

Trabajo Fin de Máster
Máster en Diseño Avanzado en Ingeniería
Mecánica

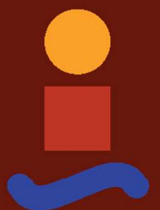
Desarrollo de Sistemas Experimentales para la
Medición de la Geometría de Vías

Autor: Miguel Rodríguez Gómez

Tutores: José Luis Escalona Franco y Pedro Urda Gómez

Dpto. Ingeniería Mecánica y Fabricación
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2023



Trabajo Fin de Máster
Máster en Diseño Avanzado en Ingeniería Mecánica

Desarrollo de Sistemas Experimentales para la Medición de la Geometría de Vías

Autor:

Miguel Rodríguez Gómez

Tutores:

José Luis Escalona Franco y Pedro Urda Gómez

Catedrático de Universidad y Profesor Ayudante Doctor

Dpto. de Ingeniería Mecánica y Fabricación

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2023

Trabajo Fin de Máster: Desarrollo de Sistemas Experimentales para la Medición de la Geometría de Vías

Autor: Miguel Rodríguez Gómez

Tutores: José Luis Escalona Franco y Pedro Urda Gómez

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2023

El Secretario del Tribunal

Resumen

En la industria ferroviaria resulta indispensable tener sistemas eficientes capaces de medir la geometría de la vía para detectar sus irregularidades y mantenerlas en buen estado. En este sentido, el Grupo de Ingeniería Mecánica ha estado trabajando en el desarrollo de sistemas embarcados para la medición de las irregularidades de las vías, contexto que engloba a este trabajo.

En primer lugar, se describen los trabajos previos que existen en la literatura y la evolución en los métodos utilizados para la auscultación geométrica de vías. Además, se resumen los aspectos principales que se han desarrollado en trabajos anteriores para poder establecer el contexto de este documento y definir los objetivos que se quieren alcanzar.

En segundo lugar, se describe el sistema para la medición de las irregularidades de larga longitud de onda incluyendo los sensores y equipos de adquisición de datos utilizados, los algoritmos de visión artificial aplicados, su implementación en una aplicación de escritorio y los resultados obtenidos a la hora de validar dichos algoritmos.

En tercer lugar, se describe el sistema diseñado para la medición de las irregularidades de corta longitud de onda, cómo se realiza el cálculo de dichas irregularidades, la implementación computacional y se exponen los resultados obtenidos.

Finalmente, se discuten las conclusiones del trabajo, realizando una comparación entre los dos sistemas indicando las ventajas e inconvenientes de cada uno.

Abstract

It is essential in the railway industry to have efficient systems capable of measuring the track's geometry in order to detect its irregularities and maintain them in good condition. In this respect, the Mechanical Engineering Group has been working on the development of embedded systems to measure track irregularities, context in which this work is included.

Firstly, previous works present in the literature are described as well as the evolution of the methods used for track geometry measurement. In addition, main issues that have been developed in previous works are resumed to establish this document's context and to define the objectives to achieve.

Secondly, the system for the measurement of long wavelength track irregularities is described, including the sensors and data acquisition equipment used, the applied computer vision algorithms, their implementation in a desktop application and the results obtained when validating such algorithms.

Thirdly, the designed system for the measurement of short wavelength track irregularities is described, how the calculation of the irregularities is done, its computational implementation and the obtained results are presented.

Finally, this work's conclusions are discussed by comparing both systems indicating their strengths and weaknesses.

Resumen	vii
Abstract	ix
Índice	xi
Índice de Figuras	xiii
1 Introducción	15
1.1. Contexto	16
1.2. Objetivos	18
2 Medición de Irregularidades de Larga Longitud de Onda	19
2.1. Descripción del Sistema	19
2.1.1 Sensores	19
2.1.1.1 Cámaras.....	19
2.1.1.2 Inertial Measuring Unit (IMU).....	20
2.1.1.3 Encóder rotatorio	21
2.1.1.4 Detector Magnético	22
2.1.2 Equipos de adquisición de datos	22
2.1.2.1 Compact RIO	22
2.1.2.2 Switch Gigabit	23
2.2. Visión Artificial	24
2.2.1 Calibración.....	24
2.2.1.1 Coordenadas homogéneas de un vector	24
2.2.1.2 Matriz de proyección.....	24
2.2.1.3 Procedimiento de calibración	25
2.2.2 Procesado de Imágenes	30
2.2.2.1 Filtrado de imagen.....	30
2.2.2.2 Cálculo de la posición del carril.....	31
2.3. Aplicación de escritorio	32
2.3.1 Recursos utilizados	32
2.3.2 Estructura del código.....	33
2.3.2.1 Arquitectura del programa	33
2.3.2.2 Clases implementadas.....	34
2.3.3 Manual de usuario.....	35
2.3.3.1 Conexión.....	36
2.3.3.2 Controles y configuración	37
2.3.3.3 Calibración de una cámara.....	38
2.3.3.4 Adquisición de datos y archivos generados.....	40
2.4. Validación y resultados experimentales	42
2.4.1 Mecanismo automático de ensayos	42
2.4.2 Resultados.....	44
3 Medición de Irregularidades de Corta Longitud de Onda: Corrugación	47
3.1. Diseño del sistema.....	47
3.2. Procesado de datos	51
3.3. Aplicación de escritorio	53

3.1.1	Recursos utilizados	53
3.1.2	Comunicación con Arduino	54
3.1.3	Clases implementadas	54
3.1.4	Manual de usuario	55
3.1.4.1	Conexión	56
3.1.4.2	Configuración y Adquisición de datos.....	57
3.1.4.3	Visualización y procesado de datos	59
3.4.	<i>Resultados</i>	62
4	Conclusiones	67
	Referencias	69

ÍNDICE DE FIGURAS

Figura 1. Definición de Irregularidades de Longitud de Onda Larga	15
Figura 2. Vista Plana del TGMS	16
Figura 3. Vía a Escala y Vehículo Instrumentado	18
Figura 4. Cámara Photonfocus	19
Figura 5. Formato de datos de las cámaras	20
Figura 6. Sensor Inercial (IMU)	20
Figura 7. Formato de Datos de la IMU	21
Figura 8. Encóder Rotatorio	21
Figura 9. Detector Magnético	22
Figura 10. Compact RIO	22
Figura 11. Switch Gigabit	23
Figura 12. Esquema Sistema Completo	23
Figura 13. Calibrador	25
Figura 14. Transformación de Imagen Completa a Líneas	30
Figura 15. Resultado del Filtrado de Imágenes	31
Figura 16. Cálculo de la Constante del Plano del Láser	31
Figura 17. Arquitectura de la Aplicación	33
Figura 18. Entorno Gráfico Inicial	35
Figura 19. Barra de Herramientas	36
Figura 20. Ventana Conexión con Cámaras	36
Figura 21. Controles	37
Figura 22. Visualización y Definición del Filtro	38
Figura 23. Ventana de Calibración	38
Figura 24. Selección de Puntos de Calibración	39
Figura 25. Orden Puntos Calibración	39
Figura 26. Resultados de la Calibración	40
Figura 27. Estado Antes de Adquisición de Datos	40
Figura 28. Adquisición de Datos y Visualización	41
Figura 29. Ventana de Representación de Señales	41
Figura 30. Mecanismo Automático de Ensayos	42
Figura 31. Modelo del Mecanismo	43
Figura 32. Comparación del Ancho de Vía	45

Figura 33. Comparación de la Nivelación	45
Figura 34. Perfilómetro Láser	47
Figura 35. Sistema de Referencia del Sensor	48
Figura 36. Switch para Conexión de Perfilómetros y PC	48
Figura 37. Alzado, Planta y Perfil de la Estructura	49
Figura 38. Vista Tridimensional de la Estructura	49
Figura 39. Pata Roscable	50
Figura 40. Montaje Encóder Rotatorio	50
Figura 41. Arduino UNO	50
Figura 42. Esquema Sistema para la Medición de la Corrugación	51
Figura 43. Imagen de la Vía a Escala	51
Figura 44. Definición de la Distancia de la Línea de Corrugación	52
Figura 45. Interfaz Gráfica de la Aplicación	56
Figura 46 Barra de Herramientas	56
Figura 47. Ventana de Selección de Perfilómetros Láser	56
Figura 48. Controles	57
Figura 49. Visualización de Imágenes	58
Figura 50. Ventana de Parámetros	58
Figura 51. Visualización de Imágenes Grabadas	60
Figura 52. Definición del Filtro	60
Figura 53. Definición de Parámetros para el Cálculo de la Corrugación	61
Figura 54. Criterio de Signos Distancia d	61
Figura 55. Representación de Resultados	62
Figura 56. Sistema Midiendo en la Vía a Escala	62
Figura 57. Montaje del Encóder	63
Figura 58. Trozo de Carril Corrugado	63
Figura 59. Carril Derecho con $d = 0$ mm	64
Figura 60. Carril Derecho con $d = 2$ mm	64
Figura 61. Carril Derecho con $d = -2$ mm	64
Figura 62. Carril Izquierdo con $d = 0$ mm	64
Figura 63. Carril Izquierdo con $d = 2$ mm	64
Figura 64. Carril Izquierdo con $d = -2$ mm	64

1 INTRODUCCIÓN

La seguridad y el confort de los pasajeros de cualquier vehículo ferroviario depende en gran medida del correcto mantenimiento del trazado ferroviario, pues la geometría de la vía determina la respuesta dinámica del vehículo que viaja sobre él [1, 2, 3]. Por ello, la industria ferroviaria muestra un gran interés en la búsqueda de nuevos sistemas para la auscultación geométrica de vías rápidos y eficientes para poder evaluar con mayor precisión el estado de la vía y determinar si se necesita reparar.

Las desviaciones de la geometría real de la vía respecto a la de diseño se denominan irregularidades y se clasifican según su longitud de onda en la normativa EN133848 [4]. Las irregularidades de onda corta, entre 10 y 1000 mm, corresponden al desgaste ondulatorio de los carriles, denominado corrugación [5]. Las irregularidades de longitud de onda larga se caracterizan por cuatro parámetros, dos de ellos relativos y los otros dos absolutos denominados ancho de vía, nivelación, alineamiento y peralte que se relacionan a través de los desplazamientos de los carriles de su posición ideal a través de las expresiones:

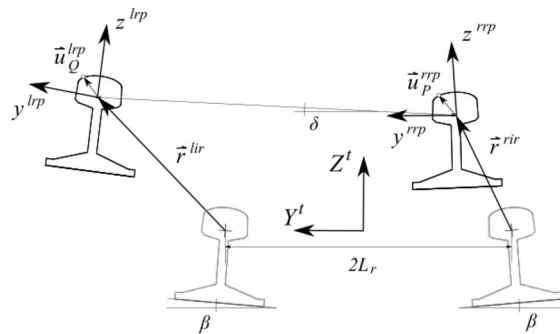


Figura 1. Definición de Irregularidades de Longitud de Onda Larga

- Alineamiento: $\frac{y^{lir} + y^{rir}}{2}$
- Ancho de vía: $y^{lir} - y^{rir}$
- Peralte: $\frac{z^{lir} + z^{rir}}{2}$
- Nivelación: $z^{lir} - z^{rir}$

Donde los valores de los desplazamientos están recogidos en los vectores definidos en la Figura 1:

$$\bar{\mathbf{r}}^{lir} = \begin{bmatrix} 0 \\ y^{lir} \\ z^{lir} \end{bmatrix} \quad \bar{\mathbf{r}}^{rir} = \begin{bmatrix} 0 \\ y^{rir} \\ z^{rir} \end{bmatrix}$$

En este sentido, se han elaborado múltiples sistemas para la medición de las irregularidades de las vías, divididos en dos vertientes. Por una parte, existen pequeños carros instrumentados operados manualmente que producen resultados muy precisos pero de forma muy ineficiente, utilizados en gran medida para la medida de la corrugación. Son capaces de medir únicamente cientos de metros al día [4, 6]. Por otra parte, existen vehículos automatizados conocidos como vehículos-laboratorio capaces de realizar una medición mucho más eficiente de la vía, pero que conllevan un gran coste para la industria [7, 8]. Además, la metodología utilizada ha ido evolucionando con el tiempo, pasando de utilizar sistemas únicamente mecánicos [9] o instrumentando un eje exclusivamente [10] a incorporar técnicas de visión artificial [11] y vehículos instrumentados completamente [12].

1.1. Contexto

Este trabajo se encuentra dentro de los proyectos de investigación llevados a cabo por el Grupo de Ingeniería Mecánica en la Escuela Técnica Superior de Ingeniería, donde se están desarrollando metodologías y sistemas para la medición de la geometría de la vía, incluyendo tanto las irregularidades de larga longitud de onda, como las irregularidades de corta longitud de onda o corrugación.

En trabajos previos se ha desarrollado un sistema embarcado para la auscultación geométrica de vías capaz de determinar las cuatro irregularidades de larga longitud de onda mediante la combinación de las medidas de sensores inerciales y de visión artificial con un modelo cinemático de sistema multicuerpo del vehículo en el que se instalan [13]. Este sistema denominado TGMS (*Track Geometry Measuring System*) se compone de un conjunto de dos cámaras de alta velocidad y dos láseres de proyección, un sensor inercial (IMU) y un sistema de odometría basado en encóder y/o balizas. A partir de las medidas obtenidas es posible calcular los cuatro parámetros que describen las irregularidades de larga longitud de onda a partir de los vectores definidos en la siguiente figura:

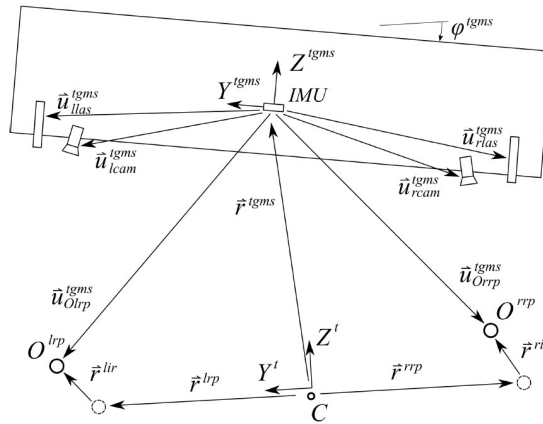


Figura 2. Vista Plana del TGMS

En ella se representa la vista plana del TGMS viajando a lo largo de una vía en un instante cualquiera. El punto C representa la línea media de la vía en su geometría teórica o de diseño, que es conocida. Los puntos O_{lrp} (*left rail profile*) y O_{rrp} (*right rail profile*) representan los centros de las secciones de los carriles izquierdo y derecho respectivamente en su posición real, que se han desplazado debido a las irregularidades una distancia dada por los vectores r^{lir} (*left irregularity*) y r^{rir} (*right irregularity*) respectivamente.

Se considera que los ángulos relativos entre el TGMS y el sistema local de la vía son pequeños, por lo que se puede asumir que las secciones de ambos carriles están contenidas en el mismo plano, teniendo así la misma coordenada de longitud de arco s asociada a la línea media de la vía. Estas secciones el resultado del corte del plano proyectado por un láser con su respectivo carril.

De esta forma, se observa que se cumplen las siguientes relaciones:

$$\begin{aligned} \vec{r}^{tgms} + \vec{u}_{O_{lrp}}^{tgms} &= \vec{r}^{lrp} + \vec{r}^{lir} \\ \vec{r}^{tgms} + \vec{u}_{O_{rrp}}^{tgms} &= \vec{r}^{rrp} + \vec{r}^{rir} \end{aligned} \quad (1.1)$$

Restando ambas expresiones se obtiene:

$$\vec{u}_{O_{lrp}}^{tgms} - \vec{u}_{O_{rrp}}^{tgms} = \vec{r}^{lrp} + \vec{r}^{lir} - (\vec{r}^{rrp} + \vec{r}^{rir}) \quad (1.2)$$

Que proyectando en el sistema de referencia local de la vía (t), sistema definido en la línea media de la vía de forma que el eje X sea tangente a ella en cada punto apuntando en la dirección de avance del vehículo se transforma en:

$$\mathbf{A}^{tgms} \left(\vec{u}_{O_{lrp}}^{tgms} - \vec{u}_{O_{rrp}}^{tgms} \right) = \mathbf{r}^{lrp} - \mathbf{r}^{rrp} + \mathbf{r}^{lir} - \mathbf{r}^{rir} \quad (1.3)$$

La matriz de orientación del sistema local del TGMS se construye a partir de tres ángulos de Euler correspondientes a los tres giros relativos de este sistema respecto al sistema local de la vía. Estos ángulos se pueden considerar pequeños, por lo que las componentes Y y Z de estas ecuaciones resultan:

$$\begin{bmatrix} 1 & -\varphi^{tgms} \\ \varphi^{tgms} & 1 \end{bmatrix} \begin{bmatrix} u_{Olrp,y} - u_{Orrp,y} \\ u_{Olrp,z} - u_{Orrp,z} \end{bmatrix} = \begin{bmatrix} 2L \\ 0 \end{bmatrix} + \begin{bmatrix} y^{lir} - y^{rir} \\ z^{lir} - z^{rir} \end{bmatrix} \quad (1.4)$$

Donde L representa el semi-ancho nominal de la vía. El último sumando representa las irregularidades relativas, por lo que reordenando la ecuación se determina la expresión para el cálculo del ancho de vía y la nivelación respectivamente:

$$\begin{bmatrix} \text{ancho vía} \\ \text{nivelación} \end{bmatrix} = \begin{bmatrix} 1 & -\varphi^{tgms} \\ \varphi^{tgms} & 1 \end{bmatrix} \begin{bmatrix} u_{Olrp,y} - u_{Orrp,y} \\ u_{Olrp,z} - u_{Orrp,z} \end{bmatrix} - \begin{bmatrix} 2L \\ 0 \end{bmatrix} \quad (1.5)$$

De forma equivalente, si se suman las expresiones iniciales se obtiene:

$$2\vec{r}^{tgms} + \vec{u}_{Olrp}^{tgms} + \vec{u}_{Orrp}^{tgms} = \vec{r}^{lir} + \vec{r}^{rir} \quad (1.6)$$

Proyectando sobre el sistema local de la vía:

$$2\mathbf{r}^{tgms} + \mathbf{A}^{tgms} (\vec{u}_{Olrp}^{tgms} + \vec{u}_{Orrp}^{tgms}) = \mathbf{r}^{lir} + \mathbf{r}^{rir} \quad (1.7)$$

Y de nuevo haciendo la suposición de ángulos pequeños, las componentes Y y Z de estas ecuaciones permiten calcular directamente las dos irregularidades absolutas de la vía, el alineamiento y el peralte de la forma:

$$\begin{bmatrix} r_y^{tgms} \\ r_z^{tgms} \end{bmatrix} + \frac{1}{2} \begin{bmatrix} 1 & -\varphi^{tgms} \\ \varphi^{tgms} & 1 \end{bmatrix} \begin{bmatrix} u_{Olrp,y} + u_{Orrp,y} \\ u_{Olrp,z} + u_{Orrp,z} \end{bmatrix} = \begin{bmatrix} \frac{y^{lir} + y^{rir}}{2} \\ \frac{z^{lir} + z^{rir}}{2} \end{bmatrix} = \begin{bmatrix} \text{alineamiento} \\ \text{peralte} \end{bmatrix} \quad (1.8)$$

Para poder aplicar estos resultados es necesario conocer la posición relativa de los carriles respecto al TGMS y la posición y orientación relativa del TGMS respecto a la línea media de la vía. Esto será el resultado de la visión artificial y de los algoritmos descritos en [13]. Los resultados presentados en este artículo son resultados *off-line*, obteniendo las medidas de los sensores inerciales y de las cámaras para la visión artificial con aplicaciones independientes y procesando estas señales a posteriori.

De cara al uso industrial de este sistema se quiere realizar la medición en tiempo real a 250 Hz. Este trabajo representa un avance en esta dirección desarrollando una aplicación única de escritorio capaz de obtener todas las señales necesarias y de procesar las imágenes en tiempo real para obtener la posición relativa de los carriles.

Paralelamente, el grupo de investigación está trabajando en algoritmos para cuantificar la corrugación de los carriles a partir de la respuesta dinámica de un vehículo que circula sobre ellos. Para ello, se dispone de una vía a escala 1:10 de 90 metros de longitud, así como de vehículos instrumentados adecuados a esta instalación.

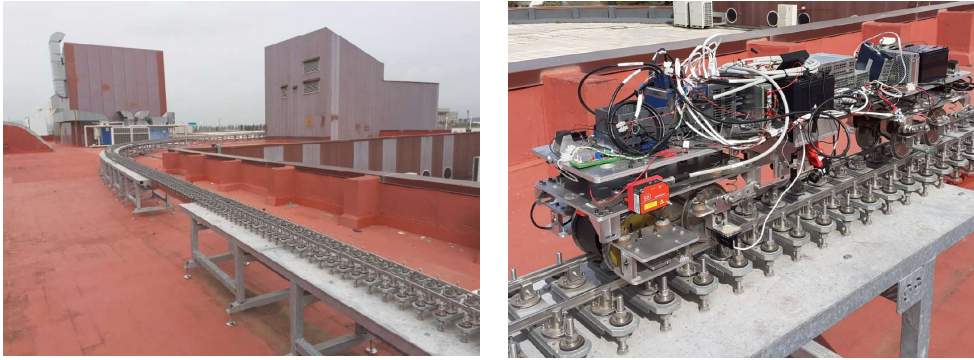


Figura 3. Vía a Escala y Vehículo Instrumentado

En esta vía existe un tramo con carriles con una superficie irregular mecanizada para simular la corrugación la cual se quiere medir utilizando perfilómetros láser de gran precisión para comprobar que el mecanizado realizado es correcto. Además, se desea poder usar estos sensores para medir la geometría de la corrugación en vías reales para poder tomar su geometría como referencia y poder contrastarla con la geometría mecanizada.

Con este objetivo en mente, en este trabajo se desarrolla un sistema para la medición de la corrugación de carriles mediante el uso de perfilómetros láser y se crea una aplicación de escritorio para poder interactuar con los sensores y obtener los resultados deseados.

1.2. Objetivos

A partir del contexto en el que se desarrolla este trabajo, se definen los siguientes objetivos:

- Unificar la adquisición de las señales de todos los sensores instalados en el TGMS en una única aplicación de escritorio.
- Implementar los algoritmos de visión artificial en tiempo real para obtener la posición relativa de los carriles a 250 Hz.
- Diseñar un sistema para la medida de la corrugación utilizando perfilómetros láser capaz de medir en la vía a escala y en una vía real.
- Desarrollar una aplicación de escritorio para interactuar con los perfilómetros láser y obtener la evolución de la corrugación a lo largo del carril.

