- Resultados para el desarrollo del sistema de medida.
- Resultados del análisis de datos.

4.3.1. Resultados para el desarrollo del sistema de medida

Inicialmente, se aplica directamente el Script de Matlab a distintas muestras, obteniéndose valores bastante variados. Aunque algunos son aceptables, la mayoría no lo son.

Se detecta que los valores adquiridos por la primera etapa presentan error de cero, ya que la medida presenta aceleración distinta de cero cuando el movimiento es nulo. Aunque este error no es muy grande, al tratarse de un proceso iterativo, el error se va acumulando hasta que termina por volverse inasumible. Para corregir este problema se lleva a cabo un proceso de calibración y de filtrado manual que permite mejorar los resultados.

Por otro lado, se observa que un gran número de muestras de valores presentan malos resultados por la aparición de errores aleatorios. Puntualmente aparecen elementos en la muestra con un valor exageradamente grande o pequeño que distorsionan la iteración. Se ha probado a realizar el procesado de datos tras eliminar estos valores y los resultados mejoran significativamente.

Los resultados obtenidos después de llevar a cabo estos procedimientos de mejora son muy satisfactorios. Aunque todavía hay muestras que presentan resultados malos, la mayoría presentan resultados aceptables, siendo posible comprobar existencia de repetibilidad y linealidad en las medidas. Por este motivo, se considera que la hipótesis es válida.

Es importante comentar sobre la linealidad, que se ha conseguido determinar la relación entre la distancia recorrida real y la calculada, de tal forma que, un centímetro real equivale a 750 unidades de distancia tras el procesado de datos.

Por último, sobre el desarrollo del sistema, la configuración de parámetros que ha resultado ser la seleccionada es la que tiene una frecuencia de muestreo de 400 Hz y NUM_READS_AVG de 16 muestras. Esto se debe a que, aunque las dos combinaciones propuestas tienen un comportamiento similar para velocidades rápidas, esta combinación de parámetros presenta un comportamiento mejor para movimientos con velocidades medias.

4.3.2. Resultados del análisis de datos

Por su parte, los resultados obtenidos del análisis de datos se pueden agrupar en cuatro grupos:

- En función de la superficie.
- En función de la velocidad.
- En función de la distancia.
- En función del modo de funcionamiento.

En primer lugar, en cuanto a los resultados en función de la superficie, ha quedado patente que el dispositivo debe operar sobre un plano horizontal, por lo que sólo serán de interés los ejes X e Y. Esto se debe a que, además de para simplificar el trabajo, de no hacerse, se introduce en los componentes X e Y del vector aceleración la proyección de la gravedad terrestre sobre estos ejes, lo que provoca que el sistema se vuelva inestable.

Sobre los resultados de las muestras en función de la velocidad del movimiento, cuanto más rápido es el movimiento, mejor es el resultado. Los movimientos lentos no serán válidos ya que impiden el cumplimiento de la hipótesis. Esto se debe a que, al realizar movimientos rápidos, la señal de aceleración compuesta por información y ruido, tiene valores de información elevados con respecto al ruido, por lo que el ruido tiene un comportamiento residual. Sin embargo, en movimientos lentos, ocurre lo contrario, la información de la señal es mucho más baja y queda enmascarada por el ruido de la señal, lo que provoca que, al realizar el proceso iterativo en el Script de Matlab, el sistema se vuelva inestable.

Para los resultados en función de la distancia, se ha comprobado que la calidad de los resultados no depende de la distancia recorrida, al menos con movimientos entre los 10 y los 40 cm. Si hay que mencionar que, para distancia muy pequeñas, de unos pocos centímetros, el sistema no funciona, ya que no se aprecia linealidad.

Por último, se ha realizado un análisis para un desplazamiento de ida y de vuelta, es decir, dos movimientos continuos. Los resultados obtenidos son positivos, lo que revela que el sistema podrá trabajar en modo de funcionamiento continuo y no tendrá que inicializarse o resetearse cada vez que se realice un movimiento.

4.4. Conclusiones

Una vez realizado el estudio preliminar, se constata que se dan los dos requisitos principales para poder validar la hipótesis:

- Repetibilidad.
- Linealidad.

Además, se ha comprobado que el sistema propuesto trabaja correctamente funcionando de forma continua, esto es, un movimiento tras otro sin parar. Los resultados indican que podrá trabajar de manera continua, al menos, un número limitado de movimientos antes de ser necesario un proceso de reseteo.

Por tanto, se concluye que la hipótesis es cierta, pero con algunas restricciones. A saber:

- Sólo se trabajará en planos horizontales, es decir, en el eje X y en el eje Y.
- Habrá que llevar a cabo un proceso de calibración previo al procesado de datos para poder eliminar el error de cero sistemático del dispositivo.
- Se realizará un proceso de filtrado para suprimir errores aleatorios en la señal que puedan afectar en el procesado de datos.
- La velocidad de operación tendrá que ser normal o rápida, evitándose los movimientos lentos.
- El desplazamiento operativo recomendado estará entre los 10 centímetros y los 40 centímetros.

Otro aspecto importante es que se ha conseguido establecer la relación que existe entre el valor de distancia recorrida calculada y el valor de distancia recorrida real. Un centímetro recorrido real es proporcional a 750 unidades obtenidas de distancia calculada tras el proceso iterativo.

Además, se ha determinado que la configuración de parámetros más óptima para la realización del trabajo es la que tiene una frecuencia de muestreo de 400 Hz y NUM_READS_AVG de 16 muestras, ya que presenta el mejor comportamiento para movimientos con velocidad media y rápida.

En resumen, se ha comprobado la hipótesis y se ha obtenido información, tanto sobre las limitaciones del sistema, como sobre las configuraciones y mejoras necesarias para desarrollar el sistema. Todo esto es necesario para poder llevar a cabo el diseño hardware del sistema que se trata en el siguiente capítulo.

5. DESARROLLO HARDWARE DEL SISTEMA

5.1. Introducción

Después de comprobar que la hipótesis se cumple y haber acotado las restricciones de funcionamiento del sistema, se puede continuar con el desarrollo del trabajo.

Hasta ahora se ha conseguido una primera versión hardware de la primera etapa “Adquisición de datos” y una primera versión software de la segunda etapa “Procesado de datos”. Esta última es funcional mientras se cumplan las condiciones operativas y se lleven a cabo los procesos de mejora (calibración y filtros).

El siguiente paso del trabajo es realizar el diseño hardware final de ambas etapas. Para ello, se modificará la primera para que envíe los datos a la segunda etapa y no al ordenador mediante el cable USB.

Así mismo habrá que pasar la segunda etapa de software a hardware e implementar el proceso de calibración y los filtros.

Además, se realiza el diseño hardware de la tercera etapa “Presentación de los resultados” para poder visualizar en la placa los resultados obtenidos tras el procesado de la segunda etapa (Figura 65).

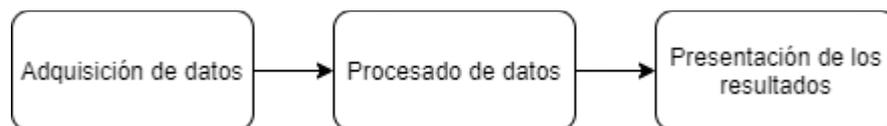


Figura 65: Procesos principales que realiza el diseño hardware.
Fuente: Elaboración propia.

Una vez se finalizan los diseños hardware de las tres etapas, se combinan para obtener el diseño de la primera versión del prototipo. Esta primera versión tiene la capacidad de trabajar en un eje.

Para que el prototipo pueda trabajar en dos ejes simultáneamente se llevan a cabo unas modificaciones en la primera versión del prototipo, dando lugar a la versión final del prototipo.

