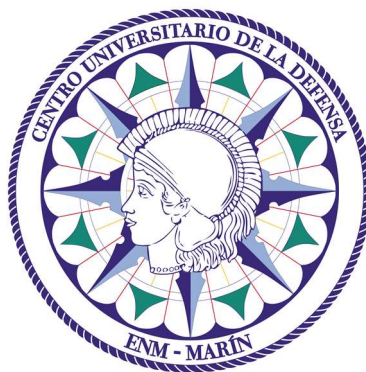




Centro Universitario de la Defensa en la Escuela Naval Militar

TRABAJO FIN DE GRADO

*Diseño e implementación de un sistema de seguimiento para
sistemas de comunicaciones basados en LASER en plataformas
navales*

Grado en Ingeniería Mecánica

ALUMNO: Guillermo Bravo Díaz-Montenegro

DIRECTORA: Paula Gómez Pérez

CURSO ACADÉMICO: 2018-2019

UniversidadeVigo



Centro Universitario de la Defensa en la Escuela Naval Militar

TRABAJO FIN DE GRADO

*Diseño e implementación de un sistema de seguimiento para
sistemas de comunicaciones basados en LASER en plataformas
navales*

Grado en Ingeniería Mecánica

Intensificación en Tecnología Naval

Cuerpo General / Infantería de Marina

UniversidadeVigo

RESUMEN

Este Trabajo Fin de Grado trata sobre el diseño e implantación de un sistema de telecomunicaciones LASER en plataformas navales. En particular, dicho sistema sería el de modular un haz de luz para comunicaciones punto a punto, aprovechando la directividad y alcance que nos proporciona el LASER. El sistema presenta dos principales ventajas en la mejora de las telecomunicaciones navales de hoy en día: ancho de banda y discreción. El ancho de banda permite llegar a tasas de transmisión equiparables a las de la fibra óptica, multiplicando las tasas actuales obtenidas mediante radiofrecuencia, y sin restricciones en cuanto a frecuencias de uso, y la directividad del LASER permitiría una discreción muy ventajosa a nivel operativo ya que la interceptación de dichas comunicaciones en el medio es altamente improbable.

Este sistema estaría compuesto por dos subsistemas, uno encargado de gestionar la comunicación en sí, y otro de gestionar la plataforma necesaria para la implementación del sistema. Este Trabajo desarrolla tanto el diseño del subsistema de adquisición y seguimiento de la estación receptora como la propia plataforma del sistema. En concreto, se diseñará un sistema de adquisición y seguimiento basado en análisis de imagen, así como una plataforma de tipo cardan.

PALABRAS CLAVE

Telecomunicaciones, FSOC, seguimiento, plataforma

AGRADECIMIENTOS

Quiero agradecer en primer lugar a las personas que me han ayudado desinteresadamente en este Trabajo Fin de Grado, aportando su ayuda o prestando sus servicios cuando no les correspondía. En particular, quiero agradecer al Sr. Casqueiro por su aportación a este Trabajo Fin de Grado, ya que ha sido el encargado de imprimir todas las piezas de la plataforma, así como a Don David y Don Manuel del Taller Mecánico de la Escuela Naval por su ayuda con la mecanización de la base estática. Les estoy profundamente agradecido a todos ellos.

Por otro lado, quiero agradecer a los cursos y foros online de programación, y a toda esa gente que participa en la difusión del conocimiento en dicho campo de forma altruista y desinteresada, sin los cuales este Trabajo Fin de Grado no habría sido posible. Publico todos los códigos que he producido con la esperanza de que algún día le sirvan a alguien, como me han servido a mí tantos otros.

Por último, quiero agradecer a mi familia, mi madre Sylvia, mi padre Juan y mis tres hermanas Inés, Blanca y Rocío, todo el apoyo incondicional durante esta etapa y durante toda la carrera. A todos ellos les dedico este Trabajo y les agradezco todo el cariño que me han dado durante todos estos años, sin el cual no estaría donde estoy.

CONTENIDO

Contenido	1
Índice de Figuras	3
Índice de Tablas.....	7
1 Introducción y objetivos.....	9
1.1 Reseña histórica	9
1.2 Contexto y objetivos	14
2 Estado del arte	17
2.1 Sistemas FSOC	17
2.1.1 Características generales.....	17
2.1.2 Limitaciones	21
2.2 Aplicaciones de las FSOC.....	27
2.2.1 Desarrollo inicial: telecomunicaciones espaciales.....	27
2.2.2 Aplicaciones dentro de la atmósfera terrestre.....	30
2.2.3 Aplicaciones en la superficie terrestre	38
2.2.4 Aplicaciones militares	47
2.3 Sistemas de adquisición y seguimiento.....	49
2.3.1 Principios de funcionamiento	49
2.3.2 Plataformas FSOC	54
3 Desarrollo del TFG.....	59
3.1 Sistema de seguimiento.....	59
3.1.1 Material.....	59
3.1.2 Código principal	63
3.1.3 Tratamiento de imágenes	65
3.2 Plataforma del sistema	74
3.2.1 Material.....	74
3.2.2 Estructura de la plataforma	78
3.2.3 Funcionamiento del sistema	83
4 Resultados / Validación / Prueba.....	87
4.1 Tratamiento de imágenes	87
4.1.1 Resolución y tasa de frames.....	87
4.1.2 Tolerancias.....	93
4.1.3 Falsa alarma	95
4.1.4 Limitaciones del sistema.....	97

4.2 Plataforma	99
4.2.1 Calibración.....	99
4.2.2 Transmisión de movimiento	103
5 Conclusiones y líneas futuras	105
5.1 Conclusiones del Trabajo Fin de Grado.....	105
5.2 Líneas futuras	106
6 Bibliografía.....	109
Anexo I: Código del microprocesador	115
Anexo II: Código del microcontrolador	119
Anexo III: Código de muestro de la luminosidad	121

ÍNDICE DE FIGURAS

Figura 1-1 Bandera para la flota de la Armada y la flota mercante (de izquierda a derecha) [3]	10
Figura 1-2 Estación de comunicación del sistema de Chappe [2].....	10
Figura 1-3 Prototipo de telégrafo [5].....	11
Figura 1-4 Retrato de Antonio Meucci junto a su invento [6]	12
Figura 2-1 Diagrama de modulaciones en FSOC [13]	19
Figura 2-2 (a) Diagrama general del funcionamiento de un sistema FSOC [10], (b) Gráfico de máximo ancho de banda en relación a la distancia (3 casos) [14]	20
Figura 2-3 (a) Esquema de funcionamiento de un sistema de óptica adaptativa convencional [14] , (b) Esquema de funcionamiento de un sistema de óptica adaptativa no convencional con enlace por radiofrecuencia auxiliar [14]	21
Figura 2-4 Ámbito de aplicación de las FSOC en función de alcance.....	21
Figura 2-5 (a)Gráfico de atenuación específica de la atmósfera (niebla) [10], (b) Gráfico de atenuación específica de la atmósfera (condiciones normales) [10]	22
Figura 2-6 Fotografía de la temperatura del aire en régimen turbulento (mediante cámara termográfica) [18]	23
Figura 2-7 Representación del haz frente a la superficie receptora [15].....	26
Figura 2-8 (a) Modelo del LLST (Lunar Lasercomm Space Terminal) [22], (b) Modelo del LLGT (Lunar Lasercomm Ground Terminal) [22]	28
Figura 2-9 Representación artística del EDRS [23]	29
Figura 2-10 Representación artística del OPALS [25].....	29
Figura 2-11 Representación funcional del sistema HALO [26].....	30
Figura 2-12 Países en los que se ha adoptado la tecnología LTE [27].....	31
Figura 2-13 (a) Zonas con cobertura 4G (en violeta) [30], (b) Zonas con cobertura 3G (en verde) [30]	32
Figura 2-14 Esquema de la carga útil de un globo de Loon: (1) antenas LTE, (2) panel solar, (3) sistema de gobierno y FSO, (4) paracaídas [28]	32
Figura 2-15 Representación artística de modo de empleo de globos Loon [28].....	33
Figura 2-16 (a) Representación artística del desplazamiento del globo Loon (1) [28], (b)Representación artística del desplazamiento del globo Loon (2) [28], (c) Representación artística del agrupamiento de globos Loon [28]	34
Figura 2-17 (a) Zonas de cobertura por globos Loon (naranja) y zonas afectadas por inundaciones (rojo) en Perú [28], (b) Fotografía de un globo Loon [28], (c) Representación geográfica de ensayo de predicción de modelos meteorológicos y algoritmos de navegación autónomos [28]	35
Figura 2-18 Póster publicitario del proyecto Aquila [33]	36
Figura 2-19 (a) Representación artística del enlace FSOC de Aquila [31] , (b) Representación artística de zona de cobertura de Aquila [31], (c) Comparativa de la zona de cobertura de Aquila frente a satélite [31].....	37
Figura 2-20 Representación artística de UAV Zephyr [35]	38

Figura 2-21 Esquema de enlace estación a estación (blackhaul communication) [29].....	40
Figura 2-22 Esquema de instalación de FSOC en CAN [17]	40
Figura 2-23 Comparativa de pérdida de paquetes frente a absorción atmosférica [36]	41
Figura 2-24 Gráfico de absorción de la atmósfera en función de la longitud de onda [14]	42
Figura 2-25 (a) Fotografía de los módulos transmisor y receptor de Ronja (perfil) [43], (b) Fotografía del módulo transmisor de Ronja en funcionamiento [43], (c) Fotografía del haz de luz en dispersión por la niebla [43].....	43
Figura 2-26 Diagrama de funcionamiento de Ronja [43].....	44
Figura 2-27 Esquema de un sistema LiFi [46]	44
Figura 2-28 Diagrama de funcionamiento de multiplexación NOMA [45]	46
Figura 2-29 Diagrama de funcionamiento de multiplexación CoMP-AJT [45].....	46
Figura 2-30 (a) Comparativa de Láser CW y USPL mediante fotografía infrarroja [23], (b) Gráfico de potencia de la señal en recepción de Láser CW (rojo) y USPL (azul) en función de la visibilidad [23]	48
Figura 2-31 (a) Póster publicitario de sistema TALON [51], (b) Módulo FSOC de TALON [51] ..	49
Figura 2-32 Filtro Bayer [54]	50
Figura 2-33 (a) sensor CMOS [66], (b) Fotografía con efecto <i>blooming</i>	51
Figura 2-34 (a) Fotografía de un PSD [55], (b) Esquema de un PSD [55]	51
Figura 2-35 Fotografía de un QPD [56]	52
Figura 2-36 (a) Enlace inicial entre dos estaciones [45], (b) desalineamiento y transmisión de posición mediante radiofrecuencia [45], (c) realineamiento [45]	53
Figura 2-37 Esquema de un sistema de seguimiento esférico [14]	54
Figura 2-38 Diagrama de tipos de plataformas FSOC [14].....	54
Figura 2-39 (a) Esquema de funcionamiento de un sistema FSOC basado en un espejo deformable MOEMS tanto para seguimiento como para comunicación [45], (b) Funcionamiento de un espejo deformable MOEMS con membrana [45]	56
Figura 2-40 Esquema de un OPA basado en cristal líquido transmissivo [45]	57
Figura 3-1 (a), (b), (c), (d), (e), (f) Representación cronológica del proceso de captura de fotograma de la Picamera [57]	60
Figura 3-2 (a) Esquema del sistema integrado de la Picamera [57], (b) Fotografía de la Picamera conectada a la Raspberry Pi [57].....	62
Figura 3-3 Fotografía de la Raspberry Pi 3 Model B v1.2 [58]	63
Figura 3-4 Soporte de la Picamera	63
Figura 3-5 Código principal del microprocesador.....	64
Figura 3-6 Diagrama de flujo del funcionamiento de la cámara	65
Figura 3-7 Ejemplo de matriz numpy YUV	66
Figura 3-8 Código del analizador	67
Figura 3-9 Diagrama de flujo del analizador de imágenes.....	68

Figura 3-10 Ejemplo de matriz de objetos de puntos luminosos	69
Figura 3-11 Código de la función CompararValor.....	70
Figura 3-12 Código de función AnhadirMatriz.....	71
Figura 3-13 Código de función PatronCuadrado	73
Figura 3-14 Código para la transmisión de órdenes por el puerto serial.....	74
Figura 3-15 Fotografía del Arduino Uno [59].....	74
Figura 3-16 Diagrama de modulación PWM en Arduino [60].....	75
Figura 3-17 (a) Fotografía de un motor 28BYJ-48 enchufado a un driver ULN2003A, (b) Circuito del bobinado interno del motor 28BYJ-48 [61]	76
Figura 3-18 (a) Vista en planta del driver ULN2003A [61], (b) Circuito para un par de pines (salida y entrada) [61], (c) Circuito esquemático del driver ULN2003A [61]	77
Figura 3-19 Representación del circuito de un Joystick acoplado a un Arduino [65]	77
Figura 3-20 Modelo del ensamble de la plataforma.....	78
Figura 3-21 Base estática de la plataforma	79
Figura 3-22 Modelo de base móvil.....	80
Figura 3-23 Modelo de rueda dentada de diámetro primitivo de 10 mm.....	80
Figura 3-24 (a) Modelo de bulón, (b) Modelo de eje transmisor de par	81
Figura 3-25 Modelo de la carcasa del módulo de comunicación	82
Figura 3-26 Modelo de rueda dentada de diámetro primitivo de 14 mm.....	82
Figura 4-1 Baliza luminosa	87
Figura 4-2 Código para prueba de resolución y tasa de <i>frames</i>	89
Figura 4-3(a) Gráfica de porcentaje de detección para una resolución de 320X280, (b) Gráfica de porcentaje de detección para una resolución de 160X120	90
Figura 4-4 Gráfica de porcentaje de detección para una resolución de 80X60.....	91
Figura 4-5 (a), (b) y (c) Modificaciones al código para la realización de la prueba de velocidad...	92
Figura 4-6 Fotografía del emplazamiento donde se llevan a cabo las pruebas de tolerancia.....	94
Figura 4-7 Gráfico de porcentaje de detección del patrón en función de la tolerancia	95
Figura 4-8 Gráfico de porcentaje de falsa alarma en función de tolerancia	96
Figura 4-9 Nivel de luminosidad de la linterna	96
Figura 4-10 Gráfico de porcentaje de falsa alarma en función de umbral	97
Figura 4-11 Fotografía de las pruebas de día en exteriores.....	98
Figura 4-12 Muestra de la luminosidad del entorno buenas condiciones meteorológicas (en torno a 240)	98
Figura 4-13 Fotografía de la baliza en buenas condiciones meteorológicas	99
Figura 4-14 (a) y (b) Partes del código donde interviene las variables de calibración	100
Figura 4-15 (a) Módulo de calibración, (b) Calibración del módulo de calibración, (c) Calibración de la plataforma.....	101

Figura 4-16 (a), (b), (c) y (d) Proceso de calibración de la plataforma.....102

Figura 4-17 Fotografía del engranaje encargado de la elevación con ruedas dentadas iguales103

Figura 4-18 Fotografía del engranaje encargado de la elevación final104

Figura Anexo III-1 (a) Valor medio de Y en la parcela en blanco, (b) Valor medio de los valores RGB y YUV de la parcecla en blanco122

ÍNDICE DE TABLAS

Tabla 1 Valores típicos de atenuación específica de la atmósfera para $\lambda = 1550$ nm [15]	25
Tabla 2 Comparativa de servicios en función de tecnología blackhaul [29].....	40
Tabla 3 Comparativa de productos FSOC.....	42
Tabla 4 Comparativa de plataformas FSOC [45].....	57
Tabla 5 Tasa de fotogramas procesados por segundo en función de la resolución de la Picamera [57]	61
Tabla 6 Características de la rueda dentada de la base móvil	79
Tabla 7 Resultados de prueba de velocidad (en órdenes por minuto).....	92

1 INTRODUCCIÓN Y OBJETIVOS

1.1 Reseña histórica

En el ámbito de las telecomunicaciones, se puede observar que a lo largo de la historia de la humanidad ha habido un gran interés por el desarrollo de medios que permitiesen establecer una transmisión de información más fiable, a mayor distancia y en el menor tiempo posible. La transmisión de información, antes de ser una necesidad personal, constituía una prioridad en la escena política y militar ya que afectaba en gran medida a la gestión y coordinación de un gran número de personas segregadas a lo largo de un vasto espacio. Es por ello que desde la Antigua Grecia se ingeniaban plataformas para la aceleración de la transmisión de información. En particular, los griegos de la época usaban antorchas como señales visuales, que, mediante un previo acuerdo sobre su significado, aseguraba una comunicación fiable entre todas las ciudades-estado para mensajes de extrema urgencia y que necesitaban estar en conocimiento de todas las autoridades de dichas ciudades a la mayor brevedad. Dichos mensajes, de no ser por este invento, solo disponían de los buques de la época para su transmisión, que tardaban varios días en recorrer las distancias que separaban las islas del archipiélago griego. Se puede citar el caso mencionado en [1], cuando en la guerra de Troya, que tuvo lugar en el siglo XII o XIII a.C. (no se sabe la fecha con precisión), la caída de la ciudad se transmitió mediante el empleo del sistema anteriormente mencionado a través de las ciudades-estado tales como Lemnos, Atenas, Eúbea y llegó hasta la ciudad de Micenas, recorriendo alrededor de 650 km en apenas horas [1]. De hecho, este método se mantendría hasta más de dos mil años después para avisar de la llegada de enemigos a nuestras costas.

En el ámbito naval, cabe reseñar que el medio predominante de comunicación eran señales visuales, y en concreto mediante gallardetes. Ya desde los tiempos de la Armada de la Corona de Castilla, en 1340, se impusieron dichas señales para dar órdenes a la hora de atacar en formación a los buques de la Corona de Aragón y perdura hasta la actualidad como medio alternativo de comunicación [2]. Este tipo de señales tuvieron tanta relevancia que el 28 de mayo de 1785 el rey Carlos III aprobó por Real Decreto la bandera de cuyos colores heredamos la actual bandera nacional. Este cambio se produjo debido a la cantidad de equivocaciones en la mar a la hora de identificar buques, ya que muchas naciones europeas estaban regidas por miembros de la Casa de Borbón y por tanto portaban pabellones de fondo blanco, de las cuales España, Francia, Sicilia o Toscana. Teniendo en cuenta los numerosos enfrentamientos entre estas naciones, estos pabellones daban pie a confusión entre aliados y enemigos, lo que empujó al rey a encomendarle al Ministro de Marina Valdés el diseño de dos banderas específicas para el ámbito marítimo, una para la Armada y otra para la flota mercante, como se muestra en la Figura 1-1 [3] [4].



Figura 1-1 Bandera para la flota de la Armada y la flota mercante (de izquierda a derecha) [3]

Pero cabe reseñar que no fue hasta la época de la Revolución francesa, en concreto en 1792, que se establece la primera red de comunicación entendida como tal, es decir el primer sistema de nodos transmisores y receptores de información, pudiendo ser esta de cualquier índole (y no un único mensaje pre pactado) [2]. Este sistema constaba de una red de torres en cuya cúspide se instalaba una barra de acero en cuyos extremos se unían dos barras más, una en cada extremo, de tal forma que la posición de las barras indicaba una sílaba, como podemos ver en la Figura 1-2. Este sistema permitía el empleo de 196 figuras o “símbolos” y aunque la velocidad de transmisión era lenta en comparación con las tecnologías de las que disponemos hoy en día y el enlace se veía muchas veces comprometido por las condiciones climatológicas (sobre todo por lluvia y niebla), se conseguían velocidades de transmisión de cientos de kilómetros por hora, lo cual supuso una revelación en comparación con los tradicionales mensajeros que tardaban varios días en recorrer lo que mediante el sistema de Chappe se conseguía en cuestión de horas sino minutos. Tanta fue la importancia de dicho avance que Napoleón Bonaparte adoptó este sistema con medio fundamental para la gestión de todo el territorio francés, cubriendo una distancia de más de 5000 km [1].

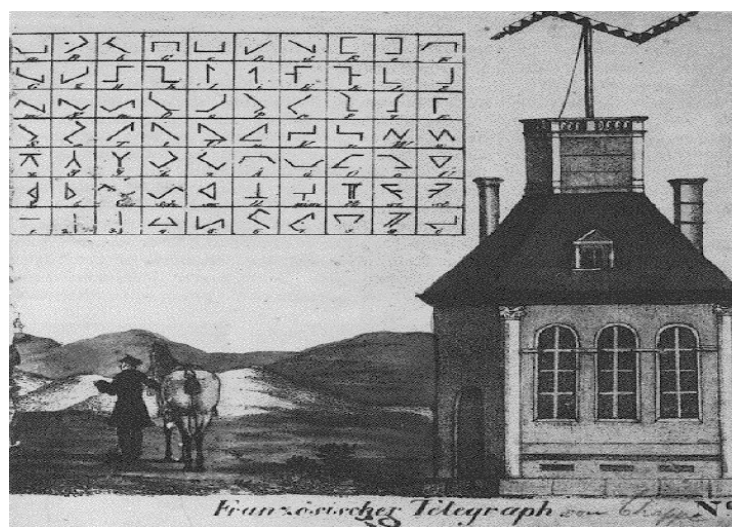


Figura 1-2 Estación de comunicación del sistema de Chappe [2]

Este sistema opto-mecánico no fue más que el principio de una auténtica avalancha de innovaciones en cuanto a telecomunicaciones se refiere, que vinieron de la mano de la Revolución Industrial. Hacia el final de ésta, los físicos de la época empezaron a estudiar en profundidad el campo de la electricidad y el magnetismo y la relación entre ambos. Descubrimientos como el de Hans Christian Oersted o André Marie Ampère relativos a la creación de un campo magnético debido al paso de una determinada corriente eléctrica en 1819 y 1820, y el de Michael Faraday relativo a la generación de corriente eléctrica mediante un campo magnético variable en 1831, fueron la base para

los avances tecnológicos que permitieron llegar a los medios de telecomunicación de los que disponemos hoy en día. Cabe también mencionar el estudio del físico escocés James Clerk Maxwell, *A Dynamical Theory of the Electromagnetic Field*, publicado en 1865 [2] y que plasma en última instancia la relación física de electricidad y magnetismo como expresiones de un mismo fenómeno, ya que demostró que tanto el campo eléctrico como el magnético se propagaban por el espacio a la velocidad de la luz, postulando que la luz fuese en sí misma una ondulación de dicha naturaleza. Este estudio supuso el pilar de toda innovación en el campo de las ondas electromagnéticas, ganando el renombre de “Segunda gran unificación de la Física”, tras los estudios del físico inglés Isaac Newton [1]. De este impulso que protagonizó la comunidad científica en el siglo XIX emanaron varios inventos cuya naturaleza marcó el camino que siguieron los avances en telecomunicaciones desde aquel entonces hasta hoy en día.

El primer invento que salió a la luz fue el telégrafo, cuyo padre fue el estadounidense Joseph Henry [2]. Este físico aplicó los hallazgos de Oersted, Ampère y Faraday y concibió un prototipo en 1829 que era capaz de establecer una comunicación entre dos partes a través de la corriente eléctrica: el telégrafo eléctrico. Este invento consistía únicamente en abrir o cerrar un circuito eléctrico mediante un interruptor, que podía o mover un estilete que hacía una marca en una bobina de papel o alimentar un dispositivo productor de ruido. Este prototipo todavía era meramente experimental y no podía implementarse como tal, y de hecho hicieron falta varios modelos como el telégrafo de los alemanes Gauss y Weber o el del ruso Schilling para desarrollar un telegráfo que pudiese ser implantado como red de comunicaciones. De hecho, el primero que consiguió establecer una conexión entre dos ciudades y que posteriormente sería el responsable de la implantación del telégrafo como medio de comunicación más rápido y eficiente en todo el país fue el americano Samuel Morse, cuyo modelo de telégrafo podemos ver en la Figura 1-3. En 1844 realizó la primera transmisión telegráfica entre Washington y Baltimore, codificando el mensaje en su famoso código Morse, en uso todavía hoy en día en varios sistemas de telecomunicaciones. Cabe destacar también el empresario Cyrus West Field, que tendió el primer cable telegráfico en 1858 entre América y Europa, uniendo por primera vez a los dos continentes [1].



Figura 1-3 Prototipo de telégrafo [5]

Poco después de los beneficios que aportó el telégrafo se patentó en 1876 el teléfono, que permitía hablar a través de un conducto eléctrico. La invención estuvo sujeta a polémica hasta 2002 ya que la primera patente la obtuvo el famoso estadounidense Alexander Graham Bell, aunque en dicha fecha el Congreso de Estados Unidos atribuyó la invención, reflejada en la Figura 1-4, al italiano Antonio Meucci, quien lo diseñó en 1854, pero supuestamente sufrió una expropiación de su invento que no pudo refutar por falta de medios frente al poder que poseía Bell por el éxito rotundo que obtuvo el

teléfono en pocos años. Este originalmente consistía en la transmisión de vibraciones mecánicas del aire (voz) en vibraciones eléctricas en un circuito mediante un electroimán. Estas vibraciones, al actuar sobre el electroimán en la estación receptora, producían una copia de las vibraciones mecánicas originales mediante una membrana y por lo tanto la voz se transmitía. Tal fue el éxito del teléfono que, en tan solo dos años después de la publicación de la patente, se inventaron los primeros tableros de conmutación para establecer el enlace entre diferentes usuarios, precursores de la conmutación de la comunicación [2].

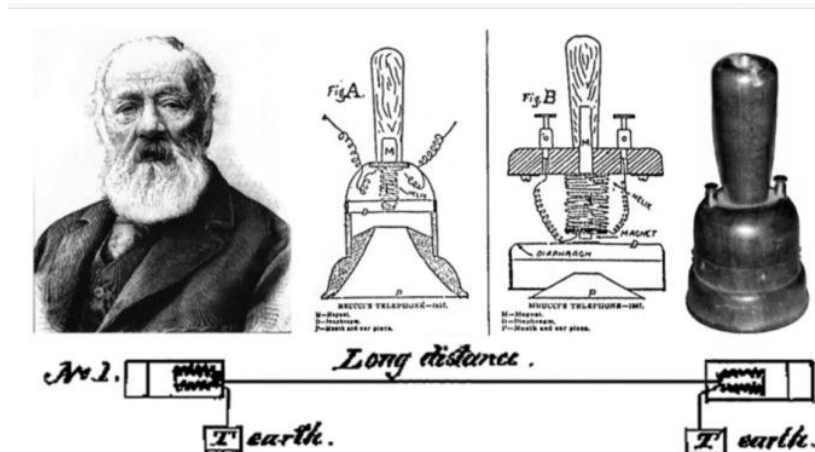


Figura 1-4 Retrato de Antonio Meucci junto a su invento [6]

El Siglo de Oro de las telecomunicaciones lo cerró el alemán Heinrich Hertz, padre de los sistemas de telecomunicación mediante el uso de ondas electromagnéticas. Este físico alemán buscaba demostrar empíricamente la teoría formulada por el escocés Maxwell acerca de la propagación del campo eléctrico y el campo magnético en el espacio en forma de onda.

Hertz fue el primero en crear dicho tipo de ondas de forma artificial, así como los equipos necesarios para la transmisión y recepción, y extrapoló su comportamiento a todas las ondas electromagnéticas conocidas, además de verificar varios postulados de las ecuaciones de Maxwell tales como la velocidad de propagación, cuyo valor medido en los experimentos se parecía mucho a los enunciados por Maxwell, es decir aproximadamente 300 000 km/s [2] [7]. Publicó sus estudios sobre las ondas electromagnéticas en 1888, estableciendo el pilar de todos los avances venideros en telecomunicaciones por ondas electromagnéticas, todavía a día de hoy en desarrollo, y permitió por lo tanto la implantación de uno de los medios de telecomunicación más usados a día de hoy. Cabe destacar el trabajo del italiano Guglielmo Marconi, gran impulsor de la radiodifusión y telegrafía sin hilos. El hecho más relevante de su carrera fue sin duda que el 12 de Diciembre de 1901 estableció un radioenlace entre el continente europeo y el americano, enlazando por primera los dos continentes sin necesidad de una infraestructura faraónica, como se podría considera el cable de 4400 km de largo para comunicaciones telegráficas de la Atlantic Telegraph Company, tendido de la mano de Field, sino tan solo con una antena formada por dos mástiles de 48 metros de alto y separados 60 metros entre sí, emitiendo a una potencia de 15 kW [1].

A partir de los inventos del siglo XIX, que marcaron una brecha temporal e iniciaron un ciclo de desarrollo intensivo en materia de telecomunicaciones, nacieron todas las tecnologías de las que disponemos hoy en día y proliferaron en infinidad de campos [1]. Además, la accesibilidad aumentó en gran medida, popularizando estos medios de comunicación. El principio del siglo XX está marcado por infinidad de mejoras en los inventos previamente mencionados, como por ejemplo las modulaciones en amplitud (AM) en 1918 o en frecuencia (FM) en 1925, estas últimas aportando mayor robustez al enlace y viéndose menos influenciadas por el ruido, y que permitieron aumentar en gran medida la tasa de transmisión mediante un mismo canal [2] [7].

La década de los 30 fue también digna de mención ya que en 1932 se realizaron los primeros estudios sobre el RADAR (*Radio Detection And Ranging*) y en 1939 se empezó a emitir televisión comercial, de mano de la NBC, aunque la televisión fuese inventada en 1925 por el escocés John Logie Baird, reflejo de la suficiente capacidad de las infraestructuras de telecomunicación de la época.

Al finalizar la Segunda Guerra Mundial, una gran cantidad de nuevas tecnologías necesitaron del desarrollo de las telecomunicaciones para su implantación. En particular, en el contexto de la Guerra Fría, en la Guerra de las Galaxias (o *Star Wars*), inicialmente emprendida por la URSS con el lanzamiento del Sputnik 1, las telecomunicaciones jugaron un rol primordial ya que de ellas dependían el éxito o el fracaso de las diferentes misiones espaciales. Dicho satélite tenía una sola función que era la de transmitir a la Tierra un pitido en forma de onda electromagnética, para que todo el mundo, tanto soviéticos como occidentales, pudiesen comprobar su existencia y operatividad.

A este acontecimiento le sucedieron hechos tan trascendentales como el primer viaje en el espacio emprendido por el cosmonauta Yuri Gagarin el 12 de abril de 1961 o el primer alunizaje el 20 de Julio de 1969, hechos que hubiesen sido inconcebibles sin los correspondientes módulos de telecomunicaciones enlazando con las bases en Tierra.

A partir de 1970, la Tercera Revolución Industrial vino de la mano del auge de la informática y en particular de los ordenadores, cuyo primer prototipo, el Harvard Mark I de la empresa IBM, vio la luz en 1944 gracias al americano Howard Aiken. Dichos ordenadores tuvieron un desarrollo muy intensivo, con hechos destacados como la invención del transistor en 1948 de mano de los estadounidenses John Bardeen y Walter Brattain o la aparición del primer ordenador comercial en 1963.

Dichos inventos no cobraron la relevancia primordial que tienen ahora hasta que, en 1969, el Departamento de Defensa de EEUU mediante la agencia DARPA, todavía en el contexto de la Guerra Fría y bajo amenaza de misiles nucleares balísticos, creó Arpanet, precursora de Internet [8]. Por temor a que, bajo un holocausto nuclear, las infraestructuras de computación fuesen destruidas únicamente por la potencia radiada, se decidió descentralizar los recursos, interconectándolos. Desde entonces, esa red fue creciendo y el número de nodos aumentó de manera exponencial, primero en el ámbito militar, luego en el académico y por último a nivel particular, pasando de 4 nodos en 1969 a más de 1200 millones a día de hoy. Esta explosión de la red, junto con algunos inventos como el teléfono móvil, requirieron de una intensificación de las infraestructuras y del desarrollo de nuevas tecnologías, como la UMTS (*Universal Mobile Telecommunications System*) cuya quinta generación (5G) se encuentra hoy en día en desarrollo [8] [9].

Por último, la fibra óptica, predecesora directa de los sistemas de telecomunicaciones como los que desarrolla este Trabajo Fin de Grado [10], supuso una revolución en las telecomunicaciones. El empleo de la luz como portadora de información ofrece la ventaja de anchos de banda elevados, debido a las frecuencias empleadas (comprendidas entre 0.1 THz y THz), lo cual se traduce en una tasa de transmisión muy elevada [1] [10]. Sin embargo, no se concebía la posibilidad de su empleo ya que la luz sufría una atenuación enorme en la atmósfera debido a los meteoros tales como niebla, nieve, lluvia, etc. Desde el siglo XIX, diferentes científicos estudiaron el confinamiento de un haz de luz en una guía mediante el principio de reflexión interna total. Se estudió en primera instancia el confinamiento en una placa de cristal lisa de la mano del físico Augustin-Jean Fresnel en 1820. Hicieron falta noventa años para que otros dos físicos, Demetrios Hondros y Peter Debye, estableciesen las ecuaciones que rigen el confinamiento de la luz en un “cable de vidrio”, y no fue hasta 1952 que el físico Narinder Singh Kapany inventó la fibra óptica como la entendemos hoy en día [1] [11].

Esta se trataba de una guía con base de silicio flexible con un índice de refracción suficiente como para permitir la reflexión interna total, pero pudiendo emplearla con mayor facilidad, debido a la flexibilidad. De hecho, las primeras aplicaciones de la fibra óptica no fueron ni mucho menos en el

campo de las telecomunicaciones sino en otros campos como en el de la medicina. En particular, hasta la primera mitad del siglo XX los médicos usaban un utensilio denominado gastroscopio, que por aquel entonces consistía en una barra alargada de metal con espejos en su interior y que servía para observar el estómago de los pacientes [1]. Dicha barra se introducía por vía oral hasta donde el médico necesitase observar, lo cual resultaba extremadamente incómodo.

En 1956, Lawrence Curtis, de la Universidad de Michigan, empleo la fibra óptica para observar el estómago de sus pacientes, inventando el actual endoscopio, mucho más versátil y cómodo. Fue de hecho el propio Curtis el que se dio cuenta de un fenómeno reseñable a la hora de emplear la fibra óptica, y es que se dio cuenta de que, si usaba un mechón de fibras y las apretaba, la luz no se transmitía de una punta a otra de las fibras, sino que parecía “gotear” entre estas. Este fenómeno se debía a que, al estar los materiales de mismo índice de refracción en contacto, la luz podía filtrarse de un conducto a otro ya que no sufría un cambio en la velocidad de propagación [1]. Dicho descubrimiento explica por qué a día de hoy las fibras están constituidas por una fibra de índice de refracción bajo extruida en el seno de otra de mayor índice.

Pero la fibra todavía presentaba su principal desventaja: el alcance. Como se comentará posteriormente en el apartado 2.1.1, se disponían de vidrios con enormes pérdidas. En 1966, Charles Kao y George Hockham realizaron un estudio que marcó el comienzo del uso de fibra óptica en telecomunicaciones. En concreto, estos dos científicos de los laboratorios de Standard Communications de Inglaterra estudiaron la viabilidad del empleo de fibra óptica en telefonía y la limitación de esta en cuanto a alcance se refiere. Llegaron a la conclusión de que la fibra podía ser perfectamente empleada en telecomunicaciones y que presentaba un gran potencial, ya que las limitaciones en alcance se debían principalmente a las impurezas de agua y metales presentes en la propia fibra [11].

Desde ese momento, diferentes investigaciones llevaron a la creación de fibras con cada vez menos pérdidas, como por ejemplo la que obtuvieron los científicos Robert Maurer, Donald Keck y Peter Schultz aplicando impurezas de titanio en sílice para aumentar la transparencia, hasta que en 1977 la compañía General Telephone and Electronics puso en práctica por primera vez a la fibra óptica. Se realizó una llamada telefónica en Long Beach, California, mediante un sistema basado en fibra óptica y que ofrecía una capacidad de 6 Mb/s. Desde entonces, la fibra óptica se ha adoptado como uno de los sistemas de telecomunicaciones preferentes, evolucionando principalmente en la creación de nuevos materiales para las fibras con pérdidas reducidas, así como nuevas técnicas de producción [11].

Se puede concluir por lo tanto que, desde la Antigüedad hasta nuestros días, el desarrollo de las telecomunicaciones ha sido uno de los pilares del desarrollo de la sociedad tal y como la concebimos hoy en día. No solo ofrece claras ventajas en las esferas políticas y militares, como lo fue desde un principio, sino que también constituye un gran interés económico para empresas y un gran interés tecnológico para otros campos de investigación como es la industria aeroespacial. Además, el desarrollo de dichas tecnologías ha visto una evolución de vertiginoso crecimiento, transformando la sociedad en y su manera de interactuar en cada momento.

1.2 Contexto y objetivos

Este Trabajo Fin de Grado se centra en el diseño de un prototipo de un sistema de telecomunicaciones basado en la tecnología LASER. Dicho tipo de telecomunicaciones se engloban en las denominadas telecomunicaciones FSOC (*Free Space Optical Communications*), y representan uno de los ejes de investigación en el campo de las telecomunicaciones. Estas tecnologías tienen un principio de funcionamiento que se asemeja al de la fibra óptica, y presentan las mismas ventajas, añadiendo el hecho de que no requieren una guía entre estaciones emisora y receptora [10]. En realidad, el sistema sería de facto el mismo, pero en vez de disponer un tendido de fibra de vidrio, el

medio sería la propia atmósfera, consiguiendo así reducir los costos de la conexión, multiplicando las posibilidades de su empleo.

Las ventajas que presenta dicho sistema son la capacidad que ofrece (aunque no llega a la de la fibra óptica), así como la reducción del coste en infraestructuras ya que no se necesita de una guía, la libertad de empleo sin necesidad de licencias como es el caso de la radiodifusión, y la versatilidad, ya que puede implementarse en multitud de escenarios. Pero la ventaja más reseñable que concierne en particular a las plataformas navales es la discrecionalidad.

A nivel operativo, muchas veces es necesario mantener una comunicación muy limitada para permanecer oculto, ya que todo empleo del espectro electromagnético conlleva el riesgo de ser detectado por el enemigo. Hoy en día, se protegen las comunicaciones con diferentes cifrados, y se consigue cierta discreción mediante el empleo de medios como el satélite o las comunicaciones visuales, pero el empleo de los satélites suele ser limitado y las comunicaciones visuales requieren de mucha proximidad y son lentas. Siendo así, normalmente hay que sopesar la necesidad de la comunicación de unidades con la exposición al enemigo. Es por ello que resulta interesante disponer de un medio de comunicación que pudiese ser empleado en todo momento, sin posibilidad de ser interceptado, ni siquiera detectado, por parte del enemigo. La discreción que ofrece el Láser, por su directividad inherente a su naturaleza física, permite obtener el grado necesario de discreción ya que sería muy complejo detectar dicho haz, atendiendo a la necesidad táctica en cuanto a comunicaciones sin restricciones.

Del principio de funcionamiento de un sistema FSOC podemos deducir que será necesario un mecanismo de apuntamiento que permita la comunicación full-duplex. Uno de los subsistemas vitales de un sistema FSOC es el sistema de adquisición y seguimiento, que permite dirigir el haz de luz estrecho en dirección de la estación receptora. Este subsistema a su vez se caracteriza tanto por la naturaleza de este, como la plataforma empleada. Dependiendo de las necesidades del sistema FSOC, normalmente relativas a su uso, se empleará un tipo u otro de seguimiento, así como una plataforma que no limite las ventajas que ofrece el subsistema de seguimiento.

El objetivo por lo tanto de dicho trabajo será el de diseñar un sistema de telecomunicaciones FSOC en una plataforma naval, y en particular el subsistema de adquisición y seguimiento, que asegure la comunicación entre las dos partes, así como la plataforma que monte dicho sistema. Para este proyecto se diseñará un sistema que permita la comunicación en condiciones climatológicas no adversas, lo que representa una de las principales trabas de estos sistemas y a día de hoy es objeto de estudio de los grandes promotores de este tipo de sistemas, pero se tendrá en cuenta el dinamismo de las plataformas que montarían dichos sistemas. En primera instancia, se estudiará un sistema de seguimiento de la estación receptora que consiste en el seguimiento de un determinado patrón de luces mediante análisis de video, para luego diseñar una plataforma donde implantar un sistema FSOC al completo en conjunto con el AF González del Tánago Landín, encargado este último de diseñar la carga útil del sistema en su Trabajo Fin de Grado *Estudio y desarrollo de un sistema de comunicaciones punto a punto basado en LASER para buques de la Armada*.

2 ESTADO DEL ARTE

2.1 Sistemas FSOC

2.1.1 Características generales

Los sistemas FSOC (*Free Space Optical Communications*), también denominados FSO (*Free Space Optics*), se asemejan a los sistemas basados en la fibra óptica [10], aunque con ciertas diferencias. Dichos sistemas comparten el empleo de la luz visible como portadora de información, permitiendo altas velocidades de transmisión [12].

La fibra óptica fue desarrollada en los años 60 y consiste en guiar un haz de luz (normalmente en el espectro visible, aunque pueden usarse frecuencias en el Infrarrojo cercano), normalmente coherente (LASER), hasta un receptor. Dicho haz de luz, mediante una modulación, transmite información, obteniendo un sistema de telecomunicaciones. La guía consiste en un cable de sílice extruido en el seno de un material con un índice de refracción más bajo, consiguiendo así la reflexión interna total y atrapando el haz de luz dentro del núcleo. Al inicio del desarrollo de esta tecnología, se disponían de fibras con pérdidas altas (alrededor de 100 dB/km), permitiendo alcances reducidos, pero el desarrollo de nuevos materiales (con pérdidas de 0,0001dB/km) permite hoy alcanzar distancias mucho mayores, como los cables submarinos de fibra óptica [10] [11].

Cabe mencionar también que, como su predecesor en el ámbito radio, se han desarrollado técnicas de multiplexación para enviar varios canales de información en un mismo conducto a través de la modulación. En concreto, se emplea el análogo de la modulación en frecuencia en las emisiones radio que es el WDM (*Wavelength-Division Multiplexing*) y en particular el DWDM (*Dense WDM*). Dicha modulación consiste únicamente en emplear diferentes longitudes de onda para diferentes canales de comunicación (es decir, diferentes colores) a través de una única guía, multiplicando así la tasa de información transmitida. La DWDM es un caso concreto en el que se emplean 8 o más longitudes de onda diferentes a través de un mismo conducto.

Debido al amplio desarrollo desde su creación, la fibra óptica esta notablemente más desarrollada que las FSOC, aunque también tuvo que hacer frente a los problemas que presentan las FSOC hoy en día, sobre todo respecto al alcance, pero ofreciendo unas prestaciones más que interesantes, sobre todo en cuanto a tasa de transmisión, discreción en la transmisión y libertad de uso en cuanto a ancho de banda en interferencias con otras emisiones se refiere. En particular, se puede mencionar el caso del cable de fibra óptica con más capacidad del mundo, el i2iCN entre Singapur y Madrás con ocho pares de fibras con 100 canales en cada uno con capacidad de transmitir a 10 Gb/s por canal mediante la modulación DWDM, que resulta en una capacidad total de diseño de 8,4 Tb/s con una ratio de error (BER: *Bit Error Rate*) de 10^{-13} . De hecho, se prevén nuevos cables con capacidad por encima de los 10

Tb/s para los próximos años. Estos avances son posibles gracias al desarrollo de nuevas fibras con menores pérdidas, pero también a nuevos receptores más sensibles, como por ejemplo los receptores que usan un oscilador local, aumentando notablemente la sensibilidad del receptor y pudiendo así emplear más canales en la misma guía y transmitir más información [10].

Las FSOC heredan de la fibra óptica los componentes básicos de la transmisión y recepción, como podemos ver en la Figura 2-2 (a). En concreto, el transmisor recibe una señal codificada en unos y ceros que revierte al modulador. Este modula un determinado haz de luz proveniente de una fuente que puede ser o un LED (*Light Emitting Diode*) o un LD (*Laser Diode*), mediante diferentes tipos de modulación.

En las FSO, existen multitud de modulaciones que se pueden poner en práctica, divididas en dos familias: SCM (*Single Carrier Modulation*) y MCM (*MultiCarrier Modulation*), como podemos ver en la Figura 2-1. Las SCM son las más sencillas de aplicar, ya que requieren de un nivel de SNR muy bajo y no suponen un gran desafío a la hora de demodular y procesar.

Entre las SCR más comunes se encuentran la OOK (*On-Off Keying*: modulación más simple en la que un “1” se codifica mediante luz y un “0” mediante ausencia de luz), la PWM (*Pulse Width Modulation*: los diferentes símbolos se codifican mediante el ancho de pulso), la PAM (*Pulse Amplitude Modulation*: el ancho de pulso y el PRF(*Pulse Repetition Frequency*) se mantienen constantes, variando la amplitud del pulso) o la PPM (*Pulse Position Modulation*: el ancho de banda y la amplitud del pulso se mantienen, variando el instante en el que se transmite el pulso dentro de una ventana concreta, es decir la posición). Las MCM son modulaciones de enorme ancho de banda, empleando una gran variedad de modulaciones del tipo OFDM (*Orthogonal Frequency-Division Modulation*) y algunas innovadoras (HCM (*Hadard Encoded Modulation*), WPDM (*Wavelet Packet Division Multiplexin*),...), aunque a día de hoy solo han resultado aplicables en condiciones de laboratorio, alcanzando hasta un ancho de banda de 224 Gb/s [12]. Por último, existen otra familia de modulaciones relacionadas con el color de la luz (CSK (*Color Shift Keying*), CIM (*Colour Intensity Modulation*), MM (*Metametric Modulation*)) , que se explicará en el punto 2.2.3, en concreto en el ámbito de LiFi [13].