2 MEDICIÓN DE IRREGULARIDADES DE LARGA LONGITUD DE ONDA

En este capítulo se describirá el sistema de medida utilizado para la medición de las irregularidades de larga longitud de onda. En primer lugar, se enumerarán los sensores y equipos que conforman el sistema de medida. A continuación, se desarrollará la visión artificial, desde la calibración de las cámaras hasta el procesado de imágenes para el cálculo de la posición del carril. Después se explicará la aplicación de escritorio desarrollada que implementa todos los algoritmos y permite al usuario interactuar con los sensores, exponiendo qué recursos utiliza, cómo se estructura el código y cómo se utiliza. Finalmente, se verificará el correcto funcionamiento de la visión artificial mostrando los resultados obtenidos en un mecanismo automático específico para su ensayo.

2.1. Descripción del Sistema

El sistema está compuesto por 2 cámaras de alta velocidad, 2 láseres de proyección de línea, un encóder rotatorio incremental, un sensor inercial (IMU) y un detector magnético conectados a un ordenador de adquisición de datos, que junto a las cámaras se conectan al ordenador mediante cable Ethernet. En esta sección se detallan cada uno de los elementos del sistema y qué tipo de señales se transmiten.

2.1.1 Sensores

2.1.1.1 Cámaras

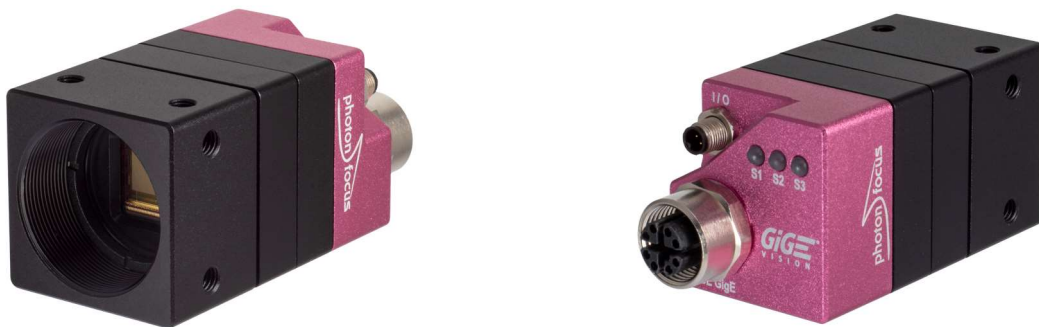


Figura 4. Cámara Photonfocus

El sistema tiene dos cámaras marca Photonfocus modelo MV0-D2048x1088-C01-3D06-768-G2 Monocromo. Estas cámaras son capaces de transmitir distinto tipo de información según el modo de trabajo que se utilice. En el modo denominado 3D, la cámara adquiere imágenes completas de 2048x1088 píxeles en escala de grises con valores entre 0 (negro) y 255 (blanco), que es equivalente a una matriz de 2048x1088 bytes. En cambio, en el modo denominado 2D la cámara obtiene la imagen completa y aplica un algoritmo de búsqueda de líneas capaz de determinar para cada columna de la matriz, a qué altura se encuentra una línea de la imagen, como puede ser la línea del láser proyectada sobre una superficie. De esta forma, la información que transmite se reduce a un vector de 2048 componentes correspondientes a la altura en píxeles de la línea en cada columna de la imagen.

Adicionalmente, se colocan dos láseres de proyección de línea que proyectarán una línea en cada carril mostrando una sección de su perfil. Cada cámara detectará esta línea en el modo de trabajo 2D y transmitirá la información de las coordenadas de la línea en la imagen en píxeles. Este modo de operación es el que se utilizará a la hora de adquirir los datos en tiempo real, pues es capaz de funcionar a alta velocidad.

En cuanto a la forma de transmisión de información, las cámaras poseen distintas configuraciones para enviar los datos. Para empezar, a la hora de trabajar en el modo 2D, se pueden agrupar imágenes en paquetes de datos con el objetivo de disminuir la frecuencia de transmisión de datos. Así, si se mide a una frecuencia de 250 Hz y se agrupan las imágenes en paquetes de 10, la frecuencia de transmisión de datos se verá reducida considerablemente a 25 Hz, disminuyendo la probabilidad de pérdida de información.

Finalmente, en el modo 2D las cámaras envían distinto tipo de información según el formato seleccionado. Para esta aplicación se utilizará el formato 3, en donde además de los datos de la imagen, se añade una marca de tiempo y otros parámetros utilizados por el algoritmo de detección de líneas. La marca de tiempo es de gran interés para comprobar si se ha perdido alguna imagen en el proceso de adquisición y transmisión de datos, pues al trabajar a frecuencia constante el incremento en la marca de tiempo entre imágenes consecutivas debe ser constante. La estructura de la información se muestra en la siguiente figura, obtenida del manual de las cámaras:

3D row	Bits								Description
	7	6	5	4	3	2	1	0	
0	PEAK[15:8]								Detected laser line coordinate. PEAK[15:4]: integer part, PEAK[3:0]: fractional part.
1	PEAK[7:0]								
2	'0'	'0'	LL_WIDTH[5:0]						LL_WIDTH: laser line width
3	LL_HEIGHT[3:0]				STAT				LL_HEIGHT: laser line height (4 MSB). STAT: Status information

Figura 5. Formato de datos de las cámaras

En este formato, asociado a cada columna de la imagen (2048 en total) se ensambla un paquete de información que contiene 4 bytes de datos que incluyen la coordenada vertical de la línea detectada, tanto su parte entera como su parte decimal, el grosor de la línea, su altura y parte de la información de la marca de tiempo. Este parámetro es una variable de 32 bits dividida entre los paquetes de las 8 primeras columnas de la imagen, teniendo cada uno 4 bits en su última fila, como muestra la figura, siendo el bit más significativo el primer bit asociado al paquete de la primera columna de la imagen. Mientras que las librerías proporcionadas por las cámaras realizan la interpretación de forma automática de la información de las coordenadas de los puntos de la línea detectada en la imagen, la información de la marca de tiempo se deberá inferir manualmente.

2.1.1.2 Inertial Measuring Unit (IMU)



Figura 6. Sensor Inercial (IMU)

El sensor inercial que se emplea en este sistema es una IMU del fabricante MicroStrain modelo 3DM-GX4-25. Este sensor digital lleva incorporados varios sensores, de los cuales se utilizarán el acelerómetro y giróscopo triaxiales para la obtención de las aceleraciones y velocidades angulares del TGMS al viajar por la vía a frecuencia constante. El acelerómetro es de tipo MEMS (*Micro Electro Mechanical System*) por lo que la aceleración que mide es la suma de las aceleraciones sufridas por el sensor más la aceleración de la gravedad proyectadas en su sistema local de referencia. La IMU se instala de forma que su sistema de referencia sea paralelo al del TGMS, como muestra la Figura 2.

La transmisión de datos se realiza a través de un puerto serie y la información recibida en cada medida se agrupa de la siguiente manera:

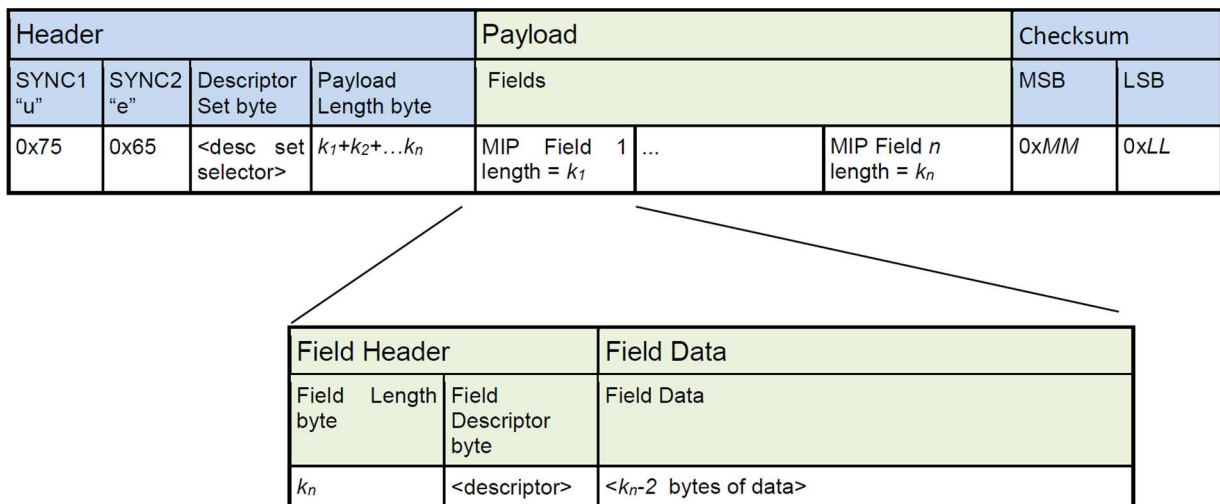


Figura 7. Formato de Datos de la IMU

Para esta aplicación, cada medida forma un paquete de 34 bytes. Los 4 primeros confirman el encabezado, y serán siempre constantes. A continuación, los siguientes 28 bytes contienen la información de los sensores, siendo los primeros 14 asociados al acelerómetro y los siguientes 14 asociados al giróscopo. En cada mitad, los dos primeros bytes son de encabezado, identificando el sensor y el tamaño de las medidas, y los siguientes 12 bytes corresponden a las medidas realizadas en las 3 direcciones del espacio, siendo cada componente una variable tipo *float* de 4 bytes cada una. Finalmente, el paquete finaliza con 2 bytes auxiliares cuyo valor sirve para comprobar que la información enviada es correcta. Esta estructura deberá ser tenida en cuenta para extraer la información de interés.

2.1.1.3 Encóder rotatorio



Figura 8. Encóder Rotatorio

El encóder rotario incremental utilizado en este sistema es un encóder del fabricante Kübler de 360 pulsos por revolución con dos canales para poder distinguir el sentido de giro. Se instalará acoplado al eje de cualquiera de las ruedas del vehículo ferroviario. A partir del recuento de pulsos, suponiendo rodadura sin deslizamiento y tomando como radio de la rueda su radio medio (r), se puede calcular la distancia recorrida por el vehículo

(s) de la siguiente manera:

$$s = \frac{2\pi r}{360} * pulsos \quad (2.1)$$

2.1.1.4 Detector Magnético



Figura 9. Detector Magnético

El sistema también puede contar con un detector magnético analógico utilizado para corregir la aproximación de la distancia recorrida dada por el encóder. Si en la vía donde circule el vehículo se colocan una serie de imanes a lo largo de ella a una distancia conocida, este sensor puede detectar cuándo el vehículo pasa sobre ellos, sabiendo exactamente su posición en ese instante.

2.1.2 Equipos de adquisición de datos

Tras describir los sensores del sistema, se procede a la descripción de los equipos utilizados para realizar la adquisición de señales a frecuencia constante y de manera síncrona.

2.1.2.1 Compact RIO



Figura 10. Compact RIO

Todos los sensores a excepción de las cámaras se deben conectar a una tarjeta o equipo de adquisición de datos que lea las distintas señales digitales y analógicas, las agrupe y envíe de alguna forma al ordenador principal. Para esto se utiliza el controlador Compact Rio cRio-9045 de National Instruments, equipo equipado con un ordenador y una FPGA (*Field-Programmable Gate Array*), con puertos de entrada y salida modulares según las tarjetas de adquisición que se instalen.

Para esta aplicación, se utilizan los módulos NI-9209 para la adquisición de la señal analógica de ± 10 V del detector magnético, NI-9401 para la adquisición de las señales digitales de los dos canales del encóder y NI-9870 para la adquisición de las medidas de la IMU a través de un puerto serie. Todas estas señales se conectan

directamente a la FPGA, que se programa utilizando el software Labview para habilitar los puertos de entrada y salida, leer las señales síncronamente a frecuencia constante, realizar el conteo de pulsos del encóder y calcular la distancia recorrida, agrupar toda la información y enviarla a una cola FIFO (*First In First Out*) para la comunicación con el ordenador del usuario.

Adicionalmente, se envía una señal digital a las cámaras correspondiente al disparador. Con ella, se determina el momento exacto de adquisición de una imagen, sincronizando de esta manera las cámaras con el resto de sensores.

2.1.2.2 Switch Gigabit



Figura 11. Switch Gigabit

Finalmente, el nexa de unión entre las cámaras y el Compact RIO es un Switch Gigabit de 8 puertos de la marca TP-Link modelo TL-SG108PE. Este equipo contiene 4 puertos PoE (*Power over Ethernet*) en los cuales se conectan las cámaras, sirviendo como alimentación como medio de tráfico de datos. En los otros 4 puertos se conectan el Compact RIO y el PC del usuario mediante cable Ethernet. De esta manera, todos los sensores se reducen a un único puerto Ethernet conectado al ordenador del usuario por donde se recibe toda la información de los sensores y se realiza la comunicación con ellos.

En definitiva, el esquema de conexión del sistema completo es el siguiente:

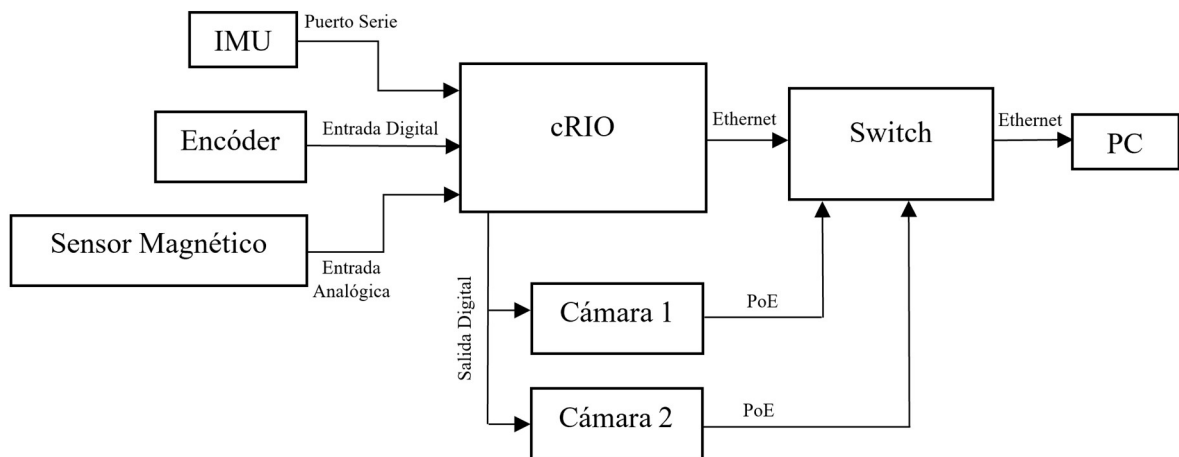


Figura 12. Esquema Sistema Completo

2.2. Visión Artificial

El procedimiento descrito en la sección 1.1 para el cálculo de las irregularidades de larga longitud de onda requiere del conocimiento de la posición relativa de los carriles respecto al TGMS. Esta magnitud es el resultado de la visión artificial que se desarrolla en esta sección. En primer lugar, es necesario realizar la calibración de las cámaras para obtener la denominada matriz de proyección, que relaciona las coordenadas de un punto de una imagen en píxeles con las coordenadas de dicho punto en el mundo real en milímetros. Después, se debe procesar la imagen para eliminar los puntos que no pertenezcan a la línea del láser proyectada en el carril y, finalmente, calcular la su posición.

2.2.1 Calibración

El método de calibración que se va a utilizar es el conocido como método de Zhang [13]. Su adaptación a la notación de la teoría de dinámica de sistemas multicuerpo se realiza en [14], artículo que servirá de referencia para lo que se desarrolla a continuación.

2.2.1.1 Coordenadas homogéneas de un vector

Antes de comenzar, es necesario introducir el concepto de coordenadas homogéneas de un vector. Dado un vector bidimensional, el vector de coordenadas homogéneas asociado es un vector de tres componentes de la forma:

$$\mathbf{v} = \begin{bmatrix} v_x \\ v_y \end{bmatrix}; \text{hom}(\mathbf{v}) = \hat{\mathbf{v}} = \begin{bmatrix} av_x \\ av_y \\ a \end{bmatrix} \quad (2.2)$$

De forma equivalente, dado un vector tridimensional, el vector de coordenadas homogéneas asociado es un vector de cuatro componentes de la forma:

$$\mathbf{v} = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix}; \text{hom}(\mathbf{v}) = \hat{\mathbf{v}} = \begin{bmatrix} av_x \\ av_y \\ av_z \\ a \end{bmatrix} \quad (2.3)$$

Con $a \neq 0$. Para obtener el vector original a partir de sus coordenadas homogéneas basta con dividir todas las componentes entre la última.

2.2.1.2 Matriz de proyección

Sea un punto P en el espacio de coordenadas del mundo real tridimensional y sea un punto P' su proyección en una imagen dada por sus dos coordenadas en la imagen en píxeles. La matriz de proyección \mathbf{P} es una matriz 3×4 que relaciona las coordenadas homogéneas del punto P con las coordenadas homogéneas del punto P' :

$$\hat{\mathbf{n}}_{P'} = \mathbf{P} \hat{\mathbf{r}}_P \rightarrow c \begin{bmatrix} \mathbf{n}_{P'} \\ \mathbf{1} \end{bmatrix} = \mathbf{P} \begin{bmatrix} \mathbf{r}_P \\ \mathbf{1} \end{bmatrix} \quad (2.4)$$

Esta expresión permite conocer la proyección de un punto en el espacio tridimensional en la imagen, pero no es suficiente para el caso contrario, pues se tienen tres ecuaciones y cuatro incógnitas (las tres componentes del vector y el parámetro c).

La matriz de proyección es el resultado de la multiplicación de dos matrices entre sí. La primera se denomina matriz de parámetros intrínsecos de la cámara, de tamaño 3×3 . Esta matriz representa la relación que existe entre las coordenadas de un punto en la imagen y de dicho punto en el mundo real proyectado en el sistema local de referencia de la cámara. Depende exclusivamente de la construcción interna de la cámara y del tamaño de la imagen que genera.

La segunda matriz se denomina matriz de parámetros externos, de tamaño 3×4 . En ella se incluye la matriz de orientación y el vector de traslación de un sistema de referencia dado en el cual se conocen las coordenadas

reales de un punto con respecto al sistema de referencia local de la cámara, cuya posición y orientación es desconocida a priori. La estructura de estas matrices viene dada por las siguientes expresiones:

$$\mathbf{P} = \mathbf{M}_{\text{int}}\mathbf{M}_{\text{ext}}$$

$$\mathbf{M}_{\text{int}} = \begin{bmatrix} \alpha & \gamma & n_x^0 \\ 0 & \beta & n_y^0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.5)$$

$$\mathbf{M}_{\text{ext}} = [\mathbf{A}_{\text{cam}}^T \quad -\mathbf{A}_{\text{cam}}^T \mathbf{r}_{\text{cam}}]$$

En general, para poder determinar las coordenadas de un punto en el espacio a partir de sus coordenadas en una imagen, son necesarias dos cámaras que observen el mismo punto desde dos puntos de vista diferentes, formando un sistema sobredeterminado de 6 ecuaciones y 5 incógnitas (las 3 coordenadas del punto y los dos parámetros c de cada cámara).

Alternativamente, si se conoce de antemano que el punto que se quiere determinar pertenece a una superficie conocida en el mundo real, el sistema de ecuaciones se cierra al añadir la ecuación de esta superficie, hecho que se utiliza a la hora de calibrar las cámaras y a la hora de realizar la medición, pues se conocerá la ecuación del plano que proyecta cada láser.

2.2.1.3 Procedimiento de calibración

Para llevar a cabo la calibración se necesita conocer la posición de un conjunto de puntos en los tres planos coordenados del sistema de referencia que se utilice. Para ello, se ha fabricado un calibrador que contiene dos tetraedros, uno para cada cámara, con una cuadrícula de 8x8 cuadrados en cada plano de 4 mm de lado y 1 mm de separación entre cada uno. El corte de las tres caras del tetraedro se toma como origen del sistema del mundo real, donde los cortes de estas caras dos a dos forman los 3 ejes coordenados, donde el eje Z es vertical.



Figura 13. Calibrador

Además, este elemento se puede atornillar rígidamente al TGMS de forma que la posición de los centros de cada tetraedro con respecto al sistema de referencia local del TGMS es conocida. La necesidad de conocer esta posición se explicará más adelante.

El procedimiento se divide en 6 pasos. Se parte de conocer la posición en el sistema local del calibrador de un conjunto de puntos en los tres planos coordenados en milímetros y de las coordenadas de esos puntos proyectados en una imagen tomada con la cámara en píxeles.

2.2.1.3.1 Paso 1: Normalización de coordenadas

Para la estabilidad numérica del procedimiento, en la implementación del método realizada en [15] se recomienda realizar una normalización de las coordenadas tanto reales como de imagen. Para ello, se utiliza la matriz de transformación:

$$\mathbf{N}_x = \begin{bmatrix} s_x & 0 & -s_x \bar{x} \\ 0 & s_y & -s_y \bar{y} \\ 0 & 0 & 1 \end{bmatrix} \quad (2.6)$$

De tal forma que los puntos normalizados de la imagen se calculan mediante la expresión:

$$\begin{bmatrix} \tilde{\mathbf{n}}_{P'} \\ \mathbf{1} \end{bmatrix} = \mathbf{N}_x \begin{bmatrix} \mathbf{n}_{P'} \\ \mathbf{1} \end{bmatrix} \quad (2.7)$$

Sabiendo que:

$$\begin{aligned} \bar{\mathbf{x}} &= \begin{bmatrix} \bar{x} \\ \bar{y} \end{bmatrix} = \frac{1}{N} \sum_{i=1}^N \mathbf{n}_{P'i} \\ s_x = s_y &= \sqrt{2}N \frac{1}{\sum_{i=1}^N \|\mathbf{n}_{P'i} - \bar{\mathbf{x}}\|} \end{aligned} \quad (2.8)$$

Para normalizar los puntos reales, se utilizan las 2 coordenadas distintas de cero, construyendo un vector de dos componentes con ellas. Además, la normalización se realiza con 6 conjuntos de puntos de forma independiente, que corresponden a dividir los puntos de la imagen en 3 según a qué plano pertenecen y los puntos reales en 3 según el plano al que pertenezcan.