5.2. Versión final hardware de la primera etapa “Adquisición de datos”

5.2.1. Funcionamiento del diseño

La primera versión hardware de la primera etapa permite seleccionar una de las cuatro posibles señales captadas por el sensor (aceleración en el eje X, en el eje Y, en el eje Z o temperatura) y enviarla al ordenador utilizando un cable USB para poder analizarla.

Ahora, esta versión final hardware de la primera etapa “Adquisición de datos” toma las señales de aceleración en el eje X y en el eje Y y las pone a su salida, para que la versión final hardware de la segunda etapa pueda leerlas. Para ello, cada señal de aceleración debe tener activado un interruptor asociado, por lo que esta versión permite trabajar sólo con el eje X, sólo con el eje Y o con los dos a la vez, en función de los interruptores.

La versión final hardware de la primera etapa se compone de dos bloques funcionales:

- Sensor.
- Procesado.

El primer bloque “Sensor” es prácticamente igual al de la primera versión, es decir, contrala el acelerómetro ADXL362 y permite obtener a su salida las señales de la aceleración en el eje X, eje Y y eje Z, así como la señal de temperatura. A diferencia de la primera versión, la versión final tiene una señal de inicio del sensor que debe estar en alto para que el sensor funcione.

El segundo bloque “Procesado” si es radicalmente distinto del de la primera versión, ya que no hay que enviar los datos al ordenador. Ahora, toma las señales de aceleración en el eje X y en el eje Y (las señales de temperatura y de aceleración en el eje Z no se emplean) y las pone a su salida, si sus respectivos interruptores están activados.

5.2.2. Descripción de los componentes del diseño

En este subapartado se describen los componentes empleados en el diseño de la versión final hardware de la primera etapa “Adquisición de datos”. Para facilitar la comprensión del lector se organiza la información en tablas, como se venía haciendo.

En el Anexo 7 se puede consultar el esquema correspondiente a la versión final hardware de la primera etapa “Adquisición de datos”.

Módulo principal

El módulo principal se llama “adquisicion.vhd” (Tabla 16). Contiene los dos bloques funcionales: “Sensor” con su módulo “adquisicion_sensor.vhd” y “Procesado” con su módulo “adquisicion_procesado.vhd”.

adquisicion.vhd

El módulo “adquisicion.vhd” se encarga del proceso de adquisición de datos del acelerómetro y permite seleccionar con que eje se trabaja.

SEÑAL	BITS	I/O	DESCRIPCIÓN
clk	1	input	Señal de reloj.
reset	1	input	Señal de inicialización asíncrono.
start_sensor	1	input	Señal de inicio para muestrear.
selx	1	input	Señal selectora para eje X.
sely	1	input	Señal selectora para eje Y.
miso	1	input	Señal MISO del protocolo SPI.
sclk	1	output	Señal de reloj del protocolo SPI.
mosi	1	output	Señal MOSI del protocolo SPI.
ss	1	output	Señal de habilitación maestro/esclavo del protocolo SPI.
done	1	output	Señal de finalización.
x	12	output	Dato de aceleración en el eje X en Ca2.
y	12	output	Dato de aceleración en el eje Y en Ca2.

Tabla 16: Descripción del módulo “adquisicion.vhd”.
Fuente: Elaboración propia.

Bloque "Sensor"

El módulo del bloque "Sensor" se llama "adquisicion_sensor.vhd". Este módulo es igual al módulo "sensor.vhd" de la primera versión (Tabla 3), salvo que tiene una señal de entrada más de tipo STD_LOGIC, es decir, de un bit, llamada "start_sensor". Esta señal cuando está activada permite que se muestren a la salida las señales de aceleración y temperatura.

Además, en este bloque "Sensor" de la versión final hardware de la primera etapa, al igual que ocurría en el bloque homónimo de la primera versión, hay instanciados los mismos dos componentes para controlar el acelerómetro y poder comunicarse con él. Son los módulos "ADXL362Ctrl.vhd" (Tabla 4) y "SPI_If.vhd" (Tabla 5).

Bloque "Procesado"

El bloque funcional "Procesado" sólo tiene un módulo llamado "adquisicion_procesado.vhd" (Tabla 17).

adquisicion_procesado.vhd

El módulo "adquisicion_procesado.vhd" permite elegir con que ejes se va a trabajar (eje X, eje Y o ambos), en función de sus interruptores asociados.

SEÑAL	BITS	I/O	DESCRIPCIÓN
clk	1	input	Señal de reloj.
reset	1	input	Señal de inicialización asíncrono.
start	1	input	Señal de inicio.
x_in	12	input	Señal de entrada de aceleración en el eje X en Ca2.
y_in	12	input	Señal de entrada de aceleración en el eje Y en Ca2.
selx	1	input	Señal selectora para trabajar en el eje X.

sely	1	input	Señal selectora para trabajar en el eje Y.
done	1	output	Señal de finalización.
x_out	12	output	Señal de aceleración en el eje X en Ca2.
y_out	12	output	Señal de aceleración en el eje Y en Ca2.

Tabla 17: Descripción del módulo “adquisicion_procesado.vhd”.
Fuente: Elaboración propia.

5.3. Versión final hardware de la segunda etapa “Procesado de datos”

5.3.1. Funcionamiento del diseño

A continuación, se desarrolla la versión final hardware de la segunda etapa “Procesado de datos”. En esta versión se ha realizado la transición del proceso de integración desde software a hardware. Además, se implementan en hardware los filtros y el proceso de calibración.

Esta segunda etapa recibe en su entrada la aceleración de uno de los ejes captada por el sensor, en binario codificado en complemento a dos, y debe entregar a su salida la distancia recorrida escalada a centímetros, también en binario codificado en complemento a dos.

La versión final hardware de la segunda etapa “Procesado de datos” (Figura 66) se divide en dos bloques principales. El primero realiza el proceso de calibración y el segundo se encarga de obtener el desplazamiento. A su vez, este segundo bloque está formado por tres bloques en serie. Uno para los filtros, otro para el proceso de integración y el último para el escalado a centímetros.

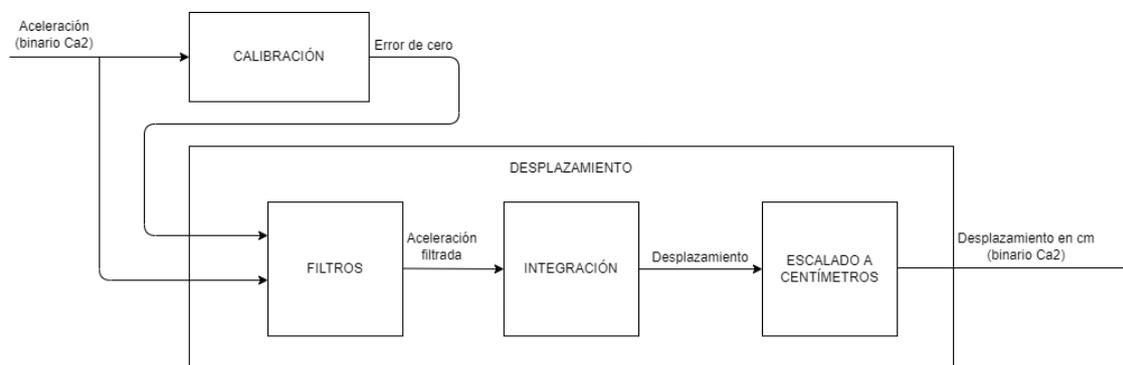


Figura 66: Esquema de la segunda etapa “Procesado de datos”.

Fuente: Elaboración propia.

En primer lugar, se debe seleccionar el modo de funcionamiento, esto es, elegir si vamos a calibrar el sistema o a obtener el desplazamiento. Para ello, cada modo lleva asociado un interruptor. Una vez se ha seleccionado un modo, se pulsa un botón para confirmar la elección, el botón de confirmación (botón de arriba).

Aunque lo más normal es empezar realizando un proceso de calibración, el sistema permite calcular el desplazamiento directamente, sin tener que hacer previamente el calibrado. De hacerse esto, no deben esperarse resultados positivos, por los motivos ya comentados en el estudio preliminar.

