

TRABAJO FIN DE MÁSTER

**MÉTODO AVANZADO DE *SCAN-MATCHING* POLAR USANDO UN
ALGORITMO DE SEGMENTACIÓN DE LÍNEAS**

Autor:

Israel Navarro Santosjuanes

Ingeniero Aeronáutico

Dirigido por:

Dr. José Manuel Cuadra Troncoso

Departamento de Inteligencia Artificial

Escuela Técnica Superior de Ingeniería Informática

Universidad Nacional de Educación a Distancia



Madrid, junio 2013

Agradecimientos

Querría agradecer a mi esposa Ana el apoyo que me ha prestado durante todo el tiempo que he cursado los estudios de post-grado en la UNED. Nada hubiera sido posible sin que ella asumiera una parte aún mayor de las tareas comunes. Gracias. Y también, cómo no, agradecer al pequeño Daniel su paciencia e infinito amor por el raro de su padre.

Al Departamento de Inteligencia Artificial de la UNED me gustaría agradecerle la oportunidad de cursar estos apasionantes estudios. En especial es más que necesario recordar a mi director José Manuel Cuadra por su sabiduría y pasión por el trabajo. Además, José Manuel es el autor de la herramienta de simulación *cybersim* que sin duda me ha permitido llegar mucho más allá de lo que yo hubiera podido alcanzar por mí mismo. Sin duda, ha sido un gran placer trabajar con un maestro de las matemáticas.

Resumen

El término *scan-matching* se refiere en general a la familia de algoritmos que resuelven el problema de la falta de alineamiento entre dos conjuntos de medidas de alcance. Concretamente, en el contexto del problema de la localización en el campo de la robótica autónoma, este tipo de algoritmos se emplea para determinar el cambio de pose de un robot. En esta aproximación, el cambio de pose se estima de manera que se optimice el alineamiento entre dos escáneres láser tomados en dos puntos diferentes de la trayectoria del robot teniendo además en cuenta las relaciones geométricas conocidas entre dichos dos puntos. Este trabajo presenta un procedimiento mejorado de *scan-matching* basado en el trabajo original de Diosi. En este documento se referirá a este nuevo algoritmo bajo el nombre de E-PSM (del inglés *Enhanced Polar Scan Matching*).

Las novedades introducidas por el algoritmo E-PSM se pueden resumir en tres puntos: (1) Las entradas al E-PSM consisten del resultado de aplicar un avanzado algoritmo de filtrado y segmentación de líneas a las lecturas del escáner láser de un robot móvil, (2) Se incluyen mejoras en el método de interpolación y de detección de puntos ocluidos; además, se presenta un método refinado de búsqueda de la orientación. (3) Un conjunto de métodos basados en optimización local o global se introducen para resolver la búsqueda de la pose óptima ya sea en su versión de búsqueda alterna o simultánea.

Finalmente se presentan los resultados obtenidos en simulaciones para diferentes niveles de ruido en las lecturas láser y/o en las medidas de odométricas del robot en tres mapas diferentes. Los resultados muestran que: (1) El algoritmo de filtrado y segmentación usado en este trabajo reduce notablemente el error en la estimación de la pose del robot generado por el algoritmo de *scan-matching*. (2) El algoritmo E-PSM y una de sus variantes basada en el algoritmo de optimización global CRS2 mejoran notablemente los resultados obtenidos con la metodología original de Diosi a la vez que mantienen los requerimientos computacionales asociados en niveles razonables.

Índice General

1	Introducción	1
2	Estado de la cuestión	3
2.1	Consideraciones generales	3
2.2	Localización del robot	3
2.2.1	Tipos de problemas	3
2.2.2	Localización global o local	4
2.2.3	Localización en entornos estáticos o dinámicos	4
2.2.4	Localización pasiva o activa	5
2.2.5	Localización por un robot o por un grupo de robots	5
2.3	Algoritmos para la localización	6
2.3.1	Método de Markov	6
2.3.2	Filtros de Kalman	6
2.3.3	Métodos de Monte Carlo	7
2.4	Localización y generación de mapas simultanea (SLAM)	8
2.4.1	Generación de mapas	8
2.4.2	Percepción robótica para el SLAM	9
2.4.3	Gestión de la incertidumbre	11
2.4.4	Otra retos en el problema del SLAM	13
2.5	Scan-matching	15
3	Procedimiento general	17
4	Algoritmo de segmentación de línea	19
4.1	Clustering mediante scale-space	19
4.2	Estimación de líneas mediante EKF	21
4.3	Extracción de líneas y merging de segmentos similares adyacentes	22
5	Procedimiento mejorado de <i>scan-matching</i>	24
5.1	Proyección de las medidas	24

Índice general	[v]
5.2 Interpolación y descartes	25
5.3 Actualización de la pose del robot	27
5.3.1 Búsqueda alterna de orientación y posición	29
5.3.2 Búsqueda simultánea de posición y orientación	33
6 Resultados	34
6.1 Búsqueda alterna de posición y orientación	39
6.2 Búsqueda simultánea de posición y orientación	59
6.3 Efecto del uso del algoritmo de segmentación	62
6.4 Ensayos en otros dos mapas adicionales con el algoritmo E-PSM	63
7 Conclusiones y trabajos futuros	73
7.1 Conclusiones	73
7.2 Trabajos futuros	74
8 Referencias	76
9 Apéndice: Ejemplos de mapas generados	83

Lista de Abreviaciones / Acrónimos

EIF	Extended Information Form
EKF	Extended Kalman Filter
E-PSM	Enhanced Polar Scan Matching
GSF	Gaussian Filter Sum
ICP	Iterative Closest Point
IDC	Iterative Dual Correspondence
IMRP	Iterative Matching Range Point
MCG	Mínimos Cuadrados Generalizados
MCO	Mínimos Cuadrados Ordinarios
PSM	Polar Scan Matching
SLAM	Simultaneous Localization and Mapping
SM	Set Membership
SRT	Sequential Ratio Test
UBB	Unknown But Bounded

Índice de figures

Figura 1: Procedimiento general	18
Figura 2: Ejemplo de aplicar el proceso de filtrado	20
Figura 3: Ejes de coordenadas de referencia (ref) y actual (cur)	24
Figura 4: Interpolación de medidas láser en el marco de referencia.....	26
Figura 5: Ejemplo de uso del criterio de descarte por oclusión	28
Figura 6: Búsqueda alterna de orientación y posición	30
Figura 7: Búsqueda alterna de orientación y posición usando algoritmo de optimización local	32
Figura 8: Mapa rectangular con una columna.....	36
Figura 9: Mapa poligonal.....	37
Figura 10: Mapa de gran tamaño	38
Figura 11: Posición absoluta del robot (x-coord) en función del tiempo.	40
Figura 12: Posición absoluta del robot (y-coord) en función del tiempo	40
Figura 13: Orientación del robot (en rad) en función del tiempo	41
Figura 14: Error acumulado en la estimación de la orientación del robot (grados).....	41
Figura 15: Trayectoria del robot en el plano XY	42
Figura 16: Error en a_x en cada <i>scan-matching</i> realizado	43
Figura 17: Error en a_y en cada <i>scan-matching</i> realizado	43
Figura 18: Error en a_{θ} en cada <i>scan-matching</i> realizado (grados).....	44
Figura 19: Valor de la función objetivo f (error en comparación) en cada <i>scan-matching</i> realizado	44
Figura 20: Escáneres de referencia y actual en el plano XY (No. Paso = 2).....	46
Figura 21: Escáneres de referencia y actual en coordenadas polares (No. Paso = 2).....	46
Figura 22: Segmentación del escáner actual (No. Paso = 2)	47
Figura 23: Segmentación del escáner actual en coordenadas polares (No. Paso = 2).....	47
Figura 24: Alineamiento de los escáneres de referencia y actual (No. Paso = 2)	48
Figura 25: Búsqueda de la orientación (No. Paso = 2)	49
Figura 26: Contorno de la función objetivo en el plano XY (No. Paso = 2)	49
Figura 27: Evolución del valor de la función objetivo en función del número -de iteraciones (No. Paso = 2)	50
Figura 28: Escáneres de referencia y actual en el plano XY (No. Paso = 78)	51
Figura 29: Escáneres de referencia y actual en coordenadas polares (No. Paso = 78).....	51
Figura 30: Segmentación del escáner actual (No. Paso = 78).....	52
Figura 31: Segmentación del escáner actual en coordenadas polares (No. Paso = 78)	52
Figura 32: Alineamiento de los escáneres de referencia y actual (No. Paso = 78).....	53
Figura 33: Búsqueda de la orientación (No. Paso = 78).....	54
Figura 34: Contorno de la función objetivo en el plano XY (No. Paso = 78)	54

Figura 35: Escáneres de referencia y actual en el plano XY (No. Paso = 30)	57
Figura 36; Escáneres de referencia y actual en coordenadas polares (No. Paso = 30)	57
Figura 37: Segmentación del escáner actual (No. Paso = 30)	58
Figura 38: Segmentación del escáner actual en coordenadas polares (No. Paso = 30)	58
Figura 39: Alineamiento de los escáneres de referencia y actual (No. Paso = 30)	59
Figura 40: Posición absoluta del robot (x-coord) en función del tiempo	65
Figura 41: Posición absoluta del robot (y-coord) en función del tiempo	65
Figura 42: Orientación del robot (en rad) en función del tiempo	66
Figura 43: Error acumulado en la estimación de la orientación del robot (deg)	66
Figura 44: Trayectoria del robot en el plano XY	67
Figura 45: Escáneres de referencia y actual en el plano XY (No. Paso = 55)	68
Figura 46: Escáneres de referencia y actual en coordenadas polares (No. Paso = 55)	68
Figura 47: Segmentación del escáner actual (No. Paso = 55)	69
Figura 48: Segmentación del escáner actual en coordenadas polares (No. Paso = 55)	69
Figura 49: Alineamiento de los escáneres de referencia y actual (No. Paso = 55)	70
Figura 50: Búsqueda de la orientación (No. Paso = 55)	71
Figura 51: Contorno de la función objetivo en el plano XY (No. Paso = 55)	71
Figura 52: Mapa poligonal generado (sin ruido)	83
Figura 53: Mapa poligonal generado (ruido odométrico de 1%)	83
Figura 54: Mapa generado (ruido odométrico de 5%)	84
Figura 55: Mapa generado (ruido odométrico de 10%)	84
Figura 56: Mapa generado (ruido en lecturas láser con desviación estándar de 5mm)	85
Figura 57: Mapa generado (ruido en lecturas láser con desviación estándar de 20mm)	85
Figura 58: Mapa generado (ruido 5% + 5mm)	86
Figura 59: Mapa generado (ruido 10% + 20mm)	86

Índice de tablas

Tabla 1: Tipos de pruebas realizadas en el simulador	35
Tabla 2: Resultados de búsqueda alterna de orientación y posición (valores medio y máximo) ...	56
Tabla 3: Resultados de búsqueda simultanea de orientación y posición.....	61
Tabla 4: Evaluación del uso del algoritmo de segmentación.....	63
Tabla 5: Resultados para varios mapas con el algoritmo E-PSM y CRS2.....	72

1 Introducción

El método del *scan-matching* (que se podría traducir como “comparación de escáneres” aunque se haga en este trabajo uso del término original inglés por la amplia difusión del mismo) es una aproximación para resolver parte del problema de la localización y generación simultánea de mapas (SLAM). El SLAM proporciona al robot la capacidad de realizar sus tareas de forma autónoma en un ambiente desconocido al permitir al robot situarse y generar mapas del entorno a partir únicamente de sus propios sensores [Leonard91, Smith90].