2.2.1.3.2 Paso 2: Cálculo de Homografías:

Si se toman los puntos pertenecientes al plano XY del calibrador y sabiendo que la matriz de rotación $\mathbf{A}_{\text{cam}}^T$ contiene por columnas los tres vectores unitarios asociados a las tres direcciones del sistema de referencia local de la cámara, la ecuación (2.4) se puede escribir de la forma:

$$c \begin{bmatrix} \tilde{\mathbf{n}}_{P'} \\ \mathbf{1} \end{bmatrix} = \mathbf{M}_{\text{int}}[\bar{\mathbf{i}} \quad \bar{\mathbf{j}} \quad \bar{\mathbf{k}} \quad -\bar{\mathbf{r}}_{\text{cam}}] \begin{bmatrix} r_x^P \\ r_y^P \\ \mathbf{0} \\ \mathbf{1} \end{bmatrix} = \mathbf{M}_{\text{int}}[\bar{\mathbf{i}} \quad \bar{\mathbf{j}} \quad -\bar{\mathbf{r}}_{\text{cam}}] = \mathbf{H}_{XY} \begin{bmatrix} r_x^P \\ r_y^P \\ \mathbf{1} \end{bmatrix} \quad (2.9)$$

Donde la matriz \mathbf{H}_{XY} define la homografía del plano XY . Desarrollando esta expresión:

$$c \begin{bmatrix} n_x^{P'} \\ n_y^{P'} \\ \mathbf{1} \end{bmatrix} = \begin{bmatrix} H_{11}^{XY} & H_{12}^{XY} & H_{13}^{XY} \\ H_{21}^{XY} & H_{22}^{XY} & H_{23}^{XY} \\ H_{31}^{XY} & H_{32}^{XY} & H_{33}^{XY} \end{bmatrix} \begin{bmatrix} r_x^P \\ r_y^P \\ \mathbf{1} \end{bmatrix} \quad (2.10)$$

Se puede sustituir la tercera ecuación en las dos primeras para eliminar la variable c , y se ensambla un sistema de ecuaciones de 2 ecuaciones y 9 incógnitas correspondientes a las 9 componentes de la matriz de homografía de la forma:

$$\begin{bmatrix} -r_x^P & -r_y^P & -1 & 0 & 0 & 0 & n_x^{P'} r_x^P & n_x^{P'} r_y^P & n_x^{P'} \\ 0 & 0 & 0 & r_x^P & r_y^P & -1 & n_y^{P'} r_x^P & n_y^{P'} r_y^P & n_y^{P'} \end{bmatrix} \begin{bmatrix} H_{11}^{XY} \\ H_{12}^{XY} \\ H_{13}^{XY} \\ H_{21}^{XY} \\ H_{22}^{XY} \\ H_{23}^{XY} \\ H_{31}^{XY} \\ H_{32}^{XY} \\ H_{33}^{XY} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (2.11)$$

Este sistema de ecuaciones se puede construir para cada punto perteneciente al plano XY y su correspondiente proyección en la imagen. De esta forma, para que se pueda resolver se necesita un mínimo de 5 puntos para tener un sistema sobredeterminado con 10 ecuaciones (2 por punto) y 9 incógnitas. Si se agrupan las ecuaciones de todos los puntos considerados se llega a un sistema lineal de expresión:

$$\mathbf{L}_{XY} \hat{\mathbf{H}}_{XY} = \mathbf{0} \quad (2.12)$$

Cuya solución óptima en el sentido de mínimos cuadrados corresponde al autovector asociado al mínimo autovalor de $\mathbf{L}_{XY}^T \mathbf{L}_{XY}$.

De manera equivalente se procede con los planos XZ e YZ , utilizando las coordenadas no nulas de los puntos del calibrador, obteniendo así dos matrices de homografía adicionales, \mathbf{H}_{XZ} y \mathbf{H}_{YZ} .

2.2.1.3.3 Paso 3: Cálculo de los Parámetros Intrínsecos

Trabajando a partir de la matriz de homografía del plano XY , despejando de la definición de matriz de homografía dada en la expresión (2.9), se obtiene que:

$$\mathbf{M}_{\text{int}}^{-1} \mathbf{H}_{XY} = [\bar{\mathbf{i}} \quad \bar{\mathbf{j}} \quad -\bar{\mathbf{r}}_{\text{cam}}] \quad (2.13)$$

Sea \mathbf{h}_i^{XY} la columna i de la matriz de homografía \mathbf{H}_{XY} . Entonces:

$$\begin{aligned} \bar{\mathbf{i}} &= \mathbf{M}_{\text{int}}^{-1} \mathbf{h}_1^{XY} \\ \bar{\mathbf{j}} &= \mathbf{M}_{\text{int}}^{-1} \mathbf{h}_2^{XY} \end{aligned} \quad (2.14)$$

Vectores que son unitarios y perpendiculares entre sí, por lo que se cumplen las relaciones:

$$\begin{aligned} (\mathbf{h}_1^{XY})^T \mathbf{M}_{\text{int}}^{-T} \mathbf{M}_{\text{int}}^{-1} \mathbf{h}_2^{XY} &= \mathbf{0} \\ (\mathbf{h}_1^{XY})^T \mathbf{M}_{\text{int}}^{-T} \mathbf{M}_{\text{int}}^{-1} \mathbf{h}_1^{XY} - (\mathbf{h}_2^{XY})^T \mathbf{M}_{\text{int}}^{-T} \mathbf{M}_{\text{int}}^{-1} \mathbf{h}_2^{XY} &= \mathbf{0} \end{aligned} \quad (2.15)$$

Definiendo la matriz simétrica \mathbf{B} de la forma:

$$\mathbf{B} = \mathbf{M}_{\text{int}}^{-T} \mathbf{M}_{\text{int}}^{-1} = \begin{bmatrix} \frac{1}{\alpha^2} & \frac{-\gamma}{\alpha^2 \beta} & \frac{n_y^0 \gamma - n_x^0 \beta}{\alpha^2 \beta} \\ \frac{\gamma}{\alpha^2 \beta^2} + \frac{1}{\beta^2} & \frac{-\gamma(n_y^0 \gamma - n_x^0 \beta)}{\alpha^2 \beta^2} - \frac{n_y^0}{\beta^2} \\ \text{SIM.} & \frac{(n_y^0 \gamma - n_x^0 \beta)^2}{\alpha^2 \beta^2} + \frac{(n_y^0)^2}{\beta^2} + 1 \end{bmatrix} \quad (2.16)$$

Y el producto:

$$\begin{aligned} (\mathbf{h}_i^{XY})^T \mathbf{B} \mathbf{h}_j^{XY} &= \\ [\mathbf{H}_{1i}^{XY} \mathbf{H}_{1j}^{XY} & \mathbf{H}_{1i}^{XY} \mathbf{H}_{1j}^{XY} + \mathbf{H}_{2i}^{XY} \mathbf{H}_{1j}^{XY} & \mathbf{H}_{2i}^{XY} \mathbf{H}_{2j}^{XY} & \mathbf{H}_{3i}^{XY} \mathbf{H}_{1j}^{XY} + \mathbf{H}_{1i}^{XY} \mathbf{H}_{3j}^{XY} & \mathbf{H}_{3i}^{XY} \mathbf{H}_{2j}^{XY} + \mathbf{H}_{2i}^{XY} \mathbf{H}_{3j}^{XY} & \mathbf{H}_{3i}^{XY} \mathbf{H}_{3j}^{XY}] \cdot \begin{bmatrix} \mathbf{B}_{11} \\ \mathbf{B}_{12} \\ \mathbf{B}_{22} \\ \mathbf{B}_{13} \\ \mathbf{B}_{23} \\ \mathbf{B}_{33} \end{bmatrix} \\ &= \mathbf{v}_{ij}^{XY} \hat{\mathbf{B}} \end{aligned} \quad (2.17)$$

Las ecuaciones (2.15) se pueden reescribir de la forma:

$$\begin{bmatrix} \mathbf{v}_{12}^{XY} \\ \mathbf{v}_{11}^{XY} - \mathbf{v}_{22}^{XY} \end{bmatrix} \hat{\mathbf{B}} = \mathbf{0} \quad (2.18)$$

Siguiendo este mismo razonamiento con las matrices de homografía de los otros dos planos, se obtienen 4 ecuaciones adicionales, formando un total de 6 ecuaciones con 6 incógnitas:

$$\begin{bmatrix} \mathbf{v}_{12}^{XY} \\ \mathbf{v}_{11}^{XY} - \mathbf{v}_{22}^{XY} \\ \mathbf{v}_{12}^{XZ} \\ \mathbf{v}_{11}^{XZ} - \mathbf{v}_{22}^{XZ} \\ \mathbf{v}_{12}^{YZ} \\ \mathbf{v}_{11}^{YZ} - \mathbf{v}_{22}^{YZ} \end{bmatrix} \hat{\mathbf{B}} = \mathbf{V} \hat{\mathbf{B}} = \mathbf{0} \quad (2.19)$$

Cuya solución es de nuevo el autovector asociado al mínimo autovalor de $\mathbf{V}^T \mathbf{V}$. Despejando de las expresiones de los coeficientes de la matriz \mathbf{B} , se calculan los parámetros intrínsecos de la cámara:

$$\begin{aligned} \lambda &= B_{33} + \frac{B_{13}^2 + n_y^0 (B_{12} B_{13} - B_{11} B_{23})}{B_{11}} \\ n_y^0 &= \frac{B_{12} B_{13} - B_{11} B_{23}}{B_{11} B_{22} - B_{12}^2} \\ \alpha &= \sqrt{\frac{\lambda}{B_{11}}} \\ \gamma &= -\frac{B_{12}^2 \alpha^2 \beta}{\lambda} \\ n_x^0 &= \frac{\gamma n_y^0}{\beta} - \frac{B_{13} \alpha^2}{\lambda} \end{aligned} \quad (2.20)$$

2.2.1.3.4 Paso 4: Cálculo de los Parámetros Extrínsecos

Partiendo de las expresiones (2.14), trabajando en el plano XY , se tiene que:

$$\begin{aligned} \bar{\mathbf{i}}_{XY} &= e \mathbf{M}_{\text{int}}^{-1} \mathbf{h}_1^{XY} \\ \bar{\mathbf{j}}_{XY} &= e \mathbf{M}_{\text{int}}^{-1} \mathbf{h}_2^{XY} \\ e_1 &= \frac{1}{\|\mathbf{M}_{\text{int}}^{-1} \mathbf{h}_1^{XY}\|} \\ e_2 &= \frac{1}{\|\mathbf{M}_{\text{int}}^{-1} \mathbf{h}_2^{XY}\|} \end{aligned} \quad (2.21)$$

En teoría se debe cumplir que $e_1 = e_2 = e = 1$, pero al ser la matriz de homografía una aproximación por mínimos cuadrados, existe un error que hace que los vectores calculados a partir de ella no sean unitarios. Por tanto, se puede calcular la matriz de parámetros extrínsecos empleando las siguientes expresiones:

$$\begin{aligned} \bar{\mathbf{i}}_{XY} &= e_1 \mathbf{M}_{\text{int}}^{-1} \mathbf{h}_1^{XY} \\ \bar{\mathbf{j}}_{XY} &= e_2 \mathbf{M}_{\text{int}}^{-1} \mathbf{h}_2^{XY} \\ \bar{\mathbf{k}}_{XY} &= \bar{\mathbf{i}}_{XY} \wedge \bar{\mathbf{j}}_{XY} \\ \bar{\mathbf{r}}_{\text{cam}}^{XY} &= -\frac{e_1 + e_2}{2} \mathbf{M}_{\text{int}}^{-1} \mathbf{h}_3^{XY} \end{aligned} \quad (2.22)$$

$$\mathbf{M}_{\text{ext}}^{XY} = [\bar{\mathbf{i}}_{XY} \quad \bar{\mathbf{j}}_{XY} \quad \bar{\mathbf{k}}_{XY} \quad -\bar{\mathbf{r}}_{\text{cam}}^{XY}] = [(\mathbf{A}_{\text{cam}}^{XY})^T \quad -\bar{\mathbf{r}}_{\text{cam}}^{XY}]$$

A la matriz de rotación debe ser ortogonal, pero como se trabaja con aproximaciones, en general el resultado no lo será. La matriz ortogonal más parecida a la calculada se puede determinar a partir de la descomposición SVD (*Singular Value Decomposition*) de la matriz:

$$[\mathbf{U}, \mathbf{S}, \mathbf{V}] = \text{svd}(\mathbf{A}_{\text{cam}}^{\text{XY}})$$

$$\mathbf{A}_{\text{cam}}^{\text{XY}} = \mathbf{UV}^T \quad (2.23)$$

Este procedimiento se repite con las matrices de homografías de los otros planos, obteniendo 3 matrices de parámetros extrínsecos distintas en general. Esto no es razonable, pues la matriz de parámetros intrínsecos es única. Este hecho da lugar al siguiente paso del procedimiento.

2.2.1.3.5 Optimización de los Parámetros

Para definir el problema de optimización, resulta conveniente trabajar con los parámetros de Rodríguez para evitar trabajar con la matriz de rotación completa de la cámara. La orientación de un sólido en el espacio se define a través del ángulo girado α alrededor de un eje de dirección \mathbf{d} , siendo los parámetros de Rodríguez:

$$\mathbf{g} = \tan\left(\frac{\alpha}{2}\right) \mathbf{d} \quad (2.24)$$

Las relaciones entre estos parámetros y las componentes de la matriz de rotación asociada son:

$$\mathbf{g} = \frac{\mathbf{1}}{\text{traza}(\mathbf{A})} \begin{bmatrix} A_{23} - A_{32} \\ A_{31} - A_{13} \\ A_{12} - A_{21} \end{bmatrix} \quad (2.25)$$

$$\mathbf{A} = \mathbf{I} + \frac{2}{\mathbf{1} + \mathbf{g}^T \mathbf{g}} (\tilde{\mathbf{g}} + \tilde{\mathbf{g}}\tilde{\mathbf{g}})$$

Siendo $\tilde{\mathbf{g}}$ la matriz antisimétrica asociada al vector \mathbf{g} .

Utilizando la matriz de parámetros intrínsecos calculada en el Paso 3 y la media de las tres matrices de parámetros extrínsecos calculadas en el apartado 4, se puede calcular la proyección de los puntos del calibrador en la imagen con la ecuación (2.4), que en general no coincidirán con las proyecciones reales en la imagen. Así, se define el error en cada punto mediante la expresión:

$$\varepsilon_{p_i} = \|\text{hom}^{-1}(\hat{\mathbf{n}}_{p_i}) - \text{hom}^{-1}(\hat{\mathbf{n}}_{p_i})\| \quad (2.26)$$

Por tanto, se buscarán aquellas matrices que minimicen la suma de los errores al cuadrado de cada punto:

$$\text{Min. } E(\mathbf{p}_{\text{cam}}) = \sum_{i=1}^N (\varepsilon_{p_i})^2 \quad (2.27)$$

Donde $\mathbf{p}_{\text{cam}} = [\alpha \quad \beta \quad \gamma \quad n_x^0 \quad n_y^0 \quad \mathbf{g}_{\text{cam}}^T \quad \mathbf{r}_{\text{cam}}^T]^T$. Dada la naturaleza del problema, se emplea el algoritmo de optimización de Levenberg-Marquardt, indicado para la minimización de un sumatorio de elementos al cuadrado. Además, como aproximación inicial se utilizarán los parámetros intrínsecos obtenidos previamente y la media de los parámetros extrínsecos calculados.

Con los nuevos valores de los parámetros, se reconstruyen las matrices de parámetros que relacionan la posición de un punto expresada en el sistema local de referencia del calibrador con las coordenadas de su proyección en la imagen.

2.2.1.3.6 Paso 6: Cambio de Sistema de Referencia

La calibración que se desea obtener es la que relaciona la posición de un punto expresada en el sistema local del TGMS con las coordenadas de su proyección en la imagen. Para ello, es necesario modificar la matriz de parámetros externos sabiendo que su contenido representa la matriz de rotación y el vector de traslación del sistema local de referencia de la cámara respecto al sistema local del calibrador.

Como se indicó anteriormente, el calibrador se atornilla rígidamente al TGMS de modo que la posición y orientación del sistema local del calibrador respecto al origen del sistema local del TGMS son conocidas. Por tanto, basta realizar un cambio de sistema de coordenadas para construir la nueva matriz de parámetros

extrínsecos:

$$\begin{aligned} \mathbf{A} &= \mathbf{A}_{\text{cal}}\mathbf{A}_{\text{cam}} \\ \mathbf{r} &= \mathbf{r}_{\text{cal}} + \mathbf{A}_{\text{cam}}\mathbf{r}_{\text{cam}} \\ \hat{\mathbf{M}}_{\text{ext}} &= [\mathbf{A}^T \quad -\mathbf{A}^T\mathbf{r}] \end{aligned} \quad (2.28)$$

2.2.2 Procesado de Imágenes

Una vez realizada la calibración, se puede proceder a la adquisición de datos en tiempo real. Como se describió en el apartado 2.1.1.1, las cámaras internamente transforman la matriz una imagen en escala de grises en un vector con la posición de la línea del láser detectada, como se muestra en la siguiente figura:



Figura 14. Transformación de Imagen Completa a Líneas

Se observa cómo, además de la línea proyectada en la cabeza del carril, existen otras proyecciones en otros elementos de la vía que la cámara es capaz de ver. Además, según las condiciones de luz ambiental, la calidad de la imagen puede verse afectada, apareciendo puntos adicionales derivados de reflejos no deseados. Es por ello que resulta necesario filtrar las imágenes en tiempo real para retener exclusivamente los puntos del carril.

2.2.2.1 Filtrado de imagen

El proceso de filtrado de una imagen se realiza directamente tras tomar una fotografía con la cámara. Consiste en definir una región de interés móvil que vaya siguiendo el movimiento del carril en la imagen. El procedimiento se realiza en los siguientes pasos:

- **Paso 1:** Antes de iniciar la adquisición de imágenes con el vehículo en reposo en su posición de partida, definir una región de interés rectangular que contenga la línea proyectada en el carril. Esta definición puede ser manual, introduciendo los extremos del rectángulo por pantalla, traducándose en unos límites superiores e inferiores en las coordenadas de los puntos de la imagen dentro de los cuales se hayan los puntos de interés. El tamaño de dicha región de interés dependerá de a qué distancia se haya colocado la cámara del carril, apareciendo la línea del láser con mayor o menor tamaño.
- **Paso 2:** Calcular el centro de gravedad de los puntos dentro de la región de interés mediante la media aritmética de sus coordenadas en la imagen.
- **Paso 3:** Comenzar con la adquisición sucesiva de imágenes. Para la imagen i , se utilizan los puntos que quedan dentro de la región de interés definida en la imagen $i-1$, calculándose de nuevo el centro de gravedad de estos puntos.
- **Paso 4:** Calcular la diferencia entre las coordenadas de los centros de gravedad de los puntos de la imagen i y la imagen $i-1$, cuantificando el desplazamiento que ha sufrido la línea del carril entre dos fotogramas consecutivos.
- **Paso 5:** Aplicar este desplazamiento al rectángulo que define la región de interés, moviendo sus límites las cantidades calculadas en el paso anterior. Esta será la nueva región de interés desplazada que se utilizará en la imagen $i+1$.

- Paso 6: Volver al Paso 2 e iterar para cada fotograma.

Cabe destacar que en este procedimiento se supone que el desplazamiento del carril entre fotogramas consecutivos es muy pequeño, siendo válidos los límites de la región de interés calculados en una imagen para la imagen siguiente. El resultado de este filtrado se muestra a continuación:

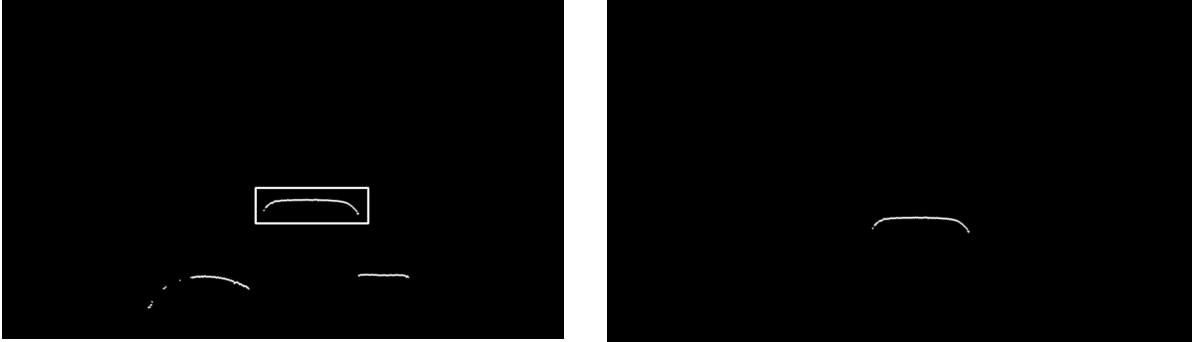


Figura 15. Resultado del Filtrado de Imágenes

2.2.2.2 Cálculo de la posición del carril

Tras filtrar la imagen, se puede calcular la posición del carril haciendo uso de la matriz de proyección calculada en el proceso de calibración. Para ello, en primer lugar se deben transformar las coordenadas de los puntos del carril en la imagen a sus coordenadas respecto al sistema local del TGMS. Las 3 ecuaciones definidas en (2.4) no son suficientes para calcular las 3 coordenadas del punto y el parámetro c , pero se sabe de antemano que los puntos pertenecen al plano proyectado por el láser. La ecuación de dicho plano cierra el sistema de ecuaciones.

