

# Diseño de la arquitectura funcional de control para un robot móvil de exploración espacial

Grado en Ingeniería Telemática



**Universidad Carlos III de Madrid**

Escuela Politécnica Superior

**Autora:** Naiara Escudero Sánchez

**Director:** Luis Moreno Lorente

**Tutor:** Alberto Valero Gómez

Curso 2011 - 2012



# Índice general

<b>1. Introducción.....</b>	<b>9</b>
1.1 Marco general .....	9
1.2 Objetivos principales .....	10
1.3 Estado del arte .....	11
1.3.1 Robots imprimibles .....	11
1.3.2 Localización mediante tecnología infrarroja .....	14
1.4 Organización de la memoria .....	16
<b>2. Análisis del problema .....</b>	<b>17</b>
2.1 Análisis de las funcionalidades requeridas .....	17
2.1.1. Localización del robot .....	18
2.1.2. Control de desplazamiento .....	30
2.1.3. Comunicación robot – base .....	33
2.2 Soluciones Hardware .....	38
2.2.1. Estructura del robot imprimible .....	38
2.2.2. Actuadores .....	44
2.2.3. SkyMega .....	46
2.2.4. Emisor infrarrojo .....	48
2.2.5. Receptor infrarrojo .....	50
2.2.6. Módulo Bluetooth .....	52
2.2.7. Acelerómetro .....	53
2.3 Soluciones Software .....	54

2.3.1. Arduino .....	54
2.3.2. OpenRDK .....	55
<b>3. Diseño de la solución .....</b>	<b>57</b>
3.1 Diseño Hardware .....	57
3.2 Diseño Software .....	59
3.2.1. Programación de la placa microcontroladora .....	60
3.2.2. Programación de los módulos de la aplicación .....	61
<b>4. Resultados y conclusiones.....</b>	<b>72</b>
<b>Referencias .....</b>	<b>74</b>

## Índice de figuras

<b>Figura 1.1:</b> Robot Curiosity .....	10
<b>Figura 1.2:</b> Ilustración de la impresora 3D denominada Darwin .....	12
<b>Figura 1.3:</b> Ilustración del robot Miniskybot .....	13
<b>Figura 1.4:</b> Ejemplos de robots imprimibles. (a) Orugator. (b) F-Track .....	13
<b>Figura 1.5:</b> (a) terrestre, (b) tierra-aire, (c) entre dispositivos de computo, (d) tierra-satélite, (e) aire-submarino, (f) Inter-satelital .....	15
<b>Figura 2.1:</b> Esquema general de una comunicación Transpondedor -Lector .....	18
<b>Figura 2.2:</b> Espectro electromagnético .....	20
<b>Figura 2.3:</b> Onda senoidal .....	21
<b>Figura 2.4:</b> Clasificación de los sistemas IR .....	23
<b>Figura 2.5:</b> Esquema general del funcionamiento de las ondas de ultrasonido ..	24
<b>Figura 2.6:</b> Detección de objetos utilizando sensores infrarrojos .....	25
<b>Figura 2.7:</b> Esquema general de una fuente de luz y un fotodetector .....	26
<b>Figura 2.8:</b> Distribución de los satélites que dan soporte a la tecnología GPS ...	27
<b>Figura 2.9:</b> Funcionamiento de la brújula atendiendo al magnetismo terrestre .	27
<b>Figura 2.10:</b> Esquema general de un inclinómetro .....	28
<b>Figura 2.11:</b> Ejes de giro de un giroscopio electrónico .....	29
<b>Figura 2.12:</b> Esquema de funcionamiento de un <i>encoder</i> óptico .....	29
<b>Figura 2.13:</b> Ejes de referencia utilizados por el acelerómetro .....	30
<b>Figura 2.14:</b> Esquema general de un motor neumático .....	31
<b>Figura 2.15:</b> Esquema general de un cilindro hidráulico .....	32
<b>Figura 2.16:</b> Esquema general de un motor eléctrico .....	32
<b>Figura 2.17:</b> Clasificación de frecuencias dentro del espectro electromagnético.	33

<b>Figura 2.18:</b> Comparación de tipos de comunicaciones inalámbricas .....	35
<b>Figura 2.19:</b> Pila de protocolos Bluetooth .....	37
<b>Figura 2.20:</b> Configuración diferencial. (a) Estructura. (b) Robot Quiróptero ....	38
<b>Figura 2.21:</b> Configuración triciclo. (a) Estructura. (b) Robot de configuración triciclo .....	39
<b>Figura 2.22:</b> Configuración Ackerman. (a) Estructura. (b) Robot de configuración Ackerman .....	39
<b>Figura 2.23:</b> Configuración sincronizada .....	40
<b>Figura 2.24:</b> Configuración omnidireccional. (a) Estructura. (b) Robot Rovio ....	41
<b>Figura 2.25:</b> Robot Rovers .....	41
<b>Figura 2.26:</b> Estructura de tracción oruga .....	41
<b>Figura 2.27:</b> Estructura básica del robot elegido para el proyecto .....	42
<b>Figura 2.28:</b> Representación del modelo diferencial en situación de giro .....	43
<b>Figura 2.29:</b> Actuador FUTABA S3003 .....	44
<b>Figura 2.30:</b> Pulsos de control del actuador FUTABA S3003 .....	45
<b>Figura 2.31:</b> Relación velocidad-ángulo proporcionada por los actuadores .....	46
<b>Figura 2.32:</b> SkyMega .....	47
<b>Figura 2.33:</b> Procesos de emisión espontánea y estimulada .....	49
<b>Figura 2.34:</b> IR383 (GaAlAs) .....	50
<b>Figura 2.35:</b> Proceso de absorción .....	51
<b>Figura 2.36:</b> Wiimote .....	52
<b>Figura 2.37:</b> Módulo BT transceptor RS232 Arduino .....	53
<b>Figura 2.38:</b> Acelerómetro ADXL345 .....	54
<b>Figura 2.39:</b> Interfaz gráfica para la programación de Arduino .....	55
<b>Figura 2.40:</b> Ejemplo de aplicación implementada a partir de OpenRDK .....	56
<b>Figura 3.1:</b> Circuito implementado para el acoplamiento de los leds .....	58

<b>Figura 3.2:</b> Estructura final del robot desde diferentes perspectivas .....	59
<b>Figura 3.3:</b> Esquema general del funcionamiento sensores-actuadores .....	59
<b>Figura 3.4:</b> Correspondencia entre velocidades continuas y discretas .....	61
<b>Figura 3.5:</b> Esquema general de los módulos .....	62
<b>Figura 3.6:</b> Esquema de la detección de los emisores infrarrojos. (a) Vista superior. (b) Vista lateral .....	62
<b>Figura 3.7:</b> Diagrama de flujo del módulo WiilocIntModule .....	63
<b>Figura 3.8:</b> Representación del cálculo de las distancias entre los leds .....	64
<b>Figura 3.9:</b> Representación de la estimación del punto medio de la base del robot .....	64
<b>Figura 3.10:</b> Representación del cálculo del ángulo de orientación del robot ....	65
<b>Figura 3.11:</b> Diagrama de flujo del módulo AccelerometerModule .....	65
<b>Figura 3.12:</b> Diagrama de flujo del módulo LocalizationModule .....	66
<b>Figura 3.13:</b> Ángulos existentes entre los puntos origen y destino .....	67
<b>Figura 3.14:</b> Representación del caso $\alpha = \beta$ .....	67
<b>Figura 3.15:</b> Representación del caso $\alpha \leq 180, \beta \leq 180$ .....	68
<b>Figura 3.16:</b> Representación del caso $\alpha \geq 180, \beta \geq 180$ .....	68
<b>Figura 3.17:</b> Representación del caso $\alpha \leq 180, \beta \geq 180$ .....	69
<b>Figura 3.18:</b> Representación del caso $\alpha \geq 180, \beta \leq 180$ .....	69
<b>Figura 3.19:</b> Diagrama de flujo del módulo ControlItModule .....	70
<b>Figura 3.20:</b> Diagrama de flujo del módulo ClientModule .....	71



# **1. Introducción**

Desde la antigüedad, el espacio ha sido uno de los principales objetos de estudio de los grandes pensadores de cada época. Científicos como Pitágoras, Aristarco, Copérnico o Galileo centraron sus esfuerzos en explicar la configuración de ese llamado Universo que nos rodea. Este afán por conocer y explorar el espacio, junto con todo aquello de lo que está compuesto, se ha mantenido a lo largo de los tiempos, con la diferencia de que el desarrollo de las tecnologías ha hecho posible avances que aquellos primeros observadores nunca hubiesen podido imaginar.

Desde 1957, fecha en la que fue lanzado el primer satélite de exploración de la órbita terrestre, la complejidad de las misiones espaciales ha crecido exponencialmente, así como la tecnología usada en cada una de ellas y los objetivos marcados en cada incursión. Los simples satélites de exploración han evolucionado a inmensas estaciones espaciales diseñadas para albergar vida durante meses o complejas naves creadas para la exploración de otros cuerpos del Sistema Solar. Tanto la tecnología como las ambiciones aeroespaciales se encuentran en continuo crecimiento, lo que hace que en la actualidad siga siendo uno de los mayores campos de estudio e inversión.

## **1.1 Marco general**

A pesar de que ya se pensase en la exploración del espacio desde tiempos remotos, hubo que esperar al desarrollo de potentes motores capaces de impulsar vehículos fuera de la atmósfera, hecho que no sucedió hasta la Segunda Guerra Mundial. Es en 1942 cuando se consiguió por primera vez llegar a los 38 km de altura, bordeando los límites de la atmósfera, mediante la utilización de motores cohete desarrollados por Alemania. Estos cohetes fueron copiados tanto por Rusia como por EEUU, que tras la finalización del conflicto emprendieron una carrera por la conquista del espacio. Fue la URSS quien tomó la delantera con las primeras incursiones, enviando el primer animal a la órbita y, más tarde, el primer hombre (1961 Yuri Gagarin) a bordo de la Vostok I. La carrera espacial entre la URSS y los USA se convirtió en un asunto de primer orden político. Finalmente, el 21 de julio de 1969, Neil Armstrong, astronauta del Apollo 11, pisó la Luna, consiguiendo la gran victoria simbólica de llegar a la Luna.

A partir de la década de los 1960 comenzaron a enviarse sondas para la exploración de otros planetas del Sistema Solar. Así en 1964 se lanzaron las

primeras sondas a Marte. Además de Marte, en 1972 la NASA envió la sonda Pioneer 10 con destino a Júpiter y en 1989 la sonda Magallanes para la exploración de Venus.

Uno de los acontecimientos más importantes de la exploración espacial llegó en 1997 cuando la Mars Global Surveyor realizó unas fotografías de la superficie marciana que podrían sugerir la existencia de agua líquida en la superficie de Marte. Para su confirmación, en 2004 la NASA envió dos exploradores gemelos: Spirit y Opportunity, que además de confirmar los vestigios de agua, detectaron un meteorito ajeno a Marte sobre su superficie. [1]

En la actualidad se continúa explorando la superficie marciana a través del explorador Curiosity, que llegó a Marte en agosto de 2012 y está encargado de caracterizar la geología general del lugar mediante el análisis de los materiales superficiales y cercanos a la superficie e interpretar los procesos de formación de rocas y suelos. Al mismo tiempo, y continuará con la búsqueda de los componentes esenciales de la vida, el agua y el carbono.



Figura 1.1: Robot Curiosity.

## **1.2 Objetivos principales**

El objetivo principal de este estudio es desarrollar un sistema de navegación listo para ser integrado en un robot de exploración de exteriores. Para ello, será necesario implementar un mecanismo capaz de estimar la posición en tiempo real del robot explorador en un entorno de tres dimensiones (3D), que simulará una superficie como podría ser la de cualquier cuerpo del Sistema Solar. Para ello se van a combinar los datos proporcionados por varios dispositivos capaces de recoger información sobre la localización de objetos, cada uno de ellos

implementado por distintas tecnologías, con el fin de obtener una estimación precisa de la posición del robot.

Además, se implementará un algoritmo de control de movimiento que permitirá al robot alcanzar de forma autónoma un punto destino de la superficie a partir de las coordenadas locales ligadas a dicho punto. Este control se llevará a cabo por medio de un algoritmo que permitirá modificar su comportamiento dependiendo de los requerimientos de cada situación, pudiendo proporcionar movimientos más precisos para trazar una trayectoria ideal en términos de longitud recorrida, o movimientos más rápidos, que permitirán al robot alcanzar el punto destino en el menor tiempo posible, al precio de recorrer una distancia mayor. Dado que el robot establece una comunicación inalámbrica con la base, será posible variar en todo momento el punto destino a alcanzar de forma remota.

## **1.3 Estado del arte**

### **1.3.1 Robots imprimibles**

La robótica ha sufrido en los últimos años un gran periodo de expansión debido a su alto potencial en cuanto a aplicaciones se refiere. La principal limitación de esta tecnología viene dada por el elevado coste de los recursos utilizados y las complejas tareas de fabricación que, además, requieren de largos tiempos de espera. Por lo tanto, el desarrollo de esta tecnología determinó la necesidad de recurrir a materiales y procesos de fabricación alternativos capaces de reducir estas limitaciones. En este marco surgieron las impresoras 3D, impresoras de prototipado rápido y bajo coste que permiten la impresión de componentes robóticos diseñados previamente mediante aplicaciones software. Estos dispositivos ofrecen la posibilidad de diseñar e imprimir piezas propiamente diseñadas de una forma fácil y rápida, además de promover el intercambio de diseños mediante archivos software entre los integrantes de esta comunidad.

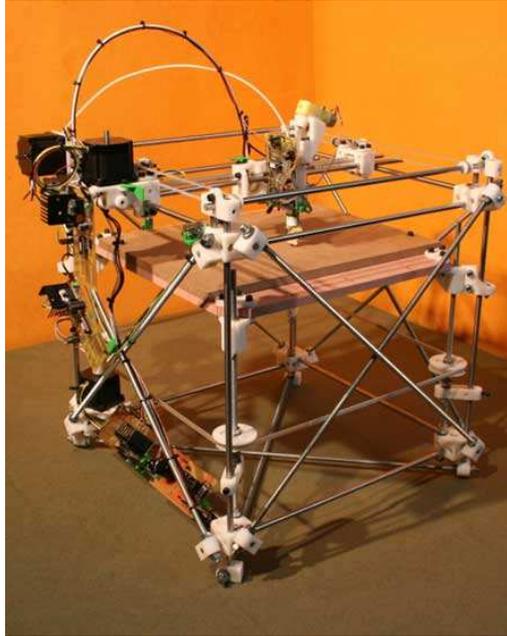


Figura 1.2: Ilustración de la impresora 3D denominada Darwin.

El origen de este tipo de impresoras se remonta al proyecto *RepRap* [2] iniciado por Adrian Bowyer en el año 2004 con el objetivo de desarrollar una máquina autoreplicante de código abierto. En mayo de 2007 el primer prototipo, llamado Darwin fue terminado y algunos días más tarde, el 29 de mayo se logró la primera replicación. Desde entonces, la comunidad *RepRap* (máquinas originales y diseños derivados) ha crecido exponencialmente [3] con una población estimada actual de alrededor de 4.500 máquinas. Inicialmente, estas impresoras no fueron diseñadas para el público en general, sino para personas con conocimientos técnicos. La evolución del proyecto *RepRap* hacia líneas de código abierto (*OpenSource*) brindó la posibilidad de que pequeñas empresas fueran apareciendo para comenzar a comercializar estos dispositivos.

Como puede suponerse, la aparición de estas impresoras, así como su relativa accesibilidad han supuesto un punto de inflexión en el campo de la robótica, abriendo sus puertas a una comunidad mucho más amplia que antes carecía tanto de recursos económicos, como materiales, para su desarrollo.

Una de las principales aplicaciones de estos dispositivos se sitúa en el terreno educativo ya que ofrece a los estudiantes la posibilidad de disponer de sus componentes físicos diseñados y simulados mediante aplicaciones software durante su aprendizaje. En esta universidad, más concretamente en el departamento de Ingeniería de Sistemas y Automática, el punto de partida de los robots imprimibles corresponde al robot móvil denominado *Miniskybot*, desarrollado íntegramente a partir de tecnologías *OpenSource*. Este robot, de accionamiento diferencial, se compone básicamente de una estructura imprimible impulsada por la acción de dos servomotores integrados en ella.

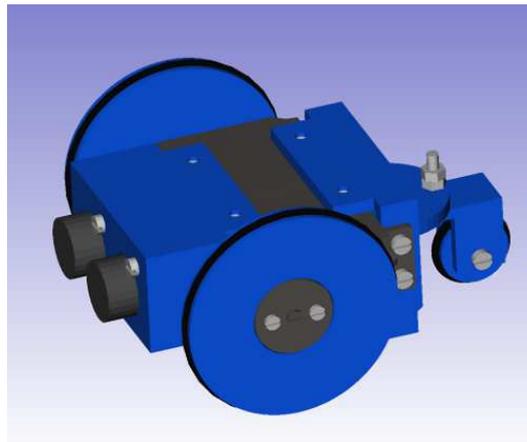


Figura 1.3: Ilustración del robot *Miniskybot*.

A partir de esta estructura básica, ha sido posible el desarrollo de otros diseños de robots imprimibles de distintas morfologías y configuraciones, todos ellos fabricados por medio de impresoras 3D. Algunos de los ejemplos de estos diseños derivados del *Miniskybot* pueden verse en la Figura 1.4:



Figura 1.4: Ejemplos de robots imprimibles. (a) Orugator. (b) F-Track

El primer diseño se trata de una versión modificada de la estructura del *Miniskybot* donde las dos ruedas han sido reemplazadas por dos orugas con el fin de proporcionar al robot cierta facilidad para moverse en terrenos irregulares. La segunda estructura, el robot *F-Track*, consta de cuatro orugas independientes y articuladas unidas a un cuerpo central, lo que le hace ideal para superar pequeños obstáculos. [4]

### 1.3.2 Localización mediante tecnología infrarroja

En la última década se ha producido un auge en las comunicaciones inalámbricas, actualmente presentes en sistemas de telefonía, comunicación móvil por satélite, acceso móvil a Internet y redes de área local inalámbricas (WLANs). La gran ventaja aportada por estas comunicaciones, la movilidad, supone también una limitación ya que la tasa de transmisión alcanzada por este tipo de conexiones es relativamente baja comparada con la tasa de las comunicaciones por cable. Dado este inconveniente, diversos cuerpos de estandarización, tanto comercial como industrial, se han visto en la necesidad de desarrollar la tercera generación de sistemas móviles de comunicaciones. La investigación y desarrollo de estos sistemas móviles de tercera generación puede ser dividido en dos clases:

- Sistemas celulares digitales de alta velocidad estandarizados internacionalmente y con movilidad como la que se obtiene en los sistemas de segunda generación.
- Sistemas de acceso móvil de banda ancha estandarizados internacionalmente, y con los cuales se tiene baja movilidad.

Una de las tecnologías más prometedoras para sistemas de acceso móviles de banda ancha, son los sistemas de comunicaciones infrarrojos (IR). Al utilizar señales ópticas, los sistemas infrarrojos de comunicaciones cuentan con un canal cuyo potencial de ancho de banda es muy grande y no están regulados en ninguna parte del planeta. Algunas de las comunicaciones más frecuentes entre sistemas infrarrojos pueden apreciarse en la Figura 1.5: [5]

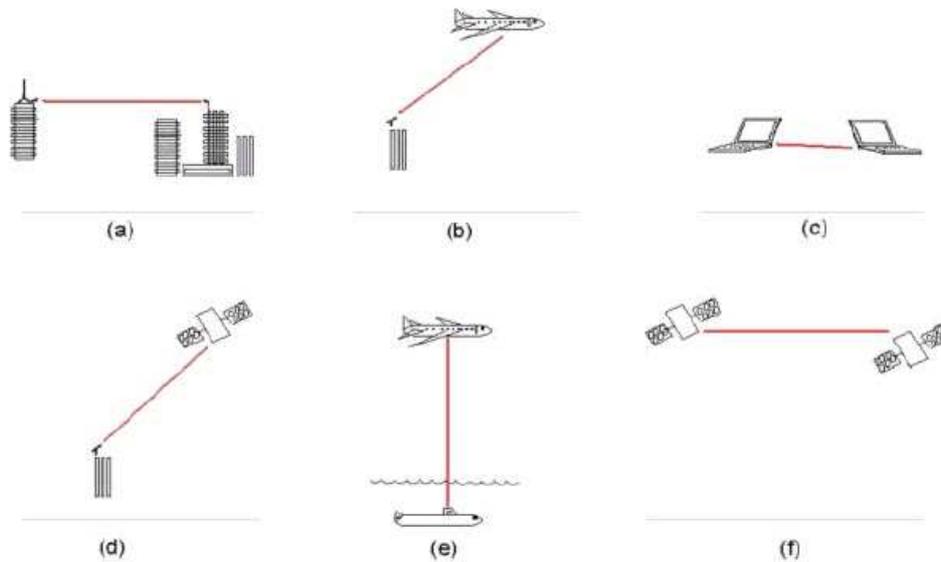


Figura 1.5: (a) terrestre, (b) tierra-aire, (c) entre dispositivos de computo, (d) tierra-satélite, (e) aire-submarino, (f) Inter-satelital.