Ahora bien, el proceso de calibración consiste en calcular la media aritmética de una muestra de 1024 valores estando la placa parada. Este valor medio se corresponde con el error de cero, que será empleado posteriormente por los

filtros para mejorar la señal. Cuando se obtiene el error de cero, es decir, se termina el proceso de calibración, se enciende un LED. De esta forma el usuario sabe que la calibración ha terminado y se ha realizado.

Por otro lado, si el modo de funcionamiento elegido es el modo de obtención del desplazamiento, se llevan a cabo los procesos de filtrado, integración y escalado sucesivamente.

El proceso de filtrado debe anular el error de cero (error aleatorio) calculado en la calibración. Además, como los valores obtenidos con la placa parada varían ligeramente (error aleatorio), restando el error de cero no se consigue establecer un nivel de cero perfecto. Al tratarse de un proceso iterativo, estos pequeños valores resultantes van añadiendo error al procesado hasta causar problemas de medición inasumibles. Para solucionar este problema, se implementa un diseño que filtra un rango determinado de valores en torno al valor de error de cero calculado en el proceso de calibración. De esta forma, si la placa está quieta, los elementos muestreados que están dentro del rango son filtrados por el sistema que asume que su valor es cero, por lo que no afectan al procesado. Sin embargo, si la placa se mueve, los elementos muestreados se salen del rango de filtrado y su valor no se ve alterado. Al realizar este proceso de esta manera, si se termina de realizar un movimiento y la placa vuelve a estar parada, se pueden volver a filtrar los nuevos elementos muestreados, lo que permite que la placa pueda trabajar de forma continua.

A continuación, se realiza el proceso de integración. El diseño desarrollado a partir del script de Matlab lleva a cabo, de forma iterativa, el algoritmo que se corresponde con el proceso de integración basado en las ecuaciones del movimiento rectilíneo uniformemente acelerado. De esta forma, partiendo de la aceleración filtrada se obtiene a la salida el desplazamiento. Además, se implementan algunas modificaciones para poder trabajar de manera continua. Tras realizar un movimiento, si la placa se queda parada, se filtran los nuevos elementos de aceleración asignándoles un valor de cero, como se ha explicado anteriormente. No obstante, con la velocidad no ocurre lo mismo. Si el proceso de integración no anula el valor de velocidad al realizar la iteración del algoritmo, habrá un valor residual de velocidad que llevará al sistema a la inestabilidad. Para corregir este inconveniente, se implementa un procedimiento que estudia si la placa realmente se ha detenido para anular también la velocidad. El procedimiento consiste en ver si durante cuatro ciclos la aceleración es nula y la velocidad muy baja. Esto se corresponde en realidad con la placa parada, por lo que la velocidad se pone a cero.

Por último, hay que escalar a centímetros el desplazamiento obtenido tras la integración. Para ello, se emplea la relación obtenida en el estudio preliminar, es decir, 750 unidades de distancia calculada equivalen a 1 centímetro de distancia real. Este valor escalado en binario codificado en complemento a dos está disponible a la salida de la segunda etapa, junto con una señal de "done", para ser empleados por la tercera etapa.

5.3.2. Descripción de los componentes del diseño

En este subapartado se describen los componentes utilizados en la versión final hardware de la segunda etapa “Procesado de datos”. Para facilitar la comprensión del lector se emplea una tabla por componente.

En el Anexo 8 se puede consultar el esquema correspondiente a la versión final hardware de la segunda etapa “Procesado de datos”.

Módulo principal

El módulo principal se llama “procesado.vhd” (Tabla 18). Contiene dos bloques principales:

- Bloque “Calibración” con su módulo “calibrado.vhd”.
- Bloque “Desplazamiento” con su módulo “posicionado.vhd”.

procesado.vhd

El módulo “procesado.vhd” realiza el procesamiento de datos para pasar de una señal de aceleración a una de desplazamiento. Este módulo permite seleccionar el modo de funcionamiento entre calibración u obtención del desplazamiento.

SEÑAL	BITS	I/O	DESCRIPCIÓN
clk	1	input	Señal de reloj.
reset	1	input	Señal de inicialización asíncrono.
start_mod0	1	input	Señal de inicio para confirmar modo de funcionamiento seleccionado.
calibrar	1	input	Señal para seleccionar el modo de funcionamiento de calibración.
posicionar	1	input	Señal para seleccionar el modo de funcionamiento de obtención del desplazamiento.
start_new_num	1	input	Señal de inicio asociada a un nuevo valor de aceleración disponible.

bin	12	input	Señal de la aceleración en Ca2.
led	1	output	Señal del LED asociado al proceso de calibración.
done	1	output	Señal de finalización.
bin_out	12	output	Señal del desplazamiento en Ca2.

Tabla 18: Descripción del módulo “procesado.vhd”.
Fuente: Elaboración propia.

Bloque “Calibración”

El bloque “Calibración” sólo tiene un módulo llamado “calibrado.vhd” (Tabla 19).

calibrado.vhd			
El módulo “calibrado.vhd” realiza el proceso de calibración, permitiendo obtener el error de cero. Para ello calcula la media aritmética de 1024 valores muestreados estando la placa parada.			
SEÑAL	BITS	I/O	DESCRIPCIÓN
clk	1	input	Señal de reloj.
reset	1	input	Señal de inicialización asíncrono.
start_mod0	1	input	Señal de inicio para confirmar modo de funcionamiento seleccionado.
calibrar	1	input	Señal para seleccionar el modo de funcionamiento de calibración.
posicionar	1	input	Señal para seleccionar el modo de funcionamiento de obtención del desplazamiento.
start_new_num	1	input	Señal de inicio asociada a un nuevo valor de aceleración disponible.
bin	12	input	Señal de la aceleración en Ca2.
led	1	output	Señal del LED asociado al proceso de calibración.
done	1	output	Señal de finalización.
error	12	output	Señal del error de cero en Ca2.

Tabla 19: Descripción del módulo “calibrado.vhd”.
Fuente: Elaboración propia.

Bloque “Desplazamiento”

El módulo del bloque “Desplazamiento” se llama “posicionado.vhd” (Tabla 20). Este módulo tiene instanciados los tres componentes que permiten realizar los procesos de filtrado, integración y escalado. Los nombres de estos componentes son “filtro.vhd” (Tabla 21), “integracion.vhd” (Tabla 22) y “escalado.vhd” (Tabla 23).

posicionado.vhd			
El módulo “posicionado.vhd” realiza el proceso los procesos de filtrado, integración y escalado.			
SEÑAL	BITS	I/O	DESCRIPCIÓN
clk	1	input	Señal de reloj.
reset	1	input	Señal de inicialización asíncrono.
start_mod0	1	input	Señal de inicio para confirmar modo de funcionamiento seleccionado.
calibrar	1	input	Señal para seleccionar el modo de funcionamiento de calibración.
posicionar	1	input	Señal para seleccionar el modo de funcionamiento de obtención del desplazamiento.
start_new_num	1	input	Señal de inicio asociada a un nuevo valor de aceleración disponible.
bin	12	input	Señal de la aceleración en Ca2.
error	12	input	Señal del error de cero en Ca2.
done	1	output	Señal de finalización.
bin_out	12	output	Señal de la aceleración filtrada en Ca2.

Tabla 20: Descripción del módulo “posicionado.vhd”.
Fuente: Elaboración propia.

filtro.vhd

El módulo “filtro.vhd” realiza el proceso de filtrado.

SEÑAL	BITS	I/O	DESCRIPCIÓN
clk	1	input	Señal de reloj.
reset	1	input	Señal de inicialización asíncrono.
start	1	input	Señal de inicio.
start_new_num	1	input	Señal de inicio asociada a un nuevo valor de aceleración disponible.
bin	12	input	Señal de la aceleración en Ca2.
error	12	input	Señal del error de cero en Ca2.
done	1	output	Señal de finalización.
bin_out	12	output	Señal de la aceleración filtrada en Ca2.

