

Procesamiento de la información recibida de un sensor láser ubicado en un vehículo tipo Quad

PROYECTO FINAL DE CARRERA

ALUMNO: Juan Alonso Tévar Ruiz
TUTOR: Enrique J. Bernabéu Soler

Ingeniería Informática
Escuela Superior de Ingeniería Informática (ETSINF)
Universidad Politécnica de Valencia (UPV)

CURSO 2011/2012

AGRADECIMIENTOS

A mis padres, por brindarme la oportunidad de cursar esta titulación, encargarse de mí y construirme como persona. A ellos les debo el haber llegado tan lejos.

A mi novia, por su apoyo incondicional y su infinita paciencia; por aguantar todo este tiempo a mi lado dándome ánimos y fuerzas para continuar.

A mis profesores, por ilustrarme en este mundo tan complicado que es la informática. Por la formación que me han aportado y los conocimientos que me han brindado.

A mis compañeros, porque todos estos años lejos de casa ellos han sido mi familia. Porque sin ellos los años de universidad no hubieran tenido sentido fuera de las aulas. Por los buenos momentos juntos y los ratos que aún nos quedan.

Al departamento, por dejarme formar parte de este proyecto que tanto me sedujo desde el primer momento.

INDICE

Resumen	5
Capítulo 1. El problema a tratar	6
Capítulo 2. Estado del arte.....	7
2.1 Introducción al concepto de robots móviles autónomos	7
2.2 Sistemas de control de robots	8
Capítulo 3. Especificación.....	12
3.1 Análisis de requisitos	12
3.2 Especificación del sistema.....	12
3.3 Planificación y estimación de costes.....	14
Capítulo 4. Desarrollo del proyecto	16
4.1 Análisis	16
4.2 Diseño	18
4.3 Implementación.....	20

Procesamiento de la información recibida de un sensor láser ubicado en un vehículo tipo Quad

Capítulo 5. Pruebas y resultados	22
5.1 Descripción de Experimentos.....	22
5.2 Resultado de las pruebas	23
Capítulo 6. Conclusiones y trabajo futuro.....	24
6.1 Conclusiones.....	24
6.2 Trabajo futuro.....	25
Referencias	26
Índice de tablas y figuras	27
Anexos.....	28

RESUMEN

El siguiente proyecto nace en el Departamento de Sistemas y Automática (DISA) de la Universidad Politécnica de Valencia (UPV), de la idea de dotar un vehículo quad de autonomía. El trabajo se divide en dos proyectos principales: la primera parte consiste en usar un telémetro láser para adquirir y procesar señales del entorno, para después analizarlas y trazar una trayectoria segura. La segunda parte se trata de tomar esta información acerca de la trayectoria y dotar al vehículo de los mecanismos adecuados para trazarla correctamente. En este caso el proyecto está vinculado únicamente a la primera parte, a la adquisición y procesamiento de la información recibida del sensor láser.

El problema a tratar surge de la dificultad que tienen los autómatas de tomar sus propias decisiones, dado que su capacidad de autonomía reside en el programa que ejecuta el robot. Se puede programar el dispositivo para que trace una ruta predeterminada, pero se corre el riesgo de encontrar algún tipo de obstáculo que impida que la tarea llegue a buen fin. De ahí que se piense en incorporar sensores capaces de interactuar con el medio de un modo *on-line* o dicho de otro modo, a medida que el robot se mueva por el entorno. Esto permitirá al autómata tomar la decisión más adecuada ante situaciones imprevistas.

De esta manera, nos encontramos en un caso en el que tenemos que saber interpretar las señales leídas por los sensores y determinar cómo debe actuar nuestro autómata en consecuencia. La manera de abordar esta situación depende del programador, aunque la idea está clara: evitar que el móvil colisione con los objetos que le rodean encontrando un camino libre por el cual poder transitar. Como este proyecto se centra en la primera parte del objetivo final, no se podrá probar sobre el vehículo en un caso real, de manera que se ha generado una interfaz gráfica con un visor donde se representarán los datos adquiridos mediante el sensor. Dicha interfaz incorporará botones que permitirán al usuario definir el rango de lectura así como la precisión de las medidas.

El presente documento recoge las fases por las que ha pasado el proyecto, desde la concepción de la idea hasta las conclusiones obtenidas del mismo y que demuestran la evolución del proyecto.

Capítulo 1.- El problema a tratar

El objetivo del presente proyecto consiste en la implementación de un software de recepción y tratamiento de los datos enviados por un telémetro láser modelo SICK LMS 220. Dicho software está orientado a dotar de autonomía un vehículo de tipo quad, un prototipo que está siendo desarrollado por el Departamento de Ingeniería de Sistemas y Automática (DISA).

El problema de la cinemática del vehículo no será tratado en este proyecto, de manera que en este caso trataremos únicamente la información de un modo off-line. No obstante, se tendrán en cuenta las dimensiones del vehículo a la hora de hacer los cálculos. La idea que se persigue es recibir las señales del dispositivo láser, tratarlas y elegir una dirección que permita al vehículo seguir avanzando sin toparse con ningún obstáculo.

Dado que hay que tratar con la información de manera off-line, se diseñará una interfaz con el fin de interactuar con el dispositivo así como de visualizar los resultados

Capítulo 2. Estado del Arte

2.1. Introducción al concepto de Robots móviles autónomos

En mecánica, el *movimiento* se define como un cambio de posición en el espacio por parte de un sistema o de alguno de sus elementos con respecto a ellos mismos o a una referencia. En referencia a los sistemas robotizados, surge el concepto de *movilidad*, que denota la capacidad del robot para realizar dichos movimientos, ya sea mediante traslación o mediante la rotación del dispositivo o de alguna de sus partes. El primero de los casos implica un desplazamiento, y es lo que se asocia al concepto de *robot móvil*.

Los robots móviles pueden desplazarse siguiendo una trayectoria programada o interactuando con el medio, en cuyo caso se habla de *autonomía*. Para que un robot sea autónomo debe ser capaz de tomar decisiones en función de la información suministrada por los sensores de acuerdo a una estrategia de actuación. Uno de los aspectos más importantes de la robótica móvil es la *navegación*, o técnica que permite al móvil desplazarse de un punto a otro, salvando los obstáculos que pueda haber por el camino, ya sea en un entorno predefinido o en uno desconocido.

El proceso de la navegación se podría dividir en tres subprocesos, atendiendo a la intencionalidad del móvil de encontrar un camino que le lleve del punto origen al punto destino:

- Proceso de percepción: reconocimiento de elementos característicos del entorno que permiten obtener un modelo del mismo, mediante el empleo de información previa y/o de la proporcionada por sensores externos
- Proceso de planificación: partiendo del modelo anterior, emplear una estrategia adecuada para la obtención de un camino libre de obstáculos que lleve a la meta (la meta puede plantearse como un punto intermedio del recorrido, de manera que se divide el problema en sub-problemas fácilmente abordables), así como de las trayectorias a seguir.
- Proceso de actuación o reacción: seguimiento del camino y reajuste de la ruta seguida en función de los cambios en el entorno, lo que

Procesamiento de la información recibida de un sensor láser ubicado en un vehículo tipo Quad

implica hacer frente a circunstancias imprevistas haciendo uso de los sensores para adquirir información en tiempo real.