En la actualidad, la tecnología IR es una de las alternativas utilizadas para la localización y rastreo de objetos móviles. Existen varias aplicaciones desarrolladas que utilizan diferente información de la comunicación IR para la detección de objetos en espacios interiores. Un ejemplo de estas aplicaciones se trata del sistema *Active Badge* uno de los primeros sistemas de localización basados en IR. En este sistema se coloca un punto de acceso de infrarrojos en cada habitación y el dispositivo móvil a localizar lleva incorporado un transceptor de infrarrojos que emite GIDs (*Globally unique identifiers*) de forma periódica en modo *broadcast*. Teniendo en cuenta que las señales se reflejan en casi todas las superficies, los *broadcast* de GID se contienen dentro de cada habitación obteniendo una localización precisa. Se actualiza la información cada 15 segundos de modo que si el individuo se mueve a gran velocidad pueden cometerse errores. La información de localización se recopila en una base de datos centralizada en el servidor. Una estimación de localización de gran precisión resulta complicada y requeriría múltiples antenas direccionales. Además, la escalabilidad en el sistema *Active Badge* es bastante pobre debido al rango limitado de los IR y no funciona correctamente en presencia de rayos de sol directos, siendo un problema en habitaciones con ventanas.

Por otro lado, el sistema *Versus*, también de localización en interiores, estima la localización estudiando la calidad del patrón de señal de IR recibido, a partir del cual calcula la distancia existente entre los dispositivos. El sistema

requiere dos transmisores situados alrededor del área a cubrir (IRTX1 y IRTX2). Los transmisores envían diferentes patrones de señales que son captados con calidades variables por la pareja de receptores (IRRXA y IRRXB) situados en el objeto a localizar. La calidad de recepción se define como el número de patrones recibidos de un tipo determinado comparando con los esperados en un intervalo de tiempo específico. [6]

Además de su aplicación para la localización y posicionamiento en entornos cerrados, la tecnología IR también es aplicable en espacios abiertos. Uno de los campos donde existe un mayor despliegue de tecnología infrarroja de localización se trata de la ingeniería militar. En este campo, son muchas las posibilidades que ofrece la esta tecnología. Algunas de las muchas posibles aplicaciones militares de las comunicaciones IR son: [7]

- Misiles aire-aire de traqueo IR, los cuales al realizar un traqueo pasivo, que dificultan su detección y evasión.
- Sistemas de contramedidas IR, capaces de detectar misiles de guiado pasivo (IR).
- Misiles portátiles tierra-aire de guiado IR contra unidades aéreas.

## **1.4 Organización de la memoria**

La presente memoria se organizará de la siguiente forma: en primer lugar se realizará un análisis del problema presentado. Una vez conocidos los requerimientos necesarios por la solución escogida, se pasará a la elección de la tecnología que hará posible su implementación. Se explicarán las características básicas del hardware y software empleados en la solución implementada a lo largo del proyecto. Se hará un análisis individual de cada tecnología utilizada además de la argumentación de la elección de cada una de ellas.

El diseño de la solución será presentado en el siguiente capítulo. En este apartado se explicará cómo se ha integrado y coordinado cada tecnología dentro de la solución desarrollada. Se especificará como influye cada una de ellas individualmente en el proyecto, así como la forma en la que se han configurado para su uso concreto. Se dividirá la solución obtenida en dos partes: el diseño hardware y el diseño software.

Para finalizar, se hará un balance de los resultados finales obtenidos tras la implementación y prueba de la solución final. Se terminará con las conclusiones y las posibles líneas de investigación futuras.

## **2. Análisis del problema**

La tecnología aeroespacial supone uno de los mayores retos de la ingeniería actual. Las condiciones y limitaciones a las que está sometida obligan a contemplar situaciones que cubran hasta el más mínimo detalle y permitan proporcionar diseños que deriven en resultados altamente precisos. Además de los impedimentos que puedan suponer entornos de extremas temperaturas, situaciones de ingravidez o exposiciones a grandes niveles de radiación, que no serán tenidos en cuenta a lo largo de este estudio, nos encontramos con el inconveniente que supone el desconocimiento total del terreno a explorar. Ante unas condiciones tan desfavorables de la que pocos datos se conocen, la localización y control de cualquier ente de exploración supone un gran desafío. Por lo tanto, la falta de conocimientos cartográficos así como la suposición de una gran irregularidad del terreno objeto de la exploración se convertirán en el mayor hándicap al que se enfrentará la solución descrita en este documento.

### **2.1 Análisis de las funcionalidades requeridas**

Antes de comenzar la implementación de la aplicación fue necesario la realización de un análisis profundo de los objetivos a alcanzar y las limitaciones y problemas que podrían ir surgiendo a lo largo de su desarrollo, con el fin de prever situaciones inesperadas que pudieran modificar la línea de estudio inicialmente marcada. A grandes rasgos, los objetivos planteados podrían ser cubiertos con el desarrollo de tres funcionalidades independientes: localización del robot, control de desplazamiento y comunicación entre robot y base (PC).

### 2.1.1 Localización del robot

El primer aspecto básico de un cuerpo de exploración en cualquier superficie supone el conocimiento de la posición de dicho elemento. Existen muchos métodos que permiten obtener la información necesaria que haga posible el seguimiento del sistema en cada momento. Dichos métodos, denominados Sistemas de Localización en Tiempo Real (RTLS), permiten precisar la posición actualizada de objetos con mayor o menor error dependiendo de la tecnología inalámbrica empleada en su desarrollo. Estos sistemas establecen una comunicación inalámbrica a través de la cual llevan a cabo el seguimiento de un *tag* o etiqueta ligado previamente al objeto de estudio. La información, recibida de forma continua y automática, se procesa para monitorizar el objeto y obtener así su localización. A diferencia de un sistema de posicionamiento global (GPS), estas aplicaciones no necesitan de coordenadas proporcionadas por un satélite para su funcionamiento. Un sistema RTLS completo está compuesto por dos elementos:

- **Transpondedor:** mecanismo instalado en el sistema sujeto a seguimiento y encargado de emitir una señal en modo *broadcast* que permita su detección.
- **Interrogator o lector:** mecanismo, habitualmente inmóvil, capaz de detectar las señales emitidas por el *transpondedor* y procesarlas con el fin de estimar la posición de éste.

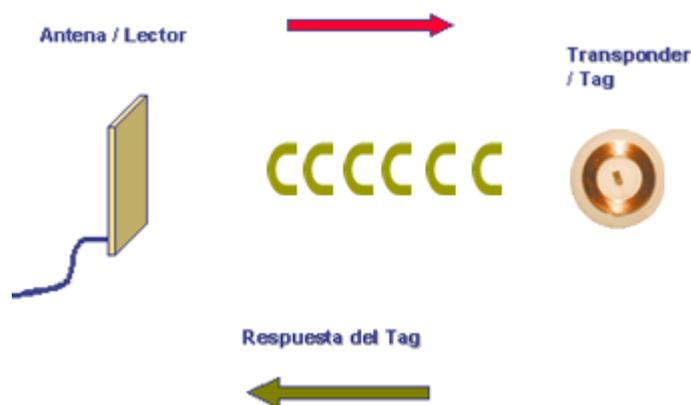


Figura 2.1: Esquema general de una comunicación *Transpondedor-Lector*.

Como se ha mencionado anteriormente, la comunicación establecida entre ambos elementos puede estar implementada a partir de diferentes tecnologías inalámbricas. Algunas de las más utilizadas son:

- **RFID:** (*Radio Frequency IDentification*) Identificación por Radio Frecuencia. Propiamente es una tecnología de identificación, aunque puede utilizarse para funciones de localización con resultados igualmente satisfactorios. Estos sistemas utilizan la información de potencia de la etiqueta RFID o *tag* como métrica de localización. A pesar de su corto alcance, ofrece una gran precisión. Supone un alto coste en infraestructuras además de poseer importantes limitaciones ligadas a la frecuencia.
- **UWB:** (*Ultra Wide Band*) Las transmisiones UWB generan pulsos aislados de corta duración que ocupan un amplio rango de frecuencias, aspecto que determina su nombre (ultrabanda) y la convierten en una tecnología idónea para localizar objetos con una precisión de centímetros en determinados escenarios. UWB utiliza principalmente la información del retardo temporal como métrica de localización. Su implementación supone un elevado coste.
- **Ultrasonido:** Sistemas basados en la utilización de nodos emisores y receptores de ultrasonidos que emplean principalmente el tiempo de vuelo de ondas ultrasónicas como métrica de localización para determinar la posición del usuario. Ofrecen una alta precisión (del orden del centímetro) pero requieren la instalación de un número elevado de nodos fijos en el entorno puesto que es necesario garantizar una línea de visión directa entre los dispositivos involucrados en el proceso de localización.
- **WiFi:** Permiten utilizar las infraestructuras de red ya desplegadas (redes de edificios, espacios públicos, etc.), por lo que acarrear un coste de infraestructura muy reducido. Por otra parte, proporcionan una baja precisión y un alcance limitado.
- **Infrarrojos (IR):** Fue la primera tecnología empleada para el desarrollo de sistemas de localización en interiores. La principal limitación de esta alternativa tecnológica es que la radiación IR no atraviesa los cuerpos, por lo que se debe asegurar la detección zonal. Las condiciones ambientales de vacío permiten que con una potencia relativamente pequeña se obtengan

alcances y razones de transmisión muy grandes en torno a cientos o miles de kilómetros y varios Gigabits por segundo (Gbps). Combinado a esto, el reducido espacio y ligereza de un sistema IR le dan una gran ventaja respecto a los sistemas de radiofrecuencia (RF) en cierto tipo de aplicaciones. Es una solución económica sujeta a una infraestructura fácil de implementar. Aunque los sistemas infrarrojos son inmunes al ruido e interferencias de tipo radioeléctrico, éstos sufren de degradaciones causadas por el ruido infrarrojo existente en ambientes exteriores e interiores, proveniente principalmente del sol y de fuentes de luz fluorescente e incandescente. [8]

Por lo tanto, de todas las tecnologías soportadas por los sistemas RTLS, la solución infrarroja será la elegida para implementar el sistema de localización de este proyecto por ser la alternativa que más se ajusta a sus necesidades, dada su simplicidad y su bajo coste. Además, supondremos que el robot, situado en la superficie del terreno a explorar, siempre será visible desde la base, que permanecerá inmóvil por encima de éste.

### Sistemas Infrarrojos:

Los sistemas infrarrojos de comunicaciones se sitúan dentro del grupo de las comunicaciones ópticas ya que la propagación de las ondas infrarrojas, que pertenecen al espectro de la luz no visible, son las perturbaciones utilizadas para establecer dicha comunicación. En la siguiente figura es posible observar el rango de longitudes de onda en las que se establece la radiación infrarroja.

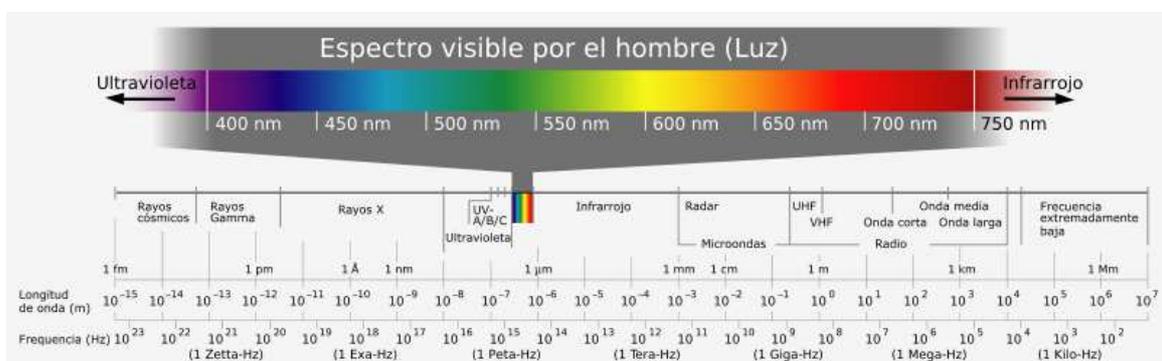


Figura 2.2: Espectro electromagnético.

Como puede observarse, la radiación electromagnética emitida por una fuente de infrarrojos se sitúa entre la luz visible, de menor longitud de onda, y las microondas, que operan a una longitud de onda mayor. Además, el espectro infrarrojo puede ser subdividido en tres secciones atendiendo al valor de longitud de onda en el que se sitúe dentro del rango específico: infrarrojo cercano (0,78 - 2,5  $\mu\text{m}$ ), infrarrojo medio (2,5 - 50  $\mu\text{m}$ ) e infrarrojo lejano: (50 - 1000  $\mu\text{m}$ ). La división de la radiación infrarroja en estos tres rangos se realiza principalmente dado que cada una de estas regiones utiliza un tipo de tecnología diferente para detectar y manipular las ondas. [9]

La longitud de onda de una señal ( $\lambda$ ) se conoce como la distancia entre puntos consecutivos de dicha onda que se sitúan en el mismo valor de fase:

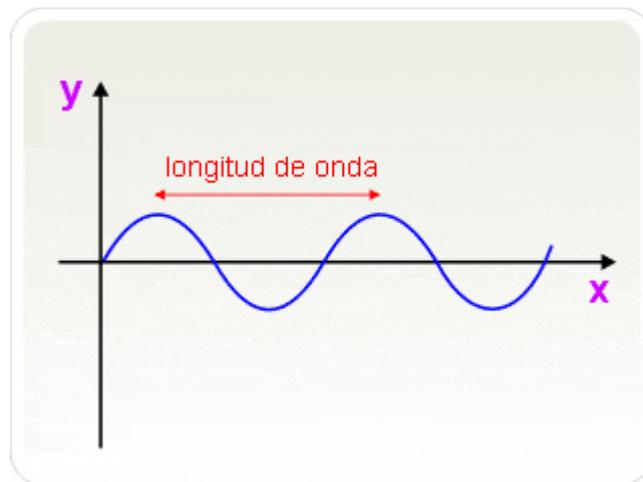


Figura 2.3: Onda senoidal.

La longitud de onda de toda señal es directamente proporcional al valor de la velocidad de propagación de dicha señal ( $v$ ) e inversamente proporcional a su frecuencia ( $f$ ). Ya que todas las señales del espectro electromagnético se propagan a la misma velocidad, esto es, a la velocidad de la luz ( $c$ ), la variación de la longitud de onda de cualquiera de estas radiaciones depende exclusivamente de variaciones en la frecuencia, como puede deducirse de la ecuación 1:

$$\lambda = \frac{v}{f} \quad \Rightarrow \quad \lambda = \frac{c}{f} \quad (1)$$

Dado que un sistema de comunicaciones infrarrojos está compuesto principalmente por un emisor y un receptor capaces de tratar este tipo de señales electromagnéticas, es necesario que ambos operen en un rango del espectro en el que sean compatibles con el fin de establecer con éxito la comunicación. Atendiendo a la disposición física de dichos componentes, es posible realizar dos tipos de clasificaciones en base a dos criterios diferentes:

- **Grado de direccionalidad.** Por un lado encontramos los enlaces dirigidos, que requieren del uso de transmisores y receptores altamente direccionales. Ambos sensores deben orientarse hacia un área común o apuntarse mutuamente para establecer el enlace. Por otro lado, los enlaces no dirigidos emplean dispositivos que operan en un amplio ángulo válido de funcionamiento, disminuyendo así dicha necesidad de orientación.

Los enlaces directos proporcionan una eficiencia máxima de potencia, ya que la señal emitida se concentra en un rango mínimo de espacio, por lo que también se consigue minimizar las pérdidas de propagación y la recepción de ruido causado por la luz ambiental. En ciertas aplicaciones, es posible establecer enlaces híbridos, en los cuales, se combinan transmisores y receptores con diferente grado de direccionalidad.

- **Línea de vista.** Referente a la existencia de una línea directa entre el transmisor y el receptor. Los enlaces de línea de vista se consiguen mediante la transmisión directa de la luz emitida entre el transmisor y el receptor. Por otra parte, en los enlaces sin línea de vista, la luz emitida por el transmisor llega al receptor generalmente después de haberse reflejado en una o varias superficies.

Los enlaces que disponen de transmisión directa maximizan la eficiencia en potencia de la señal transmitida a la vez que reducen la distorsión por multitrayectorias. Por otra parte, los enlaces sin línea de vista ofrecen una mayor facilidad de uso, mayor movilidad, y robustez, ya que la comunicación puede seguir operativa aun cuando existan obstrucciones que se interpongan entre el transmisor y el receptor.

En la siguiente figura es posible observar los diferentes tipos de sistemas infrarrojos atendiendo a los criterios de clasificación explicados: [5]

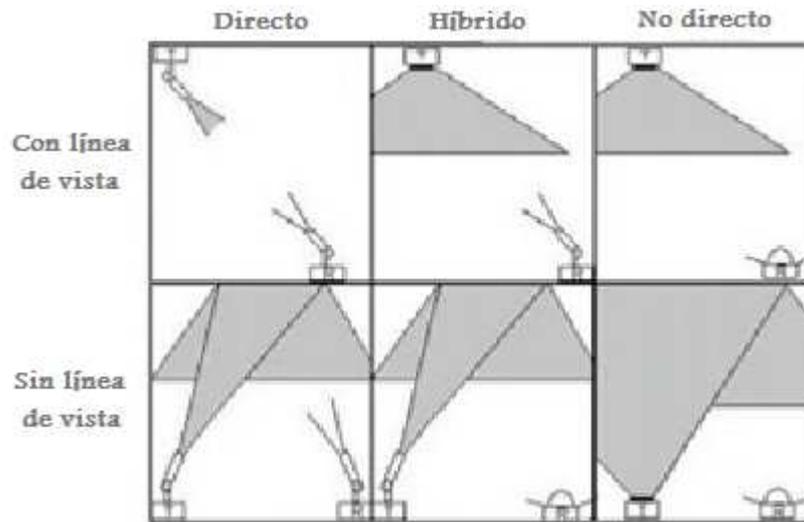


Figura 2.4: Clasificación de los sistemas IR.

Por lo tanto, según se supone la disposición de los elementos que conforman el sistema de comunicaciones a implementar en este estudio, previamente mencionado, podemos comprobar que nuestro enlace corresponderá a un sistema de comunicaciones infrarrojos no directo y con línea de vista disponible. El *interrogator* o receptor estará implementado en la estación base y el *transpondedor* o emisor estará constituido por una fuente infrarroja integrada en la superficie del robot.

Debido a la irregularidad del terreno y con el fin de conseguir un sistema de localización más robusto, se pensó en la integración de un sistema de seguimiento alternativo, que, además de aportar información sobre la posición del robot, pudiese aportar otro tipo de información adicional también útil sobre la disposición del robot en la superficie de exploración. Los dispositivos capaces de extraer información del medio físico en el que operan son conocidos como sensores. Dependiendo de donde se sitúe la percepción de dichos dispositivos, podemos encontrar dos tipos diferentes:

- **Exteroceptivos.** La exterocepción se refiere a la percepción de aspectos externos al robot como pueden ser la temperatura, presión, localización

de objetos, etc. Algunos de los sensores más utilizados para la captación de estas magnitudes externas son:

- **Sensores de ultrasonido:** Utilizados para medir distancias. Se trata de una tecnología de medida activa donde se emite una señal ultrasónica en forma de pulso para, posteriormente, recibir el reflejo de la misma o su eco. Se pueden estudiar diferentes aspectos de la señal reflejada: el tiempo de vuelo o la atenuación.

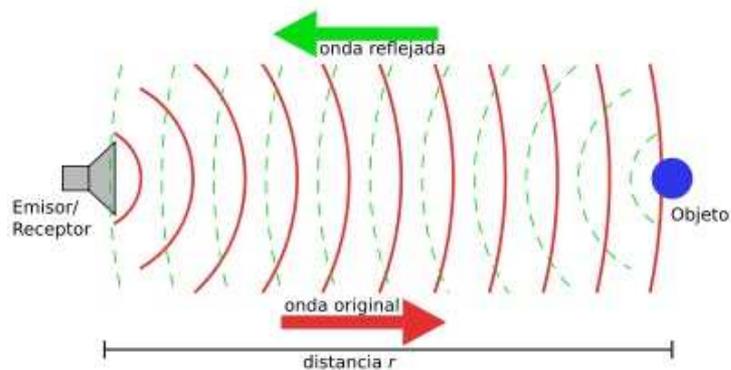


Figura 2.5: Esquema general del funcionamiento de las ondas de ultrasonido.