Tabla 21: Descripción del módulo “filtro.vhd”.
Fuente: Elaboración propia.

integracion.vhd

El módulo “integracion.vhd” realiza el proceso de integración.

SEÑAL	BITS	I/O	DESCRIPCIÓN
clk	1	input	Señal de reloj.
reset	1	input	Señal de inicialización asíncrono.

start	1	input	Señal de inicio.
acel	12	input	Señal de la aceleración filtrada en Ca2.
done	1	output	Señal de finalización.
pos	20	output	Señal del desplazamiento en Ca2.

Tabla 22: Descripción del módulo “integracion.vhd”.
Fuente: Elaboración propia.

escalado.vhd

El módulo “escalado.vhd” realiza el proceso de escalado a centímetros.

SEÑAL	BITS	I/O	DESCRIPCIÓN
clk	1	input	Señal de reloj.
reset	1	input	Señal de inicialización asíncrono.
start	1	input	Señal de inicio.
bin	20	input	Señal del desplazamiento en Ca2.
done	1	output	Señal de finalización.
bin_out	12	output	Señal del desplazamiento escalado en centímetros en Ca2.

Tabla 23: Descripción del módulo “escalado.vhd”.
Fuente: Elaboración propia.

5.4. Versión final hardware de la tercera etapa “Presentación de los resultados”

5.4.1. Funcionamiento del diseño

Una vez se obtiene el desplazamiento, el siguiente paso es mostrarlo para que el usuario del sistema pueda visualizarlo. Para conseguirlo se desarrolla la versión final hardware de la tercera etapa “Presentación de los resultados”.

En este diseño se ha optado por emplear el visualizar 7 segmentos de la placa Nexys4 DDR para mostrar los resultados, ya que es una alternativa muy eficaz y sencilla. El visualizador dispone de dos módulos de cuatro cifras cada uno. Se emplea un módulo para mostrar la información de cada eje y el dígito más significativo de cada módulo para representar el signo menos si fuese necesario.

Esta tercera etapa recibe el desplazamiento codificado en complemento a dos que ha sido calculado en el procesado de datos. Lo primero que hace es pasarlo a BCD en su bloque funcional “Binario Ca2 a BCD”. Para ello, realiza un estudio del signo y pasa el valor absoluto a BCD. De esta forma, este bloque da como resultado un conjunto de cuatro números BCD, tres para representar las cifras y uno para el signo menos o el espacio, según corresponda.

A continuación, en el segundo bloque “BCD a 7 Segmentos” se encarga de encender los segmentos correspondientes de los dígitos del visualizador 7 segmentos. Lo hace en función de los cuatro números BCD obtenidos en el primer bloque. Este segundo bloque ya está implementado para trabajar con un eje o con dos ejes (Figura 67).

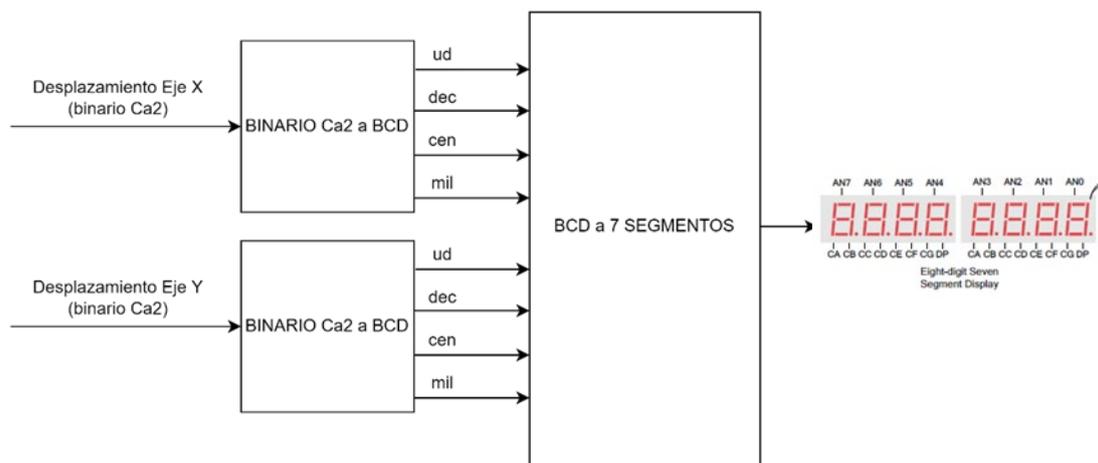


Figura 67: Esquema de la tercera etapa “Presentación de los resultados”.

Fuente: Elaboración propia.

5.4.2. Descripción de los componentes del diseño

En este subapartado se describen los componentes de la versión final hardware de la tercera etapa “Presentación de los resultados”. Como se venía haciendo hasta ahora, cada componente tiene su tabla.

En el Anexo 9 se puede consultar el esquema correspondiente a la versión final hardware de la tercera etapa “Presentación de los resultados”.

Módulo principal

El módulo principal se llama “presentacion.vhd” (Tabla 24). Tiene dos bloques principales:

- Bloque “Binario Ca2 a BCD” con su módulo “binCa2_to_4bcd.vhd”.
- Bloque “BCD a 7 Segmentos” con su módulo “visu7hex8.vhd”.

presentacion.vhd

El módulo “presentacion.vhd” permite mostrar el desplazamiento en el visualizador 7 segmentos. Este módulo permite trabajar con uno o dos ejes.

SEÑAL	BITS	I/O	DESCRIPCIÓN
clk	1	input	Señal de reloj.
reset	1	input	Señal de inicialización asíncrono.
start_x	1	input	Señal de inicio del eje X.
bin_x	12	input	Señal del desplazamiento escalado en centímetros en Ca2 del eje X.
start_y	1	input	Señal de inicio del eje Y.
bin_y	12	input	Señal del desplazamiento escalado en centímetros en Ca2 del eje Y.
an	8	output	Señal para seleccionar el dígito a encender.
ca	1	output	Señal asociada al segmento CA del dígito.

cb	1	output	Señal asociada al segmento CB del dígito.
cc	1	output	Señal asociada al segmento CC del dígito.
cd	1	output	Señal asociada al segmento CD del dígito.
ce	1	output	Señal asociada al segmento CE del dígito.
cf	1	output	Señal asociada al segmento CF del dígito.
cg	1	output	Señal asociada al segmento CG del dígito.

Tabla 24: Descripción del módulo “presentacion.vhd”.
Fuente: Elaboración propia.

Bloque “Binario Ca2 a BCD”

El módulo del bloque “Binario Ca2 a BCD” se llama “binCa2_to_4bcd.vhd” (Tabla 25).

binCa2_to_4bcd.vhd			
El módulo “binCa2_to_4bcd.vhd” pasa el número binario codificado en complemento a dos a cuatro dígitos BCD.			
SEÑAL	BITS	I/O	DESCRIPCIÓN
clk	1	input	Señal de reloj.
reset	1	input	Señal de inicialización asíncrono.
start	1	input	Señal de inicio.
bin	12	input	Señal del desplazamiento escalado en centímetros en Ca2.
done	1	output	Señal de finalización.
ud	4	output	Señal del número BCD a visualizar en el dígito de las unidades.
dec	4	output	Señal del número BCD a visualizar en el dígito de las decenas.
cen	4	output	Señal del número BCD a visualizar en el dígito de las centenas.
mil	4	output	Señal del número BCD a visualizar en el dígito de los millares.

Tabla 25: Descripción del módulo “binCa2_to_4bcd.vhd”.
Fuente: Elaboración propia.