La navegación y la localización es un aspecto muy importante y en el que se ha investigado mucho en los últimos años. El problema de los sistemas de localización y mapeo simultáneo (Simultaneous Localization and Mapping, SLAM), es su dificultad para la implantación en aplicaciones reales. Estos sistemas intentan resolver de manera concurrente dos problemas: construir un mapa del entorno, identificando las zonas libres y las ocupadas y localizarse en el mapa que se está construyendo. En el caso que nos ocupa, se construirá un *mapa de visión* a partir de las mediciones que el sensor láser obtenga del entorno.

La idea general en la planificación de trayectorias consiste en desvincular el problema de la cinemática y dinámica del robot del de la obtención de una ruta (óptima o no) libre de obstáculos.

2.2. Sistemas de control de robots

Los robots son controlados generalmente por programas que previamente han sido implementados en algún tipo de lenguaje de programación por el desarrollador. Estos programas dicen al robot lo que tiene que hacer en todo momento, como moverse hacia delante o detenerse porque hay un obstáculo. Desde los inicios de la robótica, estos programas siempre han estado limitados a lo que el diseñador haya especificado en ellos, por lo el robot siempre ejecuta las mismas acciones, no es capaz de resolver nuevas situaciones por sí solo.

Para aumentar el grado de autonomía de los robots e intentar que estos lleguen a simular diferentes comportamientos, cada vez existen más dispositivos sensoriales capaces de medir diferentes aspectos del entorno del robot, como por ejemplo la distancia hasta un obstáculo, la temperatura del ambiente, o la posición global por citar algunos de ellos.

Dejando a un lado las capacidades de los robots en cuanto a autonomía y comportamiento se refiere, todo sistema de control de un robot se basa en los mismos conceptos, los cuales permiten construir sistemas muy sencillos o sistemas realmente complejos y avanzados. En este apartado se describe brevemente los conceptos básicos por los que se rige cualquier sistema de control de robots, con el fin de comprender mejor alguno de los objetivos de este proyecto.

Procesamiento de la información recibida de un sensor láser ubicado en un vehículo tipo Quad

2.2.1. Sistemas de bucle abierto

Los sistemas de control de bucle abierto, son aquellos sistemas en los que no existe una retroalimentación, es decir, que no se utiliza la información de salida para reajustar la entrada de los datos al controlador. Un ejemplo del uso de este tipo de sistemas, sería por ejemplo si el programa de control estuviera diseñado para hacer que el robot se mueva hacia delante continuamente, sin comprobar nada.



Figura 2.1 – Sistema de bucle abierto

2.2.2. Sistemas de bucle cerrado

Los sistemas de control de bucle cerrado, son aquellos sistemas en los que existe una retroalimentación, es decir, que los datos obtenidos a la salida del controlador se utilizan de nuevo para realizar ajustes en las siguientes iteraciones del programa de control. Esto resulta muy útil cuando se necesita conocer el resultado de ciertas acciones para realizar las siguientes. Estos sistemas se utilizan con los robots sensorizados, ya que se utiliza la información de los sensores para reajustar ciertos aspectos en el comportamiento del robot.

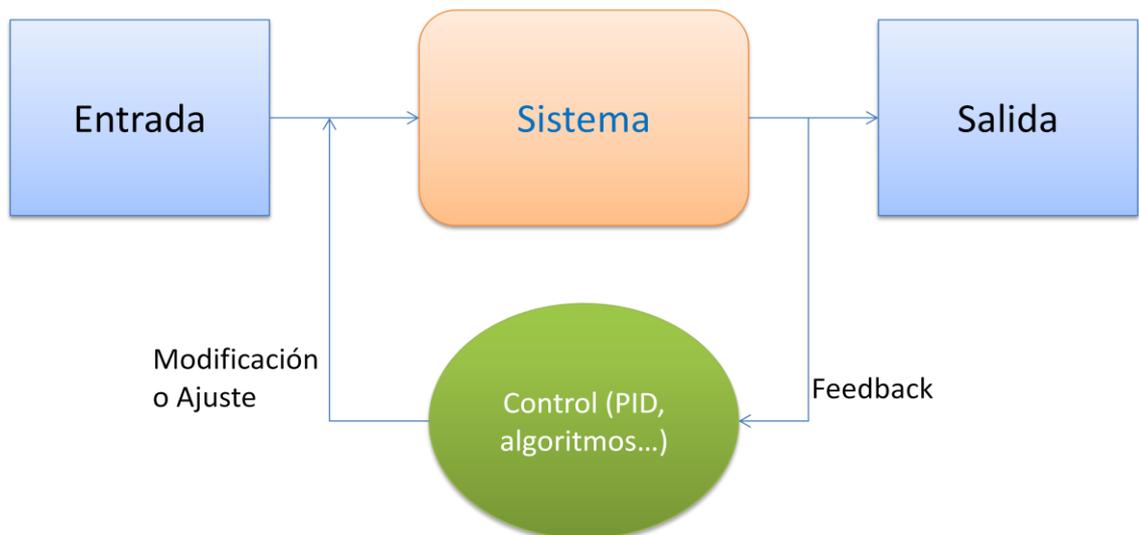


Figura 2.2 – Sistema de bucle cerrado

Procesamiento de la información recibida de un sensor láser ubicado en un vehículo tipo Quad

2.2.3. Métodos de planificación.

Se habla de *espacio de trabajo* cuando se hace referencia a los posibles casos en los que el robot se puede encontrar en el entorno en el que se va a mover. Tomando el móvil como actor principal, la interacción entre él y el medio que le rodea conforma la *configuración* existente, esto es, la posición y orientación del robot en el entorno. Al conjunto de configuraciones posibles se le llama *espacio de configuraciones*. Este conjunto se puede dividir en dos grupos o estados: configuración *libre* cuando el móvil no colisiona con ningún obstáculo y configuración en *colisión* en caso contrario. La planificación se encarga de encontrar una serie de estados en configuración libre que permitan a la máquina moverse del punto de origen al punto destino.

2.2.4. Construcción de mapas.

Para encontrar esta serie de estados libres, el robot se sirve de los sensores, con los que obtiene información del medio. Con esta información construirá un mapa donde se determina los puntos en el espacio en los que se puede presentar una configuración libre. Existen varias formas de generar estos mapas, entre ellas:

- Mapas métricos: usan figuras geométricas para representar una aproximación del entorno. En este grupo se distinguen los mapas *geométricos* que usan puntos, rectas y círculos a partir de los datos de medida y los mapas de *rejilla*, que consisten en una descomposición del entorno en celdas de área definida, determinando que áreas presentan una configuración libre y qué áreas presentan una configuración en colisión.