- **Sensores Infrarrojos:** Pueden ser utilizados para medir la distancia existente a un objeto determinado. Al igual que en los sensores de ultrasonido, dicha distancia puede ser estimada mediante el tiempo de vuelo del rayo de luz desde que es emitido hasta que se vuelve a recibir después de haber rebotado en el objeto en cuestión. La distancia también puede ser estimada a partir de ciertas relaciones geométricas existentes entre el rayo de salida, el rayo recibido y la posición del sensor.

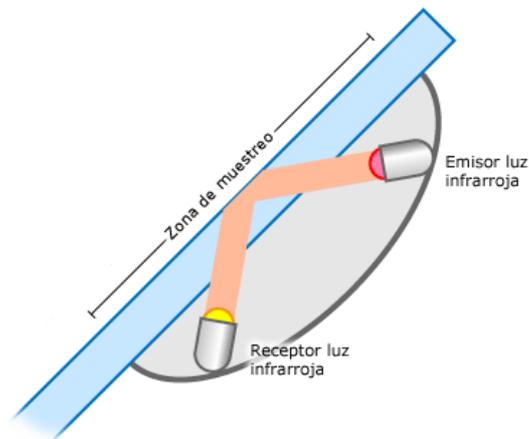


Figura 2.6: Detección de objetos utilizando sensores infrarrojos.

- **Sensores de iluminación:** Los sensores de luz son simples fotorresistencias, fotodiodos o fototransistores. Posibilitan comportamientos de un robot tales como esconderse en la oscuridad, jugar con un flash, y moverse hacia una señal luminosa. Las fotorresistencias son resistencias cuya resistividad varía con la intensidad de la luz con la que son radiadas, en muchos aspectos son parecidas a los potenciómetros. Los fotodiodos poseen una gran sensibilidad lumínica, produciendo una salida lineal en un amplio rango de niveles de luz. Responden con rapidez a los cambios de iluminación. Por último, los fototransistores son básicamente transistores cuya corriente de base es generada por la iluminación de la unión base-colector.

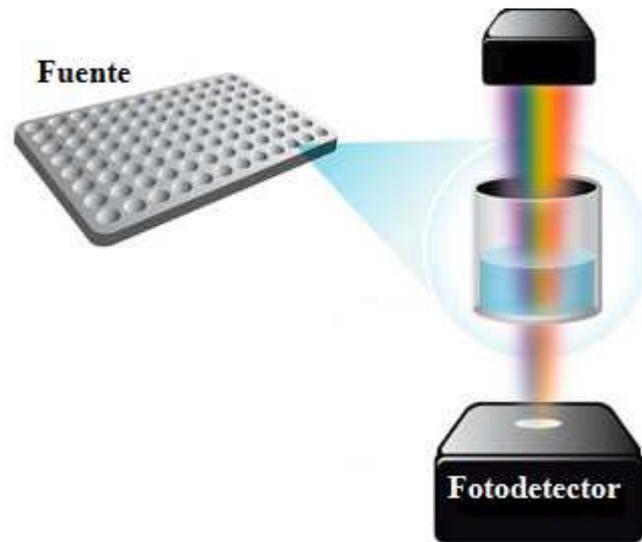


Figura 2.7: Esquema general de una fuente de luz y un fotodetector.

- **Propioceptivos.** La propiocepción se refiere a la percepción del estado interno del robot, como puede ser las medidas de carga de baterías, posición y orientación del robot, etc. Estos últimos dispositivos, que aportan datos sobre la posición y orientación del robot, también llamados sensores de posicionamiento, se pueden dividir en dos grupos:
  - o **Sensores de medida absoluta:** Dan una medida de la posición con respecto a un sistema de referencia fijo. A pesar de no acumular error con el paso del tiempo, pueden cometer errores en la medida. Los sensores de medida absoluta más comunes son:
    - **GPS (*Global Positioning System*):** Aunque no es un sensor propioceptivo, ya que necesita recibir la señal de satélites externos, tampoco es un sensor exteroceptivo. Este sistema proporciona una medida de la posición del robot y, por lo tanto, se considera un sensor de medida absoluta. Aporta información de la situación geográfica en la que se encuentra el robot móvil gracias a un conjunto de más de 20 satélites pertenecientes al departamento de defensa de los EEUU. Es el único sensor que proporciona una medida de la posición absoluta del robot que funciona en cualquier entorno exterior.



Figura 2.8: Distribución de los satélites que dan soporte a la tecnología GPS.

- **Brújula:** La brújula utiliza el campo magnético generado por la Tierra para conocer la orientación del robot. Es el único sensor de medida absoluta que mide la orientación del robot en prácticamente cualquier lugar del mundo. Como limitaciones cabe destacar que es sensible a los campos magnéticos externos y a los elementos metálicos muy cercanos al robot, ya que distorsionan el campo magnético de la Tierra.

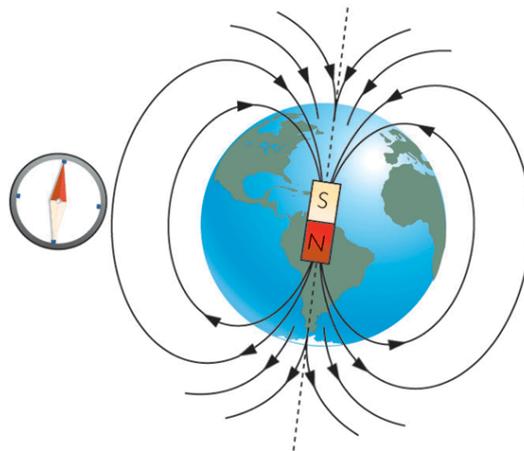


Figura 2.9: Funcionamiento de la brújula atendiendo al magnetismo terrestre.

- **Inclinómetro:** Un inclinómetro es un dispositivo simple y económico que mide la orientación del vector gravitacional. Para poder estimar la inclinación debe

estar situado en una superficie libre de fuerzas de aceleración, ya que si no la medida es errónea. Los más comunes utilizan mercurio para realizar la estimación de la inclinación. A pesar de ser muy sensible a las vibraciones, resulta muy útil cuando se trabaja en entornos irregulares.

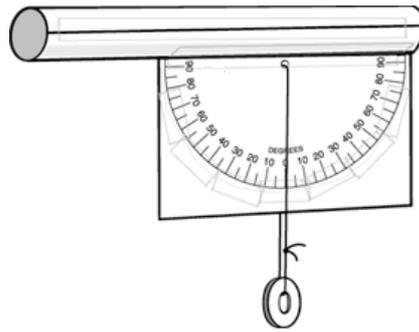


Figura 2.10: Esquema general de un inclinómetro.

- **Sensores de medida incremental:** Son sensores que proporcionan como información los incrementos de movimiento producidos con respecto a un punto fijo. El principal inconveniente se debe a los errores acumulativos existentes en dichas medidas. Los principales sensores de medida incremental son:
  - **Giroscopio electrónico:** Brújula de medida incremental, es decir, permite obtener la velocidad de rotación de un objeto, detectando así cambios en la orientación del robot. Se basa en la fuerza de Coriolis. En su interior, un material piezoeléctrico oscila a la frecuencia de resonancia, cuando se produce un giro, dicha fuerza provoca una diferencia de potencial que permite calcular la velocidad angular.

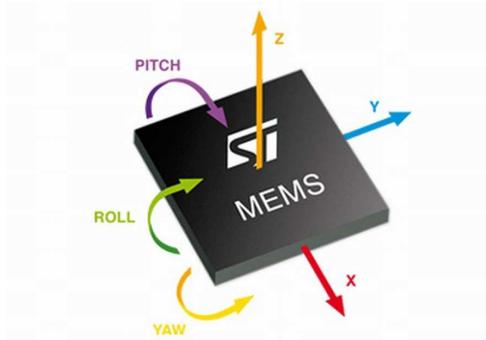


Figura 2.11: Ejes de giro de un giroscopio electrónico.

- **Encoder incremental:** Transforma movimientos angulares en impulsos digitales utilizados para detectar movimientos tanto angulares como lineales. Cuando el dispositivo es anclado a un eje, es capaz de determinar la posición angular dependiendo del número de impulsos generados durante la rotación del eje ya que un rayo de luz atraviesa los huecos opacos existentes en la superficie del disco anclado al eje.

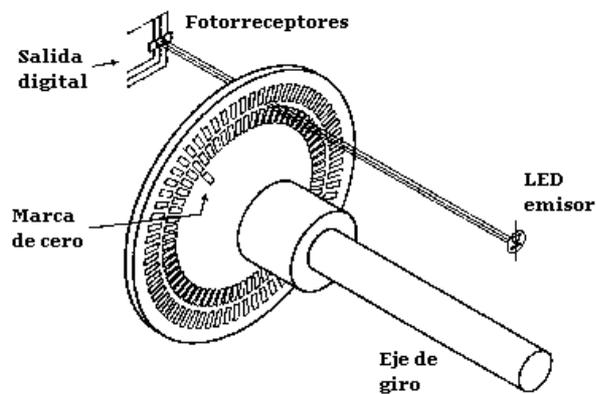


Figura 2.12: Esquema de funcionamiento de un *encoder* óptico.

- **Acelerómetros:** Dispositivos capaces de medir la aceleración lineal del objeto en el que residen respecto a un eje determinado. Es posible obtener la velocidad y, con ello, la posición actual del objeto en cuestión, aunque su principal desventaja reside en la acumulación de error y el calibrado. [10]

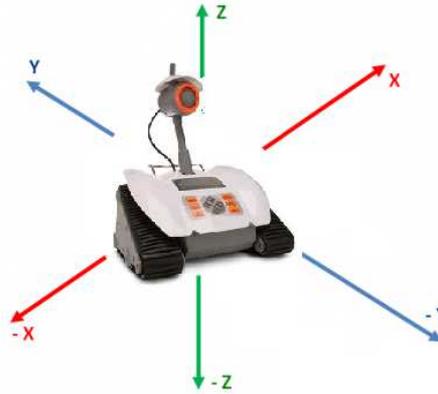


Figura 2.13: Ejes de referencia utilizados por el acelerómetro.

Tras este análisis de algunos de los sensores más utilizados en aplicaciones robóticas para la captación de datos del entorno, el acelerómetro será escogido como dispositivo de apoyo al sistema de posicionamiento implementado mediante tecnología infrarroja. Como hemos visto, el acelerómetro es capaz de calcular la posición del robot ya que aporta la aceleración existente en cada eje a cada instante, lo que brinda la posibilidad de conocer los desplazamientos realizados que, sumados a desplazamientos anteriores y conociendo el estado y posición inicial del robot, permiten calcular su posición actual.

Además de resultar un sistema de apoyo frente a posibles perturbaciones del sistema principal de posicionamiento, el acelerómetro nos aportará información adicional sobre la inclinación del robot, información muy válida dada a la gran irregularidad del terreno. Esta inclinación vendrá dada por la aceleración ejercida de la propia gravedad en el cuerpo, de la que se podrá obtener la orientación absoluta del robot.

### 2.1.2 Control de desplazamiento

El desplazamiento del robot se consigue a través de la integración de actuadores anclados en la estructura del robot. Los actuadores son dispositivos capaces de generar energía mecánica, regulada por medio de una unidad de control, a partir de diferentes fuentes de energía. Están compuestos básicamente de cuatro subsistemas coordinados entre sí:

- Sistema de accionamiento: sistema encargado de generar el movimiento.

- Sistema de transmisión: sistema que lleva a cabo la transmisión de la energía mecánica generada a otros elementos.
- Sistema reductor: sistema que regula la velocidad y el torque del actuador atendiendo a los valores establecidos.
- Sistema de control: sistema que determina el funcionamiento del actuador atendiendo a la señal con la que está siendo alimentado éste.

Es posible clasificar los actuadores en tres grupos diferentes dependiendo del tipo de energía que utilicen para generar el movimiento: [11]

- ⤴ **Neumáticos:** Utilizan aire a presión como fuente de energía. La energía acumulada en el aire comprimido es transformada en energía mecánica. A pesar de ser sencillos y robustos, no proporcionan una alta precisión, por lo que su aplicación suele derivarse a aplicaciones en las que solo se implementan dos posibles situaciones.

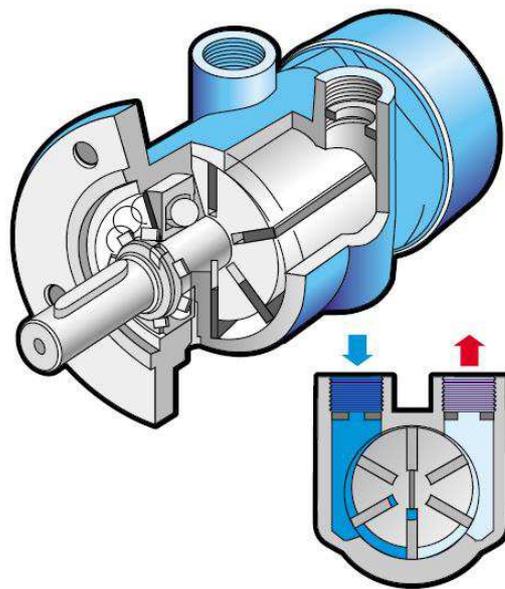


Figura 2.14: Esquema general de un motor neumático.

- ⤴ **Hidráulicos:** Fluidos comprimidos (aceites minerales o agua) transmiten la energía generada tras la pérdida de presión de un determinado caudal del fluido en cuestión. Su aplicación está orientada a robots que manejan grandes cargas ya que su compleja instalación se compensa con su gran capacidad de carga y relación peso-potencia.

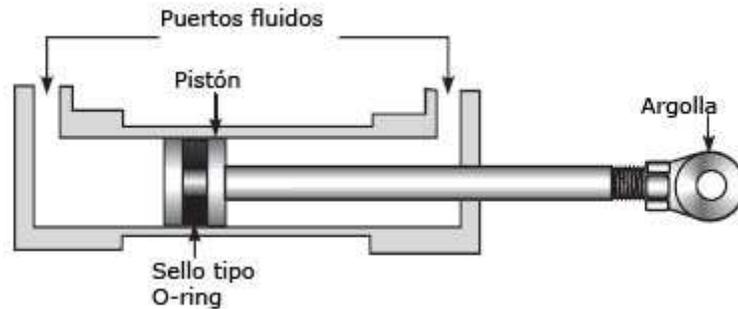


Figura 2.15: Esquema general de un cilindro hidráulico.

- ⤴ **Eléctricos:** Adquieren la energía necesaria a partir de una fuente eléctrica. Son actuadores simples y sencillos, fáciles de controlar y precisos. Podemos destacar tres tipos de actuadores eléctricos:
  - **Motores de corriente continua (servomotores):** Aportan la ventaja de poder regular la velocidad del eje, obteniendo así un rango continuo de velocidades de actuación.
  - **Motores de corriente alterna (AC):** Deben ser estimulados utilizando una señal de corriente alterna. Al igual que los motores de corriente continua, son capaces de operar en un rango continuo de velocidades.
  - **Motores paso a paso:** Convierten los impulsos eléctricos de la señal de entrada en desplazamientos angulares discretos.

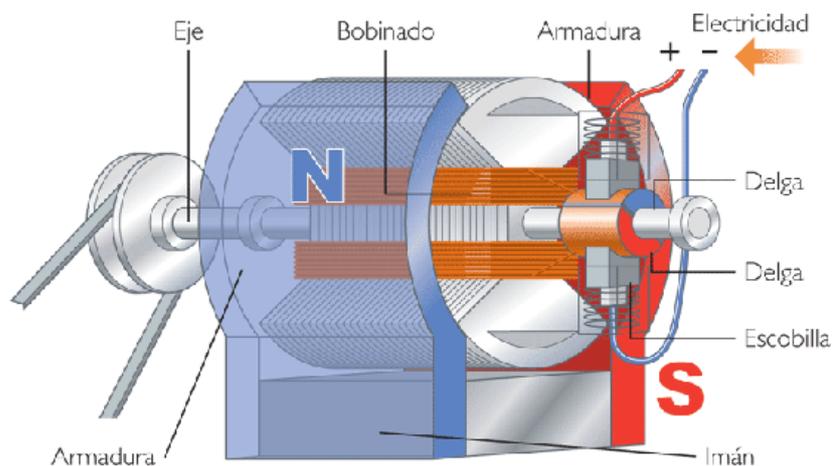


Figura 2.16: Esquema general de un motor eléctrico.

Debido al ligero peso de la estructura del robot, no será necesario el uso de potentes motores que impulsen dicho movimiento, premiando, en este caso, un bajo consumo que aporte al robot una mayor autonomía. Además se requerirá de una sencilla instalación y un control que proporcione un movimiento continuo

de cada una de las ruedas integradas en él. Por estas razones, el tipo de actuador elegido para nuestro caso se trata del motor eléctrico de corriente continua ya que proporciona unas características lo más ajustadas a nuestros requerimientos.

### 2.1.3 Comunicación robot - base

El intercambio de datos entre la estación y el robot se realizará mediante una comunicación inalámbrica, ya que queremos otorgar al ente explorador un alto grado de autonomía, que no sería posible con el uso de cables que limitasen su margen de maniobra. Como ya hemos visto, existen sistemas de comunicación inalámbricos pertenecientes a la región óptica, como la tecnología infrarroja. Para el desarrollo de la comunicación base-robot sólo tendremos en cuenta tecnologías basadas en radiofrecuencia, con el fin de implementar una solución diferente a la que se utilizó para la localización del robot.

Los sistemas de comunicación inalámbricos vía RF se basan en la propagación en el espacio de ondas de radiofrecuencia moduladas como medio de intercambio de información entre emisor/receptor. El conjunto de estas ondas, denominadas espectro radioeléctrico, que abarca tanto ondas de radio como microondas, se encuentra dentro del espectro electromagnético y se sitúa justo por debajo de la banda infrarroja.

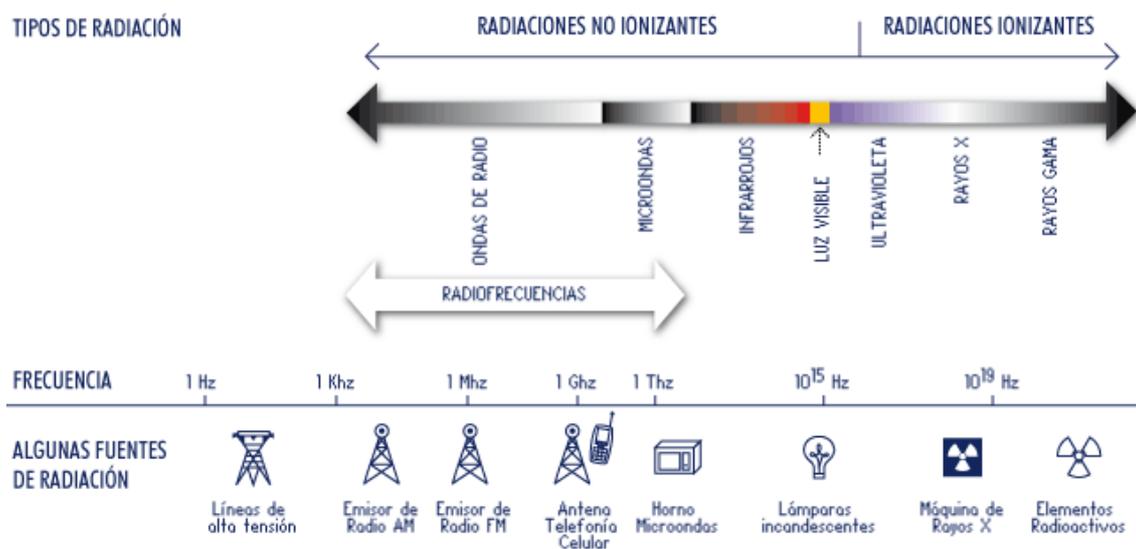


Figura 2.17: Clasificación de frecuencias dentro del espectro electromagnético.