El plano proyectado por el láser es un plano transversal a la vía, cortando a los carriles transversalmente. Este plano par tanto es paralelo al plano YZ del sistema local del TGMS, por lo que su ecuación en este sistema es:

$$x = k \tag{2.29}$$

Para determinar la constante k , se proyecta el láser sobre el calibrador atornillado rígidamente al TGMS y se mide a qué distancia se produce el corte del láser con los ejes X e Y del calibrador, magnitudes que deben coincidir ya que el calibrador se encuentra girado un ángulo de 45 grados. La siguiente figura ilustra el procedimiento:

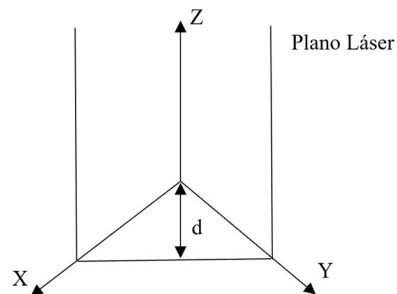
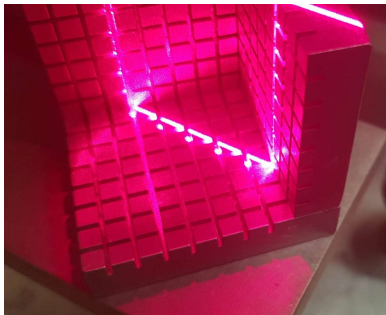


Figura 16. Cálculo de la Constante del Plano del Láser

En el montaje mostrado, la línea del láser corta al eje X del calibrador e el vértice del quinto cuadrado, lo que se traduce en una distancia de 24 milímetros. A partir de este valor, se determina la distancia d , y sabiendo la coordenada x de la posición del centro del calibrador en el sistema local del TGMS (u_x^O), la constante k viene dada por:

$$\begin{aligned}
d &= 24 \cos 45^\circ \\
k &= u_x^O - d
\end{aligned}
\tag{2.30}$$

En definitiva, el sistema de ecuaciones a resolver para cada punto del carril es:

$$\begin{cases}
c \begin{bmatrix} n_x^{P'} \\ n_y^{P'} \\ 1 \end{bmatrix} = \begin{bmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \end{bmatrix} \begin{bmatrix} u_x^P \\ u_y^P \\ u_z^P \\ 1 \end{bmatrix} \\
u_x^P = k
\end{cases}
\tag{2.31}$$

Por la naturaleza de este sistema, resulta sencillo reducirlo a un sistema de dos ecuaciones con dos incógnitas, introduciendo la última ecuación en las tres primeras y sustituyendo el valor del parámetro c dado por la tercera ecuación en el resto. De esta manera, resulta:

$$\begin{bmatrix} P_{12} - P_{32}n_x^{P'} & P_{13} - P_{33}n_x^{P'} \\ P_{22} - P_{32}n_y^{P'} & P_{23} - P_{33}n_y^{P'} \end{bmatrix} \begin{bmatrix} u_y^P \\ u_z^P \end{bmatrix} = \begin{bmatrix} (P_{31}k + P_{34})n_x^{P'} - (P_{11}k + P_{14}) \\ (P_{31}k + P_{34})n_y^{P'} - (P_{21}k + P_{24}) \end{bmatrix}
\tag{2.32}$$

Finalmente, la posición del carril respecto al sistema de referencia local del TGMS vendrá dada por la posición del centro de gravedad del perfil, calculado con la media aritmética de las coordenadas de todos los puntos del carril.

Para que este algoritmo funcione, la cámara debe poder captar en todo momento el perfil del carril completo, ya que si se rompe la simetría, el centro de gravedad de los puntos observados no coincidirá con el centro real del carril.

2.3. Aplicación de escritorio

Para la implementación de los algoritmos de visión artificial y la adquisición de las señales de todos los sensores del sistema, se ha desarrollado una aplicación de escritorio utilizando el lenguaje C++ para la programación. La elección de este lenguaje se debe a que, al ser de más bajo nivel que otros como Python, permite gestionar de manera más eficiente los recursos del ordenador. Esto da lugar a una mayor velocidad de procesamiento, necesaria para poder alcanzar el objetivo de trabajar en tiempo real.

Por otra parte, se ha decidido desarrollar la aplicación para que sea compatible únicamente con el sistema operativo Windows, ya que los recursos necesarios para la comunicación con los sensores del sistema sólo están disponibles para este sistema operativo.

2.3.1 Recursos utilizados

Para la programación de la aplicación, la compilación y la depuración del código en C++ se ha utilizado el entorno de desarrollo integrado (IDE) Visual Studio 2022. Adicionalmente, se ha hecho uso de una serie de librerías externas:

- Eigen 3: Librería matemática utilizada para implementar los algoritmos para la calibración de las cámaras y para el procesamiento de imágenes.
- ImGui: Librería para el desarrollo de la interfaz gráfica de la aplicación.
- ImPlot: Módulo adicional de ImGui para la representación de gráficas.
- GLFW (Graphics Library Framework): Librería utilizada para generar de forma sencilla una ventana para la aplicación y utilizada por ImGui.
- OpenGL (Open Graphics Library): Librería para el desarrollo de aplicaciones gráficas utilizada por ImGui.

- Glew (OpenGL Extension Wrangler Library): Librería utilizada para cargar de forma sencilla las funciones de OpenGL contenidas en el controlador de la tarjeta gráfica del PC.
- STB_Image: Librería para la lectura y escritura de imágenes, utilizada para cargar iconos en la interfaz gráfica de la aplicación.
- PF SDK (Photonfocus Software Development Kit): Librerías proporcionadas por el fabricante de las cámaras para su configuración y comunicación con el PC.
- NI FPGA C Interface API Generator (Application Programming Interface): Software de National Instruments en el que se generan archivos de código en C a partir del programa compilado de la FPGA del Compact RIO para comunicar el PC con la FPGA e intercambiar información.
- Librería estándar C++20: Librería estándar de C++ en su versión de 2020.
- Windows SDK: Librería para el desarrollo de aplicaciones para el sistema operativo Windows.

2.3.2 Estructura del código

La aplicación se ha desarrollado utilizando la programación orientada a objetos en C++ [16, 17, 18]. El código se ha dividido en una serie de clases que encapsulan las distintas partes del código de forma clara. En esta sección, se describen la arquitectura del programa, las distintas clases implementadas y las instrucciones de uso de la aplicación.

2.3.2.1 Arquitectura del programa

En rasgos generales, el programa consiste en la inicialización de la aplicación, el bucle de ejecución principal en el que se genera la interfaz gráfica a la frecuencia del monitor del ordenador (normalmente 60 Hz) y la finalización al cerrar la aplicación.

Adicionalmente, a la hora de grabar las señales de los sensores y recibir datos de ellos, es necesario hacer uso de la programación con subprocesos o multihilo. Esto es, paralelamente a la ejecución del bucle principal de la aplicación, se ejecutan los bucles de adquisición de datos, uno para cada cámara y otro adicional para el Compact RIO. Esta arquitectura presenta desafíos adicionales a la hora de comunicar los distintos subprocesos entre ellos [19]. Si bien los bucles de adquisición de datos son independientes entre sí, ya que la adquisición se sincroniza externamente, para la visualización de los datos adquiridos en tiempo real es necesario compartir los datos adquiridos en un subproceso con el bucle principal de la aplicación que controla la interfaz gráfica.

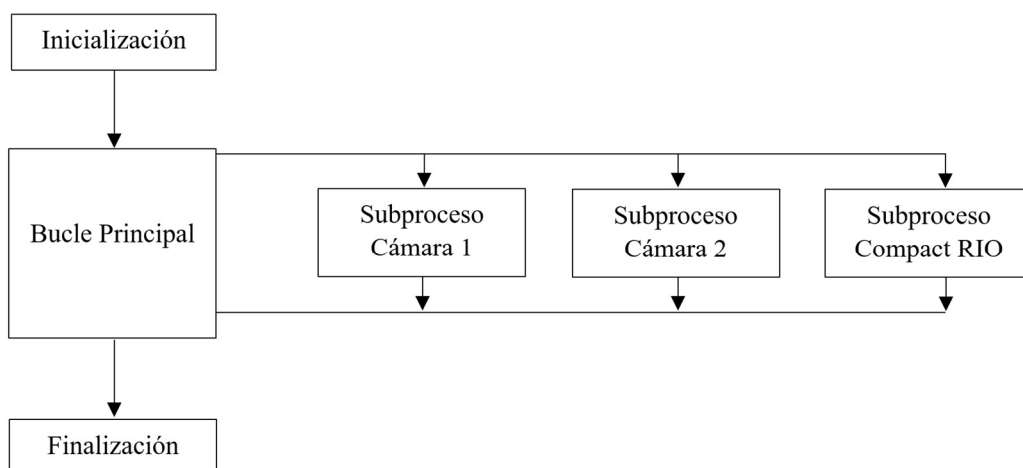


Figura 17. Arquitectura de la Aplicación

Para indicar la inicialización y terminación de la adquisición de datos, se utiliza una variable booleana atómica, variable que garantiza que dos subprocesos no puedan leer o escribir la variable a la vez, dando lugar a un comportamiento indefinido.

Para compartir los datos, se ha desarrollado un contenedor de datos específico que permite la escritura y

lectura de datos por varios subprocesos paralelamente, usando las herramientas disponibles en la librería estándar de C++ para el manejo de subprocesos.

De esta forma, se permite que cada subproceso se ejecute a una frecuencia diferente y además se utilizan los recursos disponibles en cualquier ordenador con varios núcleos en su procesador, mejorando la eficiencia computacional del código.

2.3.2.2 Clases implementadas

El código se divide en un conjunto de clases que se detallan a continuación:

- *Application*: Clase principal del programa encargada de la inicialización, ejecución del bucle principal y finalización de la aplicación. En la inicialización, se configuran las librerías utilizadas en el programa, se crea una ventana en el PC y se inician las ventanas que componen la interfaz gráfica de la aplicación. El bucle principal actualiza todas estas ventanas y recibe e interpreta las entradas de ratón y teclado. Finalmente, en la finalización se libera toda la memoria utilizada y se cierra la ventana del ordenador. Esta clase tiene una estructura de “*Singleton*”, es decir, sólo se puede crear una única instancia de esta clase a lo largo de todo el programa, pues el programa gestiona una única aplicación.
- *DockspaceWindow*: Clase para generar un espacio para el acoplamiento de ventanas dentro de la ventana principal. Además, genera la barra de herramientas principal con opciones para conectarse a los equipos y gestionar los archivos de datos.
- *ImGuiWindow*: Clase base para todas las clases asociadas a una ventana de la aplicación. Proporciona métodos básicos para abrir y cerrar las ventanas y para su actualización.
- *ConnectWindow*: Clase derivada en la que se implementa la ventana para seleccionar las cámaras disponibles a las que conectarse.
- *DevicesWindow*: Clase derivada en la que se muestran todos los equipos conectados y se controla la comunicación con ellos configurándolos, iniciando y parando la adquisición de datos e indicando el directorio de trabajo donde se guardarán los archivos de datos.
- *MessagesWindow*: Clase derivada en la que se muestran mensajes al usuario para saber si las acciones ejecutadas han sido satisfactorias o han producido errores.
- *ViewerWindow*: Clase derivada en la que se implementa un visor para una cámara que muestra la imagen completa adquirida y el resultado del algoritmo de búsqueda de líneas de la cámara. Además, en esta ventana se define la región de interés inicial para el filtrado sucesivo de las imágenes.
- *CalibrationWindow*: Clase derivada encargada de generar una ventana para realizar la calibración de una cámara. En ella se adquieren fotografías del calibrador y se seleccionan los puntos de calibración en la imagen.
- *CalibrationResultWindow*: Clase derivada en la que se muestran los resultados de la calibración realizada comparando los puntos seleccionados en la imagen y la proyección calculada a partir de la calibración.
- *DisplayWindow*: Clase derivada en la que se implementa una ventana para visualizar las imágenes en tiempo real en el proceso de adquisición de datos, comparando entre la imagen sin filtrar y filtrada.
- *SingnalsWindow*: Clase derivada para generar una ventana en la que se carguen las medidas de los sensores conectados al Compact RIO y se representen gráficamente.
- *StatsWindow*: Clase derivada en la que se recogen las medidas de los tiempos de computación del procesado de imágenes y se muestran por pantalla.
- *Network*: Clase encargada de detectar las cámaras conectadas a la red, obtener su información y seleccionar a qué cámara conectarse.
- *Camera*: Clase encargada de la conexión y desconexión con las cámaras, su configuración y la adquisición de imágenes.

- Profile: Clase encargada del procesado de imágenes, incluyendo el filtrado y el cálculo de la posición del carril implementado los algoritmos descritos en el apartado 2.2.2.
- ZhangCalibration: Clase en la que se implementa el algoritmo para la calibración de las cámaras explicado en el apartado 2.2.1.
- SumSquareErrorsFunctor: Clase necesaria para la definición de la función a optimizar, utilizada en el proceso de calibración.
- FileDialogs: Clase para gestionar los archivos de la aplicación, abriendo archivos desde el explorador de Windows y exportando los datos grabados a archivos de texto para poder ser leídos.
- Image: Clase para cargar imágenes e iconos para ser desplegados en la interfaz gráfica, ya sea cargando imágenes en archivos o las imágenes completas generadas por las cámaras.
- Logger: Clase encargada de la gestión de los mensajes desplegados en la ventana de mensajes, dividiendo los mensajes en cuatro categorías: Fallo, Advertencia, Información y Éxito.
- Timer: Clase que implementa la forma de medir los tiempos de computación de un conjunto de instrucciones dadas.
- ThreadSharedBuffer: Clase escrita en forma de plantilla en la que se implementa el contenedor del tipo de datos que se desee accesible de forma segura por varios subprocesos a la vez.

2.3.3 Manual de usuario

En este apartado se indican las instrucciones para manejar la aplicación para conectar todos los equipos, calibrar las cámaras, definir la región de interés para los filtros de las imágenes, adquirir y guardar datos y visualizar las señales de los sensores conectados al Compact RIO.

Al abrir la aplicación, se abre una ventana con el aspecto mostrado en la siguiente figura:

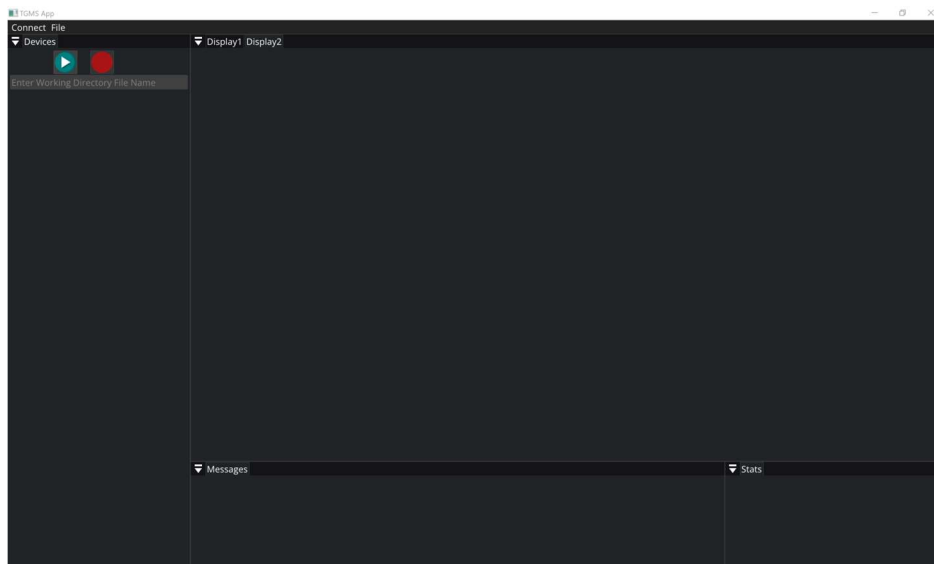


Figura 18. Entorno Gráfico Inicial

En ella, aparece una ventana en la parte inferior llamada “Messages” en la que aparecerán mensajes para saber si las acciones que se realicen son satisfactorias o producen errores. A su derecha aparece una ventana denominada “Stats”, donde aparecerán la medida del tiempo de computación en la adquisición de imágenes. A la izquierda está la ventana “Devices” donde aparecerán los equipos conectados y sus controles. Finalmente, en el centro aparecen dos ventanas llamadas “Display 1” y “Display 2”, en las cuales se mostrarán las imágenes en el proceso de adquisición de datos en tiempo real. Todas estas ventanas pueden ser alteradas de posición arrastrando el nombre de cada una, además de poder modificarse su tamaño.

Encima de todas ellas hay una pequeña barra de herramientas para conectarse a los equipos (Connect) y visualizar y exportar los archivos generados (File):

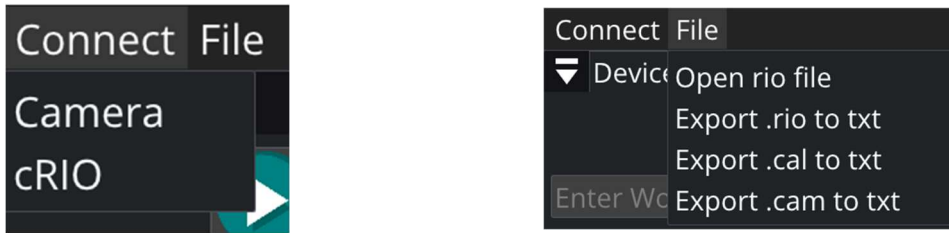


Figura 19. Barra de Herramientas

2.3.3.1 Conexión

Para realizar la conexión con las cámaras, se pulsa en la barra de herramientas Connect → Camera. El programa busca en la red las cámaras conectadas y abre una ventana con las cámaras listadas según el nombre de usuario que se le haya definido. Desplegando dicho nombre, se muestra información sobre el modelo, fabricante y dirección IP de la cámara. A la derecha se selecciona a qué cámara(s) nos deseamos conectar, pulsando finalmente el botón Connect para conectarse a ellas y Close para cerrar la ventana:

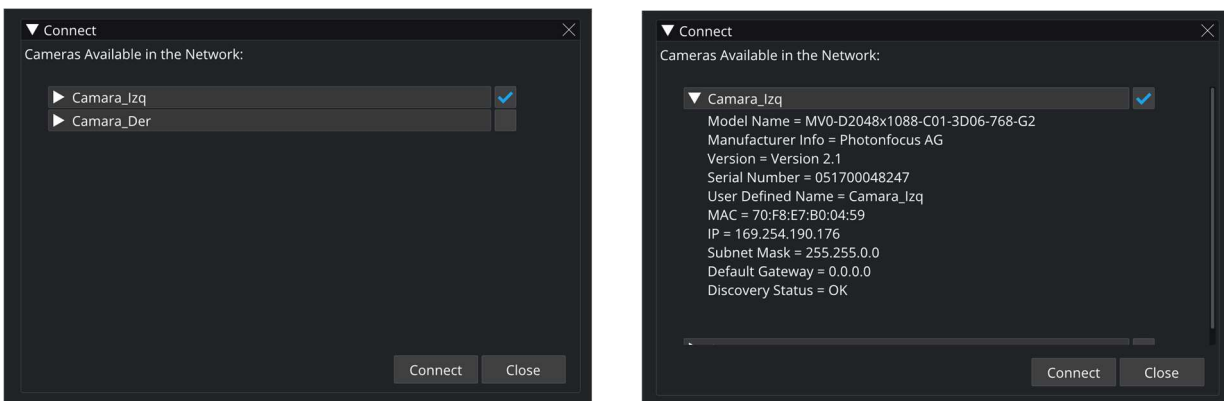


Figura 20. Ventana Conexión con Cámaras

Para establecer la conexión con el Compact RIO, basta seleccionar en la barra de herramientas Connect → cRIO, momento en el que se intenta establecer la conexión con el equipo automáticamente.

2.3.3.2 Controles y configuración

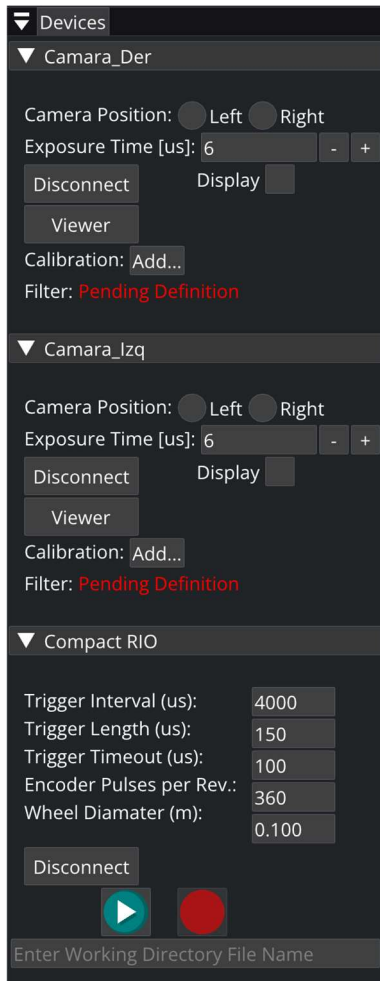


Figura 21. Controles

Los dispositivos conectados aparecen listados en la ventana “Devices”. Las cámaras aparecen con el nombre que el usuario haya definido en ellas. Los desplegados, abiertos por defecto, abren u ocultan los controles del dispositivo.

En primer lugar, se debe definir el nombre de la carpeta en la que se quiera trabajar, carpeta que se creará en el mismo directorio que el ejecutable de la aplicación. Para ello, se debe introducir el nombre en el espacio al final de la ventana y presionar Enter para definirlo.

En cuanto a las cámaras, hay que definir en qué posición se encuentran, es decir, a qué carril apuntan, el derecho o el izquierdo, mirando en la dirección de avance del vehículo. Esta información es necesaria para la calibración.

Seguidamente, se establece el tiempo de exposición, cuyo valor por defecto es el mínimo posible, de 6 microsegundos. Este es el tiempo en el que la cámara está tomando la fotografía. A mayor tiempo de exposición, más luz capta.

Después, aparece un botón para desconectar la cámara, junto a una casilla para activar o desactivar la visualización de imágenes en el proceso de adquisición de datos.

El botón viewer sirve para abrir una nueva ventana mostrada en la Figura 22 para visualizar, una cámara a la vez, la imagen completa en escala de grises y definir el filtro de imagen. Este mismo botón se utiliza para salir de este modo.

Finalmente, aparecen dos indicadores del estado de la cámara. Si no está calibrada, aparece un botón para añadir una calibración, ya sea a partir de un archivo *.cal* o realizando una nueva. Además, indica si se ha definido el filtro o no.

En cuanto al Compact RIO, se deben introducir el tiempo entre fotogramas en microsegundos, definiendo la frecuencia de adquisición de las cámaras, el tiempo en el que la señal que recibe el disparador de las cámaras es de valor digital 1 en microsegundos, el tiempo de espera máximo en microsegundos, el número de pulsos por revolución del encóder instalado y el diámetro de la rueda en la que se coloca en metros. Por debajo aparece un botón para desconectarse del equipo.

Por último, se muestran los botones de inicio y parada de la adquisición de datos, que funcionarán únicamente cuando la calibración y el filtro estén definidos en todas las cámaras conectadas.