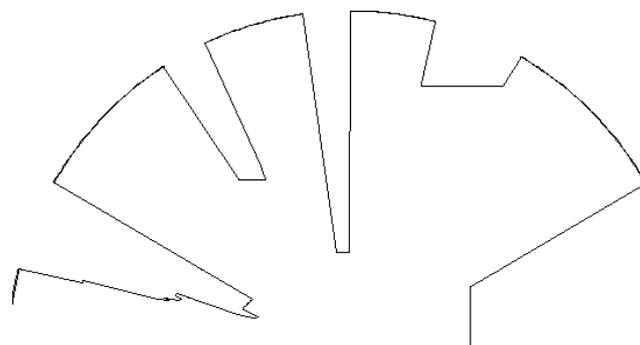


Figura 2.3 – Mapa métrico

Procesamiento de la información recibida de un sensor láser ubicado en un vehículo tipo Quad

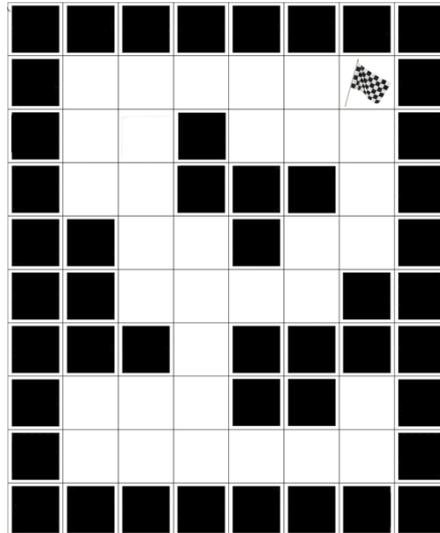


Figura 2.4 – Mapa de rejilla

- Mapas topológicos: Este tipo de mapas representan una serie de posiciones en el espacio libre en forma de nodos, que se unen después mediante arcos. Este método permite representar el entorno en forma de grafo, optimizando los cálculos de las trayectorias. En este aspecto aportan un beneficio, pero se basan en conocimientos “a priori” del entorno en que se moverán y no sólo en la información obtenida por los sensores, por lo que se suelen utilizar en entornos conocidos.
- Mapas semánticos: Se trata de una representación abstracta del entorno. Se etiqueta las partes más representativas del mismo con nombres que las identifican y permiten clasificarlas según la información almacenada. Cognitivamente, se asemeja más a la percepción humana, al etiquetar los elementos con nombres comunes (pared, puerta, suelo...).

Capítulo 3. Especificación

3.1. Análisis de requisitos

El objetivo de este proyecto es crear un software capaz de recoger, analizar e interpretar las señales enviadas por un sensor láser. Además, el programa debe ser capaz de decidir la dirección a la que se debe dirigir un robot móvil. Dado que no es posible integrar este software en el robot, se creará una interfaz gráfica con el fin de interactuar con el telémetro además de visualizar los resultados obtenidos del tratamiento de los datos.

Dicha interfaz incluirá controles que permitan al usuario definir la resolución, la distancia mínima que tomaremos como referencia para efectuar los cálculos, la anchura del vehículo y el criterio de selección de salidas, así como una pantalla donde se representará la información obtenida por el mismo. El programa debe mostrar las lecturas obtenidas además de la solución elegida por el mismo.

3.2. Especificación del sistema

El punto de partida es un vehículo de tipo quad, que desarrollado por el Instituto de Automática e Informática Industrial (ai2) de la Universidad Politécnica de Valencia. En la actualidad el vehículo está siendo modificado por el departamento para dotarlo de autonomía y no puede ser usado para implantar el software, de manera que solo se tendrán en cuenta sus medidas para la toma de decisiones. El sensor láser está pensado para colocarse en la parte delantera del dispositivo, de modo que pueda detectar los obstáculos que pueda haber frente al mismo.

3.2.1. Telémetro láser

El telémetro láser es el modelo SICK LMS 291-S05 (Figura 1), capaz de leer un plano de 2 dimensiones con una resolución desde 1° hasta 0.5° , y con un campo de visión configurable hasta 180° . La distancia máxima es regulable desde los 0m hasta los 80m. El periodo mínimo de muestreo del láser es de 13ms. Los detalles se detallan en las siguientes tablas

Procesamiento de la información recibida de un sensor láser ubicado en un vehículo tipo Quad

CARACTERISTICAS	
Campo de aplicación:	Exterior
Versión:	Rango Medio
Divisor:	Infrarrojo (905 nm)
Clase de láser:	1 (EN/IEC 60825-1), no perjudicial para la vista
Ángulo de abertura:	180 °
Frecuencia de exploración:	75 Hz
Resolución angular:	0,25°
	0,5°
	1°
Calefacción:	Opcional, mediante placa calefactora externa
Rendimiento:	0 m ... 80 m
Rendimiento máximo con reflectividad al 10%:	30 m
Tipo de código admitido:	Sí
MTBF:	50.000 h

Tabla 3.1 – Características del telémetro

RENDIMIENTO	
Tiempo de respuesta:	≥ 13 ms
Forma de objeto detectable:	Prácticamente, todas
Error sistemático:	± 35 mm
Error estadístico:	± 10 mm
Aplicación integrada:	Evaluación de campo
Número de registros de campo:	2 campos triples (6 campos)
Simultaneous processing cases:	1 (3 campos)

Tabla 3.2 – Rendimiento del telémetro

MECANICA / ELECTRONICA	
Conexión eléctrica:	2 Conector de interfaz con tomacorriente D-Sub de 9 pines (conexión soldada)
Tensión de operación:	≥ 24 V DC± 15 %
Consumo de energía:	30 W
Caja:	Fundición de aluminio
Color de caja:	Gris (RAL 7032)
Grado de protección:	IP 65
Clase de protección:	II (VDE 0106/IEC 1010-1) ¹⁾
Peso:	4,5 kg
Dimensiones:	156 mm x 155 mm x 210 mm

Tabla 3.3 – Mecánica/electrónica del telémetro

Procesamiento de la información recibida de un sensor láser ubicado en un vehículo tipo Quad

3.2.2. Interfaz gráfica

El software diseñado es una aplicación MFC (Microsoft Foundation Clases) desarrollada en un PC cuyo sistema operativo es Windows XP y el entorno de desarrollo Visual Studio 2008; el lenguaje de programación empleado en todas las modelizaciones realizadas es C++. La elección de estas características se debe fundamentalmente a que durante el último año de mi licenciatura, en la intensificación de Informática Industrial usamos este entorno en varias asignaturas y los resultados obtenidos fueron satisfactorios. Además, la librería MFC se puede entender como una serie de librerías C++ que encapsulan la funcionalidad de aplicaciones hechas para un entorno como Windows. Por otro lado, al ser de alto nivel, se puede incluir una interfaz gráfica amigable e intuitiva, haciendo más fácil su uso por parte del usuario.