Las radiocomunicaciones aportan grandes ventajas, como la posibilidad de movilidad del usuario y el escaso despliegue de infraestructura para su implementación. Sin embargo, se encuentran sujetas a ciertas limitaciones ligadas tanto a su potencia de transmisión como al ancho de banda. Ya que el espectro radioeléctrico es un medio limitado, cada sistema de comunicaciones debe utilizar el mínimo recurso posible con el fin de posibilitar la actuación de otros sistemas simultáneamente y evitar interferencias entre ellos. Las bandas útiles del espectro se encuentran, por lo general, sujetas a licencias reguladas por organismos tanto nacionales como internacionales. Además, existen ciertas normativas, denominadas protocolos estándar, que definen la velocidad de transmisión, ancho de banda o campo de aplicación de tecnologías concretas de estas comunicaciones. A continuación analizaremos algunas de las tecnologías vía RF más utilizadas con el fin de escoger entre ellas la más apropiada para establecer la comunicación robot-base.

- **Wireless USB (*Universal Serial Bus*):** Tecnología desarrollada por Cypress con el objetivo de implementar comunicaciones inalámbricas donde no es necesario trabajar en red. A pesar de su nombre, WirelessUSB, este dispositivo es implementado utilizando SPI (*Serial Peripheral Interface*) y no USB. Se trata de una interfaz SPI que empaqueta los datos y los transmite de manera inalámbrica a 2,4 GHz. Ofrece al usuario varias posibilidades de transmisión: transmisión simple entre dos dispositivos o entre un dispositivo máster y varios esclavos, en comunicación bidireccional. Dependiendo del rango, potencia y requisitos de la configuración, existen tres anchos de banda seleccionables 16, 32 o 64 kbps. La distancia normal de operación se sitúa sobre los 10 metros y su gran ventaja es su bajo coste.
- **WiFi:** Posibilita la comunicación inalámbrica entre dispositivos electrónicos. Consiste en una serie de estándares que funcionan en base a ciertos protocolos previamente establecidos. Dichos estándares fueron creados para acceder a cualquier tipo de red local inalámbrica, aunque su uso hoy se limita prácticamente a establecer conexiones a Internet. Esta tecnología surgió por la necesidad de establecer un mecanismo de conexión inalámbrica que fuera compatible entre los distintos dispositivos electrónicos. Para la implementación de esta tecnología es necesario de un punto de acceso, como el caso de los *routers*, y un dispositivo compatible con la tecnología WiFi, como una computadora capaz de acceder y establecer una comunicación con dicho punto. Dependiendo de los

dispositivos involucrados en la conexión, el rango de alcance será más o menos limitado.

- **ZigBee:** Tecnología inalámbrica de corto alcance y bajo consumo originaria de la alianza HomeRF que se definió como una solución inalámbrica de baja capacidad para aplicaciones del hogar como la seguridad y la automatización. Entre las actuales aplicaciones se encuentran la domótica, la automatización industrial, el reconocimiento remoto, la medicina, etc. El objetivo de esta tecnología no es operar a altas velocidades, ya que solo puede alcanzar tasas entre 20 y 250 kbps en un rango de 10 a 75 metros, sino utilizar sensores de consumo energético muy bajo.
- **Bluetooth (BT):** Se inició como un proyecto de *Ericsson Mobile Communications*. La tecnología inalámbrica de *Bluetooth* utiliza ondas de radio de corto alcance dando prioridad al ahorro de energía, ya que fue diseñada para ser utilizada en dispositivos personales de bajo peso. Dicha tecnología opera en la banda libre de los 2.4 GHz y permite comunicaciones *full dúplex*, donde un canal bidireccional puede soportar un tráfico entre dos dispositivos de una velocidad de hasta 64 kbps.

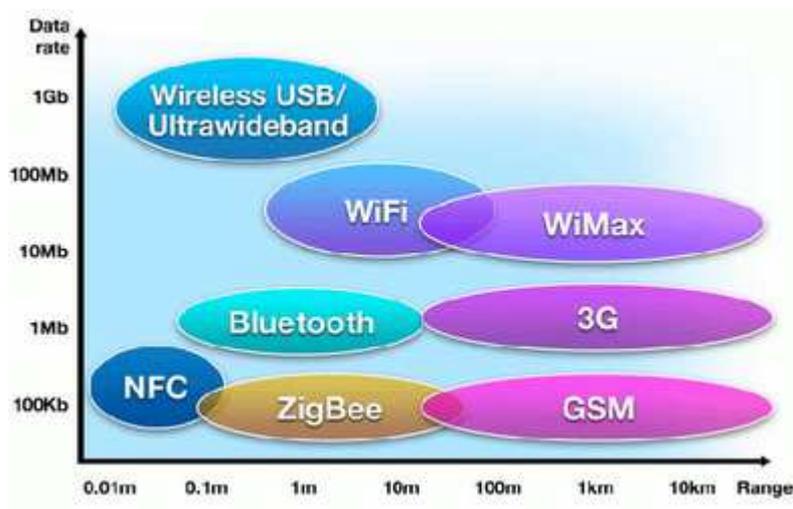


Figura 2.18: Comparación de tipos de comunicaciones inalámbricas.

Las dos últimas tecnologías presentadas parecen ser una buena solución para la comunicación inalámbrica a implementar en el proyecto. Comparando ambas, el protocolo ZigBee define un tipo de sensor para aplicaciones comerciales y de hogar, en cambio, el objetivo de *Bluetooth* es eliminar el cableado

entre dispositivos electrónicos, como puede ser el caso de los ordenadores e impresoras. *Bluetooth* está más orientado hacia la movilidad del usuario y eliminar el cableado a corta distancia. Además, *Bluetooth* ofrece la posibilidad de comunicaciones *full duplex*. Por estos motivos es el protocolo escogido para establecer la comunicación base-robot.

*Bluetooth* es una especificación que posibilita la transmisión de datos entre diferentes dispositivos mediante un enlace por radiofrecuencia. Cada dispositivo deberá estar equipado con un microchip, llamado *transceptor*, que ofrece la posibilidad de transmitir y recibir en la frecuencia de 2.4 GHz. Cada dispositivo tiene una dirección única de 48 bits basado en el estándar IEEE 802.15.1. Gracias a este protocolo, los dispositivos que lo implementan pueden comunicarse entre ellos cuando se encuentran dentro de su alcance. Debido a la naturaleza de las transmisiones, no existe la necesidad de la orientación de los dispositivos involucrados, incluso pueden situarse en habitaciones separadas, si la potencia de transmisión lo permite.

Las principales especificaciones del protocolo Bluetooth son:

- Banda de frecuencia: 2,4 GHz.
- Potencia del transmisor: entre 1 y 100 mw, típica de 2,5 mw.
- Canales máximos: hasta 3 de voz y 7 de datos por red.
- Velocidad de datos: hasta 720 Kbps.
- Rango esperado del sistema: hasta 100 metros.
- Número de dispositivos: 8 por red y hasta 10 redes.

La especificación del protocolo *Bluetooth* define el nivel físico (PHY) y el control de acceso al medio (MAC). Estos niveles inferiores de la pila de protocolos constituyen el controlador *Bluetooth* (*hardware*) que se compone de:

- **El nivel de radiofrecuencia:** Formado por el transceptor físico y sus componentes asociados. Utiliza la banda ISM (*Industrial, Scientific and Medical*) de uso no regulado a 2,4 GHz.
- **El nivel de Banda Base:** Controla las operaciones sobre bits y paquetes, realiza detección y corrección de errores, *broadcast* automático y cifrado como sus labores principales. También emite confirmaciones y peticiones de repetición de las transmisiones recibidas.
- **El nivel de Gestión de Enlace:** Es el responsable del establecimiento y finalización de conexiones, así como de su autenticación. También realiza el control del tráfico y la planificación, junto con la gestión de consumo y supervisión del enlace.

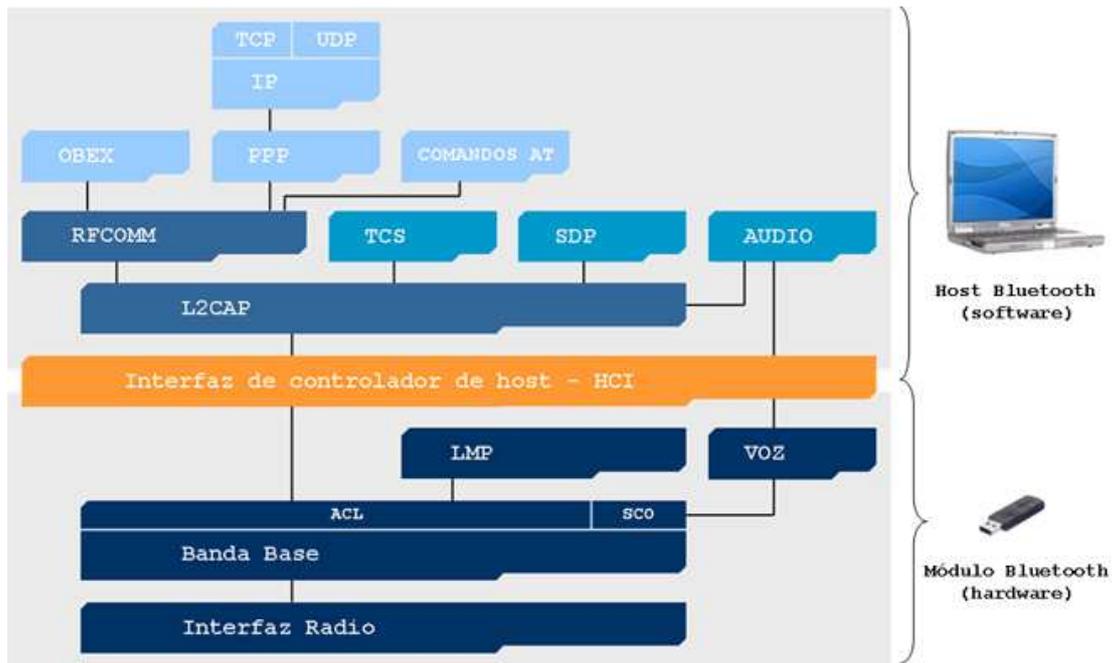


Figura 2.19: Pila de protocolos Bluetooth.

El resto de protocolos residen en el anfitrión o host *Bluetooth* y se definen a nivel software. Dicho host se comunica con el controlador utilizando un interfaz estándar. El nivel más importante del anfitrión es el protocolo de control y adaptación de enlace lógico o L2CAP:

- **Protocolo L2CAP:** La especificación Bluetooth incluye el protocolo L2CAP, que se encarga de la multiplexación de protocolos, ya que el protocolo de banda base no soporta un campo tipo para identificar el protocolo de nivel superior al que quiere transmitir la información. Puede añadir opcionalmente detección de errores y retransmisión de paquetes, así como control de flujo basado en protocolos de ventana deslizante, asignación de *buffers* y QoS (*Quality of Service*). Otra función que se realiza en el nivel L2CAP es la segmentación y recomposición de paquetes, necesaria para permitir la utilización de protocolos que utilicen paquetes.

El interfaz de controlador de host (HCI) es el encargado de realizar la comunicación entre el host y módulo Bluetooth. A partir de estos protocolos específicos de *Bluetooth*, cada fabricante implementa su capa de protocolos de aplicación. Como puede comprobarse, la especificación abierta de *Bluetooth* ofrece la posibilidad de su utilización en un gran número de aplicaciones que pueden beneficiarse de las capacidades básicas que ofrece esta tecnología inalámbrica. [12]

## 2.2 Soluciones Hardware

La estructura básica del robot es tomada como punto de partida del estudio. Complementariamente a dicha estructura, ha sido necesario disponer de distintos componentes físicos que han sido integrados en ella con el fin de obtener las funcionalidades extras requeridas para la solución planteada.

### 2.2.1 Estructura del robot imprimible

La primera decisión a tomar acerca de la solución hardware a implementar se centra en la estructura del robot. Existen diferentes configuraciones posibles de robots móviles implementados mediante ruedas para facilitar su desplazamiento. Algunas de estas configuraciones más típicas son:

- **Configuración diferencial:** Consta de dos ruedas, cada una de ellas impulsada por un motor y situadas en el eje perpendicular al movimiento del robot. Siempre y cuando ambas ruedas giren a la misma velocidad, el movimiento del robot será lineal. El giro se consigue imprimiendo una mayor velocidad en la rueda opuesta al sentido de giro. Ya que contar únicamente con dos puntos de apoyo puede ser un problema para mantener el equilibrio de la estructura, pueden utilizarse ruedas libres de giro que actúen como apoyo para su estabilización. A pesar de esta posibilidad, este tipo de robot no es apto para terrenos irregulares.



Figura 2.20: Configuración diferencial. (a) Estructura. (b) Robot Quiróptero.

- **Configuración triciclo:** Formada por tres ruedas. Las dos ruedas traseras, cuyo eje constituye el radio de curvatura, sólo sirven de soporte por lo que no están ligadas a ningún actuador. La rueda delantera ejerce la tracción y determina la dirección del robot. El inconveniente de esta configuración es la imposibilidad de realizar giros de  $\pm 90^\circ$ . Esta configuración tampoco es apta para terrenos irregulares ya que el centro de gravedad del robot tiende a alejarse de la rueda delantera en las pendientes de subida, por lo que se produce una pérdida de la tracción.

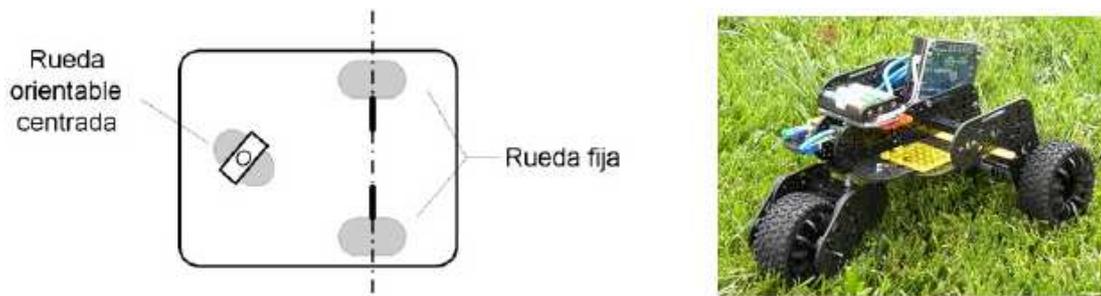


Figura 2.21: Configuración triciclo. (a) Estructura. (b) Robot de configuración triciclo.

- **Configuración de Ackerman:** Cuatro ruedas. Las dos traseras ejercen la tracción y las dos delanteras determinan la dirección. La principal característica de esta configuración es que consigue que durante el giro el ángulo de la rueda delantera interior se posicione en un ángulo mayor que el de la rueda delantera exterior, evitando así el derrape del robot.

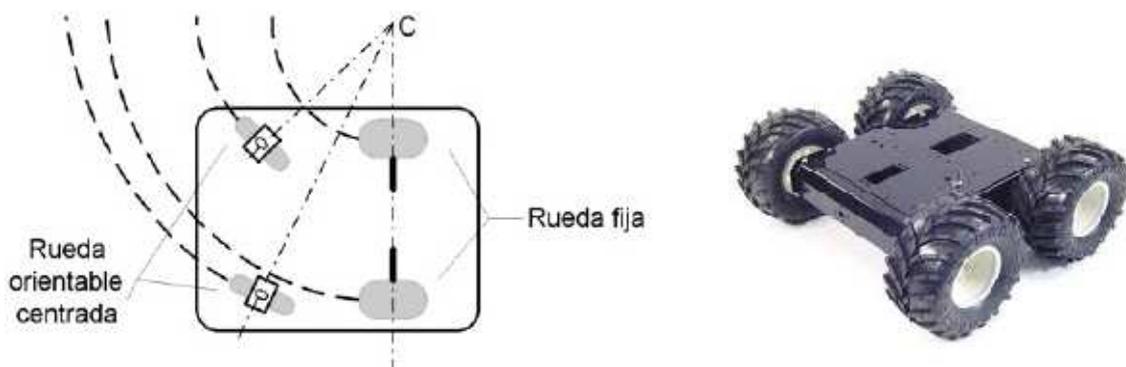


Figura 2.22: Configuración Ackerman. (a) Estructura. (b) Robot de configuración Ackerman.

- **Configuración sincronizada:** Tres o más ruedas donde todas ellas cuentan con tracción además de estar acopladas de tal forma que giran y se mueven a la misma velocidad y con la misma orientación. Con esta configuración se consigue un mínimo deslizamiento ya que todas las ruedas generan fuerzas paralelas de igual módulo. Para realizar un cambio de dirección, no es necesario el giro del robot, ya que sólo son las ruedas las que rotan.

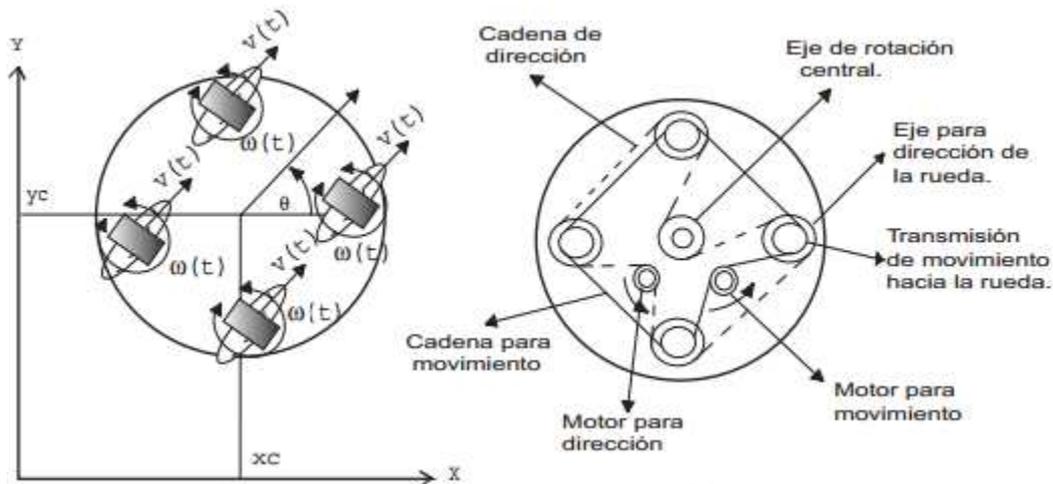


Figura 2.23: Configuración sincronizada.

- **Configuración omnidireccional:** Formada por un robot al que se le ha dotado de ruedas omnidireccionales. Este tipo de rueda se compone de una rueda estándar dotada de una corona de rodillos perpendiculares al eje de rotación. Tienen la ventaja de poder moverse en cualquier dirección sin necesidad de reorientarse, pero la coordinación entre las ruedas supone una gran complejidad.

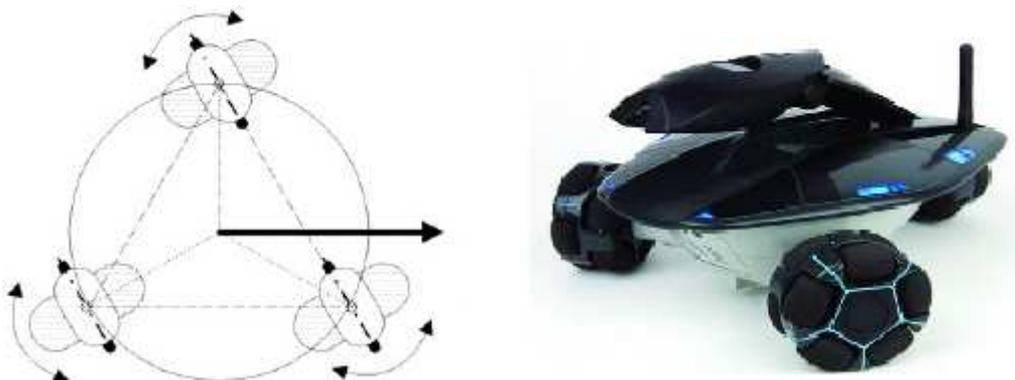


Figura 2.24: Configuración omnidireccional. (a) Estructura. (b) Robot Rovio.

- **Configuración con múltiples grados de libertad:** Esta configuración se utiliza cuando es necesaria una mayor tracción del robot. Un mayor número de ruedas constan de propulsión, por lo que se consigue un desplazamiento más potente. Como contrapartida, se produce un aumento de la complejidad de control y coordinación de todos y cada uno de los motores implicados.



Figura 2.25: Robot Rovers.