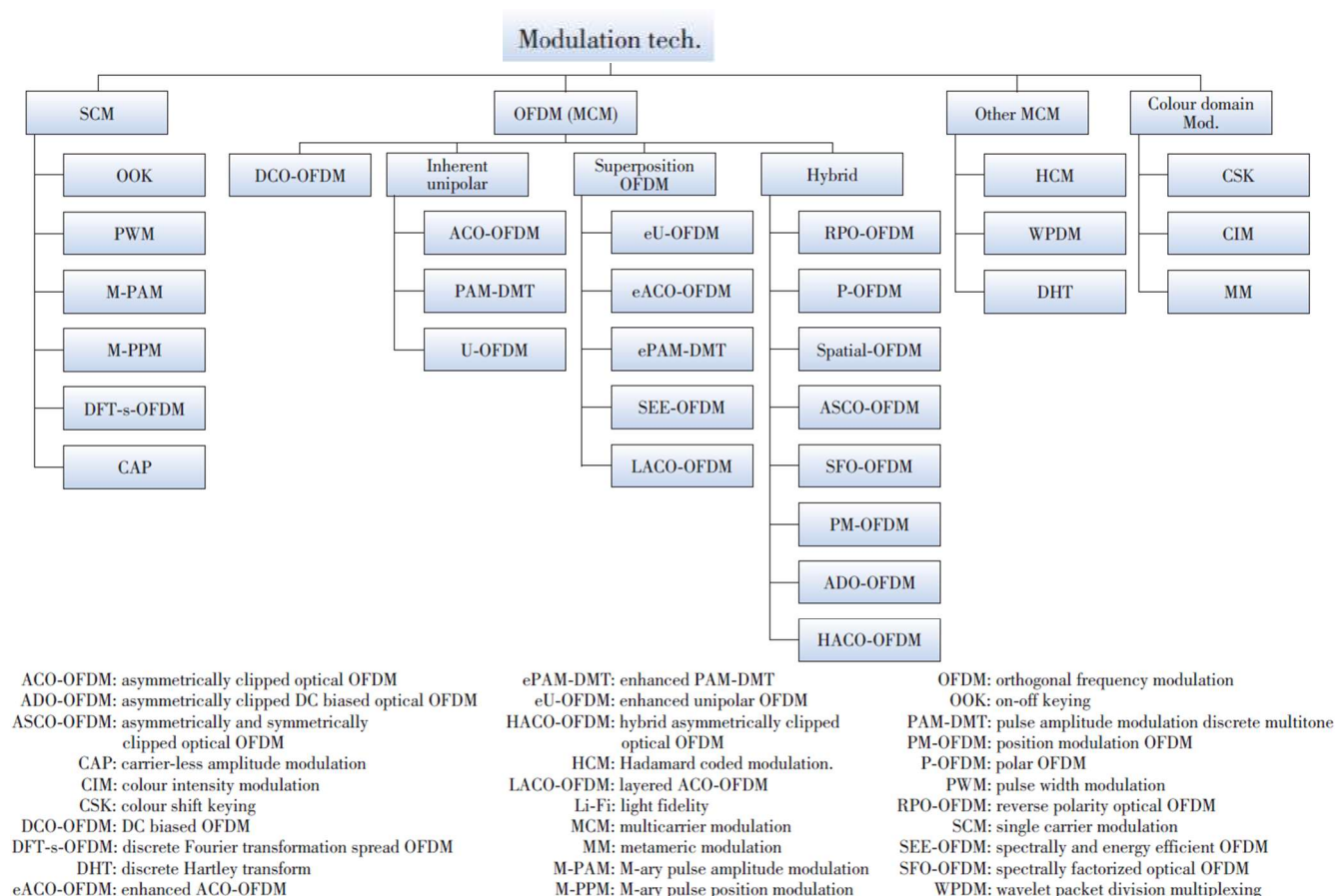
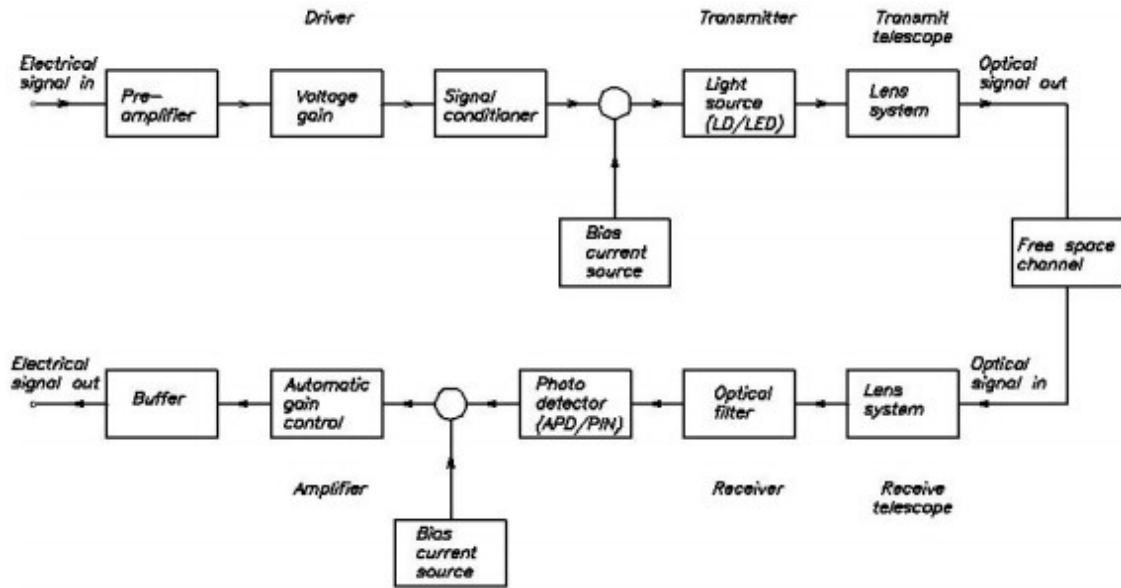
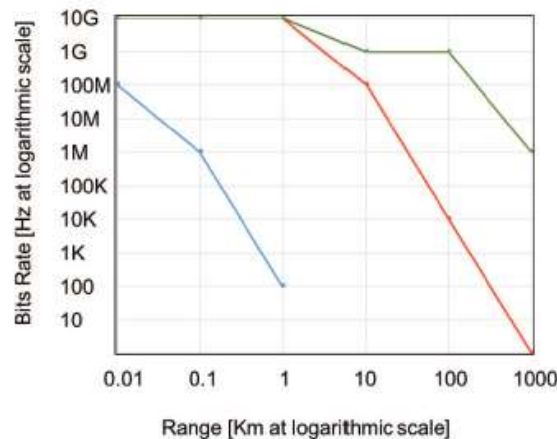


Figura 2-1 Diagrama de modulaciones en FSOC [13]

Una vez modulado, el haz de luz se enfoca mediante una lente colimada, creando un haz de rayos paralelos, que en recepción se vuelve a enfocar en un punto mediante otra lente, como podemos ver en la Figura 2-2 (a). En dicho punto, como en la fibra óptica, se sitúa uno o varios fotodetectores que transforman la señal óptica en una señal eléctrica, que se amplifica, procesa y distribuye, completando así la comunicación. Cabe reseñar que, en las FSOC, las señales que se reciben suelen ser bastante tenues, para lo que se emplean esquemas de amplificación de transimpedancia (TIA: *TransImpedance Amplifier*), que constan de un amplificador operacional que transforma la señal, que está en forma de corriente, en voltaje, lo cual impide pérdidas excesivas en el circuito. A esto hay que sumarle que en las FSOC siempre habrá un compromiso entre alcance y ancho de banda. Un FSOC operando a gran distancia tendrá un ancho de banda menor que uno operando en exactamente las mismas condiciones y en el mismo medio, pero a menor distancia, debido principalmente a que la relación señal a ruido o SNR (*Signal-to-Noise Ratio*) es proporcional a la tasa de recepción, como podemos ver en la gráfica de la Figura 2-2 (b).



(a)



(b)

Figura 2-2 (a) Diagrama general del funcionamiento de un sistema FSOC [10], (b) Gráfico de máximo ancho de banda en relación a la distancia (3 casos) [14]

Por último, aparte de las lentes convencionales para enfocar los distintos haces que se empleen, tanto de balizamiento como de la propia transmisión, se puede recurrir a una mejora en el módulo óptico para combatir una de las limitaciones mencionadas en el apartado 2.1.2, que es la refracción turbulenta. En efecto, se puede emplear la óptica adaptativa, como vemos en la Figura 2-3 (a), empleada sobre todo en los telescopios astronómicos, que consiste en detectar el frente de onda y corregir las aberraciones mediante espejos MOEMS (*Micro Opto-ElectroMechanical System*), cuya descripción en detalle se lleva a cabo en el apartado 2.3.2. Este sistema detecta las perturbaciones producidas por la atmósfera mediante un sensor de frente de onda y activa unos espejos deformables (espejos MOEMS) con resolución angular de nanoradianes (10^{-9} rad) que mitigan estos efectos, mejorando significativamente la potencia recibida y la BER (*Bit Error Rate*). Sin embargo, si el centelleo es muy intenso, se requiere de una óptica adaptativa no convencional, como la de la Figura 2-3 (b), que incluye un radioenlace con la estación receptora para corregir entre las dos las

aberraciones producidas por la atmósfera.

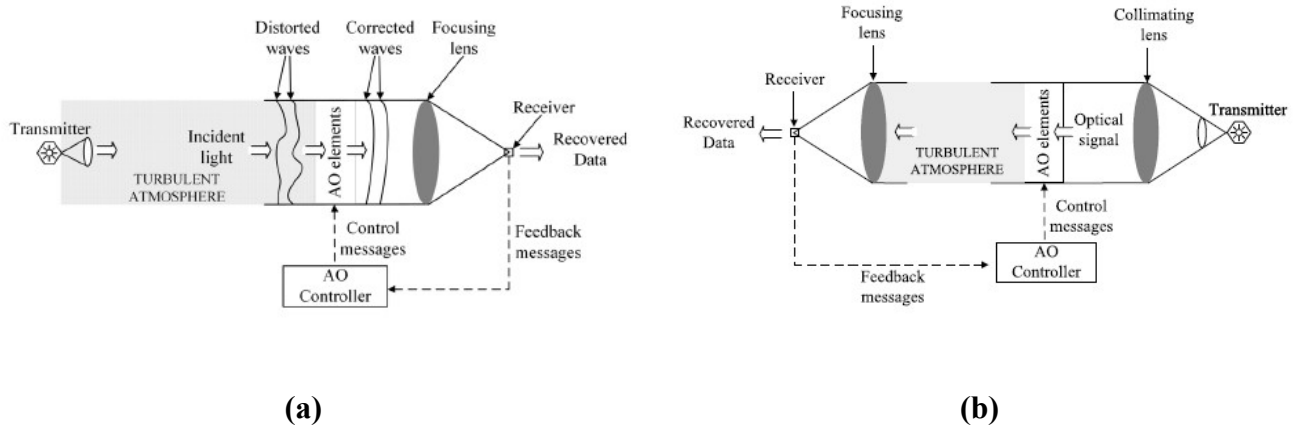


Figura 2-3 (a) Esquema de funcionamiento de un sistema de óptica adaptativa convencional [14] , (b) Esquema de funcionamiento de un sistema de óptica adaptativa no convencional con enlace por radiofrecuencia auxiliar [14]

2.1.2 Limitaciones

Los sistemas FSOC tienen como principal limitación los factores ambientales [10] [16] [15] [17]. Dependiendo del medio en el que se emplee un FSOC, este tendrá que hacer frente a todas o parte de estas limitaciones, siendo la regla general que, a mayor altura, menos restricciones, como podemos ver en la Figura 2-4. Cuanto más nos alejamos de la superficie terrestre, menos limitaciones tenemos y más eficientes son los enlaces. En concreto, veremos las pérdidas por absorción, difracción y turbulencia refractiva, relativas a la atmósfera y siempre presentes en comunicaciones con alguna estación en la superficie terrestre, así como su intensificación debido a los meteoros atmosféricos. Veremos también el efecto del desalineamiento o la propia divergencia del haz, inherente al propio sistema.

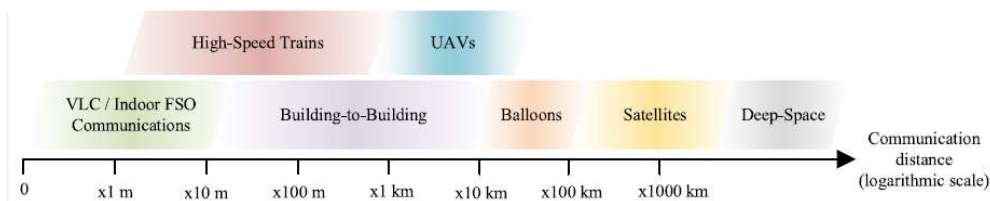


Figura 2-4 Ámbito de aplicación de las FSOC en función de alcance

En primer lugar, la absorción de la atmósfera repercute en gran medida en el alcance del sistema. En concreto, determinados gases presentes en la atmósfera como el vapor de agua (H_2O), el dióxido de carbono (CO_2) o el metano (CH_4) absorben la energía lumínica del Laser en determinadas frecuencias, pudiendo tornar la atmósfera en un medio suficientemente opaco en función de su concentración como para impedir la comunicación, como se puede apreciar entre la Figura 2-5 (a) y la Figura 2-5 (b). Además, cabe reseñar que es complicado predecir cuando estos gases se interponen en la comunicación, volviendo este fenómeno una de los principales frenos en el desarrollo de estas tecnologías [10].

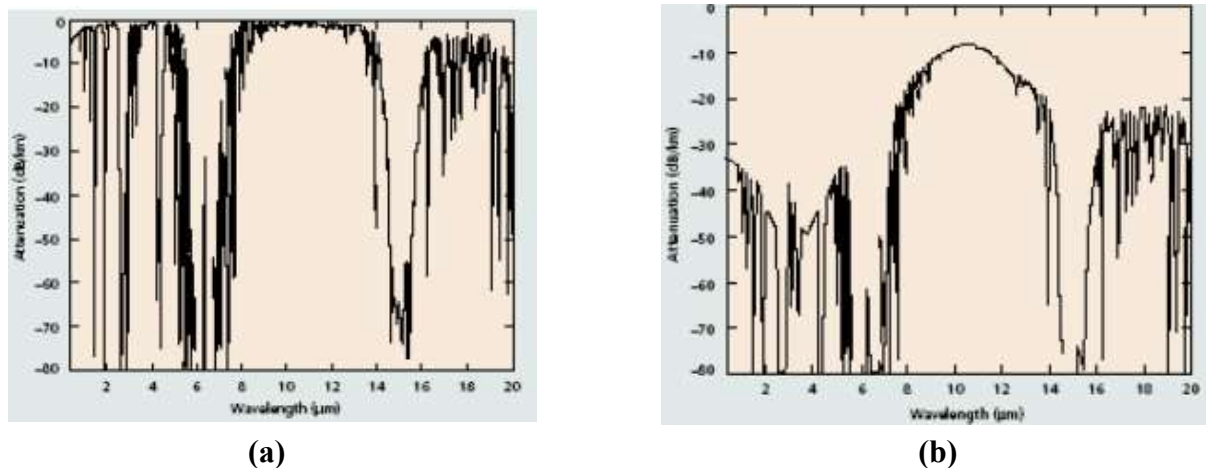


Figura 2-5 (a) Gráfico de atenuación específica de la atmósfera (niebla) [10], (b) Gráfico de atenuación específica de la atmósfera (condiciones normales) [10]

Por otro lado, la dispersión es otro factor a tener en cuenta ya que puede imposibilitar por completo el enlace [10] [15] [16]. Este fenómeno puede darse de dos formas: dispersión Mie y dispersión geométrica. La dispersión Mie corresponde a la dispersión que sufre un haz al encontrar en su dirección de propagación un obstáculo de dimensiones aproximadamente iguales a la longitud de onda del haz, mientras que la dispersión geométrica se corresponde con la dispersión sufrida cuando el haz se encuentra un obstáculo de dimensiones significativamente más grandes (más de un orden de magnitud de diferencia) en su dirección de propagación. La primera difiere de la segunda en que la dispersión producida por la partícula varía mucho en función de las dimensiones de esta, siendo por lo tanto muy aleatoria la magnitud de la dispersión en esta zona, mientras que en la otra es mucho más constante y predecible (si se conoce la posición del obstáculo, cosa que no ocurre en el análisis de las FSOs y por lo tanto la dispersión geométrica también se da con cierta aleatoriedad).

Este fenómeno puede llegar a ser perjudicial en el caso de que algún rayo se desvíe de su trayectoria y llegue al receptor, creando una interferencia destructiva. Es decir, si un rayo de luz que no iba en dirección al receptor se ve desviado y llega a este, llegará con cierto desfase respecto al rayo que sigue una trayectoria directa, lo cual se traduce en una diferencia en la fase que llega a ser destructiva si es superior a la mitad de la longitud de onda. Cabe reseñar que los obstáculos del tamaño de la longitud de onda de la luz (entre $0,4 \mu\text{m}$ y $0,8 \mu\text{m}$, como las gotas de llovizna) o superiores (como los copos de nieve) son omnipresentes en la atmósfera terrestre, en forma de partículas tales como polvo, gotas de agua, cristales de hielo, ... Su concentración determinará por lo tanto el nivel de atenuación de la atmósfera, siendo un ejemplo niebla, que no es más que una concentración de partículas hechas por moléculas de agua, y cuyo efecto de dispersión también influye en el espectro reflejado en la Figura 2-5 (b).

Otro fenómeno que influye en gran medida a la hora de establecer un enlace estable en el tiempo con una tasa de error relativamente baja es la refracción turbulenta. Para entender este fenómeno, primero se debe de entender que la atmósfera, como todo medio físico, tiene un índice de refracción asociado. Dicho índice varía ampliamente en función de varios factores como presión y temperatura, haciendo que varíe ampliamente de una zona geográfica a otra, o en una misma, pero a altitudes diferentes. Además, un cambio en el índice de refracción se traduce por un cambio en la dirección del rayo luminoso, y por lo tanto perjudica el sistema. En una zona meteorológicamente estable y donde el flujo de aire tiene un comportamiento más laminar, este hecho tiene menos relevancia, pero en zonas donde existe un régimen turbulento del aire, como el de la Figura 2-6, y coexisten masas de aire de diferentes temperaturas, el índice de refracción varía de un punto a otro del espacio, y en un mismo

punto a lo largo del tiempo. A esto se le denomina refracción turbulenta, ya que el índice de refracción del medio varía de forma imprevisible, dando sobre todo lugar al fenómeno de centelleo [10] [17].

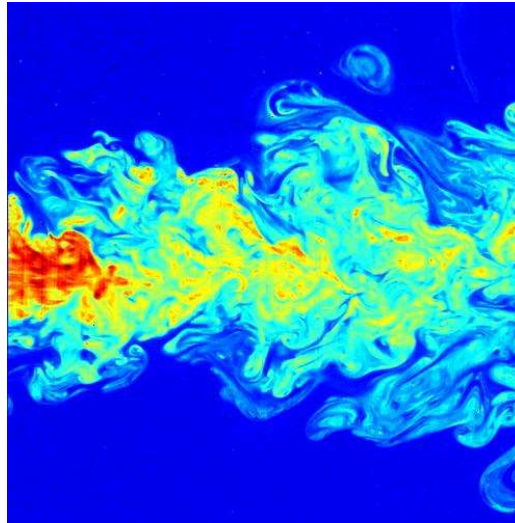


Figura 2-6 Fotografía de la temperatura del aire en régimen turbulento (mediante cámara termográfica) [18]

Sin embargo, el factor más limitante de las FSOC son con diferencia los hidrometeoros [10] [16]. Los hidrometeoros tienen lugar en la troposfera terrestre, capa atmosférica que comprende desde la superficie hasta la tropopausa (cambio del gradiente de temperatura, de negativo a positivo, normalmente a una altitud entre los 8 y los 14,5 km) [19]. Estos acentúan muy significativamente los tres efectos mencionados previamente, imposibilitando en muchos casos las comunicaciones.

Los principales hidrometeoros que incrementan las limitaciones son la niebla, la lluvia y la nieve [16], aunque otros fenómenos como las tormentas de arena o la calima también suponen una limitación notable, así como el viento, cuyo efecto en particular es el centelleo. La niebla se da con la acumulación de moléculas de agua en gotas en suspensión en una masa de aire. Esto ocurre por el enfriamiento de dicha masa húmeda por diferentes procesos (niebla de advección, niebla de radiación,...) hasta unos 3°C por encima del punto de rocío. En este punto, las moléculas de agua empiezan a condensar y a juntarse en pequeñas partículas de agua en suspensión en la masa de aire, alcanzando esta una humedad relativa del 100 %.

La niebla supone el limitante más grave para las FSOC con diferencia ya que, debido a la alta concentración de partículas que son gotas de agua, la absorción y la dispersión intensifican ampliamente su efecto en la propagación de la luz [10] [16]. Un estudio de la atenuación debido a la niebla requiere de multitud de parámetros tales como distribución de tamaños de las partículas, índice de refracción o tamaño de estas, y puede variar de una zona concreta a otra, con lo cual se prefiere la aproximación mediante la visibilidad, que da un valor de atenuación específica del banco en general mediante la fórmula (1) [16].

$$\gamma(\lambda) = \frac{3,912}{V} \left(\frac{\lambda}{550} \right)^{-q} \quad (1)$$

Donde γ es la atenuación específica en dB/km, V es la visibilidad en km, λ es la longitud de onda en nm y q es un coeficiente que depende de la visibilidad V . Para dicho coeficiente, existen dos

aproximaciones, cuyos valores atribuidos a q en función V difieren [16]. En la primera de ellas, q toma los valores reflejados en (2).

$$q = \begin{cases} 1,6 & V > 50 \text{ km} \\ 1,3 & 6 \text{ km} < V < 50 \text{ km} \\ 0,585\sqrt{V} & V < 6 \text{ km} \end{cases} \quad (2)$$

En la segunda aproximación, q toma valores similares, pero tiene en cuenta más casos, como podemos ver en la ecuación (3).

$$q = \begin{cases} 1,6 & V > 50 \text{ km} \\ 1,3 & 6 \text{ km} < V < 50 \text{ km} \\ 0,6 \times \sqrt{V} + 0,34 & 1 \text{ km} < V < 6 \text{ km} \\ 0,5 & 0,5 \text{ km} < V < 1 \text{ km} \\ 0 & V < 0,5 \text{ km} \end{cases} \quad (3)$$

Por otro lado, la nieve también supone un gran impedimento en cuanto a las FSOC ya que provoca mayormente una fuerte dispersión (aunque también absorción), siendo ésta, a diferencia de la niebla, una dispersión geométrica [16]. Los copos de nieve suelen ser más grandes que las gotas de agua, lo que produce una dispersión mayor y por lo tanto la fluctuación de la potencia de la señal a la llegada es mayor. En cuanto a la propia nieve, se tiene que diferenciar entre nieve y aguanieve, ya que los copos de la nieve suelen ser bastante más grandes que los del aguanieve. La atenuación específica (en dB/km) de la nieve tiene como fórmula general la que se refleja en la ecuación (4).

$$\alpha_{nieve} = aS^b \quad (4)$$

Donde S es la tasa de precipitación de la nieve en mm/hr, y a y b son dos coeficientes que dependen de la longitud de onda λ empleada (en nm) y de si se trata de nieve o aguanieve:

- Nieve: $a = 5,42 \times 10^{-3} \times \lambda + 5,4958776$, $b = 1,38$
- Aguanieve: $a = 1,023 \times 10^{-4} \times \lambda + 3,7855466$, $b = 0,72$

El último hidrometeoro que afecta en gran medida a las FSOC es la lluvia. Al igual que la nieve, el principal efecto es la dispersión geométrica, pero con una peculiaridad. La dispersión producida por la lluvia se denomina dispersión no selectiva ya que la lluvia varía mucho en tamaño y número, en tiempo y lugar [16]. Además, las gotas de lluvia se consideran esferas hasta un radio de 1 mm, siendo a partir de ahí considerados elipsoides de revolución aplanados, lo cual da mucha variabilidad en la forma de la gota a la hora de realizar el estudio si se tiene en cuenta que estas van desde los 100µm de radio hasta más de 1 mm. También hay que tener en cuenta que, al igual que la nieve, las gotas de lluvia atraviesan el haz rápidamente, modificando este momentáneamente, y que parte de los rayos que inciden en la gota son refractados y recuperan su dirección inicial al salir de la gota. Siendo así, cabe mencionar que, de los tres meteoros mencionados, es el que menos afecta a las FSOC, aunque ocurre con relativa mayor frecuencia. La fórmula general de la atenuación específica de la lluvia es la de la ecuación (5).

$$\beta_{displuvia} = \pi a^2 N_a Q_{disp} \left(\frac{a}{\lambda} \right) \quad (5)$$

Donde a es el radio de la gota, λ es la longitud de onda, N_a es la distribución de las precipitaciones y Q_{disp} es la eficiencia de dispersión. La distribución N_a se calcula mediante la fórmula (6).

$$N_a = \frac{R}{1,33(\pi a^3)v_a} \quad (6)$$

Donde R es la tasa de precipitación de la lluvia, a es el radio de la gota y v_a es la velocidad máxima de precipitación de una gota. La eficiencia de dispersión representa el ratio de energía dispersa en una dirección frente a la energía incidente en el obstáculo en la misma dirección. Se calcula mediante la fórmula (7).

$$Q_{disp} = \frac{\sigma_s}{\pi r^2} \quad (7)$$

Donde σ_s es la sección equivalente del obstáculo y r es el radio del obstáculo. Por último, según las condiciones climatológicas, los valores de referencia para la atenuación específica de la atmósfera serán los plasmados en la Tabla 1.

Atmospheric Condition	Attenuation
Clear weather	0.2 dB/Km
Urban (because of dust)	10 dB/Km
Rain	2-35 dB/Km
Snow	10-100 dB/Km
Light fog	120 dB/Km
Dense fog	300 dB/Km
Maritime fog	480 dB/Km

Tabla 1 Valores típicos de atenuación específica de la atmósfera para $\lambda = 1550$ nm [15]

Por otro lado, existen fenómenos que también afectan a la potencia en recepción de la señal y que no están relacionados con los meteoros, que son principalmente la *atenuación espacial de la señal* y el *desalineamiento de las estaciones* [10] [15]. La atenuación espacial se da a causa de la divergencia del haz, ya que este no es completamente directivo. Dependiendo del sistema empleado, se obtienen por lo general divergencias muy pequeñas, inferiores al mrad, pero suficientes para que parte del haz emitido no esté dirigido hacia el punto de recepción. Cabe reseñar que la divergencia en los Laser se debe principalmente a la difracción alrededor de la apertura circular por donde se emite el rayo, y cuya divergencia se rige por la fórmula (8) [10].

$$\sin \beta = \frac{1,22\lambda}{D} M^2 \quad (8)$$

Donde λ es la longitud de onda del Laser, D es el diámetro de la apertura y M es un valor adimensional relativo a la arquitectura del módulo transmisor empleado. Una fórmula empírica para la determinación de la divergencia es la (9) [15].

$$D_s = 2d \tan\left(\frac{\theta}{2}\right) \quad (9)$$

Donde para una distancia d , D_s es el diámetro del haz y por lo tanto $\frac{\theta}{2}$ se corresponde con la divergencia. De esta fórmula podemos obtener la relación de potencias de transmisión y recepción en el vacío, reflejada en la fórmula (10).

$$P_r = \left(\frac{D_l}{D_s}\right)^2 P_t \quad (10)$$

Donde P_r es la potencia en recepción, P_t la potencia en transmisión, D_s el diámetro del pulso y D_l el diámetro del receptor, esquematizados en la Figura 2-7.

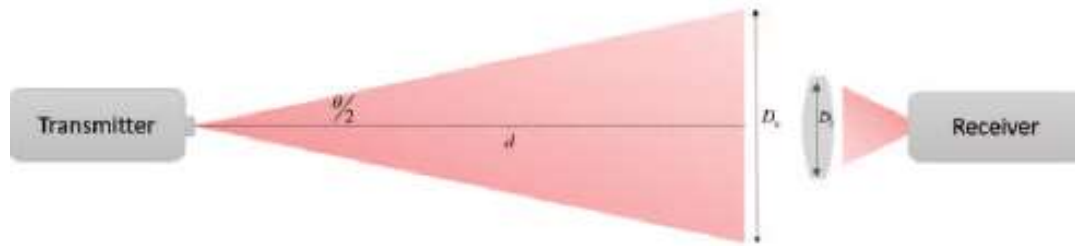


Figura 2-7 Representación del haz frente a la superficie receptora [15]

Una fórmula más general para la relación de potencias que tiene en cuenta tanto este efecto como los atmosféricos se propone en [10], correspondiendo con la fórmula (11).

$$P_r = P_t \left(\frac{A_r}{A_t}\right) T K e^{-\alpha R} \quad (11)$$

Donde P_r es la potencia en recepción, P_t la potencia en transmisión, A_r es el área de la lente de la estación receptora, A_t es el área transversal del haz, T es la eficiencia combinada del transmisor y el receptor, siendo esta el ratio entre la energía luminosa aprovechada por el elemento receptor y la energía captada por la lente, K es un parámetro que equivale a 1 para los Láser y a una fracción para los LED, R es la distancia del enlace y α es la atenuación específica de la atmósfera

Por último, cabe destacar el error que produce el desalineamiento de las estaciones. Resulta trivial deducir el efecto que tendrá sobre la comunicación el desajuste entre estación emisora y receptora, ya que la interrupción, aunque momentánea, del enlace por un desajuste se traduce en una pérdida significativa de la información y por lo tanto reduce la eficiencia del sistema, poniendo a la plataforma de dichos sistemas de telecomunicación a un nivel primordial, ya que de estas dependen en primer lugar el éxito o el fracaso de un enlace [10].

2.2 Aplicaciones de las FSOC

2.2.1 Desarrollo inicial: telecomunicaciones espaciales

Desde la década de los 90 se ha experimentado con la posibilidad de transmitir información mediante un rayo Láser en el ámbito espacial. En efecto, se detectó con éxito una emisión Láser a 6 millones de kilómetros, en concreto a la sonda americana Galileo en el año 1992. Desde entonces, la intercomunicación entre satélites y entre satélites y aeronaves a gran altitud ha experimentado un auge que perdura a día de hoy, resultando en multitud de proyectos que suponen a día de hoy los sistemas de mayor entidad en este campo de las telecomunicaciones.

La primera aplicación de las FSOC como medio de comunicación la realizó la Agencia Espacial Europea (ESA: *European Space Agency*), entre los satélites Artemis y SPOT-4 en noviembre del 2001 [20]. Esto marcó el inicio de las FSOC como tal, y supuso el primer hito relevante en cuanto a comunicación intersatelital vía Láser. De hecho, esta se podía establecer con mayor facilidad que en la superficie terrestre debido a las condiciones ideales de propagación en el vacío, alcanzando grandes distancias sin atender a ninguna de las trabas mencionadas en el apartado 2.1.2, a excepción de la alineación emisor-receptor. En diciembre de 2005, la ESA volvió a hacer historia con la primera transmisión bidireccional entre el satélite Artemis, en órbita geoestacionaria a 36 000 km de altitud, y un avión Mystère 20 equipado con el sistema LOLA (*Liaison Optique Laser Aéronavale*) y un Láser Sylex. Dicho avión estableció comunicación con el satélite en dos vuelos distintos, uno a 6 000 m de altura y otro a 10 000 m [20]. A dicha altura, la atmósfera es razonablemente transparente para la luz visible, pero ya hubo que idear un sistema de seguimiento preciso de la aeronave.

En la segunda década del siglo XXI han visto la luz multitud de sistemas de telecomunicaciones espaciales ópticos, como por ejemplo el sistema implementado a bordo del LADEE (*Lunar Atmosphere and Dust Environment Explorer*) [21]. En concreto, uno de los cuatro instrumentos de dicha sonda fue el LLCD (*Lunar Laser Communication Demonstration*) que durante el fin de 2013 y principios del 2014, proporcionó una comunicación bidireccional fiable a 385 000 km de la Tierra de hasta 622 Mb/s en transmisión desde el módulo espacial, y hasta 20 Mb/s en recepción. El módulo, de 65 libras de peso, consistía en una plataforma formada por un cardan de dos ejes en el cual iba montado un Láser infrarrojo de 0,5 W y un telescopio de 4 pulgadas de diámetro, como se puede ver en la Figura 2-8 (a). Cabe destacar que el terminal terrestre (LLGT: *Luna Lasercomm Ground Terminal*, frente al LLST: *Lunar Lasercomm Space Terminal*) consistía en un innovador OPA (*Optical Phased Array*), es decir un *array* de telescopios que, mediante las diferencias de fase en recepción, detectaban la orientación del haz proveniente del LLST. De hecho, el LLGT estaba constituido por dos *arrays*, uno compuesto por 4 telescopios de 6 pulgadas de diámetro para transmisión, y otro compuesto también por cuatro telescopios de 17 pulgadas de diámetro en recepción, como se puede ver en la Figura 2-8 (b).

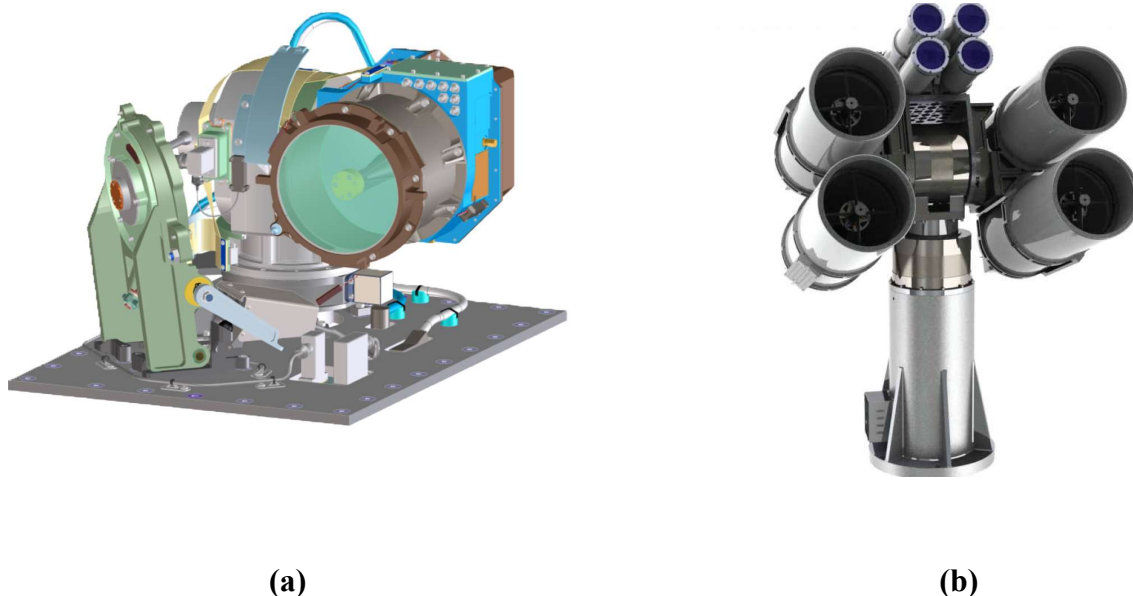


Figura 2-8 (a) Modelo del LLST (Lunar Lasercomm Space Terminal) [22], (b) Modelo del LLGT (Lunar Lasercomm Ground Terminal) [22]

Otro sistema digno de mención es el EDRS (*European Data Relay System*), cuya puesta en marcha se remonta al 2014 pero sigue su desarrollo e implementación a día de hoy [23]. Dicho sistema consiste en una serie de satélites geoestacionarios con la función de relé, de tal forma que permita la conexión de las estaciones en tierra que tienen permanentemente a la vista con otros satélites fuera de vista, como suelen estar multitud de satélites en órbitas LEO (*Low Earth Orbit*: órbitas a menos de 2 000 km de altitud) o MEO (*Medium Earth Orbit*: órbitas entre los 2 000 km y los 36 000 km, esta última siendo la órbita geoestacionaria) debido a su propósito (exploración de la Tierra, geoposicionamiento,...), así como con naves espaciales o sondas espaciales. Dicha interconexión, ilustrada en la Figura 2-9, se realizaría principalmente por telecomunicaciones ópticas, en concreto mediante los LCT (*Laser Control Terminal*) fabricados por la empresa Tesat, pionera en este tipo de telecomunicaciones, permitiendo una tasa de transmisión de más de 1 Gb/s (en concreto llegaría a tasas de 1,8 Gb/s) a distancias de 45 000 km [24], aunque también disponen de un enlace en la banda Ka.

El sistema inició su puesta a punto con la transmisión desde el satélite Sentinel 1-A, satélite cuyo objetivo es la observación de la Tierra y que orbita por lo tanto en LEO, hasta su estación en tierra de imágenes terrestres mediante el empleo del satélite de la ESA Alphasat, utilizándose entre estos un enlace FSOC. Dicho satélite normalmente tenía que almacenar la información recolectada hasta poder establecer contacto directo con el segmento terrestre, pero mediante este relé se puede retransmitir en tiempo real imágenes del globo desde casi cualquier punto de la órbita. Una vez comprobada la eficiencia del sistema, se puso en marcha el proyecto Artes 7 para la implementación de todo el sistema. Se prevé que el EDRS se encuentre totalmente operativo para el 2019, componiéndose de dos relés a bordo de los segmentos espaciales EDRS-A y EDRS-C, respectivamente situados en longitud 9°E y 31°E, con sus respectivas estaciones terrestres en Francia, Alemania, Bélgica y Reino Unido.



Figura 2-9 Representación artística del EDRS [23]

También cabe reseñar el experimento *Optical Payload for Lasercomm Science* (OPALS) que tuvo lugar a bordo de la ISS (*Internationa Space Station*) en 2014 [25], como se puede ver en la Figura 2-10. Dicho experimento fue llevado a cabo por el *Jet Propulsion Laboratory* de Caltech, bajo la dirección de la NASA, para estudiar el enlace directo desde Tierra hasta la estación, orbitando esta en una LEO. En concreto se buscaba el estudio de la calidad y estabilidad del enlace teniendo en cuenta tiempos de transmisión de 2 a 3 minutos en los cuales la ISS puede descender cientos de kilómetros, además de variar en demora a un ritmo de $0,5^\circ/\text{s}$ a $1,2^\circ/\text{s}$ y sobrevolar distintos tipos de atmósferas. De las observaciones se estudió la influencia de factores como condiciones atmosféricas, momento del día, estación del año y elevación sobre el horizonte, así como el procesamiento de datos distorsionados por las pérdidas atmosféricas o los procedimientos a seguir para establecer el enlace.

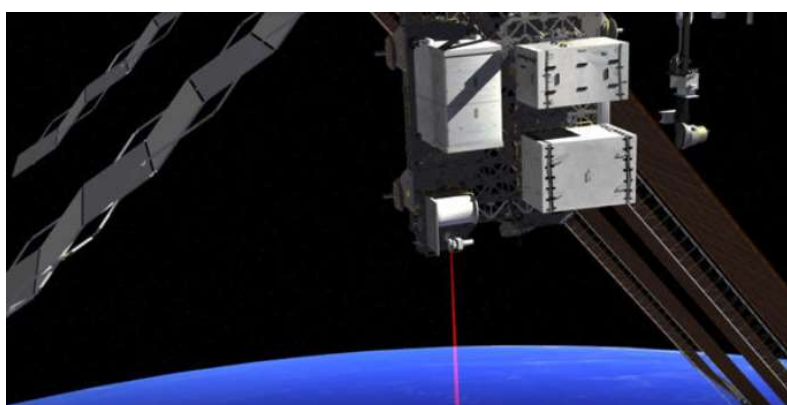


Figura 2-10 Representación artística del OPALS [25]

Por último, cabe reseñar el sistema HALO (*High Articulation Laser Optics*), en desarrollo desde 2018 por la empresa Laser Light, y que constituye la primera red de telecomunicaciones mundial basada exclusivamente en sistemas de telecomunicaciones ópticas [26]. Para el 2020, se pretende disponer la primera “*All-Optical Hybrid Global Communications Network*”, consistente en una red que combina las FSOC y la fibra óptica para multiplicar la capacidad actual de las infraestructuras de

telecomunicaciones. Dicho proyecto nace de la necesidad de satisfacer una demanda creciente de interconexión, que según la empresa se duplicará a dos años vista, debido a diversos factores como son el aumento de usuarios en la red, así como el aumento del consumo de datos por cada usuario o el aumento de proveedores de contenidos multimedia. La demanda requiere de una enorme capacidad de las infraestructuras de telecomunicaciones, y las radiocomunicaciones convencionales están saturadas, además de ofrecer prestaciones peores que las posibilidades que ofrecen los sistemas ópticos de telecomunicaciones.

Por lo tanto, para disponer una red global y de gran capacidad, *Laser Light* pretende combinar las infraestructuras existentes de fibra óptica, de gran capacidad y ampliamente instaladas a lo largo del globo, con un sistema espacial de satélites interconectados mediante FSOC, con gran capacidad y latencia reducida, ilustrado en la Figura 2-11.

Dicha red constaría de 12 satélites en órbita MEO con 48 enlaces intersatelitales disponibles, cada uno con una capacidad de 48 200 Gb/s, y de 72 estaciones en Tierra, obteniendo enlaces bidireccionales satélite-estación de hasta 200 Gb/s. Estos enlaces se establecerían por lo tanto mediante un sistema FSOC que trabaja con una longitud de onda de entre 1 525 μm y 1 565 μm , resultando en una capacidad efectiva total de la red de más de 33 Tb/s, y corresponderían a la espina dorsal del sistema, dejando la fibra óptica para la transmisión de datos en el segmento terrestre, desde la estación terrestre hasta los usuarios y viceversa, aprovechando las estructuras ya disponibles. Cabe por último reseñar que cuando se complete la implantación de este sistema, podremos hablar de *Optical Satellite-as-a-Service* (OSaaS), en analogía al empleo de fibra óptica actual dentro del marco de *Infrastructure-as-a-Service* (IaaS).

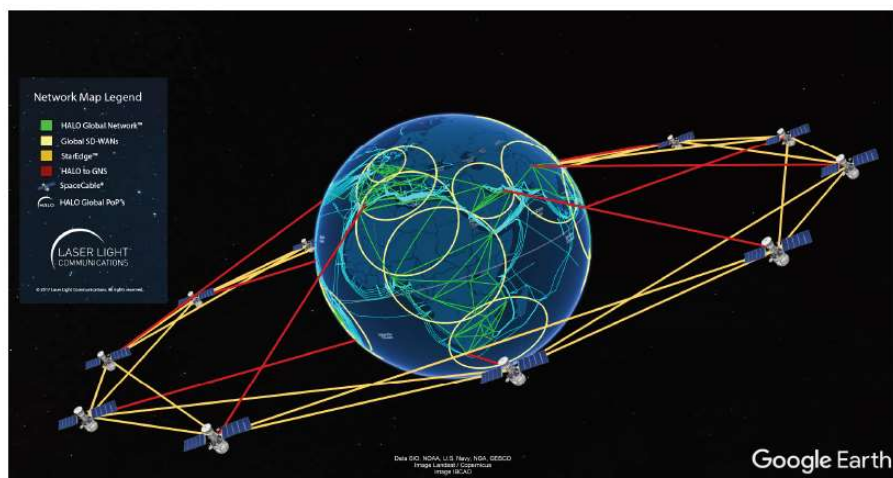


Figura 2-11 Representación funcional del sistema HALO [26]

2.2.2 Aplicaciones dentro de la atmósfera terrestre

Por dentro de la atmósfera terrestre los sistemas FSOC se ven en gran medida perjudicados, debido a las limitaciones ya mencionadas. Cuanto más nos acercamos a la superficie terrestre, más se acentúan todos los efectos adversos para dichas comunicaciones, afectando sobre todo al alcance y eficiencia (referida a la tasa de error) de estas. Los fenómenos meteorológicos tales como los movimientos de masas de aire, la nubosidad, las precipitaciones, el viento, etc. se dan en la troposfera, que es la capa de la atmósfera que comprende desde la superficie hasta entre los 8 y los 14,5 km de altitud. Por encima de esta capa, la estratosfera presenta un perfil mucho menos hostil para las comunicaciones FSOC [19]. Esta capa, que va hasta los 50 km de altitud y no responde a variaciones bruscas en el tiempo y en el espacio, sino que presenta una aumento constante de la temperatura desde el límite inferior de la capa en la que se encuentra el aire más frío y denso, con una temperatura en torno a los -55°C en la denominada tropopausa (zona que separa la troposfera de la estratosfera y en la

que el gradiente de temperatura pasa de ser negativo a positivo), hasta unas temperaturas que rondan los 0°C en la denominada estratopausa (zona que separa la estratosfera de la mesosfera y en la que le gradiente de temperatura pasa de positivo a negativo).

Así mismo, los diferentes componentes gaseosos de la estratosfera se mueven mayoritariamente en dirección horizontal, dando lugar a una capa muy estable con pocas alteraciones en su composición, temperatura y presión, lo cual se traduce en un índice de refracción que varía de manera constante en el espacio (principalmente en altitud), y de manera despreciable en el tiempo. Si a esto le sumamos que en dicha capa la humedad está muy poco presente, y que por lo tanto tampoco tendremos grandes pérdidas por absorción, podemos afirmar que el empleo de FSOC en dicha capa resulta más que posible, ya que podemos predecir con cierta exactitud la desviación del haz, y sobre todo podemos mantener un enlace relativamente estable una vez enlazadas las partes.