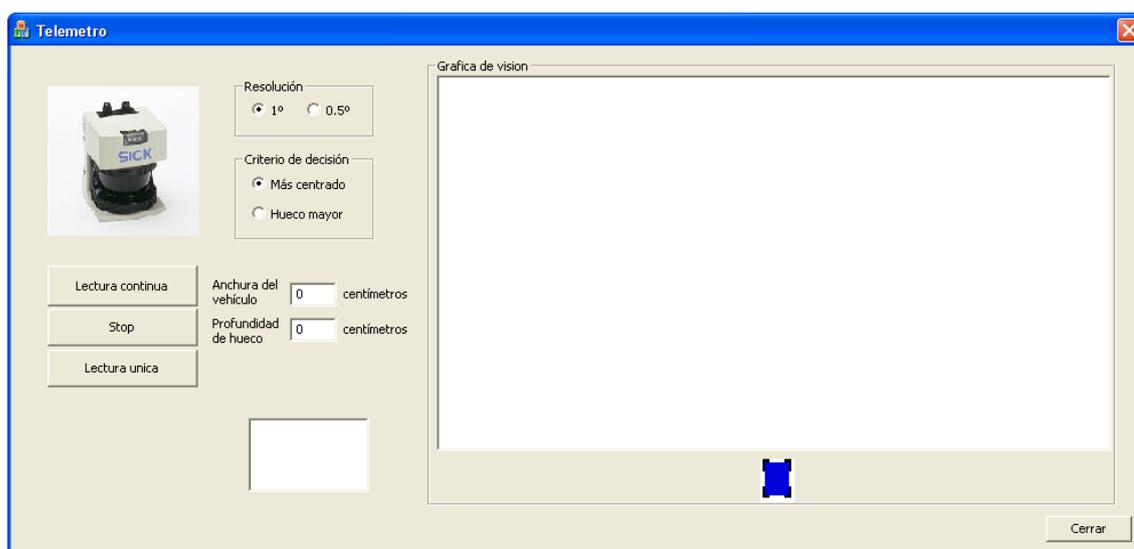


Figura 3.1 – Interfaz del programa

3.3. Planificación y estimación de costes

Dividiremos el proyecto en varias partes:

3.3.1. Recepción de los datos

Lo primero que tenemos que conseguir es establecer una conexión estable con el dispositivo y una vez establecida, recoger e interpretar las señales enviadas por el sensor. Tenemos como base unos drivers en C que reciben la información del dispositivo y la vuelca en consola. Se modificarán estos drivers para permitirnos enviar tramas al sensor de acuerdo al modo de lectura que queramos y posteriormente almacenar los datos en un vector para luego

Procesamiento de la información recibida de un sensor láser ubicado en un vehículo tipo Quad

trabajar con ellos. Además hay que configurarlos para que lean la información del puerto serie correspondiente. **15%**

3.3.2. Análisis de la información

Una vez almacenados los datos, los usaremos para realizar los cálculos oportunos para encontrar una solución. **10%**

3.3.3. Toma de decisiones

Una vez recibidos y tratados los datos, tenemos que idear un algoritmo que nos permita contemplar todos los caminos posibles para después elegir el más apropiado. Se implementarán dos métodos de selección de camino: por el hueco más grande y por el hueco más centrado. **25%**

3.3.4. Construcción de la interfaz

Para poder comunicarnos con el sensor y entender los resultados, diseñaremos una interfaz gráfica, con una serie de botones y una gráfica donde se mostrarán tanto la lectura como la posible solución calculada por nuestro programa. **40%**

3.3.5. Memoria y documentación

Una vez finalizado el software, se documentará con detalle todas las tareas realizadas. **10%**

CAPITULO 4. DESARROLLO DEL PROYECTO

4.1.- ANÁLISIS

El objetivo de este proyecto es crear un software capaz de recoger, analizar e interpretar las señales enviadas por un sensor láser. Además, el programa debe ser capaz de decidir la dirección a la que se debe dirigir un robot móvil en función de estas señales.

Dado que no es posible integrar este software en el robot, se creará una interfaz gráfica con el fin de interactuar con el telémetro además de visualizar los resultados obtenidos del tratamiento de los datos. Dicha interfaz incluirá controles que permitan al usuario definir la resolución, la profundidad mínima de hueco y la anchura mínima del mismo, así como una pantalla donde se representará la información obtenida por el mismo.

El software estará compuesto por un programa principal que actuará como intermediario entre el usuario y el dispositivo láser. El método de establecer esta comunicación será una interfaz gráfica. Por otro lado, el programa se comunicará con el sensor a través de una subrutina, para encapsular por un lado la interfaz de usuario y por otro el intercambio de datos entre la aplicación y el láser. Por último se dispondrá de una serie de funciones y algoritmos destinados a tratar los datos y obtener la solución esperada.

4.1.1.- Casos de uso

En nuestro caso, interviene únicamente un actor que es el usuario que ejecuta el programa e interactúa con él. Dispone de tres casos de uso a ejecutar: Lectura continua, Lectura única y Paro de lectura continua.

#1. Caso de uso: Lectura continua
Actores: Usuario
Precondición: El láser no está en funcionamiento
Flujo de eventos: 1.- El usuario puede elegir alguna de las opciones que presenta la interfaz como precisión de lectura o lectura máxima, o por el contrario, puede tomar las opciones por defecto y no cambiar nada. 2.- El usuario acciona el botón de <i>Lectura continua</i> situado en la parte

Procesamiento de la información recibida de un sensor láser ubicado en un vehículo tipo Quad

<p>izquierda de la interfaz.</p> <p>3.- La aplicación deshabilita la parte de configuración de precisión y lectura máxima</p> <p>4.- La aplicación muestra en la parte derecha una gráfica con los datos leídos y la solución elegida. La gráfica se actualiza constantemente con los datos actuales del sensor.</p>
<p>Excepciones:</p> <p>3.1.- La aplicación detecta un error a la hora de abrir el puerto para comunicarse con el sensor y muestra un mensaje de aviso. No mostrará nada en la parte gráfica</p> <p>3.2.- La aplicación detecta un error en los datos devueltos por el sensor y muestra un mensaje de aviso. No mostrará nada en la parte gráfica.</p>

Tabla 4.1 – Caso de uso #1

<p>#2. Caso de Uso: Lectura única</p>
<p>Actores: Usuario</p>
<p>Precondición: El láser no está en funcionamiento</p>
<p>Flujo de eventos:</p> <p>1.- El usuario puede seleccionar la precisión y el rango de lectura o quedarse con la configuración por defecto.</p> <p>2.- El usuario acciona el botón <i>Lectura única</i> situado en la parte derecha de la interfaz</p> <p>3.- La aplicación muestra en la parte derecha una gráfica con los datos leídos y la solución encontrada.</p>
<p>Excepciones:</p> <p>3.1.- La aplicación detecta un error a la hora de abrir el puerto para comunicarse con el sensor y muestra un mensaje de aviso. No mostrará nada en la parte gráfica</p> <p>3.2.- La aplicación detecta un error en los datos devueltos por el sensor y muestra un mensaje de aviso. No mostrará nada en la parte gráfica.</p>