Para poder pasar un número binario codificado en complemento a dos a un número BCD de cuatro cifras teniendo en cuenta su signo, el módulo “Binario Ca2 a BCD” instancia tres componentes: “absoluto.vhd” (Tabla 8), “signo2bcd” (Tabla 9) y “binary_to_bcd” (Tabla 26). De estos tres componentes, el primero y el segundo ya se han empleado en el desarrollo de la primera versión hardware de la primera etapa. Por este motivo, no se comentan en esta sección.

binary_to_bcd.vhd

El módulo “binary_to_bcd.vhd” pasa un número binario positivo a BCD.

SEÑAL	BITS	I/O	DESCRIPCIÓN
clk	1	input	Señal de reloj.
reset_n	1	input	Señal de inicialización asíncrono activo en baja.
ena	1	input	Señal de habilitación.
binary	12	input	Señal del desplazamiento escalado en centímetros positivo.
busy	1	output	Señal de ocupado.
bcd	12	output	Señal del número BCD.

Tabla 26: Descripción del módulo “binary_to_bcd.vhd”.
Fuente: Elaboración propia.

Bloque “BCD a 7 Segmentos”

El bloque “BCD a 7 Segmentos” sólo tiene un módulo llamado “visu7hex8.vhd” (Tabla 27).

visu7hex8.vhd

El módulo “visu7hex8.vhd” controla el visualizador 7 segmentos en función de números BCD.

Permite visualizar caracteres hexadecimales, por lo que se ha modificado para que muestre el signo menos y el espacio en lugar de las letras A y B.

SEÑAL	BITS	I/O	DESCRIPCIÓN
clk	1	input	Señal de reloj.
reset	1	input	Señal de inicialización asíncrono.
ud	4	input	Señal del número BCD a visualizar en el dígito de las unidades del módulo derecho.
dec	4	input	Señal del número BCD a visualizar en el dígito de las decenas del módulo derecho.
cen	4	input	Señal del número BCD a visualizar en el dígito de las centenas del módulo derecho.
mil	4	input	Señal del número BCD a visualizar en el dígito de los millares del módulo derecho.
ud2	4	input	Señal del número BCD a visualizar en el dígito de las unidades del módulo izquierdo.
dec2	4	input	Señal del número BCD a visualizar en el dígito de las decenas del módulo izquierdo.
cen2	4	input	Señal del número BCD a visualizar en el dígito de las centenas del módulo izquierdo.

mil2	4	input	Señal del número BCD a visualizar en el dígito de los millares del módulo izquierdo.
an	8	output	Señal para seleccionar el dígito a encender.
ca	1	output	Señal asociada al segmento CA del dígito.
cb	1	output	Señal asociada al segmento CB del dígito.
cc	1	output	Señal asociada al segmento CC del dígito.
cd	1	output	Señal asociada al segmento CD del dígito.
ce	1	output	Señal asociada al segmento CE del dígito.
cf	1	output	Señal asociada al segmento CF del dígito.
cg	1	output	Señal asociada al segmento CG del dígito.

Tabla 27: Descripción del módulo “visu7hex8.vhd”.
Fuente: Elaboración propia.

5.5. Primera versión del prototipo (1 eje)

5.5.1. Funcionamiento del diseño

Una vez se han desarrollado las versiones finales hardware de las tres etapas, éstas se combinan implementando la primera versión del prototipo. Esta primera versión permite comprobar si el sistema funciona para un eje, aunque hay varios módulos que ya aceptan el funcionamiento para dos ejes.

Para desarrollar esta primera versión del prototipo se instancian los componentes de las tres etapas en una entidad principal llamada "acel2pos.vhd". A continuación, se detalla su funcionamiento.

En primer lugar, se debe seleccionar el eje con el que se va a trabajar empleando la señal "selx" y "sely". Cuando sus interruptores asociados "sw0" y "sw1" están activos, se habilita la señal de inicio para muestrear "start_sensor" asociada al interruptor "sw2".

Una vez, iniciado el proceso de adquisición se debe llevar a cabo un proceso de calibración para que los resultados obtenidos sean óptimos. Para ello, se selecciona como modo de funcionamiento "Calibración" a través del interruptor "sw14". El interruptor del modo de funcionamiento "Desplazamiento" "sw15" debe está desactivado, de lo contrario el proceso no comienza. Una vez seleccionado de forma correcta el modo de funcionamiento, se pulsa el botón de confirmación "arriba". El proceso de calibración debe hacerse con la placa parada sobre la superficie plana sobre la que se va a trabajar. El proceso tarda aproximadamente 40 segundos y cuando finaliza el sistema enciende el LED 14 para avisar al usuario que la calibración ha terminado.

Si se pasa al modo de funcionamiento "Desplazamiento" con el interruptor "sw15" activado y el "sw14" desactivado, y se pulsa el botón de confirmación, el sistema comienza a mostrar en el visualizar 7 segmentos la cantidad de centímetros que se desplaza la placa al moverse. Si el desplazamiento es negativo, se muestra el signo menos.

Para detener el proceso y salir del modo de funcionamiento "Desplazamiento", volviendo al modo "Espera", basta con pulsar el botón de inicialización "centro".

Se ha comprobado que esta versión es funcional, por lo que se decide pasar a implementar la versión final del prototipo que emplea los dos ejes.

5.5.2. Descripción de los componentes del diseño

En este subapartado se comentan los componentes empleados en el diseño de la primera versión del prototipo.

Los diseños en VHDL de los componentes de esta primera versión del prototipo se pueden encontrar en la siguiente ruta, en el CD adjunto: Otros/Anexos/3 – Primera versión del prototipo (1 eje)/Diseño hardware. El

archivo que contiene el diseño de la primera versión del prototipo se llama “*acel2pos.xise*”.

En el Anexo 10 se puede consultar el esquema correspondiente a la primera versión del prototipo.

Módulo principal

El módulo principal del archivo “*acel2pos.xise*” lleva ese mismo nombre, “*acel2pos.vhd*” (Tabla 28). Contiene los tres bloques funcionales correspondientes a las tres etapas.

- Bloque “Adquisición de datos” con su módulo “*adquisicion.vhd*” (Tabla 16).
- Bloque “Procesado de datos” con su módulo “*procesado.vhd*” (Tabla 18).
- Bloque “Presentación de los resultados” con su módulo “*presentacion.vhd*” (Tabla 24).

acel2pos.vhd

El módulo “*acel2pos.vhd*” es el módulo principal de la primera versión del prototipo.

SEÑAL	BITS	I/O	DESCRIPCIÓN
clk	1	input	Señal de reloj.
reset	1	input	Señal de inicialización asíncrono.
start_sensor	1	input	Señal de inicio para muestrear.
selx	1	input	Señal selectora para eje X.
sely	1	input	Señal selectora para eje Y.
miso	1	input	Señal MISO del protocolo SPI.
start_mod0	1	input	Señal de inicio para confirmar modo de funcionamiento seleccionado.

calibrar	1	input	Señal para seleccionar el modo de funcionamiento de calibración.
posicionar	1	input	Señal para seleccionar el modo de funcionamiento de obtención del desplazamiento.
sclk	1	output	Señal de reloj del protocolo SPI.
mosi	1	output	Señal MOSI del protocolo SPI.
ss	1	output	Señal de habilitación maestro/esclavo del protocolo SPI.
led	1	output	Señal del LED asociado al proceso de calibración.
an	8	output	Señal para seleccionar el dígito a encender.
ca	1	output	Señal asociada al segmento CA del dígito.
cb	1	output	Señal asociada al segmento CB del dígito.
cc	1	output	Señal asociada al segmento CC del dígito.
cd	1	output	Señal asociada al segmento CD del dígito.
ce	1	output	Señal asociada al segmento CE del dígito.
cf	1	output	Señal asociada al segmento CF del dígito.
cg	1	output	Señal asociada al segmento CG del dígito.

Tabla 28: Descripción del módulo “`acel2pos.vhd`”.
Fuente: Elaboración propia.