- **Configuración oruga:** La tracción se obtiene mediante la incorporación de orugas en lugar de ruedas. Estas orugas están formadas por una correa de transmisión que se encarga de enlazar cada rueda trasera, accionada por un motor, con la rueda delantera correspondiente. Es un caso específico de configuración diferencial, ya que el desplazamiento se rige por el mismo comportamiento. Es utilizado en situaciones en las que el terreno es muy irregular ya que las orugas consiguen aumentar la superficie de contacto con el suelo, consiguiendo así un sistema más estable. [13]

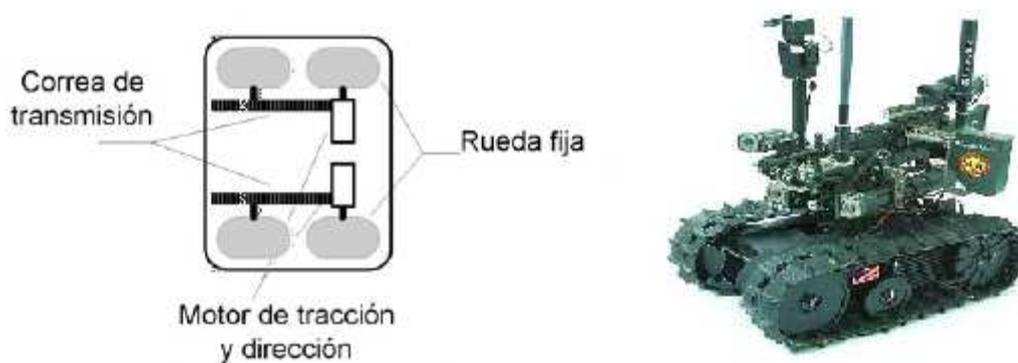


Figura 2.26: Estructura de tracción oruga.

El robot elegido para el estudio corresponde con la configuración oruga ya que se ha tenido muy en cuenta la irregularidad del terreno. Se trata de un robot imprimible cuya estructura deriva de un diseño simple, ya que el objetivo del proyecto no se quiere centrar en la complejidad del robot, si no en la implementación de una solución para la localización y control genérico de cualquier robot explorador, por diferente que sea su estructura. La decisión de utilizar un robot imprimible se basa, además de en su simplicidad, en la rápida y fácil disposición de dicha estructura, ya que es posible producirla en nuestras propias instalaciones gracias a las impresoras 3D disponibles, implicando, además, un bajo coste de fabricación.

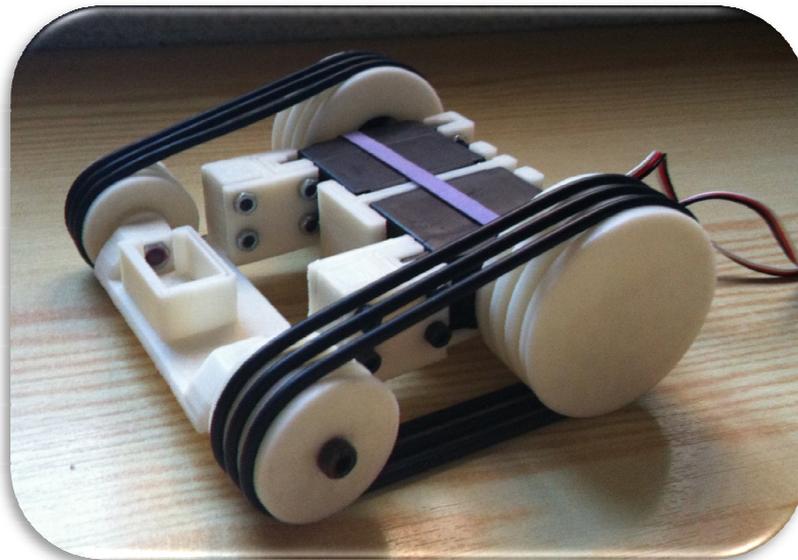


Figura 2.27: Estructura básica del robot elegido para el proyecto.

La principal ventaja de este sistema de desplazamiento radica en el hecho de que es posible el giro del robot sobre su propio eje, pudiendo realizar maniobras muy precisas en un espacio reducido. El desplazamiento se consigue coordinando el movimiento de cada una de las dos orugas que trabajan independientemente, cada una de ellas impulsadas por un motor diferente. El valor de la velocidad de cada una de estas orugas queda determinado por el valor de la velocidad lineal ( $v$ ) y angular ( $w$ ) que se quiera proporcionar al robot. A continuación vamos a realizar un estudio de la cinemática de esta configuración para obtener las ecuaciones según las cuales se rige el desplazamiento del robot escogido y poder así controlar su movimiento.

La ecuación general de la velocidad de una rueda, es:

$$v = \text{radio} \cdot w \quad (2)$$

donde *radio* se refiere al radio de la rueda en cuestión. Por lo tanto, las velocidades de la rueda derecha,  $V_d$ , e izquierda,  $V_i$ , serán:

$$V_d = \text{radio} \cdot w_d \quad (3)$$

$$V_i = \text{radio} \cdot w_i \quad (4)$$

A partir de estas ecuaciones, es posible calcular la distancia recorrida por la rueda derecha,  $S_d$ , por la rueda izquierda,  $S_i$ , y la distancia recorrida por el punto central del robot,  $S$ :

$$S_d = (a + L) \theta \quad (5)$$

$$S_i = a \cdot \theta$$

$$S = \theta \cdot R \quad (6)$$

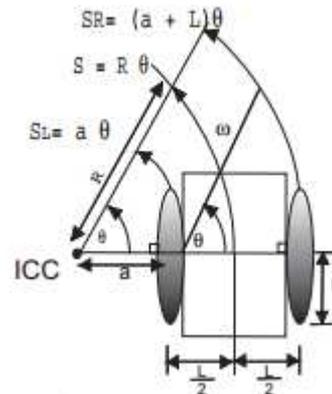


Figura 2.28: Representación del modelo diferencial en situación de giro.

Como puede verse en la Figura 2.28,  $R$  corresponde al radio de curvatura de la trayectoria del robot,  $a$  corresponde la distancia entre la rueda interna del giro y el centro de giro (ICC),  $\theta$  representa el ángulo de giro y  $L$  la distancia entre las ruedas. Despejando en función de  $S$ , obtenemos:

$$S_d = S_i + L \cdot \theta \quad (7)$$

Derivando esta ecuación y despejando, podemos obtener  $\omega$ :

$$V_d = V_i + L \cdot \omega \quad \Rightarrow \quad \omega = \frac{V_d - V_i}{L} \quad (8)$$

Para obtener la velocidad lineal del sistema completo, calculamos primero  $S$ , a partir de la suma de las ecuaciones de  $S_r$  y  $S_i$ , y despejamos:

$$S_i + S_d = 2a\theta + L\theta = 2\theta\left(a + \frac{L}{2}\right) \quad \Rightarrow \quad S = \frac{S_d - S_i}{2} \quad (9)$$

Una vez calculado  $S$ , derivamos la ecuación para obtener  $v$ :

$$v = \frac{V_d - V_i}{2} \quad (10)$$

Por tanto:

$$S = \begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \frac{V_d + V_i}{2} \\ \frac{V_d - V_i}{L} \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ -\frac{1}{L} & \frac{1}{L} \end{bmatrix} \begin{bmatrix} V_i \\ V_d \end{bmatrix} \quad (11)$$

Despejando de esta ecuación en función de  $V_d$  y  $V_i$ , obtenemos las expresiones que nos permiten conocer la velocidad con la que debemos estimular cada una de las ruedas para conseguir que el robot siga una trayectoria deseada:

$$\begin{bmatrix} V_i \\ V_d \end{bmatrix} = \begin{bmatrix} 1 & \frac{-L}{2} \\ 1 & \frac{L}{2} \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (12)$$

## 2.2.2 Actuadores

Como se determinó anteriormente, el tipo de actuador escogido para ser utilizado en el proyecto se trata del actuador eléctrico. Dentro de los actuadores eléctricos, el modelo escogido para ser integrado en la estructura del robot corresponde al actuador FUTABA S3003.



Figura 2.29: Actuador FUTABA S3003.

Se trata de un modelo de tipo estándar diseñado para ser ligero y rápido. Ya que no proporciona una elevada potencia, puesto que no puede soportar grandes cargas, su consumo es especialmente bajo. Cada servomotor necesita ir conectado a tres terminales: toma de alimentación (5V), toma de tierra (GND) y señal de control. Esta señal de control será la que determine el movimiento del motor en cada instante. Dicha señal se trata de una onda cuadrada de voltaje, donde la duración del nivel alto ( $T_{on}$ ) marcará la posición a adquirir por el servo a cada momento, es decir, el funcionamiento de estos motores se rige por un sistema de control de los denominados PWM (*Pulse Width Modulation*).

La señal a ser interpretada por este modelo de actuadores se corresponde a una señal digital (0 ó 1) y periódica, de 50 Hz de frecuencia, es decir, un periodo de 20 ms. La máxima anchura de pulso permitida y que establece la posición del servo en uno de sus extremos (90°) se establece en 2,3 ms, por otro lado, la mínima anchura contemplada, que establece la posición en el otro extremo (-90°), corresponde a un intervalo de 0,3 ms de nivel alto.

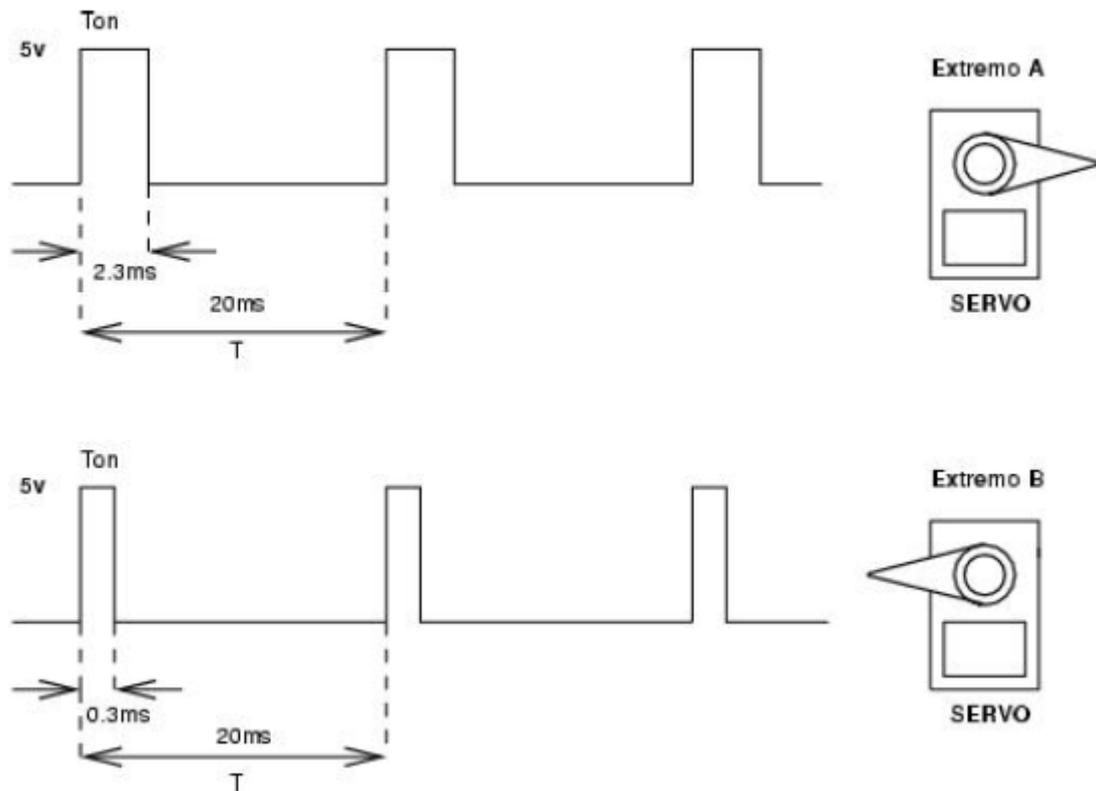


Figura 2.30: Pulsos de control del actuador FUTABA S3003.

A partir de estos datos es posible obtener la ecuación que relaciona la anchura del pulso de la señal de control ( $T_{on}$ ) con el ángulo adquirido por el eje del actuador:

$$\theta = 90 \cdot (t - 1,3) \quad (13)$$

Dado que el giro del motor está limitado a 180°, es necesario realizar una modificación del hardware original de dichos servos con el fin de conseguir que el giro proporcionado sea completo, es decir, un giro de 360°. Dicha modificación se basa en eliminar el tope físico que impide al servo su giro continuo. De esta forma, el comportamiento del servo en función de la anchura de pulso será:

- ▲  $T_{on} = [0,3 \text{ ms}, 1,3 \text{ ms}]$ : Se produce un giro constante y antihorario. La velocidad del motor aumenta según disminuye la anchura del pulso.

- ⤴  $T_{on} = 1,3 \text{ ms}$ : El servo permanece en esa posición.
- ⤴  $T_{on} = (1,3 \text{ ms}, 2,3 \text{ ms}]$ : Se produce un giro constante en sentido horario. La velocidad de giro aumenta según aumenta  $T_{on}$ .

Tras un estudio experimental documentado en [4], se puede comprobar cómo varía la velocidad del actuador modificado en función al ángulo asignado al mismo:

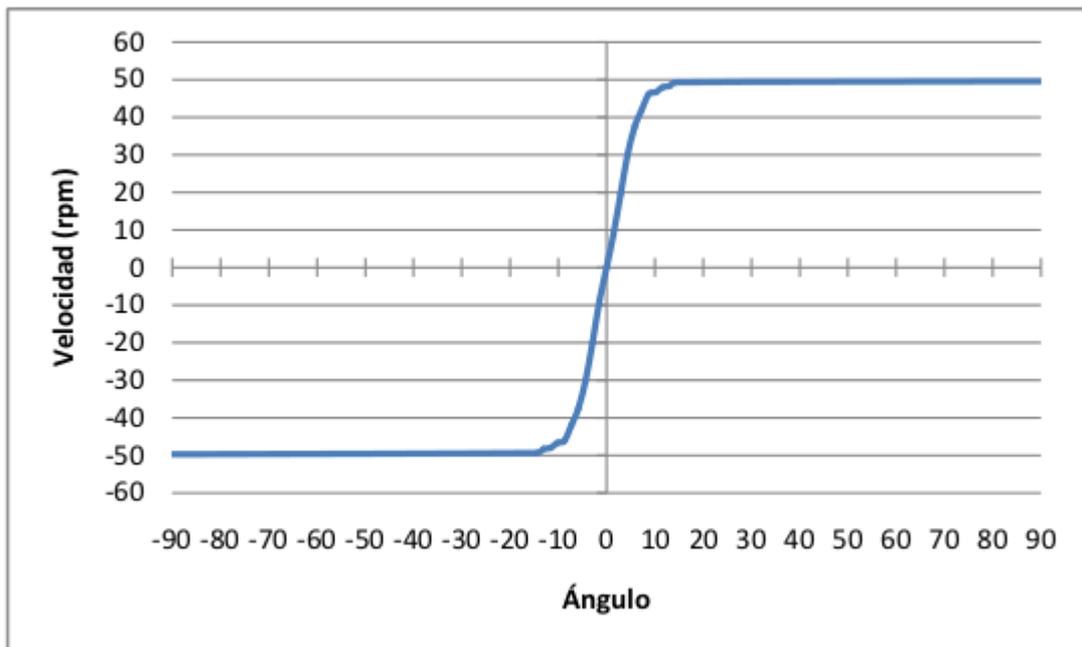


Figura 2.31: Relación velocidad-ángulo proporcionada por los actuadores.

Como puede verse en la Figura 2.31, la velocidad proporcionada por el actuador varía en el rango  $[-50, 50]$  rpm (revoluciones por minuto).

### 2.2.3 SkyMega

Para el control de todos los dispositivos electrónicos fue necesaria la utilización de un microcontrolador. Un microcontrolador es un circuito integrado programable compuesto por una unidad central de procesamiento, una memoria y periféricos de entrada/salida. Con el fin de facilitar el uso de estos dispositivos en cuanto a la utilización de los periféricos y sus propias funcionalidades, cabe la posibilidad de ser integrado en una placa que facilite el acceso a los pines, aportando cierto grado de comodidad a su manipulación,

además de posibles aplicaciones extras. Estas placas compuestas por un microcontrolador y los circuitos de conexión correspondientes son denominadas también tarjetas microcontroladoras.

La tarjeta microcontroladora acoplada en la estructura del robot y encargada de dar funcionalidad y coordinar el hardware integrado en él se trata de la tarjeta "SkyMega". Sus principales características son:

- ✦ Microprocesador ATMEGA328 a 16 GHz.
- ✦ Conexión de hasta 8 servomotores de 5V.
- ✦ Posibilidad de comunicación a través del puerto I2C entre varias tarjetas.
- ✦ Hasta dos conectores de I2C.
- ✦ Conector de alimentación tipo mólex.
- ✦ Conector de grabación.
- ✦ Led de prueba.
- ✦ Pulsador de pruebas.
- ✦ Micro-interruptor on/off.
- ✦ Led de power-on.
- ✦ Slot de expansión para conectar sensores.

La alimentación requerida por la SkyMega viene dada por un conector mólex de 2 vías que proporcione entre 4,5 y 6 voltios. Su conexión al PC se realiza a través de un cable USB-serie FTDI (*Future Technology Devices International*). Esta tarjeta fue diseñada para su uso en aplicaciones robóticas modulares, por lo que se adecúa perfectamente a nuestro planteamiento. Además, es hardware libre y totalmente compatible con Arduino. [14]

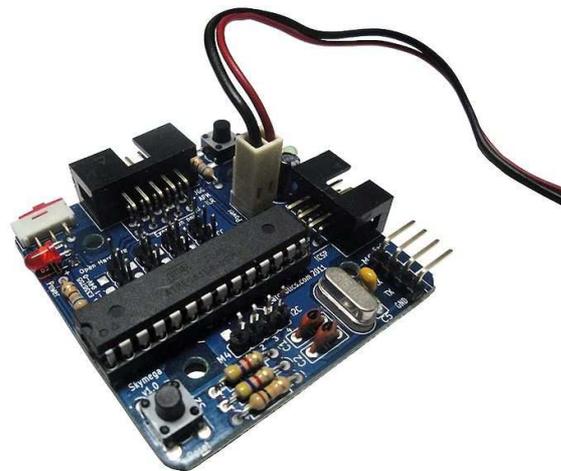


Figura 2.32: SkyMega.

Arduino es una plataforma de electrónica abierta para la creación de prototipos basada en software y hardware flexibles y fáciles de usar. Puede tomar información del entorno a través de la conexión de sensores a sus pines de entrada y puede actuar sobre aquello que le rodea controlando luces, motores y otros actuadores. El microcontrolador integrado en la placa Arduino se programa mediante el lenguaje de programación Arduino y el entorno de desarrollo Arduino. [15]

El hecho de que la placa escogida sea compatible con la tecnología de Arduino, abre la posibilidad de que sea programada en el lenguaje específico proporcionado por esta tecnología, además de en C. La elección de la tarjeta microcontroladora SkyMega frente al propio Arduino se basa en la sencillez que proporciona esta placa. Al estar diseñada específicamente para su uso en robótica, proporciona facilidades ausentes en la placa original, a la vez de reducir recursos que no son necesarios para nuestra aplicación. A causa de este hecho, conseguimos un ahorro de energía además de reducir las dimensiones físicas de la placa.

#### 2.2.4 Emisor infrarrojo

Como se ha mencionado, la fuente principal de información que aporta datos sobre la posición del robot se implementa mediante tecnología infrarroja. Para ello es necesario incorporar a la superficie robot una serie de emisores infrarrojos capaces de transmitir una señal que sea posible interpretar por parte de un receptor.

Los emisores infrarrojos están fabricados de un material denominado semiconductor, un material con propiedades especiales que le hace capaz de la emisión de luz. Los electrones en este tipo de materiales se encuentran situados en bandas de energía, entre las cuales se sucederán transiciones energéticas de estos electrones. Existen dos situaciones diferentes en las que puede darse una transición: si la transición es de una posición de mayor energía a otra de menor, entonces no se producirá ninguna emisión de luz, absorbiendo el propio material la energía perdida. En el caso contrario, el salto de un electrón de una banda energética menor a una mayor puede generar la emisión de fotones.

Existen dos tipos diferentes de emisiones que pueden darse en un emisor infrarrojo: emisión espontánea y emisión estimulada:

- **Emisión espontánea:** Un conjunto de electrones situados en un nivel superior pierden energía y saltan de forma espontánea a un nivel inferior liberando energía en forma de un fotón.

- **Emisión estimulada:** Los electrones situados en el nivel superior sufren un proceso de pérdida de energía inducido aplicando un campo eléctrico de frecuencia adecuada. El electrón salta por tanto al nivel inferior emitiendo un fotón de igual energía. La diferencia en este caso es que el fotón emitido posee la misma frecuencia, fase, polarización y se propaga en el mismo sentido que el electrón que lo generó, de esta forma, la energía radiada corresponde ahora con dos fotones idénticos.