El *scan-matching* se puede realizar en dos o tres dimensiones y se puede aplicar a entornos estáticos o dinámicos. Se cuentan también tanto con aproximaciones locales [Lu94] como globales [Tomono04]. En una solución local, la pose del robot es estimada comparando un par de escáneres láser del entorno [Weiss95]. Empezando desde una posición conocida, el proceso de *scan-matching* es repetido en intervalos regulares. Para cada segmento avanzado, se almacena el escáner obtenido en la posición inicial (llamado escáner de referencia o *reference scan*). El escáner obtenido al completar el segmento (llamado escáner actual o *current scan*) se compara con el escáner de referencia. De esta comparación. Se calcula la pose del robot relativa a su pose inicial. Este proceso proporciona correcciones a los errores en la posición y orientación calculados usando únicamente las medidas odométricas disponibles al robot. En el *scan-matching* global [Borthwick94, Cox01], el escáner actual se compara contra un mapa global de características (construido a priori o durante la exploración) o una base de datos de escáneres.

La clave del proceso de *scan-matching* es la metodología usada para asociar y comparar los datos del escáner actual con los datos del escáner de referencia. Algunos de estos métodos son de característica a característica, de puntos a línea, y de puntos a puntos. En las aproximaciones de característica a característica, los elementos como los segmentos [Gutmann02], esquinas, o extremos [Lingemann04] son extraídos de los escáneres y comparados. Estas características no tienen por qué ser elementos físicos concretos; por ejemplo, una aproximación basada en el uso de la función de correlación cruzada se presenta en [Weiss95]. En este caso, los dos escáneres son reemplazados por sus representaciones estocásticas (histogramas) y la comparación se resuelve encontrando el máximo de función de correlación cruzada. En las aproximaciones de puntos a puntos [Cox01], los puntos del escáner actual son comparados con un modelo poligonal del mundo consistente únicamente de segmentos lineales. Algunos de las aproximaciones más prominentes de este tipo son: *Iterative Closest Point* (ICP) [Besl92], *Iterative Matching Range Point* (IMRP) y el *Iterative Dual Correspondence* (IDC) [Lu97]. Estas tres aproximaciones están definidas en coordenadas cartesianas y no se aprovechan de la ventaja de usar coordenadas polares que es el sistema nativo a la generación de escáneres láser. Para remediar esto, Diosi propone en [Diosi07a] un método novel denominado *Polar Scan Matching* (PSM). La principal ventaja del PSM es su habilidad de buscar asociaciones por la simple comparación de los puntos con el mismo ángulo de orientación. Este sencillo procedimiento elimina el costoso proceso de la búsqueda de los puntos correspondientes.

De esta exposición, debería ser evidente que cualquier factor que degrade la precisión de las lecturas laser tiene un efecto negativo pronunciado en las prestaciones ofrecidas por cualquier algoritmo de *scan-matching*. Resulta por tanto de vital importancia poseer la capacidad de comprender y gestionar todas las fuentes de ruido en ellas. Se pueden distinguir varias fuentes de ruido: (1) Errores intrínsecos en las lecturas debido a errores en los aparatos de medición, (2) Errores de calibración y, (3) Irregularidades del entorno (dependencias en las propiedades refractivas de las superficies, presencia de superficies no lisas, etc.). El efecto de las dos primeras se puede minimizar (o por lo menos ser acotado) usando técnicas de calibración o especificaciones técnicas. Sin embargo, esto no es así con el tercer caso siendo ésta posiblemente la mayor fuente de error con la que además contamos una limitada capacidad de control. Esto no es de extrañar ya que resulta de la naturaleza física del entorno. En el marco de esta necesidad de descartar el ruido presenta en las lecturas láser, Cuadra [Cuadra11a] desarrolló un algoritmo para la extracción de líneas que hace uso de un modelo de ruido genérico lo que le permite ser aplicado a entornos de superficies con muy diferentes características.

El resto de este trabajo está organizado como se indica a continuación:

- El capítulo 2 proporciona una revisión del estado actual de la investigación robótica en los campos de la localización, el SLAM y el *scan-matching*
- El capítulo 3 introduce el procedimiento general de SLAM basado en *scan-matching* propuesto en este trabajo incluyendo su relación con la aproximación de Diosi en el que está basado
- El capítulo 4 proporciona un resumen del algoritmo de filtrado y segmentación de línea usado en este trabajo para proporcionar los datos de entrada al proceso de *scan-matching*
- El capítulo 5 da los detalles del algoritmo de *scan-matching* desarrollado aquí incluyendo algunas variantes que hacen uso de técnicas de optimización local y/o global
- El capítulo 6 presenta los resultados obtenidos al probar la propuesta presente en un simulador de robot en diferentes mapas y con niveles diferentes de ruido en los sensores odométricos y los sensores láser. Estos resultados se comparan con los obtenidos por el algoritmo original de Diosi y las variantes del E-PSM basadas en la optimización local y/o global.
- El capítulo 7 expone las conclusiones e identifica posibles líneas futuras de trabajo
- El capítulo 8 lista los detalles de las referencias usadas en este trabajo
- El apéndice muestra varios ejemplos de mapas generados de un entorno poligonal con varios niveles de ruido en los sensores odométricos y láser.

2 Estado de la cuestión

2.1 Consideraciones generales

Uno de los principales retos a la hora de crear robots que puedan ser autónomos de forma efectiva es el problema de la auto-localización. Para que un robot pueda planear su ruta debe ser capaz de reconocer en que parte de su entorno se encuentra. De otro modo, el robot probablemente se va a limitar a desplazarse de forma aleatoria lo que no supone un comportamiento útil para la mayoría de aplicaciones. Además, si el mapa del entorno del robot no se encuentra disponible, deberá ser capaz de trazarlo a la vez se ubique él mismo en el mapa en tiempo real. Estos problemas forman parte del objetivo de más alto nivel referente a la definición, gestión y actualización del modelo interno del robot.

En general, los robots tienen acceso a información parcial y ruidosa del entorno a través de sensores de diferentes tipos. En entornos dinámicos, la fusión de información es vital para que el modelo del mundo creado sea tan preciso como sea posible. La fusión de información para la localización se consigue típicamente a través del uso de técnicas probabilísticas como los filtros de Kalman o los de partícula. También, la integración de la información sensorial a lo largo del tiempo para conseguir filtrar el ruido es esencial para llegar a buenas estimaciones. Este tipo de integración se puede realizar con filtros de Kalman, con el método de Monte Carlo o con aproximaciones de Markov. En general, los métodos de Monte Carlo [Metropolis49] proporcionan las mejores prestaciones cuando se encuentran discontinuidades significativas en los datos ya que en estos casos la suposición de una densidad de probabilidad Gaussiana usada por Kalman [Kal01] es poco precisa. Otra aproximación común es la del EKF (del inglés, *Extended Kalman Filter*) que proporciona la capacidad de analizar sistemas no-lineales aunque sea a costa de un mayor costo computacional [Wan00].

En el campo de los robots móviles autónomos, se encuentran en general dos tipos de escenarios: uno en el que el entorno es conocido por lo menos en parte y en el que el robot se debe ubicar y otro en el que el entorno es desconocido y es el robot quien debe generar un mapa. Este último caso se conoce como el problema del SLAM (del inglés, *Simultaneous Localization and Mapping*) y es frecuentemente objeto del uso de técnicas de fusión sensorial [Thrun05, Siciliano08].

2.2 Localización del robot

2.2.1 Tipos de problemas

El problema de la localización de un robot consiste en identificar donde está dado un mapa de su entorno. Nótese que el problema de la localización se suele definir más precisamente como la búsqueda de la pose del robot (que consiste de una posición y una orientación) con respecto a un eje de coordenadas dado. Cuando un robot se mueve y estima su pose por sus propios medios realmente no puede estar seguro de cuales son su posición y orientación. El robot sólo puede

aportar una estimación de su pose pero sin una garantía perfecta. Esta estimación o creencia se suele presentar como una función de distribución de probabilidad, $bel(x)$, donde x representa la pose del robot.

A lo largo de su recorrido, el robot recoge información de su entorno a través de los sensores de los que dispone. Estos sensores pueden proporcionar datos con cierta precisión acerca de la pose del robot en el instante en que son usados. Sin embargo, nunca van a ser totalmente precisos y van por tanto a introducir una cierta cantidad de ruido en la estimación de la pose del robot. En el marco probabilístico, las observaciones z llevan a un modelo de observación dado por $p(z/x)$ que es una función de probabilidad.

Los problemas de localización son normalmente clasificados según diferentes aspectos de su funcionamiento, propósito y entorno.

2.2.2 Localización global o local

En este caso se clasifica el problema por el tipo de conocimiento inicialmente disponible y durante el tiempo de ejecución. Tres tipos de problemas se pueden diferenciar:

- Seguimiento de la posición. Se asume que la pose inicial del robot es perfectamente conocida. Esto quiere decir que el algoritmo de localización debe estimar la pose basándose en el valor previo. Este último suele tener errores pequeños que se pueden corregir en el modelo para obtener buenos resultados.
- Localización global. En este caso, la pose inicial es desconocida. El robot inicia su movimiento en alguna parte de su entorno; esto implica que no se puede hacer suposición alguna sobre el límite del error presente en la estimación de la pose.
- Robot secuestrado. Ésta es una variante del problema anterior con la dificultad añadida que durante su operación el robot puede ser inadvertidamente transportado de forma arbitraria a otra parte de su entorno. Aquí se revela crucial la capacidad de cómo detectar si cierta estimación de una pose es incorrecta.

2.2.3 Localización en entornos estáticos o dinámicos

La variabilidad dinámica del entorno puede tener un impacto significativo en la dificultad del problema de localización. Por lo común, los entornos se distinguen en dos tipos:

- Entornos estáticos. En este tipo de entornos, el robot es el único elemento dinámico; todos los elementos del entorno ajenos al robot permanecen indefinidamente en su misma pose.
- Entornos dinámicos. En éstos, la posición y configuración de algunos objetos cambia con el tiempo. En el entorno de una oficina, algunos ejemplos representativos serían las personas, los muebles que se pueden mover (como las sillas), las puertas, ventanas, luminosidad, etc.

Como sería lógico asumir, el problema de la localización es más complejo en entornos dinámicos. Se puede identificar dos aproximaciones principales para adaptar los algoritmos de localización en este tipo de casos:

- Incluir las entidades dinámicas en el vector de estado del robot. Esta metodología permite a la vez mapear el entorno pero requiere del uso de modelos complejos y de grandes recursos computacionales.
- Filtrar las medidas de los sensores para eliminar efectos causados por efectos dinámicos no modelados.

2.2.4 Localización pasiva o activa

Esta característica se refiere a la capacidad del algoritmo de localización de influenciar la trayectoria del robot.

- Localización pasiva. En este caso, el algoritmo de localización se limita a observar cómo opera el robot. Normalmente, esto quiere decir que la trayectoria del robot es determinada por otro algoritmo diseñado para llevar a cabo ciertas tareas asignadas al robot.
- Localización activa. Por contra, en este tipo de problemas el algoritmo de localización tiene un efecto directo sobre el control de movimiento del robot. Este hecho le concede la oportunidad de elegir la trayectoria del robot para obtener mejores medidas del entorno y obtener una mejor estimación de la pose.