5.6. Versión final del prototipo (2 ejes)

5.6.1. Funcionamiento del diseño

Tras comprobar que la primera versión del prototipo funciona correctamente para un eje, se modifica obteniendo la versión final del prototipo para que pueda trabajar con los dos ejes.

El funcionamiento es análogo al de la primera versión y la interfaz de usuario no cambia.

Se prueba que el sistema funciona correctamente.

5.6.2. Descripción de los componentes del diseño

Las modificaciones realizadas para desarrollar esta versión final del prototipo no implican cambiar ningún módulo, más allá de tener que instanciar algunos componentes dos veces en lugar de una (una para cada eje). Por este motivo, las tablas descriptivas de cada componente, al no cambiar, no se añaden en esta sección.

Los componentes que se han tenido que instanciar dos veces son:

- El módulo “procesado.vhd”, es decir, se instancian dos segundas etapas de “Procesado de datos”, una para cada eje.
- El módulo “binCa2_to_4bcd.vhd” para pasar los datos de desplazamiento de las segundas etapas de cada eje a los 8 números BCD a representar en el visualizador (cuatro para el eje X y cuatro para el eje Y).

Los diseños en VHDL de los componentes de esta versión final del prototipo se pueden encontrar en la siguiente ruta, en el CD adjunto: Otros/Anexos/4 – Versión final del prototipo (2 ejes)/Diseño hardware. El archivo que contiene el diseño de la primera versión del prototipo se llama “acel2pos_xy.xise”.

En el Anexo 11 se puede consultar el esquema correspondiente a la versión final del prototipo.

6. APLICACIONES

6.1. Introducción

Hasta ahora se ha logrado desarrollar un diseño hardware que permite obtener la distancia recorrida en los dos ejes a la vez realizando movimientos continuos. El sistema tiene algunas restricciones, ya comentadas en el Capítulo 4, pero ofrece un rango amplio de aplicaciones.

En el presente trabajo, dado que se busca un dispositivo que permita determinar la posición de objetos, se llevan a cabo aplicaciones en este sentido.

Para desarrollar estas aplicaciones se han implementado varios diseños hardware, uno por cada aplicación propuesta. Se ha partido del diseño hardware de la versión final del prototipo, por lo que las entradas y salidas anteriores del sistema con las que interactúa el usuario no varían, salvo el LED que indica que el proceso de calibración se ha realizado que ahora pasa a ser el LED RGB verde.

Para llevar a cabo las aplicaciones, se han realizado varias modificaciones en el diseño de la versión final del prototipo. Estas modificaciones consisten en añadir un nuevo módulo en la entidad principal y cambiar las señales de ésta última para que el sistema funcione correctamente. Al haber tres aplicaciones distintas se desarrollan tres módulos nuevos. De manera que se instancia un módulo nuevo distinto para cada aplicación.

En los siguientes apartados se describen las aplicaciones, sus diseños y los resultados obtenidos en cada una de ellas.

6.2. Primera aplicación

6.2.1. Funcionamiento del diseño

La primera de las aplicaciones consiste en que el dispositivo determine la posición en la que se halla la placa al desplazarse sobre una superficie con ocho zonas delimitadas (Figura 68). La placa indica en que zona se encuentra iluminando uno de sus LEDs, de tal forma que cada LED va asociado a una zona distinta.

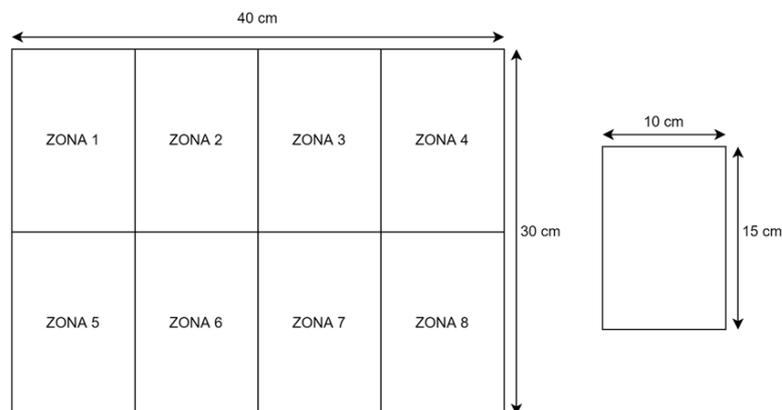


Figura 68: Superficie de la primera aplicación.
Fuente: Elaboración propia.

Debe conocerse la posición inicial para poder realizar los cálculos. Por este motivo, la placa debe comenzar en la zona 1.

Del mismo modo que en la versión final del prototipo, el sistema sigue permitiendo visualizar la distancia desplazada en los dos ejes a medida que se mueve.

6.2.2. Descripción de los componentes del diseño

El nuevo módulo instanciado en la entidad principal encargado de implementar la primera aplicación se llama “demo01.vhd” (Tabla 29). Este módulo está compuesto por dos componentes “demo01_zona.vhd” (Tabla 30) y “demo01_led.vhd” (Tabla 31).

Los diseños en VHDL de esta primera aplicación están disponibles en la siguiente ruta, en el CD adjunto: Otros/Anexos/5 – Primera aplicación/Diseño hardware. El archivo que contiene el diseño de la primera aplicación se llama “acel2pos_xy_demo01.xise”.

demo01.vhd

El módulo “demo01.vhd” implementa la primera aplicación. Enciende el LED asociado a la zona en la que se encuentra la placa después de calcular dicha zona, a partir de las distancias recorridas en los dos ejes y conociendo la posición inicial del sistema.

SEÑAL	BITS	I/O	DESCRIPCIÓN
clk	1	input	Señal de reloj.
reset	1	input	Señal de inicialización asíncrono.
start_x	1	input	Señal de inicio del eje X.
pos_x	12	input	Señal del desplazamiento escalado en centímetros en Ca2 del eje X.
start_y	1	input	Señal de inicio del eje Y.
pos_y	12	input	Señal del desplazamiento escalado en centímetros en Ca2 del eje Y.
led	8	output	Señal del LED a encender.

Tabla 29: Descripción del módulo “demo01.vhd”.
Fuente: Elaboración propia.

demo01_zona.vhd

El módulo “demo01_zona.vhd” calcula la zona en la que está la placa a partir de las distancias recorridas en los dos ejes y conociendo la posición inicial del sistema.

SEÑAL	BITS	I/O	DESCRIPCIÓN
clk	1	input	Señal de reloj.
reset	1	input	Señal de inicialización asíncrono.
start_x	1	input	Señal de inicio del eje X.
pos_x	12	input	Señal del desplazamiento escalado en centímetros en Ca2 del eje X.
start_y	1	input	Señal de inicio del eje Y.
pos_y	12	input	Señal del desplazamiento escalado en centímetros en Ca2 del eje Y.
done	1	output	Señal de finalización.
zona	4	output	Señal con la zona calculada.

Tabla 30: Descripción del módulo “demo01_zona.vhd”.
Fuente: Elaboración propia.

demo01_led.vhd

El módulo “demo01_led.vhd” enciende el LED asociado a la zona en la que se encuentra la placa.

SEÑAL	BITS	I/O	DESCRIPCIÓN
clk	1	input	Señal de reloj.
reset	1	input	Señal de inicialización asíncrono.
start	1	input	Señal de inicio.
zona	4	input	Señal con la zona calculada.
done	1	output	Señal de finalización.
led	8	output	Señal del LED a encender.

Tabla 31: Descripción del módulo “demo01_led.vhd”.
Fuente: Elaboración propia.

6.2.3. Resultados

Una vez implementado el diseño de la primera aplicación, se prueba y los primeros resultados son buenos, aunque hay que mencionar que los resultados, para ser satisfactorios, necesitan que el usuario adquiera cierta práctica para realizar los movimientos a la velocidad correcta y sin levantar la placa. Para facilitar este proceso se realizan movimientos en línea recta.