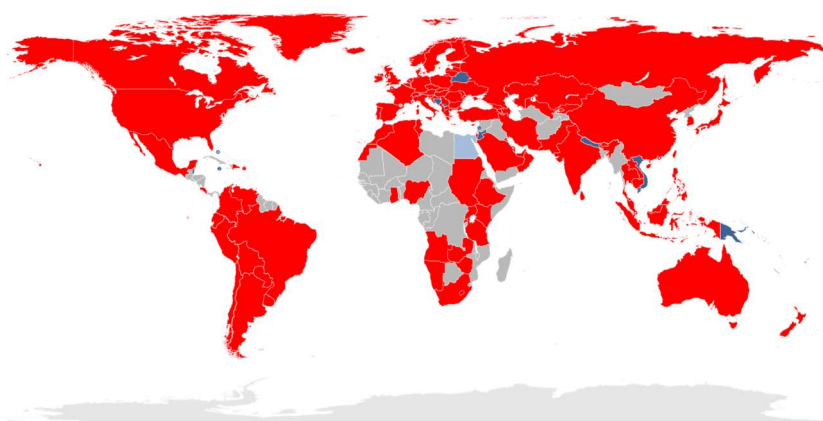


Figura 2-12 Países en los que se ha adoptado la tecnología LTE [27]

Un sistema innovador que recurre a las FSOC en la estratosfera es el proyecto Loon, desarrollado por la empresa Google, en concreto por su departamento de I+D denominado Google X [28]. Dicho sistema, cuyo primer prototipo vio la luz en 2011, fue concebido primordialmente como infraestructura para conectar a todas aquellas zonas que no disponían de ningún medio de telecomunicación y por lo tanto estaban aisladas de la red. En efecto, las diferentes generaciones de la TMA (Telefonía Móvil Automática) requieren una mayor infraestructura, a la vez que ofrecen mayores prestaciones. Desde la transmisión de datos reducida de la segunda generación, que se limitaba a los SMS, el Fax y la conexión muy limitada a la red, con una tasa máxima de transmisión de 9,6 kb/s, hasta las prestaciones de la última generación (4G), englobadas en el estándar LTE (*Long Term Evolution*), que permiten velocidades de hasta 100 Mb/s a baja movilidad, se ha requerido la implantación de infraestructuras cada vez más costosas [9] [29].

Es por lo cual dichas tecnologías se implementan en primera instancia en ciudades muy pobladas, donde la densidad de población permite a los ISP (*Internet Service Provider*: generalmente las compañías telefónicas) sacar beneficio de las inversiones en infraestructuras que requiere el servicio, dejando al margen las zonas más remotas y menos habitadas, ya que requieren de unas infraestructuras parecidas, pero ganando significativamente menos clientes. Además, se da el caso de que las infraestructuras necesarias puedan resultar inasumibles para el país entero, quedando este totalmente marginado en cuanto a conectividad se refiere, como se refleja en la Figura 2-12. A esto hay que sumar todos aquellos acontecimientos que accidentalmente pueden dejar una o varias zonas geográficas fuera de cobertura, como desastres naturales, guerras, etc., dejando en total entorno a un 60% de la población mundial fuera de cobertura.

Un ejemplo es que, a día de hoy, en la península ibérica, aun siendo España un país totalmente desarrollado no se tiene cobertura ni siquiera de 3G en la totalidad del territorio peninsular nacional, como podemos ver en las Figura 2-13 (a) y (b). Esta segregación de la conectividad resulta en una necesidad en la población de dichas zonas de acceder a la red, necesidad que puede considerarse como prioritaria por infinidad de razones de índole económica, educativa, social,... Es por ello que la empresa norteamericana Google, con cierta motivación altruista pero respondiendo a una demanda del mercado, ha puesto en marcha un sistema de telecomunicaciones basado en globos estratosféricos que es capaz de proveer a las zonas más remotas de cobertura LTE con una infraestructura más que asequible.



(a)



(b)

Figura 2-13 (a) Zonas con cobertura 4G (en violeta) [30], (b) Zonas con cobertura 3G (en verde) [30]

En concreto, dicho sistema estaría formado por un globo estratosférico capaz de aguantar hasta 190 días de forma completamente autónoma, y la carga útil como tal, como se puede ver en la Figura 2-17 (b). Esta está compuesta por dos antenas LTE que ofrecen la cobertura de telefonía móvil mencionada, unos paneles solares eficientes, que llegan a producir hasta 1,72 GJ de energía eléctrica, y que permiten el funcionamiento del sistema durante el día y la recarga de las baterías para posibilitar el funcionamiento de noche, y un módulo de control (llamado capsula de vuelo) [28], como se puede ver en la Figura 2-14.

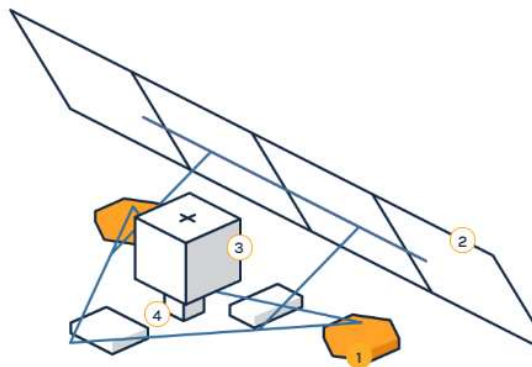


Figura 2-14 Esquema de la carga útil de un globo de Loon: (1) antenas LTE, (2) panel solar, (3) sistema de gobierno y FSO, (4) paracaídas [28]

En el módulo de control coexisten dos subsistemas, uno de gobierno autónomo y otro de interconexión con otros globos, mediante un enlace FSO. La arquitectura de este sistema reside en el hecho de que cada globo puede comportarse como un relé de gran capacidad, mediante el empleo de FSOC, y de gran distancia, debido al medio empleado. Como se puede ver en la Figura 2-15, un repetidor en tierra enlazado con un globo mediante FSOC transmite y recibe datos a un ISP a alta velocidad. Este a su vez puede transmitir o recibir datos de otros globos mediante FSOC y puede dar cobertura LTE a los usuarios situados en su zona de cobertura. Además, debido a la altura, la superficie cubierta por cada globo es francamente más grande cualquier zona de cobertura de cualquier infraestructura terrestre. Siendo así, desde el terminal de un usuario los datos se recibirían en el globo situado en su zenit, viajando a continuación de globo en globo hasta una estación conectada al ISP.



Figura 2-15 Representación artística de modo de empleo de globos Loon [28]

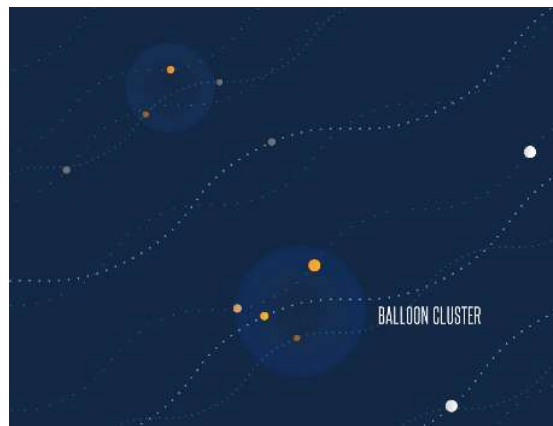
Cabe reseñar que los globos son capaces de guiarse en el espacio sin necesidad de motor ya que aprovechan de una forma ingeniosa las corrientes de aire de la estratosfera, principalmente de componente horizontal. Estas corrientes varían de dirección en función de la altura, pero suelen ser de dirección relativamente paralela al suelo, lo cual permite redirigir al globo solamente con hacerlo ascender o descender, como se puede observar en las Figura 2-16 (a) y (b). Los globos Loon llevan un sistema de gobierno autónomo que, mediante algoritmos para toma de decisiones y modelos predictivos de las corrientes, son capaces de guiar el globo con relativo acierto. Este aspecto permite no solo situar al globo en un lugar deseado, pero también permite mantener una red de globos en suficiente proximidad los unos de los otros como para asegurar la comunicación entre estos y por lo tanto la comunicación con el ISP, como ilustra la Figura 2-16 (c) [28].



(a)



(b)

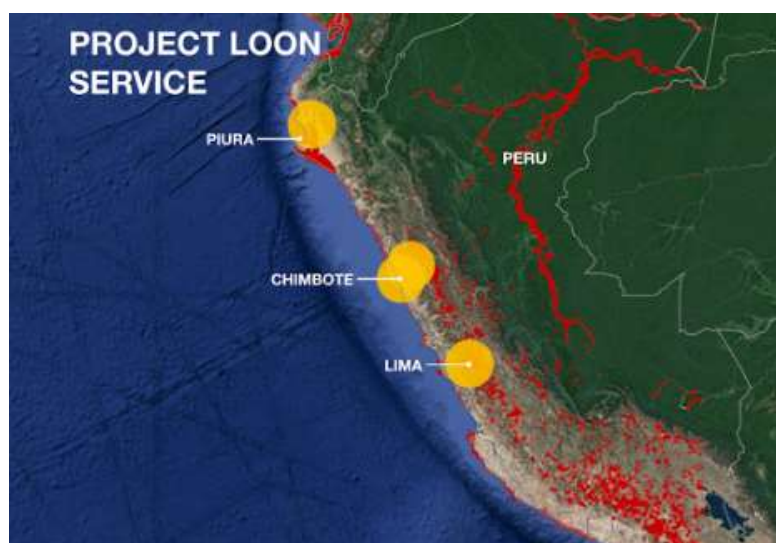


(c)

Figura 2-16 (a) Representación artística del desplazamiento del globo Loon (1) [28], (b) Representación artística del desplazamiento del globo Loon (2) [28], (c) Representación artística del agrupamiento de globos Loon [28]

Por último, el sistema Loon, cuya apariencia se puede ver en la Figura 2-17 (c), ha sido empleado con éxito en varias ocasiones, como en India con la colaboración con AP Fiber para dar mayor conectividad al estado indio de Andhra Pradesh, pero la reseña más remarcable es el rol que jugó durante las inundaciones que sufrió Perú en marzo de 2017. En dicha fecha acaecieron diluvios torrenciales con subsecuentes inundaciones masivas. Más de 800 localidades se declararon en estado de emergencia y las infraestructuras de telecomunicaciones se vieron severamente castigadas por los meteoros. Fue entonces cuando el equipo de Loon, en conjunto con Telefónica y el gobierno peruano, pudieron ofrecer una solución al problema en cuanto a conectividad, dando servicio en una superficie de más de 40 000 km² durante más de 7 semanas y con un tráfico total de 160 Gb, como se puede ver en la Figura 2-17 (a) [28].

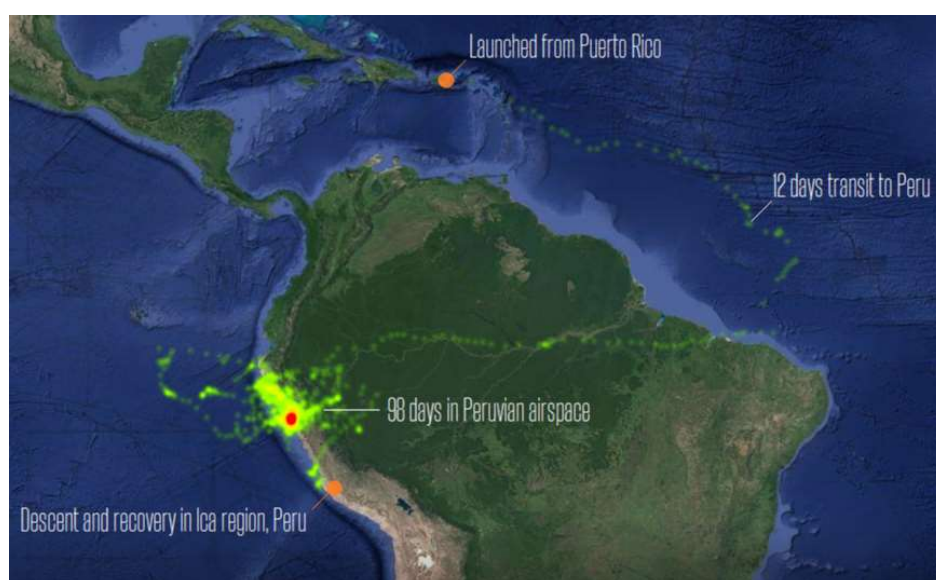
Irónicamente, los modelos predictivos de viento estratosférico y los algoritmos de aeronavegación fueron probados con éxito en la misma zona un año antes, despegando los globos desde Puerto Rico, realizando una travesía de 12 días hasta Perú y manteniéndose en el espacio aéreo peruano hasta 14 semanas, como se puede ver en la Figura 2-17 (c). Huelga decir que la experiencia ganada de dichas pruebas contribuyó en gran medida al éxito de su posterior empleo.



(a)



(c)



(a)

Figura 2-17 (a) Zonas de cobertura por globos Loon (naranja) y zonas afectadas por inundaciones (rojo) en Perú [28], (b) Fotografía de un globo Loon [28], (c) Representación geográfica de ensayo de predicción de modelos meteorológicos y algoritmos de navegación autónomos [28]

Otro proyecto, competidor del proyecto Loon, es el proyecto Aquila. Este proyecto data del 2014, cuando la compañía Facebook, a través del Facebook Connectivity Lab, se lanzó al mundo de las telecomunicaciones FSO con el mismo objetivo que perseguía Google con Loon, que no era más que favorecer la conectividad de las personas. Yael Maguire, director del departamento de Ingeniería de Facebook, añade en [31] a los motivos ya compartidos con Google el hecho de que en 2015, más del 90% de la población mundial tenía acceso a sistemas de telecomunicaciones TMA de segunda generación (2G), pero que era altamente improbable que se llegase al 100% debido principalmente al hecho de que un quinto de la población mundial vivía en la extrema pobreza, es decir ganando 1,25 \$/día o menos, y las compañías que invirtiesen en las infraestructuras necesarias para dar una cobertura

totalmente global seguramente no podrían vender el servicio a un precio accesible para dicho sector de la población recuperando a la vez la inversión.

La principal diferencia con los balones de Loon es que Aquila consiste también en un UAV (*Unmanned Aerial Vehicle*) pero de naturaleza monoplana. El proyecto de Facebook consistía en un avión de grandes dimensiones, alcanzando una envergadura de 43 m (superior a la de un Boeing 737), que volaría en la estratosfera, en concreto entre 18 y 27 km de altitud. Uno de los objetivos del proyecto era que tuviese una gran autonomía, en concreto hasta 3 meses. Por lo tanto, se tuvo que optimizar al máximo el diseño de la plataforma en todos los aspectos.

En primer lugar, como menciona el Ingeniero Jefe del Equipo de Aviación de Facebook Andy Cox en [32], el avión necesitaba ser a la vez rígido pero muy poco pesado. Para ello, se empleó como material principal para la estructura fibra de carbono T-700 curada, tres veces más duro que el acero, pero francamente más ligero. Además, se emplearon cuatro motores eléctricos muy eficientes, que consumían una potencia total de 5 000 W (lo consumido por tres secadores), y un sistema de alimentación constituido por baterías de gran energía y paneles solares distribuidos a lo largo de las alas. El conjunto, contando con la carga útil (es decir el sistema de telecomunicaciones a emplear) pesaría alrededor de 400 kg, o en otras palabras un tercio del peso de un coche eléctrico convencional (sabiendo que la mitad del peso del avión se corresponde con el de las baterías), y alcanzaría velocidades de hasta 80 mph en altura, donde el aire tiene una densidad igual al 5% de la densidad del aire al nivel del mar, como se puede ver en la Figura 2-18.

Además, para minimizar el arrastre de las alas, se emplearon alas de curvatura reducida, es decir con un ángulo de ataque reducido, y de alargamiento elevado. La reducción del ángulo de ataque permite reducir el arrastre que producen las alas, que no es más que la fuerza que ejerce el aire en oposición al movimiento del ala en el seno del fluido, lo cual hace a la plataforma más eficiente en cuanto a energía empleada en el movimiento se refiere. Sin embargo, la reducción del ángulo de ataque va en detrimento de la sustentación del avión, el cual necesita por lo tanto de un fuselaje de alargamiento mayor, es decir de un ratio envergadura-cuerda media mayor, siendo esta última la media de las distancias entre el borde de ataque y el borde de salida de la totalidad del ala. Es por esto último que la plataforma presenta un aspecto tan alargado.

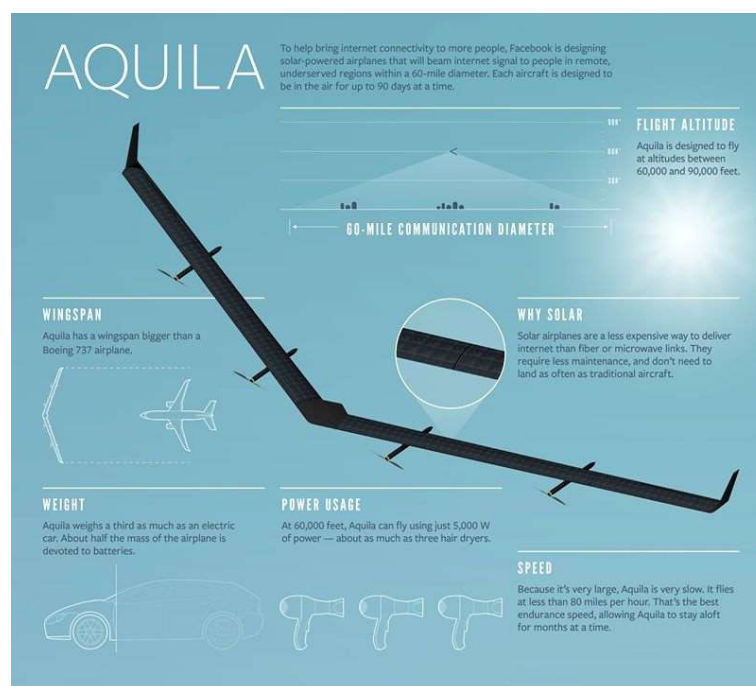


Figura 2-18 Póster publicitario del proyecto Aquila [33]

En cuanto a la carga útil del proyecto, el proyecto Aquila es muy parecido al de Loon, ya que se pretendía disponer de varios aviones en las zonas que requiriesen de conectividad, obteniendo por cada avión una zona de cobertura de telefonía móvil de aproximadamente 60 millas, como se puede ver en la Figura 2-19 (c). Dicho tráfico sería por lo tanto redirigido hacia a otros UAVs mediante FSOC, como se representa en la Figura 2-19 (a), con un alcance máximo de entre 200 km y 300 km y con una capacidad de más de 50 Gb/s, hasta el receptor terrestre del ISP, que hubiese enlazado de igual modo mediante FSOC, alcanzado tasas de transmisión y recepción de entre 30 y 40 Gb/s [31]. Cabe reseñar que el proyecto Aquila no fue diseñado para atender a los terminales de usuario directamente, en contraposición a su análogo Loon, sino que requería de antenas de pequeño porte en recepción, siendo pues ideado para dar cobertura a torres de telecomunicación pequeñas o directamente a parabólicas, como se representa en la Figura 2-19 (b).

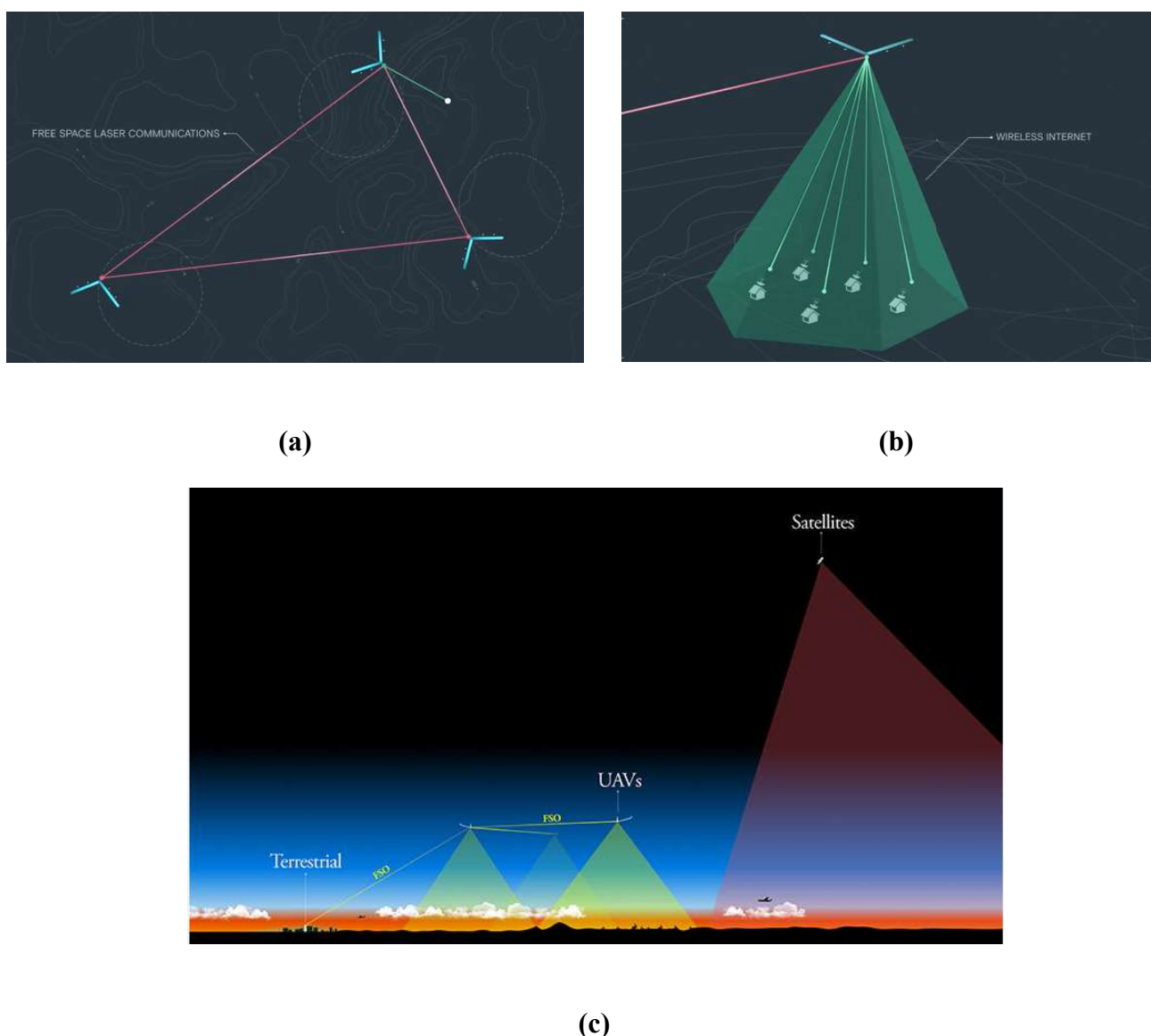


Figura 2-19 (a) Representación artística del enlace FSOC de Aquila [31] , (b) Representación artística de zona de cobertura de Aquila [31], (c) Comparativa de la zona de cobertura de Aquila frente a satélite [31]

El sistema fue desarrollado hasta el punto de que se llegó a construir un prototipo completo en 2016 en las instalaciones de la empresa en Bridgwater, en el Reino Unido, que fue ensamblado en Estados Unidos, en concreto en Arizona. Dicho prototipo se probó exitosamente en un vuelo que se

preveía de una hora pero que acabó durando 96 minutos debido al buen funcionamiento el 28 de junio de 2016 [31] [33]. Sin embargo, todo fue bien hasta el aterrizaje, en el cual el avión se quedó corto en la entrada en pista y sufrió grandes daños estructurales, que sucesivas investigaciones atribuyeron a un fallo estructural. Otro prototipo fue realizado en 2017, de mismas características que su predecesor, pero con alguna mejora como la incorporación de *spoilers* a las alas, que no son más que láminas situadas en la cara superior del ala que perturban el flujo de aire en dicha cara, reduciendo la sustentación del ala y permitiendo un mejor control del descenso de la aeronave. El 22 de mayo de 2017 volvió a volar el Aquila durante 1 hora y 46 minutos, considerándose esta vez todo un éxito de principio a fin.

Sin embargo, sería la última vez que volaría ya que, en 2018, Facebook decidió cerrar el proyecto debido principalmente a la falta de competitividad en relación a la plataforma respecto a sus rivales, y en particular Airbus [34]. En el contexto de los pseudo-satélites, denominados HAPS (*High Altitude Platform Station* o también *High Altitude Pseudo-Satellite*), que no son más que plataformas situadas entre 20 y 50 km de altitud y que realizan diferentes misiones (sin FSOC implicadas) como vigilancia e inteligencia, radiotelecomunicaciones o monitorización del tiempo, Airbus decidió desarrollar su avión Zephyr desde el 2003, representado en la Figura 2-20. Este avión también fue diseñado para los mismos propósitos que el Aquila, siendo primordiales la autonomía y la altitud, pero de la mano de una empresa pionera en el sector y con casi 40 años de experiencia como es Airbus [35]. De hecho, ya en el 2010, el Zephyr 7 rompió el recorrer de mayor tiempo de vuelo de una aeronave no tripulada sin repostaje, durando 336 horas, 22 minutos y 8 segundos en vuelo. Además, la empresa subsidiaria de Airbus, Tesat, había desarrollado ya los terminales LCT, a bordo de los satélites del EDRS mencionado en el apartado 2.2.1, que ofrecían prestaciones mucho más interesantes que los objetivos que se propusieron con Aquila. Siendo así, la empresa americana abandonó el proyecto Aquila, uniéndose de hecho al desarrollo de la conectividad en sistemas HAPS con Airbus.



Figura 2-20 Representación artística de UAV Zephyr [35]

2.2.3 Aplicaciones en la superficie terrestre

En altitudes más bajas, al nivel de la superficie terrestre, las limitaciones atmosféricas para las FSOC se vuelven francamente más perjudiciales para cualquier tipo de aplicación. A este nivel tendremos que contar con todos los fenómenos meteorológicos que acontecen en la troposfera, como son la lluvia, la niebla, la nieve, las partículas como polvo o aerosoles, así como el viento. Como se ha mencionado en el apartado 2.1.2, a estos niveles tendremos que hacer frente a pérdidas por absorción, dispersión o centelleo entre otras. Es por lo cual las aplicaciones existentes a día de hoy de las FSOC

presentan prestaciones mucho más reducidas, sobre todo respecto al alcance de estas, y por lo tanto no son de tanta entidad como las que se implementan a mayores alturas, a excepción a lo mejor de la tecnología Li-Fi que comentaremos más adelante.

En concreto, las FSOC se dividen en dos familias, las de largo alcance, que presentan las mismas características que las aplicaciones previamente vistas y que basan su funcionamiento en el empleo de un Láser, y las VLC (*Visible Light Communication*), que emplean en vez de un rayo Láser un haz de luz proveniente de LEDs. Estos sistemas tienen un alcance operativo inferior a los 4,5 km en cualquier caso, como se muestra en [36], pero consiguen altas velocidades de transmisión, por encima del Gb/s, y pueden ir acoplados a sistemas de telecomunicaciones convencionales si la distancia a salvar entre la estación transmisora y la receptora en un enlace en concreto es demasiado grande. El record actual en cuanto a ancho de banda lo ostenta la Universidad de Oxford, que en condiciones de laboratorio consiguió una tasa de 224 Gb/s, utilizando fibra óptica tanto en el emisor como en el receptor y una modulación DWDM, típica de la fibra, pero con una propagación intermedia sin guía [12].

En primer lugar, las FSOC que se emplean a nivel de la superficie terrestre de mayor alcance siguen siendo las que emplean un haz de luz Láser. Se emplean en diferentes ámbitos, como por ejemplo en las “*blackhaul communications*”, es decir las comunicaciones entre las infraestructuras troncales de telecomunicaciones y las infraestructuras de servicio [29]. En este ámbito, se debe primero entender que a medida que las nuevas generaciones de TMA (Telefonía Móvil Automática) evolucionan y ofrecen mejores prestaciones y capacidades, las infraestructuras también van quedando obsoletas y necesitan de nuevos medios para abastecer a la demanda de los usuarios.

Hoy en día asistimos al denominado “blackhaul bottleneck”, que no es más que la saturación de las comunicaciones no entre las torres y los usuarios, sino entre las torres y las infraestructuras troncales. Estos enlaces requieren de una capacidad enorme y en crecimiento exponencial, debido al aumento de usuarios y de consumo de cada usuario. En la Figura 2-21, podemos ver la arquitectura de los sistemas TMA, y en la Tabla 2 podemos apreciar que existen tres tecnologías para poder solventar la mencionada congestión, cada una con ventajas e inconvenientes.

Las microondas (MMW: *MiliMeter Wave*, ondas electromagnéticas de la banda EHF (30GHz-300GHz)), ofrecen un alcance considerable, llegando hasta los 20 km de distancia [29], pero teniendo en cuenta las posibilidades que ofrece la última generación de telefonía automática (4G), solo podría dar servicio completo a un solo cliente.

Es por lo que dos posibilidades parecen ser las únicas viables, la fibra óptica y las FSOC. Las primeras son las idóneas, ya que ofrecen la mayor velocidad, pudiendo dar servicio completo a más de 200 clientes de 4G. Sin embargo, el tendido de los cables es costoso y en multitud de zonas no es viable, aparte de aparatoso en cuanto a la obtención de licencias para cavar, así como de la gestión de la propia obra, cosa que en el dominio de las FSOC no ocurre. Sin embargo, las FSOC ofrecen una capacidad bastante interesante (hasta 100 clientes en servicio completo) a un coste reducido, pero su operatividad depende en gran medida de la distancia y los meteoros, pudiendo operar a una distancia máxima de 4,5 km pero reduciendo su distancia operativa por debajo de los 500 m en caso de lluvia densa, niebla, tormenta de arena, etc.

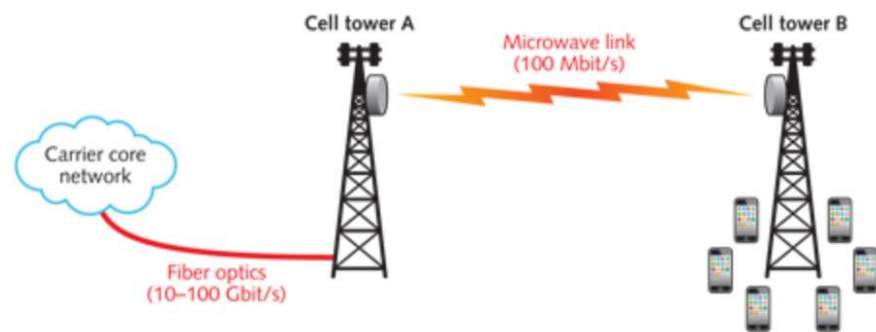


Figura 2-21 Esquema de enlace estación a estación (blackhaul communication) [29]

Backhaul technology	Backhaul speed	Number of mobile users for each backhaul technology (by network; individual download speed)			
		(2G; 10 kbit/s)	(2.5G; 200 kbit/s)	(3G; 5 Mbit/s)	(4G; 100 Mbit/s)
Copper	10 Mbit/s	1,000	50	2	0.1
Microwave	100 Mbit/s	10,000	500	20	1
Fiber optics	20 Gbit/s	2,000,000	100,000	4,000	200
Free-space optical	10 Gbit/s	1,000,000	50,000	2,000	100

Tabla 2 Comparativa de servicios en función de tecnología blackhaul [29]

Por otro lado, diferentes empresas se han volcado en el negocio del desarrollo de las FSOC en multitud de ámbitos. Entre ellos se encuentran la conectividad entre edificios, útil en campus y centro de negocios en los que una misma institución posee un complejo de varios edificios y necesita de conectividad en todos ellos, además de una red en común en la mayoría de los casos [10] [17]. Para dichas instituciones es mucho más interesante contratar una sola línea y distribuirla internamente por aspectos económicos, administrativos y de seguridad y control, que contratar una para cada edificio. Las FSOC permiten por lo tanto crear una CAN (*Campus Area Network*: red en una zona geográfica limitada, inferior a una ciudad, pero mayor que una casa, es decir entre MAN y LAN) robusta, segura y de gran capacidad, como se representa en la Figura 2-22.

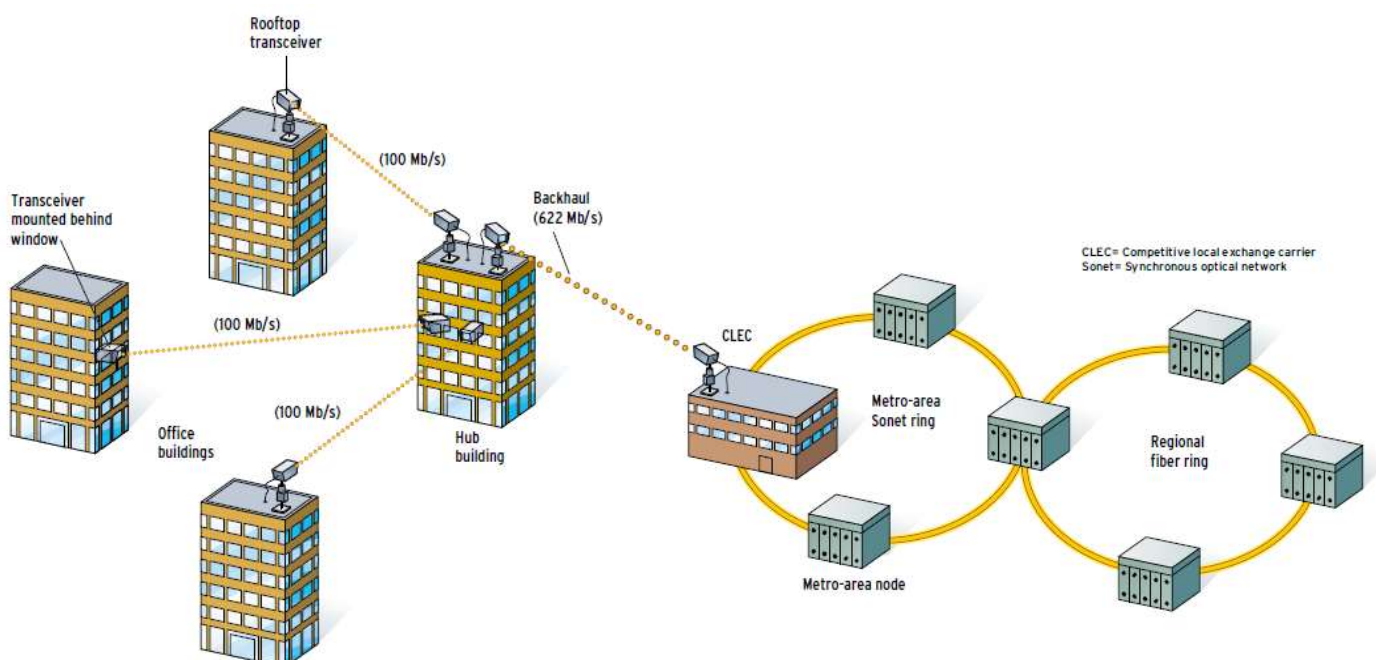


Figura 2-22 Esquema de instalación de FSOC en CAN [17]

Otro ámbito también es el de las comunicaciones que requieren una latencia baja, como pueden ser los servicios de emergencia, policía o el ámbito militar, así como comunicaciones críticas de gran capacidad, principalmente basadas en fibra óptica, pero con redundancia delegada en FSOC [37]. Cabe destacar que estos sistemas se emplean muchas veces acoplados en estos ámbitos con enlaces de ondas MMW en algunos enlaces, tanto para salvar una distancia demasiado grande para las FSOC como para asegurar la comunicación si la situación meteorológica no permite el empleo de estas. Un estudio reciente de la empresa Space Photonics en conjunto con la Harris Corporation, la University of Central Florida y la NASA, muestra que, a 4,5 km de distancia, unas condiciones ambientales no adversas y transmitiendo a 0.52 Gb/s, los paquetes sufren unas pérdidas de más del 35 % en las FSOC, pero mediante técnicas de buffering dinámico, consiguen reducir esta cifra hasta en torno a 4%, estas últimas debidas mayormente al prototipo en sí y no al enlace [36], como podemos ver en la Figura 2-23.

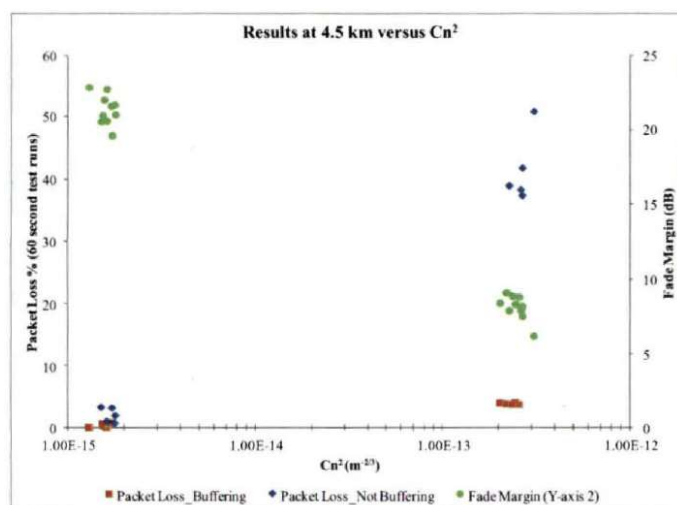


Figura 2-23 Comparativa de pérdida de paquetes frente a absorción atmosférica [36]

Es la mayor distancia a la que se ha conseguido emplear de manera fiable las FSOC, aunque por requerimientos del buffer la tasa de datos ha de ser de 0.52 Gb/s y no los 10 Gb/s que permiten como máximo estos sistemas (a menor distancia). También cabe reseñar la contribución de la empresa Attochron, que trabajando en conjunto con la Army de los Estados Unidos ha aplicado la tecnología de los USPL (*Ultra Short Pulse Laser*) para disminuir en gran medida la atenuación de la señal luminosa. Dichos estudios se comentarán más adelante en el apartado 2.2.4. Algunas de las principales empresas que compiten en estos sistemas son Lightpointe Communications Inc., Artolink, CableFree: Wireless Excellence, Space Photonics Inc. y EC Systems, cuyos modelos comparamos en la Tabla 3. Dichos modelos están en constante desarrollo, y algunas casas como Lightpointe Communications o Space Photonics preveen el desarrollo de sistemas de hasta 10 km de alcance, con mayor capacidad y fiabilidad. Para las casas con varios sistemas, se han expuesto los que mejores prestaciones ofrecen. En caso de tener varios modelos con diferentes rangos de acción, se han mencionado los de mayor tasa, así como los de mayor alcance.

Compañía	Modelo	Alcance operativo (en buenas condiciones)	Capacidad	Longitud de onda	Tipo	Latencia
Lightpointe Communications Inc. [37]	Aire X-Stream Series, Model X-FSO	2800 m	1,25 Gb/s	850 nm	Full-duplex	< 20 ns
Artolink [38]	Artolink M1-GE-L	4400 m	1 Gb/s	1550 nm	Full-duplex	< 100 ns
	Artolink M1-30GE	1500 m	30 Gb/s	1550 nm	Full-duplex	< 5 ns
CableFree: Wireless Excellence [39]	G 2000	2000 m	1,5 Gb/s	780 nm	Full-duplex	-
Space Photonics [40]	LaserFire	2500 m	10 Gb/s	1550 nm	Full-duplex	-
EC Systems [41] [42]	EL-10G	1500 m	10,3125 Gb/s	1550 nm	Full-duplex	20 – 50 ns
	EL-1GL	4400 m	1,25 Gb/s	1550 nm	Full-duplex	< 125 ns

Tabla 3 Comparativa de productos FSOC

Cabe reseñar de estos sistemas que las longitudes de onda son elegidas con ciertos criterios determinados. Concretamente, la atmósfera presenta ventanas concretas en las que la absorción es francamente menor para un rango concreto de longitudes de onda, mientras que para otras la absorción crece significativamente. El Láser con $\lambda = 1550$ nm es uno de los favoritos, ya que además de estar situado en una ventana relativamente amplia, como podemos en la Figura 2-24 [10] [15].

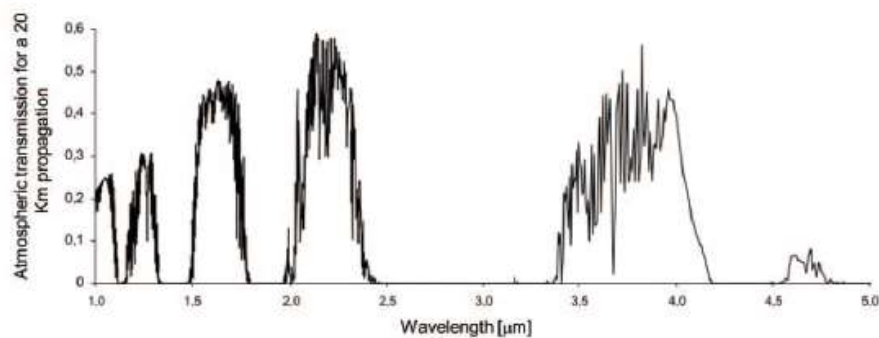


Figura 2-24 Gráfico de absorción de la atmósfera en función de la longitud de onda [14]

En segundo lugar, se puede hablar de una familia particular de FSOC: las VLC (*Visible Light Communications*). Dentro de las FSOC observamos en los últimos años el desarrollo de un nuevo tipo de tecnologías con un elevado potencial y gran crecimiento, que se distinguen de sus hermanas principalmente en la fuente de luz [43] [44] [45]. Las VLC, en vez de emplear una luz coherente y directiva como el Láser, emplea luz incoherente, es decir luz proveniente de una fuente convencional.

Para dichas comunicaciones se emplean normalmente LEDs (*Light Emitting Diode*), por su rapidez en cuanto a respuesta cuando se excita con una tensión. El espectro de dichas comunicaciones es el mismo que el de todas las FSOC, es decir con una frecuencia superior a 300 GHz, dentro del dominio del IR cercano, la luz visible y la luz UV (bandas no reguladas), y que todo lo mencionado anteriormente en cuanto a limitaciones, modulaciones y principio de funcionamiento se aplica igualmente a dichas comunicaciones.

Dentro de las VLC podemos reseñar como primer hito el proyecto RONJA (*Reasonable Optical Near Joint Access*), desarrollado por Karel Kulhavý de los laboratorios Twibright en 2001 [43]. Este sistema tiene como principal característica que es una UCT (*User Controlled Technology*, término acuñado el autor del proyecto), es decir, y en palabras de la empresa, que se tiene el derecho de uso, el derecho de acceso a la documentación fuente relativa al producto para su estudio, modificación y/o implementación de cualquier mejora, el derecho de manufacturar y vender el producto y el derecho de publicar versiones modificadas del producto. Así pues, toda la documentación relativa a la producción está disponible online sin restricciones, y venden diferentes componentes sobre todo relativos al terminal en sí. En cuanto al proyecto en sí, ofrece tres prototipos distintos:

- Ronja Tetropolis: 10 Mb/s de capacidad full-duplex a un máximo de 1,4 km empleando LEDs emitiendo una longitud de onda de 625 nm (Figura 2-25 (a)). Se emplean con conectores RJ45.
- Ronja Inferno: mismas características que Tetropolis, pero empleando luz IR y con un alcance máximo de 1,25 km
- Ronja 10M Metropolis: de mismas características que Tetropolis, pero para conexiones AUI y lentes convencionales (no de propósito de telecomunicaciones)



(a)



(b)



(c)

Figura 2-25 (a) Fotografía de los módulos transmisor y receptor de Ronja (perfil) [43], (b) Fotografía del módulo transmisor de Ronja en funcionamiento [43], (c) Fotografía del haz de luz en dispersión por la niebla [43]

Todos los prototipos tienen la misma arquitectura del sistema, reflejada en la. Cada estación dispone de un segmento transmisor en el cual un se modula una señal mediante un LED potente, como se puede apreciar en las Figura 2-25 (b) y (c), así como de un segmento receptor en el que la señal llega a un diodo PIN de 5mm, se amplifica mediante un TIA (*TransImpedance Amplifier*) y se procesa. Estos dos están conectados a un transceptor que sirve de interfaz entre el enlace de datos y la capa física de la comunicación. Cabe reseñar que estos productos suelen caseros, así que no tienen unas especificaciones técnicas muy precisas.

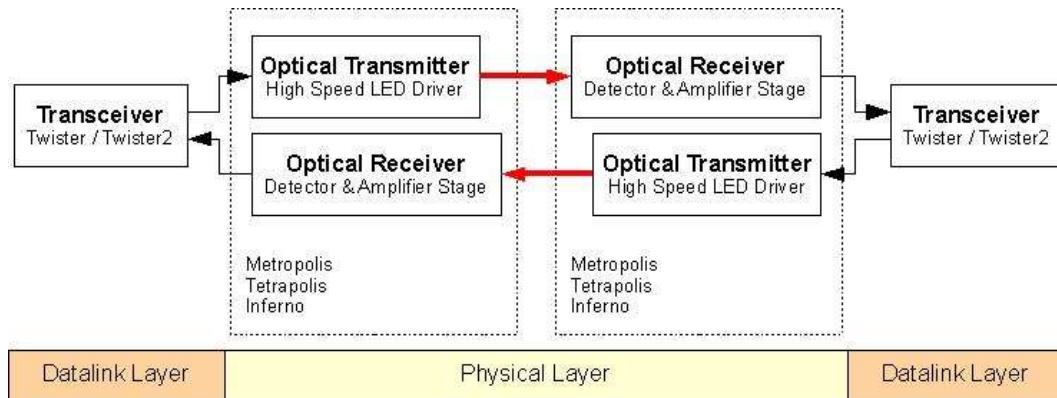


Figura 2-26 Diagrama de funcionamiento de Ronja [43]

Por otro lado, la tecnología VLC que está despuntando con gran diferencia es la que se denomina con el acrónimo LiFi (*Ligh Fidelity*) [44] [45]. Esta tecnología fue en primer lugar desarrollada por el escocés Harald Haas, profesor de Ingeniería en la Universidad de Edimburgo, a partir de 2011, aunque la idea, según el propio Haas, le vino cuatro años antes tras un coloquio acerca de la repartición del espectro de las radiofrecuencias en la World Radiocommunications Conference de Suiza. En concreto, se le ocurrió como aprovechar lo que en el espectro de dichas frecuencias y en el ámbito de la modulación paso banda (OFDM: *Orthogonal Frequency-Division Modulation*) es un defecto, que es el elevado PAPR (*Peak-to-Average Power Ratio*) que tienen estas modulaciones, en un contexto donde ese defecto fuese una ventaja [44].