Las aproximaciones activas tienden a obtener mejores resultados que las pasivas. Sin embargo, tienden a ser insuficientes para obtener un comportamiento útil para el robot si se aplican por sí solas. En la mayoría de casos, el robot debe ejecutar otras tareas con lo que es práctica habitual combinar la tarea de localización con la de propia del robot en los algoritmos de control del mismo.

2.2.5 Localización por un robot o por un grupo de robots

Este aspecto del problema trata sobre el número de robots que estén llevando a cabo la tarea de localización:

- Mono-robot. El caso más estudiado a fondo es aquel en el que el problema de localización es resuelto por un único robot. Su única ventaja es la sencillez ya que todos los datos sensoriales provienen de la misma plataforma y no hay problemas de comunicación o coordinación a tratar.
- Multi-robot. Este problema se puede tratar como si fueran varios problemas mono-robot y resolverlo de la misma manera. Pero si los robots tienen la habilidad de detectarse entre sí, las estimaciones pueden ser usadas para validar la de los otros siempre y cuando sus posiciones relativas estén disponibles.

2.3 Algoritmos para la localización

Se presentan a continuación algunos de los algoritmos más comúnmente usados para la resolución de problemas de localización

2.3.1 Método de Markov

La localización de Markov es una aplicación directa del filtro de Bayes al problema de la localización. El mapa se incluye como entrada al modelo de medida y a menudo se incorpora también al modelo de movimiento. El algoritmo transforma una estimación probabilística en un momento de tiempo $t-1$ a una estimación probabilística en un momento de tiempo t . Así se mantiene la probabilidad para cada pose posible del robot en el espacio de estados. Con el método de Markov también se puede afrontar el problema de la localización global y el problema del robot secuestrado.

La localización de Markov se ha probado en un prototipo de guía de museo robotizado [Fox99]. Este robot fue desarrollado específicamente para entornos dinámicos y para resolver tanto el problema de localización global como el problema del robot secuestrado (en la práctica, como resultado de fallos temporales en los algoritmos de localización).

2.3.2 Filtros de Kalman

La localización por filtros de Kalman típicamente requiere que el punto de arranque del robot sea conocido; así, esta metodología se centra en el problema del seguimiento de ruta. Al contrario que en el método de Markov, se mantiene una estimación $bel(x_t)$ que es una función gaussiana representable por su media y desviación estándar. El filtro de Kalman se basa en un modelo de sistema dinámico lineal discretizado en el tiempo. Se asume que tanto el sistema como las medidas son perturbados por ruido blanco gaussiano (es decir, sin correlación en el tiempo). Así, se asume que para cada paso en el tiempo el modo en el que el ruido perturba el sistema y las medidas no depende valores pasados o futuros.

El filtro de Kalman incluye dos pasos: (1) Se realiza una predicción de la salida en base a la evolución lineal del sistema dinámico con ruido; es decir se produce una estimación apriorística. (2) Las medidas basadas en los sensores son combinadas con la predicción para producir una estimación a posteriori de la pose del robot [Bishop01]. Si no hay ninguna lectura de los sensores disponible, la estimación generada por el modelo de movimiento tiende a degradarse porque se basa únicamente en las estimaciones a priori sin ninguna corrección posterior.

En general, se asume que un mapa dado está representado por un conjunto de características cuyos identificadores son conocidos sin incertidumbre. Este hecho permite establecer una

correspondencia única entre las medidas asociadas a las características en diferentes pasos del tiempo. Si la correspondencia entre las características observadas en dos instantes de tiempo es incierta, la identificación de las características se debe hacer durante el mismo proceso de localización. Una de las estrategias más simples y más usadas es la de correspondencia por máxima probabilidad [Howard02]. En ésta, se determina primero el valor más probable de la variable de correspondencia y entonces se aplica el filtro descrito tomando la correspondencia elegida por supuesta. Con este tipo de aproximaciones, la localización por filtro de Kalman se puede extender al problema de localización global.

Roumeliotis et al. [Roumeliotis00] presentan una combinación de estimación Bayesiana con filtrado de Kalman para localización que permite mejores resultados al relajar las suposiciones que hace el filtro de Kalman sobre el ruido en las medidas. En el contexto de la localización multi-robot, Roumeliotis et al. [Roumeliotis02] aplican el principio del filtro de Kalman dividido en varios filtros más pequeños de comunicación en el que cada robot tiene su filtro para procesar los datos de sus propios sensores. El intercambio de información entre dos filtros es necesario sólo cuando dos robots se detectan entre sí y miden su posición relativa. Cuando esto ocurre, entra en acción un filtro de Kalman más global capaz de tener en cuenta toda la información disponible.

2.3.3 Métodos de Monte Carlo

Esta otra importante familia de algoritmos consiste en la aplicación de los filtros de partículas al problema de la localización. La estimación $bel(x_t)$ se representa por un conjunto de partículas M , $X_t = x_t^{[1]}, x_t^{[2]}, \dots, x_t^{[M]}$. La estimación inicial $bel(x_0)$ se obtiene de la generación aleatoria de M partículas de pose desde la distribución a priori $p(x_0)$ asignando a cada una de ellas un factor de importancia de $1/M$. Esta aproximación se puede aplicar tanto al problema de seguimiento de una ruta como al de localización global.

En general, la estimación inicial $bel(x_0)$ es un conjunto de partículas aleatorias de pose uniformemente generadas sobre todo el espacio de poses posibles. Al obtener información de los sensores, el algoritmo de Monte Carlo analiza cada partícula y le asigna un factor de importancia teniendo en cuenta el modelo de medida el cual contabiliza el efecto de la partícula estimada, la observación actual, y el mapa. Los factores de importancia computados se usan a continuación para el proceso de re-muestreo que genera nuevas partículas que se encontrarán concentradas en torno a las posiciones más probables. De nuevo, las nuevas medidas asignan pesos de importancia no uniformes al conjunto de partículas con lo que el centro de la masa de probabilidad acumulada se tenderá a centrar en la localización más probable del robot.

El algoritmo de Monte Carlo tiene la ventaja de no estar limitado por las suposiciones sobre el sistema y la naturaleza del ruido que usan los filtros de Kalman. La precisión de la pose estimada se incrementa con el tamaño del conjunto de partículas aunque esto conlleva un mayor coste computacional.

Una extensión común al algoritmo de Monte Carlo es la inclusión de reglas heurísticas para añadir partículas aleatorias al conjunto. Con esto se pretende evitar que gradualmente desaparezcan las partículas de las localizaciones que no se encuentran entre las más probables. Esto permite que el algoritmo se recupere de errores en la localización y le permite afrontar retos como el del secuestro del robot.

El uso de los métodos de Monte Carlo con un número de partículas variable se presenta en [Heinemann06] con el objeto de obtener un buen balance entre prestaciones y demanda de recursos computacionales en una aplicación como la de fútbol robótico. En [Montesano05] se presenta una investigación sobre como un sistema basado en visión puede ser usado para que un par de robots se localicen usándose a sí mismos y a otras características de su entorno. En este caso se prueban una versión extendida del algoritmo de Kalman (EKF), el método de Monte Carlo y una combinación de los dos. En este caso, se observó que la aproximación de Monte Carlo es más robusta que la de Kalman en la estimación inicial y que todos los métodos proporcionan resultados similares al converger a la localización verdadera. El método combinado ofrece el mejor compromiso entre robustez y eficiencia. En [Gutmann02] se presenta también una comparación entre el uso de filtros de Kalman, método de Monte Carlo y sus combinaciones. En general, los métodos combinados proporcionan mejores resultados que los originales.

2.4 Localización y generación de mapas simultanea (SLAM)

En los problemas de tipo SLAM, no se conoce con certeza ni el mapa del entorno ni la pose del robot. Las entradas al algoritmo se limitan a las medidas de los sensores $z_{1:t}$ y los comandos de control $u_{1:t}$ del robot; las dos entradas pueden ser sujeto de ruido. Todo esto impone un problema significativamente más complejo que el de la localización ya que es el robot quien debe trazar un mapa del robot a la vez que recorre su entorno y se localiza en él. La habilidad de estimar tanto el mapa como la pose del robot se obtiene habitualmente formulando las correlaciones estadísticas que existen entre la pose del robot y los puntos notables del mapa y entre estos mismos puntos notables. Una estrategia típica para aumentar la precisión del algoritmo es re-observar los puntos notables del mapa ya identificados. Chen et al. [Chen07] proporcionan una completa revisión del progreso en este campo de investigación.

2.4.1 Generación de mapas

Desde el inicio de esta actividad investigadora hace más de 25 años, dos tipos de métodos han sido identificados para la creación de mapas [Bekey05]:

- *Mapas métricos basados en mallas.* Los mapas se crean a partir de representaciones basadas en las medidas del espacio que obtiene el robot [Moravec85]. En un mapa métrico de interiores, la información usada puede incluir la longitud de las paredes, la anchura de los pasillos, las distancias entre intersecciones, etc. Con esta aproximación,

el robot debe ser equipado con sensores capaces de estimar la distancia a cada objeto detectado. Es usual representar el espacio que está siendo mapeado como una malla espaciada de manera regular.

- *Mapas topológicos*. Al contrario que en el caso anterior, la representación del espacio no se basa en medidas precisas del entorno sino en la presencia de elementos notables [Kuipers91, Chown95]. En el contexto de un mapa de interiores, esto puede representar puertas, intersecciones entre pasillos, etc. Para poder construir un mapa basado en elementos notables, el robot debe estar dotado de sensores que le permitan identificarlos, típicamente a través de sensores de visión. Esta metodología se corresponde con la aproximación seguida por los humanos de guiarse por elementos reconocibles en su camino sin realizar ningún tipo de medida.

Desde los años 1990s las aproximaciones probabilísticas han sido dominantes en el campo de la generación de mapas. De hecho, uno de los hitos más importantes en este campo fue la publicación de una serie de artículos por Smith [Smith86, Smith90] que mostraban una aproximación probabilística muy exitosa que condujo a la realización de numerosos trabajos de investigación en este mismo sentido. Solo unos pocos artículos [DiMarco03, DGLCV01] se abstuvieron de seguir esta corriente probabilística.

Una metodología muy exitosa para afrontar el problema de la generación de mapas es la de su generación incremental. Para generar mapas con este método la información captada con los sensores láser es incorporada a un mapa global usando una aproximación avariciosa (*greedy*). Al computar la pose ideal del robot (por máxima probabilidad, etc.) los datos del láser actuales se incorporan usando como base dicha aproximación. Una vez incorporados al mapa, los escáneres son indistinguibles unos de otros y no hay ya un mecanismo para corregirlos en caso de error.

Thrun [Thrun00] presenta un método para generar mapas que combina un método de expectación máxima con estimadores de máxima probabilidad de la generación incremental del mapa. Gutmann et al. [Gutmann00] proponen la correlación de los escáneres para comprobar la consistencia del mapa en dos fases. Primero, los últimos escáneres se alinean y después de varias actualizaciones se aplica un procedimiento de detección de bucles para alinear los mapas locales.

Es importante también destacar la variedad de entornos físicos en los que las técnicas de SLAM han sido aplicadas. Así, en [Kim04, Hygounenc04] se estudian los resultados de aplicar SLAM a un problema de reconocimiento de terrenos aéreo. En [Sujan05, Sujan04] se aplica el SLAM a problemas de exploración planetaria, como los riscos y otros terrenos del planeta Marte. De todos modos, la mayoría de las aplicaciones del SLAM se han centrado en el estudio de la navegación de un robot en un entorno interior altamente estructurado.

2.4.2 Percepción robótica para el SLAM

Los sensores más comúnmente usados comprenden los basados en láser, los basados en sónar, y

los basados en sistemas de visión artificial.