Tabla 4.2 – Caso de uso #2

<p>#3. Caso de Uso: Paro de Lectura</p>
<p>Actores: Usuario</p>
<p>Precondición: El programa se encuentra en modo <i>Lectura continua</i>. El láser se encuentra en funcionamiento.</p>
<p>Flujo de eventos:</p> <p>1.- El usuario acciona el botón <i>Stop</i> situado en la parte izquierda de la interfaz.</p> <p>2.- La aplicación habilita de nuevo los botones de selección de precisión y el área de distancia máxima.</p> <p>3.- En el área de la parte derecha permanecerá la última gráfica mostrada.</p>

Tabla 4.3 – Caso de uso #3

Procesamiento de la información recibida de un sensor láser ubicado en un vehículo tipo Quad

4.2.- DISEÑO

Como se ha definido en el análisis, el software contará con 3 módulos principales: interfaz de usuario, comunicación con el dispositivo y algoritmos de toma de decisiones.

4.2.1.- Interfaz de usuario:

La interfaz se basa en las librerías MFC (Microsoft Foundation Classes). Estas librerías constituyen un paquete bastante potente con el que crear interfaces amigables e intuitivas para el usuario. Nuestra interfaz consta de tres botones en la parte izquierda, un par de grupos de botones radiales junto a un campo de texto en la parte central y un área destinada a albergar una gráfica en la parte izquierda.

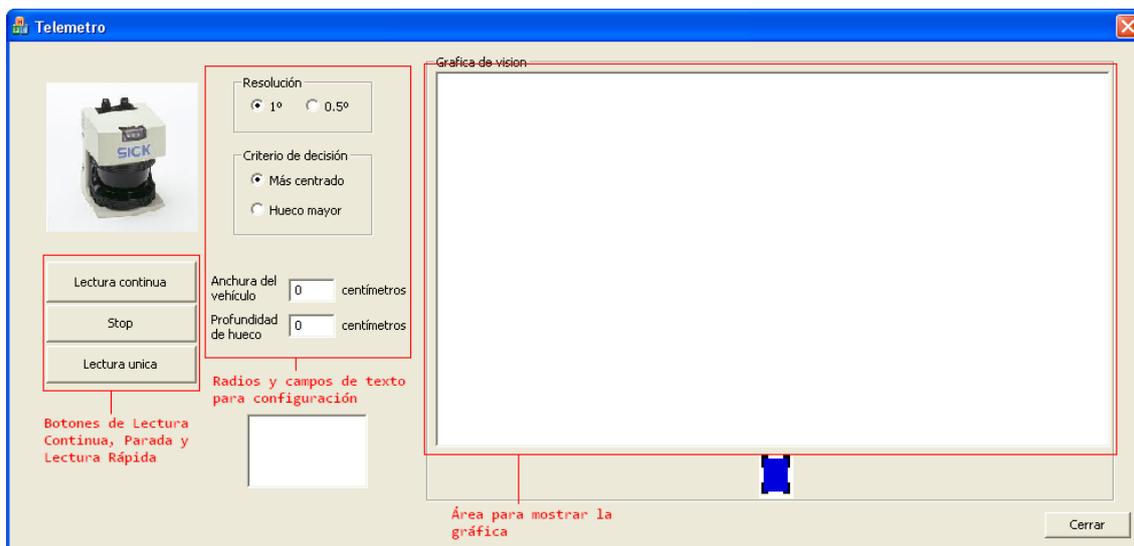


Figura 4.1 –Interfaz gráfica detallada

4.2.2.- Comunicación con el dispositivo.

Para comunicarse con el telémetro láser, partimos de unos drivers proporcionados por el departamento y desarrollados en lenguaje C para un dispositivo de características similares, de la misma marca pero un modelo anterior. Apoyándonos en ellos y adecuándolos a las características de este modelo, se creará un subproceso que se ejecutará paralelamente a la interfaz, con el fin de conseguir que la lectura no acapare el foco de la aplicación y la interfaz siga recibiendo acciones por parte del usuario, a la vez que lee continuamente los datos enviados por el sensor. Este hilo será el encargado de abrir el puerto de conexión con el sensor, enviar los parámetros de configuración adecuados al telémetro y recoger los datos generados por el mismo. Una vez obtenidos los datos, se almacenarán en 2 estructuras: Una es

Procesamiento de la información recibida de un sensor láser ubicado en un vehículo tipo Quad

un vector que almacena la distancia recorrida por cada rayo (el láser emitirá un rayo cada $0,5^\circ$ o 1°) y el valor del ángulo en el que se emite el rayo. La otra es un vector de puntos resultado de transformar los valores de distancia y ángulo a coordenadas cartesianas. Esta parte está desarrollada en C/C++, y no usa más que estructuras estándar de datos y operaciones básicas de lectura y escritura en ficheros.

4.2.3.- Algoritmos de toma de decisiones.

Tenemos un último módulo donde encontramos funciones de tratamiento y algoritmos de toma de decisiones sobre los datos obtenidos. Este módulo está desarrollado en C++ y su funcionamiento se divide en dos partes fundamentales. Una parte obtiene los huecos libres de obstáculos a la distancia definida por el usuario. Una vez obtenidos, se pasan a la segunda parte, que se encarga de decidir que hueco es el más adecuado basándose en uno de los dos criterios a decidir por el usuario: el hueco más grande o el más centrado.

4.2.3.1.- Obtención de huecos

La manera de decidir qué huecos son susceptibles de formar parte de una solución se basa en las medidas obtenidas por cada rayo. En primer lugar, declaramos una estructura *Hueco* que consiste en un punto *inicio*, un punto *fin* y un valor en coma flotante *anchura*. Antes de empezar, el usuario definirá mediante uno de los cuadros de texto de la interfaz la distancia más corta a la que se puede dirigir el vehículo. Esta distancia será el discriminante que determinará si un punto es o no un obstáculo. Tomaremos cada una de las medidas obtenidas y si la distancia es mayor a la definida por el usuario, asumiremos que se puede pasar. De este modo, recorreremos en orden el conjunto de valores devueltos por el láser y el primero que cumpla la condición de ser mayor que el valor de referencia lo tomaremos como *inicio* del hueco y nos quedaremos con los valores cartesianos del punto anterior (el último que supone un obstáculo). Seguiremos recorriendo los valores hasta encontrar un punto cuya distancia sea menor a la de referencia, tomaremos esta medida como *fin* del hueco y nos quedaremos con los valores cartesianos de este punto. De este modo, tenemos dos puntos *inicio* y *fin* en el espacio cartesiano que representan los bordes del espacio por donde podríamos pasar. La distancia entre ellos representa la *anchura* del hueco. Eso sí, antes de dar este hueco por válido, tenemos que comprobar si la anchura del hueco es suficiente para que el vehículo pueda pasar sin problemas. La validación de este hueco pasa nuevamente por comparar la anchura con un parámetro proporcionado por el usuario mediante el campo de texto correspondiente en la interfaz. Una

Procesamiento de la información recibida de un sensor láser ubicado en un vehículo tipo Quad

vez validado se almacena en un vector, si lo rechazamos por tener una anchura insuficiente se descarta. Posteriormente seguimos recorriendo el conjunto de valores en busca del siguiente punto que esté más alejado del vehículo que la distancia mínima, que tomaremos como *inicio* del siguiente hueco.