Para evaluar con más detalle el comportamiento del sistema en esta primera aplicación, se prueba a realizar un movimiento bordeando la superficie total, zona a zona, en sentido horario. Esto se realiza diez veces.

Los resultados de esta prueba son muy positivos, consiguiéndose que la placa acierte siempre las zonas en la que encuentra durante nueve de los diez intentos. La vez que no se ha conseguido se debe a que el sistema, en mitad de uno de los movimientos, se ha inestabilizado al no trabajar en el plano horizontal durante el movimiento.

Se decide aumentar la dificultad con una segunda aplicación.

6.3. Segunda aplicación

6.3.1. Funcionamiento del diseño

Dado que en la primera aplicación el sistema se comporta de forma muy positiva, se aumenta la dificultad incrementando el número de zonas en la misma superficie (Figura 69). Ahora hay un total de 15 zonas.

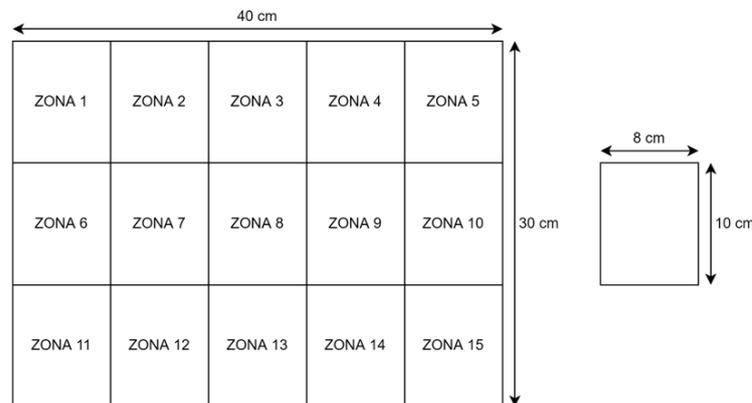


Figura 69: Superficie de la segunda aplicación.
Fuente: Elaboración propia.

En esta segunda aplicación, la posición inicial de la placa debe ser la zona 8.

Por lo demás, el procedimiento utilizado es análogo al de la primera aplicación.

6.3.2. Descripción de los componentes del diseño

En esta segunda aplicación, los nuevos módulos instanciados son prácticamente iguales a los de la primera aplicación. Por tanto, sólo se mencionarán las diferencias.

En primer lugar, la señal de los LEDs a encender tiene 15 bits y no 8, como en la primera aplicación.

Por otro lado, se realizan modificaciones leves en el diseño "demo01_zona.vhd" para que pueda calcular la zona con las nuevas dimensiones de la misma.

Los diseños en VHDL de esta segunda aplicación están disponibles en la siguiente ruta, en el CD adjunto: Otros/Anexos/6 – Segunda aplicación/Diseño hardware. El archivo que contiene el diseño de la segunda aplicación se llama "acel2pos_xy_demo02.xise".

6.3.3. Resultados

A pesar de haber aumentado la dificultad, los resultados son también bastante buenos.

En esta segunda aplicación se ha realizado otra prueba. Se ha realizado un movimiento en forma de infinito (∞), realizando la siguiente secuencia de desplazamiento empleando líneas rectas: zona 8, zona 4, zona 10, zona 14, zona 2, zona 6, zona 12 y vuelta a la zona 8. Este ciclo se ha repetido diez veces.

Nuevamente, la placa consigue mostrar de forma correcta todas las zonas de la secuencia un gran número de veces. Concretamente, lo consigue siete veces. En esta aplicación es más difícil completar un ciclo de forma perfecta, porque el sistema se vuelve inestable con mayor facilidad. Esto se debe a dos factores:

- En la primera aplicación se describe la forma de un rectángulo realizando cuatro movimientos. Ahora, en esta segunda aplicación para describir la forma del infinito hacen falta siete movimientos, lo que incrementa las posibilidades de errores (mala medición por parte del usuario ya que el movimiento es más complejo, más error acumulado en el proceso iterativo...).
- Por otro lado, en la primera aplicación los movimientos realizados son horizontales y verticales, y, por tanto, más fáciles de realizar para el usuario. En esta segunda aplicación, todos los movimientos son diagonales, aumentando la dificultad para el usuario.

Se ha probado movimientos horizontales y verticales en esta segunda aplicación y los resultados obtenidos mejoran notablemente. Por tanto, se puede concluir que una fuente muy importante en estas aplicaciones es el error humano, cometido por el usuario.

6.4. Tercera aplicación

6.4.1. Funcionamiento del diseño

Finalmente, tras los buenos resultados obtenidos hasta ahora, se decide implementar una versión inalámbrica del sistema.

En esta aplicación se busca emular el comportamiento de un sistema de medida inalámbrico en una industria, donde se evita la presencia de un operario in situ o la necesidad de emplear cableado para la transmisión de la información.

La aplicación propuesta transmite la información de forma inalámbrica empleando el módulo HC-12 (Figura 70). La placa emisora envía la zona en la que se encuentra a otra placa igual que actúa como receptora. Las transmisiones entre los módulos HC-12 y las placas Nexys4 DDR utilizan el protocolo UART.

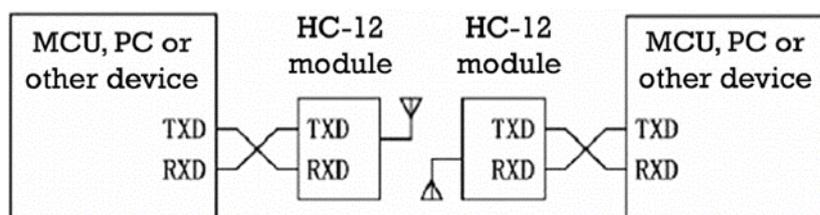


Figura 70: Esquema del sistema empleando el módulo HC-12 para transmitir de forma inalámbrica.
Fuente: [11] HC-12 Wireless Serial Port Communication Module User Manual.

El funcionamiento de la placa transmisora es el mismo que en la segunda aplicación, con la única diferencia que ahora envía a través del módulo HC-12 la zona en la que se encuentra.

Por su parte, la placa receptora si tiene un funcionamiento nuevo.

La placa receptora se mantiene a la espera de datos por parte de la transmisora. Cuando se produce transmisión, se avisa al usuario encendiendo el LED RGB azul.

Al recibir la zona en la que se encuentra la placa transmisora, la receptora muestra esta zona en los dos dígitos menos significativos del visualizador 7 segmentos. Además, en los LEDs se codifica en BCD la zona.

Cuando la placa transmisora pasa al modo “Espera”, esta deja de transmitir. En ese momento, en la placa receptora se apaga el LED RGB azul. Los números en el visualizar correspondientes a la última zona recibida se mantienen, es decir, no se elimina del visualizador esa información cuando deja de recibir datos. En caso de querer resetear el visualizador a cero, puede pulsarse el botón de inicialización “centro”.

6.4.2. Descripción de los componentes del diseño de la placa transmisora

En este diseño de la placa transmisora de la tercera aplicación, los nuevos módulos instanciados son los mismos que en la segunda aplicación. A mayores se instancia una UART (Tabla 13) en el nuevo módulo que lleva a cabo las aplicaciones “demo01.vhd”. Además, se hacen los cambios pertinentes de añadir las señales nuevas en este módulo como en la entidad principal “acel2pos_xy.vhd”.

Los diseños en VHDL están disponibles en la siguiente ruta, en el CD adjunto: Otros/Anexos/7 – Tercera aplicación/Diseño hardware. El archivo que contiene el diseño de la placa transmisora de la tercera aplicación se llama “acel2pos_xy_demo04_tx.xise”.

6.4.3. Descripción de los componentes del diseño de la placa receptora

La entidad principal del diseño de la placa receptora instancia una serie de componentes que ya se han empleado en otros diseños, a saber:

- “uart.vhd” (Tabla 13) para recibir datos por el módulo HC-12.
- “binary_to_bcd.vhd” (Tabla 26) para pasar la información de la zona codificada en binario a BCD.
- “visu7hex8.vhd” (Tabla 27) para representar la zona codificada en BCD en el visualizador 7 segmentos.