Los sistemas de medición de alcance basados en láser consisten de sistemas activos altamente precisos. Tienen sin embargo la desventaja de que se limitan a medir la distancia entre dos puntos lo que dificulta su uso en labores de reconocimiento semántico (reconocimiento y seguimientos de objetos). Los sistemas basados en sónar son rápidos y baratos pero no muy precisos. Aún así, varias aplicaciones con sónar han sido reportadas [Kleeman95, Kleeman99] desde el inicio de la actividad investigadora en este campo. Por otro lado, cuando la tarea del SLAM depende únicamente de los sensores odométricos del robot, los errores tienden a ser muy grandes. Esto se debe a que en este caso no hay forma de corregir los pequeños errores que se van acumulando en la estimación de pose del robot. Existen aún otras posibles fuentes de información para un robot como pueden ser otros tipos de sensores de corto alcance (infrarrojos, táctiles) pero los resultados observados son generalmente pobres [Thrun02]. Se deduce que es necesario dotar al robot de sensores de relativamente largo alcance.

Una de las técnicas de percepción más populares es la basada en sensores de visión artificial. Este método es destacable por largo alcance, alta resolución, y propiedad pasiva (no emite energía lo que permite su combinación con sensores sensibles al calor como los infrarrojos). La investigación en el campo de la visión artificial ha producido grandes avances que pueden ser usados para su aplicación al problema del SLAM. Algunos de los avances más relevantes incluyen el modelado de escenas para la construcción de mapas del entorno, el análisis del movimiento de la cámara y muchos otros. Gracias también a los avances en hardware, en las matemáticas aplicadas a la visión por ordenador y en las técnicas de abstracción de características las soluciones al SLAM basadas en visión artificial han obtenido resultados muy robustos.

Wolter et al. [Wolter04] enfatizan que para la resolución del SLAM el problema debe descomponerse en dos aspectos: (1) El manejo de las características del mapa (extracción de características del entorno a partir de datos sensoriales) y (2) la gestión de la incertidumbre. En este sentido las técnicas de visión artificial han conseguido realizar el seguimiento y reconocimiento de características bastante bien y capacitan una reconstrucción tridimensional robusta que resulta esencial para la construcción del mapa. Por otro lado, las teorías de estimación probabilística ofrecen numerosos paradigmas para minimizar el error en presencia de ruido. De esto que haya un movimiento muy fuerte en recientes esfuerzos investigadores para combinar ambas técnicas.

Otra metodología que puede resultar fundamental es la del *scan-matching* ya que permite establecer asociaciones de los datos observados en dos instantes diferentes de la trayectoria del robot. Con esto se pretende definir un método que permita calcular el desplazamiento relativo del robot en diferentes pasos de su movimiento. Más información al respecto se presenta en el capítulo 2.5

2.4.3 Gestión de la incertidumbre

La gestión del ruido presente en los sensores y comandos de control de un robot se suele realizar de forma probabilística en las soluciones del SLAM. Desde este punto de vista de formulación probabilística, las metodologías de SLAM toman dos formas:

- *Online SLAM*. En este procedimiento se estiman las probabilidades posteriores de la pose actual a la vez que las del mapa: $p(x_t, m | z_{1:t}, u_{1:t})$. Esto requiere la estimación de las variables que persisten el tiempo t . Normalmente estos algoritmos descartan las medidas y controles una vez procesadas.
- *Full SLAM*. En esta caso, el procedimiento consiste en estimar las probabilidades posteriores sobre toda la trayectoria $x_{1:t}$ a la vez que el mapa en vez de estimar sólo la pose actual x_t .

Los tres métodos más comunes para abordar esta problemática son los de Markov, Kalman, y Monte Carlo de forma paralela a lo que ya se habían discutido anteriormente en el marco del problema de la localización.

El algoritmo de SLAM original [Thrun05] es el basado en el filtro de Kalman extendido (EKF). Este algoritmo usa el EKF para un *online SLAM* usando el principio de máxima probabilidad para la asociación entre datos. Es una aproximación que incluye una serie de suposiciones que implican ciertas limitaciones: (1) el mapa debe estar compuesto de puntos notables con tan poca ambigüedad como sea posible, (2) como en otros algoritmos de Kalman, la suposición de ruido gaussiano blanco para los modelos de movimiento y percepción del robot debe ser aceptable, (3) el algoritmo sólo puede procesar el reconocimiento positivo de puntos notables pero no puede notar su ausencia.

Otro ejemplo de aproximación de SLAM, presentado en [Thrun05], es el denominado algoritmo EIF (*Extended Information Form*). Este algoritmo tiene en común con el EKF en que representa la estimación posterior con funciones gaussianas pero, al contrario que el EKF, el algoritmo EIF resuelve el problema de *Full SLAM*. Así, en vez de definir las probabilidades posteriores sobre el mapa y la pose más reciente, define las probabilidades posteriores sobre el mapa y la trayectoria completa del robot. El algoritmo EKF representa pues una aproximación incremental que permite al robot crear un mapa del robot sin problemas de gestión de memoria mientras que el EIF es óptimo para problemas en los que el tamaño del mapa es fijo y se puede mantener un espacio de memoria dedicado para todos los datos necesarios hasta el momento en que se construya el mapa.

Madhavan et al. [Madhavan04] presentan una versión distribuida del filtro extendido de Kalman para la localización cooperativa multi-robot y la consecuente generación de mapas en zonas exteriores. Esta aproximación mejora las estimaciones generadas gracias a la información sobre

las posiciones relativas que unos robots tienen de otros. Al compartir cada robot la información que posee, todos los robots pueden obtener partes desconocidas del mapa del entorno para añadir las a su propia versión y, en caso de solape, rectificarlas. Otros algoritmos para el SLAM multi-robot se presentan en [Rekleitis02], [Mourikis06] y [Thrun01]. Stroupe et al. [Stroupe00] presentan una variante de este tipo de algoritmos cooperativos pero aplicada a la localización de objetos en vez de la auto-localización. En [Julier03], se ofrece otra alternativa en la que varios filtros implementan su propia versión del algoritmo SLAM con la capacidad de intercambiar información entre ellos antes de generar los mapas.

El *ScanSLAM* es una generalización del EKF-SLAM basada en escáneres láser que no necesita de ningún modelo geométrico. Nieto et al. [Nieto05, Nieto07] propusieron este algoritmo general en el que las características son directamente definidas a partir de las medidas láser y la pose del robot cuando los datos fueron adquiridos. En [Nieto07] también se muestra una técnica de muestreo para computar la matriz de varianzas de ICP y un procedimiento de scan-matching para ampliar las plantillas de puntos notables.

Diosi y Kleeman [Diosi05] aplican el *ScanSLAM* al PSM tomando la inspiración de la metodología de Atlas. La implementación es similar a la de Nieto et al. en el que un punto notable sólo se actualiza si la pose del robot es lo suficientemente cercana a él. Una extensión a un *ScanSLAM* múltiple se presenta en un entorno simulado fusionando las informaciones producidas por cada láser antes de incorporar el escáner [Tungadi07].

Otra aproximación denominada *graphSLAM* para solucionar el problema del SLAM se basa en la definición de las relaciones entre las diferentes entidades del mapa o de la pose del robot. En este caso, la pose y el mapa se representan internamente como un grafo donde las aristas se construyen a partir de restricciones particulares a cada vértice.

Biber et al. [Biber06] proponen una técnica para realizar múltiples procesos de *scan-matching* simultáneos para ser empleados en un SLAM relacional. En este método, el *scan-matching* se basa en la minimización de una función de energía y la pose resultante se usa en un grafo donde los nodos con los escáneres láser y los vértices representan las relaciones binarias entre ellos.

Bosse et al. [Bosse03, Bosse04] desarrollaron Atlas, una aproximación híbrida métrico-topológica diseñada para entornos de gran tamaño. El grafo se define de modo que los nodos representan los marcos locales y los vértices las transformaciones entre ellos. Dentro de cada marco se establece un algoritmo de mapas por *scan-matching* para cerrar los bucles. La metodología se puede aplicar tanto a mapas locales basados en características como a los derivados por medio de *scan-matching* a partir de escáneres láser.

Iser et al. [Iser08] describen una aproximación de *scan-matching* para realizar mapas topológicos. Grisetti et al. [Grisetti08a, Grisetti08b] desarrollan un algoritmo de optimización en tiempo real que mejora los resultados del *graphSLAM* por medio de una búsqueda estocástica por gradiente y

el uso de una parametrización basada en los árboles de la red. Como se asume que las restricciones al grafo son dadas, se puede usar en conjunto a la metodología Atlas para obtener dichas asociaciones de datos. Ciertas extensiones a este trabajo permiten también la resolución del problema de SLAM tridimensional.

Cabe destacar finalmente la presencia de otras aproximaciones que no hacen uso de este tipo de técnicas probabilísticas. Concretamente, la teoría de estimación de pertenencia a conjuntos (SM, del inglés *Set Membership*) proporciona una alternativa útil usada en [DiMarco01, DiMarco03, Milanese96]. Estas técnicas formulan la incertidumbre para el SLAM con la suposición de que el error es desconocido pero acotado (UBB, del inglés *Unknown But Bounded*) en vez de usar modelos estadísticos. Esta aproximación modela el ruido en el movimiento y pose absoluta del robot así como el ruido en los sensores externos de los que se miden, por ejemplo, el alcance a otros objetos. El concepto es que todos estos ruidos están acotados por ciertos valores escalares que permite incluirlos en conjuntos de medida. Los conjuntos de medida se integran con el conjunto de lo factible (que representa el modelo dinámico del robot) para dar los resultados del SLAM.

2.4.4 Otra retos en el problema del SLAM

Aparte de la gestión de la incertidumbre existen otros retos importantes a la hora de elaborar con éxito un algoritmo que resuelva el problema del SLAM.

Asociación de datos

Este problema se refiere al de encontrar la correspondencia entre características observadas en posiciones diferentes. Dos aplicaciones comunes de este tipo de asociación serían la del emparejamiento de dos escenas consecutivas o la del cierre de un bucle de una trayectoria larga cuando el robot ha de reconocer un punto notable por el que ya ha pasado. Formalmente, si un ítem n_t se usa para etiquetar la característica $i f_t^i$ adquirida en un tiempo t y si $n_t = j \leq N$ donde N es el número total de características en el mapa, entonces la característica i corresponde a la característica j observada anteriormente. De otro modo, si $n_t > N$, f_t^i es una característica nunca vista antes. Una técnica habitual para determinar el valor de n_t es aplicar el principio de máxima probabilidad. Por ejemplo, la probabilidad de una asociación de datos n_t puede ser evaluada a través de la derivación de las aproximaciones de Chapman-Kolmogorov, Monte Carlo y Bayes [Montemerlo02]. Al probar todas las características observadas, el resultado del máximo de la probabilidad con respecto n_t se compara con un umbral determinado empíricamente para determinar si $i f_t^i$ corresponde a una característica j ya existente o si es una característica completamente nueva. Otras técnicas usadas para resolver el problema de la asociación de datos se pueden encontrar en [Hahnel03, Neira01].