4.2.3.2.- Decisión de hueco óptimo

Aunque se vayan a representar todos los huecos que pueden formar parte de la solución, el algoritmo decidirá cuál es el más apropiado. El usuario elegirá en la interfaz uno de los dos criterios disponibles, el hueco más grande o el más centrado. La obtención del hueco más grande es inmediata, puesto que la estructura *Hueco* almacena tanto los puntos de inicio y fin como la anchura. En este caso basta con quedarse con el hueco que presente un valor de anchura mayor. En el caso del hueco más centrado, basta con calcular el punto medio de cada uno de los huecos, quedándonos con el que presente un valor X en coordenadas cartesianas más cercano a 0, puesto que el origen de coordenadas se ubica en la posición del láser.

4.3.- IMPLEMENTACION

El primer problema al que nos enfrentamos es establecer una vía de comunicación entre el PC y el dispositivo. Para ello se dispone de un cable RS232 que conecta el puerto del PC de este tipo con la entrada/salida de datos del telémetro. Internamente el programa crea un descriptor de fichero mediante el cual escribirá tramas de comando y recibirá los datos resultantes de las lecturas efectuadas.

Una vez establecida la conexión, se presenta el problema de saber interpretar la información proporcionada por el láser. Este dispositivo devuelve por el puerto RS232 una serie de datos correspondientes a la distancia desde el dispositivo hasta el objeto más cercano. Por tanto, devolverá un valor por cada rayo emitido por el dispositivo. Por ejemplo, leyendo en un rango de 180° con una resolución de 1°, obtendremos 181 valores (0° ~ 180°). Antes de comenzar el volcado de datos por parte del sensor, es necesario enviar una trama al láser con una serie de bytes que determinarán el comportamiento del mismo.

Para obtener los datos, haremos una llamada a una subrutina para evitar perder el foco en la ventana principal. Esta subrutina es un hilo que se ejecutará paralelamente al programa principal con el único fin de comunicarse con el sensor. El hilo se ejecutará de forma continua, favoreciendo el reconocimiento del entorno a medida que el autómata va avanzando. La

Procesamiento de la información recibida de un sensor láser ubicado en un vehículo tipo Quad

subrutina abrirá el puerto serie la primera vez que se ejecute el bucle principal y enviará los parámetros adecuados para las opciones seleccionadas mediante la interfaz. Una vez recuperados los datos, los almacenaremos en dos vectores. Uno contendrá tanto la distancia en centímetros desde el dispositivo hasta el primer objeto encontrado por el rayo como el ángulo que forma dicho rayo y el otro contendrá los puntos en el espacio en coordenadas cartesianas correspondientes a la localización de los distintos objetos, tomando como origen de coordenadas el propio sensor. Una vez los tengamos, los pasaremos al programa principal, donde serán tratados.

Mediante el programa principal se puede definir una profundidad mínima de hueco. Esto quiere decir que para distancias menores que esta profundidad, asumiremos que no es una solución posible. Puede variar entre 1 y 81 metros, aunque no tiene sentido tomar un valor inferior a 2-3 metros, teniendo en cuenta las limitaciones del vehículo para efectuar un giro. De este modo, si el usuario propone una distancia inferior a los 3 metros, el programa asumirá que esta distancia no es válida y trabajará con una distancia mínima de 3 metros. Los vectores que contienen las lecturas del láser se pasarán a un algoritmo, que se encarga en primer lugar de calcular los huecos por los que puede pasar nuestro robot y posteriormente elegirá el más conveniente. El criterio de selección de hueco hacia el que acudir, o *salida* depende nuevamente de dos factores. Se puede elegir el hueco más centrado (el que requiera menor giro del autómatas) o el hueco de mayor tamaño. Una vez elegido, la aplicación dibujará una gráfica en la parte destinada a tal efecto.

La gráfica mostrada en el apartado *Gráfica de visión* incluye dos semicircunferencias en línea negra discontinua, que representan la distancia máxima y la mitad de la distancia máxima (81 y 40,5 metros respectivamente), una línea azul continua que dibuja el contorno de los objetos detectados por el sensor, una o varias líneas rectas rojas punteadas que supone las posibles soluciones o huecos y una línea continua más gruesa que representa la solución elegida como más adecuada en dirección a la salida. El programa refrescará esta información, y actualizará adecuadamente la gráfica de visión del robot y la salida elegida por el mismo.

CAPITULO 5. PRUEBAS Y RESULTADOS

5.1.- Descripción de experimentos

Dado que no se puede integrar el software en el propio vehículo, las pruebas a ejecutar se limitarán a pruebas de comportamiento simulado y pruebas de rendimiento.

Experimento #1.

La primera prueba realizada fue una simulación de movimiento del vehículo con el láser incorporado. Subido el láser a una silla con ruedas de las que hay en el laboratorio, el experimento consistía en empujar la silla por el laboratorio para corroborar que la gráfica se actualizaba correctamente, con una frecuencia aceptable y adecuadamente a lo que se esperaba

Experimento #2.

La segunda prueba fue una prueba de rendimiento. Se ejecutó el programa en uno de los PCs de laboratorio y se dejó en funcionamiento, en modo de lectura continua durante 30-40 minutos. El objetivo del experimento era comprobar cómo afectaba el funcionamiento continuado al rendimiento del PC, si consumía muchos recursos, etc.

Experimento #3.

Este experimento fue llevado a cabo al mismo tiempo que el #2. Se dejó el programa en funcionamiento durante mucho tiempo para comprobar que la frecuencia de las lecturas realizadas era adecuada con respecto a la frecuencia con que el láser proporciona datos. La idea inicial suponía que si después de algún tiempo con el programa en marcha no se producían lags entre la representación gráfica de la lectura y lo que se esperaba de la lectura actual, la frecuencia de lectura era correcta.

Procesamiento de la información recibida de un sensor láser ubicado en un vehículo tipo Quad

5.2.- Resultado de las pruebas

Experimento #1.

Tras varias rutas alrededor del laboratorio, la respuesta de la gráfica era la esperada, con una tasa de refresco adecuada y una representación bastante fiable.

Experimento #2.

Se comprobó de acuerdo al administrador de tareas de Windows que el programa utiliza una cantidad de memoria fija que no variaba por mucho tiempo que estuviera el programa en funcionamiento. En cuanto al uso de la CPU, oscilaba regularmente entre unos valores, sin crecer demasiado. Se puede concluir, que el programa utiliza unos recursos fijos que no se incrementan con el tiempo de ejecución del programa.