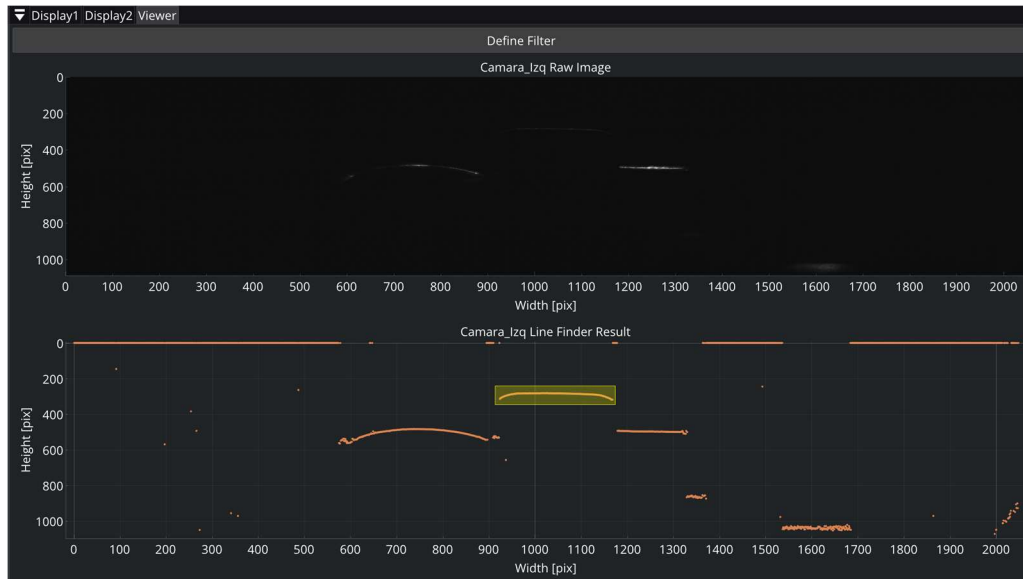


Figura 22. Visualización y Definición del Filtro

En cuanto al modo de visualización, se muestran en la gráfica superior la imagen real en escala de grises y en la gráfica inferior el resultado del algoritmo interno de detección de líneas de la cámara. En esta gráfica aparece un rectángulo amarillo que se debe mover y escalar con el ratón hasta contener en él el perfil del carril únicamente, y pulsando en el botón en la parte superior de la ventana se define este rectángulo como la región de interés en el instante inicial. Además, en la gráfica inferior se puede mover la imagen arrastrándola con el ratón, y se puede hacer zoom con la rueda del ratón.

2.3.3.3 Calibración de una cámara

Para calibrar una cámara se puede optar por cargarla desde un archivo *.cal* generado previamente desde el explorador de archivos que se abre automáticamente, o iniciar una calibración nueva. En este último caso, se abre una ventana nueva, mostrada a continuación:

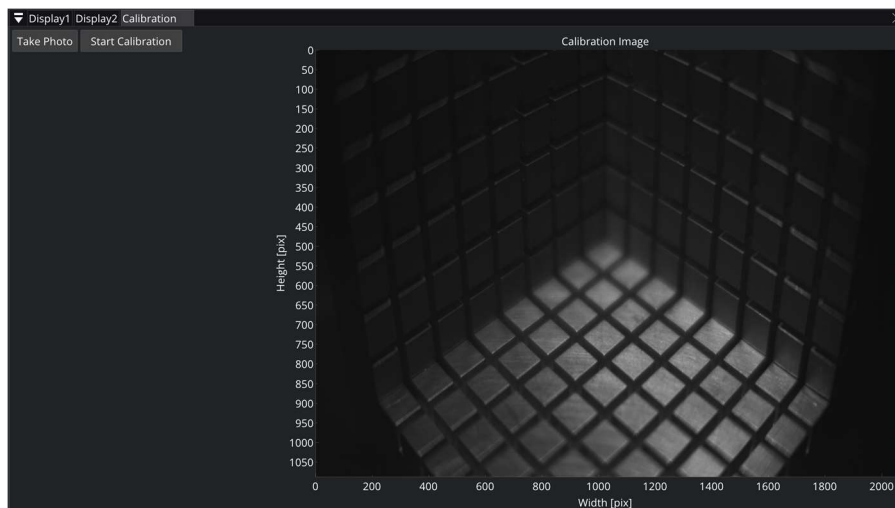


Figura 23. Ventana de Calibración

Primero se debe tomar una fotografía del calibrador presionando el botón Take Photo. La imagen se despliega en la gráfica de la derecha. Se pueden tomar tantas imágenes como sean necesarias hasta obtener una imagen satisfactoria. Después se debe pulsar el botón Start Calibration para comenzar el proceso.

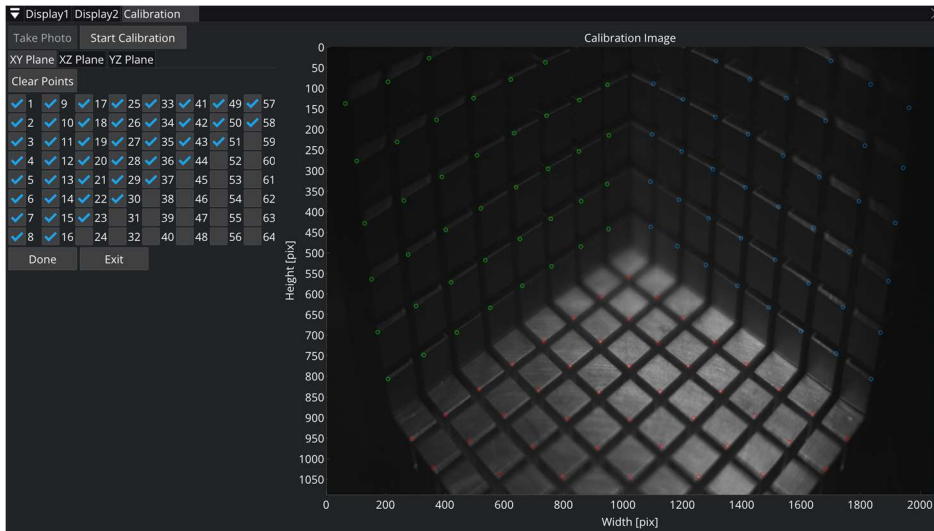


Figura 24. Selección de Puntos de Calibración

En el proceso de calibración, se trabaja por planos independientemente, seleccionando la pestaña adecuada. Los puntos en la imagen se seleccionan directamente en ella pulsando sobre la gráfica en la posición en la que se encuentren los vértices de los cuadrados del calibrador. Sólo los vértices de la diagonal más alejada del centro del calibrador son válidos, como muestra la Figura 24. La selección en cada plano se debe realizar en un orden determinado. Los vértices se enumeran de 1 a 64 como se muestra a continuación. Además, hay que seleccionar qué puntos se han seleccionado cuáles no en la tabla de casillas de la izquierda, bien porque se salen de la imagen o porque no sea posible apreciarlos.

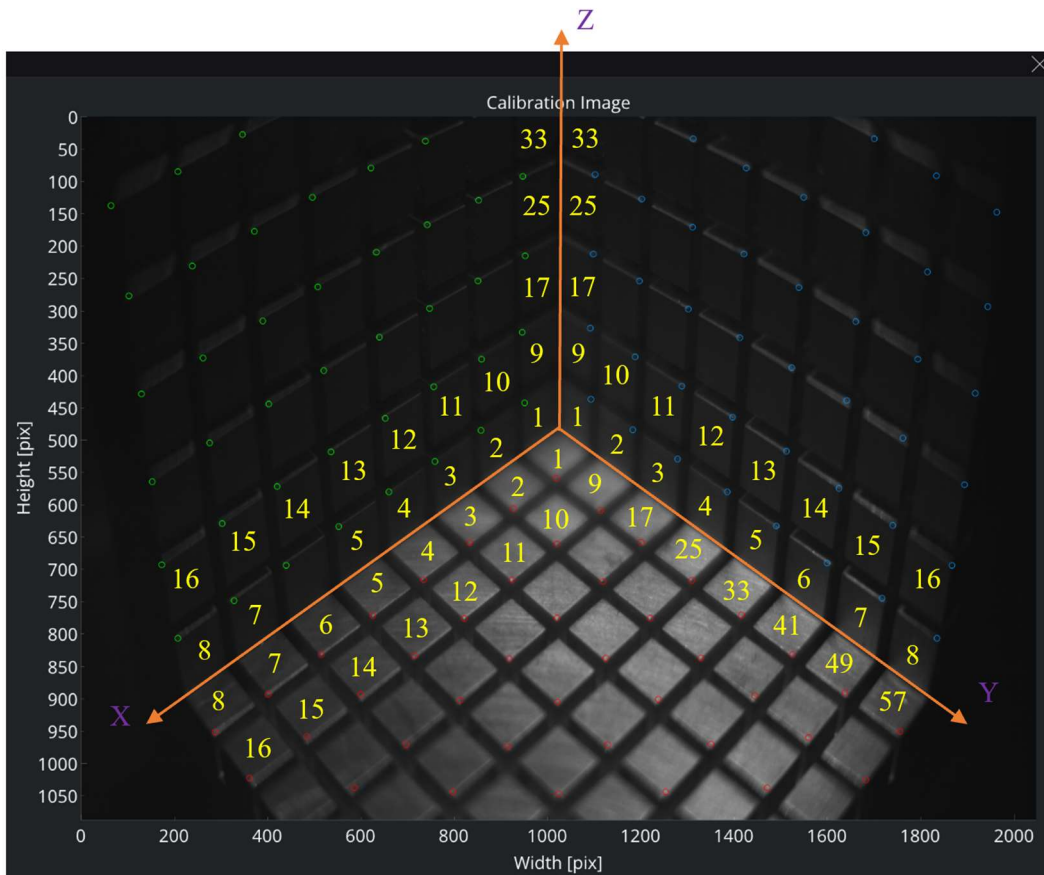


Figura 25. Orden Puntos Calibración

Una vez definidos todos los puntos, se pulsa el botón Done que inicia el algoritmo de calibración. El botón Clear Points permiten borrar la definición de los planos de todos los puntos del plano en el que se trabaje. El

botón Exit cierra la ventana de calibración.

Al finalizar el cálculo, se mostrará un mensaje de si ha habido algún problema a la hora de realizar la optimización, y se abre una nueva ventana que muestra la comparación entre los puntos seleccionados en la imagen y los puntos resultantes de utilizar la matriz de proyección de la cámara calculada:

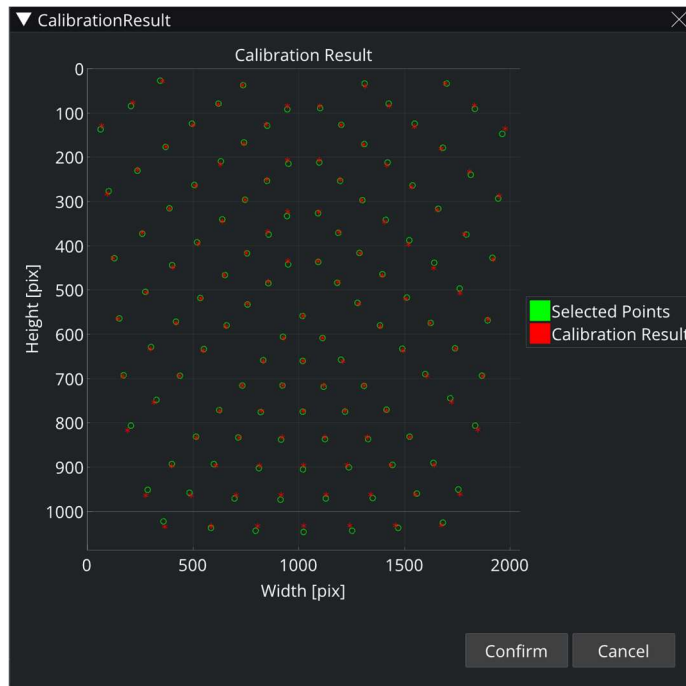


Figura 26. Resultados de la Calibración

Si los resultados son correctos, se confirma la calibración con el botón Confirm, momento en el que se guardan los resultados en un archivo `.cal` utilizando el nombre identificativo de la cámara dentro de una carpeta generada en el directorio de trabajo llamada Calibration. En caso contrario, se cancela en pulsando el botón Cancel y se puede proceder a iniciar el proceso nuevamente.

2.3.3.4 Adquisición de datos y archivos generados

Una vez definida la calibración y los filtros de las cámaras, activando uno de los displays, se debe llegar a una situación como la siguiente:

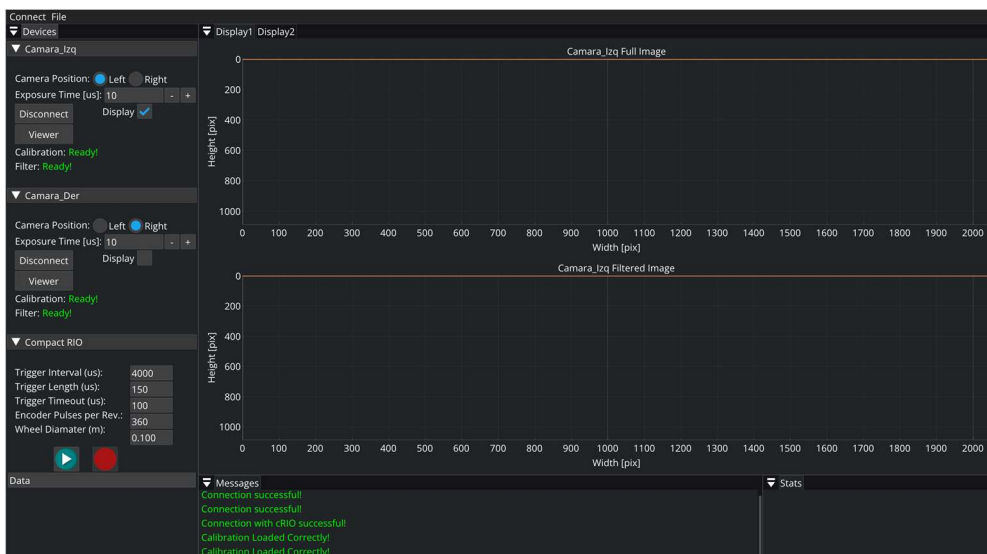


Figura 27. Estado Antes de Adquisición de Datos

Una vez alcanzada esta situación, se puede empezar la adquisición. En el display aparece en la gráfica superior la imagen sin filtrar resultado del algoritmo de la cámara y una gráfica inferior con la imagen filtrada. Además, se muestran los tiempos de computación de cada cámara en la ventana “Stats”.

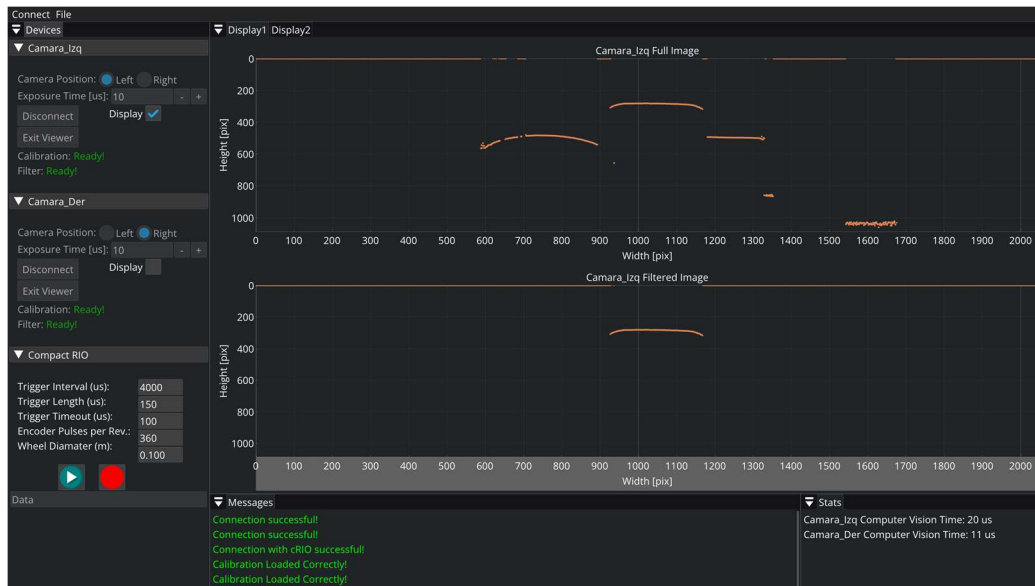


Figura 28. Adquisición de Datos y Visualización

Al parar la adquisición, se generan en el directorio de trabajo dos carpetas adicionales. La primera, llamada Camera Results, contiene archivos *.cam* con los resultados de la visión artificial. Se segunda, de nombre cRIO Measurements contiene un archivo *.rio* con las medidas de los sensores incorporados en el Compact RIO.

Utilizando la barra de herramientas File, se pueden exportar estos archivos y la calibración en archivos de texto que se generan en el mismo directorio del archivo seleccionado con su mismo nombre, cada uno con un encabezado indicando qué datos contiene. Adicionalmente, se pueden visualizar las señales de los sensores del Compact RIO seleccionando File → Open rio file. Al hacerlo, se abre una nueva ventana con un listado de las señales disponibles y una gráfica donde se representan:

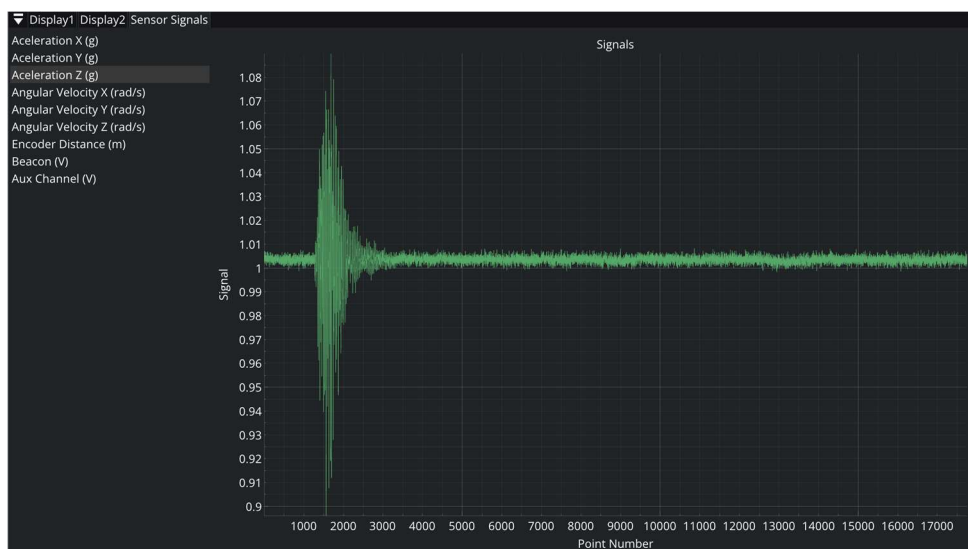


Figura 29. Ventana de Representación de Señales

Se puede elegir entre las tres componentes de la aceleración medida en la IMU, las tres componentes de la velocidad angular medida en la IMU, la distancia recorrida según las medidas del encóder, la señal del detector magnético y la señal de un canal auxiliar existente en caso de necesitar incorporar algún sensor adicional en el sistema.

2.4. Validación y resultados experimentales

Por último, en esta sección se presenta la validación de los algoritmos de visión artificial descritos y de su implementación en tiempo real. Para poder evaluar la precisión de la visión artificial exclusivamente, se ha utilizado un mecanismo automático de ensayos en el cual las cámaras están en una posición fija, eliminando así los efectos acoplados que existen derivados de la dinámica de un vehículo si se hicieran ensayos con el sistema montado en él.

2.4.1 Mecanismo automático de ensayos

El mecanismo automático de ensayos está compuesto por cuatro actuadores eléctricos lineales cuyo movimiento es controlado para mover dos trozos de carril montados sobre el mecanismo de manera que se reproduce el movimiento de los carriles debido a las irregularidades que se observaría si se estuviese montado en un vehículo que viaja a velocidad constante.

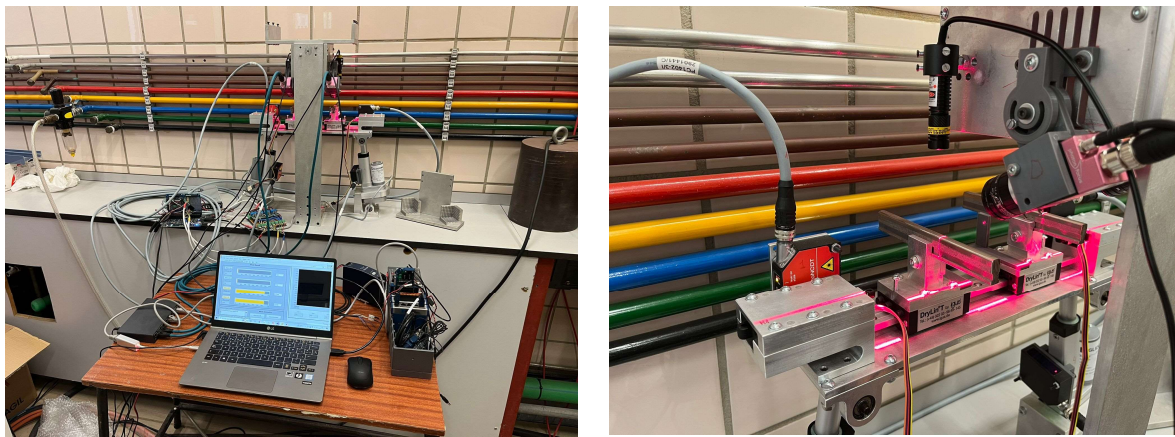


Figura 30. Mecanismo Automático de Ensayos

Para poder contrastar los resultados medidos con las cámaras, el mecanismo también incorpora cuatro láseres para medir distancia. Con ellos se monitoriza la longitud de los actuadores y se puede reconstruir la posición real de los carriles.

Con el fin de controlar los actuadores para que realicen un movimiento determinado y para poder reconstruir las posiciones de los carriles se realiza un modelo cinemático del mecanismo utilizando la teoría de dinámica de sistemas multicuerpo:

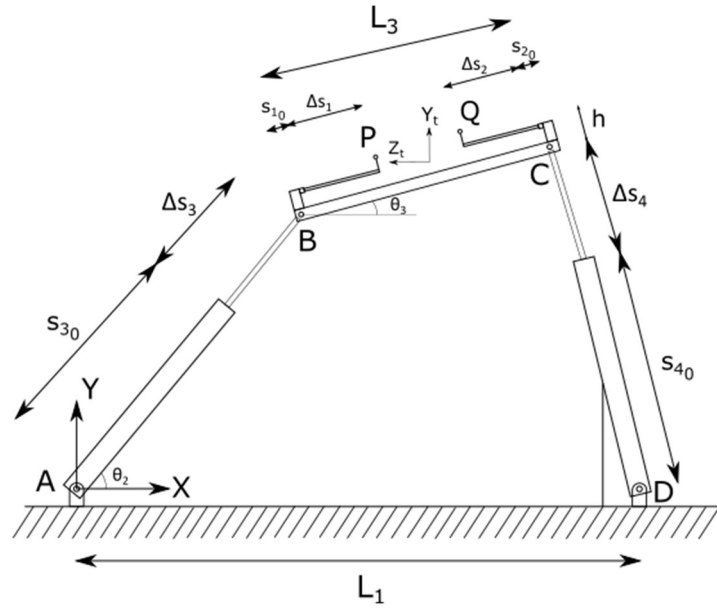


Figura 31. Modelo del Mecanismo

El mecanismo es esencialmente un mecanismo de cuatro barras en donde dos de sus barras tienen longitud variable y se fija la orientación de una de ellas. Es un sistema de cuatro grados de libertad, correspondiente a las longitudes de los cuatro actuadores lineales. El sistema global de coordenadas se sitúa en el punto A y los puntos P y Q representan la posición de los centros de los carriles izquierdo y derecho respectivamente. Además, se define un sistema de referencia adicional correspondiente al sistema local de la vía cuya posición es fija y se elige de forma que los actuadores no se encuentren próximos a un final de carrera al colocar los carriles en su posición nominal sin irregularidades.