Cierre de bucle

El propósito de efectuar el cierre de un bucle es construir un mapa consistente para entornos a

gran escala. Es habitual encontrar problemas cuando el robot vuelve a la posición de partida ya que por culpa de los errores acumulados a lo largo de la trayectoria del robot el mapa reconstruido no suele ser consistente. Es decir, el bucle de la trayectoria no se suele cerrar correctamente. Una asociación de datos correcta es necesaria para identificar de forma inequívoca los puntos notables ya vistos antes y es esto lo que permitiría cerrar el bucle. Una vez que esto se ha conseguido se pueden aplicar diferentes técnicas para corregir el mapa como por ejemplo la de la combinación del filtro de Kalman y el método de máxima probabilidad. En [Kaess05] se proporciona una comparativa de las aproximaciones usadas para resolver este problema. Del artículo se observa que los métodos de Monte Carlo y de máxima probabilidad tienen la capacidad de resolver este problema mientras que el método de Kalman no lo puede conseguir por sí solo. También, una importante aproximación para la resolución del problema del cierre de bucle usando visión por ordenador se presenta en [Se05].

SLAM usando referencias visuales únicamente

Existe un gran reto en la inicialización en todas las soluciones visuales al problema del SLAM debido a las limitaciones de la visión artificial. Generalmente, no es posible calcular la distancia a cada objeto con precisión de una sola medida con lo que sólo el ángulo a cada punto notable del entorno se encuentra disponible. Por ello, una estimación de la posición de cada punto notable del entorno no va a ser posible hasta que se realicen diferentes medidas desde diferentes puntos de vista [Solà05]. A partir de aquí, se pueden identificar dos tipos de aproximaciones:

- Inicialización retrasada. Debido a que la visión artificial pierde una dimensión del entorno, se deben definir requerimientos base para ser satisfechos. Esto lleva a tener que almacenar una serie de poses del robot junto con sus correspondientes medidas. Así la inicialización de la estimación de los puntos notables del entorno se retrasa hasta que la base de medidas es suficiente como para establecer una estimación gaussiana. Un ejemplo de aproximación para hacer esto incluye la detección del alcance a los objetos por filtros de partículas [Davison03]. La inicialización se retrasa hasta que la varianza es más pequeña que un valor establecido por una estimación gaussiana creada por EKF.
- Inicialización sin retrasos. [Kwok04] presenta una aproximación en la que un conjunto de todas las hipótesis incluyendo la de la posición de todos los puntos notables del entorno se incluyen en el mapa inicial. En sucesivas observaciones la prueba de cociente secuencial (SRT, del inglés *Sequential Ratio Test*) basado en la probabilidad es usada para eliminar las hipótesis incorrectas y se usan aquellas con máxima probabilidad para actualizar el mapa. [Solà05] presenta un sistema SLAM basado en una solución federada para compartir información. Esta técnica permite lanzar la computación desde una inicialización basada en un filtro gaussiano de suma (GSF, del inglés *Gaussian Filter Sum*). En principio, esta técnica sigue un conjunto de mapas, uno para cada hipótesis lo que causa que el coste computacional crezca geoméricamente. Sin embargo, en [Solà05] se presenta una alternativa para que el incremento computacional sea solamente lineal.

En general, los métodos sin retrasos son superiores a los que no incluyen esta metodología por dos motivos: (1) Los métodos sin retrasos evitan el innecesario almacenamiento de poses y medidas, (2) En casos de navegación por entornos exteriores puede que no se llegue a tener los requerimientos para una base suficiente si el robot se mueve en línea recta y las cámaras enfocan hacia delante.

El problema del robot secuestrado

Cómo se describió anteriormente, no hay un algoritmo de localización que garantice la ausencia de fallos. En general, son puntos notables globales y sensores (como el GPS) los que pueden ayudar a recuperar el robot. Interesantemente, las soluciones propuestas para este problema pueden ayudar a mejorar los algoritmos de SLAM. En [Wang05], se presenta una aproximación de SLAM desacoplado (D-SLAM, del inglés *Decoupled SLAM*) donde se gestiona el problema como uno de localización y otro de generación de mapas con puntos notables estáticos. Los dos problemas se resuelven en paralelo. Uno representa el problema de SLAM normal y el otro es el problema del robot secuestrado dado por la observación actual y el mapa generado en la última secuencia. La solución normal de SLAM se usa a menos que se pueda cerrar el bucle; en este caso la solución dada por la aproximación global es superior y reemplaza a la aproximación rutinaria. Esta combinación ofrece mejores estimaciones de localización que las obtenidas por cualquiera de las otras dos técnicas por separado.

2.5 Scan-matching

El término *scan-matching* se refiere a la familia de algoritmos que resuelven el problema de la falta de alineamiento entre dos conjuntos de medidas de alcance obtenidas por sensores láser del entorno o de un escáner y su entorno. Del mismo modo se denomina *scan-matcher* al algoritmo que proporciona el cambio de pose necesario para conseguir que los dos escáneres se alineen para obtener el movimiento relativo y resolver el problema de la localización del robot.

Uno de los algoritmos más importantes creados para este propósito por Besl y McKay [Besl92] en 1992 es el llamado ICP (del inglés, *Iterative Closer Point*). En su caso, la aplicación de interés fue el estudio de las imágenes en tres dimensiones, siendo los autores capaces de demostrar la convergencia monótonica del algoritmo al punto local mínimo más cercano. En 1997, Lu y Milios [Lu97] publicaron una de las primeras aplicaciones del ICP a la robótica móvil. Su trabajo demostró la capacidad de esta metodología para resolver el problema de la localización. Así, en [Lu97] se exploran dos algoritmos: (1) El primero explora las direcciones tangentes y se le considera como el más rápido, (2) El segundo, que es el ICP clásico de punto a punto resuelto por la técnica de mínimos cuadrados, es más lento pero más estable. Ambos métodos pueden ser usados secuencialmente constituyendo el núcleo del método IDC (del inglés, *Iterative Dual Correspondences*). Rusinkiewicz y Levoy [Rusinkiewicz01] proporcionan una clasificación de los

tipos más comunes de algoritmos ICP incluyendo sus diferencias en velocidad de convergencia. Se concluye que la métrica punto a punto junto con la búsqueda de puntos correspondientes basada en proyección asegura los resultados más confiables y una convergencia más rápida que el ICP clásico de punto a punto.

Debido al ruido intrínseco en las medidas y en el movimiento del robot, los escáneres láser no pueden ser superpuestos completamente. Esto lleva a definir técnicas especiales para eliminar los puntos fuera de lugar (*outliers*) en el conjunto de correspondencias y así reducir el error en la búsqueda de la pose del robot. En [Chetverikov02], Chetverikov propone una versión robusta del ICP denominada *Trimmed ICP* y, en [Zhang94] Zhang demuestra un método adaptativo para el descarte de los puntos fuera de lugar.

Muchas otras variantes del algoritmo ICP han sido propuestas con el objeto de mejorar su velocidad y robustez. Censi [Censi08] demuestra que un método basado en una métrica punto a línea (que usa la información dada por el vector normal a la superficie) converge cuadráticamente en un número de pasos finitos. Minguez [Minguez05, Minguez06] usa una métrica de minimización diferente a la norma euclidiana común que permite tener en cuenta las componentes de traslación y rotación proporcionando una robustez muy alta. Diosi y Kleeman [Diosi05, Diosi07b] proponen un método para mejorar la velocidad de convergencia minimizando el tiempo dedicado a la búsqueda de correspondencias. Para ello, se define el problema en coordenadas polares (algoritmo PSM) dando lugar a soluciones más estables y rápidas que el ICP clásico.

En otro tipo de aproximaciones con éxito se encuentra la comparación de histogramas [Bosse07] en la que se generan dos de estos histogramas: uno para la rotación y otro para la traslación. Cabe también destacar el trabajo publicado en [Konolige99] y [Olson09] en las que el proceso de *scan-matching* se basa en el uso de tablas de correlación. Otros autores han investigado el uso de algoritmos genéticos [ZeSu07] aplicados a la solución del PSM.

3 Procedimiento general

Este trabajo presenta un procedimiento mejorado de PSM, basado en el trabajo original de Diosi [Diosi07a], que en este trabajo se va a denominar E-PSM (del inglés, *Enhanced Polar Scan Matching*). La metodología consiste de un algoritmo de *scan-matching* relativo, de punto a punto, en coordenadas polares. Esta aproximación fue elegida por la rapidez que otorga trabajar en el sistema de coordenadas nativo del escáner láser y la facilidad que el PSM permite a la hora de buscar correspondencias entre los datos de dos escáneres láser.

Se presentan varias mejoras sobre el PSM original con el objeto de mejorar su precisión pero manteniendo en lo posible su rapidez y, por tanto, su capacidad para ser usado en tiempo real. En primer lugar, un avanzado algoritmo de segmentación de línea es aplicado a las lecturas de los sensores láser en vez de las normas heurísticas usadas en [Diosi07a]. En segundo lugar, se añade un nuevo criterio para seleccionar los puntos del escáner aptos para el proceso de comparación. Finalmente, se definen nuevas mejoras en los métodos usados para mejorar los procesos de búsqueda de la orientación y la posición del robot. Como alternativa, se incluye la posibilidad de usar técnicas de optimización local o global para buscar la posición y orientación del robot de forma simultánea.

El algoritmo de *scan-matching* es aplicado cada vez que el robot detecta un cambio significativo en su pose (usando las medidas de sus sensores odométricos) o, en su defecto, después de que pase una cantidad predeterminada de tiempo. El procedimiento de *scan-matching* que se efectúa al cumplir una de estas dos condiciones se presenta en la Figura 1. Se pueden distinguir claramente cuatro pasos principales:

- Filtrado y segmentación de los escáneres de referencia y actuales (*reference scan & actual scan*). Para ello se usan los algoritmos diseñados por Cuadra [Cuadra11b] según son descritos en el Capítulo 4.
- Proyección e interpolación del *current scan* al eje de coordenadas de referencia siguiendo la aproximación de Diosi [Diosi07a]. Información adicional con respecto a este proceso se presenta en el capítulo 5.1
- Descarte de los datos de alcance del escáner que no deban ser usados por no ser adecuados para el proceso de comparación. El capítulo 5.2 proporciona un resumen de la aproximación propuesta en este trabajo.
- Búsqueda de la pose del robot que permita alinear de manera óptima el *current scan* y el *actual scan*. Se presentan dos opciones en las que se puede realizar la búsqueda de orientación y posición de manera alterna o simultánea según se presenta en los capítulos 5.3.1 5.3.2 respectivamente.