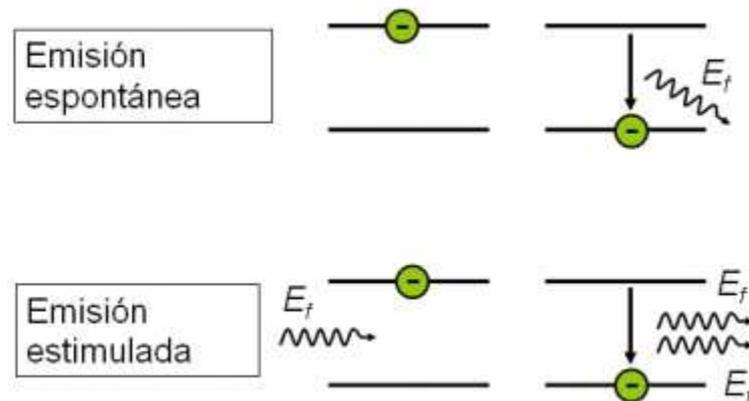


Figura 2.33: Procesos de emisión espontánea y estimulada.

Los emisores infrarrojos pueden clasificarse básicamente en dos tipos: el Diodo Emisor de Luz (LED) y el Diodo Láser (LD).

- **LED:** Capaz de emitir radiación óptica dependiendo de la intensidad eléctrica con la que sea estimulado. La luz emitida proviene casi exclusivamente de emisiones espontáneas, por lo que la radiación será menos direccional. Existen tres tipos: de emisión por superficie, donde la luz generada se propaga en todas las direcciones; de emisión lateral, con una superficie de emisión más reducida para su mejor acoplo, y leds superluminiscentes, con mayor potencia de emisión.
- **Diodo Láser:** Al igual que el LED, es capaz de liberar energía óptica tras su estimulación eléctrica. La gran diferencia es que esta energía corresponde a emisiones estimuladas, por lo que se consigue una potencia y una direccionalidad mucho mayor que en el caso del LED.

A pesar de que el láser posee una potencia óptica de salida notablemente mayor que la proporcionada por el LED, la línea de emisión generada por el LED es al menos diez veces mayor, característica de la que nos podremos beneficiar para hacer el campo de localización del robot más amplio que en el caso del láser. [16]

Tras un estudio de los componentes LED existentes actualmente en el mercado, el modelo escogido por su compatibilidad con nuestra tecnología fue el IR383, por estar diseñado específicamente para sistemas de transmisión al aire libre.



Figura 2.34: IR383 (GaAlAs).

Estos leds poseen unas características de emisión y funcionamiento que encajan perfectamente con los requerimientos de nuestra aplicación. De su hoja de datos se pueden obtener sus principales características de funcionamiento:

- ♣  $V_R = 5v$ .
- ♣  $V_F = 1,4v$ .
- ♣  $I_F = 100mA$ .
- ♣  $\lambda_F = 940nm$ .
- ♣ Ángulo de emisión =  $20^\circ$ .
- ♣ Potencia disipada =  $100mw$ .
- ♣ Intensidad radiada =  $20mw/sr$ .

### 2.2.5 Receptor infrarrojo

El proceso en el cual cierta energía óptica, infrarroja en este caso, es detectada se denomina absorción. Cuando un fotón con una longitud de onda adecuada, que dependerá de la energía de banda prohibida ( $E_g$ ), incide sobre un electrón, se produce una cesión de energía por parte del fotón que provoca el salto del electrón a una banda de valencia mayor. La característica principal de los receptores infrarrojos radica en el hecho de que son capaces de transformar la luz recibida en corriente eléctrica mediante este proceso.

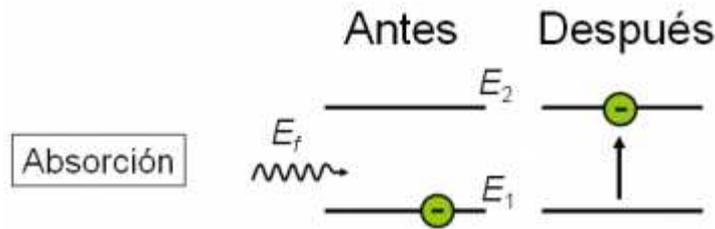


Figura 2.35: Proceso de absorción.

Para completar la funcionalidad de la localización mediante sistema infrarrojo ha sido necesaria la utilización de un receptor capaz de detectar la energía óptica emitida por los leds incorporados en la superficie del robot. En lugar de diseñar y fabricar un propio receptor, se ha optado por la opción de utilizar un dispositivo ya existente y fabricado para esta misma finalidad. El dispositivo se trata de un dispositivo denominado *WiiRemote* o *WiiMote* desarrollado por la empresa de videojuegos Nintendo. Este dispositivo sirve de mando controlador de la consola conocida como Wii lanzada en el año 2006. Se trata de un controlador inalámbrico que permite a los jugadores realizar movimientos naturales, los cuales se usan como entrada de datos, gracias a sus sensores de movimiento y su sistema de posicionamiento espacial. Destaca sobre todo por la capacidad de detección de movimiento en el espacio y la habilidad de apuntar hacia objetos en la pantalla. El mando está compuesto básicamente por 3 dispositivos que son los que le aportan estas funcionalidades:

- **Acelerómetro de 3 ejes.** Mide la inclinación del mando en 3 ejes y se transmite a la consola a través del enlace bluetooth. Este acelerómetro mide las fuerzas de gravedad en los tres ejes principales.
- **Emisor / Receptor Bluetooth:** Para comunicarse con la consola, el *WiiMote* dispone de un chip emisor y receptor de datos por Bluetooth, proporcionado por la empresa especializada en comunicaciones Broadcom Technologies. Este chip, modelo BCM2042 ofrece una velocidad de transferencia de 2.1 Mbps y un consumo relativamente bajo. Gracias a este protocolo se establece una comunicación entre el *WiiMote* y el PC.
- **Sensor de infrarrojos:** Este elemento es similar al de una cámara de visión infrarroja y se utiliza para indicar hacia donde apunta el usuario con el mando, por lo que el mando se usa como puntero. Para su funcionamiento se requiere una fuente de luz infrarroja que pueda detectar el sensor para transmitir a la consola su posición. Para ello la videoconsola Wii suministra una barra con leds infrarrojos llamada "*Wii Sensor Bar*" que se coloca encima o debajo de la pantalla de TV. Este receptor infrarrojos, constituido por una cámara de alta resolución, dispone de un filtro infrarrojo de tal forma que la cámara solo es capaz de recibir imágenes de fuentes emisoras de infrarrojos. Esta cámara dispone de un rango de

visión efectiva de 33 grados horizontales y 23 grados verticales. Percibe la radiación infrarroja de un punto y envía la posición X e Y de dicho punto, respecto de la imagen que capta la cámara en ese momento, que tiene una resolución de 1024x768. De esta forma, el punto superior izquierdo de la imagen captada por la cámara correspondería a la posición (0,0), el punto superior derecho al par de coordenadas (0,1023), la esquina inferior izquierda a la posición (767,0) y, finalmente, la esquina inferior derecha a las coordenadas (1023, 767). El mando dispone de un chip desarrollado por PixArt Imaging Inc. que posiciona el punto detectado sobre el plano y envía la posición vía Bluetooth al dispositivo al que esté conectado. Este chip es capaz de procesar cuatro puntos diferentes simultáneamente, por tanto, es posible posicionar hasta cuatro emisores infrarrojos de los cuales el chip se encarga de enviar las coordenadas de posición de cada uno. El muestreo de dichos puntos se realiza cada 100 ms. [17]



Figura 2.36: Wiimote.

## 2.2.6 Módulo Bluetooth

La transmisión de datos entre la base y el robot se lleva a cabo mediante una comunicación *Bluetooth*. Para esta comunicación se ha escogido un módulo Bluetooth que va incorporado en la estructura del robot y que es compatible con la placa incorporada en el robot.



Figura 2.37: Módulo BT transceptor RS232 Arduino.

Las principales características de este dispositivo son:

- ⤴ Alimentación: 3.6 – 6 V.
- ⤴ Protocolo Bluetooth: Bluetooth specification v2.0.
- ⤴ Frecuencia: 2.4 GHz.
- ⤴ Modulación: GFSK (*Gaussian Frequency Shift Keying*).
- ⤴ Potencia de emisión: 4 dBm.
- ⤴ Sensibilidad: -84 dBm.

Actualmente existen gran variedad de modelos de adaptador esclavo BT en el mercado, todos ellos con características muy similares. La elección de este modelo de entre los demás compatibles se ha realizado esencialmente por motivos de disponibilidad y coste.

### 2.2.7 Acelerómetro

La otra fuente de información sobre la posición del robot se obtiene mediante la incorporación de un acelerómetro en la base del robot. Además de los requerimientos de alimentación, ha sido necesario localizar un acelerómetro de 3 ejes que pudiese ser conectado a través de la interfaz I2C (*Inter-Integrated Circuit*), ya que la otra posibilidad, la interfaz SPI, es la encargada de soportar el módulo *Bluetooth*, por lo que no está disponible.

Al igual que en el caso del módulo *Bluetooth*, actualmente existen muchas posibilidades en el mercado de acelerómetros compatibles con nuestra placa que cumplan todos los requisitos deseados. Finalmente, se ha optado por el modelo ADXL345 por su consumo, resolución y coste.

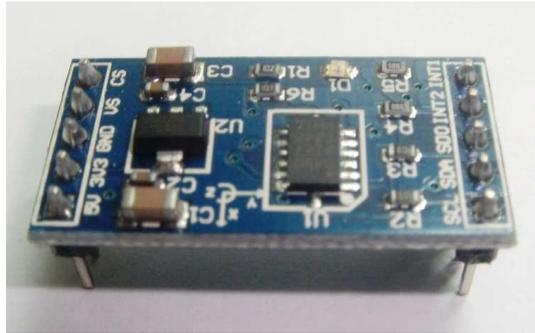


Figura 2.38: Acelerómetro ADXL345.

El ADXL345 es un reducido acelerómetro de bajo consumo que proporciona medidas en los 3 ejes con alta resolución (13 bits) y es capaz de detectar hasta 16 g. Los datos digitales de salida corresponden a palabras de 16 bits y es accesible a través del SPI o la interfaz digital I2C. El ADXL345 es muy adecuado para medir la aceleración estática de la gravedad en aplicaciones de detección, así como la aceleración dinámica resultante del movimiento. Su alta resolución (4 mg/LSB) permite la medición de los cambios de inclinación de menos de 1,0°. Es capaz de detectar la presencia o ausencia de movimiento y la caída libre.

## 2.3 Soluciones Software

Para la implementación de la solución software se han empleado dos lenguajes de programación diferentes. Uno de ellos, Arduino, se ha utilizado para especificar el comportamiento del microprocesador incorporado en la placa del robot, y el otro, OpenRDK es utilizado para la implementación de la aplicación que controlará el robot desde la base.

### 2.3.1 Arduino

El lenguaje Arduino está basado en C/C++ por lo que soporta todas las construcciones de C estándar y algunas de las funcionalidades de C++. La estructura básica de un programa implementado mediante este lenguaje requiere la codificación de dos funciones obligatorias:

- **setup:** Se emplea para inicializar variables, establecer el estado de los pins, inicializar librerías, etc. Esta función se ejecutará una única vez después de que se conecte la placa Arduino a la fuente de alimentación, o cuando se pulse el botón de reinicio de la placa.

- **loop:** Se ejecuta de manera continua. Se usa para controlar de forma activa la placa Arduino.

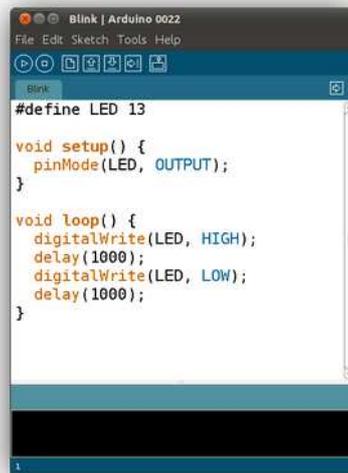


Figura 2.39: Interfaz gráfica para la programación de Arduino.

Además de estas dos funciones básicas, es posible la creación y utilización dentro del propio programa de otras funciones complementarias, tal y como se realiza en el lenguaje C.

A pesar de que la placa microcontroladora escogida para el proyecto permitía la posibilidad de su codificación en lenguaje C, finalmente se ha optado por Arduino ya que, a pesar de ser un lenguaje nuevo y poco extendido (al contrario que C), proporciona una serie de librerías que facilitan la configuración y gestión de los dispositivos que van a ser interconectados en la placa, como puede ser el caso de los servos y el módulo BT.

### 2.3.2 OpenRDK

OpenRDK es un software que proporciona una estructura modular centrada en el desarrollo de sistemas distribuidos para la robótica. Hasta ahora ha sido aplicado con éxito en diversas aplicaciones con robots heterogéneos. En esta estructura, la entidad principal es un proceso denominado agente. Un módulo es un solo hilo que se ejecuta en el interior del proceso agente, por lo que un agente puede estar formado por uno o varios módulos coordinados entre sí por dicho agente. La configuración del agente se especifica en un archivo de

configuración que debe contener un listado de los módulos que lo forman, junto con los valores de sus parámetros y la interconexión de éstos.

Los módulos se comunican mediante un repositorio en el que publican algunas de sus variables internas (parámetros, entradas y salidas), denominadas propiedades. Un módulo define sus propiedades durante la inicialización, después, el valor de las propiedades es publicado en el repositorio por lo que otros módulos, dentro del mismo agente o en otro agente diferente, puede acceder a dichos datos. La ventaja es que el acceso a las propiedades remotas es transparente, además de conseguir una reducción de la memoria compartida en el caso de las propiedades locales.

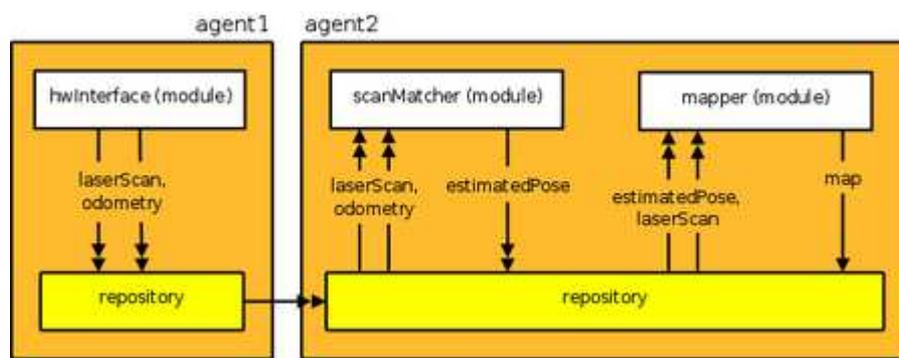


Figura 2.40: Ejemplo de aplicación implementada a partir de OpenRDK.

En la Figura 2.40 puede verse el ejemplo de una aplicación compuesta por dos agentes ejecutándose en dos máquinas diferentes, uno de ellos formado por un único módulo, y el otro agente, formado por dos módulos. En este caso, `hwInterface` recupera datos de un telémetro láser y de odometría y los deposita en el repositorio. El módulo `scanMatcher` utiliza un algoritmo de análisis con el fin de estimar la posición del robot utilizando los datos aportados por estos sensores. Por último, el `asignador` utiliza la posición estimada, junto con los datos aportados por el escáner láser, para construir un mapa del entorno.

Una alternativa al uso de OpenRDK como estructura software de la aplicación final, viene dada por el lenguaje ROS (*Robot Operative System*). ROS provee las bibliotecas y herramientas necesarias para ayudar a los desarrolladores de software a crear aplicaciones robóticas. Proporciona abstracción del hardware, controladores de dispositivos, bibliotecas, visualizadores, paso de mensajes, gestión de paquetes y mucho más. [18]

Ambas herramientas pertenecen al grupo de los llamados *OpenSource* (software libre), por lo que se distribuyen gratuitamente. A pesar de constituir ambas buenas soluciones para la implementación de la aplicación, finalmente se

ha optado por el uso de OpenRDK ya que es este software ha sido utilizado previamente en el departamento y se prevé su uso futuro, por lo que la aplicación desarrollada a lo largo de este proyecto podrá servir como punto de partida de otros estudios, abogando así por la reutilización del software.

### 3. Diseño de la solución

Una vez tomadas todas las decisiones sobre las tecnologías con las cuales implementar la solución del proyecto, se comienza con su desarrollo. Esta solución va a ser dividida en dos partes: una parte hardware en la que se explica la integración de todos los dispositivos hardware, así como el circuito electrónico que los soporta, y una parte software que muestra cómo ha sido programada tanto la placa microprocesadora como la aplicación general para coordinar y conseguir la funcionalidad completa del sistema.

#### 3.1 Diseño Hardware

El componente hardware principal al que todos los demás dispositivos electrónicos van conectados es la placa microcontroladora. Esta placa es la encargada de dar soporte y funcionalidad a los servomotores, al módulo bluetooth, a los emisores led y al acelerómetro, todos ellos encaramados en la estructura del robot. A continuación se explica cómo ha sido realizada la integración y conectividad de cada uno de estos dispositivos:

- **Servomotores:** Para conseguir el movimiento del robot oruga ha sido necesaria la integración de dos servomotores, cada uno de ellos proporcionando tracción a cada una de las dos ruedas traseras. Como se ha especificado anteriormente, cada actuador debe estar alimentado a 5V para su correcto funcionamiento, por lo que se ha realizado una conexión directa a la placa aprovechando las salidas facilitadas para hasta ocho servos a ese voltaje.
- **Módulo bluetooth:** Dotado con 4 pines de salida correspondientes a la alimentación, GND, RX y TX, también se ha conectado directamente a la placa, utilizando en este caso los pines facilitados para la conexión serie.
- **Emisores led:** En un principio, se pensó en la utilización de un único emisor infrarrojo encargado de soportar la funcionalidad de localización

del robot. Durante el desarrollo de la solución se desechó esta idea dado que la información aportada por un único punto en el plano aportaba conocimiento sobre la posición del robot, pero imposibilitaba el cálculo de su orientación. Para salvar este impedimento se ha determinado la utilización de tres emisores dispuestos en forma de triángulo isósceles, donde la menor de las distancias, la base del triángulo, corresponde con el eje formado por las ruedas del robot.

Para la implantación de estos tres leds en la superficie de la estructura se ha diseñado un circuito que cumple con las necesidades de voltaje y corriente requeridas por la hoja de especificación de los leds.

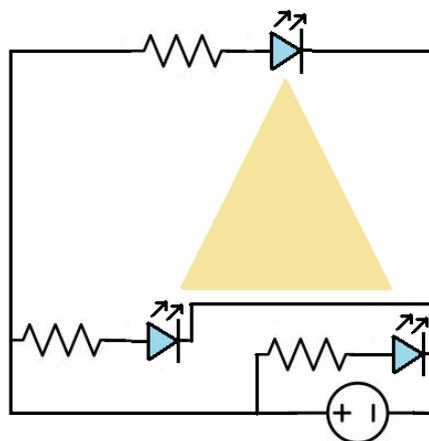


Figura 3.1: Circuito implementado para el acoplamiento de los leds.

Como puede verse en la Figura 3.1, los tres leds se disponen en paralelo con el fin de que todos ellos compartan la misma tensión de alimentación, los 5V proporcionados por la placa microcontroladora. Además, en cada una de las ramas se incluye una resistencia que regula la corriente que atraviesa cada led, adecuando dicha corriente a las especificaciones. Ya que cada led puede soportar un máximo de 100 mA y que existe una caída de tensión de 5V, entonces:

$$V = I \cdot R \quad \Rightarrow \quad \frac{V}{I} = \frac{5}{0,1} = 50 \, \Omega \quad (14)$$

Por lo que se ha optado por utilizar tres resistencias de 51 ohmios incorporadas cada una en cada rama del circuito.

- **Acelerómetro:** La conexión del acelerómetro tampoco se ha realizado mediante una conexión directa, ya que necesita de un regulador de tensión adicional para su acoplamiento. Como puede comprobarse en la hoja de características del acelerómetro, es necesario estimular a este dispositivo con un voltaje de 3,3V para su correcto funcionamiento. Ya que la placa solo es capaz de proporcionar voltajes de 5V, se ha incluido un regulador

entre ambos dispositivos capaz de adaptar el voltaje proporcionado (5V) al voltaje requerido (3,3V).

- **Batería:** Para la alimentación de la tarjeta microcontroladora se ha acoplado un portapilas que aloja en su interior cuatro pilas, de 1,5V cada una, suministrando un total de 6V a todo el sistema.