Para modelar la cinemática del mecanismo, se utilizan el siguiente vector de coordenadas, que contiene los ángulos de giro de las barras 2 y 3 y los incrementos de longitud de los cuatro actuadores:

$$\mathbf{q} = [\theta_2 \quad \theta_3 \quad \Delta s_1 \quad \Delta s_2 \quad \Delta s_3 \quad \Delta s_4]^T \quad (2.33)$$

Dichas coordenadas están sometidas a dos restricciones derivadas de las ecuaciones de lazo del mecanismo de cuatro barras:

$$\begin{aligned} (s_{30} + \Delta s_3) \cos \theta_2 + L_3 \cos \theta_3 + (s_{40} + \Delta s_4) \cos \theta_4 - L_1 &= 0 \\ (s_{30} + \Delta s_3) \sin \theta_2 + L_3 \sin \theta_3 - (s_{40} + \Delta s_4) \sin \theta_4 &= 0 \end{aligned} \quad (2.34)$$

Donde los parámetros s_{i0} representan la longitud de los actuadores cuando están completamente retraídos. De esta manera, las coordenadas independientes son los cuatro incrementos de longitud de los actuadores, variables impuestas a la hora de accionar el mecanismo, siendo las dos dependientes los dos ángulos de las barras 2 y 3.

Para calcular la longitud de los actuadores en cada instante a partir de unas irregularidades dadas, se parte de la posición de los carriles respecto al sistema local de la vía. Tomando como referencia la Figura 1, la posición de los puntos P y Q es:

$$\begin{aligned} \bar{\mathbf{r}}_P &= \begin{bmatrix} L_r \\ 0 \end{bmatrix} + \begin{bmatrix} y^{lir} \\ z^{lir} \end{bmatrix} \\ \bar{\mathbf{r}}_Q &= \begin{bmatrix} -L_r \\ 0 \end{bmatrix} + \begin{bmatrix} y^{rir} \\ z^{rir} \end{bmatrix} \end{aligned} \quad (2.35)$$

A continuación, se expresan estos vectores en el sistema global de coordenadas cambiando de signo su primera componente ya que se observa que los ejes X global e Y' local apuntan en sentidos opuestos. Finalmente, para obtener la posición global, se suma a ambos vectores la posición del origen de coordenadas

del sistema de referencia local de la vía:

$$\begin{aligned}\mathbf{r}_P &= \begin{bmatrix} -L_r \\ 0 \end{bmatrix} + \begin{bmatrix} -y^{lir} \\ z^{lir} \end{bmatrix} + \begin{bmatrix} u_x^{O^t} \\ u_y^{O^t} \end{bmatrix} \\ \mathbf{r}_Q &= \begin{bmatrix} L_r \\ 0 \end{bmatrix} + \begin{bmatrix} -y^{rir} \\ z^{rir} \end{bmatrix} + \begin{bmatrix} u_x^{O^t} \\ u_y^{O^t} \end{bmatrix}\end{aligned}\quad (2.36)$$

Por último, se expresa la posición de estos puntos en función de las coordenadas generalizadas del sistema:

$$\begin{aligned}\mathbf{r}_P &= \begin{bmatrix} (s_{30} + \Delta s_3) \cos \theta_2 \\ (s_{30} + \Delta s_3) \sin \theta_2 \end{bmatrix} + \begin{bmatrix} \cos \theta_2 & -\sin \theta_2 \\ \sin \theta_2 & \cos \theta_2 \end{bmatrix} \begin{bmatrix} s_{10} + \Delta s_1 \\ h \end{bmatrix} \\ \mathbf{r}_Q &= \begin{bmatrix} (s_{30} + \Delta s_3) \cos \theta_2 \\ (s_{30} + \Delta s_3) \sin \theta_2 \end{bmatrix} + \begin{bmatrix} \cos \theta_2 & -\sin \theta_2 \\ \sin \theta_2 & \cos \theta_2 \end{bmatrix} \begin{bmatrix} L_3 - (s_{20} + \Delta s_2) \\ h \end{bmatrix}\end{aligned}\quad (2.37)$$

Igualando las ecuaciones (2.37) con las ecuaciones (2.36) y añadiendo las ecuaciones de restricción (2.34) se ensambla un sistema de 6 ecuaciones para determinar las 6 coordenadas generalizadas a partir de las irregularidades de la vía en un instante dado. Los incrementos de longitud obtenidos serán los que habrá que imponer en los actuadores para conseguir la posición deseada.

Como los actuadores no consiguen llegar exactamente a la posición deseada, se reconstruye dicha posición a partir de la medida de los láseres de medida de distancia. Estos sensores proporcionan un incremento de posición y no la distancia absoluta. La posición de referencia o “cero” de los láseres corresponde a la posición en la que todos los actuadores están completamente retraídos. Por tanto, gracias a las coordenadas que se han elegido, los láseres miden directamente los incrementos de longitud de los actuadores dados por las coordenadas Δs_i .

Así, la reconstrucción de la posición de los carriles se reduce a resolver las ecuaciones de restricción para obtener las coordenadas independientes del sistema y calcular la posición con las ecuaciones (2.37).

Finalmente, para poder comparar las medidas obtenidas con los láseres y con las cámaras es necesario expresar ambas en el mismo sistema de referencia, ya que la visión artificial proporciona resultados respecto al sistema de referencia local del TGMS. Su posición es conocida y es paralelo al sistema local de la vía. De esta manera, se aplica una transformación equivalente a la realizada para el sistema de la vía a los resultados medidos con el sistema desarrollado.

2.4.2 Resultados

A continuación, se representan los resultados obtenidos en un ensayo experimental realizado en el mecanismo automático de ensayos. Se realiza la comparación de las irregularidades de ancho de vía y nivelación, ya que para su cálculo se incluyen los resultados de ambas cámaras conjuntamente. El cálculo se realiza despejando las irregularidades de las expresiones (2.36) y aplicando las definiciones de estas irregularidades.

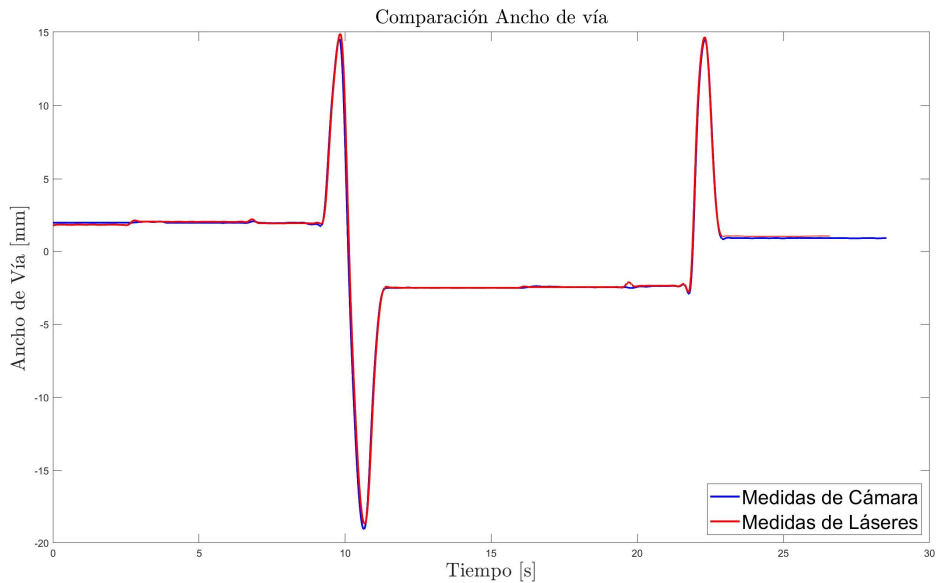


Figura 32. Comparación del Ancho de Vía

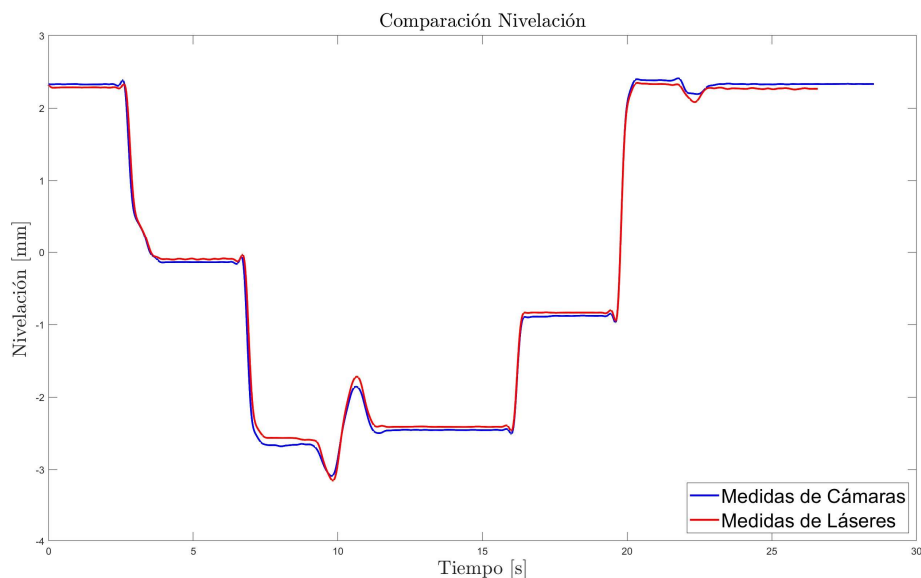


Figura 33. Comparación de la Nivelación

Se observa un gran grado de acuerdo entre las medidas de los láseres y los resultados de la visión artificial. Además, se han impuesto irregularidades muy grandes para abarcar el tamaño completo de la imagen, de forma que se comprueba que la calibración es válida sea cual sea la posición del perfil del carril en la imagen. Además, de esta manera se verifica el buen funcionamiento del filtrado de las imágenes ya que la región de interés de la imagen es capaz de seguir el movimiento del carril por muy grande que este sea.

Por otra parte, se ha medido el tiempo de computación necesario para realizar el procesado de las imágenes para verificar que los algoritmos trabajan en tiempo real. Se ha medido a una frecuencia de 250 Hz, agrupando las imágenes en paquetes de 10, por lo que el PC recibe información de las cámaras a una frecuencia de 25 Hz.

Para permanecer en tiempo real, se deben procesar las 10 imágenes en menos de 40 ms. Se ha ejecutado la aplicación en un ordenador con procesador Intel(R) Core(TM) i7-9750H de 2.60 GHz, en donde el tiempo de procesamiento no supera los 70 microsegundos por imagen. Esto supone un total de 0.7 ms, menos del 2% del tiempo disponible para procesar cada paquete de 10 imágenes, garantizando el funcionamiento en tiempo real de la visión artificial.

3 MEDICIÓN DE IRREGULARIDADES DE CORTA LONGITUD DE ONDA: CORRUGACIÓN

En este capítulo se describirá el sistema diseñado para la medición de la corrugación tanto en la vía a escala como en una vía real. En primer lugar, se presentará el sistema diseñado para esta aplicación, incluyendo los sensores utilizados y la estructura fabricada en donde se han instalado los sensores. A continuación, se describirá el proceso de cálculo de la evolución de la corrugación a lo largo del carril. Seguidamente, se mostrará la aplicación de escritorio desarrollada para el manejo de los sensores, la adquisición de datos y el cálculo y representación de resultados, para finalizar mostrando los resultados obtenidos midiendo en la vía a escala.

3.1. Diseño del sistema

El sistema de medida utiliza dos perfilómetros láser de la marca Micro-Epsilon, modelo scanCONTROL 3002-100B/L. Este dispositivo incorpora un proyector de línea láser azul y un sensor fotosensible o cámara apuntando en una dirección determinada con un filtro de luz para detectar únicamente la longitud de onda de la luz del láser. Este sensor adquiere imágenes a una frecuencia determinada o según una señal externa que active el disparador.



Figura 34. Perfilómetro Láser

Al estar la posición relativa del láser y de la cámara fijada, el propio sensor tiene internamente una calibración realizada por el fabricante capaz de pasar los puntos de coordenadas de imagen a coordenadas en el plano del láser. Además, cuenta con un algoritmo de detección de líneas, por lo que la información que transmite es un conjunto de 1024 puntos con las coordenadas X y Z reales en milímetros de los puntos en el plano definido por el láser, tomando como sistema de referencia la base del sensor, como muestra la siguiente figura:

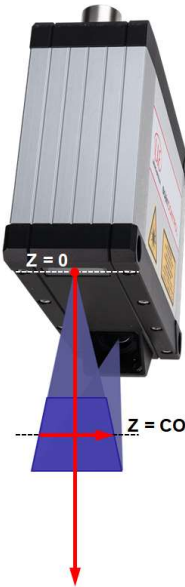


Figura 35. Sistema de Referencia del Sensor

Además, el trapecio sombreado en color más oscuro delimita la zona de medida fuera de la cual el sensor no capta al láser. Esta circunstancia hace que, para medir el perfil de un carril sea necesario colocar el sensor encima de la vía a una altura determinada.

La información se transmite a través de un cable Ethernet, uno por sensor. Para poder establecer la conexión con los dos perfilómetros a la vez con el PC del usuario, es necesario utilizar un Switch donde se conecten los dos sensores y el ordenador.



Figura 36. Switch para Conexión de Perfilómetros y PC

El dispositivo utilizado es un switch de la marca TP-Link modelo D-Link GO-SW-5G, con 5 puertos Gigabit a los que se conectará con cable Ethernet cada uno de los perfilómetros y el ordenador del usuario.

Adicionalmente, para medir sucesivamente los perfiles de un carril, es necesario mover los sensores a lo largo de la vía de forma estable sin que sufran movimientos verticales derivados de vibraciones u otros fenómenos. Para solventar este problema, se ha diseñado una estructura de 2 metros de longitud en la que se han instalado los perfilómetros láser.

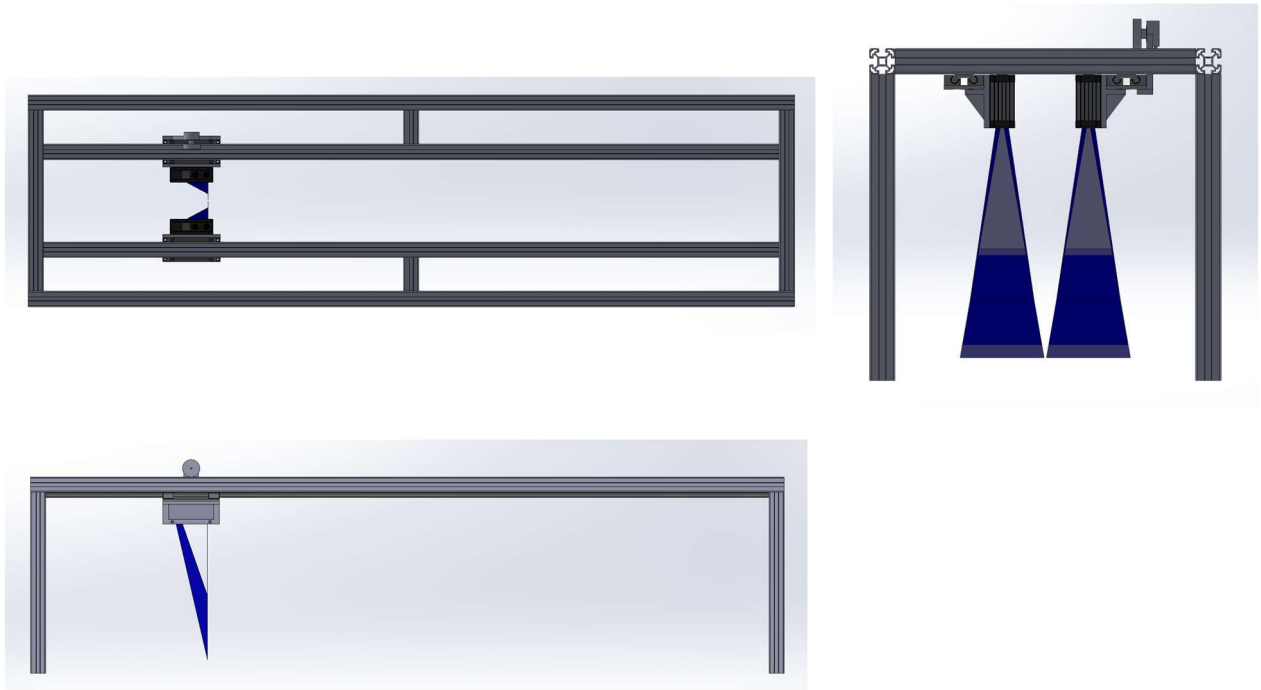


Figura 37. Alzado, Planta y Perfil de la Estructura



Figura 38. Vista Tridimensional de la Estructura

Está compuesta de una serie de perfiles de aluminio ensamblables mediante tornillería con dos guías sobre los que deslizan unos patines a los que se les colocan los sensores. De esta forma, se consigue deslizar suavemente los sensores sobre la vía. Además, en los cuatro pilares se enroscan unas patas roscables que permiten variar la altura de los puntos de apoyo de la estructura. Con esto, se garantiza que el sensor esté a la altura adecuada para poder medir y también se consigue nivelar la estructura a la hora de medir en una vía real en un terreno sinuoso.



Figura 39. Pata Roscable

Finalmente, como los sensores se deslizarán manualmente a lo largo de la estructura, no es posible obtener una frecuencia de adquisición espacial constante si se adquieren imágenes con frecuencia de adquisición constante. Para conseguir que cada fotograma esté espaciado una longitud determinada en el carril, se incluye un encóder rotatorio en la estructura, montado sobre uno de los patines. Este encóder estará acoplado a una rueda que rodará sobre un perfil de la estructura.

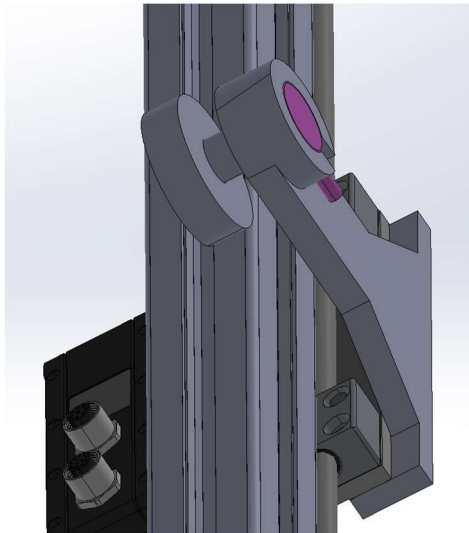


Figura 40. Montaje Encóder Rotatorio

El encóder utilizado es el mismo que el incluido en el sistema de visión artificial del capítulo anterior, con 360 pulsos por revolución y dos canales digitales. Para adquirir estas señales, realizar el recuento de pulsos y enviar una señal digital a los perfilómetros para activar el disparador del sensor cada cierto número de pulsos, se ha utilizado el microcontrolador Arduino UNO.



Figura 41. Arduino UNO

Este microcontrolador sirve para alimentar el encóder, adquirir y procesar su señal y enviar señales digitales para la adquisición de imágenes a una frecuencia espacial constante.

En definitiva, el sistema completo se compone de dos perfilómetros láser y un encóder rotatorio instalados en una estructura de 2 metros de longitud conectados a un Arduino y un Switch que se conectan a su vez al PC del usuario. El esquema completo es el siguiente:

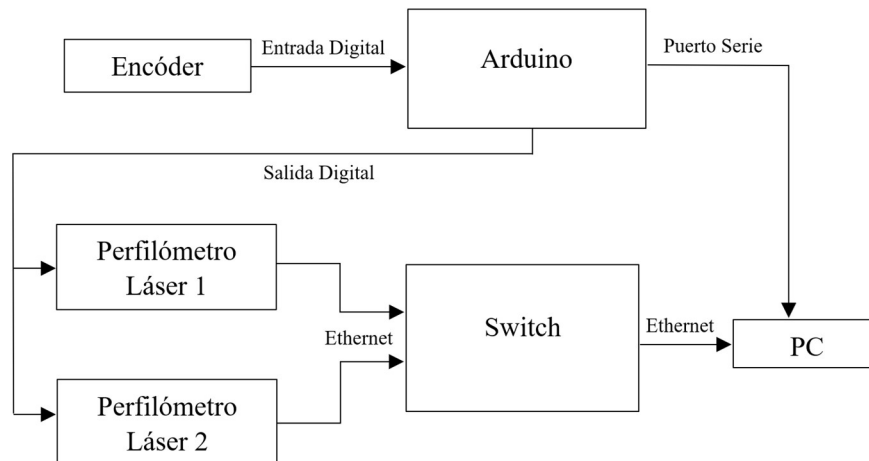


Figura 42. Esquema Sistema para la Medición de la Corrugación

3.2. Procesado de datos

Como se ha explicado en la sección anterior, una imagen recibida de un perfilómetro láser consiste en un conjunto de 1024 puntos con coordenadas X y Z en milímetros expresadas en el sistema local del sensor. Para calcular la evolución de la corrugación a lo largo del carril, se van a reutilizar los algoritmos de visión artificial usados en el capítulo anterior.