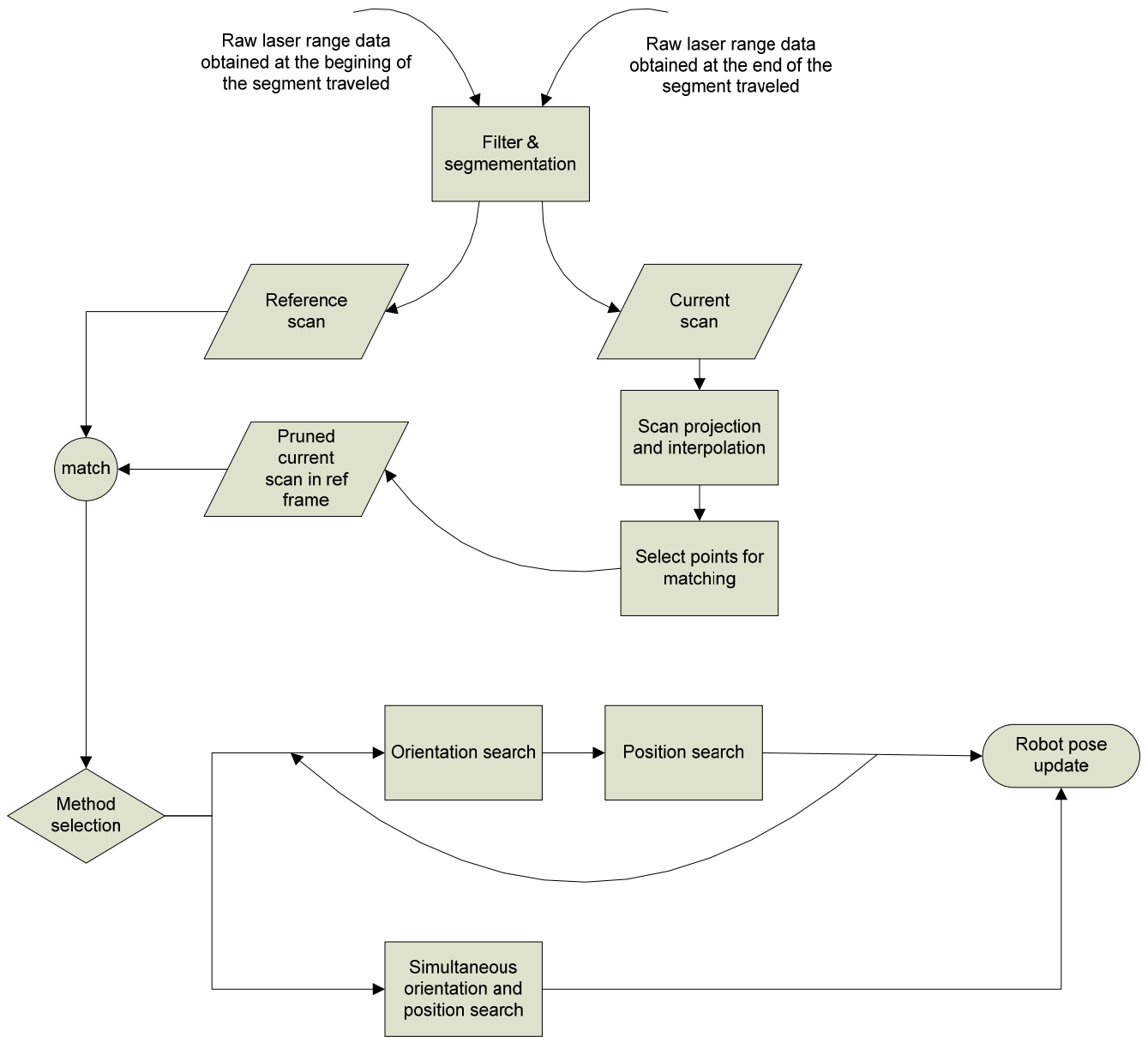


Figura 1: Procedimiento general

4 Algoritmo de segmentación de línea

Cuadra [Cuadra11b] desarrolló un nuevo algoritmo para la extracción de líneas en coordenadas polares. Para poder manejar niveles de ruido variables en las lecturas del escáner láser, este algoritmo usa un modelo de ruido en el que se asume que los errores tienen una desviación estándar proporcional a una función conocida de la medida esperada. El proceso completo se compone de tres pasos principales: (1) Un procedimiento de *clustering* mediante *scale-space* (2) Un algoritmo de regresión de línea basado en EKF partiendo de los puntos semillas identificados en el paso anterior y (3) Un procedimiento de *merging* para segmentos similares adyacentes.

La Figura 2 (se observa el robot trasladándose de izquierda a derecha) presenta un ejemplo de la aplicación de este algoritmo a un par de escáneres (*reference scan* y *actual scan*) con una desviación estándar en sus medidas de 20mm. Se presenta en verde los escáneres ruidosos originales y en azul el resultado de aplicar el algoritmo de filtrado y segmentación. Este ejemplo sirve de muestra de la gran capacidad de este algoritmo para eliminar el ruido del entorno y facilitar la tarea del proceso de *scan-matching* como se presentará en capítulos posteriores. A continuación se presentan más detalles de los procesos del algoritmo de filtrado y segmentación según se presentan en [Cuadra11b].

4.1 Clustering mediante scale-space

El procedimiento de *clustering* conocido como *scale-space* ha sido adaptado al problema de obtención de segmentos en coordenadas polares. Esta técnica se aplica con el objetivo de determinar de forma aproximada los segmentos del entorno y usar sus puntos internos como semillas para inicializar un filtro de Kalman como se expone en la sección siguiente.

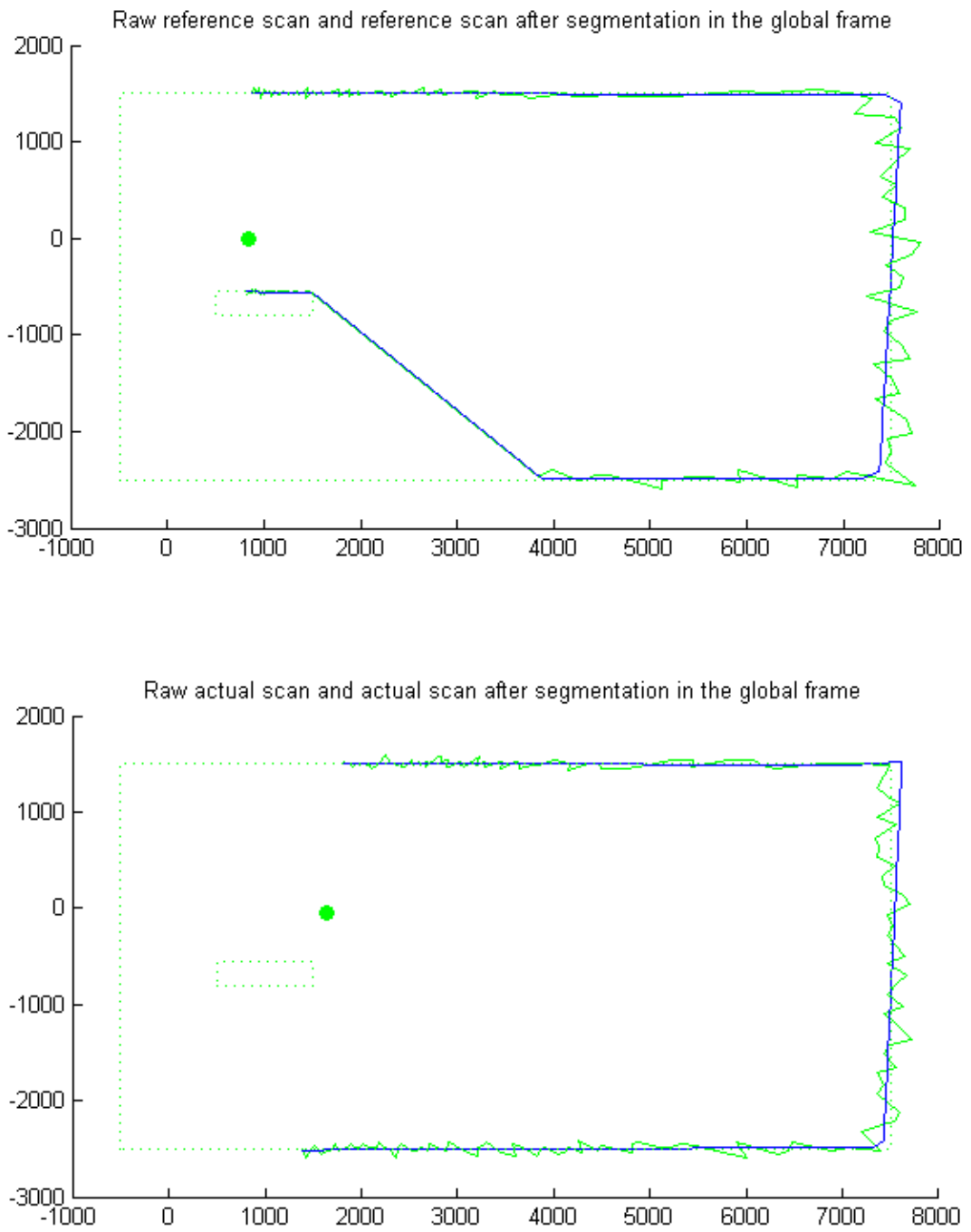


Figura 2: Ejemplo de aplicar el proceso de filtrado

Para la extracción de los extremos de los segmentos que determinan un entorno 2D se utilizan la primera y segunda derivadas de la representación *scale-space* de la señal. Esta aproximación surgió de la observación de la forma de las señales de mediciones radiales en coordenadas polares. Según se expone en [Cuadra11b], las pruebas con entornos simulados sin ruido muestran que mediante las curvas de nivel cero de ambas derivadas era posible localizar casi todos los extremos. Para el caso de mediciones con ruido, se incluyó un filtro para la reducción del ruido en el proceso de *scale-space*. Este filtro se basa en una caracterización previa de los efectos del ruido en las curvas de nivel obtenidas por el procedimiento de *scale-space*. Esta caracterización se obtuvo mediante el estudio estadístico de las curvas generadas para cada derivada, al alterar valores equiespaciados de un segmento usando patrones de alteraciones fijas de diversas intensidades. Adicionalmente, esta caracterización se refina mediante un procedimiento de filtrado estadístico on-line que elimina gran parte de las curvas de nivel generadas por el ruido, adaptándose al nivel de ruido presente en cada momento.

El filtrado estadístico también proporciona un nivel inferior del parámetro de escala para el procedimiento del Árbol de Intervalos de Witkin [Witkin84] que extrae las características más estables del entorno (los posibles extremos de segmentos) a partir de las curvas de nivel. La elección del nivel inferior se realiza de una forma diseñada para el problema concreto de las mediciones radiales y adaptada al nivel de ruido presente en el entorno en cada momento. El filtrado de curvas de nivel y el nivel inferior adaptativos aumentan sensiblemente la velocidad de recorrido del árbol y disminuyen el grado de fragmentación en la representación obtenida del entorno.

Por último, el proceso de *clustering* incluye un procedimiento para la fusión de la información obtenida a partir de ambas derivadas sobre los posibles extremos de segmentos. A partir de esta información se eligen zonas en el interior de los posibles segmentos. Estas zonas constituirán las semillas para la inicialización de los filtros de Kalman presentados en la sección siguiente.

4.2 Estimación de líneas mediante EKF

Cuadra [Cuadra11b] extiende el trabajo original de Peña [Peña85], quien había formulado un filtro de Kalman equivalente al procedimiento de estimación lineal por Mínimos Cuadrados Ordinarios (MCO), al procedimiento de Mínimos Cuadrados Generalizados (MCG) lineales contemplando además el caso en el que dicha varianza pueda ser desconocida bajo ciertas condiciones. En este caso, Cuadra adaptó el método de estimación en dos etapas, conocido como MCG factibles, al filtro de Kalman. Nótese también que el MCG incluye un modelo con varianza del error no constante al contrario que en el MCO. Se incluyó además un test de detección de *outliers*.

Con el objetivo de usar la ecuación de la recta en forma polar y así modelar adecuadamente la toma de mediciones radiales, el procedimiento de estimación en dos etapas se extendió además a modelos no lineales mediante un Filtro de Kalman Extendido. Así, Cuadra usa un modelo de medidas en el que la desviación típica del ruido es proporcional a la distancia medida esperada con constante de proporcionalidad desconocida.

El filtro propuesto proporciona, para los dos niveles de ruido y las dos amplitudes, resultados de precisión comparable a los obtenidos mediante el conocido método de Gauss-Newton (aplicándolo también en dos etapas). Los resultados de la comparación con Gauss-Newton ponen de manifiesto la validez de dicha fórmula también para el caso no lineal.