Los diseños en VHDL están disponibles en la siguiente ruta, en el CD adjunto: Otros/Anexos/7 – Tercera aplicación/Diseño hardware. El archivo que contiene el diseño de la placa transmisora de la tercera aplicación se llama “acel2pos_xy_demo04_rx.xise”.

6.4.4. Resultados

En primer lugar, la transmisión inalámbrica de esta aplicación se realiza correctamente y no afecta a los resultados obtenidos.

En esta tercera aplicación, se repiten las pruebas de la primera y de la segunda. Los resultados obtenidos son similares. El correcto funcionamiento del sistema depende en gran medida de que el usuario realice los movimientos de forma correcta. Ahora bien, de hacerlo, se puede conseguir medir el desplazamiento de distancias muy cortas, de incluso unos pocos centímetros.

Por este motivo, se concluye que el sistema de medida de distancia y detección de posición se ha realizado con éxito, cumpliendo con éxito las tareas de las tres aplicaciones propuestas.

7. CONCLUSIONES

El presente trabajo partía con el objetivo de estudiar y desarrollar un sistema que fuese capaz de medir el desplazamiento y determinar la posición de un objeto a partir de su aceleración y siendo previamente conocida la posición inicial. A la vista de los resultados finales, se puede concluir que el objetivo del trabajo se ha alcanzado con creces.

Primeramente, ha habido que validar la hipótesis realizando el estudio preliminar. En él, después de realizar las primeras versiones de las dos primeras etapas y analizar los datos, se determinó que los dos requisitos para poder aceptar la hipótesis como cierta, aunque con restricciones, se dan:

- Repetibilidad.
- Linealidad. Además, se ha determinado la relación existente entre la distancia real recorrida y la obtenida en valores binarios por el procesado (1 centímetro real equivale a 750 en valor binario).

Por otro lado, se comprobó que el sistema propuesto puede trabajar funcionando de manera continua al menos un número de veces limitado, es decir, un movimiento tras otro sin ser necesario un reseteo entre ambos.

Algunas de las restricciones para el cumplimiento de la hipótesis implican la implementación forzosa de ciertas mejoras en las versiones posteriores del sistema:

- Proceso de calibración para eliminar el error de cero sistemático de la señal de aceleración.
- Proceso de filtrado de errores aleatorios que afectan en el procesado de datos, siendo necesario filtrar tanto la aceleración como la velocidad.

Ahora bien, para que se cumpla la hipótesis hay que respetar, además de las restricciones que implican mejoras en el diseño del sistema, algunas condiciones de funcionamiento obtenidas en el estudio preliminar:

- Sólo se trabajará en planos horizontales, es decir, en el eje X y en el eje Y, sin inclinar la placa.
- La velocidad de operación tendrá que ser normal o rápida, evitándose los movimientos lentos.
- El desplazamiento será de 10 a 40 centímetros aproximadamente, evitándose los desplazamientos excesivamente cortos.

Finalmente, del estudio preliminar se extrae que la mejor configuración de parámetros para la realización del prototipo es la que tiene una frecuencia de muestreo de 400 Hz y NUM_READS_AVG de 16 muestras, ya que presenta un buen comportamiento tanto para movimientos a velocidad media como a velocidad rápida.

A continuación, se realizó el desarrollo hardware de las tres etapas, dando lugar a las versiones primera y final del prototipo, ambas funcionales.

A pesar de las limitaciones, el sistema tiene múltiples aplicaciones posibles. Dado que el trabajo busca desarrollar un dispositivo capaz de detectar la posición, se sigue esta línea, implementando y probando tres aplicaciones.

Los resultados de las pruebas de las aplicaciones ponen de manifiesto algunas conclusiones:

- El sistema propuesto es funcional.
- El error humano es muy importante a la hora de realizar mediciones con el sistema, de forma que, si el operario realiza movimientos prohibidos por las limitaciones del trabajo, el sistema se vuelve inestable.
- El sistema obtiene mejores resultados para movimientos simples, más cortos, en línea recta y en vertical u horizontal (y no en diagonal). Esto se debe a que es más sencillo para el usuario realizarlos correctamente.
- Aunque el sistema puede trabajar de forma continua, al hacerlo, las probabilidades de que el procesado se vuelva inestable aumentan. Esto es provocado por múltiples factores (mayor dificultad para el usuario, acumulación del error durante el procesado que termine por provocar su inestabilidad...).

En definitiva, el sistema desarrollado es funcional, con resultados muy positivos, pero han de cumplirse las restricciones para alcanzarlos.

A modo de cierre, comentar que alguna posible línea de continuación a este trabajo podría tratar la mejora del funcionamiento, eliminando algunas de estas restricciones, como, por ejemplo, la obligación de trabajar en un plano horizontal. La nueva versión del prototipo podría trabajar en espacios tridimensionales.

8. BIBLIOGRAFÍA

- [1] Analog Devices, Inc. (2012). Micropower Three-Axis $\pm 2g/\pm 4g/\pm 8g$ Digital-Output MEMS Accelerometer [Datasheet].
- [2] Ashenden, P. J. (2008). The Designer's Guide to VHDL [Libro].
- [3] Dhaker, P. (2018). Introduction to SPI Interface [Artículo]. Recuperado de <https://www.analog.com/en/analog-dialogue/articles/introduction-to-spi-interface.html> .
- [4] Digilent, Inc. (2016). Nexys4 DDR FPGA Board Reference Manual [Datasheet].
- [5] Ebel, F.; Nestel, S. (2003). FESTO Textbook FP 1110 Proximity Sensors [Libro].
- [6] G. Webster, John; Eren, H. (2014). Measurement, Instrumentation, and Sensors Handbook [Libro].
- [7] MathWorks, Inc. (2022). MATLAB [Artículo]. Recuperado de <https://es.mathworks.com/products/matlab.html> .
- [8] Miron, R. (2020). Conceptos fundamentales de los FPGA: ¿Qué son los FPGA y por qué son necesarios? [Artículo]. Recuperado de Número 782 (Enero 2020) de la Revista Española de Electrónica.
- [9] Peña, E.; Legaspi, M. G. (2020). UART: A Hardware Communication Protocol Understanding Universal Asynchronous Receiver/Transmitter [Artículo]. Recuperado de <https://www.analog.com/en/analog-dialogue/articles/uart-a-hardware-communication-protocol.html> .
- [10] Pini, A. (2021). Los fundamentos de los sensores de proximidad: Su selección y uso en la automatización industrial [Artículo]. Recuperado de <https://www.digikey.es/es/articles/the-fundamentals-of-proximity-sensors-selection-and-use-industrial-automation> .
- [11] Rozee, R. (2016). HC-12 Wireless Serial Port Communication Module User Manual [Datasheet].
- [12] Trimberger, S. M. (2015). Three Ages of FPGAs: A Retrospective on the First Thirty Years of FPGA Technology [Artículo]. Recuperado de <https://ieeexplore.ieee.org/document/7086413>.
- [13] Xilinx, Inc. (2022). ISE Design Suite [Artículo]. Recuperado de <https://www.xilinx.com/products/design-tools/ise-design-suite.html> .
- [14] Xilinx, Inc. (2022). ISE WebPACK Design Software [Artículo]. Recuperado de <https://www.xilinx.com/products/design-tools/ise-design-suite/ise-webpack.html> .

[15] Xilinx, Inc. (2022). What is an FPGA? Field Programmable Gate Array [Artículo]. Recuperado de <https://www.xilinx.com/products/silicon-devices/fpga/what-is-an-fpga.html> .

[16] Young, H. D.; Freedman, R. A. (2009). SEARS ZEMANSKY Física Universitaria [Libro].