Al igual que con las cámaras del sistema anterior, las imágenes contienen el perfil del carril de otros elementos de la vía que no son de interés:

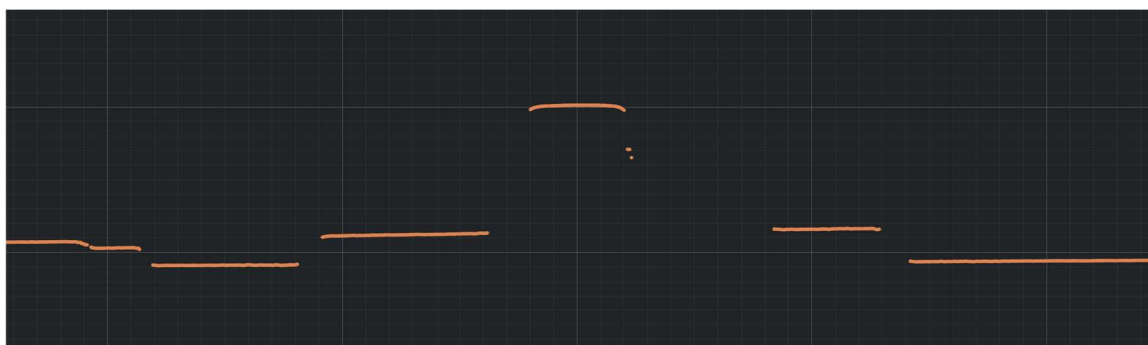


Figura 43. Imagen de la Vía a Escala

Por ello, se utiliza el mismo algoritmo de filtrado de imágenes descrito en el apartado 2.2.2.1, definiendo una región de interés inicial que capture únicamente el perfil del carril y vaya siguiéndolo conforme se vaya moviendo en la imagen.

Una vez filtrados los puntos innecesarios, se calcula en qué posición se encuentra el carril respecto al láser calculando el centro de gravedad de los puntos del carril. En este caso, se trabaja directamente con las coordenadas reales en milímetros, no siendo necesario utilizar la calibración para calcular estas coordenadas en función de la proyección de la imagen. Este paso lo realiza internamente el sensor de forma automática si se configura correctamente.

Con todo esto, se tienen un conjunto de secciones del carril equiespaciadas entre sí una distancia conocida y cuya posición respecto al perfilómetro láser es conocida. Para cuantificar la corrugación, se debe evaluar la evolución de la altura de un punto del perfil del carril en función de la coordenada longitudinal o longitud de arco (s) del carril. Además, para comparar si la corrugación varía en función de la línea elegida, se representará esta evolución en distintas posiciones del perfil del carril, a una distancia transversal d de su centro de gravedad:

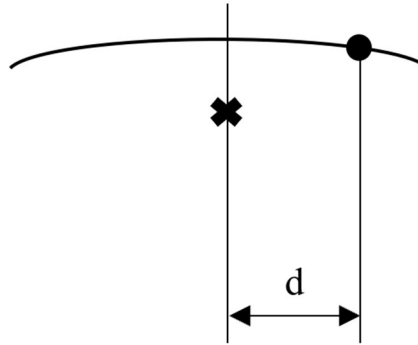


Figura 44. Definición de la Distancia de la Línea de Corrugación

Como el perfil del carril está discretizado en una serie de puntos obtenidos con un perfilómetro, es posible que esta distancia no coincida con la coordenada transversal de ninguno de estos puntos. Para solucionar este inconveniente, se ha utilizado el punto de coordenada transversal más cercana a la deseada.

En general, la altura de un punto del perfil puede variar por efecto de la corrugación del carril, pero también por efecto de las irregularidades de larga longitud de onda o simplemente por no estar la estructura de los láseres perfectamente paralela al carril. Además, la cuantificación de la corrugación se realiza a través de su longitud de onda y amplitud, no de la variación de la altura del perfil del carril.

Es por todo esto que resulta interesante pasar los resultados al dominio de la frecuencia (espacial en este caso) en donde se pueden distinguir los efectos de las irregularidades de larga longitud de onda en la altura de los carriles y de corta longitud de onda, además de cuantificar la severidad del defecto mediante su amplitud.

Para realizar esta transformación, en primer lugar se debe restar a las medidas obtenidas su media ya que esta va a ser distinta de cero y muy grande en comparación con las variaciones de altura. A continuación, calcula la Transformada de Fourier de esta señal (x) empleando el algoritmo de la FFT (*Fast Fourier Transform*), que proporciona el espectro de la señal expresado mediante los coeficientes complejos:

$$c_k = \frac{1}{N} \sum_{j=0}^{N-1} x_j e^{i2\pi \frac{kj}{N}} \quad (3.1)$$

Donde N es el número de puntos en la señal. De todos ellos, sólo la primera mitad contiene información, ya que la segunda mitad resulta ser simétrica a la primera. El valor absoluto de estos coeficientes representa la mitad de la amplitud del armónico asociado, por lo que se calculará el doble del valor absoluto.

Las frecuencias (espaciales) asociadas a cada uno de los coeficientes calculados dependen de la longitud total en la que se ha medido (L), empezando en 0 y estando equiespaciadas en una cantidad de valor $1/L$. Como se quiere obtener la longitud de onda de la corrugación, basta representar las amplitudes frente a la inversa de las frecuencias.

3.3. Aplicación de escritorio

Para conectar, configurar y recibir datos de los sensores y procesar los datos recibidos para calcular la corrugación de los carriles, se ha desarrollado una aplicación de escritorio utilizando el lenguaje de programación C++ empleando muchos de los recursos utilizados en la aplicación descrita en el capítulo anterior.

Si bien previamente se buscaba una gran eficiencia computacional para realizar el procesado de datos en tiempo real, en esta aplicación no es necesario, ya que se necesita la medida completa de la corrugación para poder transformarla al dominio de la frecuencia. Por tanto, en esta aplicación se hará la adquisición de datos y se realizará su procesado a posteriori.

En cuanto a la arquitectura del programa, se va a emplear la misma filosofía, realizando la adquisición de datos en dos subprocesos independientes al bucle de ejecución principal que genera la interfaz gráfica, uno para cada sensor. La comunicación con el Arduino, novedad en esta aplicación, se explicará con detalle en el apartado 3.1.2.

3.1.1 Recursos utilizados

Para la programación, depuración y compilación del código se ha utilizado el entorno integrado de desarrollo Visual Studio 2022. Para programar y configurar el microcontrolador Arduino, se ha utilizado el entorno integrado de desarrollo de Arduino. Adicionalmente, se han utilizado librerías desarrolladas por terceros.

Para el programa del microcontrolador, se ha empleado la librería `Encoder.h`, disponible para su descarga en el entorno de desarrollo. Esta librería se encarga automáticamente de recibir las señales de los dos canales del encóder rotatorio y realizar el recuento de pulsos. Esta librería realiza este recuento a partir de los flancos de subida y de bajada de las señales digitales de ambos canales, por lo que se multiplica por cuatro el número de pulsos por revolución del encóder, siendo en este caso un total de 1440 pulsos.

Para la aplicación de escritorio, se han reutilizado muchos de los recursos empleados en la programación de la aplicación para la medición de las irregularidades de larga longitud de onda, que se listan a continuación:

- Eigen 3: Librería matemática utilizada para el cálculo de la transformada de Fourier utilizando la FFT.
- ImGui: Librería para el desarrollo de la interfaz gráfica de la aplicación.
- ImPlot: Módulo adicional de ImGui para la representación de gráficas.
- GLFW (Graphics Library Framework): Librería utilizada para generar de forma sencilla una ventana para la aplicación y utilizada por ImGui.
- OpenGL (Open Graphics Library): Librería para el desarrollo de aplicaciones gráficas utilizada por ImGui.
- Glew (OpenGL Extension Wrangler Library): Librería utilizada para cargar de forma sencilla las funciones de OpenGL contenidas en el controlador de la tarjeta gráfica del PC.
- STB_Image: Librería para la lectura y escritura de imágenes, utilizada para cargar iconos en la interfaz gráfica de la aplicación.
- scanControl SDK (Software Development Kit): Librerías proporcionadas por el fabricante de los perfilómetros para su configuración y comunicación con el PC.
- Librería estándar C++20: Librería estándar de C++ en su versión de 2020.
- Windows SDK: Librería para el desarrollo de aplicaciones para el sistema operativo Windows.

3.1.2 Comunicación con Arduino

La comunicación entre el Arduino y el PC se realiza a través de un puerto serie, conectándose la placa en un puerto USB del ordenador. El microcontrolador debe recibir del ordenador la señal de empezar o terminar la lectura de las señales del encóder, la señal de reestablecer el contador a cero, el valor de dos parámetros correspondientes al número de pulsos por cada señal a emitir al disparados de los láseres para controlar la frecuencia espacial de adquisición de imágenes y el tiempo en la que esta señal digital se mantiene en valor alto. Además, debe enviar al ordenador el recuento de pulsos en un instante dado y el número de señales enviadas al disparador para comprobar que el número de imágenes recibidas coincide.

El programa en el microcontrolador en un conjunto de instrucciones que se ejecuta continuamente en bucle, al igual que la aplicación principal en el ordenador. Sin embargo, estos bucles no están sincronizados de ninguna manera y se ejecutan a frecuencias diferentes. Ante esta situación, se utilizan en el Arduino los llamados eventos de puerto serie.

Un evento es un hecho que ocurre en un sistema que se comunica al programa para que este actúe en consecuencia. Por ejemplo, presionar una tecla en el teclado genera un evento que se envía a una aplicación y esta lo procesa realizando o no una acción concreta. El evento de puerto serie en el Arduino ocurre cuando se recibe un byte de información a través del puerto serie. Este hecho se notifica al programa que se ejecuta en el microcontrolador y actuará en consecuencia.

De esta forma, desde el lado del ordenador se envía un byte de datos, lo que corresponde en C++ a una variable de tipo carácter (`char`), según el siguiente código:

- Carácter 'a': Señaliza al Arduino que comience la lectura de las señales del encóder.
- Carácter 'b': Señaliza al Arduino que termine la lectura de las señales del encóder.
- Carácter 'c': Señaliza al Arduino que reestablezca el contador de pulsos a cero.
- Carácter 'd': Señaliza al Arduino que lea por el puerto serie y cargue el valor de los dos parámetros variables del programa.
- Carácter 'e': Señaliza al Arduino que envíe el valor de los contadores de pulsos y de señales enviadas a los perfilómetros por el puerto serie.

A través de este código, es posible controlar el Arduino desde el PC, y este es capaz de mostrar el valor de los contadores de pulsos en los instantes necesarios según la frecuencia de ejecución del programa principal.

3.1.3 Clases implementadas

El código de la aplicación de escritorio se organiza en una serie de clases, algunas de las cuales ha sido reutilizadas de la aplicación del capítulo anterior:

- `Application`: Clase principal del programa encargada de la inicialización, ejecución del bucle principal y finalización de la aplicación. En la inicialización, se configuran las librerías utilizadas en el programa, se crea una ventana en el PC y se inician las ventanas que componen la interfaz gráfica de la aplicación. El bucle principal actualiza todas estas ventanas y recibe e interpreta las entradas de ratón y teclado. Finalmente, en la finalización se libera toda la memoria utilizada y se cierra la ventana del ordenador. Esta clase tiene una estructura de "*Singleton*", es decir, sólo se puede crear una única instancia de esta clase a lo largo de todo el programa, pues el programa gestiona una única aplicación.
- `DockspaceWindow`: Clase para generar un espacio para el acoplamiento de ventanas dentro de la ventana principal. Además, genera la barra de herramientas principal con opciones para conectarse a los equipos y gestionar los archivos de datos.
- `ImGuiWindow`: Clase base para todas las clases asociadas a una ventana de la aplicación. Proporciona métodos básicos para abrir y cerrar las ventanas y para su actualización.
- `ConnectWindow`: Clase derivada en la que se implementa la ventana para seleccionar los perfilómetros láser disponibles a los que conectarse.
- `DevicesWindow`: Clase derivada en la que se muestran todos los perfilómetros conectados y se

controla la comunicación con ellos configurándolos, iniciando y parando la adquisición de datos e indicando el directorio de trabajo donde se guardarán los archivos de datos.

- `MessagesWindow`: Clase derivada en la que se muestran mensajes al usuario para saber si las acciones ejecutadas han sido satisfactorias o han producido errores.
- `DisplayWindow`: Clase derivada en la que se implementa una ventana para visualizar las imágenes en tiempo real en el proceso de adquisición de datos.
- `ArduinoWindow`: Clase derivada en la que se implementa la interfaz para la comunicación con el Arduino.
- `ParametersWindow`: Clase derivada en la que se implementa una ventana para editar los parámetros de configuración de los perfilómetros.
- `FileViewerWindow`: Clase derivada en la que se implementa una ventana para la visualización a posteriori de las imágenes grabadas en un ensayo.
- `FilterWindow`: Clase derivada en la que se implementa una ventana para la definición de la región de interés inicial necesaria para el filtrado y procesado de las imágenes.
- `ResultsWindow`: Clase derivada en la que se implementa la visualización de los resultados de la corrugación calculada.
- `Profiler`: Clase encargada de la conexión y desconexión con los perfilómetros laser, su configuración y la adquisición de imágenes.
- `Arduino`: Clase encargada de establecer la conexión con el Arduino y comunicarse con él.
- `SerialPort`: Clase en la que se implementa la lectura y escritura de datos a través de un puerto serie del ordenador.
- `FileDialogs`: Clase para gestionar los archivos de la aplicación, abriendo archivos desde el explorador de Windows y exportando los datos grabados a archivos de texto para poder ser leídos.
- `Image`: Clase para cargar imágenes e iconos para ser desplegados en la interfaz gráfica, cargando imágenes desde archivos del ordenador.
- `Logger`: Clase encargada de la gestión de los mensajes desplegados en la ventana de mensajes, dividiendo los mensajes en cuatro categorías: Fallo, Advertencia, Información y Éxito.
- `Timer`: Clase que implementa la forma de medir los tiempos de computación de un conjunto de instrucciones dadas.
- `ThreadSharedBuffer`: Clase escrita en forma de plantilla en la que se implementa el contenedor del tipo de datos que se desee accesible de forma segura por varios subprocesos a la vez.

3.1.4 Manual de usuario

En este apartado se incluyen las instrucciones para el manejo de la aplicación, incluyendo cómo conectarse a los sensores y Arduino, su configuración, la adquisición de datos y qué archivos se generan y cómo visualizar y procesar los datos para el cálculo de la corrugación.

Al abrir la aplicación, se abre una ventana en el ordenador con la siguiente interfaz:

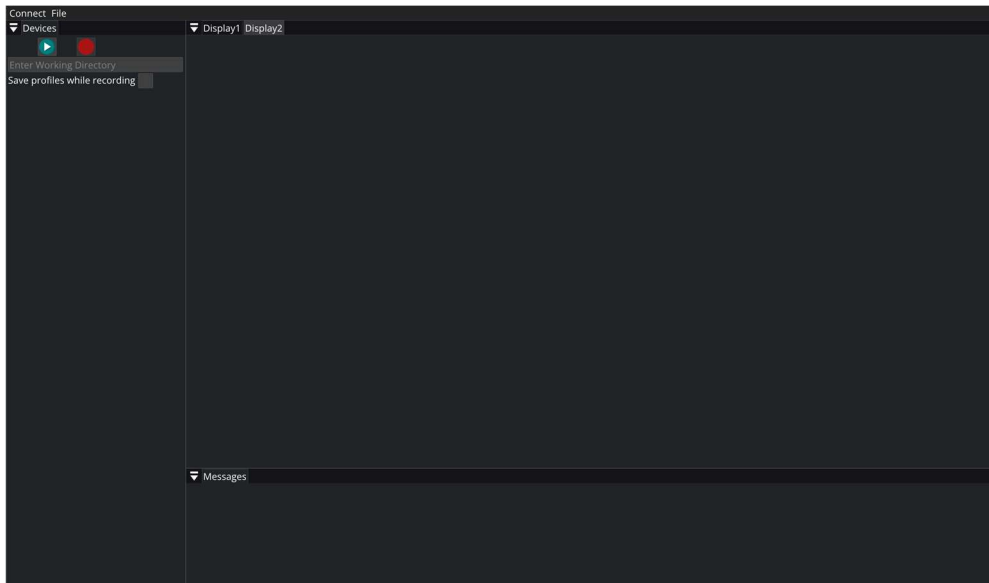


Figura 45. Interfaz Gráfica de la Aplicación

En ella, aparece una ventana en la parte inferior llamada “Messages” en la que aparecerán mensajes para saber si las acciones que se realicen son satisfactorias o producen errores. A la izquierda está la ventana “Devices” donde aparecerán los perfilómetros láser conectados y sus controles. Finalmente, en el centro aparecen dos ventanas llamadas “Display 1” y “Display 2”, en las cuales se mostrarán las imágenes en el proceso de adquisición de datos en tiempo real. Todas estas ventanas pueden ser alteradas de posición arrastrando el nombre de cada una, además de poder modificarse su tamaño.

Encima de todas ellas hay una pequeña barra de herramientas para conectarse a los equipos (Connect) y visualizar, calcular la corrugación y exportar los archivos generados (File):



Figura 46 Barra de Herramientas

3.1.4.1 Conexión

Para establecer la conexión con un perfilómetro láser se selecciona en la barra de herramientas la pestaña Connect → Profiler. Al hacerlo, se abrirá la siguiente ventana, donde aparecen los perfilómetros conectados a los que se puede conectar la aplicación:

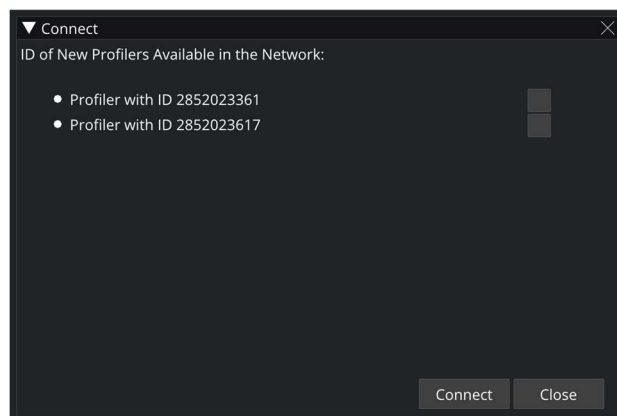


Figura 47. Ventana de Selección de Perfilómetros Láser

Cada sensor se identifica con un número único y para conectarse al dispositivo deseado, se selecciona en el recuadro de la derecha y se pulsa el botón Connect. Finalmente, se cierra la ventana con el botón Close.

Para establecer la conexión con el Arduino, en la barra de herramientas se selecciona Connect → Arduino, apareciendo desplegada una ventana, como se muestra en la Figura 46, en la que se debe introducir el nombre del puerto USB al que se haya conectado el Arduino. Este nombre se compone de las letras “COM” seguidas del número del puerto, por ejemplo, COM3. Para establecer la conexión se debe pulsar Enter tras introducir la información.

3.1.4.2 Configuración y Adquisición de datos

Los perfilómetros conectados aparecen en la ventana “Devices”, y si la conexión con el Arduino es exitosa, aparece encima de ella una nueva ventana llamada “Arduino”.

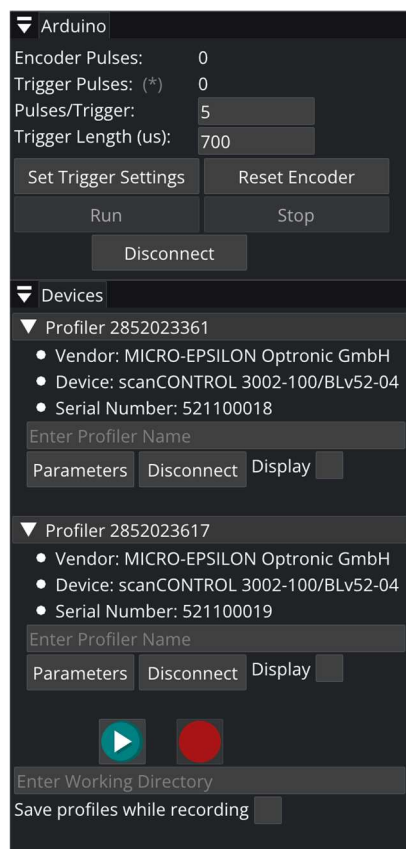


Figura 48. Controles

En la ventana “Arduino” aparecen dos indicadores que muestran en tiempo real el recuento de pulsos del encóder y el número de señales emitidas al disparador de los perfilómetros láser. A continuación, se pueden introducir el valor de los parámetros del programa del Arduino: el número de pulsos por cada señal al disparador y el tiempo en microsegundos que esta señal está en nivel alto. Una vez introducidos, se cargan en el Arduino pulsando el botón Set Trigger Settings. El botón Reset Encoder reestablece los contadores a cero, y los botones Run y Stop inician y paran la adquisición de las señales del encóder por parte del Arduino. Estos botones no se podrán pulsar hasta que no se hayan cargado los parámetros. Finalmente, la desconexión con la placa se realiza con el botón Disconnect.

En la ventana “Devices” aparece la información del fabricante, modelo y número de serie de los perfilómetros conectados. Seguidamente se puede definir un nombre identificativo para el láser, nombre que se utilizará en los archivos de datos obtenidos con el sensor. A continuación, aparecen los botones Parameters para abrir la ventana de configuración de parámetros del sensor y Disconnect para desconectarlo. La casilla Display habilita la visualización de las imágenes al tiempo que se van grabando.

Finalmente, aparecen los botones de iniciación y parada de la adquisición de datos, seguidos de una barra donde se debe introducir el directorio de trabajo expresado a partir del directorio donde se encuentre el ejecutable de la aplicación. La última casilla se debe activar si se quiere guardar los datos en archivos *.img*. Si no se selecciona, se pueden observar las imágenes, pero no se registran.