La formulación anterior fue en principio realizada a partir de un conjunto de puntos (semilla) situado a un extremo del segmento. Dado que en los casos de aplicación no se sabe de antemano la posición de la semilla con respecto al segmento, Cuadra diseñó procedimiento de manera que los puntos se fueran introduciendo consecutivamente a ambos lados de la semilla. Con este procedimiento, denominado bilateral, se obtiene una reducción importante en los errores máximos de las estimaciones, manteniendo los mismos valores medios del error. La complejidad temporal del filtro, incluido el test de detección de *outliers* es $O(n)$, siendo n el número de mediciones.

4.3 Extracción de líneas y merging de segmentos similares adyacentes

Las semillas obtenidas mediante el procedimiento de *clustering* son usadas para la inicialización de filtros de Kalman definidos anteriormente. Para ello se obtiene a partir de los datos de una semilla una estimación de los parámetros de la recta. Esta estimación se obtiene por el procedimiento de la mediana repetida, que es robusto ante la presencia de *outliers* y parte de un criterio de minimización distinto al de mínimos cuadrados. Esta diferencia permite suponer cierta independencia entre las estimaciones obtenidas mediante la mediana repetida y de métodos basados en mínimos cuadrados (como el filtro de Kalman propuesto), con el objetivo de reutilizar los puntos de las semillas en el proceso de filtrado. El test de detección de *outliers* necesita que se hayan filtrado varios puntos para que sus valores críticos no sean elevados. La inclusión de la semilla permite al test alcanzar esos valores críticos cuando en filtro está saliendo de la semilla por lo que facilita la detección temprana de *outliers*. Este test permite además definir un criterio para la detección de extremos de segmentos en la estimación secuencial mediante el filtro de Kalman. La detección temprana de *outliers*, debida a la inclusión de la semilla, concede al procedimiento la capacidad de extraer segmentos fiables más cortos.

Cuadra también estableció un criterio, basado en las varianzas unitarias, para resolver posibles solapes entre segmentos. Estos solapes aparecen especialmente cuando un nivel alto de ruido

hace que los límites de los segmentos no estén nada claros. El procedimiento también busca zonas que hayan quedado sin asociar a segmentos, que no correspondan a observaciones que faltan, y realiza en ellas un proceso de estimación. Tras esta fase del procedimiento todos los puntos del haz están clasificados como asociados a un segmento, como puntos aislados o como medidas que faltan. Para completar el procedimiento de segmentación, Cuadra usó la equivalencia entre la teoría de la regresión clásica y el filtro propuesto para establecer un criterio de mezcla de segmentos adyacentes colineales de manera que se reduzca la posible fragmentación de la representación del entorno obtenida.

Para comprobar el rendimiento y la eficiencia del procedimiento expuesto en este capítulo, en [Cuadra11b] se realizaron experimentos estadísticos con diversos niveles de ruido, en diversos mundos simulados y usando dos resoluciones angulares para el sensor. En estos experimentos se comparó el rendimiento del método propuesto con el del popular *Split and Merge* (SM) y se realizaron pruebas para establecer la corrección del método (ajuste de parámetros de la recta y localización de extremos) y la exactitud de las estimaciones. Cuando el ruido era bajo, el método de segmentación propuesto ha ofrecido resultados similares a los del popular *Split and Merge*, superándolo en la localización de extremos. A niveles de ruido alto el rendimiento del nuevo método propuesto se mostró netamente superior resaltando especialmente notable el bajo nivel de fragmentación producido. Esto implica que el método propuesto proporciona representaciones del mundo consistentes a simple vista, además de estadísticamente aceptables.

5 Procedimiento mejorado de *scan-matching*

5.1 Proyección de las medidas

Una vez que las medidas de láser actuales y de referencia han sido filtradas del ruido presente en las mismas, debemos encontrar la forma de poder compararlas. Para ello, se debe elegir un punto común de trabajo; en esta investigación, se ha elegido para este propósito el marco original de referencia. Esto significa que debemos determinar los valores de las lecturas de alcance obtenidas en la posición final del robot (*current scan*) como si hubieran sido obtenidas en la posición inicial (es decir, en el mismo marco que el usado para determinar el *reference scan*). Siguiendo a Diosi [Diosi07a], para cada retorno del láser con medida de alcance $r_{cur,i}$ obtenido desde la posición final del robot, sus coordenadas y orientación en el marco de referencia están dadas por la Ecuación 1 y la Ecuación 2. La Figura 3 proporciona una referencia visual con todos los parámetros usados en las ecuaciones citadas previamente.

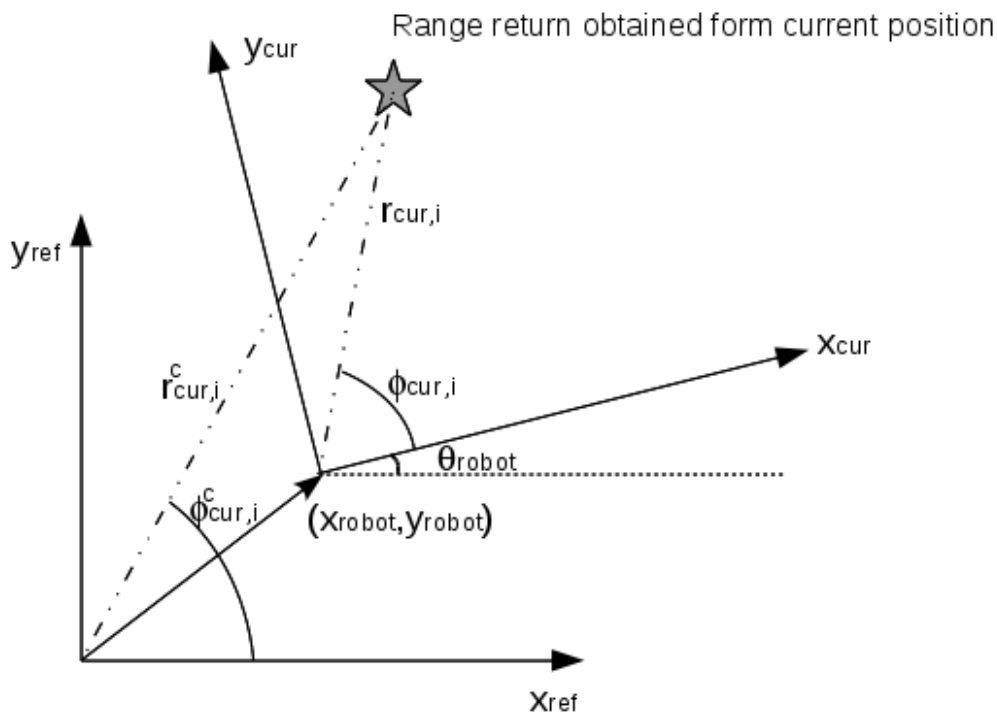


Figura 3: Ejes de coordenadas de referencia (ref) y actual (cur)

$$r_{cur,i}^c = \sqrt{(r_{cur,i} \cos(\theta_{robot} + \phi_{cur,i}) + x_{robot})^2 + (r_{cur,i} \sin(\theta_{robot} + \phi_{cur,i}) + y_{robot})^2} \quad (1)$$

$$\phi_{cur,i}^c = \text{atan2}(r_{cur,i} \sin(\theta_{robot} + \phi_{cur,i}) + y_{robot}, r_{cur,i} \cos(\theta_{robot} + \phi_{cur,i}) + x_{robot}) \quad (2)$$

donde

- $r_{cur,i}^c$ es la medida de alcance actual computada desde el marco de referencia correspondiente a la medida tomada en el ángulo $\phi_{cur,i}$
- $\phi_{cur,i}^c$ es el ángulo computado en el marco de referencia correspondiente a $r_{cur,i}^c$
- $r_{cur,i}$ es la medida de alcance obtenida por el robot desde su posición final en el ángulo $\phi_{cur,i}$
- $\phi_{cur,i}$ es el ángulo en la que la medida $r_{cur,i}$ fue obtenida (en el marco de referencia actual)
- $x_{robot}, y_{robot}, \theta_{robot}$ corresponden a las coordenadas cartesianas y orientación del robot en su posición final
- El subíndice i hace referencia al hecho de que la medida obtenida en el ángulo $\phi_{cur,i}$ es la obtenida por el sensor i que se encuentra situado en esta orientación del robot.

5.2 Interpolación y descartes

El siguiente paso consiste en interpolar los nuevos datos de alcance computados en los ángulos $\phi_{cur,i}^c$ a los ángulos en los que las medidas de alcance fueron originalmente obtenidas, $\phi_{ref,i}$. Además, se deben identificar qué puntos (ya sea en la lectura actual o en la de referencia) no son adecuados para ser parte del proceso de comparación que sigue a continuación. El resultado de este proceso debe ser un conjunto de parejas $\{r_{ref,i}^f, r_{cur,i}^f\}$ para cada ángulo de referencia $\phi_{ref,i}$ en el que haya una medida actual y de referencia de alcance válida.

Para los puntos contenidos en un mismo segmento (según lo identificado por el algoritmo de segmentación) se aplica una interpolación lineal básica. Es importante destacar que en este trabajo no se han aplicado el tipo de reglas heurísticas usadas por Diosi y otros autores para descartar medidas erróneas. En este trabajo es el algoritmo de segmentación el encargado de asegurarse que las medidas no validas (aquellas más allá del alcance máximo definido o fuera de cualquier segmento) no sean tenidas en cuenta en el proceso de *scan-matching*. La Figura 4 muestra un ejemplo para el caso de un robot dotado de únicamente siete sensores láser espaciados uniformemente en la su parte delantera. En este sencillo caso, la tarea de interpolación consistiría de calcular para $\phi_{ref,30}$ y $\phi_{ref,60}$ los correspondientes $r_{cur,30}^f$ y $r_{cur,60}^f$ usando las medidas calculadas $r_{cur,0}^c$, $r_{cur,30}^c$, $r_{cur,60}^c$ y $r_{cur,90}^c$ (no mostradas en la Figura 4).