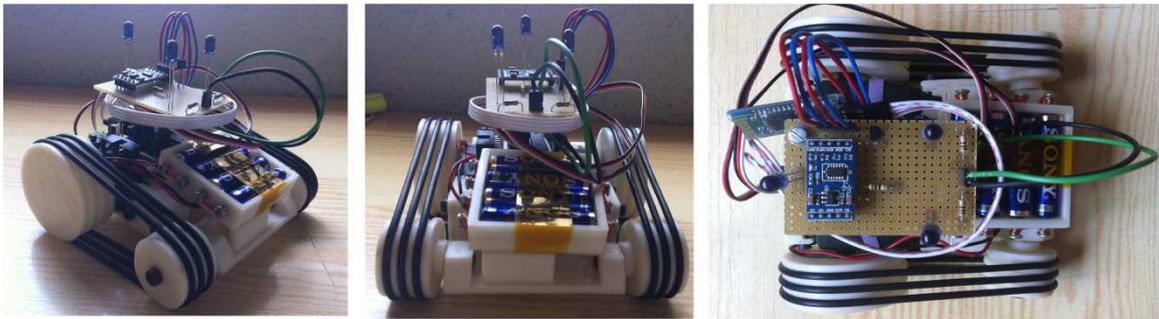


Figura 3.2: Estructura final del robot desde diferentes perspectivas.

### 3.2 Diseño Software

La solución software es la encargada de otorgar la funcionalidad completa a los dispositivos electrónicos mencionados anteriormente. El objetivo general del robot es lograr localizarse a partir de los datos obtenidos de los sensores incorporados en él, asignando su posición actual a unas coordenadas absolutas que representen dicho punto en el espacio. Una vez localizado, ofrece la posibilidad de alcanzar otra localización diferente a partir de las coordenadas de dicho punto. Para ello es necesaria una continua actualización de los datos de posición y control de movimientos del robot. La solución software es la encargada de recopilar todos los datos proporcionados por los sensores, interpretarlos y otorgar al robot la movilidad necesaria para alcanzar la posición final.



Figura 3.3: Esquema general del funcionamiento sensores-actuadores.

El software implementado se divide en dos aplicaciones diferentes: una corresponde al programa codificado en la placa controladora, la otra a la aplicación desarrollada en la base, ambas comunicadas mediante una conexión *Bluetooth*.

### 3.2.1 Programación de la placa microcontroladora

Como se ha explicado anteriormente, la programación de la placa microcontroladora ha sido realizada en lenguaje Arduino. El código implementado se encarga básicamente de dos funciones:

- **Gestión de datos del acelerómetro:** Periódicamente, la tarjeta debe leer la información proporcionada por el acelerómetro acoplado al robot. Esta información corresponde a las variaciones de la aceleración en cada uno de los 3 ejes de posición. Estas aceleraciones son utilizadas para calcular el incremento de la posición actual del robot y poder obtener así la posición absoluta. Una vez realizados los cálculos, las coordenadas  $x$ ,  $y$  y  $z$  del punto en el espacio en el que se encuentra el robot son enviadas a la base a través de la comunicación *Bluetooth*.
- **Control de los motores:** Por otra parte, la placa recibe por ese mismo medio de comunicación los valores de la velocidad lineal ( $v$ ) y velocidad angular ( $w$ ) que debe adquirir el robot en cada momento con el fin de alcanzar el punto destino. A partir de estos valores, el programa aquí ejecutado calcula como estimular cada uno de los servos para alcanzar dichas velocidades. Posibles situaciones:
  - o  $v = 0$  y  $w = 0$ : En este caso, la base está comunicando al robot que se detenga, por lo que los actuadores no deben ejercer ninguna acción.
  - o  $v = 0$  y  $w \neq 0$ : Solo existe velocidad angular, el robot debe girar sobre su propio eje. Ambas ruedas deben moverse a la misma velocidad, pero en sentido contrario, dependiendo del sentido de giro.
  - o  $v \neq 0$  y  $w = 0$ : Ya que solo existe velocidad lineal, el robot avanza en línea recta. Ambas orugas son impulsadas por la misma velocidad.
  - o  $v \neq 0$  y  $w \neq 0$ : El robot debe avanzar a la vez que gira. Ambas ruedas rotarán en el mismo sentido, pero a diferentes velocidades.

Una vez conocido el tipo de desplazamiento a realizar, se lleva a cabo el cálculo preciso de las velocidades a adquirir por cada una de las orugas. Este cálculo se basa en las posibilidades físicas de los servomotores, ya que, como hemos indicado anteriormente, están limitadas entre  $[-50, 50]$  rpm. Dado que, en la práctica, el comportamiento de las velocidades imprimidas por los servos es casi discreto, se ha optado también por la discretización de las velocidades a asignar. De esta forma, cada rueda puede adquirir 5 velocidades diferentes en sentido de rotación horario, y otras 5 velocidades diferentes en el sentido antihorario. El algoritmo de cálculo de la velocidad final es el siguiente: dada una velocidad máxima especificada como parámetro, se establece una correspondencia entre las posibles velocidades a recibir, es decir, el rango  $[-V_{\max}, V_{\max}]$  y las 10 velocidades admitidas. Para establecer esta correspondencia se divide  $2 \cdot V_{\max}$  entre las 10 velocidades, obteniendo así los rangos en los que operarán cada una de las velocidades discretas. Una vez recibida una nueva velocidad, se comprueba a que rango pertenece y, por lo tanto, la velocidad discreta a asignar.

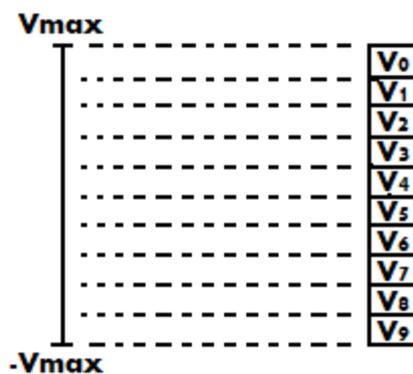


Figura 3.4: Correspondencia entre velocidades continuas y discretas.

Una vez calculadas las velocidades discretas correspondientes a cada una de las ruedas, se actualiza el valor de los servos con el fin de que adquieran este nuevo valor.

### 3.2.2 Programación de los módulos de la aplicación

La aplicación implementada en la base es la encargada de recopilar los datos de la localización, tanto de la tecnología infrarroja como del acelerómetro y, una vez conocida la posición actual, calcular la trayectoria hacia la posición final. Para ello, la aplicación obtiene el punto inicial y el punto final y calcula la velocidad lineal y angular a adquirir por el robot.

El funcionamiento de la aplicación se reparte en cinco módulos, todos ellos conectados por un repositorio general que hace posible el intercambio de datos entre un módulo y otro.

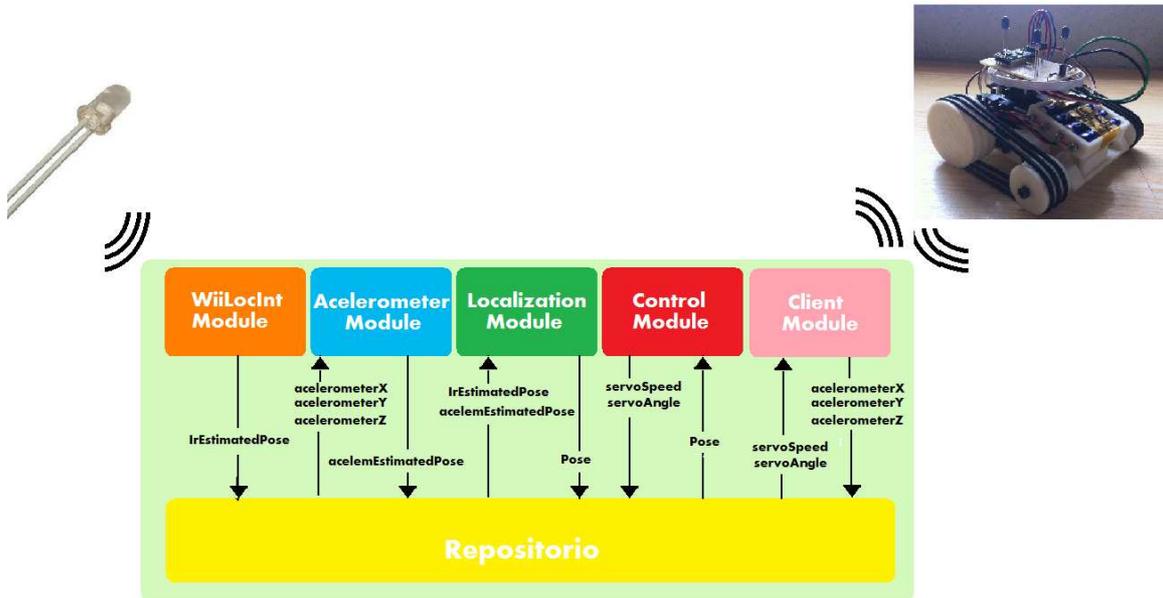


Figura 3.5: Esquema general de los módulos.

Una vez conocemos el flujo de datos entre cada uno de los módulos, es posible especificar lo que internamente realiza cada uno de ellos:

- WiiLocIntModule:** Encargado de la recepción e interpretación de los datos infrarrojos utilizados para la localización del robot. Cada uno de los tres leds posicionados en la superficie del robot son detectados por el receptor infrarrojo, el *Wiimote*. Tal y como se ha explicado, este dispositivo funciona como una cámara capaz de detectar sensores infrarrojos. Los puntos detectados en su campo de visión son posicionados sobre el plano paralelo a la superficie de visión, haciendo corresponder las coordenadas de cada punto con cada uno de los píxeles con los que está formada la imagen captada.

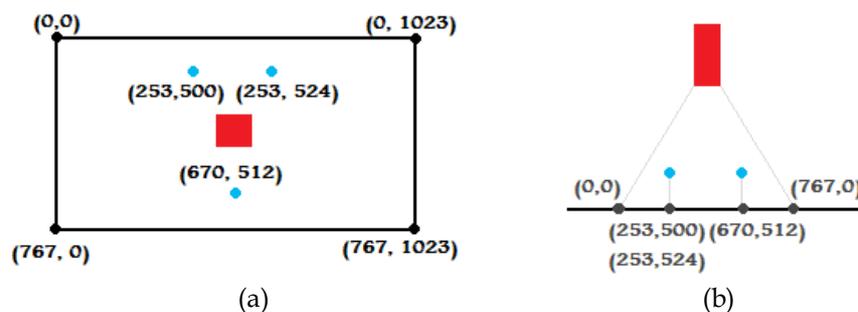


Figura 3.6: Detección de los emisores infrarrojos. (a) Vista superior. (b) Vista lateral.

Este módulo se encarga de abrir un puerto de comunicación *Bluetooth* entre el *Wiimote* y el ordenador mientras la recepción de datos es correcta. Una vez iniciada la conexión, el receptor realiza una lectura periódica de las coordenadas  $x$  e  $y$  de cada uno de los tres leds y envía la información por puerto creado. A partir de esta información, es posible conocer la posición y orientación del robot, valor que se deposita en el repositorio a través de la variable "IrEstimatedPose", para su uso por parte de otros módulos. Esta variable se corresponde a un punto orientado en el espacio. Los datos se continúan actualizando mientras la conexión esté abierta.

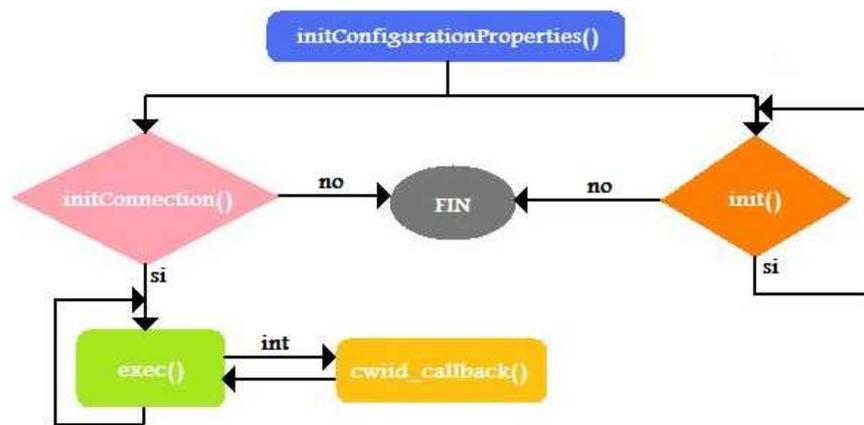


Figura 3.7: Diagrama de flujo del módulo WiilocIntModule.

Según podemos ver en la Figura 3.7, este módulo está compuesto de cinco funciones:

- *initConfigurationProperties()*: Declara e inicializa las propiedades utilizadas por dicho módulo.
- *initConnection()*: Establece la conexión *Bluetooth* entre el receptor infrarrojo y el ordenador. Si el puerto de comunicación no se encuentra disponible, se termina la ejecución del módulo.
- *init()*: Comprueba periódicamente, cada 100 ms, si la conexión permanece operativa. Cuando la comunicación deja de estar activa, se termina la ejecución del módulo.
- *exec()*: Su función es actualizar el valor de la variable "IrEstimatedPose" almacenada en el repositorio. Se ejecuta continuamente. Cuando se reciben datos por parte del receptor infrarrojos, una interrupción detiene el proceso para ejecutar la función *cwiid\_callback*.
- *cwiid\_callback()*: Encargada de obtener las coordenadas de cada uno de los tres leds y ejecutar el algoritmo de cálculo de la posición estimada a partir de dichos datos.

Como explicamos anteriormente, los leds instalados en la superficie del robot forman un triángulo isósceles. A partir de las coordenadas recibidas es posible hacer corresponder cada uno de los puntos conformados por estas coordenadas con cada uno de los vértices del triángulo. De esta forma, es posible estimar la posición del robot con su correspondiente orientación. El proceso se realiza mediante los siguientes pasos:

- Se calculan las distancias existentes entre cada uno de los tres leds. La menor distancia obtenida se supone que es la base del triángulo.

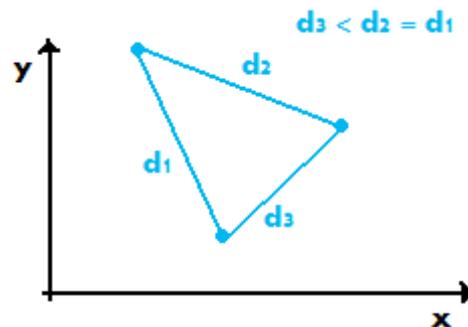


Figura 3.8: Representación del cálculo de las distancias entre los leds.

- Se establece como posición estimada el punto medio de dicha base.

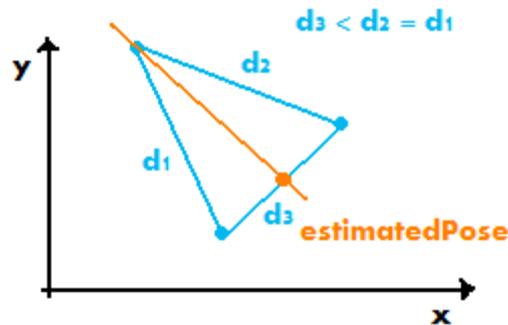


Figura 3.9: Representación de la estimación del punto medio de la base del robot.

- Para establecer la orientación, se calcula el ángulo existente entre la mediatriz de la base del triángulo y el eje x de coordenadas.

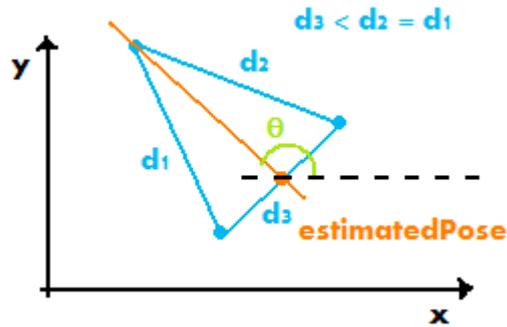


Figura 3.10: Representación del cálculo del ángulo de orientación del robot.

A partir de estos cálculos, ya es posible actualizar el valor de la variable que almacena la posición estimada del robot y disponer de ella accediendo al repositorio.

- ▲ **AcelerometerLocModule:** Proporciona la localización del robot estimada a partir del acelerómetro incorporado en él. El valor de las coordenadas  $x$ ,  $y$  y  $z$  estimadas por el acelerómetro son depositadas en el repositorio. Cada 0,5 segundos, este módulo realiza una lectura de dichas coordenadas y calcula el punto del plano ligado a ellas, asignándolo a la variable "acelemEstimatedPose". Una vez el punto es calculado, se actualiza su valor en el repositorio.

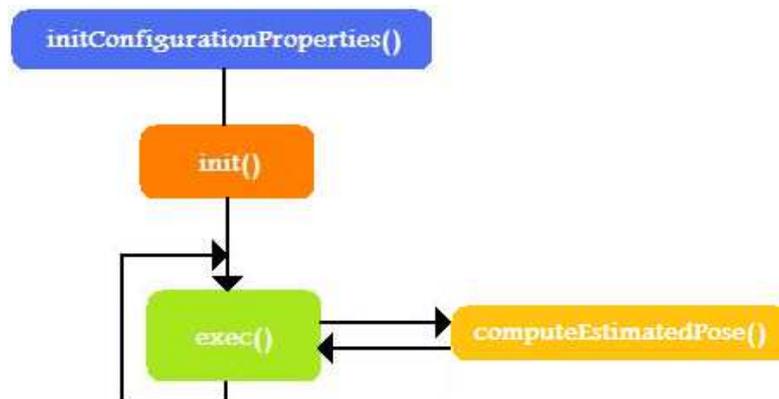


Figura 3.11: Diagrama de flujo del módulo AcelerometerModule.

En Figura 3.11 puede comprobarse las funciones de las que se compone este módulo:

- *initConfigurationProperties()*: Declara e inicializa las propiedades que van a ser utilizadas.
- *init()*: Determina el periodo de ejecución de la función *exec*. En este caso establece que dicha función se ejecute cada 0,5 segundos.
- *exec()*: Se ejecuta continuamente. Obtiene del repositorio el valor de las coordenadas de posición estimadas por el acelerómetro y ejecuta

la función *computeEstimatedPose*. Finalmente, actualiza el valor de la posición estimada albergado en el repositorio.

- *computeEstimatedPose()*: Calcula un punto del espacio con su respectiva orientación a partir de unas coordenadas.

- ^ **LocalizationModule**: Es el módulo encargado de integrar los datos de localización que aportan tanto el módulo de localización infrarroja como el módulo del acelerómetro. Realiza la lectura de los puntos estimados por cada uno de ellos y calcula el punto que será el considerado como la posición correcta del robot. El funcionamiento de este módulo puede llevarse a cabo de dos formas: cuando los datos por parte del acelerómetro no están disponibles, el software solo utiliza el punto estimado por la tecnología infrarroja y determina que la localización final es la indicada por dicho módulo. Por otro lado, cuando existen datos de posicionamiento proporcionados por el acelerómetro, se utilizan ambos puntos para el cálculo de la estimación final.

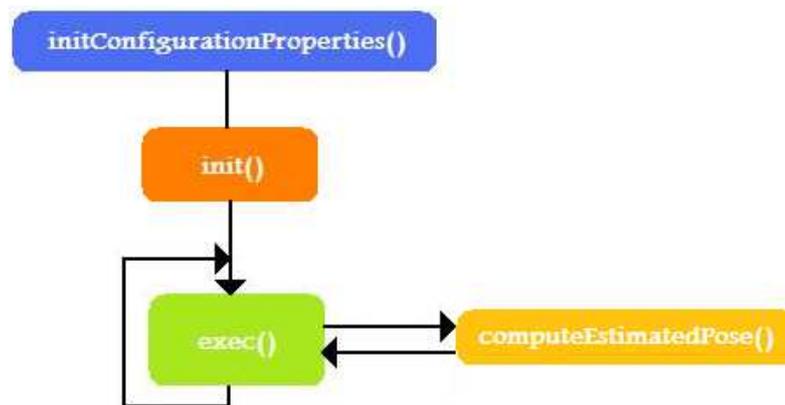


Figura 3.12: Diagrama de flujo del módulo LocalizationModule.

Tal y como muestra Figura 3.12, el módulo está formado por cuatro funciones:

- *initConfigurationProperties()*: Declara e inicializa las propiedades que van a ser utilizadas.
- *init()*: Establece que la función *exec* se ejecute cada 0,5 segundos.
- *exec()*: Se ejecuta continuamente.
- *computeEstimatedPose()*: Determina la posición del robot utilizando las estimaciones realizadas a partir del dispositivo infrarrojo y el acelerómetro.