La tecnología LiFi trata de conseguir una transmisión de datos a través de la luz corriente, debido al gran número de sistemas de iluminación que cuentan con LEDs y por lo tanto a su fácil instalación [45], como podemos ver en la Figura 2-27. Este sistema se propone para interiores, y por lo tanto para comunicaciones de corta distancia. Siendo así, todas las limitaciones en cuanto a FSOC se pueden despreciar, obteniendo un modelo de propagación extremadamente determinista que se puede incluso aprovechar para optimizar el sistema, como veremos más adelante. Como particularidad de este sistema frente a los demás vistos anteriormente, es que este permite no solo a comunicación punto a punto, sino también la comunicación punto a multipunto y multipunto a punto, lo cual la hace un sistema de red inalámbrica íntegro [46].

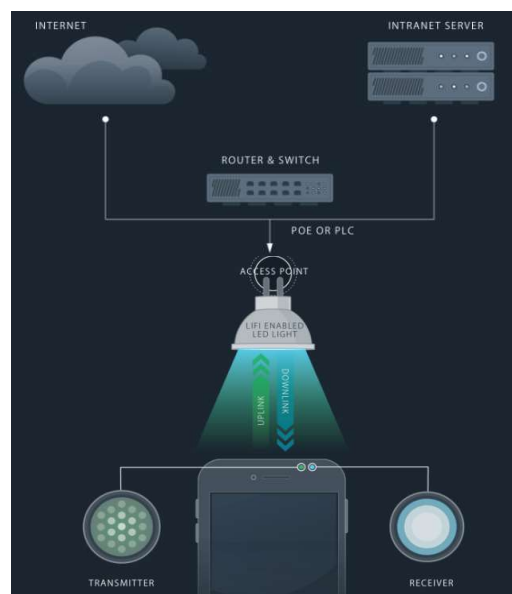


Figura 2-27 Esquema de un sistema LiFi [46]

En concreto, se puede emplear la luz sin que notemos ningún cambio respecto de la luz normal debido a que el ojo humano es incapaz de detectar centelleos con frecuencias por encima de los 100 Hz. Es por ello que cualquier tipo de modulación suficientemente rápida no tendrá ningún efecto en cuanto al empleo de un sistema LiFi para iluminación. Además, presenta la ventaja de que el rango de los puntos de acceso queda confinado entre los límites físicos de la estructura en la que se ha dispuesto el emisor, evitando interferencias (aunque no todas, veremos a continuación tipos de multiplexación para estos sistemas).

En cuanto a las modulaciones que se emplean, las modulaciones clásicas de OOK, PAM y PPM presentan un problema respecto a las interferencias entre símbolos y las distorsiones en el fotodiodo, sobre todo a tasas de transmisión elevadas y debido al aspecto multimodal del sistema. Por lo tanto, una primera aproximación es la propia técnica de modulación paso banda (OFDM). Esta consiste en aprovechar varios canales definidos por una portadora y, por lo tanto, pasar información mediante la repartición de la transmisión de esta en los distintos canales. Sin embargo, como estos sistemas también tienen que cumplir con la función de iluminación, la IEEE propuso una nueva modulación: la CSK (*Color-Shift Keying*). El principio de dicha modulación es el de codificar las intensidades de los colores primarios de la luz (rojo, azul y verde) mediante el formato RGB con el objetivo de mantener un color de la luz constante. La superposición de estos colores determina el color de la luz resultante, pero su intensidad puede ser modulada para pasar información [13] [45].

En cuanto al aspecto de red inalámbrica, al ser el único sistema FSOC que no está proyectado únicamente para una comunicación punto a punto, necesita a mayores de un diseño con una componente de multiplexación. La primera que se concibió fue la clásica TDMA (*Time Division Multiple Acces*), en el cual cada terminal dispone de una ventana de tiempo para transmitir información. Pero, sucesivas investigaciones han desarrollado nuevos métodos en los cuales se aprovecha el modelo de propagación de la luz. Como ya se mencionó, el modelo de propagación en estos sistemas es previsible con un grado de precisión bastante elevado. Esto se debe a que no está sujeto a ninguna limitación atmosférica, ni a ningún desalineamiento entre emisor y receptor, y por lo tanto lo único que afecta a la propagación es la pérdida por la distancia entre emisor y receptor. Siendo así, una segunda aproximación a la multiplexación fue la SDMA (*Space Division Multiple Access*). En este caso, se pretende dividir espacialmente la cobertura del punto de acceso, asignando a cada LED o grupo de LEDs un terminal concreto. En efecto, un sector que puede variar en el tiempo se corresponde con un único usuario, utilizando para ello algoritmos de búsqueda exhaustiva de terminales [45].

Una tercera aproximación es la familia de las NOMA (*Non-Orthogonal Multiple Access*), siendo predominante la relativa a la potencia de la señal. Esta se basa en la precisión de la estimación de la intensidad de la señal, y la variación que esta sufre con la distancia, ya que el SINR (*Signal-to-Interference/Noise Ratio*) varía en gran medida a lo largo de la propagación de la onda. La NOMA aprovecha este hecho y la posibilidad de predecir la intensidad de la señal en recepción para transmitir los datos de todos los terminales simultáneamente, pero a potencias distintas, de forma que la potencia radiada de una señal determinada es la necesaria para que esté por encima de la SINR mínima, pero por debajo de las señales con más potencia. Siendo así, a los terminales más cercanos al punto de acceso les corresponderá una potencia más baja, mientras que a los más alejados les corresponderá una potencia más elevada.

Para aislar la señal en recepción, tenemos en cuenta que el equipo no detecta señales inferiores a la SINR mínima y por lo tanto las señales de terminales más cercanos no se detectan. Sin embargo, para eliminar las señales de mayor potencia se emplea un esquema SIC (*Successive Interference Cancellation*), en el cual el terminal sucesivamente repite un ciclo que consiste en aislar la señal de mayor potencia y restarla a la señal recibida, como podemos ver en la Figura 2-28. Siendo así, el terminal acaba aislando la señal de menor potencia, que es la que le corresponde, y la procesa. Este

sistema presenta a día las mejores prestaciones, ya que presenta una capacidad mayor y gran flexibilidad [45].

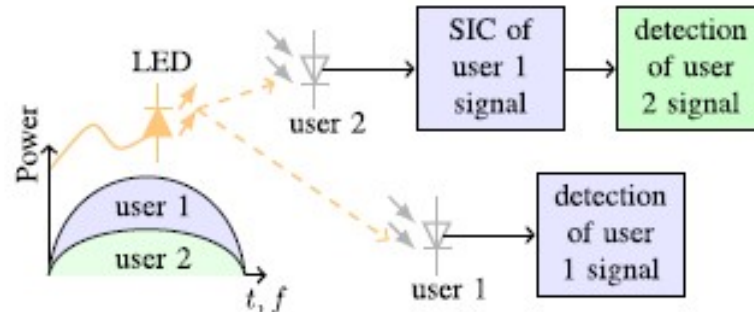


Figura 2-28 Diagrama de funcionamiento de multiplexación NOMA [45]

Sin embargo, una mejora de las NOMA se ha implementado respecto a la posibilidad de implementar estos sistemas en conjunto, y por lo tanto de tener más de un punto de acceso en la misma habitación. Si tenemos a algún usuario en una zona donde se solapan la cobertura de dos o más puntos de acceso, este sistema puede no funcionar ya que no podemos asegurar que la señal con menor potencia recibida por el terminal es la que le corresponde, ya que podría ser otra de otro punto de acceso. Por lo tanto, la mejora que ha surgido de este hecho es la técnica CoMP-AJT (*Combined Multi-Point-Aided Joint Transmission*). Para emplear esta técnica, se requiere inicialmente de algoritmos complejos para la detección de los nodos en las zonas de solape. Una vez detectados, todos los puntos de acceso transmiten las señales de todos los nodos en dichas zonas a una misma potencia, suficiente para que desde un punto de acceso le llegue al nodo más alejado. Al estar a la misma potencia, el nodo aísla varias señales a la vez, correspondientes a los nodos en zona de solape. Sin embargo, en el punto de acceso, se recurre a un TPC (*Transmit Pre-Coding*) que asigna cada punto de acceso a un terminal. Mediante este TPC, un terminal descodifica la señal que le corresponde de las señales de los demás nodos en zona de solape y se consigue por tanto la comunicación [45], como se puede ver en la Figura 2-29.

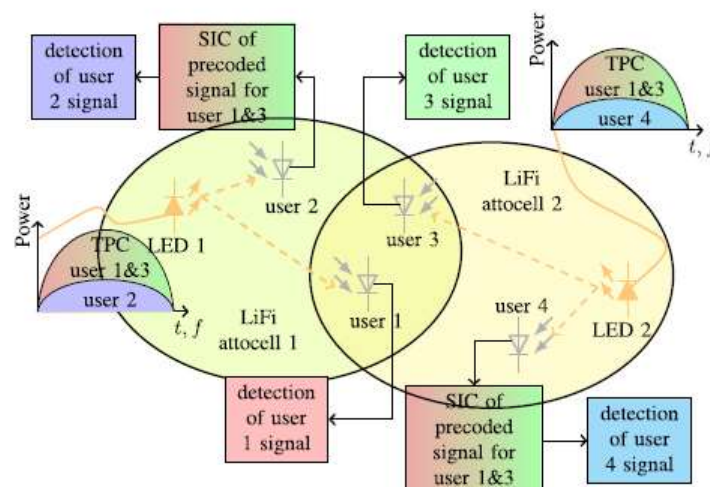


Figura 2-29 Diagrama de funcionamiento de multiplexación CoMP-AJT [45]

En cuanto al estado de la implantación de esta tecnología hoy en día, cabe reseñar que la Universidad de Edimburgo sigue investigando en esta tecnología. Se puede citar el reciente estudio de abril de 2017, en el cual un equipo de investigadores ha conseguido alcanzar velocidades de transmisión de hasta 11,95 Gb/s basándose en un micro-LED violetas de nitruro de Galio (GaN) [47]. Por otro lado, la IEEE ha creado, dentro del grupo de trabajo 802.15 centrado en redes inalámbricas personales (WPAN: Wireless Personal Area Network), un grupo especializado para la estandarización de redes de comunicación mediante luz visible: el Task Group 7 (TG7). Dicho grupo trabaja desde enero del 2009 en la implantación de unos estándares de comunicación mediante LiFi para su posible difusión masiva, y cabe reseñar su aportación en cuanto a modulación de la luz se refiere, desarrollando el CSK. Por último, la empresa pureLiFi comercializa a día de hoy los sistemas que funcionan con esta tecnología, consiguiendo una capacidad efectiva superior a 1 Gb/s, unas 100 veces superior a su competidor: el WiFi [46].

2.2.4 Aplicaciones militares

En el ámbito militar, las FSOC no se han implementado extensivamente debido a los defectos que presentan. Los impedimentos en la estabilidad del enlace suponen todavía a día de hoy una desventaja que los sistemas de telecomunicaciones militares no se pueden permitir, sistemas que deben de operar en condiciones muy adversas y que requieren gran fiabilidad y robustez. Sin embargo, esto no quiere decir que las FSOC hayan pasado desapercibidas en dicho ámbito, sino más bien todo lo contrario. Las agencias de investigación militares, y en particular las de los Estados Unidos, se han interesado desde principios de siglo en estas tecnologías, y han contribuido en gran medida a su desarrollo, aunque las aplicaciones sean escasas.

El proyecto en el ámbito FSO más antiguo data del 2003. En efecto, en el presupuesto de la DARPA (*Defense Advanced Research Projects Agency*) de dicho año, se menciona por primera el proyecto ORCA (*Optical & RF Combined Adjunct*) [48]. Este proyecto emana de un proyecto anterior, THOR (*Tera Hertz Operational Reachback*), que buscaba crear un sistema de telecomunicaciones troncal de gran capacidad y que pudiese ser empleado en el ámbito terrestre, aéreo y naval. En THOR, se pretendía además que el sistema permitiese una capacidad de alta maniobrabilidad, creando así lo que acuñaron como MFSO (*Mobile Free Space Optics*). Para implementar dicha capacidad, se ideó el apéndice ORCA, sistema híbrido pero con una componente de FSOC, y desde dicho año la agencia ha ido invirtiendo en el proyecto, como podemos ver en los presupuestos de 2012 [49] y 2013 [50]. Para su desarrollo, se menciona el denominado ORCLE (*Optical & RF Communications Link Experiment*), que hace la función de prueba de concepto de la carga útil de ORCA, que no es más que el módulo encargado de las FSOC, y en el que han participado diferentes empresas como por ejemplo AOA Xinetics. No se sabe más acerca del proyecto, aparte de que sigue financiado y por lo tanto en proceso.

Sin embargo, cabe destacar la aportación de otros centros de investigación americanos del ámbito militar, en concreto de la Army. En marzo de 2005 se emitió un estudio de la Dirección de Sistemas de Transmisión de la Comandancia de Ingeniería de Sistemas de la Información (USAISEC: *Transmission Systems Directorate, U.S. Army Information Systems Engineering Command*) titulado “*Analysis of Free Space Optics as a Transmission Technology*”, en el que se estudia el potencial de las FSOC, así como sus limitaciones [10]. Por lo tanto, ORCLE y el mencionado estudio son pruebas fehacientes de que ya desde el 2003 las FSOC están muy presentes en los investigadores de los diferentes organismos de investigación militares.

A día de hoy, también hay otro organismo de la Army americana que está trabajando en las FSOC y que es digno de mención, cuyo estudio se basa en una tecnología innovadora. En 1999, el químico Ahmed H. Zewail ganó el Premio Nobel de Química por sus trabajos en la femtoquímica. Zewail estudió las reacciones químicas en tiempos muy reducidos para observar como los átomos de las moléculas se reorganizaban, inaugurando así un nuevo campo de la química. Para ello desarrolló una nueva tecnología: los USPL (*Ultra Short Pulse Laser*). Estos Láseres emiten pulsos extremadamente

reducidos, del orden de los picosegundos (10^{-12} s) e incluso de los femtosegundos (10^{-15} s), lo cual permite, para una potencia media de radiación, alcanzar un pico de radiación muy elevado. Tanto es así que con estos Láseres se estudia un nuevo campo en el de la óptica, que es el de la óptica no lineal. En efecto, la intensidad alcanzada hace que la interacción con la materia se vea distorsionada, y principios físicos de la óptica como es la superposición no sean aplicables.

Es de dicha interacción distorsionada que el Virginia Military Institute, en conjunto con la empresa Attochron, ha estudiado los efectos beneficios del empleo de estos Láseres no como modulación, sino en lugar de los Láseres de onda continua (CW: *Continuous Wave*) que se emplean a día de hoy [29]. Dichos estudios descubrieron que el empleo de USPL permite, en condiciones atmosféricas adversas como lluvia o niebla, una ganancia en recepción de 25 a 30 dB, como se puede ver en la gráfica de la Figura 2-30 (b), así como una ganancia de 15 dB en condiciones favorables pero con gran turbulencia, debido a que sufre menos distorsión, como se puede ver en la Figura 2-30 (a). Para ello emplearon USPL con una longitud de onda de 1550 nm y con pulsos de 100 fs, alcanzando una velocidad de 1 Gb/s a 3 km de distancia.

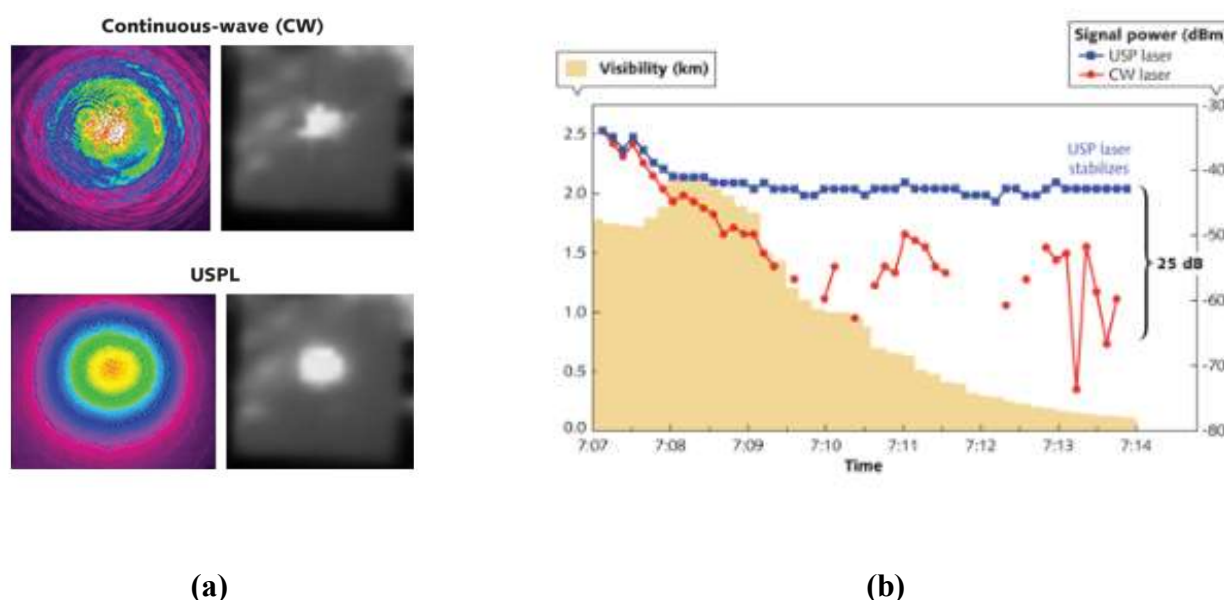


Figura 2-30 (a) Comparativa de Láser CW y USPL mediante fotografía infrarroja [23], (b) Gráfico de potencia de la señal en recepción de Láser CW (rojo) y USPL (azul) en función de la visibilidad [23]

Por último, cabe destacar el sistema TALON (*Tactical Line-of-sight Optical Network*), actualmente en desarrollo por el Naval Research Laboratory (NRL) de la Navy americana, dentro del programa de Future Naval Capabilities (FNC), en conjunto con un consorcio de empresas, a saber, Exelis Inc., Novasol Inc., IJK Inc. y LGS Inc [51], y reflejado en las Figura 2-31 (a) y (b). El desarrollo de este sistema prometedor consta de tres fases, encontrándonos a día de hoy en la tercera, y habiendo completado las dos fases anteriores con resultados brillantes.

El sistema tiene el mismo principio de funcionamiento que los sistemas convencionales terrestres mencionados en el apartado 2.2.3, y fija su objetivo en la implementación tanto en un segmento terrestre, como uno naval y uno aéreo. Este sistema supone una revolución en las FSOC ya que es el primer sistema de este estilo que ha asegurado una comunicación a más de 50 km de distancia gracias a numerosos avances y técnicas contra las limitaciones, aunque todavía sigue en fase de desarrollo.

La fase 1 culminó con un ensayo del prototipo en el Camp Pendleton, asegurando un enlace de más de 4 km de distancia con un ancho de banda de 100 Mb/s, una latencia de menos de 50 ms y una tasa de pérdidas inferior al 1%. Cabe destacar que el sistema se implantó en un mástil de 33 metros de alto y a pesar de un viento de 30 nudos no hubo pérdidas por fallo en el sistema de seguimiento. La

fase 2 culminó a su vez con un ensayo en la Naval Air Weapons Station de China Lake, ensayo que fué todo un éxito ya que se obtuvieron los mismos resultados que en el primer ensayo, pero a 50 km de distancia y con una tasa de pérdidas inferior a 0,1%. Dicho sistema está por lo tanto capacitado para transmitir tanto datos como VoIP (Voz sobre IP) y videoconferencias. Actualmente, las líneas de investigación se centran sobre todo en el seguimiento automático, la seguridad, la gestión de la red, así como la SWAP (*Space, Weight And Power*).



(a)



(b)

Figura 2-31 (a) Póster publicitario de sistema TALON [51], (b) Módulo FSOC de TALON [51]

2.3 Sistemas de adquisición y seguimiento

2.3.1 Principios de funcionamiento

Para asegurar la comunicación entre las partes de un sistema FSO, se requiere de un sistema que permita el seguimiento de una estación remota para poder establecer el enlace [14]. A excepción del caso particular de la tecnología LiFi, todos los sistemas de FSO necesitan de un apuntamiento al módulo receptor de la estación complementaria, ya que al tratarse de un haz extremadamente directivo (del orden de los mrad), no podemos contar con que, una vez en el aire, la onda se propagará hasta la estación receptora, como es el caso por ejemplo de las radiocomunicaciones. Siendo así, diversos sistemas han sido desarrollados, cada uno con ventajas e inconvenientes, aunque cabe destacar que en la mayoría de las aplicaciones se encuentran más de uno, complementándose los unos a los otros. En concreto, se tendrá que sopesar la precisión del sistema con la movilidad y el FOV (*Field of View*), aplicando en cada caso lo que más convenga.

En primer lugar, un sistema ampliamente empleado en las comunicaciones es el de la detección de una baliza luminosa. Desde la implantación de un simple LED para aplicaciones en tierra hasta el empleo de un Láser con un FOV (*Field Of View*) suficiente como para mantener un satélite en seguimiento, es el método que a día de hoy ofrece mayores posibilidades en cuanto a precisión y alcance se refiere [14] [52]. En concreto, este método consiste simplemente en disponer un haz con una dispersión elevada de forma que, al detectarlo la estación complementaria, esta se centra en la baliza, percibiendo la estación de esta última el haz de la otra. Las balizas suelen ser un haz Láser que mediante una lente se hace divergir, aunque también pueden ser LEDs potentes. En cuanto al detector de la baliza, se suelen emplear FPA (*Flat-Plane Arrays*) o PSD (*Position Sensitive Diode*) para seguimientos menos precisos, y QPD (*Quadrant Photo Detector*) para seguimientos que requieran de mayor precisión.

Los FPA son *arrays* de fotodetectores que cuando la luz incide en ellos, liberan una cantidad determinada de electrones debido al efecto fotoeléctrico. Dicha cantidad dependerá de la intensidad

luminosa y del tiempo de exposición, y puede ser medida de forma que se le atribuya un valor determinado de luminosidad. Si además tenemos en cuenta que se emplea un filtro Bayer, es decir un filtro de colores primarios (rojo, azul y verde) como el de la Figura 2-32, dispuesto por encima de los fotodetectores, podemos determinar en cada fotodetector (o “píxel”) un color con una determinada intensidad. De esta forma, conseguimos obtener una imagen y por lo tanto sirve para detectar un haz de luz de una baliza determinada [14] [53]. Los FPA pueden ser de dos tipos: CCD (*Charged-Couple Device*) o sensores CMOS (*Complementary Metal-Oxide-Semiconductor*).

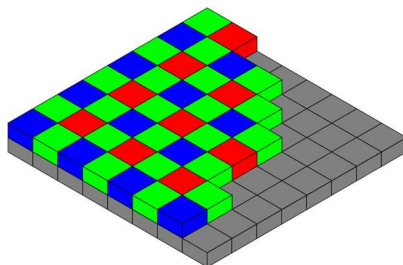
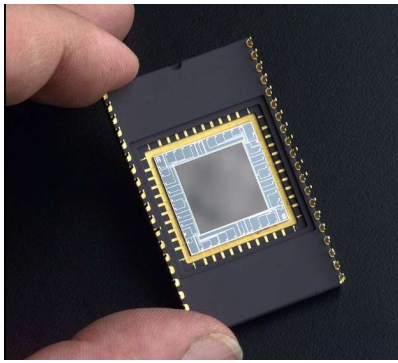


Figura 2-32 Filtro Bayer [54]

Los CCD consisten en capas de algún material fotoactivo dispuesto en parcelas (llamadas fotositos) encima de una capa de óxido de silicio, a su vez dispuesto encima de una capa de silicio dopado positivamente (normalmente con boro). Cuando los fotones inciden en el fotosito, hacen liberar electrones debido al efecto fotoeléctrico, que quedan atrapados entre el óxido de silicio y el silicio dopado, en lo que comúnmente se conocen como “pozos de potencial”. Dichos electrones son luego transferidos a unos canales de silicio cargados aún más positivamente (denominado silicio p+, que puede ser por una concentración mayor de iones positivos o por un voltaje inducido), que conducen los electrones para su procesamiento. Estos canales estaban antiguamente dispuestos de manera uniaxial, lo que hacía que la información se tenía que procesar por filas, pero a día de hoy se implementan en dos ejes, lo cual permite un procesamiento mejor y más rápido. En apariencia, son muy parecidos a los CMOS, como el de la Figura 2-33 (a).

En cuanto a los sensores CMOS, como el de la Figura 2-33 (a), emplean una disposición de transistores MOSFET de tipo n y p de tal forma que cuando los fotones inciden en el fotosito, se transfiere un voltaje a las “Gates” de los MOSFET, dejando pasar la corriente. Este tipo de FPA ofrece la ventaja de que permite amplificar la señal de cada fotosito, ya que se pueden integrar amplificadores en cada uno de ellos, mientras que en los CCD la señal se amplifica fuera de la matriz. Además, los CCD están sujetos a un efecto denominado “*blooming*”, que se refiere a cuando la intensidad de la luz es demasiado elevada y los pozos de potencial se saturan, parte de la carga eléctrica negativa se transfiere a los fotositos adyacentes, creando un efecto de deslumbre en la imagen, como en la Figura 2-33 (b). Los FPA se emplean normalmente para las cámaras de los dispositivos electrónicos, aunque cabe reseñar que el más popular es el CMOS, utilizado a día de hoy en la mayoría de cámaras de los teléfonos móviles, cámaras DSLR, etc. En las FSOC, permiten por lo tanto seguir una determinada baliza con una frecuencia definida, pero debido a que estos dispositivos no permiten determinar la inclinación del rayo incidente se suelen emplear en seguimientos de menor precisión, o acoplados a otros sistemas de mayor precisión [14] [53].



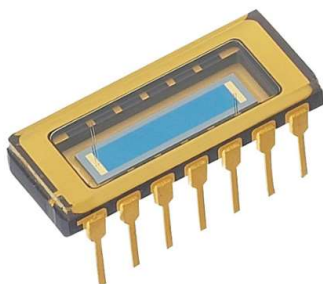
(a)



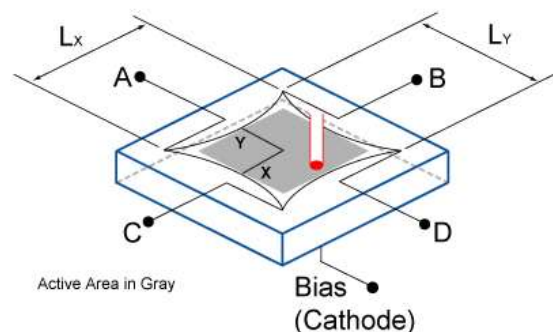
(b)

Figura 2-33 (a) sensor CMOS [66], (b) Fotografía con efecto *blooming*

Otro detector más sencillo que se suele emplear a la hora de detectar una baliza es el PSD (*Position Sensitive Diode*), que podemos ver en la Figura 2-34 (a). Este diodo fue inventado en 1957 por J.T. Wallmark, y aprovecha también el efecto fotoeléctrico, que no es más que la liberación de electrones debido a la absorción de la energía electromagnética de una onda de tal naturaleza, además del denominado efecto lateral. Estos diodos son de naturaleza PIN, es decir que constan de tres partes: un semiconductor negativo, un semiconductor positivo y un semiconductor sin dopar (denominado intrínseco) entre estos. En el PSD, estos semiconductores están dispuestos de forma que el bloque central se corresponde con la parte intrínseca del diodo, la base del diodo se corresponde con la parte del semiconductor positivo y el semiconductor negativo se corresponde con cuatro electrodos dispuestos en los bordes del semiconductor intrínseco, correspondientes con los bornes A, B, C y D en la Figura 2-34 (b). Cuando se emplea como un fotodiodo, el diodo PIN se polariza a la inversa sin llegar a la saturación, de forma que cuando un fotón con energía suficiente genera un electrón, lo que produce es un par electrón-hueco en el diodo de forma que se crea una corriente. Dicha corriente es diferente en cada uno de los electrodos, y la diferencia entre estas depende de la desviación respecto al centro del diodo del punto de incidencia del fotón, lo que permite así obtener las coordenadas del punto de incidencia del fotón respecto del centro del diodo. Al igual que con los FPA, obtenemos una desviación respecto al centro pero no un ángulo, lo que resulta en un sistema menos preciso [14].



(a)



(b)

Figura 2-34 (a) Fotografía de un PSD [55], (b) Esquema de un PSD [55]

Por último, los QPD (*Quadrant Photo Detector*), como el de la Figura 2-35, se imponen como los detectores más precisos, y a la vez más sencillos en su arquitectura [14] [55]. Estos se componen únicamente de varios fotodiodos (también de estructura PIN en polarización inversa, aunque ocasionalmente se emplean diodos de tipo PN) dispuestos alrededor del centro del diodo. En general se emplean 4 diodos, y el funcionamiento de este sistema consiste en medir la diferencia de intensidad entre las cuatro partes, pudiendo así determinar con que inclinación el rayo incide en el plano de los fotodiodos. Siendo así, al redirigirse el diodo, las intensidades se equiparán cuando el vector normal del plano del diodo sea colinear a la dirección del rayo incidente, y por lo tanto queden enfrentadas las dos estaciones participantes en el enlace. Este método se emplea en multitud de aplicaciones a larga distancia y que requieren de gran precisión, como pueden ser los satélites [14] [52].

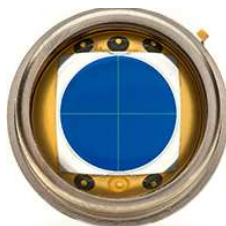


Figura 2-35 Fotografía de un QPD [56]

En segundo lugar, otro sistema empleado con recurrencia para adquisición y seguimiento de la estación receptora son los sistemas que emplean un enlace secundario por radiofrecuencia. Este enlace puede estar previsto tanto como un enlace secundario en caso de que las FSOC fallen como simplemente una parte del sistema de adquisición y seguimiento. Como se ha visto en el 2.2.3, los sistemas FSOC pueden estar acoplados a un sistema de radiotelecomunicación convencional, sobre todo en aplicaciones terrestres, y este puede también participar parcial o totalmente en la labor que nos atañe en este capítulo [14]. Para tal propósito, el módulo de adquisición y seguimiento debe de tener datos de la posición de la estación propia mediante un receptor GPS o un receptor GPSD (GPS Diferencial) para mayor precisión (este último del orden de los centímetros), y opcionalmente de la cinemática (rumbo, velocidad, acelerómetro,...). Siendo así, las estaciones difunden su posición y los datos relevantes de su cinemática para que la estación complementaria adquiera la estación propia, la siga y en caso de perderla, la readquiera con facilidad, como se puede ver en la secuencia de las Figura 2-36 (a), (b) y (c).

Cabe reseñar que la omnidireccionalidad de la radiodifusión permite una comunicación constante, además de la nimia importancia que supone en pérdidas para el enlace los meteoros a los obstáculos, sobre todo a bajas frecuencias y en comparación con las FSOC. Este tipo de seguimiento permite un gran dinamismo, teniendo un rango de acción en la mayoría de los casos no limitante (veremos a continuación que las plataformas también suponen un limitante tanto en precisión, como en rango y velocidad angular de seguimiento), una velocidad de seguimiento alta y una gran capacidad de readquisición. Es por lo tanto idóneo para sistemas móviles como pueden ser sistemas montados en un avión, en un coche o en un tren a alta velocidad.

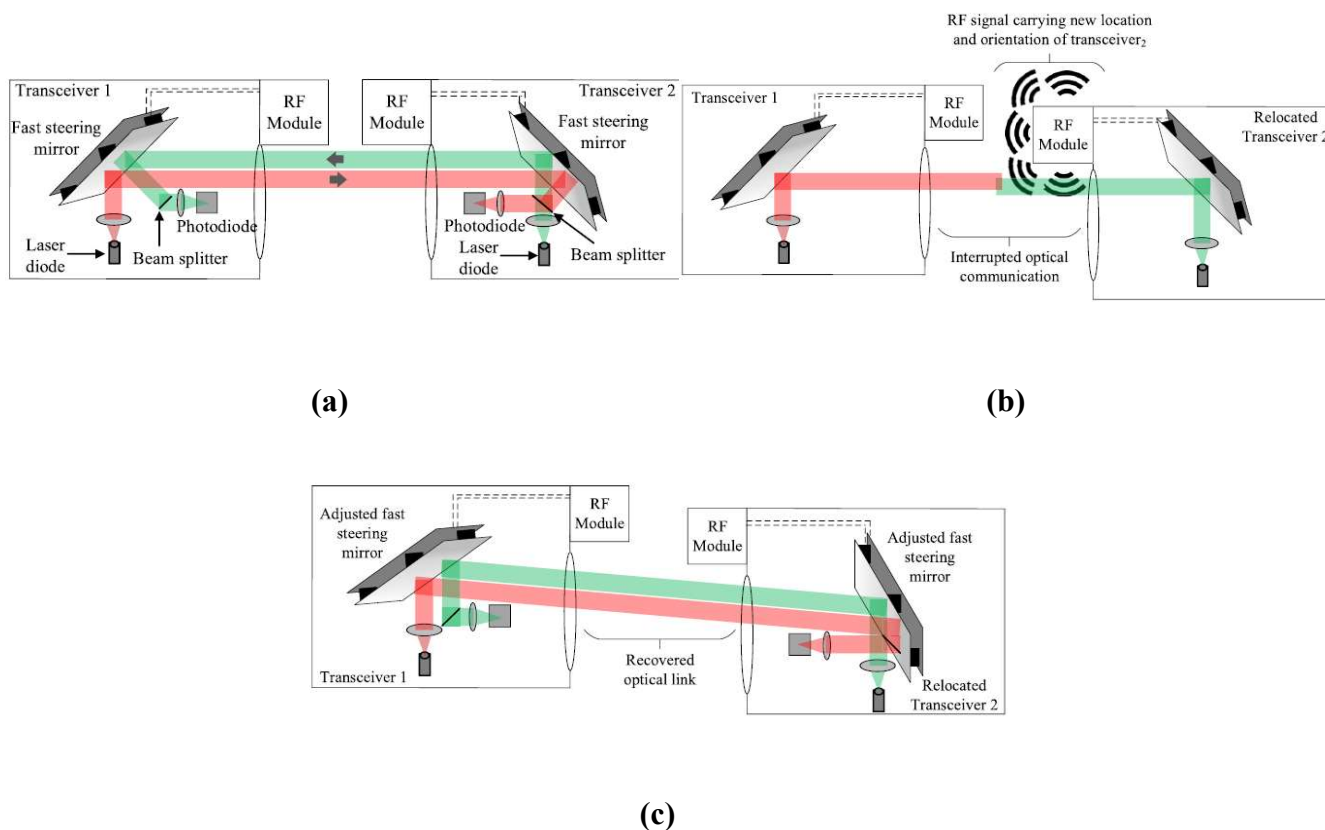


Figura 2-36 (a) Enlace inicial entre dos estaciones [45], (b) desalineamiento y transmisión de posición mediante radiofrecuencia [45], (c) realineamiento [45]

Por último, varios documentos proponen un sistema que consista en que las dos estaciones, en la fase de adquisición, realicen un patrón de búsqueda mediante un haz de luz estrecho [14] [53]. Dichas estaciones harían un patrón definido (ráster, rotatorio en sentido horario o antihorario,...), que puede ser optimizado, como exhibe el estudio [53], y estarían constantemente mandando un mensaje SYN, a la espera de realizar un “*Three-way handshake*”. Cuando coincidan enfrentadas, las estaciones realizarán el mencionado protocolo mediante el intercambio de mensajes SYNACK y ACK, pasando a continuación datos sobre la cinemática de cada uno para efectuar el seguimiento mutuo. Este sistema permitiría discreción, a la vez que una buena capacidad de seguimiento en un contexto demandante por el dinamismo de las estaciones. Sin embargo, la readquisición es algo más tediosa y por lo tanto requiere de alguna mejora. En esta línea, se han planteado sistemas como esferas de fotodiodos y diodos emisores, en las que no existe un movimiento mecánico de la plataforma y por lo tanto de sus sensores, sino que estos están siempre “a la escucha”, y por lo tanto haría mucho más eficiente el *modus operandi* mencionado en este párrafo. Dicha esfera estaría dividida en celdas siguiendo un patrón de colmena, y corresponderían con una dirección determinada del espacio, como se puede ver en la Figura 2-37.

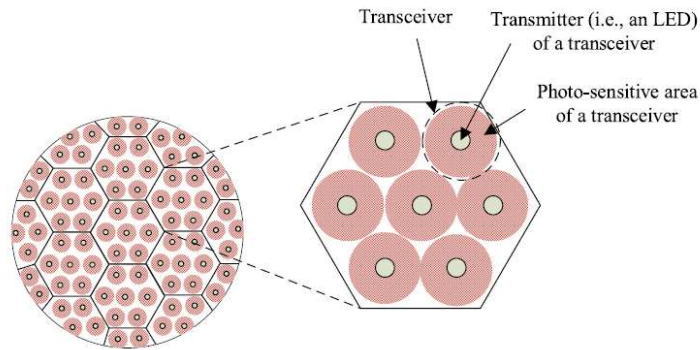


Figura 2-37 Esquema de un sistema de seguimiento esférico [14]

A día de hoy, se suelen emplear varios de estos sistemas en conjunto, creando un sistema de adquisición y seguimiento híbrido. En dichos sistemas se suele disponer de varios sistemas, explotando las ventajas de cada uno en un momento preciso, como por ejemplo en un sistema de seguimiento satelital, en el que se suele emplear un sistema menos preciso, pero de mayor rango y velocidad, como es el caso de los FPA, para las fases de adquisición y seguimiento inicial, a la par con un sistema basado en QPD, que permite hacer un seguimiento más preciso [14].

2.3.2 Plataformas FSOC

Las diferentes modalidades de plataformas presentes a día de hoy ofrecen múltiples ventajas y desventajas que atañen directamente a la adquisición y seguimiento de las estaciones remotas, ya que de la precisión, velocidad de actuación y rango de acción de estas depende en última instancia las características que ofrece el sistema. No solo la naturaleza del sistema influye, sino también la maquinaria empleada, suponiendo esta en la mayoría de los casos el limitante de los sistemas de seguimiento en cuanto a los aspectos mencionados previamente. En este apartado veremos las diferentes familias de plataformas existentes a día de hoy, que son las reflejadas en la Figura 2-38 [14].

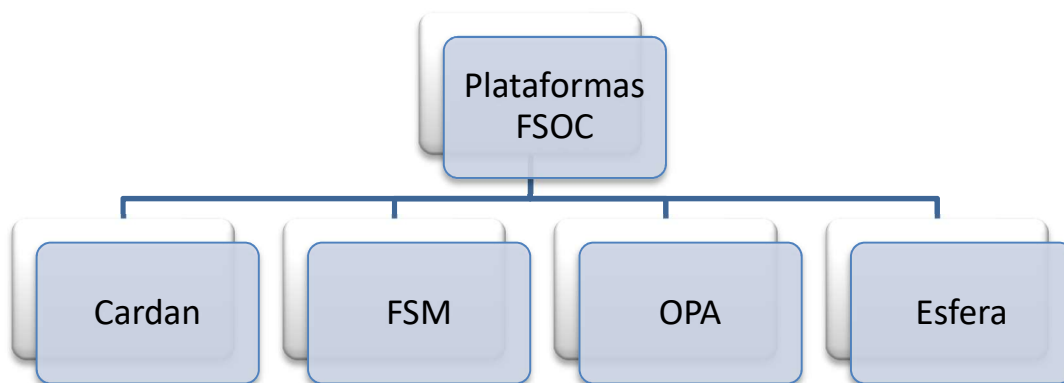


Figura 2-38 Diagrama de tipos de plataformas FSOC [14]

La primera familia que se puede mencionar es la de los cardan, es decir sistemas que permiten orientar la fuente mediante giros en dos o tres ejes. La estructura en sí del sistema se basa en el giro de todo el módulo de transmisión mediante motores y una transmisión mecánica precisos. Se usan principalmente en casos en los que el rango del sistema FSOC ha de ser amplio, es decir en casos en los que se requiera que el sistema FSOC sea capaz de cubrir un amplio abanico de direcciones en el espacio. Además, aunque no sean los más precisos a día de hoy, se pueden conseguir resoluciones de seguimiento del orden de los μ rad, resolución que es suficiente en la mayoría de los casos [14]. Un caso en el que son favoritos estos sistemas son los sistemas FSOC embarcados en satélites, así como sus estaciones complementarias en Tierra. Dichos casos requieren de un amplio rango, sobre todo en

los dos ejes de giro perpendiculares al haz de luz, así como de precisión, y por lo tanto este tipo de plataforma resulta idónea para el diseño. Un claro ejemplo es el módulo del LLCD mencionado en el apartado 2.2.1, que consiste en una plataforma cardan de dos ejes con un sistema de seguimiento basado en los QPD [52] [22].

En segundo lugar, la familia de los espejos, más concretamente de FSM (*Fast Steering Mirrors*), es la familia que permite obtener de las resoluciones mejores (solo igualadas por OPA), por debajo del μ rad, así como velocidades angulares [14]. Estos sistemas se basan en redirigir el haz Láser con la ayuda de un espejo, lo que permite mayor precisión a la vez que velocidad de seguimiento, debido principalmente a su estructura liviana. Los FSM se dividen en tres subfamilias dependiendo del mecanismo de orientación del espejo: los convencionales, los PZT (piezoeléctricos) y los deformables.

Los espejos convencionales son espejos lisos de alta reflectividad accionados por bobinas. Estos tienen en la propia estructura del espejo una serie de imanes que son atraídos o repelidos por dichas bobinas, parecidas a las bobinas presentes en los altavoces. Estos espejos permiten el rango mayor dentro de los FST y aportan robustez, así como la posibilidad de usar espejos más grandes que en las dos otras subfamilias, pero se ven afectados en tanto en cuanto la frecuencia de respuesta a los estímulos está limitada por las bobinas [14].

Por otro lado, los espejos accionados por materiales piezoeléctricos consisten en material de este tipo, apilado en capas en el seno de un cilindro de acero, que mediante un voltaje modulado se expande y permite orientar el FSM. Estos sistemas permiten mover espejos de un tamaño considerable y tienen una resolución extremadamente buena en comparación con los espejos convencionales. Sin embargo, son los espejos más limitados en cuanto rango ya que este se encuentra normalmente por debajo del mrad, lo cual les sitúa normalmente en la familia de los híbridos [14].

Por último, la tecnología MEMS (*Micro ElectroMechanical System*) se aplica en las FSOC, dando lugar a la tercera subfamilia de espejos deformables [14]. Situándose en el ámbito de los MOEMS (*Mechanic Opto-ElectroMechanical System*), los espejos deformables están constituidos de superficies reflectantes micrométricas, así como de electrónica basada principalmente en silicio y maquinaria a dicha escala. Su funcionamiento se basa en actuadores independientes los unos de los otros mueven unas superficies reflectantes, o a una membrana reflectante y deformable dispuesta encima de estos, pudiendo modificar la onda, como podemos ver en la Figura 2-39 (b). Actuando en esta, puede modificar tanto el frente de onda para reconstruirlo después de las deformaciones que este haya sufrido en su propagación por el medio o para redirigir el haz hacia una dirección determinada, como su fase o amplitud. Estos espejos requieren de poca potencia para operar, y ofrecen una gran versatilidad, ya que son muy precisos a la par que rápidos en cuanto a respuesta se refiere. No están tan limitados como los piezoeléctricos, pero suelen necesitar de un cardan para ampliar el rango, ya que este en muchas ocasiones no suele ser suficiente. Una particularidad que tienen estos espejos es que pueden ser empleados tanto para posicionamiento como para modulación de una señal. En efecto, al poder modificar la amplitud y fase de la señal, estos espejos pueden ser empleados a la vez como módulo de seguimiento y de transmisión, ahorrando un sistema adicional únicamente dedicado a esta, como es el caso en la Figura 2-39 (a).