Experimento #3.

Por más que pasaba el tiempo, la gráfica seguía refrescándose de forma periódica, aparentemente con la misma frecuencia que el láser realizaba lecturas. Tras 30 minutos de ejecución, la gráfica mostraba en tiempo real los resultados de las lecturas. Se puede concluir que la frecuencia de refresco de la gráfica es igual o mayor a la de lectura del láser.

CAPÍTULO 6. CONCLUSIONES Y TRABAJO FUTURO

6.1.- Conclusiones

En el presente proyecto se perseguía establecer una vía de comunicación entre un PC y un dispositivo de medición láser y una vez establecida la comunicación, el intercambio de datos y la interpretación de los mismos.

Si bien el proyecto está orientado a formar parte de otro mayor en el futuro, la meta intermedia que se presentaba en este desarrollo era el análisis e interpretación de los datos, preparando el camino para desarrollos venideros. Para llevarlo a cabo se ha optado por encaminar el trabajo a la comunicación con el dispositivo, enviándole órdenes y recibiendo respuestas. Más en detalle, los desarrollos llevados a cabo son:

- Establecimiento de la conexión mediante un puerto serie RS232.
- Construcción y envío de tramas al dispositivo
- Recepción y tratamiento de los datos
- Construcción de una interfaz gráfica donde representar los datos obtenidos

Así mismo, la interfaz ha servido de herramienta para que el usuario pueda comunicarse con el láser de una forma más dinámica y eficiente, enviando los parámetros adecuados a las necesidades del usuario.

El dispositivo láser ha resultado ser una herramienta bastante fiable, con una precisión más que aceptable y una estabilidad a la hora de percibir el entorno satisfactoria. De este modo, el telémetro se perfila como una herramienta muy útil de cara al proyecto futuro de dotar de autonomía a un vehículo móvil.

En cuanto al algoritmo de tratamiento de datos, obtención de soluciones y elección de la más adecuada satisface la meta original aportando soluciones válidas. Aunque en el futuro, la parte de selección de la solución óptima se verá modificada por algoritmos de búsqueda global, ya que a este nivel todas las soluciones encontradas responden a un criterio local, al no disponer de una meta final. No obstante, la aportación de este algoritmo de soluciones “a corto plazo” cubre las necesidades que presentaba el proyecto.

Procesamiento de la información recibida de un sensor láser ubicado en un vehículo tipo Quad

6.2.- Trabajo Futuro

En el futuro, se espera poder usar el resultado del proyecto como parte de la automatización de un vehículo tipo Quad. En concreto, este programa se encargará de comunicarse con el sensor láser, tratar la información y determinar qué zonas del entorno son susceptibles de formar parte de la solución. Una vez obtenidas las soluciones, se enviarán a un algoritmo de decisión global que tendrá en cuenta la posición actual del robot respecto de la meta y efectuará los mecanismos de control adecuados para poner el vehículo rumbo a la ruta seleccionada.

REFERENCIAS

- **Telémetro SICK LMS291-S05**
 - online_data_sheet_LMS291-S05_en_20110204_1459.pdf
 - LMS_Quick_Manual_V1_1.pdf
 - www.sick.com
 - www.mysick.com

- **Programación de la interfaz gráfica**
 - Apuntes de las asignaturas de la Titulación: IDP/EDA/MTP
 - <http://msdn.microsoft.com/es-es/library/ms123401.aspx>
 - www.codeguru.com
 - www.ucancode.net

- **Información sobre el estado del arte**
 - Giamarchi, Frederic – ROBOTS MÓVILES (ISBN: 8428327769)
 - José Andrés Somolinos Sánchez - AVANCES EN ROBÓTICA Y VISIÓN POR COMPUTADOR (ISBN: 8484271994)

INDICE DE TABLAS Y FIGURAS

Figuras

2.1.- Sistema de bucle abierto.....	9
2.2.- Sistema de bucle cerrado.....	9
2.3.- Mapa métrico	10
2.4.- Mapa de rejilla.....	11
3.1.- Interfaz de la aplicación.....	14
4.1.- Interfaz detallada.....	18

Tablas

3.1.- Características técnicas del telémetro.....	13
3.2.- Rendimiento del telémetro	13
3.3.- Mecánica/electrónica del telémetro	13
4.1.- Caso de uso #1	16
4.2.- Caso de uso #2	17
4.3.- Caso de uso #3	17

ANEXO I - MANUAL DE USUARIO

1.- INTRODUCCION

El presente manual pretende guiar al usuario a la hora de usar la aplicación que controla el funcionamiento del telémetro láser SICK LMS 291-S05. Dicho telémetro es un aparato de medida para uso en exteriores, con un rango de hasta 80 metros.

El programa sirve para poner en práctica las diversas configuraciones que permite el láser, mostrando los resultados en un área diseñada a tal efecto. Dicho programa, consta de varios botones de selección donde se ajusta la configuración deseada para que el resultado sea lo más parecido a lo que el usuario necesita.

2.- INTERFAZ

La pantalla principal se divide en tres partes fundamentales: de izquierda a derecha, la primera parte consta de tres botones para iniciar o detener la lectura, después encontramos una serie de paneles que determinarán el modo de lectura y por último un área donde se representarán los resultados.

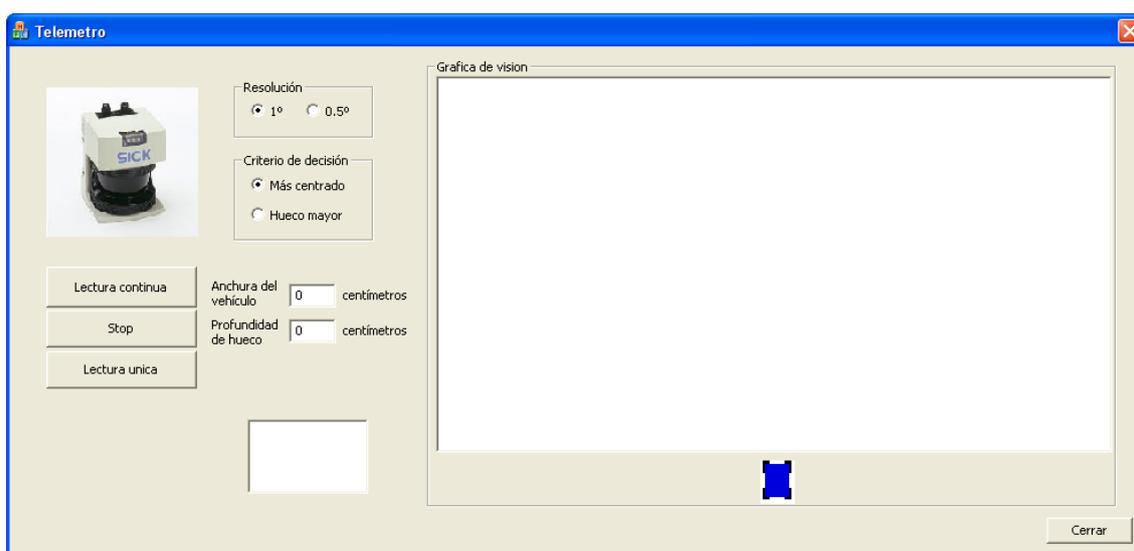


Figura I.1 - Interfaz gráfica

Procesamiento de la información recibida de un sensor láser ubicado en un vehículo tipo Quad

3.- USO

3.1.- CONFIGURACIÓN DE LA LECTURA.