- ^ **ControlModule**: Es el módulo principal. Se encarga de obtener la posición del robot y la posición destino y calcular los valores de velocidad lineal y angular que debe adquirir éste para alcanzar su objetivo. El algoritmo del cálculo de estas velocidades se realiza en los siguientes pasos:

- En primer lugar se realiza una lectura de los puntos de origen y destino almacenados en el repositorio.
- Una vez almacenados estos puntos con su orientación, se calcula el ángulo de error existente entre el punto inicial y el final. Como puede verse en la Figura 3.13, para realizar el cálculo vamos a definir  $p_o$  como el punto origen,  $p_f$  como el punto destino,  $\alpha$  como el ángulo existente entre la recta que une ambos puntos con el eje  $x$ ,  $\beta$  como el ángulo entre el eje del robot y el eje  $x$  y  $\gamma$  como el ángulo existente entre este eje y la línea que une  $p_o$  y  $p_f$ , llamado ángulo de error.

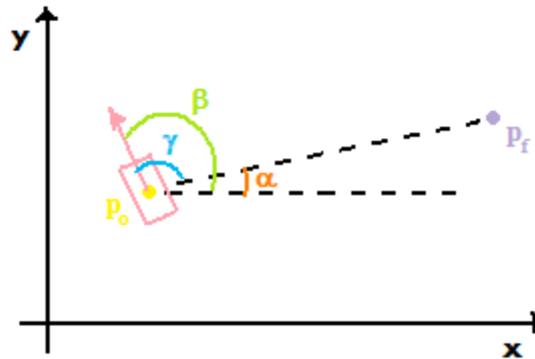


Figura 3.13: Ángulos existentes entre los puntos origen y destino.

Existen cinco posibles situaciones:

- $\alpha = \beta$ : El robot está orientado directamente al punto destino, por lo que el ángulo de error es cero.

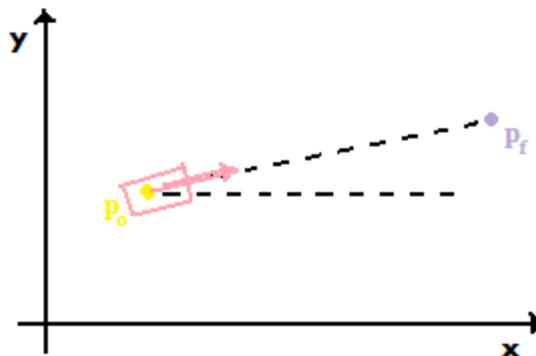


Figura 3.14: Representación del caso  $\alpha = \beta$ .

- $\alpha \leq 180$  y  $\beta \leq 180$ : Como puede verse en la imagen, el robot debe girar a la derecha para orientarse correctamente. En este caso, el ángulo de error corresponde con la diferencia entre  $\beta$  y  $\alpha$ .

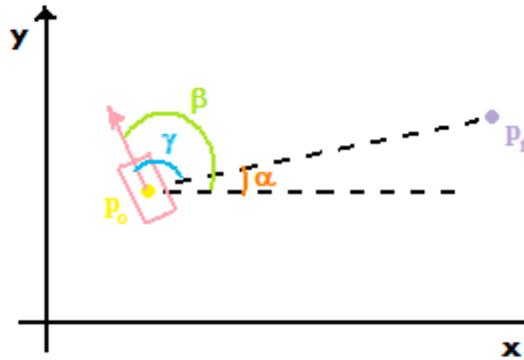


Figura 3.15: Representación del caso  $\alpha \leq 180$ ,  $\beta \leq 180$ .

$$\gamma = \beta - \alpha \quad (15)$$

- $\alpha > 180$  y  $\beta > 180$ : Se obtiene un ángulo de error negativo, por lo que se produce una rotación en sentido antihorario.

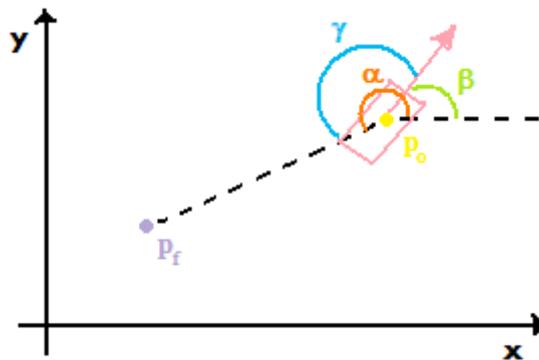


Figura 3.16: Representación del caso  $\alpha \geq 180$ ,  $\beta \geq 180$ .

$$\gamma = -(\alpha - \beta) \quad (16)$$

- $\alpha \leq 180$  y  $\beta \geq 180$ : En este caso se obtiene un ángulo de error negativo por lo que resulta óptimo que la rotación se realice en sentido antihorario ya que el ángulo a recorrer es menor en este sentido.

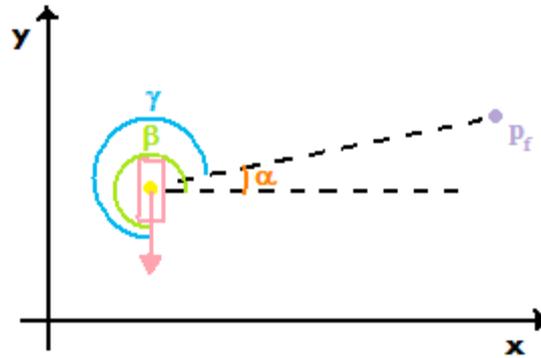


Figura 3.17: Representación del caso  $\alpha \leq 180$ ,  $\beta \geq 180$ .

$$\gamma = -(360 - \beta + \alpha) \quad (17)$$

- $\alpha \geq 180$  y  $\beta \leq 180$ : El ángulo de error es positivo por lo que se producirá una rotación en sentido horario.

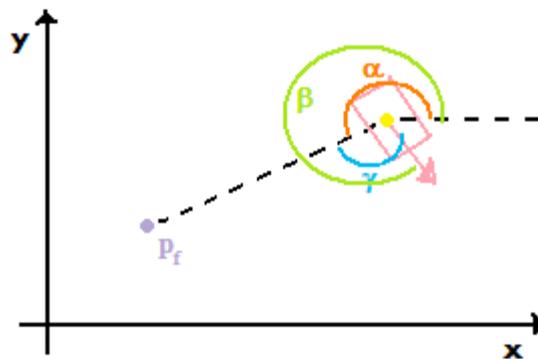


Figura 3.18: Representación del caso  $\alpha \geq 180$ ,  $\beta \leq 180$ .

$$\gamma = \beta - \alpha \quad (18)$$

- Se calcula la velocidad lineal a partir del ángulo error previamente obtenido y de los parámetros de velocidad máxima y constante de linealidad. Éste último parámetro, cuyo rango de valores oscila entre  $(0, 1]$ , es introducido para determinar la precisión o velocidad de los movimientos trazados por el robot. Como puede apreciarse en la ecuación (19) para valores cercanos a cero, se obtendrán reducidas velocidades de movimiento, por lo que el robot trazará movimientos más precisos. Por otra parte, cuando la constante de linealidad sea igual a uno, el robot podrá alcanzar la velocidad máxima, realizando movimientos más rápidos e imprecisos. La ecuación del cálculo de la velocidad lineal viene dada por:

$$v = V_{max} \left( 1 - \frac{\gamma}{180} \right) k_v \quad (19)$$

- o Se calcula la velocidad angular de la misma forma que la velocidad lineal:

$$w = W_{max} \left( \frac{\gamma}{180} \right) k_w \quad (20)$$

Por lo tanto, cuando el ángulo de error es cercano a 180°, es decir, la máxima desviación posible respecto al punto destino, obtenemos una velocidad lineal cercana a cero, pero una velocidad angular elevada. El caso contrario sucede cuando el ángulo de error es mínimo, ya que entonces el robot se encuentra casi orientado hacia el punto destino por lo que no es necesaria una importante rotación del robot.

- o Finalmente se actualiza el repositorio incluyendo los nuevos valores de velocidad angular y lineal obtenidos.

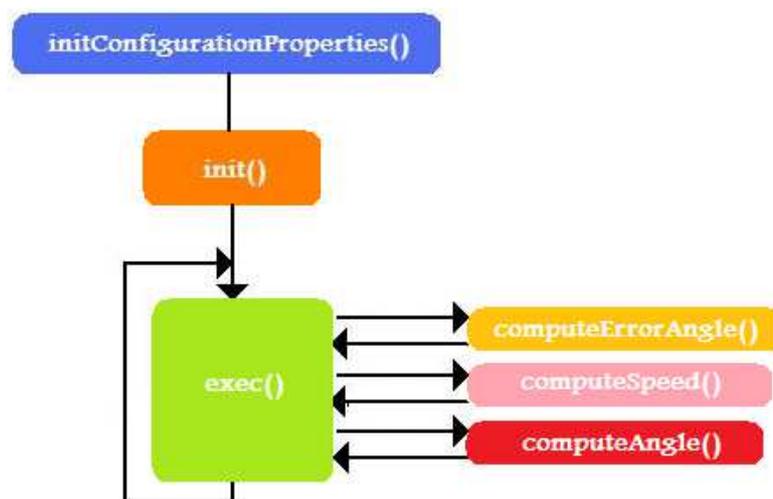


Figura 3.19: Diagrama de flujo del módulo ControlModule.

Todos estos cálculos se realizan dentro de las seis funciones que forman este módulo, tal y como puede comprobarse en la Figura 3.19:

- *initConfigurationProperties()*: Declara e inicializa las propiedades que van a ser utilizadas.
- *init()*: Establece el periodo de ejecución de *exec* en 1 segundo.
- *exec()*: Se ejecuta continuamente. Tras obtener el valor de los puntos origen y destino, ejecuta las funciones *computeErrorAngle*, *computeSpeed* y *computeAngle*. Finalmente, actualiza el valor de la velocidad lineal y angular almacenado en el repositorio.
- *computeErrorAngle()*: Calcula el ángulo de error existente entre el punto de origen y el punto destino.

- *computeSpeed()*: Determina la velocidad lineal que debe adquirir el robot para alcanzar el punto destino en función del ángulo de error.
- *computeAngle()*: Determina la velocidad angular que debe adquirir el robot para alcanzar el punto destino también en función del ángulo de error.

△ **ClientModule**: Establece la comunicación entre el ordenador, que hará la función de base, y el robot. Como se ha mencionado anteriormente, esta comunicación se realiza a través de una conexión *Bluetooth*. Para ello el software implementado en este módulo abre un puerto de comunicación entre el dispositivo *Bluetooth* del ordenador y el dispositivo *Bluetooth* instalado en el robot y configura una conexión entre ambos. Una vez establecida la conexión, se procede a un continuo intercambio de datos. Al igual que el módulo de localización, éste también actúa de dos formas diferentes dependiendo de la disponibilidad del acelerómetro. Cuando el acelerómetro no se encuentra disponible, el módulo sólo tiene que encargarse de la emisión de datos. Cuando el acelerómetro se encuentra operativo, además de la emisión de datos, debe encargarse de la recepción periódica de éstos. Los datos emitidos por parte de este módulo se corresponden con la velocidad lineal y angular a adquirir por el robot, calculadas por el módulo anterior. Los datos recibidos son cada una de las tres coordenadas del plano proporcionadas por el acelerómetro, que son depositadas directamente en el repositorio.

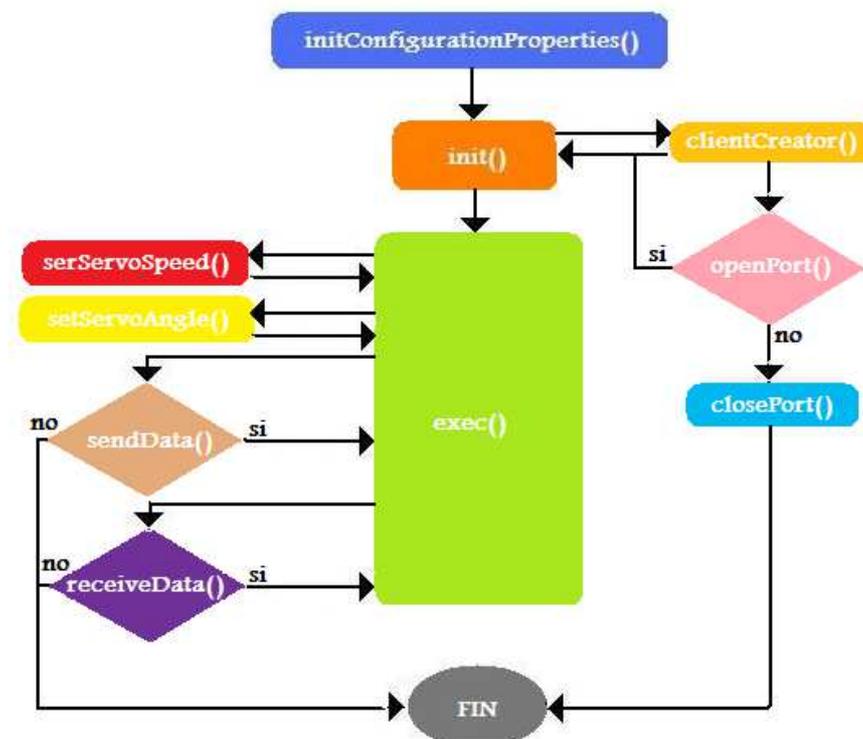


Figura 3.20: Diagrama de flujo del módulo ClientModule.

La Figura 3.20 muestra el diagrama de flujo de este módulo:

- *initConfigurationProperties()*: Declara e inicializa las propiedades que van a ser utilizadas.
- *init()*: Establece el periodo de ejecución de *exec* en 100 milisegundo y llama a la función *clientCreator()*.
- *exec()*: Se ejecuta continuamente. En primer lugar llama a las funciones *setServoSpeed()* y *setServoAngle()*. Después ejecuta la función *sendData()*. En este punto, comprueba si el acelerómetro está disponible: si lo está, pasa a ejecutar *receiveData()* y actualiza el valor de las coordenadas *x*, *y* y *z* del repositorio con los datos recibidos; si no lo está, vuelve al inicio de la función.
- *clientCreator()*: Crea el cliente para la comunicación BT donde se almacenarán los datos a enviar o recibidos. Llama a la función *openPort()*.
- *openPort()*: Comprueba si el puerto de comunicación se encuentra disponible. Si lo está, se encarga de abrir dicho puerto para establecer la conexión, si no está operativo, ejecuta la función *closePort()*.
- *closePort()*: Cierra el puerto de comunicación.
- *setServoSpeed()*: Actualiza el valor de la velocidad lineal consultando su valor en el repositorio.
- *setServoAngle()*: Actualiza el valor de la velocidad angular consultando su valor en el repositorio.
- *sendData()*: Si el puerto permanece abierto, envía por él los valores de la velocidad lineal y angular. Si la conexión no se encuentra disponible, se termina la ejecución del programa.
- *receiveData()*: Recibe a través del puerto, mientras éste esté operativo, los valores de las coordenadas determinadas por el acelerómetro.

## **4. Resultados y conclusiones**

Tras la implementación y prueba de la aplicación descrita a lo largo del documento, se puede concluir que los resultados finales han sido satisfactorios ya que se ha logrado alcanzar el objetivo inicialmente marcado desarrollando un sistema de navegación válido. En las pruebas realizadas en entornos de 2D el robot es capaz de localizarse y alcanzar el punto destino tal y como se estableció en un principio. No obstante, tras las pruebas realizadas ha sido posible detectar ciertas limitaciones de la aplicación sujetas, casi en su totalidad, a los

componentes físicos utilizados en su fabricación. Las limitaciones que han podido ser observadas a lo largo de los test realizados son:

- Área de exploración reducida: Dado que el receptor infrarrojos utilizado en la aplicación posee un rango de visión efectiva de 33 grados horizontales y 23 grados verticales, el área en el que el robot puede ser detectado queda reducido a una extensión que apenas supera el metro cuadrado. Esta limitación no viene dada solamente por el receptor infrarrojo, sino también por los emisores, ya que la intensidad irradiada por estos dispositivos no puede superar los 20 mw.
- Limitación en el envío y recepción de datos en la conexión *Bluetooth* establecida entre el ordenador y el robot: Durante las pruebas se ha observado errores en dicha comunicación en los casos en los que los datos del acelerómetro están disponibles. Como podemos recordar, cuando este dispositivo está operativo, la comunicación *Bluetooth* establecida debe permitir el flujo de datos en ambos sentidos. Cuando la comunicación recibe esta carga de datos de ambos extremos se satura y se produce una pérdida la conexión.
- Movimientos no continuos del robot: Se ha observado que la trayectoria que traza el robot hasta el punto final no es una trayectoria totalmente limpia. Esto se debe a las velocidades discretas suministradas por parte de los servomotores, además de la continua actualización y modificación de dicha trayectoria por parte del algoritmo implementado.
- Escasa diferencia entre los varios modos del algoritmo configurable: Como se ha explicado anteriormente, el algoritmo de control de movimiento permite cierta configuración que aporta modificaciones en la trayectoria a seguir en cuanto a velocidad y precisión de movimientos. Estas diferentes configuraciones son casi inapreciables en esta aplicación dado el reducido rango de velocidades aportado por los actuadores utilizados.
- Conocimiento previo de las coordenadas: Una vez fijado el receptor infrarrojos y antes de poner en funcionamiento el sistema de navegación, es necesaria la lectura de dos puntos de la superficie con el fin de establecer las relaciones entre el sistema de coordenadas dispuesto por el receptor y el área de operación en sí.

Tras estos resultados podemos concluir que la solución obtenida ha alcanzado los objetivos propuestos utilizando los componentes más básicos del mercado. Los resultados son favorables para la tecnología utilizada, sin embargo el uso del sistema de navegación en entornos no tan ideales requerirá la utilización de tecnologías más complejas que puedan resolver las limitaciones descritas. Como trabajo futuro cabe destacar la prueba del sistema de navegación en entornos 3D.

## **Referencias**

- [1] Jose Luís Pérez Díaz. *Astronomía, mecánica celeste y exploración espacial: Exploración espacial*. Marzo 2005.
- [2] Rhys Jones, Patrick Haufe, Edward Sells, Pejman Iravani, Vik Olliver, Chris Palmer, and Adrian Bowyer. *RepRap-the replicating rapid prototyper*. Robotica, January 2011.
- [3] Erik de Bruijn. *On the viability of the open source development model for the design of physical objects. Lessons learned from the RepRap project*, November 2010.
- [4] Julián Marín Mato. *Control remoto de un robot imprimible de exteriores. Proyecto fin de carrera*. Universidad Carlos III de Madrid. Diciembre 2011.
- [5] Luis Andreula. *Comunicaciones Infrarrojas*.
- [6] Programa Sentinel. *Sistema flexible de localización en interiores basado en redes de sensores ZigBee. E10 Localización en interiores*. Tecnológico Fundación Deusto. Programa Ekinberri 2008.
- [7] Rodrigo Larenas Torrealba. *Aplicaciones de la tecnología Infrarroja*. Revista General de Marina. 1999.
- [8] Carlos Iván Loja Cajas, Juan Pablo Moscoso Moscoso. *Diseño y construcción de un prototipo de sistema de posicionamiento local en tiempo real mediante red inalámbrica 802.11 WI-FI*. Tesis Pregrado. Ingeniería Electrónica CUE. Diciembre 2009.

- [9] Douglas A. Skoog, Stanley R. Crouch, F. James Holler. Principios de Análisis Instrumental. Cengage Learning. Sexta Edición. 2008.
- [10] Álvaro Sánchez Miralles. Sensores y actuadores. Octubre 2006.
- [11] Barrientos, A., Penñín, L. F., Balaguer, C., y Aracil, R. (1997). Fundamentos de robótica. McGraw-Hill, Interamericana de España, S.A.
- [12] Francisco Martín Archundia Papacetzzi. Wireless Personal Area Network (WPAN) & Home Networking. Proyecto fin de carrera. Universidad de las Américas Puebla (Méjico). Diciembre 2003.
- [13] Nelson Rolando Cárdenas Calle, Galo Rene Simbaña Quinde. Diseño y construcción de un robot móvil didáctico para estudios experimentales de robótica cooperativa. Tesis pregrado. Ingeniería Electrónica CUE. Agosto 2007.
- [14] Juan González-Gómez, Andrés Prieto-Moreno, Itziar Lima, Ricardo Gómez. Proyecto SkyMega,
- [15] Proyecto Arduino, <http://www.arduino.cc>. Consulta en Agosto de 2012.
- [16] J. A. Martín Pereda. Sistemas y redes ópticas de comunicaciones. Pearson, 2004.
- [17] Ricardo Bonache Samaniego. Wiimote Hack. Proyecto fin de carrera. Universitat Rovira i Virgili. Junio 2009.
- [18] Proyecto Source Forge, <http://openrdk.sourceforge.net>. Consulta en Agosto de 2012.