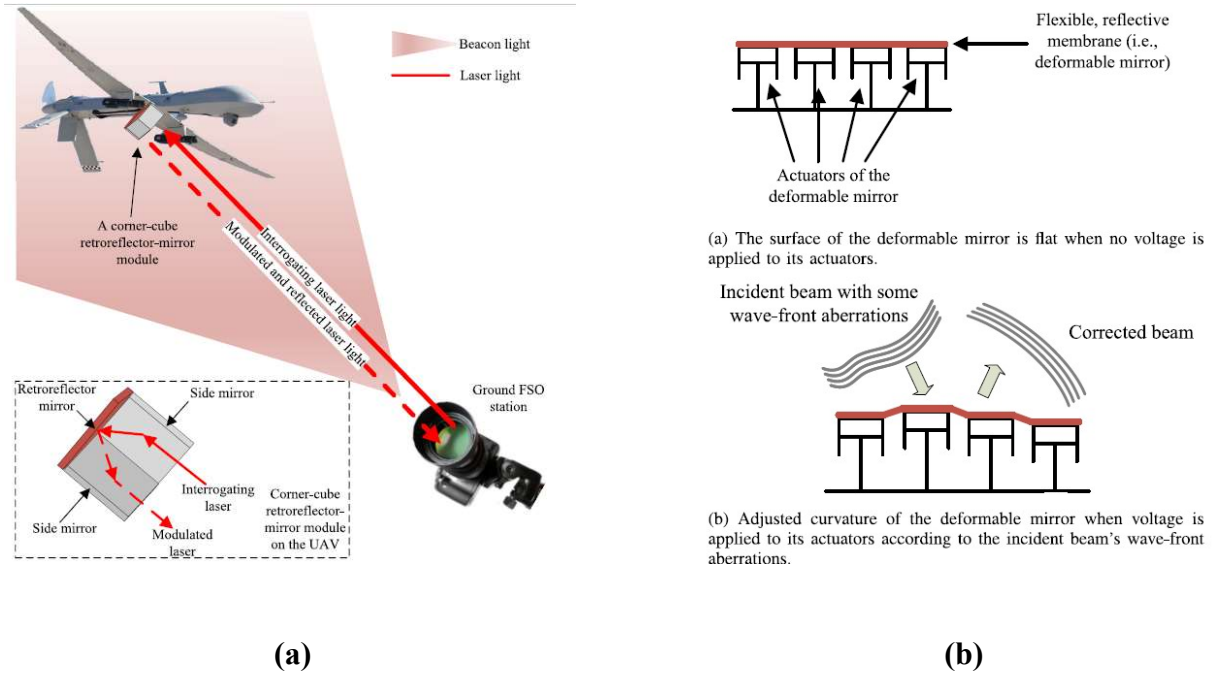


Figura 2-39 (a) Esquema de funcionamiento de un sistema FSO basado en un espejo deformable MOEMS tanto para seguimiento como para comunicación [45], (b) Funcionamiento de un espejo deformable MOEMS con membrana [45]

Los OPA (*Optical Phased Arrays*) nacen de los de los “*Phased arrays*”, ampliamente usados en ámbitos como la detección RADAR (*Radio Detection And Ranging*). Estos últimos no son más que un *array* de antenas que transmiten la misma señal, pero con cierto desfase. Este desfase entre cada onda permite modificar la dirección del frente de onda, desfase que se controla por lo tanto para dirigir el haz hacia una determinada dirección del espacio. Además, estos *arrays* permiten una direccionalidad bastante elevada respecto a las antenas convencionales, lo cual se traduce en una ganancia elevada, y un SLL (*Secondary Lobe Level*) extremadamente reducido. De este principio de modificar el frente de onda mediante un desfase entre diferentes puntos emisores nacen los OPA, cuyo funcionamiento es idéntico, pero en el dominio de las ondas visibles. En los OPA, la fase de la luz se controla para orientar el haz mediante tecnología basada en cristal líquido (LC: *Liquid Cristal*), cuya tecnología se basa en la manipulación del índice de refracción. La luz se crea en una misma fuente, y se hace pasar por una matriz de LC [14]. Esta está subdividida en píxeles que no son más que conductos diminutos con cristal líquido, material con una estructura cristalina que puede reorganizarse en base a la aplicación de un campo eléctrico (mediante una cadena uniaxial de decenas de miles de electrodos microscópicos) y cuya reorganización resulta en un cambio del índice de refracción del medio. El índice de refracción a su vez representa una determinada velocidad de propagación de la luz en el medio, como podemos ver en la fórmula (12).

$$n = \frac{c}{v_p} \quad (12)$$

Donde n es el índice de propagación, c la velocidad de propagación en el vacío y v_p la velocidad de propagación en el medio. Modificando v_p , el haz de luz entrante en el píxel tardará más o menos en salir de este que los demás, lo cual resulta en una desfase al igual que en su análogo de las radiofrecuencias, como se puede apreciar en la Figura 2-40. Modificando el voltaje aplicado en cada píxel podremos determinar una dirección del espacio en concreto en la que el haz se dirigirá. Cabe reseñar que este tipo de OPA se emplean con LC denominados transmisivos, pero también se pueden emplear LC reflectivos, en los que el desfase proviene de la reorganización de los cristales cuyas superficies reflejan la luz. En este caso, el simple camino que recorre la luz entre los cristales la

desfasa. Estos sistemas también ofrecen una precisión por debajo del μrad , además de ser un equipo que requiere de poca potencia, tener un rango amplio y ser versátil ya que pesa poco, y por lo tanto puede ser acoplado a otro sistema de seguimiento fácilmente. Además, cabe reseñar que la falta de elementos mecánicos a mayores lo convierte en un equipo muy fiable, ventaja que lo convierte en idóneo para aplicaciones en las que reparar el dispositivo sería imposible o muy costoso (por ejemplo, en el espacio).

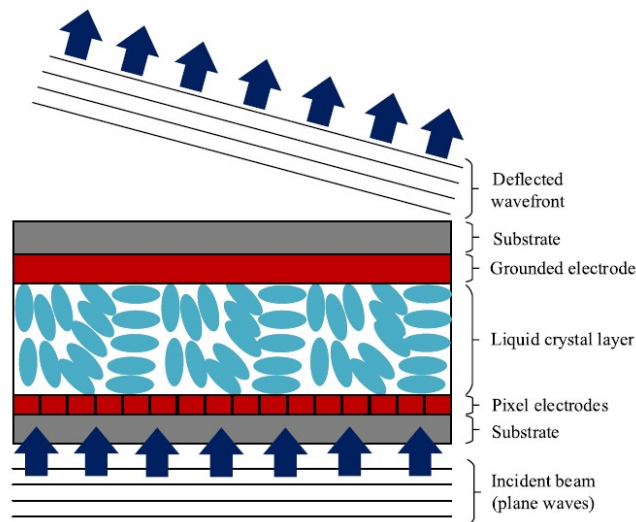


Figura 2-40 Esquema de un OPA basado en cristal líquido transmissivo [45]

Por último, la esfera de la que se habló en el apartado 2.3.1, es una plataforma que, aunque todavía no se ha materializado en ningún proyecto en concreto, sería de gran utilidad en el caso de que la precisión no fuese primordial, pero la respuesta y sobre todo el rango si lo fueran. Esta esfera que estaría dividida en parcelas de fotoreceptores dispuestos en un patrón de colmena, permitiría una resolución del orden del mrad (aunque todavía está por desarrollar), pero al no disponer de ningún elemento mecánico y ofrecer una superficie tan extensa, sería un sistema extremadamente fiable y rápido en cuanto a seguimiento, ya que este depende solo del procesamiento de las corrientes provenientes de las parcelas.

A modo de conclusión, podemos resumir todas las plataformas de sistemas FSO en la Tabla 4.

ATP Mechanism	Pointing Resolution (rad)	Field of Regard	Angular Steering Speed	Mechanics	Dimensionality	Application Scope
Gimbal-based	μrad	Wide	Low	Mechanical	2D	Any
Mirror-based	Sub- μrad	Narrow	High	Mechanical, piezoelectric and electromechanical	2D, 3D	Any
Gimbal-Mirror Hybrid	Sub- μrad	Wide	High	Mechanical, piezoelectric and electromechanical	2D, 3D	Any
Liquid-crystal-based	Sub- μrad	Narrow for one stage	High	Electrical	2D	Terrestrial communications
Spherical Node Design	N/A	360°	N/A	No moving parts (spatial re-use)	3D	Ground/vehicle-to-vehicle/ground

Tabla 4 Comparativa de plataformas FSOC [45]

3 DESARROLLO DEL TFG

3.1 Sistema de seguimiento

3.1.1 Material

Para desarrollar el sistema de seguimiento del sistema FSOC, emplearemos como sensor principal una Raspberry Picamera. Este dispositivo es un CCD implementado con un filtro Bayer convencional, con lectura de los fotositos uniaxial. En concreto, el funcionamiento interno de la cámara se basa en el hecho de que esta no dispone de un diafragma físico que se abre para la captura un tiempo determinado y se cierra, como puede ser el caso de las cámaras DSLR. En la Picamera, se emplea un reseteo sucesivo en cada fila, y una lectura de esa fila en un tiempo determinado después de su reseteo [57]. En efecto, si tenemos una Picamera de 8x8 fotositos sin filtro Bayer, lo que recibiríamos es lo que se muestra cronológicamente en las Figura 3-1 (a), (b), (c), (d), (e) y (f). Aunque en los fotositos haya un valor acumulado de electrones, con el reseteo se consigue obtener un valor equiparado a los fotositos de las celdas anteriores.

El tiempo de exposición, que es el tiempo que tarda en registrarse los valores de todo un fotograma, tiene una duración mínima. Para leer todas las líneas de un fotograma, se requiere un tiempo mínimo determinado, proporcional a la resolución a la que se está empleando la cámara. Para maximizar la tasa de fotogramas, se empleará el menor tiempo de exposición posible, pero para emplear el máximo tiempo de exposición posible, se empleará la mínima tasa posible. El tiempo de exposición mínimo está relacionado con la capacidad de procesamiento del GPU (*Graphics Processing Unit*), mientras que el máximo está relacionado con el hardware, y en concreto con el máximo tiempo posible en el que un fotosito puede estar almacenando una lectura sin saturar (es decir sin saturar los pozos de potencial). Siendo así, la fórmula general para el tiempo de exposición será la (13).

$$t_{exp} = n \times t_{fila} \quad (13)$$

Donde t_{exp} es el tiempo de exposición de un fotograma, n el número de filas en este y t_{fila} es el tiempo en el que se tarda en procesar una fila. Este último es el que esta por lo tanto limitado. El tiempo t_{fila} mínimo tiene que ver con la velocidad máxima a la que se puede procesar las lecturas, mientras que el tiempo t_{fila} máximo tiene que ver con la propia saturación de los fotositos. La tasa de fotogramas vendrá determinada por la fórmula (14).

Sensor elements									->	Frame 1								
0	0	0	0	0	0	0	0	0										
0	0	0	0	0	0	0	0	0										
0	0	0	0	0	0	0	0	0										
0	0	0	0	0	0	0	0	0										
0	0	0	0	0	0	0	0	0										
0	0	0	0	0	0	0	0	0										
0	0	0	0	0	0	0	0	0										
0	0	0	0	0	0	0	0	0										
0	0	0	0	0	0	0	0	0										

(a)

Sensor elements									->	Frame 1								
0	0	0	0	0	0	0	0	0	Rst									
1	1	1	1	1	1	1	1	1										
1	1	1	1	1	1	1	1	1										
1	1	1	1	1	1	1	1	1										
1	1	1	1	1	1	1	1	1										
1	1	1	1	1	1	1	1	1										
1	1	1	1	1	1	1	1	1										
1	1	1	1	1	1	1	1	1										
1	1	1	1	1	1	1	1	1										

(b)

Sensor elements									->	Frame 1								
1	1	1	1	1	1	1	1	1										
0	0	0	0	0	0	0	0	0	Rst									
2	2	2	2	2	2	2	2	2										
2	2	2	2	2	2	2	2	2										
2	2	2	2	2	2	2	2	2										
2	2	2	2	2	2	2	2	2										
2	2	2	2	2	2	2	2	2										
2	2	2	2	2	2	2	2	2										
2	2	2	2	2	2	2	2	2										

(c)

Sensor elements									->	Frame 1								
2	2	2	2	2	2	2	2	2										
1	1	1	1	1	1	1	1	1										
0	0	0	0	0	0	0	0	0	Rst									
3	3	3	3	3	3	3	3	3										
3	3	3	3	3	3	3	3	3										
3	3	3	3	3	3	3	3	3										
3	3	3	3	3	3	3	3	3										
3	3	3	3	3	3	3	3	3										
3	3	3	3	3	3	3	3	3										

(d)

Sensor elements									->	Frame 1								
3	3	3	3	3	3	3	3	3	->	3	3	3	3	3	3	3	3	3
2	2	2	2	2	2	2	2	2										
1	1	1	1	1	1	1	1	1										
0	0	0	0	0	0	0	0	0	Rst									
4	4	4	4	4	4	4	4	4										
4	4	4	4	4	4	4	4	4										
4	4	4	4	4	4	4	4	4										
4	4	4	4	4	4	4	4	4										

(e)

Sensor elements									->	Frame 1								
4	4	4	4	4	4	4	4	4		3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3	->	3	3	3	3	3	3	3	3	3
2	2	2	2	2	2	2	2	2										
1	1	1	1	1	1	1	1	1										
0	0	0	0	0	0	0	0	0	Rst									
5	5	5	5	5	5	5	5	5										
5	5	5	5	5	5	5	5	5										
5	5	5	5	5	5	5	5	5										

(f)

Figura 3-1 (a), (b), (c), (d), (e), (f) Representación cronológica del proceso de captura de fotograma de la Picamera [57]

$$T = \frac{1}{t_{exp}} \quad (14)$$

Donde T es la tasa de fotogramas procesados por segundos (se emplea la unidad fps: frames per second) y t_{exp} es el tiempo de exposición en segundos. Tendremos por lo tanto las relaciones mencionadas en (15).

$$T_{m\acute{a}x} = \frac{1}{t_{exp_{min}}} = \frac{1}{n \times t_{fila_{min}}}, \quad T_{m\acute{i}n} = \frac{1}{t_{exp_{m\acute{a}x}}} = \frac{1}{n \times t_{fila_{m\acute{a}x}}} \quad (15)$$

Los rangos posibles de T vienen reflejados en la documentación de la cámara, como podemos en la Tabla 5 (la columna FOV (*Field Of View*) se corresponde con el rango de visión para dicha resolución).

#	Resolution	Aspect Ratio	Framerates	Video	Image	FoV
1	1920x1080	16:9	1 < fps <= 30	x		Partial
2	2592x1944	4:3	1 < fps <= 15	x	x	Full
3	2592x1944	4:3	1/6 <= fps <= 1	x	x	Full
4	1296x972	4:3	1 < fps <= 42	x		Full
5	1296x730	16:9	1 < fps <= 49	x		Full
6	640x480	4:3	42 < fps <= 60	x		Full
7	640x480	4:3	60 < fps <= 90	x		Full

Tabla 5 Tasa de fotogramas procesados por segundo en función de la resolución de la Picamera [57]

Por otro lado, la arquitectura de todo el hardware implicado en la cámara es algo más complejo, estando la mayoría de este en la propia Raspberry Pi que en el dispositivo. La cámara en sí dispone solo de amplificadores y registradores, que manda a la Raspberry mediante un bus. En esta última se encuentra tanto el GPU como el CPU (*Central Processing Unit*), que gestionan la información [57].

La secuencia que sigue la cámara, correspondientes con los números en la Figura 3-2 (a), son las siguientes:

1. La cámara envía la información a través del bus al GPU
2. El GPU procesa dicha información y construye la imagen. Los formatos empleados para dicho proceso son o bien RGB, que devuelve un valor para la intensidad de cada color primario en cada píxel, o YUV, que devuelve la luminosidad de cada píxel (valor Y) y la cromaticidad de cada uno de estos en un plano cromático que consta de un parámetro U y otro V. Estos formatos se pueden a mayores codificar mediante diversos códigos como MPEG para fotografías, o H264 para los vídeos.
3. Al mismo tiempo, el código que ejecutamos llama a determinadas funciones de la cámara (que en realidad son métodos dentro de las librerías de la cámara).
4. Estas funciones instancian a su vez otras de una capa de menor nivel: la MMAL (*MultiMedia Abstraction Level*).
5. La MMAL se emplea para gestionar la comunicación entre CPU y GPU a través de un interfaz: el VCHI (*VideoCore Host Interface*). El GPU emplea la RAM de la Raspberry Pi para almacenar los fotogramas o vídeos procesados, y redirige al VHCI a la posición concreta de la imagen en la memoria mediante un DMA (*Direct Memory Access*)
6. El VHCI entrega los datos al MMAL.
7. El MMAL devuelve los datos a la librería de la cámara
8. La librería de la cámara devuelve los datos instanciados por el código.

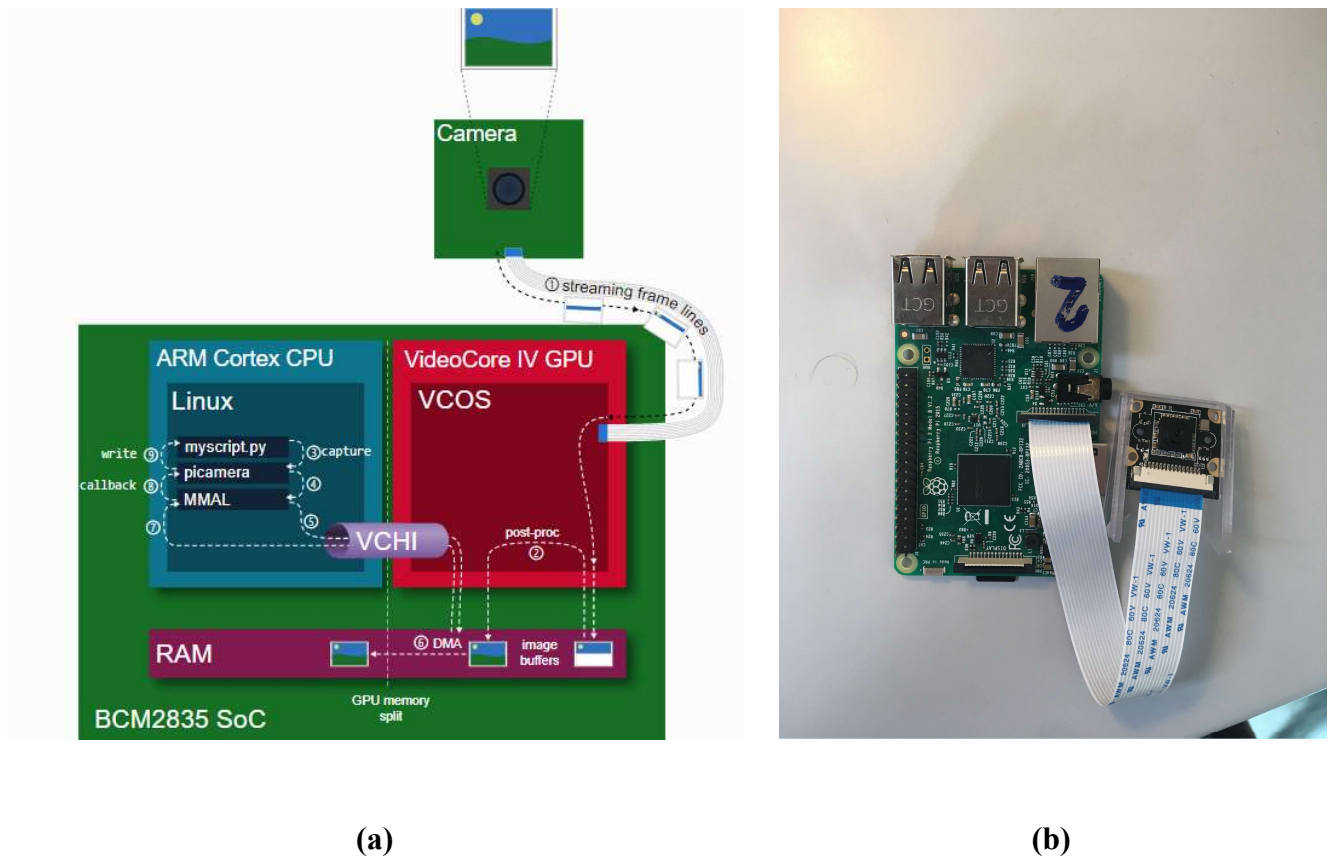
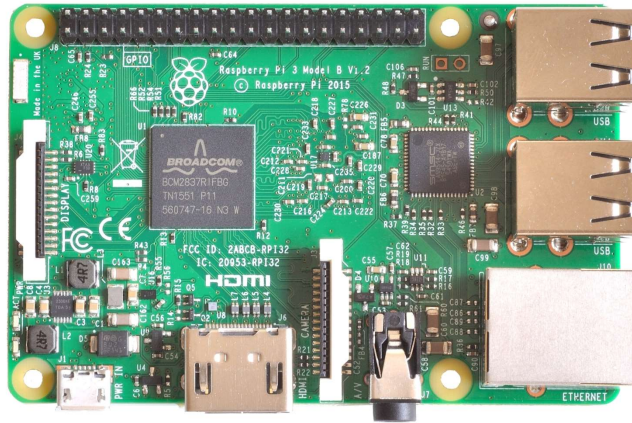


Figura 3-2 (a) Esquema del sistema integrado de la Picamera [57], (b) Fotografía de la Picamera conectada a la Raspberry Pi [57]

Además, podemos citar la propia Raspberry Pi, que no es más que un ordenador a escala reducida y que cuenta con la arquitectura básica de un ordenador (CPU, GPU, RAM,...) además de una serie de pines para aplicaciones del ámbito electrónico. El modelo del que dispondremos será el Raspberry Pi Model B v1.2, que podemos ver en la Figura 3-2 (b) y en la Figura 3-3. Este modelo cuenta con el microchip BCM2837, que incluye un procesador de cuatro núcleos ARM Cortex A53, que opera a una velocidad de procesamiento de 1,2 GHz [58]. Además, acerca del GPU, se trata del modelo VideoCore IV y opera a una velocidad de 400 MHz. Cabe destacar que requiere de un sistema operativo para su empleo, que en nuestro proyecto será el desarrollado por la Raspberry Pi Foundation (institución desarrolladora de la Raspberry Pi), el Raspbian, basado en el sistema operativo Debian, que no es más que un sistema operativo de código libre. Dicho elemento será el elemento principal de nuestro sistema tanto de seguimiento como de accionamiento de la plataforma. Todos los componentes (sensores, controlador,...) irán subordinados a este, y será por lo tanto el que ejecute el código base del sistema y del que emanen todas las órdenes para realizar el seguimiento. Por último, para usar este dispositivo se requiere de periféricos como un ratón, un teclado y un monitor, o de un cable Ethernet para permitir un control remoto del microprocesador a través de una conexión a la red de la Raspberry Pi a través del ordenador remoto.



Por último, usaremos un soporte para la cámara que se ha adquirido. Dicho soporte consiste en dos partes, una que soporta la cámara mediante dos hendiduras en las lengüetas del soporte, y otra que fija la carcasa a la plataforma mediante unión mecánica, como podemos en la Figura 3-4. El soporte cuenta con dos alas con un agujero de 5 mm de diámetro en cada una para dicha unión. Cabe reseñar que la carcasa está concebida de tal forma que la cámara se sujeta con una inclinación de 23,6° respecto a la vertical del suelo, inclinación que se tendrá cuenta a posteriori para que la cámara quede totalmente a la vertical.

3.1.2 Código principal

En primer lugar, para emplear la cámara, se deberá instanciar el objeto de la librería correspondiente a esta para determinar ciertos parámetros de funcionamiento. En concreto, determinaremos la resolución de la cámara, así como la tasa de fotogramas, como podemos ver en la Figura 3-5, aunque la librería ofrece multitud de opciones como capturar un determinado número de fotogramas o emplear un temporizador. Para evitar emplear el nombre completo del objeto,

utilizaremos la herramienta de Python denominada gestor de contexto, que permite denominar un método u objeto con otro nombre más práctico. Siendo así, el objeto de la cámara con los parámetros lo denominaremos “camara”.

```

140 with picamera.PiCamera(resolution='160X120', framerate=60) as camara:
141     with MiAnalizador(camara) as analizador:
142         port = serial.Serial('/dev/ttyACM0', 9600)
143         camara.start_preview(alpha=128)
144         sleep(2)
145         camara.start_recording(analizador, 'yuv')
146         try:
147             while True:
148                 camara.wait_recording(1)
149         finally:
150             camara.stop_recording()

```

Figura 3-5 Código principal del microprocesador

A continuación, se emplea el gestor de contexto para la función de tratamiento de imágenes, que denominaremos “analizador”, y se inicializa el modo “*preview*”, que permite ver en streaming lo que está grabando la cámara, sin opción de tratamiento de imagen. Este modo se emplea únicamente para ver lo que ve la cámara a la vez que se emplea, sin opción a tratar las imágenes o codificarlas en un formato determinado y posteriormente guardarlas en un archivo. El parámetro “alpha” corresponde con el grado de opacidad de la presentación en pantalla, siendo 255 totalmente opaco, es decir que solo se ve el video, y 0 totalmente transparente, de forma que se aprecia ligeramente el “*preview*”, pero se ve sobre todo la presentación de la Raspberry. Usaremos este modo “*preview*” como herramienta a la hora de realizar las diferentes pruebas del sistema.

Por último, se deja dos segundos de margen para que la cámara se inicialice (recomendado por el fabricante) y se empieza a grabar. Cabe reseñar que, como explicaremos más adelante, se empleará un tratamiento de imagen basado en el formato YUV, con lo cual se utilizará este formato, al igual que el propio analizador, que se empleará como “*output*” del video (que comentaremos en el siguiente párrafo). El bucle “While True” permite grabar en bucle, finalizando con la orden de parar de grabar cuando se interrumpe el programa (gracias a la estructura “Try-Finally”).

Cabe reseñar que, para dar instrucciones al microcontrolador, emplearemos la comunicación serial. En efecto, le pasaremos al microcontrolador a través del puerto USB una serie de instrucciones para que realice los movimientos adecuados, en forma de letras mediante el código ASCII. Siendo así, para establecer dicha comunicación serial, habrá que indicar a la Raspberry qué puerto usar y el valor de la tasa de símbolos por segundos (en baudios) que habrá que usar, siendo en nuestro caso el puerto “/dev/ttyACM0”. Cabe reseñar que, en recepción, a parte de establecer una lógica acorde con las señales enviadas, habrá que establecer la misma tasa ya que sino el buffer del microcontrolador saturará y se romperá el enlace. Para aclarar el funcionamiento del código, el flujograma de la Figura 3-6 refleja el modo de proceder del microcontrolador. La función “analizador” se analizará en detalles en el apartado 3.1.3.

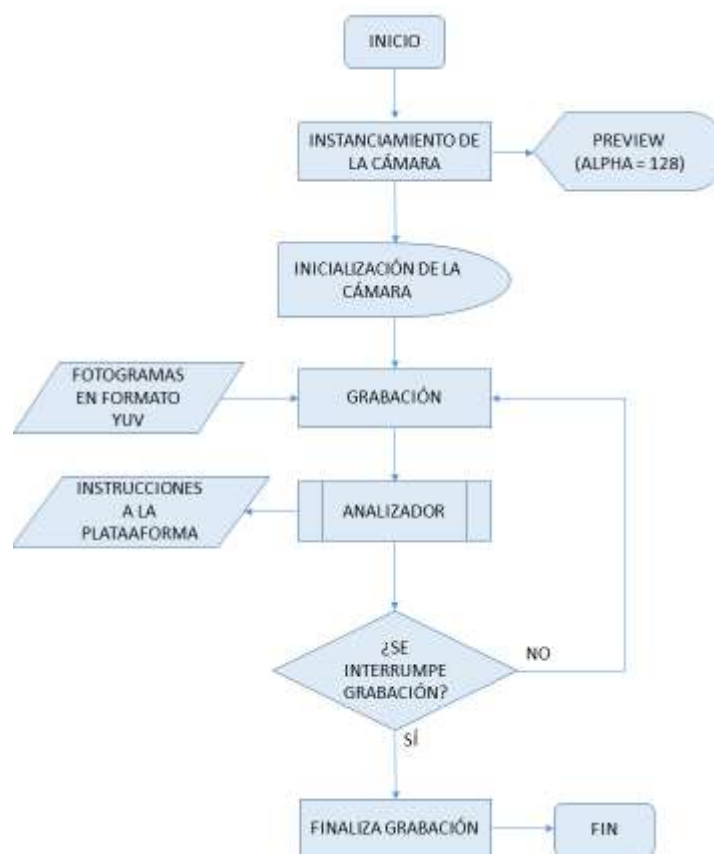


Figura 3-6 Diagrama de flujo del funcionamiento de la cámara

3.1.3 Tratamiento de imágenes

En cuanto al analizador de imágenes, este se corresponderá con la carga útil del código, ya que gracias a este podremos lograr el objetivo que es el reconocimiento de una baliza determinada. Primero hay que entender como procede la cámara tal y como la hemos configurado. Al inicializar la cámara, el destino final de estas puede ser tanto un archivo, en cuyo caso se escribirá una cadena de texto (string) al inicializar la grabación, como un objeto con una función “write”. Esta consiste únicamente en pasar los datos en una larga cadena, que se almacenan en un buffer, hasta que se transmite la totalidad de la imagen y se instancia el método “flush”, que transfiere todos los datos del buffer al objeto destino y lo libera. El analizador por lo tanto será dicho objeto con la función “write”.

Para analizar las imágenes, es necesario emplear otra librería, distinta a la propia de la cámara, que es la de PiYUVAnalysis. Esta librería permite únicamente pasar los datos del formato YUV a una matriz de características especiales, y necesita heredar los parámetros de la cámara para su funcionamiento, para lo cual se implementan en el método “__init__” del analizador, método que se ejecuta con prioridad cada vez que se instancia el analizador. La matriz que devuelve dicha librería corresponde con los valores Y (luminosidad), U y V (crominancia: color del píxel en una escala de dos coordenadas), de cada píxel de la cámara. Además, cabe reseñar que esta librería revierte los datos a un método del analizador que obligatoriamente ha de llamarse “analyze”. En efecto, el objeto analizador tendrá que tener obligatoriamente un método con tal nombre, ya que sino no funcionará el programa.

Sin embargo, es de suma importancia saber que el tipo de matriz no es el “array” convencional de Python, sino que los desarrolladores emplean un tipo específico denominado “numpy array”, que no es más que un tipo de array que se tratará con la librería específica NumPy (que se importa también como “numpy” y con el gestor de contexto “np”). Esta librería está específicamente desarrollada para

computación científica, y en concreto para operar con matrices de múltiples dimensiones, aunque también ofrece otras prestaciones como implementación con lenguaje C++ o Fortran u operaciones en el ámbito del álgebra lineal o la transformada de Fourier. En nuestro caso, este tipo de matrices tienen características y métodos que difieren de los “*arrays*” convencionales, lo cual nos aventaja en determinados momentos, pero nos dificulta mucho en otros, y que veremos más adelante. En concreto, la librería PiYUVAnalysis revierte una matriz de tres dimensiones, la primera correspondiendo al número de fila (coordenada Y), la segunda correspondiendo al número de columna (coordenada X) y la tercera correspondiendo al valor Y, U y V del píxel (esta tendrá por lo tanto tres posiciones posibles: 0 para el valor Y, 1 para el valor U y 2 para el valor V). Cabe destacar que, al igual que en el formato RGB, estos valores pueden ir del 0 al 256. Un ejemplo de dicha matriz es la de la Figura 3-7.

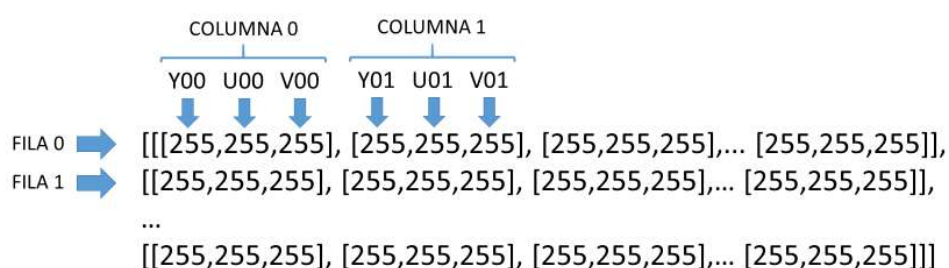


Figura 3-7 Ejemplo de matriz numpy YUV

El objetivo de nuestro sistema es que reconozca puntos luminosos, y por lo tanto tendremos en cuenta la luminosidad de los píxeles de la imagen para dicho reconocimiento. No se emplearán colores ya que estos se distorsionan bastante en función de los ajustes automáticos que realiza la cámara. Llegados a este punto, hay que tener en cuenta varios puntos para el diseño del código. Queremos que el sistema sea capaz de detectar puntos luminosos, pero estos no tienen por qué coincidir con un píxel. En otras palabras, un determinado punto luminoso puede corresponder con más de un píxel, y por lo tanto habrá que crear objetos en el código, que no son más que matrices de píxeles adyacentes que representan un mismo punto luminoso.

A mayores, y relativo sobre todo a la componente naval del sistema, debemos de tener en cuenta que una baliza luminosa que empleemos no tiene por qué coincidir con el punto más luminoso del fotograma. En el entorno marino existen multitud de señales luminosas que pueden ser más luminosas que la baliza de la estación receptora como pueden ser faros, luces de barcos (sobre todo barcos pesqueros) o las luces de las ciudades costeras, además de que en una plataforma naval no interesa emitir un foco de luz excesivamente luminoso debido a la indiscreción que eso supone. Por lo tanto, emplearemos una baliza luminosa que consista en un patrón determinado de luces, y desarrollaremos por lo tanto un código que permita diferenciar de entre una multitud de puntos luminosos, el centro del patrón empleado. Dicho patrón tiene que ser uno que sea suficientemente improbable de darse por sí solo, y por lo tanto los patrones en forma de línea o triángulo no valen. Se ha elegido por lo tanto el cuadrado, forma bastante difícil de darse por sí sola, pero relativamente sencilla de programar.

En cuanto al código del analizador, se implementará por lo tanto en un método que se denomine “analyze”, de forma que la interfaz de la cámara detecte el código de tratamiento de imagen. El código se representa en la Figura 3-8, junto con el flujograma correspondiente en la Figura 3-9.

```

77 class MiAnalizador (PiYUVAnalysis):
78     def __init__(self, camara):
79         super (MiAnalizador, self). __init__ (camara)
80
81     def analyze (self, a):
82         M = []
83         Xm = []
84         Ym = []
85         umbral = 180
86         Y = a[:, :, 0]
87         XRES = 193
88         YRES = 90
89         print("inicio")
90         A = np.argwhere(Y >= umbral)
91         if A.shape != (0, 2) :
92             M.append(np.array(A[0]).reshape(1, 2))
93             A = np.delete(A, 0, 0)
94             while A.shape != (0, 2):
95                 contador = 0
96                 for i, obj in enumerate(M):
97                     obj = np.array(obj)
98                     elem = A[0]
99                     elem = np.array(elem)
100                     [x, y] = CompararValor(obj, elem)
101                     if x.shape != (0,) and y.shape != (0,):
102                         M[i] = AnhadirMatriz(M[i], elem)
103                         A = np.delete(A, 0, 0)
104                         contador += 1
105                     break
106                 if contador == 0:
107                     M.append(np.array(A[0]).reshape(1, 2))
108                     A = np.delete(A, 0, 0)
109             for i, obj in enumerate(M):
110                 Xmed = int(np.mean(obj[:, 1]))
111                 Ymed = int(np.mean(obj[:, 0]))
112                 Xm.append(Xmed)
113                 Ym.append(Ymed)
114             if len(Xm) != 0:
115                 B = PatronCuadrado(Xm, Ym)
116                 if B != None:
117                     [X, Y] = B
118                     print(X, ' ', Y)
119                     if X < XRES/2-1 and Y > YRES/2:
120                         port.write(("q".encode()))
121                     elif XRES/2-1 <= X <= XRES and Y > YRES/2:
122                         port.write(("w".encode()))
123                     elif X > XRES/2 and Y > YRES/2:
124                         port.write(("e".encode()))
125                     elif X > XRES/2 and YRES/2-1 <= Y <= YRES/2:
126                         port.write(("d".encode()))
127                     elif X > XRES/2 and Y < YRES/2-1:
128                         port.write(("c".encode()))
129                     elif XRES/2-1 <= X <= XRES/2 and Y < YRES/2-1:
130                         port.write(("x".encode()))
131                     elif X < XRES/2-1 and Y < YRES/2-1:
132                         port.write(("z".encode()))
133                     elif X < XRES/2 and YRES/2-1 <= Y <= YRES/2:
134                         port.write(("a".encode()))

```

Figura 3-8 Código del analizador

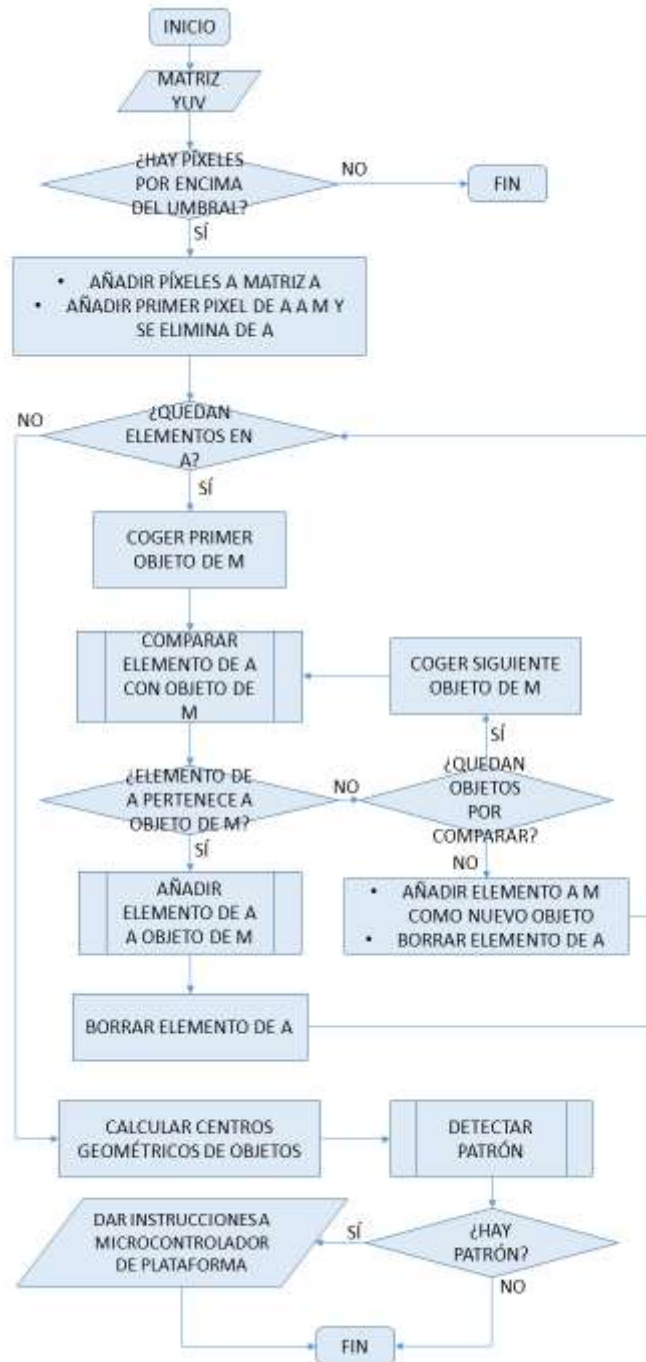


Figura 3-9 Diagrama de flujo del analizador de imágenes

En el código, una de las ventajas que nos permite una matriz numpy es que se pueden acceder a valores específicos en cada una de las dimensiones, mientras que en los *arrays* convencionales solo se puede acceder a elementos de la primera dimensión. En otras palabras, si la matriz de la Figura 3-7 fuese un *array* convencional, este sería solo un *array* de *arrays*, y por lo tanto solo podríamos acceder a una fila en concreto. Con NumPy, podemos escoger el valor que en este caso nos interesa, que son todos los valores en posición 0 de la última dimensión (correspondiente con el valor Y), quedándonos una matriz con el mismo número de dimensiones, pero con solo el valor Y en cada una de ellas. Otra de las ventajas que aporta, que se explota a continuación en el código, es que permite, mediante el método “*argwhere*”, encontrar valores en la matriz que cumplan un requisito, devolviendo la posición de estos en la matriz. Siendo así, el primer paso que tomaremos será determinar un umbral luminoso a

partir del cual tendremos en cuenta esos píxeles como puntos luminosos. En el punto 4.1.3, veremos que umbral pondremos en el código. Cabe reseñar que dicha función devuelve un *numpy array* de dos dimensiones, la primera siendo los pares de coordenadas de todos los píxeles cuya luminosidad supere el umbral, y la segunda correspondiendo con la coordenada Y o X (en ese orden) de dichos pares. A esta matriz la denominaremos “A”, y a los diferentes pares de coordenadas le denominaremos elementos.

Una vez tenemos las posiciones de los píxeles luminosos, tenemos que agrupar estos en grupos de píxeles que representan al punto luminoso, a los que llamaremos objetos. Como se ha mencionado, un punto luminoso normalmente se corresponde con un grupo de píxeles adyacentes que superan el umbral, y por lo tanto hay que juntarlos. Para ello, se empleará una matriz M en la que guardaremos las coordenadas con las mismas dimensiones que devuelve el método “argwhere”, pero añadiendo por lo tanto una tercera dimensión, que se corresponderá con cada objeto. Siendo así, tendremos en la primera dimensión diferentes matrices de pares de coordenadas que se corresponderán con los objetos, en la segunda tendremos los propios pares y en la tercera la coordenada Y o X de cada par, como se puede ver en la Figura 3-10.

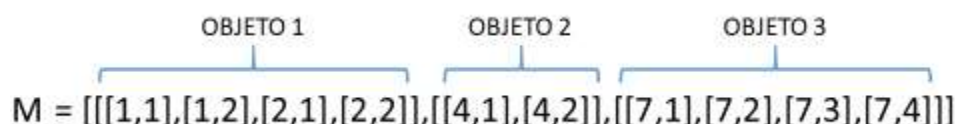


Figura 3-10 Ejemplo de matriz de objetos de puntos luminosos

Esta tarea de ordenar los diferentes pares de coordenadas de píxeles luminosos en objetos es la que supone el mayor reto de todo el programa y para la que habrá que hacer uso de dos funciones específicas que veremos más adelante. Como se ha planteado el código para cumplir con este objetivo es, en rasgos generales, que para cada elemento de A se compruebe si este corresponde con algún objeto de M, es decir si hay algún elemento de algún objeto adyacente al elemento de A en cuestión. Si es el caso, se añade a dicho elemento, y sino, al ser un punto a tener en cuenta (por ser luminoso), se añade como un objeto nuevo.

En el código podemos ver dos bucles: el bucle “while” y el bucle “for”, este último estando dentro del primero. Lo que se pretende es ir pasando elementos de A a M, trillándolos ya sea como objetos nuevos o como parte de otros objetos. Para eso, iremos borrando los elementos de A procesados, de forma que cuando se procese toda la matriz A, se salga del bucle “while”. Dentro de este bucle, mediante un bucle “for”, analizaremos si el elemento en cuestión puede asimilarse a un objeto mediante la función CompararValor que explicaremos más adelante, en cuyo caso se añadirá a este mediante otra función: AnhadirMatriz (que también se explicará más adelante). Si no, se añade a la matriz M simplemente mediante el método “.append”. Cabe reseñar que se emplea un buque “while” y un bucle “for” ya que, si usásemos dos bucles “for” como se podría pensar en primer lugar, los elementos que vayamos añadiendo a los objetos no se tendrán en cuenta, además de complicar el uso de los índices ya que a medida que eliminamos elementos de A, los índices se corren.

Una vez procesados todos los elementos de A, se procede a calcular los centros geométricos de los objetos, que no son más que la media de las coordenadas en cada eje. Estas coordenadas representarán a cada objeto a la hora de reconocer un patrón. Para ello, se empleará una tercera función, denominada PatronCuadrado, que comentaremos al final de este apartado, pero que en resumidas cuentas determina si hay un patrón cuadrado de luces. Si este existe, devolverá como parámetros las coordenadas del centro de este, sino, no devolverá nada. Por último, en caso de reconocimiento de un patrón y teniendo en cuenta la resolución de la pantalla para determinar que columna y que fila de píxeles están a la

mitad de la pantalla, se manda a través de la comunicación serial al microcontrolador una orden en función del cuadrante donde esté el centro del cuadrado, actuando este de forma que centre el cuadrado en el centro de la cámara. En cuanto al código del analizador, solo queda decir que se ejecuta de nuevo con cada fotograma de la cámara, con lo que no hace falta inicializar todas las variables a cero.

La función CompararValor, cuyo código viene reflejado en la Figura 3-11, devuelve algún valor, tanto de coordenadas en Y como en X, si se da el caso de que un elemento (que se dará como *input* a la función) corresponda con un objeto de la matriz M. La lógica que se aplicará en cuanto a pertenencia a un objeto es que el elemento en cuestión sea adyacente a este, o en otras palabras que el elemento tenga alguna coordenada que sea mayor o menor que algún elemento de un objeto de uno, aunque se tendrá en cuenta una tolerancia. Para ello, usaremos la función que hemos empleado anteriormente para buscar elementos en un matriz numpy: el método “.argwhere”. El inconveniente es que en este caso, queremos implementar una búsqueda de tal forma que verifique que x' e y' cumplen las ecuaciones de (16).

$$x - tol \leq x' \leq x + tol, \quad y - tol \leq y' \leq y + tol \quad (16)$$

Donde x e y son las coordenadas de cada elemento del objeto, x' e y' son las coordenadas del elemento en proceso de clasificación y tol es la tolerancia. Una fácil implementación de estas ecuaciones sería posible con un operador lógico AND, pero un inconveniente que presenta en este caso Numpy es que en el método “.argwhere” no se pueden usar dichos operadores, pudiendo haber solo un argumento. Es por ello que se implementa el bucle “for” seguido de un “if”, equivalente a la otra parte de la ecuación. Cabe reseñar que se coge los valores de las matrices resultantes de la búsqueda (matrices x e y en el código) a partir de la segunda posición ya que la primera corresponde con un cero, empleado únicamente para inicializar las variables como matrices numpy. Además, la tolerancia es necesaria ya que la cámara sufre diversos efectos a la hora de procesar las imágenes (*blooming*, perturbación de la lente,...), sobre todo para las pruebas en el laboratorio, que son a poca distancia. Por lo tanto, se empleará una tolerancia de 1 en general (es decir solo se tomarán píxeles adyacentes), pero pudiendo aumentarse para la experimentación.

```

17 def CompararValor(obj,elem):
18     tol = 3
19     x = np.zeros((1)).astype(int)
20     x1 = np.argwhere(obj[:,1] <= elem[1]+tol)
21     for x2 in x1:
22         if obj[x2,1] >= elem[1]-tol:
23             x = np.append(x,x2,axis = 0).astype(int)
24     x = x[1:]
25     y = np.zeros((1))
26     y1 = np.argwhere(obj[:,0] <= elem[0]+tol)
27     for y2 in y1:
28         if obj[y2,0] >= elem[0]-tol:
29             y = np.append(y,y2,axis = 0)
30     y = y[1:].astype(int)
31     return([x,y])

```

Figura 3-11 Código de la función CompararValor

La segunda función del código es la más corta de las tres, como podemos ver en la Figura 3-12, pero se basa en un principio complejo relacionado con la librería NumPy. La función AnhadirMatriz se emplea para añadir a un objeto ya existente, un nuevo elemento. Esto requeriría de un diseño nimio si se tratase de una matriz convencional, ya que se usaría simplemente el método “.append”, pero en las matrices numpy, el método “.append” equivalente tiene muchas limitantes, sobre todo respecto a las

dimensiones. Cuando queremos añadir algo a una matriz numpy, si esta es de una sola dimensión, así como lo que queremos añadir, no hay ningún inconveniente.