La visualización de las imágenes aparece en las ventanas “Display 1” y “Display 2” que se mover y poner en paralelo como muestra la siguiente figura:



Figura 49. Visualización de Imágenes

La ventana de parámetros muestra las distintas opciones para configurar cómo se desea trabajar con los perfilómetros:

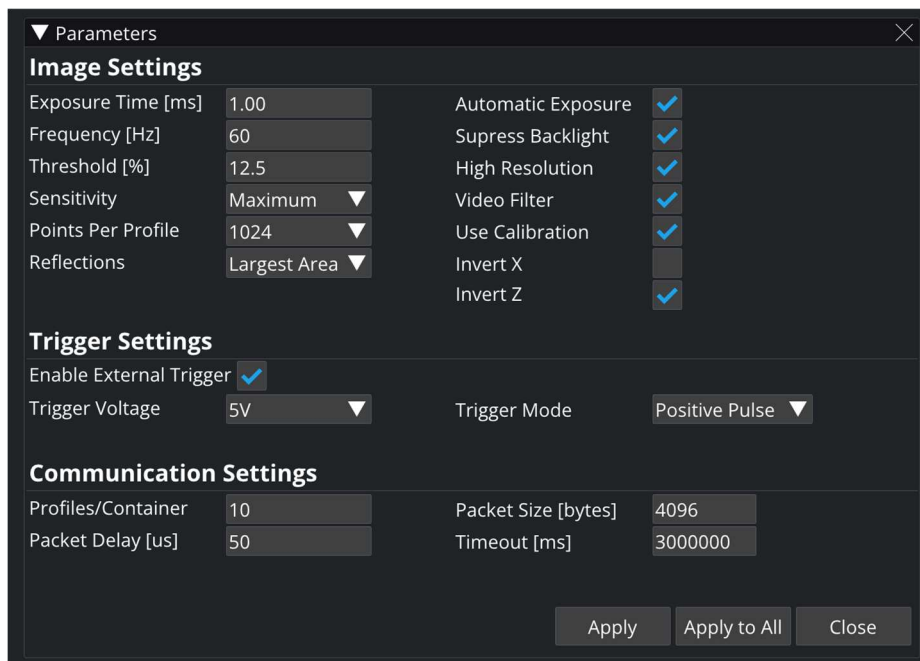


Figura 50. Ventana de Parámetros

Los parámetros se agrupan en parámetros de imagen, parámetros del disparador y parámetros de comunicación. Los primeros consisten en:

- Exposure Time: Tiempo en milisegundos donde el sensor fotosensible capta la imagen. Mayor tiempo de exposición se traduce en mayor cantidad de luz captada.
- Frequency: Frecuencia de adquisición de imágenes en caso de que se desactive el disparador.
- Threshold: Valor en porcentaje del nivel de luz máxima por debajo del cual no se considera suficiente para pertenecer a la imagen.
- Sensitivity: Nivel de sensibilidad del sensor fotosensible, pudiendo ser máxima, alta, media, baja y mínima.
- Points per Profile: Puntos por cada imagen. Es indicativo, no se puede variar.

- Reflections: Indica qué reflejo procesar, para distinguir el reflejo real del láser de otros debido a la luz ambiental. Se puede elegir entre el primero que llegue al sensor, el último, el de mayor área, el de mayor intensidad, un único reflejo diciendo que no existen otros, o todos.
- Automatic Exposure: Si se activa, el sensor calcula automáticamente el tiempo de exposición óptimo, tomando como referencia el introducido manualmente.
- Suppress Backlight: Suprimir la luz de fondo para mejorar la calidad de la imagen.
- High Resolution: Emplear el perfilómetro en modo de alta resolución.
- Video Filter: Aplicar un filtro interno del sensor para eliminar posibles reflejos indeseados.
- Use Calibration: Utilizar la calibración interna del sensor para obtener coordenadas en milímetros.
- Invert X: Invertir el eje transversal de coordenadas del sistema de referencia local del sensor.
- Invert Z: Invertir el eje vertical de coordenadas del sistema de referencia local del sensor. Útil si la imagen que se obtiene está invertida.

Los parámetros del disparador consisten en:

- Enable External Trigger: Activar o desactivar el uso del disparador activado con señal externa.
- Trigger Voltage: Voltaje del valor alto de la señal externa, pudiendo ser de 5 o 24 voltios.
- Trigger Mode: Indica en qué situación se activa el disparador. Puede activarse en un flanco de subida, en un flanco de bajada, en un pulso de voltaje positivo o en un pulso de voltaje negativo.

Por último, los parámetros de comunicación son:

- Profiles/Container: Número de imágenes a agrupar en paquetes para disminuir la frecuencia de transmisión de datos.
- Packet Delay: Tiempo de desfase en microsegundos en el envío de datos cuando hay conectados más de un sensor.
- Packet Size: Tamaño de los paquetes en bytes que se pueden transmitir en el cable ethernet conectado al ordenador.
- Timeout: Tiempo máximo de espera en milisegundos para recibir un paquete de información. Si se supera, se suspende la conexión con el sensor.

3.1.4.3 Visualización y procesado de datos

Tras realizar la adquisición de datos y generar los archivos *.img*, en la pestaña File de la barra de herramientas se puede seleccionar la opción Open Image File con la que se abre una nueva ventana en la que se representa una imagen. En la barra superior se puede modificar el número de la imagen que se quiera visualizar, y con las flechas izquierda y derecha del teclado se puede avanzar de fotograma en fotograma.

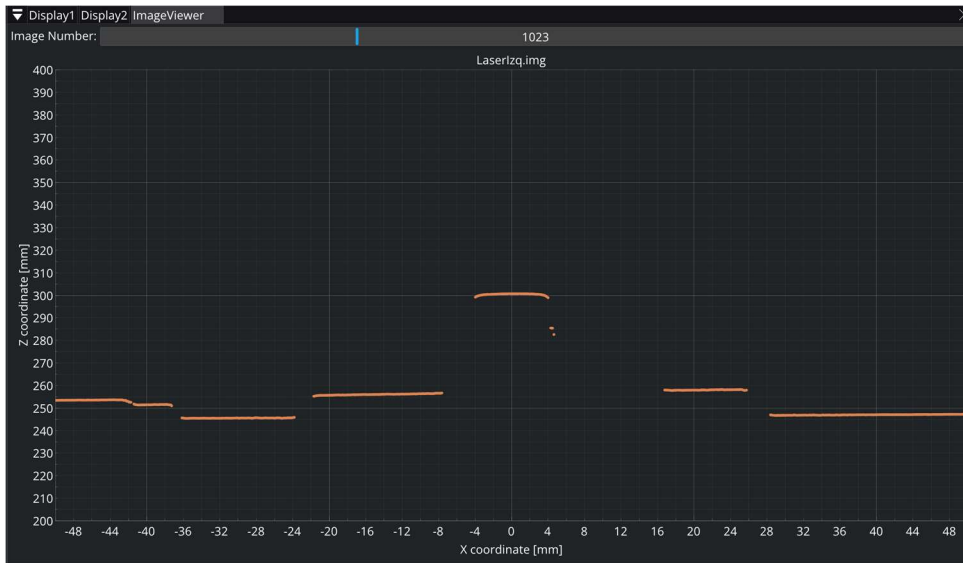


Figura 51. Visualización de Imágenes Grabadas

La opción Export to txt per image exporta cada imagen del archivo *.img* seleccionado en un archivo de texto independiente que contiene las coordenadas de los 1024 puntos que la forman. Se crea una nueva carpeta en el directorio del archivo seleccionado con su mismo nombre, donde se generan los archivos de texto enumerados en orden.

Finalmente, para el cálculo de la corrugación, se selecciona la opción Calculate corrugation from image file, en donde se carga el archivo *.img* seleccionado. Inmediatamente, se abre una ventana donde se representa la primera imagen que contiene el archivo y un rectángulo amarillo que debe ser desplazado y escalado para definir la región de interés de la imagen donde se encuentre el perfil del carril.



Figura 52. Definición del Filtro

Establecida la posición del rectángulo, se pulsa el botón OK, momento en el que se cierra esta ventana y se abre una nueva donde se deben introducir los datos necesarios para el cálculo de la corrugación.

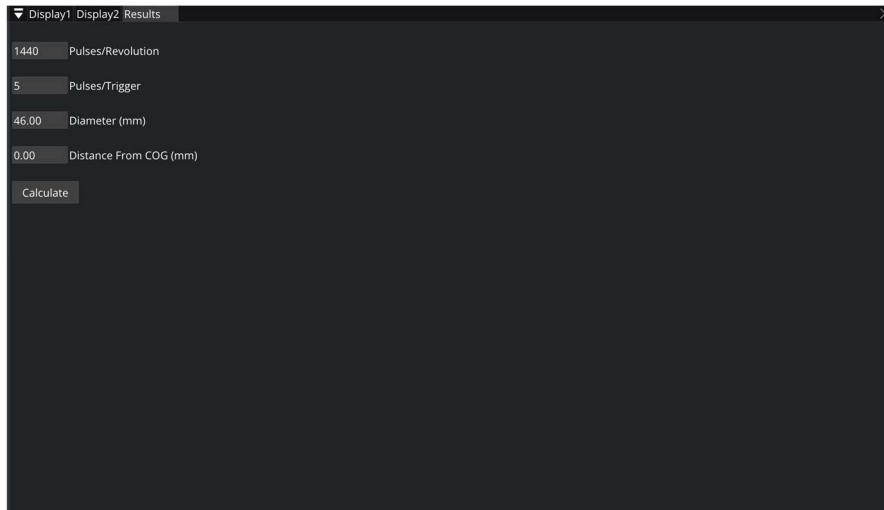


Figura 53. Definición de Parámetros para el Cálculo de la Corrugación

Los parámetros que se deben definir son el número de pulsos por revolución del encóder, el número de pulsos por cada señal enviada al disparador del perfilómetro láser, el diámetro (D) de la rueda montada en el encóder que rueda sobre la estructura en milímetros y la distancia transversal al centro de gravedad (d) donde se quiere medir la corrugación. Esta distancia sigue el siguiente criterio de signo:

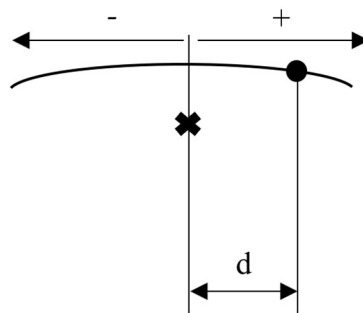


Figura 54. Criterio de Signos Distancia d

Con estos parámetros, se calcula la distancia que hay entre dos imágenes consecutivas de la forma:

$$\Delta s = \frac{\pi D}{\text{pulsos/rev.}} * \text{pulsos/disparador} \quad (3.2)$$

Una vez introducidos los valores, se pulsa el botón Calculate, y se muestran a la derecha las representaciones gráficas de la altura del punto del perfil del carril seleccionado y su contenido en frecuencia.

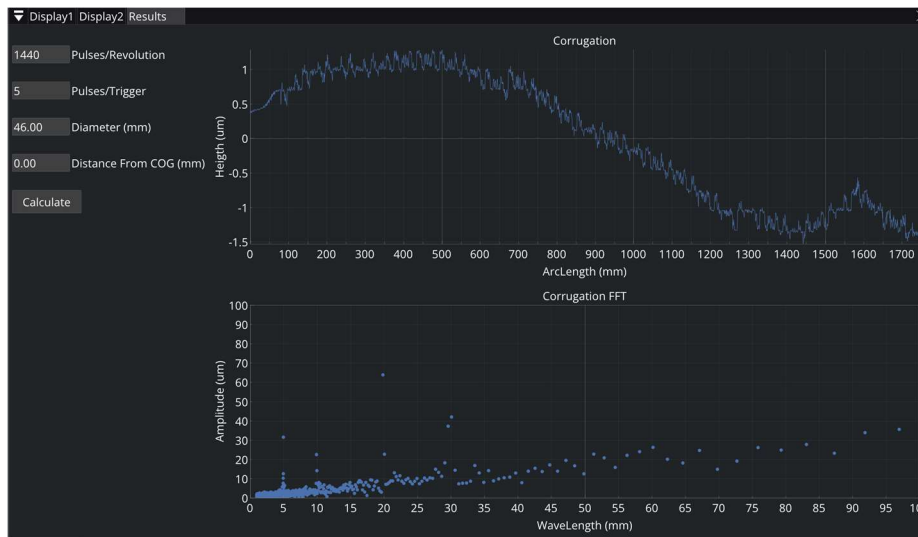


Figura 55. Representación de Resultados

3.4. Resultados

El sistema diseñado se ha utilizado para medir la corrugación que existe en un segmento de la vía a escala instalada en la azotea de la escuela. Se ha montado la estructura, fabricado las piezas para la sujeción de los sensores y puesto a punto el sistema para poder realizar estas medidas.



Figura 56. Sistema Midiendo en la Vía a Escala



Figura 57. Montaje del Encóder

La corrugación en la vía es el resultado de mecanizar un carril con una cuchilla con la forma del perfil del carril imponiendo una evolución a lo largo del carril con una curva analítica que está formada por la suma de cuatro armónicos con las siguientes características:

Tabla 1. Características de la Corrugación

Longitud de Onda (mm)	Amplitud (μm)
5	45
10	75
20	60
30	30



Figura 58. Trozo de Carril Corrugado

De esta forma, se esperan obtener en el espectro de longitudes de onda cuatro puntos en estas longitudes de onda y con estas amplitudes. Además, se espera que los resultados sean independientes de la línea seleccionada para el cálculo, ya que el mecanizado del carril es uniforme transversalmente.

A continuación, se representan los resultados obtenidos en los carriles izquierdo y derecho en distintas líneas:

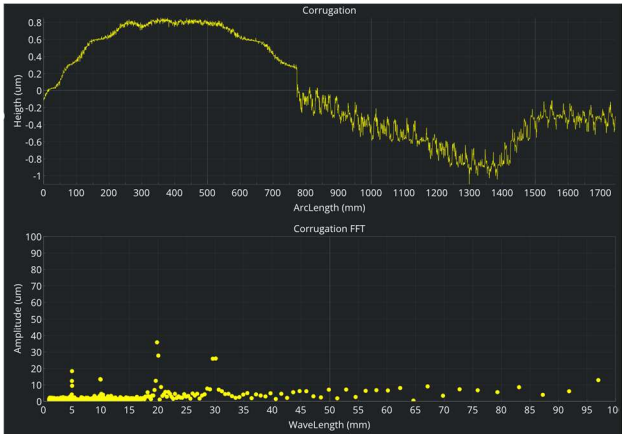


Figura 59. Carril Derecho con $d = 0$ mm

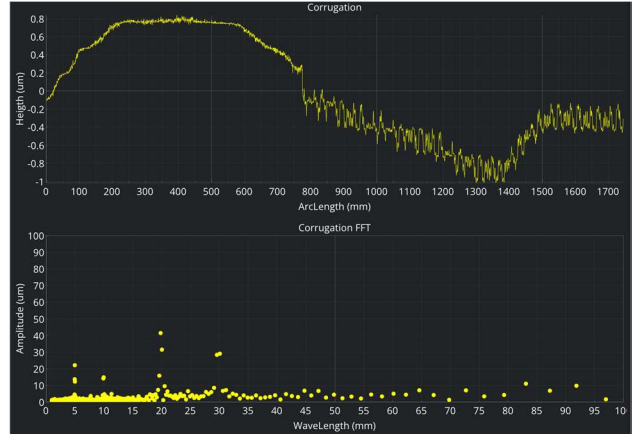


Figura 60. Carril Derecho con $d = 2$ mm

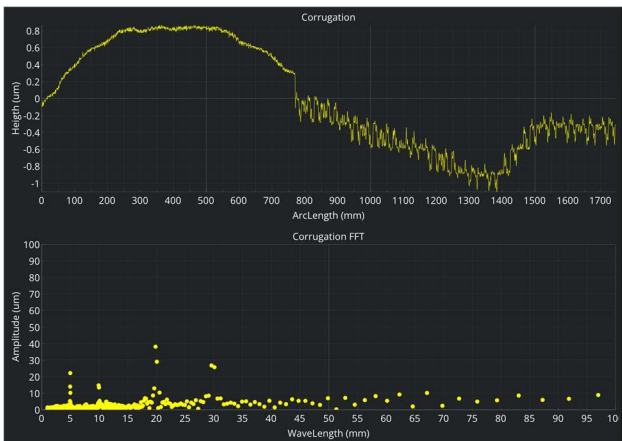


Figura 61. Carril Derecho con $d = -2$ mm

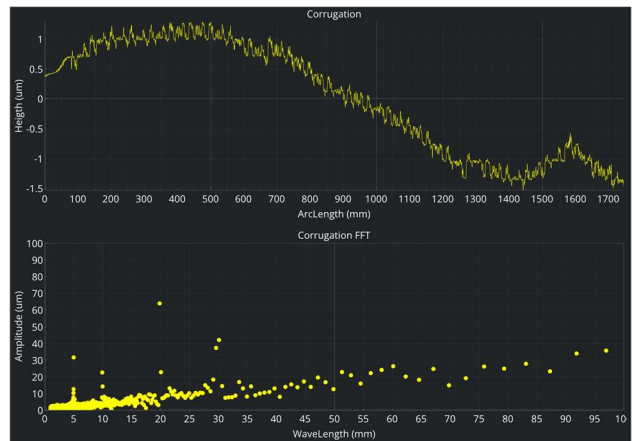


Figura 62. Carril Izquierdo con $d = 0$ mm



Figura 63. Carril Izquierdo con $d = 2$ mm



Figura 64. Carril Izquierdo con $d = -2$ mm

En primer lugar, en la evolución de la corrugación en el carril derecho aparecen dos zonas claramente diferenciadas unidas por una discontinuidad. Esto es debido a que en el segmento medido existe una primera zona de carril sin corrugación y una segunda zona con corrugación. Estas dos zonas son dos tramos de carriles separados, siendo la discontinuidad la unión entre ambos.

Por otra parte, se comprueba que los resultados en ambos carriles son independientes de la distancia transversal d , resultado esperado. Además, en el dominio de la frecuencia aparecen claramente los picos en las longitudes de onda esperadas de 5, 10, 20 y 30 mm.

Finalmente, en cuanto a las amplitudes, en el carril izquierdo se consiguen obtener los valores teóricos salvo para la longitud de onda de 10 mm, donde se obtiene un valor más pequeño del teórico. En cambio, en el carril derecho, el valor de las amplitudes se subestima en general, hecho que puede ser debido a que se está utilizando la señal completa y no solo el segmento de la corrugación. Sin embargo, se observa que la señal tiene un aspecto muy similar en todos los registros.

4 CONCLUSIONES

En este trabajo se han desarrollado dos sistemas para la medición de las irregularidades de larga y corta longitud de onda respectivamente. El primer sistema unifica un conjunto de sensores y equipos de adquisición en una única aplicación de escritorio y es capaz de realizar la adquisición y procesado de imágenes en tiempo real a una frecuencia de 250 Hz, cumpliendo los dos primeros objetivos establecidos inicialmente

El segundo sistema ha sido diseñado para poder realizar medidas de la corrugación de carriles tanto en la vía a escala como en vías reales gracias a la flexibilidad de su montaje. Además, se controla a través de una aplicación de escritorio donde se configuran los sensores, graban y procesan los datos y se representan los resultados. De esta forma, se cumplen los dos últimos objetivos definidos al principio de este trabajo.

Pese a utilizar sensores diferentes, la tecnología y principios físicos que utilizan son muy similares. Las cámaras tienen la ventaja de poder ser instaladas en cualquier posición siempre que sean capaces de captar la proyección del láser, mientras que los perfilómetros láser no proporcionan esta posibilidad. En contrapartida, las cámaras necesitan ser calibradas por el usuario, mientras que los perfilómetros incorporan la suya propia. Además, de esta forma, la precisión de las medidas del perfilómetro es fija y conocida, mientras que la precisión de las medidas realizadas con las cámaras puede variar en función del tamaño del perfil del carril en la imagen.

Finalmente, el desarrollo del primer sistema representa una evolución hacia la medida de las irregularidades de larga longitud de onda en tiempo real y presenta muy buenas oportunidades para incorporar el resto de algoritmos que tienen en cuenta la dinámica del vehículo. La visión artificial ocupa menos del 2% del tiempo de computación disponible para garantizar el funcionamiento en tiempo real, dejando mucho margen para implementar todos los algoritmos.

En cuanto al segundo sistema, se observa que es capaz de medir la corrugación con precisión, pudiendo llevarlo a una vía real para evaluar exactamente la geometría del defecto generado por el desgaste ondulatorio en el carril.

REFERENCIAS

- [1] A. H. Wickens, «Fundamentals of Rail Vehicle Dynamics: Guidance and Stability» *CRC Press* 2005.
- [2] C. Esveld, «Modern Railway Track», *MRT-productions*, Zaltbommel, 2006.
- [3] S. Iwnicki, «Handbook of Railway Vehicle Dynamics» *Taylor and Francis Group*, 2006, <https://doi.org/10.1201/9781420004892>.
- [4] Q. Chen, X. Niu, L. Zuo, T. Zhang, F. Xiao, Y. Liu, J. Liu, «A railway track geometry measuring trolley system based on aided INS» *Sensors* vol. 18 (2) 538, 2018, <https://doi.org/10.3390/s180205388>.
- [5] K. H. Oostermeijer, «Review on short pitch rail corrugation studies» *Wear* vol. 265 (9-10), 2008.
- [6] Q. Chen, X. Niu, Q. Zhang, Y. Cheng, «Railway track irregularity measuring by gnss/ ins integration» *Navigation* vol. 62 (1), 2015 83–93, <https://doi.org/10.1002/navi.78>.
- [7] H. Tsunashima, Y. Naganuma, T. Kobayashi, «Track geometry estimation from car-body vibration» *Vehicle Syst. Dyn.* Vol. 52 (sup1) ,2014, 207–219, <https://doi.org/10.1080/00423114.2014.889836>.
- [8] P.F. Westeon, C.S. Ling, C. Roberts, C.J. Goodman, P. Li, R.M. Goodall «Monitoring vertical track irregularity from in-service railway vehicles» *Proc. Inst. Mech. Eng., Part F: J. Rail Rapid Transit* vol. 221 (1), 2007, 75–88, <https://doi.org/10.1243/0954409JRRT65>.
- [9] J. Theurer, J. Hansmann, K. Riessberger, «Measuring apparatus for rail head running surface irregularities» *uS Patent* 4.391.134, 1981.
- [10] J. Trefouel, M. Centil. «Railroad vehicle for measuring the geometrical parameters of railroad track» *uS Patent* 5.203.089, 1993.
- [11] J. Therer, B. Lichtberger, «Method of surveying a track» *uS Patent* 7.050.923 B2, 2006.
- [12] G.A. Carr, C. Diaz, J. Bloom. «Method and apparatus for track geometry measurement» *uS Patent* 6.634.112 B2, 2003.
- [13] Z. Zhang. «A flexible new technique for camera calibration» *Technical Report* MSR-TR-98-71, Microsoft Research, 1998.
- [14] Escalona, J.L. «Kinematics of motion tracking using computer vision» *arxiv.org*, 2020.
- [15] Burger, W. «Zhang's Camera Calibration Algorithm: In-Depth Tutorial and Implementation» *Technical Report* HGB16-05. University of Applied Sciences, Upper Austria, 2016.
- [16] Lafore, R. «Object Oriented Programming in C++» 4th ed. Indianapolis: Sams, 2002.
- [17] Liberty, J. Jones, B. «Teach Yourself C++ in 21 Days» 5th ed. Indianapolis: Sams, 2005.
- [18] Lee, M. «C++ Programming for the Absolute Beginner» 2nd ed. Boston: Course Technology PTR, 2009.
- [19] Williams, A. «C++ Concurrency in Action: Practical Multithreading» New York: Manning Publications, 2012.