Antes de iniciar una lectura, conviene tener en consideración las alternativas de lectura que nos ofrece el telémetro. Si observamos el panel de configuración, veremos unos botones circulares en la parte superior y unos campos de texto en el centro.

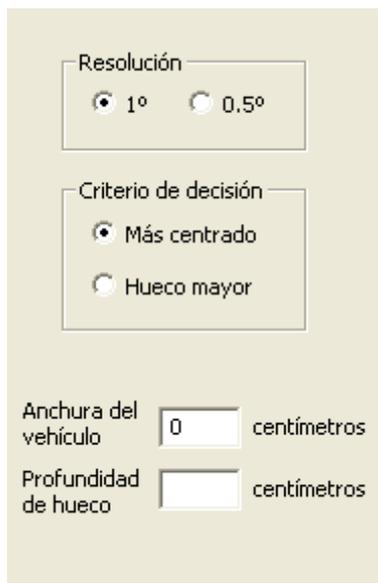


Figura I.2 - Panel de configuración

El primer grupo de botones circulares nos permiten elegir la precisión con que se emitirán los rayos. Hay que tener en cuenta que el rango de visión del aparato tiene una amplitud de 180° . Por defecto, estará seleccionada la opción 1° , que provocará que el láser emita 181 rayos, uno cada grado. Si hacemos clic sobre el botón de $0,5^\circ$, el láser emitirá un total de 361 rayos, cada $0,5^\circ$.

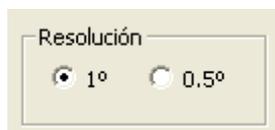


Figura I.3 - Botones de resolución

El segundo grupo de botones tiene que ver con las soluciones que mostrará el visor. Cada vez que se efectúe una lectura por parte del telémetro, los datos de la misma se mostrarán en el área del visor, además de las posibles salidas que haya determinado el programa como válidas. No obstante, además de mostrar todas las posibilidades, se decidirá por una dependiendo del criterio que se marque con estos botones. Por defecto está marcado el "Más centrado", haciendo que el programa elija la salida que esté más en el centro y que

Procesamiento de la información recibida de un sensor láser ubicado en un vehículo tipo Quad

requiera un giro menor para el vehículo donde esté instalado el equipo. Si el usuario selecciona la opción “Hueco mayor”, el programa se decantará por elegir aquella salida cuya anchura sea mayor.

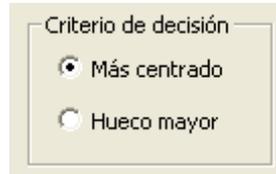


Figura I.4 - Botones de decisión

Por último, los campos de texto permiten únicamente valores numéricos. Por defecto no tendrán valor, aunque el programa internamente adecuará estos valores para que las lecturas obtenidas y representadas tengan un valor más o menos realista. El campo de Anchura del vehículo espera obtener la anchura en centímetros del vehículo donde va instalado el láser, para así poder descartar todos aquellos huecos cuya amplitud sea menor a la anchura del vehículo (si el vehículo no cabe, no lo tomaremos como solución). Por otro lado, el campo *Profundidad del hueco* determina la distancia mínima a la que debe estar un obstáculo para que consideremos la opción de dirigirnos hacia él. Esto quiere decir que todo lo que esté a una distancia menor a la introducida en este campo, será considerado un obstáculo, mientras que lo que se encuentre más lejos será susceptible de formar parte de una posible solución. Este campo se usa para evitar considerar elementos demasiado cercanos como salida. Este campo como mínimo será 200 centímetros (2 metros) para dar margen al vehículo para efectuar giros. Por defecto, contaremos con una anchura del vehículo de 150 centímetros y una profundidad mínima de 200 centímetros.

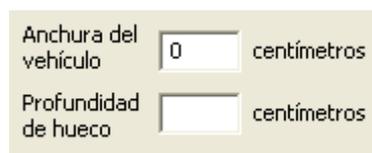
Una interfaz de usuario con dos campos de texto. El primer campo está etiquetado "Anchura del vehículo" y contiene el número "0", con "centímetros" a la derecha. El segundo campo está etiquetado "Profundidad de hueco" y está vacío, con "centímetros" a la derecha.

Figura I.5 - Campos de texto

Una vez introducidas las opciones deseadas, procedemos a iniciar la lectura.

3.2.- LECTURA

Para iniciar la lectura, existen unos botones en la parte izquierda de la interfaz.

El botón *Lectura continua* pone el láser a realizar lecturas constantemente, a razón de una cada 300 milisegundos, aproximadamente. Toma los parámetros

Procesamiento de la información recibida de un sensor láser ubicado en un vehículo tipo Quad

establecidos por el usuario, inhabilita los campos para evitar que se modifiquen mientras se está leyendo y representa la información periódicamente en la parte del visor. La rutina asociada a este botón se ejecutará ininterrumpidamente hasta que el usuario presione el botón *Stop*.

El botón *Stop* detiene la lectura continuada, vuelve a habilitar las opciones de configuración y deja el programa preparado para la próxima lectura. En la gráfica se mantendrá la representación de la última lectura del láser.

Por último el botón *Lectura única* toma los parámetros elegidos y realiza una única lectura. Los valores obtenidos se representarán en el área del visor.

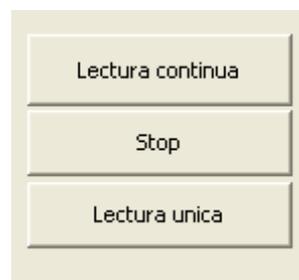


Figura I.6 - Botones de lectura y paro

3.3.- EL VISOR

Para cada lectura realizada por el dispositivo, el programa mostrará en la parte derecha una gráfica con dos semicircunferencias en línea negra discontinua que marcan la distancia máxima de lectura (80 m) y la distancia media (40 m), a modo de referencia. También dibujará una línea continua azul que representa el contorno de los elementos y objetos del entorno. Por último, mostrará las soluciones posibles con una serie de líneas punteadas en rojo, y la solución elegida como la óptima en una línea más gruesa.

Para una mejor comprensión, la representación se realiza considerando que el lector se encuentra en el centro de la parte inferior de la gráfica. Se incluye un icono que representa el vehículo

Para la opción de lectura continua, esta gráfica se refrescará automáticamente cada vez que el dispositivo realice una lectura, obteniendo una representación actualizada del entorno.