Sin embargo, si alguno de los dos es de dos dimensiones, la librería automáticamente aplica otro método, el “`.flatten`”, que devuelve una matriz de una sola dimensión con los valores de la matriz que queríamos añadir a continuación de los valores ya existentes en la otra matriz. Sin embargo, si queremos añadir una matriz de dos o más dimensiones a otra también de dos o más dimensiones, NumPy solo permite la operación en caso de que las dimensiones sean las mismas, y el número de elementos dentro de estas sea el mismo. Por ejemplo, en nuestro caso, si queremos añadir una matriz a un objeto de M, esta tendrá que ser también de dos dimensiones, y además con el mismo número de elementos.

Sabiendo esto, la función `AnhadirMatriz` tiene en cuenta este limitante y procede en concordancia. En primera instancia, las entradas de la función son las mismas que las de la función del párrafo anterior, es decir el objeto al que se debe añadir el elemento, y el propio elemento a añadir. Para realizar la fusión, tenemos que crear una matriz de mismas dimensiones y número de elementos que el objeto. Para ello se procede de la siguiente manera, se crea una matriz de ceros de una sola dimensión, en la que habrá tantos ceros como número de elementos en el objeto menos dos, ya que hay que añadir el elemento a añadir. A continuación, se añade el elemento, que tiene dos dimensiones, y como se ha mencionado, la matriz resultante tiene los ceros iniciales más los dos valores del elemento a añadir, todo en una dimensión.

Utilizando el método “`.reshape`”, se crea una matriz de dos dimensiones en las que vuelve a recuperar la forma que tiene el objeto. Siendo así, tenemos dos matrices de mismas dimensiones (dos) y con el mismo número de elementos, con lo cual las añadimos sin problemas. Por último, para librarse de los ceros, emplearemos una técnica de Python llamada máscara. Al emplear el método “`.any`”, la librería devuelve una matriz con valores booleanos en función del argumento que hayamos puesto, así como la dimensión. En nuestro caso, además, aprovechamos que la librería NumPy asume una matriz de números iguales (matriz de ceros, de unos,...) como el propio número cuando empleamos el método “`.any`” en una dimensión de matrices. En otras palabras, el método “`.any`”, cuando lo empleamos en la dimensión 1 (que es la dimensión de los pares de coordenadas, es decir la segunda dimensión), analiza las diferentes matrices y, en nuestro caso, analiza las que tienen en su interior todo ceros, interpretando el cero del argumento como una matriz de ceros de misma magnitud que las matrices analizadas (aunque también se podría poner por ejemplo un valor concreto `[a,b]`). Siendo así, devuelve para cada elemento de toda la matriz un valor “`True`” si hay coordenadas, y `False` si el elemento es `[0,0]`. Esta matriz de booleanos es por la tanto la máscara, que, al aplicarla a la matriz de origen, nos dará una nueva matriz en la que los elementos compuestos únicamente por ceros se verán enmascarados. Es decir, la máscara hace que los elementos coincidentes con un “`False`” se anulen, devolviendo solo los que corresponden con un valor “`True`”. De este modo conseguimos la matriz que nos interesa, con el objeto inicial de M y el elemento añadido en este.

```

8  def AnhadirMatriz(obj,elem):
9      a = np.zeros((len(obj)*2-2)).astype(int)
10     elem = np.append(a,elem).reshape((len(obj),2))
11     obj = np.append(obj,elem,axis=0)
12     n = obj
13     inds = np.any(n != 0, axis=1)
14     n_zero = n[inds, :]
15     return (n_zero)

```

Figura 3-12 Código de función `AnhadirMatriz`

Por último, la última función se encargará de reconocer un patrón de puntos luminosos. El patrón elegido para nuestro sistema será un cuadrado, ya que resulta complicado encontrar de por sí una disposición parecida en el ámbito marino, además de ser sencillo de implementar. En el ámbito marino podemos encontrar multitud de patrones recurrentes, como pueden ser una línea de puntos luminosos, que pueden fácilmente darse en la costa, con las luces de los pueblos costeros (farolas, comercios,...). Así mismo, tampoco es raro encontrar patrones alineados verticales, sobre todo en ciudades o puertos con estructuras altas e iluminadas, y por lo tanto la combinación de estas, equivalente a una cruz, también podría ser relativamente probable de ocurrir. Sin embargo, formas geométricas más complicadas son más difíciles de encontrar, como puede ser un triángulo equilátero o un cuadrado.

Se ha elegido el cuadrado como patrón, aunque sería objeto de estudio determinar un patrón luminoso de reducida probabilidad de falsa alarma en el ámbito marino, teniendo en cuenta los propios buques, la luminosidad de la costa o el reflejo del sol en la mar. A parte, consideraremos que tanto estación emisora como receptora poseen de un sistema de estabilización adecuado, y que por lo tanto ni la cámara ni la baliza están girados, respecto a la horizontal ni a la vertical, estando la baliza perfectamente alineada con la vertical al suelo, así como la cámara. Siendo así, la función recibe dos matrices que contiene, para cada objeto, la media de sus coordenadas en cada eje. La matriz X_m tiene por lo tanto la media de todas las coordenadas en el eje de las abscisas para cada objeto, correspondiendo por lo tanto un valor a un objeto, y de mismo modo para la matriz Y_m . Cabe reseñar que en este caso las dos matrices son *arrays* convencionales de Python, y por lo tanto se emplearán los métodos clásicos de estas para operar.

En cuanto al modo de proceder del código en sí, plasmado en la Figura 3-13, este se divide en cuatro partes. La primera parte, que se corresponde con el doble bucle “for”, añade a una matriz X las coordenadas de los píxeles cuya coordenada X se repita, y sus correspondientes coordenadas Y en una matriz denominada Y . El objetivo es tener en cuenta solo estos píxeles ya que como hemos mencionado, la baliza y la estación emisora están estabilizados. Aun así, se tendrá en cuenta una tolerancia ya que el patrón se ve perturbado por efectos de la cámara a la hora de captura de imágenes, por la propia construcción de este (que no es perfecta) y por la orientación de este. Además, el condicional “if” permite no sobrescribir los datos en las matrices X e Y . La segunda parte es la de purgar los píxeles cuya coordenada Y no se repite, debido a lo expuesto previamente. Para ello se utilizará una matriz auxiliar en la que se guardarán los índices de dichos píxeles, para luego borrar los valores correspondientes a dichos píxeles. Se procede en dos bucles distintos ya que, si borrásemos directamente los píxeles de Y en el primer bucle, habría valores que no se tendrían en cuenta por el hecho de que se van borrando valores de la matriz y el bucle “for” opera mediante índices.

En tercer lugar, para detectar un cuadrado, buscaremos una diagonal de este. Para ello, mediante los dos bucles “for”, buscaremos si existe un píxel cuya diferencia de abscisas sea igual a la de ordenadas, con cierta tolerancia. Por último, el código verifica que, en el caso de haber encontrado una posible diagonal, existen los otros dos vértices (de nuevo, con cierta tolerancia). Si es el caso, devuelve el punto medio de la diagonal, que tomaremos como centro de gravedad del cuadrado (aunque este no sea una figura geométrica perfecta en la mayoría de los casos). Cabe destacar que este reconocimiento devuelve unas coordenadas si detecta un patrón, pero en caso de haber más de uno, devuelve el primero que encuentre.

```

33 def PatronCuadrado (Xm,Ym):
34     X = []
35     Y = []
36     tol = 5
37     I = []
38     for i, x1 in enumerate(Xm):
39         for j, x2 in enumerate(Xm):
40             if x1-tol <= x2 <= x1+tol:
41                 if X.count(x2) != Xm.count(x2):
42                     X.append(x2)
43                     Y.append(Ym[j])
44     for k, y1 in enumerate(Y):
45         contador1 = 0
46         for l, y2 in enumerate(Y):
47             if y1-tol <= y2 <= y1+tol:
48                 contador1 += 1
49         if contador1 == 0:
50             I.append(k)
51     contador2 = 0
52     for l in I:
53         X.pop(l-contador2)
54         Y.pop(l-contador2)
55         contador2 += 1
56     if len(Y) > 1:
57         for m, n in enumerate(X):
58             for o, p in enumerate(X):
59                 if p > n:
60                     difx = p-n
61                     dify = abs(Y[o]-Y[m])
62                     if dify-tol <= difx <= dify+tol:
63                         contador3 = 0
64                         for q, r in enumerate(X):
65                             if n-tol <= r <= n+tol:
66                                 if Y[o]-tol <= Y[q] <= Y[o]+tol:
67                                     contador3 += 1
68                             if p-tol <= r <= p+tol:
69                                 if Y[m]-tol <= Y[q] <= Y[m]+tol:
70                                     contador3 += 1
71                         if contador3 >= 2:
72                             XMED = (n+p)/2
73                             YMED = (Y[m]+Y[o])/2
74                             return(XMED, YMED)

```

Figura 3-13 Código de función PatronCuadrado

Por último, para transmitir la señal se empleará la librería de Raspberry para emplear el puerto Serial, como se puede ver en la Figura 3-14. Dicho puerto se activará al principio del funcionamiento, como se puede ver en la Figura 3-5, y las órdenes se transmitirán en forma de letras. En primer lugar, se ha dividido el FOV (*Field Of View*) de la cámara en ocho sectores de forma que coinciden con las letras alrededor de la letra “s” en un teclado español. Siendo así, las letras “a”, “w”, “d” y “x” representaran giros en un solo eje, es decir o un giro en azimut o en elevación, ya que estas corresponden con los sectores cardinales desde el centro de la cámara, y las letras “q”, “e”, “c” y “z” representarán un giro en los dos ejes, ya que corresponde con los cuadrantes restantes. Como se verá en las pruebas de calibración mencionadas en el apartado 4.2.1, las variables XRES e YRES representan lo que la cámara deberá de considerar como centro para las órdenes y se ajustarán mediante una calibración previa del conjunto del sistema. Cabe reseñar que la comunicación serial emplea el código UTF-8 (*8-bit Unicode Transformation Format*) para el envío de caracteres, y por ello es necesario emplear el método “.encode()”.

```

114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136

if len(Xm) != 0:
    B = PatronCuadrado(Xm,Ym)
    if B != None:
        [X,Y] = B
        print(X,' ',Y)
        if X < XRES/2-1 and Y > YRES/2:
            port.write(("q".encode()))
        elif XRES/2-1 <= X <= XRES and Y > YRES/2:
            port.write(("w".encode()))
        elif X > XRES/2 and Y > YRES/2:
            port.write(("e".encode()))
        elif X > XRES/2 and YRES/2-1 <= Y <= YRES/2:
            port.write(("d".encode()))
        elif X > XRES/2 and Y < YRES/2-1:
            port.write(("c".encode()))
        elif XRES/2-1 <= X <= XRES/2 and Y < YRES/2-1:
            port.write(("x".encode()))
        elif X < XRES/2-1 and Y < YRES/2-1:
            port.write(("z".encode()))
        elif X < XRES/2 and YRES/2-1 <= Y <= YRES/2:
            port.write(("a".encode()))

```

Figura 3-14 Código para la transmisión de órdenes por el puerto serial

3.2 Plataforma del sistema

3.2.1 Material

Para el desarrollo de la plataforma de este sistema FSOC se pretende construir un sistema cardan que permita el giro tanto en azimuth como elevación para permitir el apuntamiento entre estación emisora y estación receptora. Se ha elegido dicho sistema tanto por su componente mecánica como por su sencillez, robustez y posible desarrollo con materiales genéricos de un laboratorio de electromecánica.

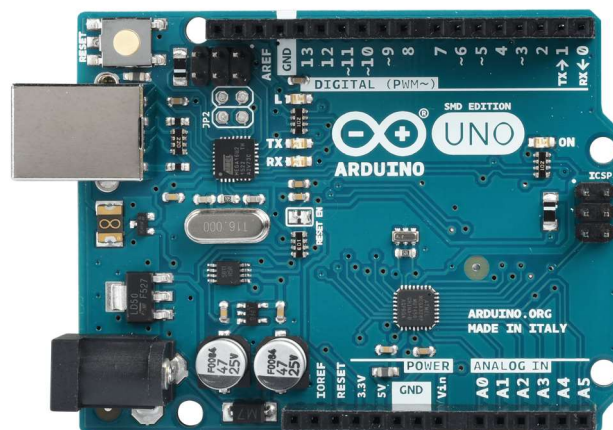


Figura 3-15 Fotografía del Arduino Uno [59]

Así como todo el sistema depende de la Raspberry, todo el accionamiento de la plataforma se hará a través de un microcontrolador Arduino Uno, correspondiente con la Figura 3-15. Este tipo de microcontroladores ofrecen varias ventajas respecto a la Raspberry Pi en cuanto a proyectos de electrónica y es que aparte de funcionar a voltajes más altos (5 V frente a 3,3 V para la Raspberry Pi), los Arduinos son francamente más robustos (impedancias mucho más grandes en los pines, esquemas Pull-up integrados en los pines,...) y disponen de ADC (Analog-to-Digital Converter) [59].

Las diferencias principales entre un microcontrolador, como Arduino, y un microprocesador, como la Raspberry Pi, residen en primer lugar en la frecuencia de procesamiento. El orden de magnitud de la frecuencia de procesamiento de los microprocesadores son los GHz (como el caso de la

Raspberry Pi empleada que es de 1,2 GHz) mientras que el de los microcontroladores es de decenas de MHz (como es el caso del Arduino Uno, que es de 16 MHz) [58] [59]. Otra diferencia fundamental es que un microcontrolador se puede asimilar a un circuito integrado, en tanto que ofrece la ventaja de aplicar una lógica determinada a un circuito, necesitando solo de implantar esta lógica mediante la IDE de Arduino y de proporcionar una fuente de alimentación. Por otro lado, la complejidad de un microcontrolador requiere en primer lugar de un sistema operativo que permita gestionar todos los componentes de este, y a continuación de multitud de instrucciones para su uso efectivo, que requiere además en muchas ocasiones de periféricos como un ratón, un teclado y un monitor o de una conexión en remoto.

El modelo de microcontrolador que se empleará en este proyecto será el Arduino Uno, que consta de 14 pines digitales tanto como para medir como para transmitir (se configuran o en un modo o en otro). Estos pines siguen la lógica TTL (Transistor-Transistor Logic), en la que se asimilar un “0” para un valor entre 0 V y 0,8 V, y un “1” entre 2,2 V y 5 V. Además, 6 de estos pines constan de una modulación PWM para ser capaces de recrear una salida analógica. Mediante la emisión de pulsos de mayor o menor ancho, se puede conseguir una tensión media que varíe y por lo tanto poder emitir una señal que recree una señal analógica, como se puede apreciar en la Figura 3-16. En el caso de Arduino, se pueden emitir 256 valores diferentes de tensión mediante los pines PWM, entre 0 V y 5 V [60].

Además, disponen de 5 pines para medir valores analógicos. Estos pines solo permiten medir valores (no permiten emitir) pero tienen mayor resolución, ya que el ADC divide el rango de funcionamiento del Arduino (0-5 V) en 1024 valores de unos 5 mV de diferencia entre estos (más precisamente 4,88 mV). Por último, ofrecen, al igual que los pines de la Raspberry Pi, unos pines para alimentación, que se componen de pines para dar tensión (uno a 5 V y otro a 3,3 V) y otros dos de masa [59].

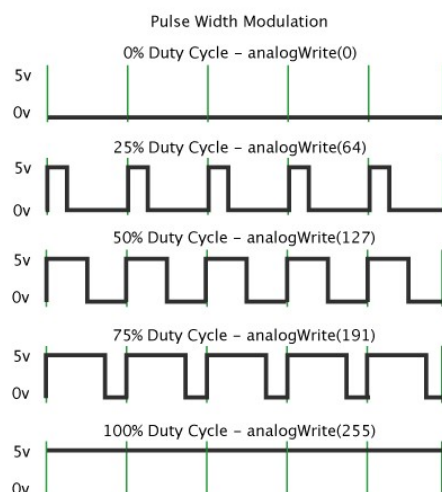
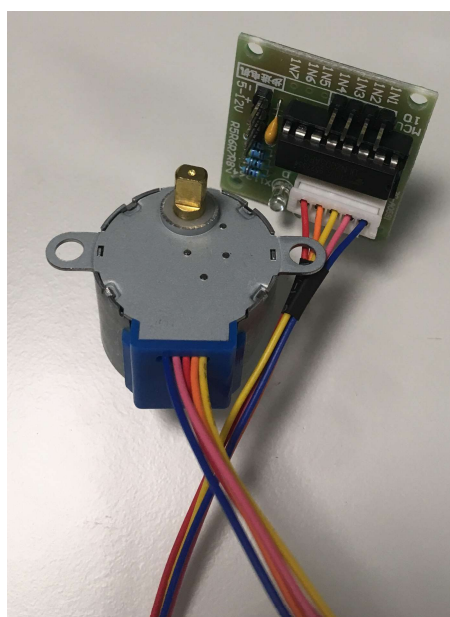


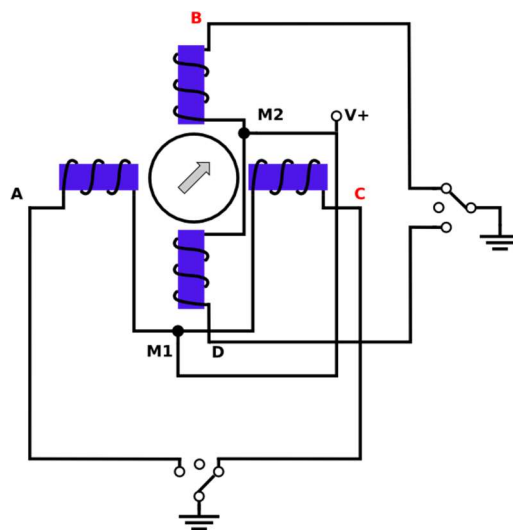
Figura 3-16 Diagrama de modulación PWM en Arduino [60]

Por otro lado, se emplearán dos motores paso a paso 28BYJ-48, como el de la Figura 3-17 (a). Este tipo de motores tienen un rotor está polarizado magnéticamente y rodeado de bobinas. Al inducir un voltaje en una bobina, esta atrae al motor, avanzando este último por “pasos”. Los motores paso a paso 28BYJ-48 son unipolares, lo cual quiere decir que consta de cuatro bobinas divididas en dos pares cuyos bornes positivos son compartidos (cable rojo), como se muestra en la Figura 3-17 (b). Siendo así, las bobinas quedan enfrentadas, creando campos opuestos al excitarse (pero no se excitan a la vez).

Esta disposición permite que, para cambiar el sentido de giro del rotor, no haga falta cambiar el voltaje de alimentación del motor. Estos motores ofrecen ventajas muy interesantes ya que permiten controlar con precisión el movimiento de un rotor, además de ofrecer un par de retención. Cuando el motor está parado, la bobina correspondiente se mantiene excitada, actuando como un freno. Estos motores permiten dar también medios pasos, lo cual simplemente consiste en que, si una bobina está excitada y por lo tanto atrae al polo correspondiente del rotor, se excita la siguiente bobina, creando un campo tal que el rotor encuentra la posición de equilibrio entre una bobina y la otra. El rotor da por lo tanto medio paso, lo que permite mayor precisión. En concreto, los motores 28BYJ-48 operan con una tensión nominal de alimentación de entre 5 V y 12 V, aunque solo requieren de una intensidad de 55 mA. Ofrecen un par de 36 N.m y tienen una reducción mecánica de relación 1/64. Si a esto le sumamos que el motor tiene 32 pasos por vuelta, obtenemos que el eje del motor da 2048 pasos por vuelta [61].



(a)



(b)

Figura 3-17 (a) Fotografía de un motor 28BYJ-48 enchufado a un driver ULN2003A, (b) Circuito del bobinado interno del motor 28BYJ-48 [61]

Para emplear estos motores, se conectan a un driver ULN2003A que consta de 7 pines de entrada y 7 de salida, como se puede ver en la Figura 3-18 (a). Este driver permite operar en condiciones de mayor potencia, aguantando hasta los 500 mA con un voltaje máximo de 50 V. El driver se compone de puertas lógica NOT que consisten en un esquema de transistores Darlington NPN, como se puede ver en la Figura 3-18 (b). Cuando no se aplica un voltaje a la base del primer transistor, es decir al pin de entrada, los transistores actúan como diodos en polarización inversa y no dejan pasar la corriente por el colector. A la salida, el pin común está al voltaje máximo del circuito de salida y el pin de salida está conectado al borne negativo del circuito de salida, de forma que la tensión que cae en el circuito hace que haya menos tensión en el pin de salida que en el pin común, y por lo tanto el circuito quede abierto ya que el diodo conectado al pin común se encuentra en polarización inversa.

En relación al circuito esquemático representado en la Figura 3-18 (c), al aplicar una tensión a la puerta (un 1), la puerta emite un 0, que no es más que el nivel de potencial nulo, o masa. Siendo así, existe una diferencia de potencial entre el pin común y el pin de salida (que está conectado a masa) y por lo tanto la corriente circula. Como en realidad funciona es que el potencial aplicado en la base de los transistores los “abre”, permitiendo el paso de la corriente. El emisor de estos está conectado a

masa, lo cual permite que la corriente circule. El montaje de los transistores similiar a un esquema de Darlington permite por lo tanto que circule una corriente de gran intensidad a través de la puerta hasta la masa. Cabe reseñar que el diodo que conecta el pin de salida y el común permite que en este caso no se cortocircuite el circuito, permitiendo así que la corriente vaya del pin común al circuito, luego al pin de salida y por último a masa a través de los transistores.

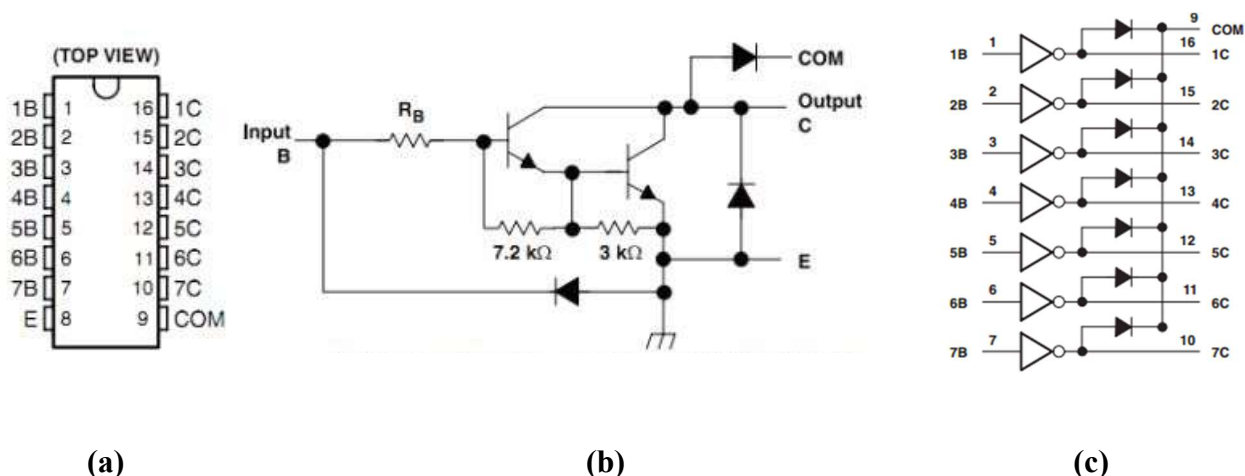


Figura 3-18 (a) Vista en planta del driver ULN2003A [61], (b) Circuito para un par de pines (salida y entrada) [61], (c) Circuito esquemático del driver ULN2003A [61]

En tercer lugar, usaremos un Joystick de la casa Arduino para el control inicial de la plataforma. Dicho Joystick consiste únicamente en dos potenciómetros a los que se les aplica 5 V de tensión y ofrecen una resistencia variable en función de la posición del Joystick, como se puede ver en la Figura 3-19. Un potenciómetro mide el ángulo desplazado en un plano ortogonal a la base, y el otro mide el ángulo desplazado en un plano ortogonal a la base y al plano anterior. Para ello, se emplean los pines analógicos del Arduino, correspondiendo la posición inicial del Joystick con la mitad del voltaje aplicado en los dos potenciómetros, lo que da una lectura en el Arduino de valor 512. Siendo así, las lecturas de los potenciómetros varían de 0 a 1024, que corresponden con los límites del Joystick.

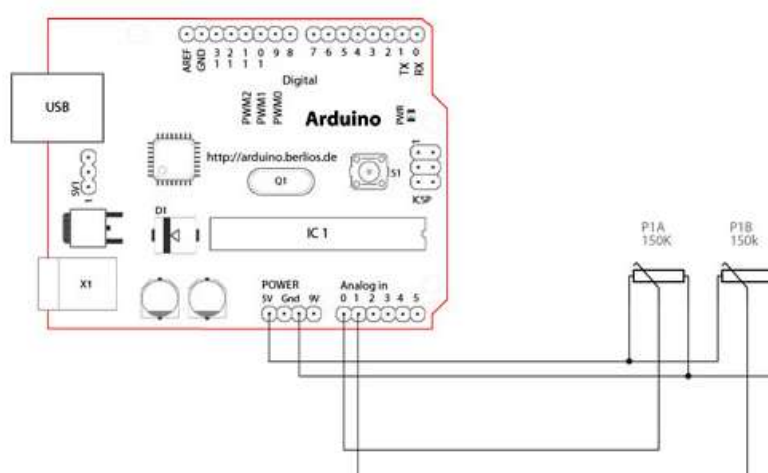


Figura 3-19 Representación del circuito de un Joystick acoplado a un Arduino [65]

Por último, se empleará para el desarrollo de la plataforma cables para interconectar los diferentes elementos, así como algunas piezas que se comentarán en el apartado 3.2.2 *Estructura de la plataforma*.

3.2.2 Estructura de la plataforma

El diseño de la plataforma de la estructura empieza teniendo como principal necesidad la capacidad de poder ajustar la dirección del módulo de comunicación tanto en azimut como en demora. Para ello, emplearemos dos motores paso a paso 28BYJ-48, uno para cada movimiento, y mediante un sistema de engranajes, transmitiremos este movimiento a la plataforma. La plataforma en sí consta de un total de ocho piezas, como se puede ver en la Figura 3-20, que se diseñarán específicamente para la plataforma y se imprimirán mediante impresoras 3D que son: una base estática, una base móvil, una carcasa para el módulo de comunicación, dos ejes, uno a cada lado de la carcasa, que se unirán a esta en el ensamblado final, y tres ruedas dentadas, diseñadas para encastrar en el eje del motor paso a paso. La tercera irá encastrada en el extremo de uno de los ejes (el más largo), que se ha diseñado para tener la misma forma del eje del motor, para permitir el giro de la carcasa en elevación. A esto le sumaremos tres piezas que se adquirirán ya hechas, que son dos rodamientos radiales, uno para cada eje, y uno axial para la plataforma móvil.

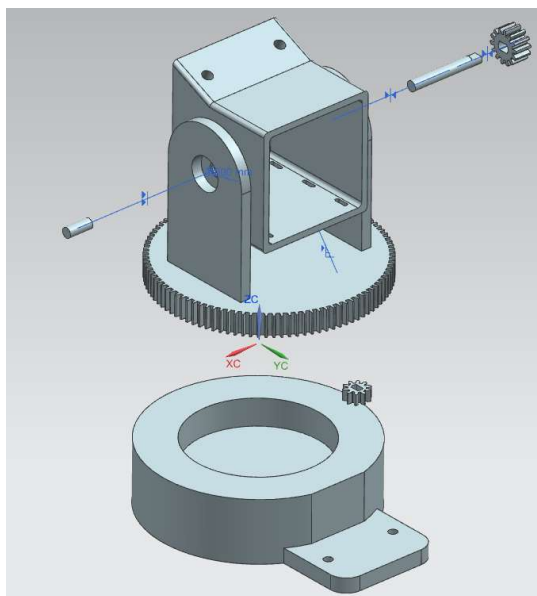


Figura 3-20 Modelo del ensamblado de la plataforma

La base estática consiste únicamente en un pedestal para toda la plataforma. Cabe reseñar dos partes en concreto, que son la lengüeta en la parte frontal, y la hendidura en el centro de la pieza. La lengüeta sirve como soporte para el motor paso a paso que se encargará de la dirección en azimut de la plataforma. Este irá con la parte opuesta al eje en contacto con la lengüeta, alineando los agujeros de las alas del motor con los de la lengüeta (de mismas dimensiones: 5 mm).

Para unirlos, se empleará una unión mecánica consistente en un tornillo con rosca métrica M4 (según ISO 7046) de 60 mm de longitud y una rosca métrica M4. El tornillo irá enroscado con la cabeza en la parte posterior de la lengüeta, quedando la tuerca apretada con el ala del motor. Se ha elegido esta disposición para permitir apoyar la base en una superficie plana y que la base apoye en su totalidad en esta (para lo que se tendrán que apretar los tornillos concienzudamente).

La hendidura central tiene las dimensiones exactas del rodamiento axial que se empleará para soportar el peso del resto de la plataforma, siendo este un rodamiento de bolas de la casa NSK de diámetro exterior de 65 mm y de diámetro interior de 45 mm, con una anchura de 14 mm y fabricado en acero, como podemos ver en la Figura 3-21.

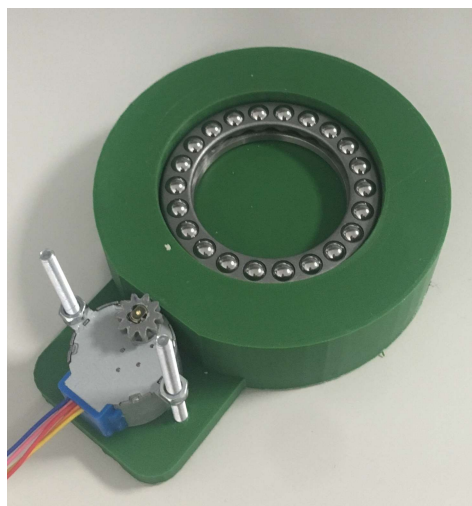


Figura 3-21 Base estática de la plataforma

La base móvil, que podemos ver representada en la Figura 3-22, es la pieza más compleja de la plataforma, ya que a parte de estar diseñada de tal forma que pueda sostener el módulo de comunicación y permita su orientación en elevación, sostiene uno de los motores en una de las aletas, además de que la base propiamente dicha sea en sí una rueda dentada. En concreto, la pieza en sí consta de dos aletas cuya función es la de ejercer de pilares del módulo de comunicación, transmitiendo el peso hacia la base y posteriormente al rodamiento axial. Una de las aletas tiene dos agujeros pasantes para la fijación del segundo motor paso a paso. Este irá de manera similar al otro motor en la base estática, de forma que la cabeza del tornillo con rosca métrica M4 vaya en la cara interior de la aleta (cara que enfrenta la otra aleta), y la tuerca, de mayor espesor, vaya en la aleta del motor, quedando el eje de este apuntando hacia el exterior de la base. Así conseguimos que la tuerca no roce contra la carcasa del módulo de comunicación, ya que no hay espacio suficiente entre esta y la aleta. Además, la rueda dentada en la base se ha diseñado con los parámetros de la Tabla 6. Cabe reseñar que la anchura de los dientes y de la base permite cierta holgura a la hora de ensamblar los componentes, ya que las tolerancias a tener en cuenta son bastante grandes debido al alcance del proyecto y los medios disponibles.

Las aletas cuentan cada una con un agujero pasante para alojar un rodamiento radial que permita apoyar los ejes de la carcasa, permitiendo el libre giro de autovector colinear con el centro del eje. El rodamiento elegido es de la casa Akozon, también de bolas, de diámetro exterior 11 mm, diámetro interior 5 mm y de 5 mm de ancho, todo ello hecho de acero. Por último, debajo de la rueda dentada, se ha dispuesto un apéndice de forma que la parte superior del rodamiento axial no tenga que ir pegada, sino que el apéndice ofrece una resistencia física al movimiento transversal del anillo.

Módulo	Diámetro primitivo	Número de dientes	Tipo de diente	Ángulo de presión	Anchura del diente	Addendum	Dedendum
1 mm	100 mm	100	Evolvente	20°	10 mm	1 mm	1,25 mm

Tabla 6 Características de la rueda dentada de la base móvil

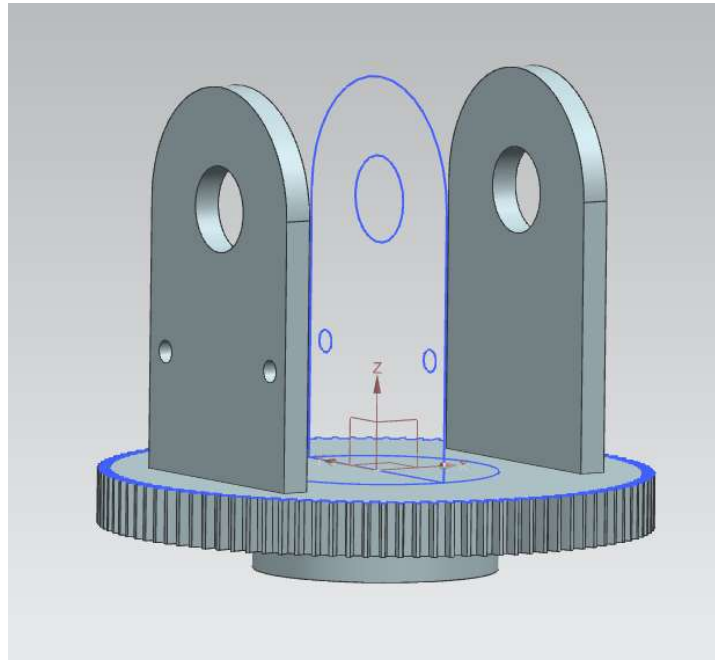


Figura 3-22 Modelo de base móvil

En cuanto a las ruedas dentadas, se han realizado de forma que encastran en el eje de los motores paso a paso empleados. Tienen las mismas características que la de la base móvil, a excepción de la anchura, que será de 5 mm, y del diámetro primitivo, que será de 10 mm, lo que da un número de dientes de 10, como podemos ver en la Figura 3-23. Tendremos por lo tanto dos engranajes en la plataforma, uno constituido por la rueda dentada de un motor y la de la base móvil, que permite el giro el azimut y donde tendremos una reducción de 1/10, y otro constituido por dos ruedas idénticas, una encastrada en el eje de un motor y otra encastrada en un eje de la carcasa, este último de mismas dimensiones que el del motor, que permite el giro en elevación. Este último engranaje no tiene reducción alguna, siendo su única función la de transmitir el par ejercido por el motor al eje.

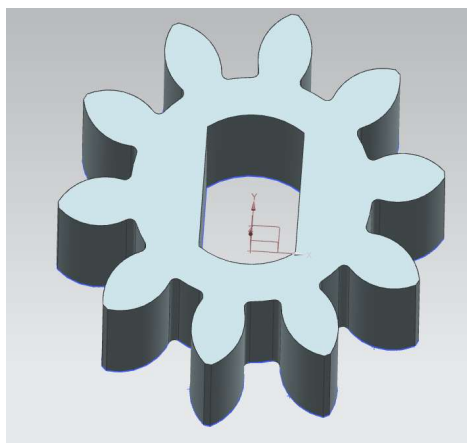
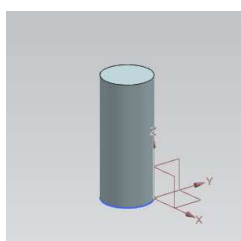
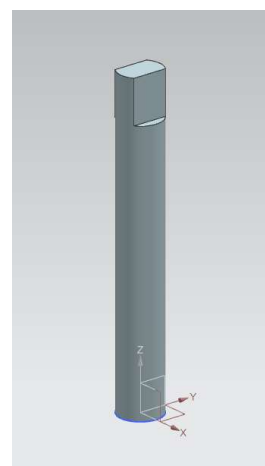


Figura 3-23 Modelo de rueda dentada de diámetro primitivo de 10 mm

Para sostener la carcasa se emplearán dos ejes de diámetro 5 mm que irán apoyados en los rodamientos y de los cuales uno será más largo (correspondiente con la Figura 3-24(b)), ya que aparte tendrá que alcanzar el eje del motor para poder rotar. Este tendrá además el extremo correspondiente al engranaje de la misma forma que el eje del motor, de tal forma que se pueda emplear el mismo engranaje. El otro, correspondiente con la Figura 3-24 (a), simplemente se podrá considerar como un mero bulón, ya que será de longitud reducida debido al poco espacio entre la carcasa y la aleta.



(a)



(b)

Figura 3-24 (a) Modelo de bulón, (b) Modelo de eje transmisor de par

Por último, la carcasa del módulo de comunicación tendrá forma cúbica, a excepción de un apéndice en la parte superior para el soporte de la cámara, como podemos ver en la Figura 3-25. Para las dimensiones, se han tenido en cuenta las del propio módulo de comunicación, cuyo diseño lo ha realizado el AF González del Tánago Landín en su Trabajo Fin de Grado *Estudio y desarrollo de un sistema de comunicaciones punto a punto basado en LASER para buques de la Armada*. Las dimensiones del módulo que nos atañen son las del cilindro que encierra todo el sistema, de diámetro 49 mm y de longitud 100 mm. Por lo tanto, del cubo de lado 56 mm se ha vaciado una proyección de un cuadrado de 50 mm de lado en sentido perpendicular al eje, de manera que el cilindro quede con poco juego, encastrado en el seno de este (ya que las tolerancias son grandes). Además, se han añadido 6 agujeros pasantes en forma de hipódromo para la sujeción del cilindro con corbatas. Por último, el apéndice de la parte superior del cubo tiene una inclinación de $23,56^\circ$ respecto a la horizontal, de manera que el soporte de la cámara, que tenía la misma inclinación respecto a la vertical del suelo, quede perfectamente alineado con la misma. Además, cuenta con dos agujeros pasantes para la sujeción del soporte, en los cuales también se empleará la unión mecánica tornillo-tuerca empleada para la sujeción de los motores.

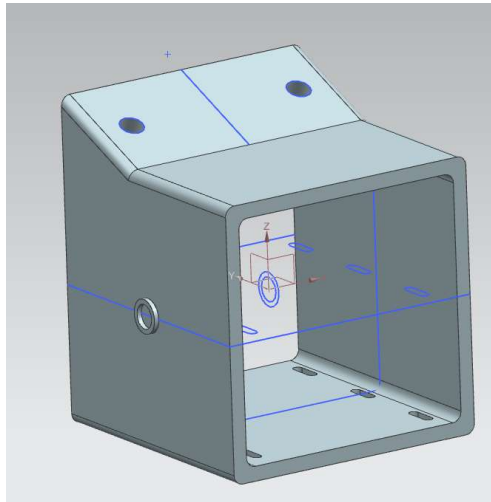


Figura 3-25 Modelo de la carcasa del módulo de comunicación

Sin embargo, tras las pruebas realizadas de la plataforma, que se mencionan en el apartado 4.2.2, se tuvo que realizar un ajuste y diseñar una rueda dentada nueva en el eje donde se encastra la rueda del motor que regula la elevación. Tras numerosas pruebas, el eje no aguantaba la fuerza que ejercía el engranaje en la línea de presión, ya que el momento flector ejercido era demasiado grande y la deformación hacía que el engranaje se desajustase, además de verse la unión con la carcasa francamente perjudicada. Para remediar la situación se ha recurrido a disponer un rodamiento adicional encima del motor, más cerca del engranaje, para reducir el momento flector del eje. Por la forma del motor, se tuvo que disponer mediante una fijación de fortuna, consistente en cinta aislante y en relleno, para evitar que el rodamiento se mueva. Además, se tuvo que crear una rueda dentada más grande, de 14 dientes en vez de 10 (y por lo tanto con un diámetro primitivo de 14 mm) para que el engranaje estuviese ajustado, como el de la Figura 3-26.



Figura 3-26 Modelo de rueda dentada de diámetro primitivo de 14 mm

3.2.3 *Funcionamiento del sistema*

El núcleo para el control de la plataforma es el microcontrolador, subordinado tanto a las instrucciones del microprocesador como a las órdenes del Joystick, y que a su vez controla los motores para orientar el sistema. Este será el encargado de gestionar los motores, y por lo tanto regulará tanto su velocidad como desplazamiento al recibir una orden. Además, se tendrá que programar de tal forma que las órdenes recibidas por el puerto serial o por el Joystick y las acciones tomadas con los motores sean acordes. En otras palabras, si la orden dada por el microcontrolador o por el Joystick es la de mover la plataforma al cuadrante superior izquierdo, los motores tendrán que operar de forma que el movimiento siga las instrucciones. Además, se dispondrá de un Joystick para realizar el posicionamiento inicial de la plataforma en dirección de la estación receptora.

Como podemos ver en el código del Anexo II: Código del micro, este es sencillo, siendo prioritario las órdenes que da la Raspberry a las del Arduino. Si el microcontrolador dispone de datos en el buffer del puerto serial, este los ejecutará con prioridad a las órdenes del Joystick. Aunque el buffer del microcontrolador tenga una capacidad de 64 bytes, nunca se dará el caso de que sature ya que como veremos en el apartado 4.1.1, el programa de tratamiento de imágenes operará a unas velocidades francamente menores que las de procesamiento del Arduino. Siendo así, se tendrá además que en realidad las órdenes del microprocesador no interferirán con las del Joystick, ya que tras la lectura de una orden desde el puerto serial, el Arduino desechará esa información y hará varios bucles hasta que vuelva a haber una orden (suponiendo que haya continuidad en la detección del patrón). Sin embargo, las órdenes del Joystick son mediante lecturas directas de voltaje y por lo tanto siempre se podrá dar una instrucción mediante este. Cabe reseñar que este efecto se reduce cuantos más pasos elijamos para cada orden, como veremos a continuación. En nuestro caso, cogeremos 5 pasos, que como veremos implica que se puede usar el joystick mientras que el microcontrolador recibe información.

Para establecer la lógica de los motores, es necesario establecer unos criterios, así como tener en cuenta su programación. El código se ha diseñado de tal forma que cuando la cámara detecte un patrón en un cuadrante, lo envíe mediante una letra, que corresponde con el cuadrante en un teclado español si tomamos la letra “s” como origen, siendo las letras “a”, “w”, “d” y “x” una orden para un movimiento de giro uniaxial, y las letras “q”, “e”, “c” y “z” uno de giro biaxial, como se ha mencionado en el apartado 3.1.3. Los cuadrantes son en relación a la cámara vista de frente, cosa que también se tiene en cuenta en el código del microprocesador. Sabiendo el cuadrante, también es necesario saber que en la librería de Arduino, un paso positivo equivale a un paso en sentido antihorario si miramos al eje de frente, y por lo tanto uno negativo equivale a uno horario. Por último, estableceremos que el motor encargado del movimiento azimutal sea el 1, mientras que el 2 será el de la elevación. En cuanto a la velocidad de estos, se ha comprobado experimentalmente que los motores 28BYJ-43 no giran a más de 937 rpm, ya que a partir de esta velocidad el eje no sigue las fases de los estatores. Sin embargo, esta será francamente inferior cuando el sistema funciona en automático, por las limitaciones del microprocesador a la hora de analizar imágenes, y en manual, por el procesamiento del microcontrolador.

Si empleamos pasos unitarios para cada orden, teniendo en cuenta que el sistema opera en torno a 500 órdenes/minuto (en condiciones óptimas), como se verá en el apartado 4.1.1, y que en cada vuelta el motor da 32 pasos, obtendríamos la velocidad calculada en la ecuación (17).

$$\omega = \frac{n}{N} \times R \Rightarrow \omega = \frac{500}{32} \times \frac{1}{64} \approx 0,24 \text{ rpm} \quad (17)$$

Donde ω son las revoluciones por minuto a la que opera el motor, n son los pasos por minutos (equivalente al número de órdenes por minuto), N es el número de pasos por vuelta y R es la reducción

interna del motor 28BYJ-48. Esta velocidad es francamente baja, si además añadimos que el engranaje azimutal tiene a mayores una reducción de 1/10. Sin embargo, para optimizar el código, utilizaremos una velocidad máxima de referencia, que será de 900 rpm (para asegurar que el motor no fuerce mucho, ya que ha a mucha velocidad puede saltarse pasos por no seguir las fases de los estatores debido a la carga). Siendo así, para maximizar la eficiencia del sistema, se empleará los tiempos muertos existentes entre las órdenes. Si cogemos una tasa de 450 órdenes/minuto (valor conservador ya que los valores medidos en el apartado 4.1.1 son en condiciones ideales), se puede calcular el tiempo que transcurre en entre mediante la ecuación (18).

$$T = 450 \frac{\text{órdenes}}{\text{min}} = 7,5 \frac{\text{órdenes}}{\text{seg}} \Rightarrow t = \frac{1}{7,5} \approx 0,13 \frac{\text{seg}}{\text{orden}} \quad (18)$$

Donde T es la tasa de órdenes por minuto y t es el tiempo entre cada orden. Por lo tanto, podemos aprovechar este tiempo para mover los motores, de forma que, si calculamos el número de pasos que puede dar el motor en ese tiempo a 900 rpm y lo implantamos en el código, aprovecharemos al máximo la capacidad del código. Por lo tanto, calculamos el número máximo de pasos que puede dar el motor entre una orden y la siguiente con la ecuación (19).

$$p = t \times \frac{\omega}{60} \times N \Rightarrow p = 0,13 \times \frac{900}{60} \times 32 = 62,4 \text{ pasos} \quad (19)$$

Donde p es el número de pasos que se puede dar en un tiempo t , a una velocidad ω (en rpm). N por lo tanto, vuelve a ser el número de pasos por vuelta. Tenemos que entre una orden y la siguiente, se podrán dar como máximo 62 pasos. Cabe reseñar que sobrepasar dicha velocidad, incluso ajustar mucho el funcionamiento a esta, es francamente negativo ya cuando el microcontrolador mueve los motores, no procesa la información que le llega al buffer y este puede llenarse si la tasa es superior de la esperada, o si el microcontrolador opera a una velocidad de operación menor (si la velocidad de rotación se ve disminuida por la velocidad de procesamiento del microcontrolador). Además, hay que tener en cuenta también que el engranaje encargado de regular la elevación es frágil, y por lo tanto la inercia del módulo de comunicaciones le limita mucho. De hecho, para este motor sí que usaremos una avance de pasos unitarios, ya que sino el eje parte (como veremos en el apartado 4.2.2).

Sin embargo, para fijar el número de pasos a emplear, tendremos que tener en cuenta un balance entre resolución angular y velocidad. En efecto, hay que tener en cuenta que, a mayor número de pasos por orden, menor resolución tendremos angularmente, lo cual puede resultar en un movimiento de vaivén si esta es demasiado grande, debido a que el objetivo se sitúa a medio camino entre el inicio de los pasos y el final del movimiento. Si cogemos un paso unitario la resolución será la que se calcula en la ecuación (20).

$$\text{resolución} = \frac{360^\circ \times p}{N} \times R \Rightarrow \text{resolución} = \frac{360^\circ \times 1}{32} \times \frac{1}{64} \approx 0,18^\circ \quad (20)$$

Donde p es el número de pasos por orden, N es el número de pasos por vuelta y R es la reducción interna del motor. Si cogemos el máximo número de pasos, es decir 62, tendremos la resolución calculada en (21).

$$\text{resolución} = \frac{360^\circ \times 62}{32} \times \frac{1}{64} \approx 11,16^\circ \quad (21)$$

Como podemos ver, esta resolución es intolerable. Para nuestro código, que como veremos en el apartado 4.1.2 tiene un alcance estimado de entre 5 y 10 m (con una probabilidad de falsa alarma aceptable), emplearemos un número de pasos de 5, lo cual resulta en la resolución calculada en (22).

$$resolución = \frac{360^\circ \times 5}{32} \times \frac{1}{64} \approx 0,9^\circ \quad (22)$$

La resolución calculada en (21) resulta en la resolución en distancia transversal a la plataforma calculada en (23).

$$\Delta d = 10 \times \tan(0,9^\circ) \approx 0,157 \text{ m} = 15,7 \text{ cm} \quad (23)$$

4 RESULTADOS / VALIDACIÓN / PRUEBA

4.1 Tratamiento de imágenes

4.1.1 Resolución y tasa de frames

Para procesar las imágenes, tendremos que ajustar varios parámetros para adaptar el tratamiento del sistema al hardware empleado y a su capacidad. Debido a las limitaciones de las plataformas empleadas, sobre todo a la de la Raspberry que es la que mayor trabajo computacional realiza, tendremos que ajustar para optimizar el funcionamiento del sistema. Los valores que se ajustarán para optimizar el sistema serán la resolución, la tasa de *frames* por segundos y la tolerancia insertada en la función PatronCuadrado, que se explicará más adelante.

Para las pruebas, se empleará una baliza consistente en cuatro linternas dispuestas en una en cada uno de los vértices de una estructura cuadrada, como se puede ver en la Figura 4-1. Las linternas están constituidas por tres LEDs blancos y un panel reflector para reflejar la mayor cantidad de la luz de los LEDs hacía la lente de la linterna. La estructura que se empleará para sostener las linternas será un cuadrado hecho de tubos de PVC de 1 m de lado cada uno, unidos por codos de plástico, cuyo uso es recurrente en la fontanería. Se unirán los tubos a los codos mediante un pegamento resistente, y las linternas a los codos mediante cinta aislante.

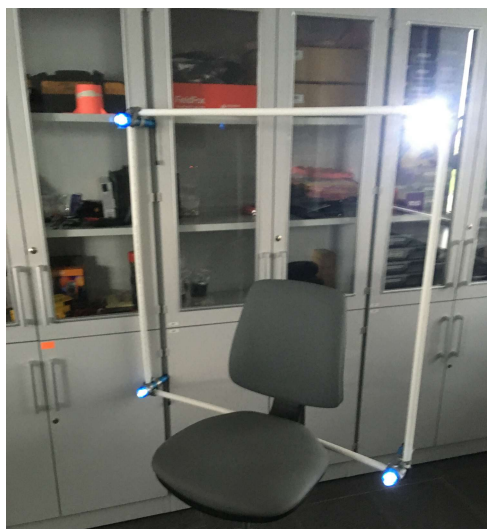


Figura 4-1 Baliza luminosa

En primer lugar, por lo tanto, se determinará la resolución y la tasa de *frames* por segundo que se deberá emplear. Para ello, se han realizado dos experimentos, uno en el que se experimentará el porcentaje de detecciones en condiciones de laboratorio (idóneas) y otro en el que se determinará la velocidad a la que el sistema funciona en función de los dos parámetros estudiados. Los valores que se contemplan para el funcionamiento del sistema en cuanto a resolución guardan una proporción 4:3, siendo estos 320X240, 160X120 y 80X60. El fabricante estipula que para la resolución 680X480, el rango de tasas de *frames* para el funcionamiento óptimo de la cámara está entre 60 y 90, como se puede ver en la Tabla 5. Siendo esta la referencia más precisa disponible, cogeremos este mismo rango, estudiando en concreto las tasas de 60, 70, 80 y 90. Además, aunque luego se determinará la tolerancia a emplear con otra prueba, emplearemos en la prueba de porcentaje de detecciones todo el rango de tolerancias, que como veremos más adelante será de 1 a 13.

Para la ejecución de la primera prueba, dispondremos de condiciones idóneas para el funcionamiento del sistema. Estas se harán en un entorno completamente oscuro, siendo los únicos puntos de luz los de la baliza. Esta se situará en una posición completamente vertical, enfrentada a la cámara y la base del cuadrado estará perfectamente alineada con el suelo. Se sujetará la baliza de forma que durante el experimento no se mueva, siendo las mismas condiciones para todas las mediciones, y se pondrá a una distancia cercana para asegurar su detección (a 2 m). Además, cabe reseñar que emplearemos un umbral de 200 para la prueba, aunque no influye demasiado ya que, además de estar en un entorno completamente oscuro, la baliza está suficientemente cerca como para que los puntos luminosos abarquen más de un píxel, y por lo tanto haya píxeles con una luminosidad de 240, que es la real de las linternas, como podemos ver en la Figura 4-9. Esto último no pasa en pruebas que haremos a continuación, cuando la distancia de la baliza es mayor.

Además, usaremos una modificación del código empleado en el sistema, de forma que, en vez de dar órdenes, cuenta las veces que el código detecta algo pero no es un patrón, y las veces que detecta un patrón. Para ello, emplearemos dos contadores como atributos del objeto analizador, que guardarán la cuenta de las veces que detecta un patrón (`self.contador1`) y las que detecta algo pero no un patrón (`self.contador2`), como se puede ver en la Figura 4-2.

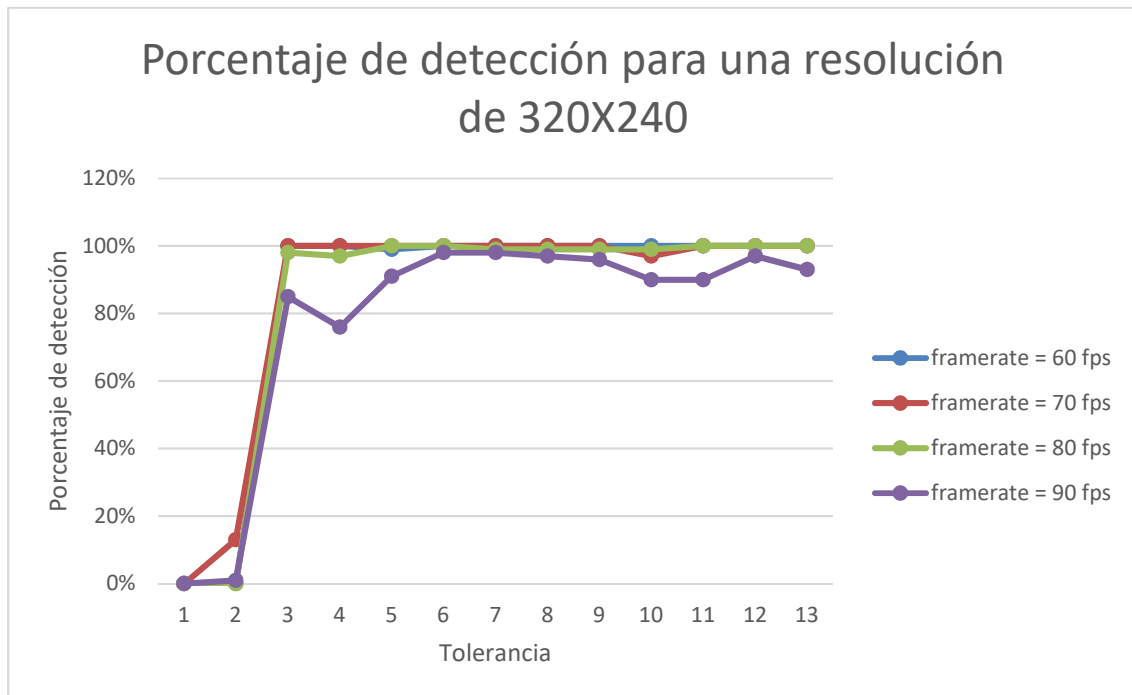
```

77 class MiAnalizador (PiYUVAnalysis):
78     def __init__(self,camara):
79         super (MiAnalizador,self).__init__(camara)
80         self.contador1 = 0
81         self.contador2 = 0
82
83     def analyze (self,a):
84         M = []
85         Xm = []
86         Ym = []
87         umbral = 100
88         Y = a[:, :, 0]
89         U = a[:, :, 1]
90         V = a[:, :, 2]
91         print("inicio")
92         A = np.argwhere(Y >= umbral)
93         if A.shape != (0,2) :
94             M.append(np.array(A[0]).reshape(1,2))
95             A = np.delete(A,0,0)
96             while A.shape != (0,2):
97                 contador = 0
98                 for i,obj in enumerate(M):
99                     obj = np.array(obj)
100                     elem = A[0]
101                     elem = np.array(elem)
102                     [x,y] = CompararValor(obj,elem)
103                     if x.shape != (0,) and y.shape != (0,):
104                         M[i] = AnhadirMatriz(M[i],elem)
105                         A = np.delete(A,0,0)
106                         contador += 1
107                         break
108                 if contador == 0:
109                     M.append(np.array(A[0]).reshape(1,2))
110                     A = np.delete(A,0,0)
111             for i,obj in enumerate(M):
112                 Xmed = int(np.mean(obj[:,1]))
113                 Ymed = int(np.mean(obj[:,0]))
114                 Xm.append(Xmed)
115                 Ym.append(Ymed)
116             if len(Xm) != 0:
117                 B = PatronCuadrado(Xm,Ym)
118                 if B != None:
119                     [X,Y] = B
120                     self.contador1 += 1
121                 else:
122                     self.contador2 += 1
123             print('contador1 ',self.contador1,'          contador2',self.contador2)

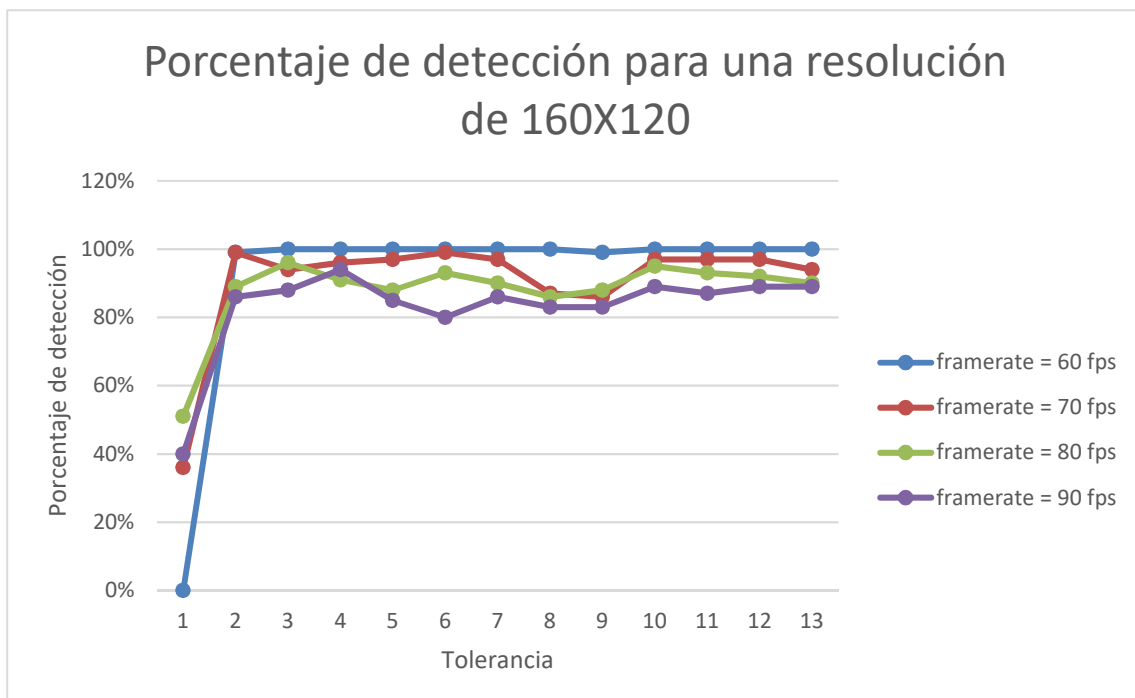
```

Figura 4-2 Código para prueba de resolución y tasa de *frames*

Los resultados de las pruebas se han plasmado a modo de tres gráficas, correspondientes con las Figura 4-3 (a), (b) y la Figura 4-4. En dichas gráficas, podemos ver superpuestas distintas curvas, cada una para una tasa de *frames* por segundo en concreto.



(a)



(b)

Figura 4-3(a) Gráfica de porcentaje de detección para una resolución de 320X280, (b) Gráfica de porcentaje de detección para una resolución de 160X120

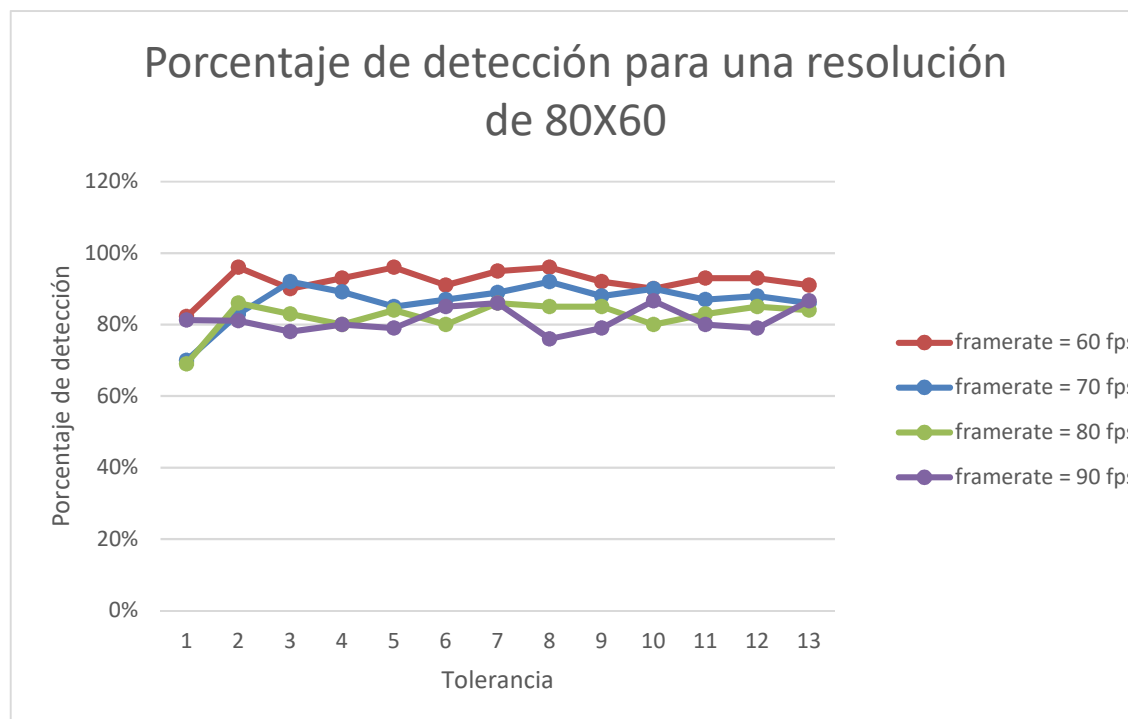


Figura 4-4 Gráfica de porcentaje de detección para una resolución de 80X60

Analizando las tres gráficas, podemos escudriñar varias conclusiones en relación con el funcionamiento de la cámara. Como podemos ver en las gráficas, a mayor resolución, mayor es el porcentaje de detección del patrón. Además, con una misma resolución, a mayor tasa de *frames* por segundo, menor es el porcentaje de detección del mismo. Por último, en cuanto a la tolerancia, podemos ver que influye sobre todo para valores pequeños, es decir cuando se ajusta mucho el margen de error del patrón. Esto se debe a que la baliza no es perfectamente cuadrada, y a que desde la cámara al centro de la baliza hay un cierto ángulo (mínimo) lo cual hace que la proyección de la baliza no sea perfectamente cuadrada. De estos resultados podemos ver que la resolución que más nos interesa es la de 320X240, así como la de 160X120 con una tasa de *frames* baja (60 fps).

La segunda tratará de determinar cuál de las resoluciones y tasas mencionadas en el párrafo anterior es la más interesante, y por lo tanto la que usaremos como parámetros en general para la cámara. La segunda trata de determinar la velocidad a la que el sistema de tratamiento de imágenes, compuesto tanto por la cámara como por el microcontrolador, es capaz de detectar un patrón y enviar una orden acorde con el movimiento a seguir de la plataforma por el serial. El sistema de tratamiento de imágenes operará a una velocidad que dependerá tanto de la resolución como de la tasa de *frames*, añadiendo un tiempo para la transmisión de la orden por el puerto serial, que también requiere al microcontrolador de un esfuerzo considerable de procesamiento. Las condiciones de la prueba son idénticas a las de la prueba anterior, aunque fijaremos la tolerancia a un valor de 10 para que no influya en el resultado (ya que en esta prueba solo influye a valores muy pequeños. Para la realización de la prueba, también se ha retocado el código fuente, como vemos en las Figura 4-5 (a), (b) y (c), de forma que añadiremos un solo contador de atributo al objeto del analizador, además de delimitar el tiempo de grabación. En vez de un bucle “while True” (bucle permanente), utilizaremos el método “.wait_recording(60)” que permite grabar durante un tiempo determinado (que en nuestro caso será de un minuto).

```

77 class MiAnalizador (PiYUVAnalysis):
78     def __init__(self,camara):
79         super (MiAnalizador,self).__init__(camara)
80         self.contador = 0

```

(a)

```

115         if len(Xm) != 0:
116             B = PatronCuadrado(Xm,Ym)
117             if B != None:
118                 [X,Y] = B
119                 print(X, '          ',Y)
120                 if X < XRES/2-1 and Y > YRES/2:
121                     port.write("q".encode())
122                 elif XRES/2-1 <= X <= XRES and Y > YRES/2:
123                     port.write("w".encode())
124                 elif X > XRES/2 and Y > YRES/2:
125                     port.write("e".encode())
126                 elif X > XRES/2 and YRES/2-1 <= Y <= YRES/2:
127                     port.write("d".encode())
128                 elif X > XRES/2 and Y < YRES/2-1:
129                     port.write("c".encode())
130                 elif XRES/2-1 <= X <= XRES/2 and Y < YRES/2-1:
131                     port.write("x".encode())
132                 elif X < XRES/2-1 and Y < YRES/2-1:
133                     port.write("z".encode())
134                 elif X < XRES/2 and YRES/2-1 <= Y <= YRES/2:
135                     port.write("a".encode())
136                 elif XRES/2-1 <= X <= XRES/2 and YRES/2-1 <= Y <= YRES/2:
137                     self.contador += 1

```

(b)

```

141 with picamera.PiCamera(resolution='160X120', framerate=60) as camara:
142     with MiAnalizador(camara) as analizador:
143         port = serial.Serial('/dev/ttyACM0',9600)
144         camara.start_preview(alpha=128)
145         sleep(2)
146         camara.start_recording(analizador,'yuv')
147         try:
148             camara.wait_recording(60)
149         finally:
150             camara.stop_recording()

```

(c)

Figura 4-5 (a), (b) y (c) Modificaciones al código para la realización de la prueba de velocidad

Una vez realizada la prueba, se han plasmado los resultados en la Tabla 7.

		RESOLUCIONES		
		320X280	160X120	80X60
FPS	60	224	492	550
	70	310	501	558
	80	352	517	565
	90	369	540	571

Tabla 7 Resultados de prueba de velocidad (en órdenes por minuto)

Podemos ver por lo tanto que, a mayor resolución, menor velocidad de procesamiento tiene la cámara. Sin embargo, a mayor tasa de *frames*, mayor velocidad de procesamiento y, por lo tanto, más órdenes por minuto. En nuestro sistema, nos interesa la mayor velocidad posible ya que el movimiento del motor dependerá de esta.

De las dos pruebas, podemos concluir que, si buscamos una probabilidad de detección alta, a la vez que una velocidad de procesamiento alta para una tasa de órdenes alta, los parámetros a emplear son una resolución de 160X120 con una tasa de *frames* de 60 fps. Esta combinación nos permite una gran capacidad de detección (en torno a 100% en las pruebas realizadas para una tolerancia superior a 2), y la máxima velocidad posible para esa capacidad, ya que, aunque una resolución de 320X280 tiene una capacidad de detección muy buena, la velocidad de procesamiento es más de 100 unidades inferior al caso elegido.

4.1.2 Tolerancias

Como se ha visto en el punto 3.1.3, se han empleado tolerancias a lo largo del código para optimizar su funcionamiento. Tanto en la función *ComparaValor* del código, que identifica píxeles luminosos a un objeto determinado, como en la función *PatronCuadrado*, encargada de reconocer el patrón de luces, las tolerancias permiten darle al tratamiento de imágenes eficiencia ya que este se ve influenciado por factores tanto internos (calidad de la cámara, funcionamiento del GPU y CPU,...) como externos (precisión de la arquitectura de la baliza, orientación de esta, luz ambiental,...) que dificultan mucho su funcionamiento.

La primera tolerancia implementada influye a la hora de juntar píxeles en objetos mediante la función *CompararValor*, mencionada en el apartado 3.1.3. Dicha tolerancia se implementa por lo tanto en el código de comparación de los píxeles con los objetos ya existentes, y permite justamente añadir píxeles que no estén exactamente situados en una posición adyacente al objeto. Usaremos esta tolerancia sobre todo para corregir los efectos de la lente de la cámara, que produce reflejos en el sensor, y para corregir los casos en los que la intensidad de la luz sea demasiado elevada y se produzca efecto de blooming. Los dos efectos son altamente variables y dependen tanto de la luz de la fuente, como de la luz de fondo y de los ajustes internos que hace la cámara en función de estos dos. Aun así, cabe reseñar que estos factores surgen principalmente en pruebas en condiciones de laboratorio, principalmente cuando la baliza se posiciona a distancias cortas (menos de 3 m). Por lo tanto, para eliminar esos efectos, se empleará una tolerancia de 3 únicamente para las pruebas de laboratorio (cosa que no influye en los resultados de las pruebas ya que los únicos puntos luminosos son los de la baliza) mientras que se usará una tolerancia de 1 para las demás pruebas.

La segunda tolerancia que nos afecta es la de detección del patrón cuadrado. Esta tolerancia se implementa en la función *PatronCuadrado* mencionada en el apartado 3.1.3, función encargada de detectar el patrón cuadrado de puntos luminosos de la baliza. La tolerancia juega por lo tanto un rol determinante en esta función, y por extensión en todo el proceso de tratamiento de imágenes, ya que de esta depende en última instancia la detección de un patrón o no. Para determinar que tolerancia es la óptima para el sistema, haremos una prueba consistente en determinar para cada tolerancia el porcentaje de detección del patrón. Sin embargo, las condiciones de la prueba cambiarán respecto a las realizadas anteriormente.

Esta prueba se hará en exteriores, con un fondo luminoso, aunque se hará de noche (porque como veremos más adelante, durante el día el sistema se ve muy limitado). El lugar donde se podrá la cámara es la ventana norte del Laboratorio de Radar del edificio de Laboratorios de Investigación de la Escuela Naval Militar, dirigida hacia el cuartel Marqués de la Victoria, como podemos ver en la Figura 4-6. Además, probaremos la detección a distintas distancias: 3 m, 5 m, 10 m y 15 m. Las pruebas se harán con la baliza correctamente orientada (a la vertical y con la base paralela a la horizontal), pero se moverá la baliza transversalmente a la dirección de la cámara, tanto a un lado y a otro, como hacia arriba y hacia abajo. Por último, haremos una prueba sin el patrón para detectar el porcentaje de falsa alarma, cuyos resultados se plasmarán más adelante. Cabe reseñar que durante toda la prueba se mantendrá una resolución de 160X120 con una tasa de 60 fps, además de que se usará el mismo código de detección de patrón que el empleado para la primera prueba mencionada en el apartado 4.1.1

y una tolerancia de 1 en la función de CompararValor, modificando por lo tanto únicamente en cada muestra la tolerancia en la función PatronCuadrado.



Figura 4-6 Fotografía del emplazamiento donde se llevan a cabo las pruebas de tolerancia

Como podemos ver en la gráfica de la Figura 4-7, los resultados muestran que una tolerancia de más de 9 permite detectar un patrón con relativo éxito hasta 15 m, mientras que unas tolerancias pequeñas son poco eficaces, sobre todo a distancias muy cortas o muy largas. Sin embargo, como se verá en el punto 4.1.3, justamente a partir de una tolerancia de 9, el porcentaje de falsa alarma se dispara. Por lo tanto, podemos determinar que la tolerancia óptima para el código es o 7 u 8, ya que permiten una detección casi segura hasta 5 m y permiten llegar hasta 10 m, con una probabilidad de falsa alarma nimia. Elegiremos una tolerancia de 7 en pro de dar la máxima precisión, manteniendo una distancia de detección máxima y una probabilidad de falsa alarma mínima. Cabe por último reseñar que la distancia máxima del sistema no se corresponde con la reflejada en la gráfica (menor de 15 m con las tolerancias elegidas), ya que esta está relacionada también con la probabilidad de falsa alarma y los factores que en ella influyen, y que se tratarán en el apartado 4.1.3. Por lo tanto, la distancia máxima del sistema dependerá de la probabilidad máxima de falsa alarma tolerable, así como del porcentaje de detección mínimo tolerable.

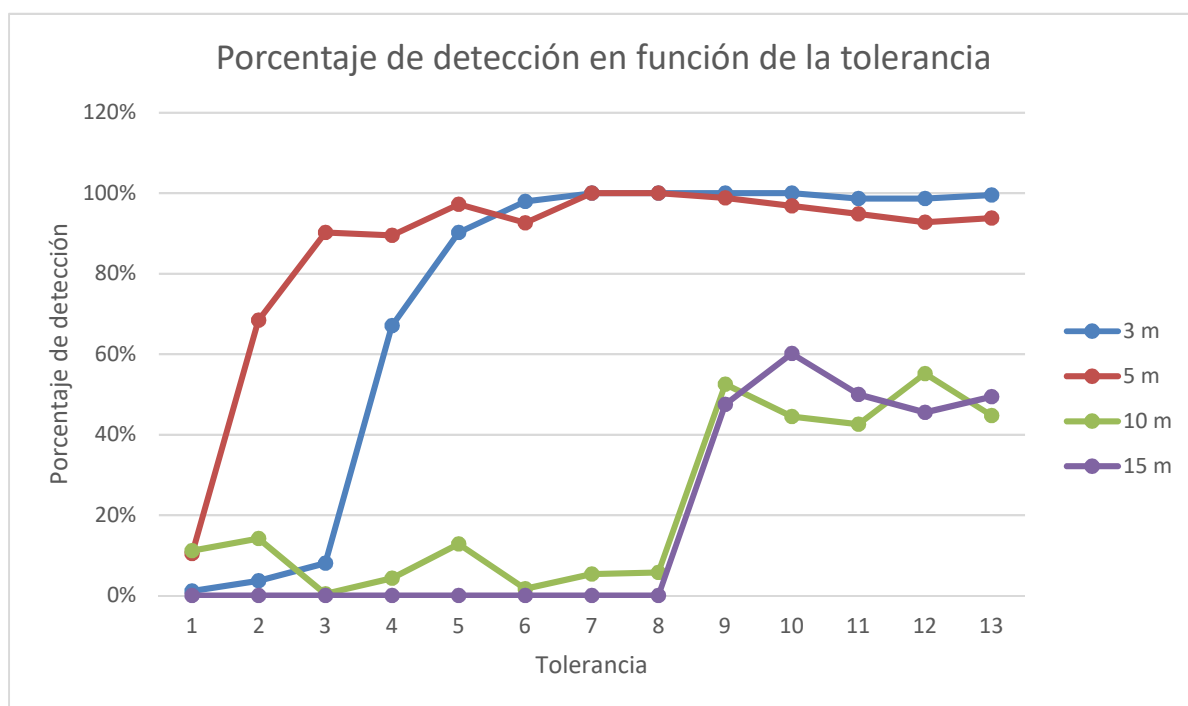


Figura 4-7 Gráfico de porcentaje de detección del patrón en función de la tolerancia

4.1.3 Falsa alarma

Para determinar con precisión los parámetros óptimos de nuestro sistema, deberemos tener también presente el porcentaje de falsa alarma, siendo este el porcentaje de veces en los que detecta erróneamente un patrón. Este sistema puede fallar y detectar un patrón cuando no lo hay, o cuando lo hay, pero no es el de la baliza. En este Trabajo Fin de Grado, analizaremos el porcentaje de falsa alarma de un entorno luminoso cuando no está la baliza, aunque hay que tener en cuenta que también puede ocurrir que, aunque la baliza esté en pantalla, detecte otro patrón que no se corresponda con la baliza.

En primer lugar, es trivial deducir que el porcentaje de falsa alarma dependerá de la tolerancia del código. En efecto, a mayor tolerancia, mayor probabilidad de que el analizador de imágenes detecte ciertos puntos luminosos que ocasionalmente coincidan con un patrón cuadrado, aunque la cámara esté en estático. Por lo tanto, probaremos cual es la probabilidad de falsa alarma en el mismo entorno que en el de la prueba mencionada en el punto 4.1.2. Probaremos por lo tanto diferentes tolerancias, con la cámara funcionando con los parámetros ya mencionados, y determinaremos a partir de qué tolerancia el porcentaje de falsa alarma resulta excesivo. Los resultados de dicha prueba se reflejan en la Figura 4-8.

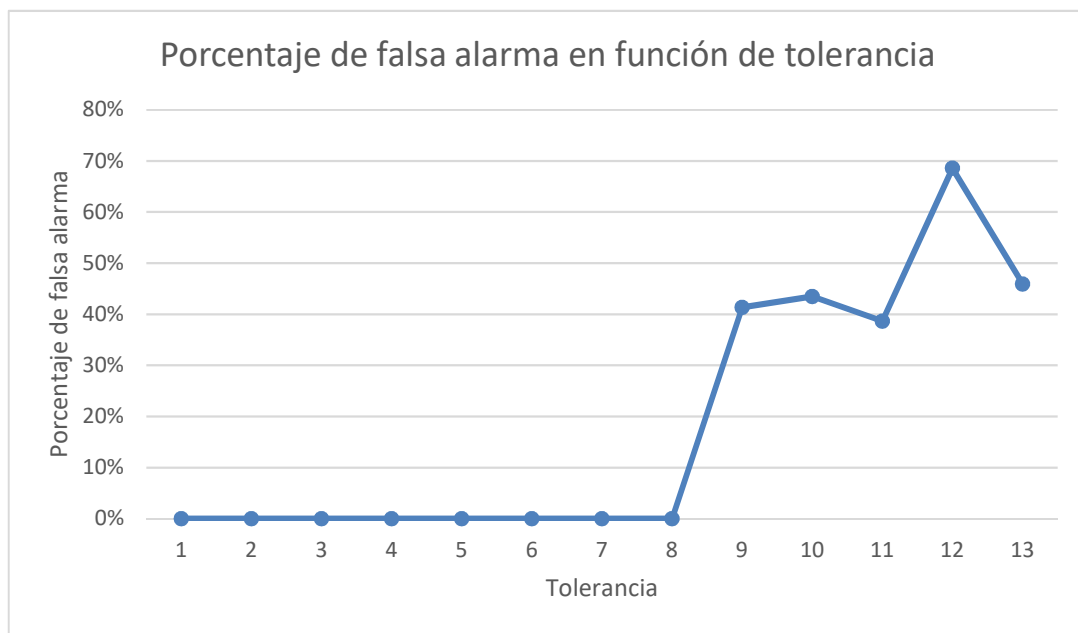


Figura 4-8 Gráfico de porcentaje de falsa alarma en función de tolerancia

Como podemos ver, a partir de una tolerancia de 9, el porcentaje de falsa alarma aumenta por encima del 35%. Siendo así, como se ha mencionado en el apartado 4.1.2, una tolerancia superior a 8 no es interesante, y por lo tanto la óptima será o 7 u 8.

Sin embargo, otro factor reseñable que influye notablemente en el porcentaje de falsa alarma es el del umbral utilizado en las pruebas. En efecto, el nivel de luminosidad que se especifique influye notablemente en la probabilidad de falsa alarma, ya que, debido a la resolución, los píxeles pueden no medir un nivel de luminosidad correspondiente con el de la linterna. Mediante el código del Anexo III: Código de muestro de la luminosidad, podemos medir el nivel de luminosidad de una parcela de píxeles en concreto. Si medimos el nivel de la linterna, obtenemos un nivel que oscila entre 240 y 243, como podemos ver en la Figura 4-9. Sin embargo, este nivel solo se puede obtener en mediciones muy cercanas a la cámara.

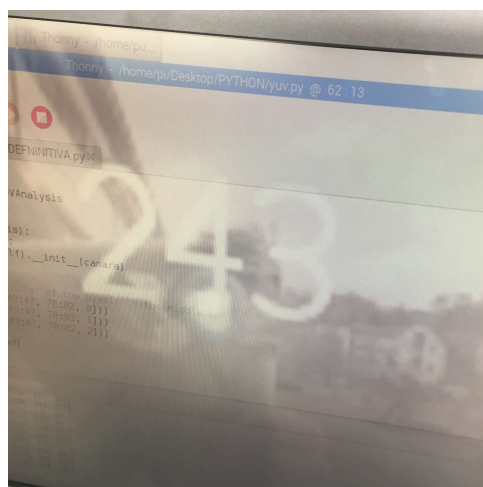


Figura 4-9 Nivel de luminosidad de la linterna

Con la resolución que empleamos, al alejar la baliza de la cámara, los focos luminosos no ocupan un píxel en su totalidad. Por lo tanto, el nivel luminoso del píxel baja, de forma que es necesario ajustar el umbral. Este ajuste va de la mano de un aumento del porcentaje de falsa alarma, ya que, al bajar el umbral, más píxeles se tienen en cuenta como luminosos y pueden interferir en la detección de un patrón. Siendo así, se ha realizado una prueba de falsa alarma en las mismas condiciones que la prueba anteriormente realizada, pero fijando la tolerancia a 7. Los resultados de la Figura 4-10 muestran claramente que, a partir de un umbral de 190, las interferencias del entorno son excesivas y por lo tanto no es viable la detección de un patrón. Para nuestro código, emplearemos un umbral de 200, umbral que permite por lo tanto asegurar una probabilidad de falsa alarma casi nula pero que permita reconocer la baliza.

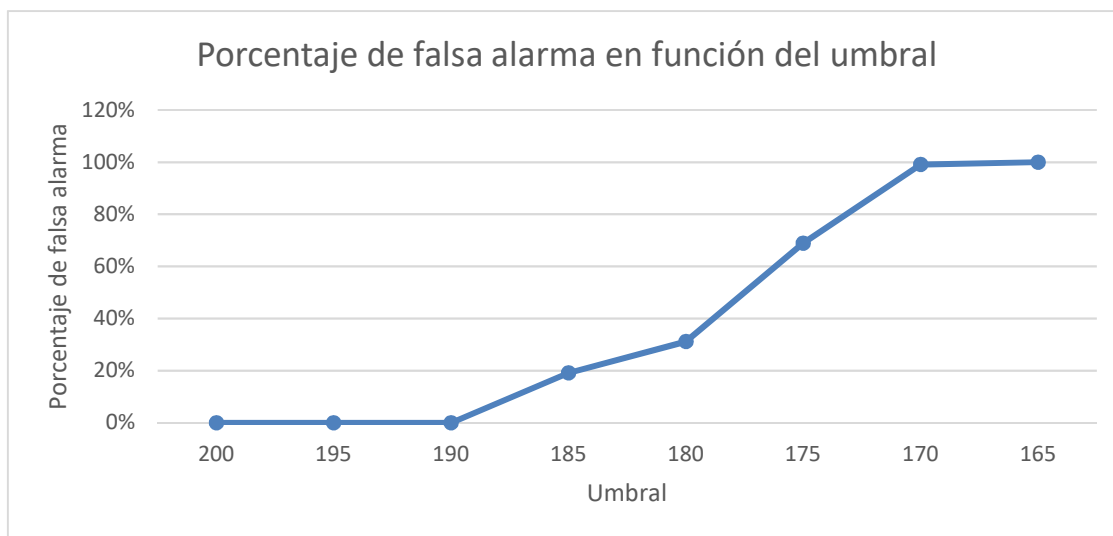


Figura 4-10 Gráfico de porcentaje de falsa alarma en función de umbral

4.1.4 Limitaciones del sistema

En cuanto al tratamiento de imágenes, el sistema tiene grandes limitaciones debido al hardware empleado. Al igual que se ha realizado pruebas de noche, como las de los apartados 4.1.2 y 4.1.3, también se ha realizado pruebas de día para probar el funcionamiento del sistema. Dichas pruebas resultaron en un fracaso a la hora de probar el sistema por diversos motivos. Como veremos más adelante, la limitación se debe a tres factores: la luminosidad propia del entorno, la capacidad del microprocesador y los ajustes de la cámara. Cabe reseñar que en las pruebas realizadas se empleó una resolución 160X120 con una tasa de 60 fps, un umbral de 200, que se subió a 240 para las pruebas a una distancia corta (inferior a 3 m), una tolerancia de 7 y en el mismo emplazamiento de las pruebas en exteriores de los apartados 4.1.2 y 4.1.3. Las pruebas se realizaron entorno a las 1300 de la tarde, cuando el sol está en su máxima altura (y por lo tanto ilumina más), en un día con condiciones meteorológicas buenas, como podemos ver en la Figura 4-11.



Figura 4-11 Fotografía de las pruebas de día en exteriores

El primer factor limitante que encontramos a la hora emplear el sistema en un sistema como tal es efectivamente el nivel lumínico del ambiente. Simplemente midiendo con el código del Anexo III: **Código de muestro de la luminosidad**, la luminosidad del entorno, podemos ver que efectivamente hay multitud de puntos muy luminosos, alcanzando los niveles de luminosidad de la linterna a corta distancia (es decir valores en torno a 240), como podemos ver en la

Figura 4-12. Además, estos valores son de gran variabilidad, sobre todo los correspondientes al cielo y a las nubes. Por lo tanto, esa gran densidad de puntos lumínicos es francamente perjudicial para el sistema, siendo además una densidad altamente variable. De las pruebas realizadas con el mismo código que en la prueba realizada en el apartado 4.1.2, aún aumentado la resolución a 320X240 y probando diferentes tolerancias a distancias cortas (menores de 3 m), solo se ha podido concretar que con una tolerancia por debajo de 3, el código no detecta nada, y por encima solo detecta un porcentaje de falsa alarma entorno al 50%.

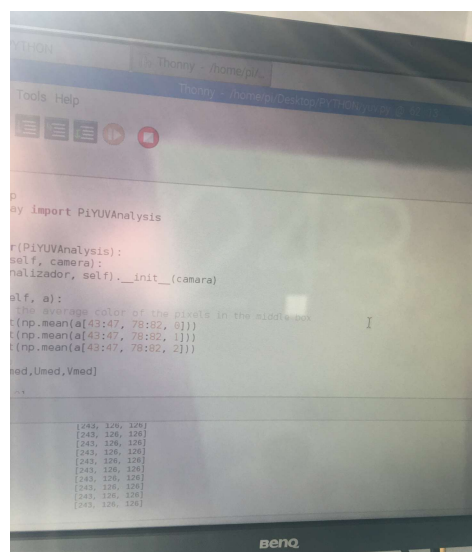


Figura 4-12 Muestra de la luminosidad del entorno buenas condiciones meteorológicas (en torno a 240)

El segundo factor limitante que influye en el sistema es la capacidad del microprocesador. Durante la noche o en condiciones donde no hay gran luminosidad en el entorno, el microprocesador es capaz de realizar las operaciones en un tiempo razonable. Sin embargo, en un entorno luminoso como este, en el que como ya se ha mencionado una gran parte de los píxeles cumplen el requisito de estar por encima del umbral, los datos a procesar crecen de manera sustancial. Tal cantidad de datos, al ser procesados, resultan en un crecimiento exponencial de las operaciones a realizar, lo cual satura la máquina. Esto se traduce en que, detectando o no un patrón, el código emite un resultado en un tiempo exageradamente largo, volviendo inútil por lo tanto el resto del sistema. En concreto, se ha medido que los resultados del código para cada tratamiento de imagen se emitían con una cadencia inferior a la unidad por minuto, llegando en ocasiones a menos de una cada dos minutos.

Por último, el funcionamiento del sistema se ve completamente anulado por un tercer factor, que son los ajustes internos que hace la cámara. De forma automática, la cámara procede con ajustes automáticos para optimizar la calidad de las imágenes. Cuando el entorno es así de luminoso, los ajustes de la cámara anulan completamente los puntos luminosos de las linternas empleadas, quedando solo los del entorno. Como podemos ver en la Figura 4-13, a simple vista podemos ver que los puntos luminosos quedan totalmente anulados, no viéndose ningún punto luminoso destacable en los vértices de la baliza.



Figura 4-13 Fotografía de la baliza en buenas condiciones meteorológicas

Por lo tanto, podemos concluir que el sistema no puede operar en entornos excesivamente luminosos, siendo estos entornos en los que se sobrepase un umbral en torno a 200. Dichos entornos pueden ser perfectamente escenarios de noche o en cercanía a los crepúsculos matutinos y vespertinos, así como escenarios con cielo cubierto. En dichos escenarios, es de reseñar que la máxima velocidad de procesamiento es la expuesta en las pruebas mencionadas en el apartado 4.1.1, resultando en torno a 500 órdenes al microcontrolador como máximo.

4.2 Plataforma

4.2.1 Calibración

En cuanto a la plataforma, se han realizado en primera instancia pruebas para corregir desajustes entre la cámara y la plataforma en sí. La cámara va montada en un soporte, el cual está sujeto a la cámara mediante una unión mecánica consistente en dos tornillos y dos tuercas. Además, se ha diseñado un pedestal para la cámara de forma que esta quede a la vertical, ya que el montaje de la cámara tiene una inclinación de $23,56^\circ$, como se ha mencionado en el apartado 3.2.2. Sin embargo, debido a las tolerancias de las piezas debido a la impresora 3D y al juego entre los diferentes

componentes, la alineación de la cámara con la plataforma no es perfecto. Se requiere por lo tanto de una calibración del sistema al completo.

Dicha calibración se hará en primer lugar mediante unas variables que se insertarán en el código. A la hora de decidir qué orden enviar a la cámara, emplearemos estas variables para determinar que píxeles corresponden con el eje del módulo de comunicación de la plataforma, sobre el que va montado la cámara. Las variables en sí se denominarán XRES e YRES, fijándolas al principio del código del analizador e implementándolas al final del código, a la hora de decidir qué orden dar a través del serial en función del centro del patrón, como se puede ver en las Figura 4-14 (a) y (b).

```

77 class MiAnalizador (PiYUVAnalysis):
78     def __init__(self,camara):
79         super (MiAnalizador,self).__init__(camara)
80
81     def analyze (self,a):
82         M = []
83         Xm = []
84         Ym = []
85         umbral = 180
86         Y = a[:, :, 0]
87         XRES = 160
88         YRES = 120

```

(a)

```

115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137

```

```

if len(Xm) != 0:
    B = PatronCuadrado(Xm,Ym)
    if B != None:
        [X,Y] = B
        print(X, '          ', Y)
        if X < XRES/2-1 and Y > YRES/2:
            port.write(("q".encode()))
        elif XRES/2-1 <= X <= XRES and Y > YRES/2:
            port.write(("w".encode()))
        elif X > XRES/2 and Y > YRES/2:
            port.write(("e".encode()))
        elif X > XRES/2 and YRES/2-1 <= Y <= YRES/2:
            port.write(("d".encode()))
        elif X > XRES/2 and Y < YRES/2-1:
            port.write(("c".encode()))
        elif XRES/2-1 <= X <= XRES/2 and Y < YRES/2-1:
            port.write(("x".encode()))
        elif X < XRES/2-1 and Y < YRES/2-1:
            port.write(("z".encode()))
        elif X < XRES/2 and YRES/2-1 <= Y <= YRES/2:
            port.write(("a".encode()))
        elif XRES/2-1 <= X <= XRES/2 and YRES/2-1 <= Y <= YRES/2:
            self.contador += 1

```

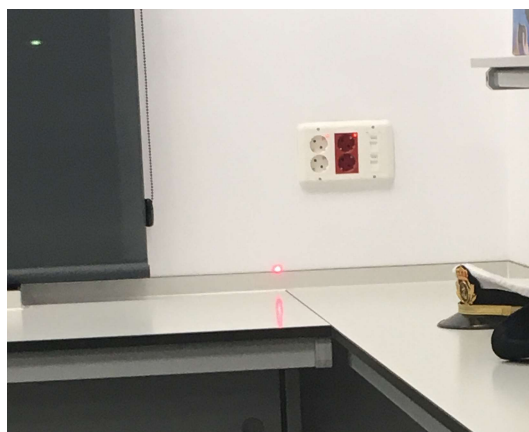
(b)

Figura 4-14 (a) y (b) Partes del código donde interviene las variables de calibración

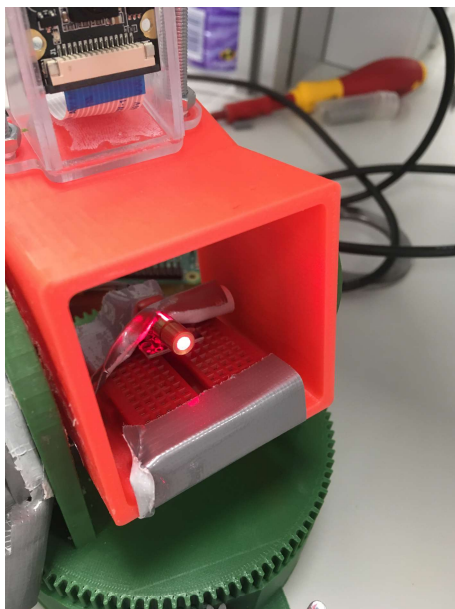
Para la prueba de calibración, se dispondrá de un diodo Láser que ha sido previamente calibrado en una superficie plana. Para ello, se montará encima de una Protoboard y se fijará con cinta aislante de forma que el diodo quede en la misma posición en la Protoboard, como se puede ver en la Figura 4-15 (a). El conjunto se calibrará en una superficie plana, ajustando el conjunto de forma que el Láser apunte en la dirección deseada, como podemos ver en la Figura 4-15 (b). Una vez calibrado, se fijará el conjunto dentro del módulo de comunicación, como se observa en la Figura 4-15 (c).



(a)



(b)



(c)

Figura 4-15 (a) Módulo de calibración, (b) Calibración del módulo de calibración, (c) Calibración de la plataforma

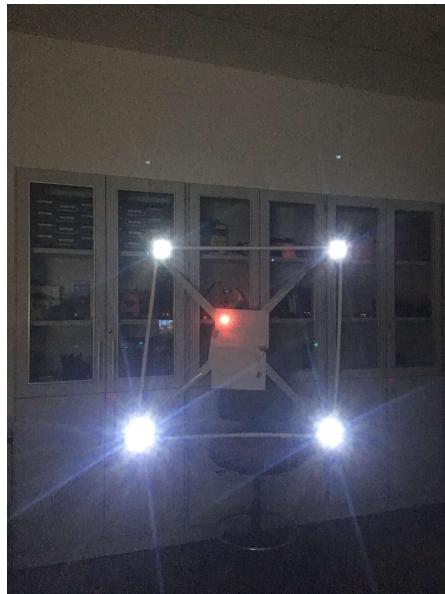
Una vez preparado el montaje, se empezará por poner como variables XRES e YRES en el código la resolución a la que se está operando, que como se ha mencionado en el apartado 4.1.1 será de 160X120. Una vez dispuestos los parámetros iniciales, la prueba de calibración consistirá simplemente en hacer funcionar el código del sistema, de forma que el sistema se alinee con lo que considere que es el centro del patrón. La baliza se mantendrá fija en una posición, de forma que a cada prueba se alteren los parámetros XRES e YRES hasta que el Láser termine por apuntar al centro de la baliza, como se pueda ver en las Figura 4-16 (a), (b), (c) y (d). Para ello, se ha dispuesto de un panel de papel en el centro de la baliza de forma que se posible ver el desvío del punto Láser respecto al centro del patrón.



(a)



(b)



(c)



(d)

Figura 4-16 (a), (b), (c) y (d) Proceso de calibración de la plataforma

Los resultados de la prueba muestran que el sistema se calibra para unos valores de $XRES = 193$ e $YRES = 90$. Cabe reseñar que la calibración no es válida para un periodo de tiempo prolongado o cuando el sistema se mueve ya que dicho sistema tiene partes que no están totalmente fijas, como es la propia cámara que está encastrada en el soporte y puede moverse debido a movimientos de la plataforma o a tirones del bus. Es por lo tanto recomendable una calibración previa a cada uso del sistema, así como usar el propio sistema de comunicación para hacer la calibración.

4.2.2 Transmisión de movimiento

Para probar el correcto funcionamiento del sistema, se ha realizado en primer lugar pruebas con el Arduino y mediante el Joystick para probar la movilidad de la plataforma. En dichas pruebas, varios cambios tuvieron que tenerse en cuenta para la implementación final de la plataforma, y en concreto de los componentes relacionados con el movimiento de elevación.

Una vez ensambladas todas las piezas, cuyas únicas uniones fijas son la de los rodamientos radiales en los agujeros de las aletas de la base móvil, la de los extremos de los ejes en la carcasa del módulo de comunicaciones (ambas mediante adhesivo), la de los motores a la plataforma y la de la cámara (estas dos últimas mediante unión mecánica), se probó la transmisión de movimiento en el engranaje azimutal con éxito. Sin embargo, al probar el engranaje encargado de ajustar la elevación, se comprobó que este sufría una deformación exagerada ya que está a demasiada distancia del rodamiento. La transmisión de la fuerza ejercida por el engranaje en la línea de presión de este se transmite a la estructura a la altura del rodamiento, lo que da lugar a un momento flector importante, que deforma el eje, desajustando el engranaje. El resultado era que el engranaje en vez de girar, acompañaba el movimiento del piñón, hasta que llegaba un punto en que él se trababa. Para remediar esta situación, se tomaron dos medidas. La primera consiste en disponer otro rodamiento apoyado sobre el motor y más cerca del engranaje. De esta manera, el momento flector se reducía considerablemente (proporcionalmente a la distancia), de forma que ya no se deformaba tanto. Para ello, se tuvo que subir el motor, además de sujetar el rodamiento mediante cinta aislante y con relleno a los lados para evitar que este rodase sobre el motor, como se puede apreciar en la Figura 4-17.

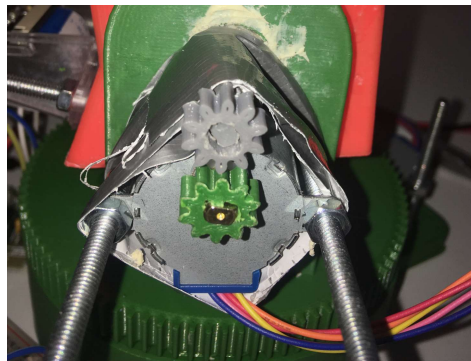


Figura 4-17 Fotografía del engranaje encargado de la elevación con ruedas dentadas iguales

Además, la segunda medida que se tomó, y que fue a consecuencia de la primera, fue la creación de otra rueda dentada. Se había concebido el sistema de forma que el eje necesitase la misma rueda que el eje del motor, ambas encastradas en sus respectivos ejes. Sin embargo, resultó que esta se quedaba corta debido a la disposición del eje, y se diseñó por lo tanto otra de diámetro primitivo 14 mm (cuando los anteriores eran de 10), ya que como se puede ver en la Figura 4-17, las crestas de los dientes se tocan, lo cual significaba que los diámetros primitivos estaban separados dos veces el addendum, o lo que es lo mismo dos veces el módulo, que era de 1 mm. El diseño de dicha rueda dentada se puede ver en el apartado 3.2.2, y corresponde con la Figura 4-18. Mediante la nueva disposición del motor se consiguió eliminar el problema, funcionando el sistema correctamente.

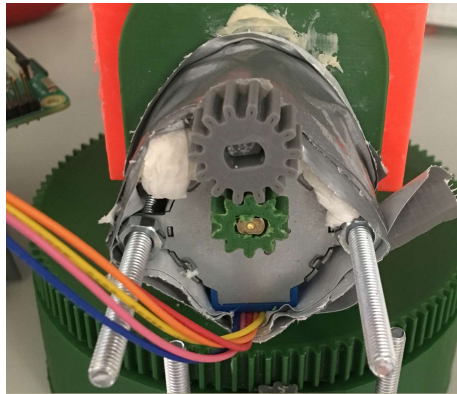


Figura 4-18 Fotografía del engranaje encargado de la elevación final

Por otro lado, otro inconveniente surgió con el otro extremo del eje. Este iba unido en el seno de un pedestal en el costado de la carcasa mediante adhesivo. Sin embargo, resultó que la unión no aguantaba la transmisión del par entre eje y carcasa, despegándose con gran facilidad. Para remediarlo, se perfiló el extremo del eje, dejándolo con una forma similar a la del otro extremo, y se perforó el costado de la carcasa, de 3 mm de ancho, practicando una hendidura de forma que el extremo perfilado encastrase en dicha hendidura. Para asegurar la unión, se aplicó también adhesivo, resultando en éxito a la hora de probar el sistema.

A la hora de probar distintas velocidades y pasos para la determinación de pasos a dar entre orden y orden, como se ha mencionado en el apartado 3.2.3, el extremo del eje partió. Esto se debe a que la inercia de la carcasa y la cámara es demasiado para el eje, y a la hora de mover el eje a 900 rpm (aunque haya una reducción de 1/64 en el motor y de 1/1,4 en el motor) la aceleración del motor produce un momento torsor excesivo. Siendo así, se tendrá este hecho como una limitación del sistema, en el que por lo tanto el motor solo podrá dar pasos unitarios por cada orden. Sin embargo, no supone una limitación excesiva ya que al ser el objetivo de este Trabajo Fin de Grado el diseño y la implementación en plataformas navales, no se requiere una corrección grande en elevación sino más bien en azimut.

5 CONCLUSIONES Y LÍNEAS FUTURAS

5.1 Conclusiones del Trabajo Fin de Grado

Los dos objetivos planteados para este Trabajo Fin de Grado fueron el de diseñar e implementar tanto un sistema de seguimiento para un sistema FSOC, como una plataforma para dicho sistema. Para cumplir con dichos objetivos, se ha diseñado una plataforma que permite el apuntamiento mediante un movimiento de toda la plataforma, así como un sistema de seguimiento basado en una baliza luminosa. La dificultad por lo tanto de este TFG recae en primera instancia en la variedad de ámbitos en los que se ha trabajado, siendo el del sistema de seguimiento un ámbito eminentemente informático, aunque también de componente electrónico, y el de la plataforma un ámbito tanto electrónico como mecánico.

En primer lugar, en cuanto al sistema de seguimiento, se ha propuesto una solución consistente en la detección de un patrón de puntos luminosos mediante un tratamiento de imágenes con un microprocesador Raspberry Pi y una Picamera. Se ha elegido dicho modo de proceder mediante un patrón debido a la componente marítima de este proyecto, ya que el entorno marino puede resultar ser muy luminoso debido a la densidad de focos de luz en ciudades, puertos, plataformas navales,... Dicho propósito se ha cumplido, desarrollando un sistema que permite detectar el mencionado patrón y dar las directrices pertinentes para permitir un seguimiento de este mediante la orientación de la plataforma. En concreto, se ha concebido un programa que es capaz de discernir un patrón cuadrado mediante un análisis de la luminosidad de todos los píxeles, y posterior tratamiento de imágenes.

Además, se han realizado estudios sobre el rendimiento del sistema para su optimización mediante el estudio de diferentes parámetros tanto relativos al propio código desarrollado como relativos al hardware empleado. Cabe reseñar que ciertos parámetros han sido concebidos a la hora de desarrollar el propio código, no habiendo sido concebidos por los desarrolladores de los componentes empleados, como es el caso de las tolerancias, lo cual supone una gran ventaja del código ya que le otorga versatilidad. Además, en los estudios se han empleado códigos auxiliares que permitieron medir con precisión los resultados de las distintas pruebas, resumiéndose en conclusiones en cuanto al modo de funcionamiento más interesante para cumplir con los objetivos propuestos, aunque también se destacan los balances generales de los parámetros que se han estudiado, como por ejemplo el balance entre porcentaje de falsa alarma, porcentaje de detección y distancia de la baliza, o el balance entre velocidad de procesamiento y resolución.

En segundo lugar, en cuanto a la plataforma, se ha optado por una opción sencilla a la vez que viable, dado el material disponible. Como se ha mencionado en el apartado 2.3.2, existe una amplia gama de plataformas para un sistema FSO que permiten llegar a precisiones del orden de los nanoradianes, pero cuya complejidad y necesidad de recursos sobrepasan por lo general el alcance de este proyecto. A día de hoy, este tipo de plataformas son objeto de numerosos estudios para su

desarrollo, y requieren de una tecnología puntera, como son el caso de los espejos deformables o de las esferas mencionadas en el apartado 2.3.2. Por lo tanto, se ha optado por una opción viable tanto en tiempo como en recursos, dentro del alcance de este Trabajo Fin de Grado, consistente en una plataforma orientable de tipo cardan. El diseño e implementación se han llevado a cabo con éxito ya que se ha conseguido que la plataforma opere según las instrucciones dadas por el microprocesador, a través de un microcontrolador, y por lo tanto cumpla con la función de orientar el módulo de comunicaciones. Podemos también destacar los estudios de optimización que se han hecho acerca de la plataforma, que son la calibración y la optimización de la velocidad. Mediante la inserción de ciertas variables en el código, se ha podido calibrar la plataforma para maximizar la precisión del sistema y corregir los posibles errores de desalineamiento entre los diferentes componentes del sistema. Además, la optimización de la velocidad permite reducir en gran medida una desventaja del sistema que es la velocidad de rotación de los motores, mejorando las prestaciones generales del mismo.

Sin embargo, cabe reseñar que el sistema no es perfecto, y necesita de un desarrollo posterior para llegar a ser un prototipo que sea operativo para los sistemas de comunicaciones navales. En los distintos estudios realizados se han mencionado igualmente cuales son las limitaciones del sistema, tanto en el sistema de seguimiento como en la plataforma. En concreto, en cuanto al sistema de seguimiento, podemos afirmar que hemos llegado al techo de la tecnología empleada, ya que la principal limitante viene del hecho de que tanto la cámara como el microprocesador operan en condiciones límite y se saturan fácilmente en condiciones adversas. Sin embargo, en el caso de la plataforma, existen dos limitantes, la principal heredada del microprocesador y de la velocidad de procesamiento, que es la velocidad de operación de los motores, y la segunda que es el diseño de la transmisión de movimiento en elevación, que necesitó de un reajuste adicional para poder operar pero que limita la velocidad de operación del motor.

Por lo tanto, podemos concluir que se ha cumplido satisfactoriamente con los objetivos especificados, ya que no solo se ha diseñado e implementado, sino que se ha estudiado y optimizado los dos sistemas objeto de este TFG. Aunque con ciertas limitantes, se ha conseguido un sistema funcional que permite cumplir la función de un sistema de adquisición y seguimiento de un sistema FSO, siendo además concebido para una aplicación de carácter naval.

5.2 Líneas futuras

En cuanto a los futuros desarrollos de este Trabajo Fin de Grado, cabe reseñar que en primer lugar se pueden realizar ciertos estudios e incorporar nuevas tecnologías que permitirían mejorar el sistema en sí. Además, se pueden implementar sistemas de adquisición y seguimiento alternativos que funcionen de manera complementaria al sistema desarrollado, creando por lo tanto un sistema híbrido, más versátil y robusto.

En primer lugar, como se ha mencionado en el apartado 3.1.3, para seguir optimizando el sistema en sí será necesario la realización de un estudio que indague acerca de cuál es el patrón más robusto (es decir con menor probabilidad de ser detectado erróneamente) y adecuado para su empleo en un sistema de adquisición y seguimiento de un sistema FSO en una plataforma naval. También se puede mejorar las prestaciones usando un sistema que en vez de emplear tratamientos de imágenes mediante una cámara que opera en el rango de frecuencias de la luz visible, procese imágenes tomadas con una cámara que opere en frecuencias del infrarrojo. En estas frecuencias el “ruido” de fondo es menor y por lo tanto mejorarían la probabilidad de detección del patrón, lo que se traduce en un aumento de la distancia y de la fiabilidad del sistema. Por otro lado, se puede evitar el problema de la falsa alarma mediante una retroalimentación del subsistema de comunicación al sistema de adquisición y seguimiento. En efecto, este porcentaje de falsa alarma puede verse casi anulado mediante una comunicación con la carga útil del sistema FSO, detectando cuando hay comunicación y cuando no. De esta manera, si se detecta un patrón y no hay comunicación, el sistema de adquisición pasa a buscar otro patrón, hasta que el enlace se forme.

Por otro lado, y como ya hemos mencionado en el apartado 5.1, el llegar al techo de la tecnología empleada en lo que al microprocesador se refiere hace que sea necesario la implementación de un procesador más potente. El núcleo del sistema es el tratamiento de imágenes, y toda limitante que tenga este se transmite al resto del sistema, sobre todo en cuanto a velocidad se refiere. Por lo tanto, cabe plantearse la sustitución del microprocesador empleado por otro con una velocidad de procesamiento mayor. De esta manera, se podrán emplear resoluciones mejores, con menores porcentajes de falsa, aumentando las distancias de operación, así como las velocidades de seguimiento. Cabe también reseñar que un rediseño de la parte motriz encargada de la elevación se requiere si se quiere aumentar la velocidad de operación en elevación. Debido a los problemas mencionados en el apartado 4.2.2, es necesario una reestructuración del sistema, teniendo en cuenta que los materiales obtenidos mediante impresión 3D son deformables y poco resilientes, lo cual hace que el sistema sea poco robusto mecánicamente. Una mejora significativa puede ser el uso de un metal como eje, cuyas propiedades son mucho más interesantes que las del plástico empleado.

Por último, y como suele ocurrir con los sistemas FSO, sobre todo en los de la superficie terrestre, sería muy interesante combinar este sistema de detección de un patrón de balizas luminosas con otro sistema de adquisición y seguimiento, de forma que se complementen. Este sistema mediante balizas puede llegar a ser extremadamente preciso, sobre todo teniendo en cuenta el alcance que tendría hoy en un día una FSOC, que no supera los 5 km en la superficie terrestre, sin tener en cuenta además que las condiciones marítimas suelen ser por lo general más perjudiciales para estos sistemas. Sin embargo, no es un sistema que permita gran velocidad de seguimiento, ya que este puede perturbar la detección del patrón y dificultar el seguimiento. Sin embargo, la incorporación de otro sistema permitiría solventar esas carencias, siendo el seguimiento por radiofrecuencia el candidato más favorable. En efecto, un sistema por radiofrecuencia permitiría una gran capacidad tanto de adquisición y readquisición, mejorando tanto tiempos como distancias, como una gran velocidad de seguimiento. Este sistema híbrido permitiría el establecimiento del enlace no solo con plataformas navales, pero con plataformas más dinámicas como son las terrestres e incluso las aéreas.

Por lo tanto, este sistema puede ser mejorado mediante la incorporación de nuevos elementos, así como la realización de ciertos estudios, pero alcanzaría su máxima operatividad con la combinación del sistema reseñado con un sistema de adquisición y seguimiento mediante radiofrecuencia, multiplicando las posibilidades de empleo del sistema.

6 BIBLIOGRAFÍA

- [1] J. Palfreman, Dirección, *Light Speed*. [Película]. National Geographic.
- [2] R. M. Dr. Estepa Alonso, «Tema 1: Evolución histórica de las telecomunicaciones,» de *Apuntes de la asignatura Arquitectura de Redes, Sistemas y Servicios*, 2004.
- [3] J. Á. Abeilhé, «Historia de la Bandera de España,» Ejército de Tierra, [En línea]. Available: <http://www.ejercito.mde.es/unidades/Madrid/ihycm/Actividades/Ciclos-Divulgativos/vexi-historia-bandera.html>.
- [4] «Origen de la Bandera Española,» Armada Española, [En línea]. Available: <http://www.armada.mde.es/ArmadaPortal/page/Portal/ArmadaEspañola/conocenos/especiales/prefLang-es/02dia-de-las-fuerzas-armadas--02dia-de-las-fuerzas-armadas-2017--01historia-armada#origen-de-la-bander-espanola>.
- [5] «Estación Morse fabricada por Lorenz,» antiguedades.es, [En línea]. Available: <https://www.antiguedades.es/telegrafos-vendidos/antiguo-telegrafo-o-estacion-morse-fabricada-por-lorenz-preciosa-1890>.
- [6] M. Bandera, «Perdenti 14: Antonio Meucci,» Missioni Consolata, 13 mayo 2016. [En línea]. Available: <http://www.rivistamissioniconsolata.it/2016/05/13/perdenti-14-antonio-meucci/>.
- [7] G. Garratt, *The Early History of Radio from Faraday to Marconi*, Londres: The Institution of Engineering and Technology, 1994.
- [8] R. A. Cacheda, *Apuntes de la asignatura Fundamentos de Redes de Ordenadores*, 2018.
- [9] Dr. Núñez Ortuño, «Tema 6: Sistemas de Telecomunicaciones actuales,» de *Apuntes de la asignatura Sistemas de Telecomunicaciones*, 2017.
- [10] T. Garlington, J. MAJ Babbitt y G. Long, «Analysis of Free Space Optics as a Transmission Technology,» U.S. Army Information Systems Engineering Command (USAISEC), Transmission Systems, 2005.
- [11] «Las comunicaciones modernas: la revolución del láser y la fibra óptica,» The National Academy of Sciences, [En línea]. Available:

- http://www7.nationalacademies.org/spanishbeyonddiscovery/tec_007520-05.html.
- [12] A. Gómez, K. Shi, C. Quintana, M. Sato, G. Faulkner, B. C. Thomsen y D. O'Brien, «Beyond 100-Gb/s Indoor Wide Field-of-View Optical Wireless Communications,» IEEE Photonics Technology Letters, 2015.
 - [13] H. Haas, «Comprehensive Summary of Modulation Techniques for Lifi,» University of Edinburgh.
 - [14] Y. Kaymak, R. Rojas-Cessa, J. Feng, N. Ansari, M. Zhou y T. Zhang, «Survey on Acquisition, Tracking, and Pointing Mechanisms for Mobile Free-Space Optical Communications,» IEEE Communications surveys and tutorials, 2018.
 - [15] L. Rapaport, R. Yozevitch y B. Ben-Moshe, «Ad Hoc FSO Communication,» Kinematics & Computational Geometry lab. Ariel University, Israel, 2017.
 - [16] P. Singhal, P. Gupta y P. Rana, «Basic Concept of Free Space Optics Communications (FSO): An Overview,» IEEE ICCSSP, 2015.
 - [17] H. A. Willebrand y B. S. Ghuman, «Fiber optics without fiber,» IEEE Spectrum, 2001.
 - [18] P. Gómez-Esteban, «[Mecánica de fluidos I] Turbulencia,» El Tamiz, 10 octubre 2013. [En línea]. Available: <https://eltamiz.com/2013/10/10/mecanica-de-fluidos-turbulencia/>.
 - [19] H. Zell, «Earth's Atmospheric Layers,» NASA, 7 Agosto 2017. [En línea]. Available: https://www.nasa.gov/mission_pages/sunearth/science/atmosphere-layers2.html. [Último acceso: 11 Febrero 2019].
 - [20] «Another world first for Artemis: a Laser link with an aircraft,» 18 Diciembre 2006. [En línea]. Available: http://www.esa.int/Our_Activities/Telecommunications_Integrated_Applications/Another_world_first_for_Artemis_a_laser_link_with_an_aircraft. [Último acceso: 11 Febrero 2019].
 - [21] «Lunar Laser Communication Demonstration, NASA's first space Laser Communication System Demonstration,» 2013. [En línea]. Available: https://www.nasa.gov/sites/default/files/llcdfactsheet.final_.web_.pdf. [Último acceso: 11 Febrero 2019].
 - [22] «Lunar Laser Communication Demonstration, NASA First Space Laser Communication System Demonstration,» Nautical Aeronautics and Space Administration, 2013.
 - [23] «European Data Relay Satellite System (EDRS) Overview,» European Space Agency, [En línea]. Available: <https://artes.esa.int/edrs/overview>.
 - [24] «Tesat Laser Portfolio,» Tesat-Spacecom GmbH & Co, 2019. [En línea]. Available: <http://www.tesat.de/en/laser>.
 - [25] M. J. Abrahamson, O. V. Sindiy, B. V. Oaida, S. Fregoso, J. N. Bowles-Martínez, M. Kokorowski y M. W. Wilkerson, «OPALS Mission System Operations Architecture for an Optical Communications Demonstration on the ISS,» Jet Propulsion Laboratory, California Institute of Technology., Pasadena, California, 2014.
 - [26] «Laser Light Global Limited, Executive Summary,» Laser Light, 2018. [En línea]. Available: https://docs.wixstatic.com/ugd/2f1301_c206daba55294b15a1cfa26a3f7d1819.pdf.
 - [27] «LTE (telecomunicaciones),» Wikipedia, [En línea]. Available:

- [https://es.wikipedia.org/wiki/LTE_\(telecomunicaciones\)](https://es.wikipedia.org/wiki/LTE_(telecomunicaciones)).
- [28] «Loon,» Google X, 2011. [En línea]. Available: <https://loon.co>.
- [29] I. Kim, T. Chaffee, R. Fleishauer, P. Szajowski, A. Braga, D. Beering y D. C. Hurley, «Advances in communications: New FSO provides reliable 10 Gbit/s and beyond backhaul connections,» *Laser Focus World*, 14 Octubre 2013. [En línea]. Available: <https://www.laserfocusworld.com/articles/print/volume-49/issue-10/features/advances-in-communications-new-fso-provides-reliable-10-gbit-s-and-beyond-backhaul-connections.html>.
- [30] D. Pérez, «Como saber si hay cobertura de datos 4G donde vivo, y qué compañía es mejor,» *El Español*, 15 enero 2016. [En línea]. Available: <https://elandroidelibre.elespanol.com/2016/01/cobertura-4g-3g-espana.html>.
- [31] Y. Maguire, «Building communications networks in the stratosphere,» Facebook Connectivity Lab, 30 Julio 2015. [En línea]. Available: <https://code.fb.com/connectivity/building-communications-networks-in-the-stratosphere/>.
- [32] A. Cox, Interviewee, *Inside Facebook's Conncetivity Lab*. [Entrevista]. 30 Julio 2015.
- [33] M. Zuckerberg, «The technology behind Aquila,» Facebook, 21 Julio 2016. [En línea]. Available: <https://www.facebook.com/notes/mark-zuckerberg/the-technology-behind-aquila/10153916136506634>.
- [34] Y. Maguire, «High altitude connectivity: the next chapter,» Facebook Connectiviity Lab, 27 Junio 2018. [En línea]. Available: <https://code.fb.com/connectivity/high-altitude-connectivity-the-next-chapter/>.
- [35] «Zephyr,» Airbus, [En línea]. Available: <https://www.airbus.com/defence/uav/zephyr.html#capabilities>.
- [36] R. Peach, C. Visone, G. Burdge, J. Vickers, T. Leclerc, P. Sauer, L. Andrews, R. Phillips, J. E. Valencia y J. Kiriazes, «Performance of a 10 Gbps FSO System Implementing Novel Beam Tracking and,» Space Photonics Inc., Kennedy Space Center Florida, 2018.
- [37] LigthPointe Communications Inc., «Aire X-Stream Series, Free Space Optics, Model X-FSO,» 2018. [En línea]. Available: <https://nebula.wsimg.com/d874e04cdeef31ed313763787104058d?AccessKeyId=C1431E109BF92B03DF85&disposition=0&alloworigin=1>.
- [38] «Artolink products,» Artolink, [En línea]. Available: http://artolink.com/page/products/free_space_optics_Artolink_10Gbps/.
- [39] CableFree: Wireless Excellence, «Free Space Optics (FSO), CableFree Gigabit range, Overview,» 2018. [En línea]. Available: <https://www.cablefree.net/pdf/CableFree%20FSO%20Gigabit%20Datasheet.pdf>.
- [40] Freedom Photonics LLC, «FP8100 Free Space Communications Link,» 2018. [En línea]. Available: https://freedomphotonics.com/wp-content/uploads/2018/03/Data_Sheets_Free_Space_Optical_Communications_Link-FP8100.pdf.
- [41] EC System, «EL-10G,» 2018. [En línea]. Available: http://www.ecsystem.cz/ec_system/download/el-10g.pdf.

- [42] EC System, «EL-1GL,» 2018. [En línea]. Available: http://www.ecsystem.cz/ec_system/download/el-1gl.pdf.
- [43] «Ronja,» Twibright Laboratories, 2001. [En línea]. Available: <http://ronja.twibright.com/about.php>.
- [44] P. Daukantas, «Optical Wireless Communications: The New "Hot Spots"?», *Optics and photonics news*, p. 8, 2014.
- [45] H. Haas, L. Yin, Y. Wang y C. Chen, «What is LiFi?», *Journal of lightwave technology*, 2016.
- [46] pureLiFi, [En línea]. Available: <https://purelifi.com/>.
- [47] M. Sufyan Islim, R. X. Ferreira, X. He, E. Xie, S. Videv, S. Viola, S. Watson, N. Bamiedakis, R. V. Penty, I. H. White, A. E. Kelly, E. Gu, H. Haas y M. D. Dawson, «Towards 10 Gb/s orthogonal frequency division multiplexing-based visible light communication using a GaN violet micro-LED,» *Photonics Research*, vol. 5, n° 2, p. 9, 2017.
- [48] Department of Defense, «2005 Budget Estimates: Research, Development, Test and Evaluation, Defense-wide,» 2004.
- [49] Department of Defense, «2012 Budget Estimates: Research, Development, Test & Evaluation; Defense-Wide,» 2011.
- [50] Department of Defense, «2013 President's Budget Submission: Research, Development, Test & Evaluation, Defense-Wide,» 2012.
- [51] C. Reynolds, «Exelis completes US Naval Research Laboratory evaluation of high-speed laser-based communications technology,» Exelis Inc., 25 Noviembre 2013. [En línea]. Available: <https://www.harris.com/press-releases/2013/11/exelis-completes-us-naval-research-laboratory-evaluation-of-high-speed-laser>.
- [52] T. Nguyen y K. Cahoy, «Laser Beacon Tracking for Free-space Optical Communication on Small-Satellite Platforms in Low-Earth Orbit,» Massachusetts Institute of Technology, 2015.
- [53] J. Wang, J. M. Kahn y K. Y. Lau, «Minimization of acquisition time in short-range free-space optical communication,» Optical Society of America, 2002.
- [54] «Sensor CMOS,» Wikipedia, [En línea]. Available: https://es.wikipedia.org/wiki/Sensor_CMOS.
- [55] «Position Sensing Detectors / Quadrant Position Detectors,» Thorlabs Inc., [En línea]. Available: https://www.thorlabs.com/NewGroupPage9.cfm?ObjectGroup_ID=4400.
- [56] «Quadrant PIN photodiodes (QP),» First Sensor AG, [En línea]. Available: <https://www.first-sensor.com/en/products/optical-sensors/detectors/quadrant-pin-photodiodes-qp/>.
- [57] «Picamera documentation,» Picamera, [En línea]. Available: <https://picamera.readthedocs.io/en/release-1.12/>.
- [58] «Raspberry Pi,» Raspberry Pi Foundation, [En línea]. Available: <https://www.raspberrypi.org/>.
- [59] «Arduino Uno Rev3,» Arduino, [En línea]. Available: <https://store.arduino.cc/arduino->

uno-rev3.

- [60] T. Hirzel, «PWM,» Arduino, [En línea]. Available: <https://www.arduino.cc/en/Tutorial/PWM>.
- [61] A. Sperduti, «Tutorial Motor Paso a Paso - Stepper Motor,» ArS Electrónica, 19 abril 2015. [En línea]. Available: <https://blog.ars-electronica.com.ar/2015/04/tutorial-stepper-motor-paso-a-paso.html>.
- [62] J. Sun, M. Watts, A. Yaacobi y E. Timurdogan, «Optical Phased Arrays». Massachusetts Institute of Technology. Massachusetts, EEUU. Patente WO 2014/110017 A1, 17 Julio 2014.
- [63] «Narinder Singh Kapany Chair in Opto-electronics,» University of California Santa Cruz, [En línea]. Available: <https://southasia.ucsc.edu/endowed-chairs/narinder-singh-kapany.html>.
- [64] Space Photonics Inc., «LaserFire, Free Space Optical Communications Terminal,» 2018. [En línea]. Available: http://spacephotonics.com/pdfs/data_sheets/Free_Space_Optics_Information_flyer_2012.pdf.
- [65] D. Cuartielles y M. Banzi, «Interfacing a Joystick,» Arduino, [En línea]. Available: <https://www.arduino.cc/en/Tutorial/JoyStick>.

ANEXO I: CÓDIGO DEL MICROPROCESADOR

```
import picamera
import numpy as np
from picamera.array import PiYUVAnalysis
from picamera.color import Color
from time import sleep
import serial

def AnhadirMatriz(obj,elem):
    a = np.zeros((len(obj)*2-2)).astype(int)
    elem = np.append(a,elem).reshape((len(obj),2))
    obj = np.append(obj,elem,axis=0)
    n = obj
    inds = np.any(n != 0, axis=1)
    n_zero = n[inds, :]
    return (n_zero)

def CompararValor(obj,elem):
    tol = 3
    x = np.zeros((1)).astype(int)
    x1 = np.argwhere(obj[:,1] <= elem[1]+tol)
    for x2 in x1:
        if obj[x2,1] >= elem[1]-tol:
            x = np.append(x,x2,axis = 0).astype(int)
    x = x[1:]
    y = np.zeros((1))
    y1 = np.argwhere(obj[:,0] <= elem[0]+tol)
    for y2 in y1:
        if obj[y2,0] >= elem[0]-tol:
            y = np.append(y,y2,axis = 0)
    y = y[1:].astype(int)
    return([x,y])

def PatronCuadrado (Xm,Ym):
    X = []
    Y = []
    tol = 7
    I = []
    for i, x1 in enumerate(Xm):
        for j, x2 in enumerate(Xm):
            if x1-tol <= x2 <= x1+tol:
                if X.count(x2) != Xm.count(x2):
                    X.append(x2)
                    Y.append(Ym[j])
    for k, y1 in enumerate(Y):
        contador1 = 0
        for l, y2 in enumerate(Y):
            if y1-tol <= y2 <= y1+tol:
                contador1 += 1
        if contador1 == 0:
            I.append(k)
    contador2 = 0
    for l in I:
        X.pop(l-contador2)
        Y.pop(l-contador2)
        contador2 += 1
    if len(Y) > 1:
        for m, n in enumerate(X):
```

```
for o, p in enumerate(X):
    if p > n:
        difx = p-n
        dify = abs(Y[o]-Y[m])
        if dify-tol <= difx <= dify+tol:
            contador3 = 0
            for q, r in enumerate(X):
                if n-tol <= r <= n+tol:
                    if Y[o]-tol <= Y[q] <= Y[o]+tol:
                        contador3 += 1
                if p-tol <= r <= p+tol:
                    if Y[m]-tol <= Y[q] <= Y[m]+tol:
                        contador3 += 1
            if contador3 >= 2:
                XMED = (n+p)/2
                YMED = (Y[m]+Y[o])/2
                return (XMED, YMED)

class MiAnalizador (PiYUVAnalysis):
    def __init__(self, camara):
        super (MiAnalizador, self). __init__ (camara)

    def analyze (self, a):
        M = []
        Xm = []
        Ym = []
        umbral = 180
        Y = a[:, :, 0]
        XRES = 193
        YRES = 90
        A = np.argwhere(Y >= umbral)
        if A.shape != (0, 2) :
            M.append(np.array(A[0]).reshape(1, 2))
            A = np.delete(A, 0, 0)
            while A.shape != (0, 2):
                contador = 0
                for i, obj in enumerate(M):
                    obj = np.array(obj)
                    elem = A[0]
                    elem = np.array(elem)
                    [x, y] = CompararValor(obj, elem)
                    if x.shape != (0,) and y.shape != (0,):
                        M[i] = AnhadirMatriz(M[i], elem)
                        A = np.delete(A, 0, 0)
                        contador += 1
                    break
                if contador == 0:
                    M.append(np.array(A[0]).reshape(1, 2))
                    A = np.delete(A, 0, 0)
            for i, obj in enumerate(M):
                Xmed = int(np.mean(obj[:, 1]))
                Ymed = int(np.mean(obj[:, 0]))
                Xm.append(Xmed)
                Ym.append(Ymed)
            if len(Xm) != 0:
                B = PatronCuadrado(Xm, Ym)
                if B != None:
                    [X, Y] = B
                    print(X, ' ', Y)
                    if X < XRES/2-1 and Y > YRES/2:
                        port.write(("q".encode()))
                    elif XRES/2-1 <= X <= XRES and Y > YRES/2:
                        port.write(("w".encode()))
```

```
elif X > XRES/2 and Y > YRES/2:
    port.write(("e".encode()))
elif X > XRES/2 and YRES/2-1 <= Y <= YRES/2:
    port.write(("d".encode()))
elif X > XRES/2 and Y < YRES/2-1:
    port.write(("c".encode()))
elif XRES/2-1 <= X <= XRES/2 and Y < YRES/2-1:
    port.write(("x".encode()))
elif X < XRES/2-1 and Y < YRES/2-1:
    port.write(("z".encode()))
elif X < XRES/2 and YRES/2-1 <= Y <= YRES/2:
    port.write(("a".encode()))

with picamera.PiCamera(resolution='160x120', framerate=60) as camara:
    with MiAnalizador(camara) as analizador:
        port = serial.Serial('/dev/ttyACM0', 9600)
        camara.start_preview(alpha=128)
        sleep(2)
        camara.start_recording(analizador, 'yuv')
        try:
            while True:
                camara.wait_recording(1)
        finally:
            camara.stop_recording()
```


ANEXO II: CÓDIGO DEL MICROCONTROLADOR

```
#include <Stepper.h>

Stepper stepper1(32,4,6,5,7);
Stepper stepper2(32,8,10,9,11);
int value0;
int value1;
char letter;

void JoystickLogic() {
    value0 = analogRead(A0);
    value1 = analogRead(A1);
    if(value1 <= 400) {
        stepper1.step(-1);
    }
    if(value1 >= 600) {
        stepper1.step(1);
    }
    if(value0 <= 400) {
        stepper2.step(1);
    }
    if(value0 >= 600) {
        stepper2.step(-1);
    }
}

void SerialLogic(char letter) {
    switch(letter) {
        case 'w':
            stepper2.step(-1);
            break;
        case 'x':
            stepper2.step(1);
            break;
        case 'a':
            stepper1.step(5);
            break;
        case 'd':
            stepper1.step(-5);
            break;
        case 'q':
            stepper1.step(5);
            stepper2.step(-1);
            break;
        case 'e':
            stepper1.step(-5);
            stepper2.step(-1);
            break;
        case 'z':
            stepper1.step(5);
            stepper2.step(1);
            break;
        case 'c':
            stepper1.step(-5);
            stepper2.step(1);
            break;
    }
}

void setup() {
```

```
    stepper1.setSpeed(900);  
    stepper2.setSpeed(900);  
    Serial.begin(9600);  
}  
  
void loop() {  
    if(Serial.available() > 0) {  
        letter = Serial.read();  
        SerialLogic(letter);  
    } else {  
        JoystickLogic();  
    }  
}
```

ANEXO III: CÓDIGO DE MUESTRO DE LA LUMINOSIDAD

Este código es una adaptación del código de muestro en [57] para medir los valores de intensidad mediante el formato YUV. El código hace la media del valor Y de una parcela determinada de píxeles, y lo expone en pantalla (en el “*preview*”). Para ello, toma los valores de los píxeles en formato RGB y los transforma mediante una matriz de conversión. A parte de la luminosidad, el código también imprime los tres valores de la parcela del formato RGB, así como los tres del formato YUV.

```
import picamera
import numpy as np
from picamera.array import PiYUVAnalysis

class MiAnalizador(PiYUVAnalysis):
    def __init__(self, camera):
        super(MiAnalizador, self).__init__(camara)

    def analyze(self, a):
        # Convert the average color of the pixels in the middle box
        Ymed = int(np.mean(a[43:47, 78:82, 0]))
        Umed = int(np.mean(a[43:47, 78:82, 1]))
        Vmed = int(np.mean(a[43:47, 78:82, 2]))

        YUVm = [Ymed, Umed, Vmed]

        Y = a[:, :, 0]
        U = a[:, :, 1]
        V = a[:, :, 2]

        camara.annotate_text = str(Ymed)

        YUV = np.dstack((Y, U, V))[:90, :160, :].astype(np.float)
        YUV[:, :, 0] = YUV[:, :, 0] - 16 # Offset Y by 16
        YUV[:, :, 1:] = YUV[:, :, 1:] - 128 # Offset UV by 128
        # YUV conversion matrix from ITU-R BT.601 version (SDTV)
        #
        #      Y      U      V
        M = np.array([[1.164, 0.000, 1.596], # R
                     [1.164, -0.392, -0.813], # G
                     [1.164, 2.017, 0.000]]) # B

        # Take the dot product with the matrix to produce RGB output, clamp the
        # results to byte range and convert to bytes
        RGB = YUV.dot(M.T).clip(0, 255).astype(np.uint8)
        R = int(np.mean(RGB[43:47, 78:82, 0]))
        G = int(np.mean(RGB[43:47, 78:82, 1]))
        B = int(np.mean(RGB[43:47, 78:82, 2]))
        col = [R, G, B]
        print (col[:], ' ', YUVm[:])

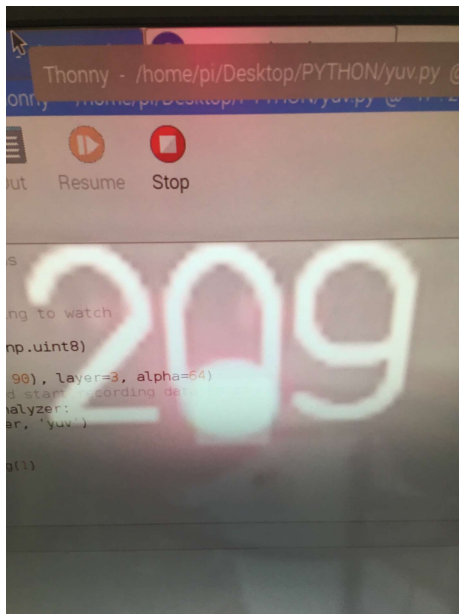
with picamera.PiCamera(resolution='160x90', framerate=20) as camara:
    # Fix the camera's white-balance gains
    camara.awb_mode = 'off'
    camara.awb_gains = (1.4, 1.5)
    # Draw a box over the area we're going to watch
    camara.start_preview(alpha=128)
```

```

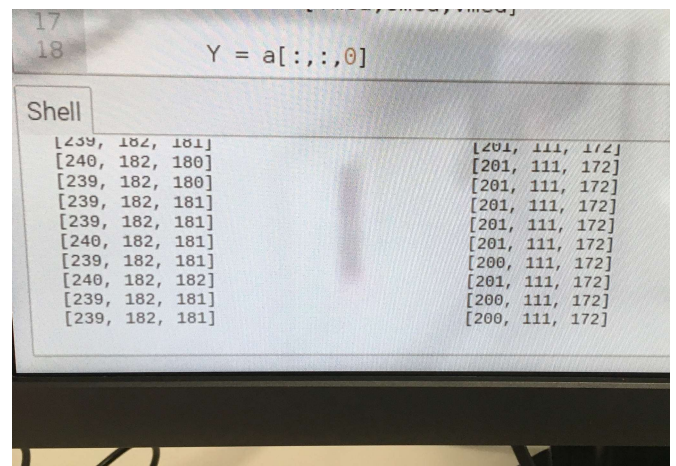
box = np.zeros((96, 160, 3), dtype=np.uint8)
box[43:47, 78:82, :] = 0x80
camara.add_overlay(box, size=(160, 90), layer=3, alpha=64)
# Construct the analysis output and start recording data to it
with MiAnalizador(camara) as analyzer:
    camara.start_recording(analyzer, 'yuv')
    try:
        while True:
            camara.wait_recording(1)
    finally:
        camara.stop_recording()

```

Un ejemplo del resultado que da este código se puede ver en la Figura Anexo III-1 (a), donde podemos ver la media de los valores Y de la parcela en blanco, en grande. Además, el código imprime en la “Shell” la media de los valores RGB y YUV, como se puede apreciar en la Figura Anexo III-1(b).



(a)



(b)

Figura Anexo III-1 (a) Valor medio de Y en la parcela en blanco, (b) Valor medio de los valores RGB y YUV de la parcecla en blanco