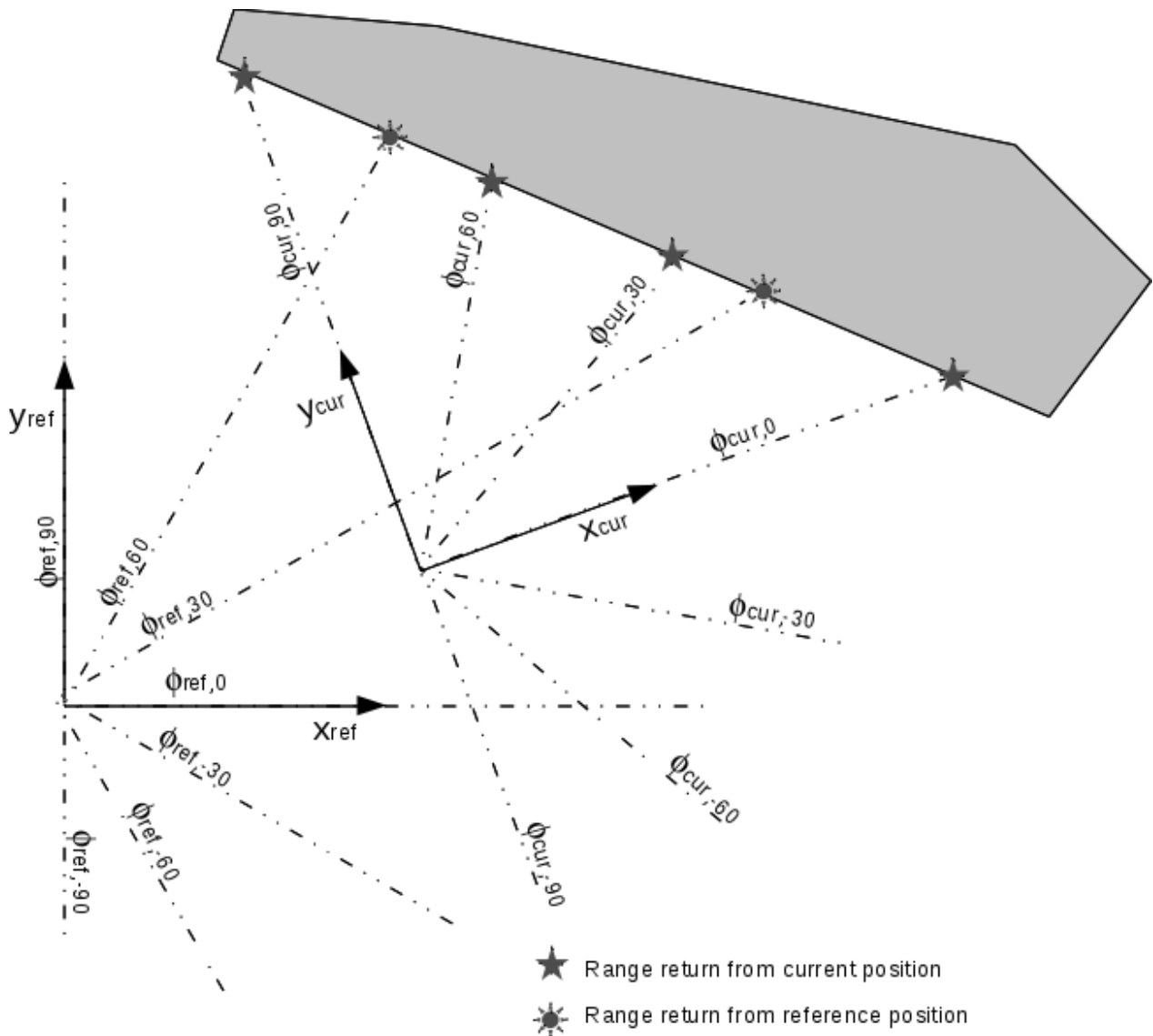


Figura 4: Interpolación de medidas láser en el marco de referencia

Diversas estrategias han sido definidas para tratar los casos de ocultamiento. En primer lugar, cuando varias medidas interpoladas $r_{cur,i}^f$ se computan para un mismo ángulo de referencia $\phi_{ref,i}$ solamente el menor de los valores disponibles es tenido en cuenta. Además, en este trabajo, se añade un nuevo criterio para descartar aquellas lecturas láser de referencia que no pueden ser observadas desde la posición final del robot (es decir, la lectura de referencia está fuera de la visión del robot desde la posición actual). En el caso particular en que el robot tenga sensores únicamente en el área que abarca 180° delante de sí mismo, este criterio se puede simplificar al cálculo de la coordenada x de la medida de alcance de referencia en los ejes de coordenadas

definidos por la posición actual del robot, como se presenta en la Ecuación 3.

$$x_{ref,i}^{cur} = (r_{ref,i} \cos(\phi_{ref,i}) - x_{robot}) \cos(\theta_{robot}) + (r_{ref,i} \sin(\phi_{ref,i}) - y_{robot}) \sin(\theta_{robot}) \quad (3)$$

donde $x_{ref,i}^{cur}$ es el valor de la coordenada x de la medida de referencia en el eje de coordenadas actual.

Con esta definición todas las lecturas de alcance de referencia con un valor de $x_{ref,i}^{cur}$ por debajo de cero se asumen invisibles desde la posición actual y son por tanto descartadas del proceso de *scan-matching*. Un ejemplo mostrando el resultado del proceso completo se presenta en la Figura 5. Nótese que la gráfica superior (en coordenadas polares) proporciona los valores de los $r_{cur,i}^c$ para este ejemplo (cruces, círculos y triángulos); en la misma gráfica, la línea discontinua representa las medidas de referencia contra las que los valores de $r_{cur,i}^c$ son comparados. Los puntos de los datos de los $r_{cur,i}^c$ fueron marcados con una cruz si se han considerados válidos para su uso en el proceso de *scan-matching*. Aquellos marcados con un círculo corresponden a los puntos $r_{cur,i}^c$ descartados por: (1) No tener un punto de correspondencia en la lectura actual, o (2) por haber sido identificado como no-válido por el algoritmo de segmentación. Los puntos marcados con un triángulo invertido corresponden a aquellos descartados por el nuevo criterio de oclusión definido en este capítulo. La gráfica inferior de la Figura 5 representa la misma información pero representado en el plano x-y junto con las posiciones inicial y final del robot (punto blanco y punto negro sólido respectivamente).

5.3 Actualización de la pose del robot

La pose actualizada del robot debe ser determinada de forma que minimice el error en la comparación entre las lecturas láser actuales y de referencia definidas anteriormente. Para ello, se debe en primer lugar establecer una medida del error en dicha comparación. Se debe recordar que después de la proyección de las lecturas y el proceso de descarte, se va a disponer (para aquellos ángulos de referencia para los que se dispone de lecturas válidas de referencia y actuales) de una correspondencia unívoca entre $r_{ref,i}$ y $r_{cur,i}^f$. De estos se puede definir, como muestra la Ecuación 4, la función objetivo f que será minimizada por el algoritmo de búsqueda:

$$f = \sum_{i=1}^n (r_{ref,i} - r_{cur,i}^f)^2 \quad (4)$$

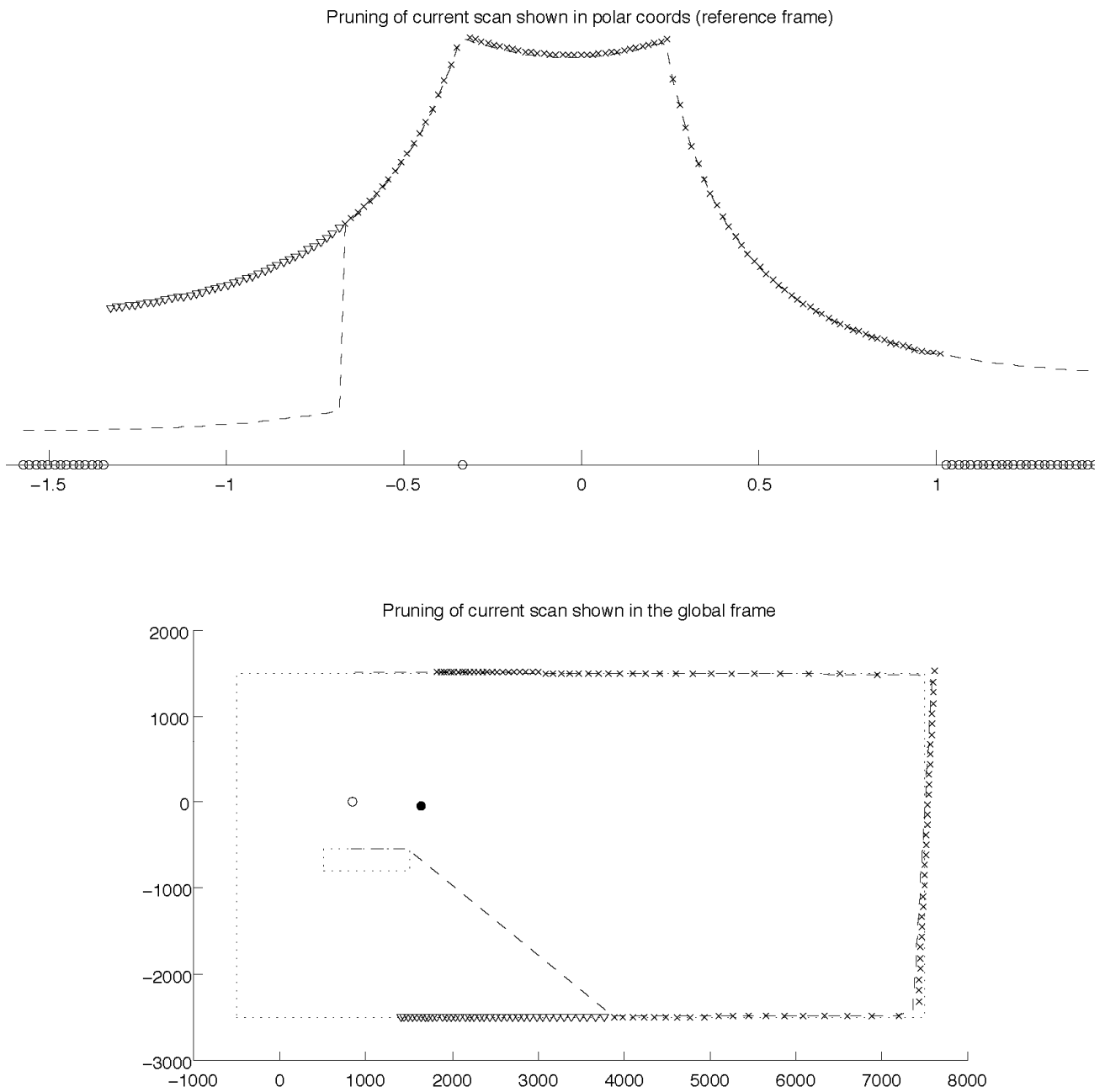


Figura 5: Ejemplo de uso del criterio de descarte por oclusión

Dos procedimientos generales se proponen para resolver este problema y se presentan en las secciones 5.3.1 y 5.3.2

5.3.1 Búsqueda alterna de orientación y posición

Siguiendo la aproximación general de Diosi, en este método se realiza primero la búsqueda de la orientación del robot θ_{robot} y a continuación se realiza la búsqueda de la posición del robot (x_{robot}, y_{robot}) . Este proceso es repetido hasta que se cumple con un criterio de convergencia dado o se alcanza el máximo de iteraciones definido. La Figura 6 muestra un esquema general del proceso de actualización de la pose. Una variante a este procedimiento se proporciona en la Figura 7.

El proceso de búsqueda de la orientación se basa en una aproximación de fuerza bruta. Arrancando de la estimación inicial obtenida de los sensores de odometría, una re-evaluación sistemática de la función objetivo f se realiza para cada grado en el intervalo $[\theta_{robot} - 10, \theta_{robot} + 10]$. Al contrario que en [Diosi07a], no se asume que los procesos de interpolación y descarte permanezca invariante con el proceso de cambio de orientación que implica la búsqueda del θ_{robot} . Así, aunque requiera mayor coste computacional, la aproximación desarrollada en este trabajo ofrece la ventaja de permitir la re-evaluación de la función objetivo para cualquier θ_{robot} arbitrario. Si no se encuentra un mínimo relativo en una primera instancia, el proceso de evaluación sistemático se expande en la dirección de error decreciente hasta alcanzar un mínimo relativo. Es importante notar que para cada re-evaluación de la función objetivo f para los diferentes θ_{robot} se va a requerir la realización de los procesos de proyección, interpolación y descarte. Una vez que el mínimo relativo es encontrado, se realiza en proceso de búsqueda refinado para cada 0.01 grados en el intervalo $[\theta_{robot} - 0.5, \theta_{robot} + 0.5]$.

Para la actualización de la posición del robot, la diferencia entre las lecturas láser de referencia y las actuales proyectadas se modela con una aproximación de primer orden como en [Diosi07a], ver Ecuación 5.

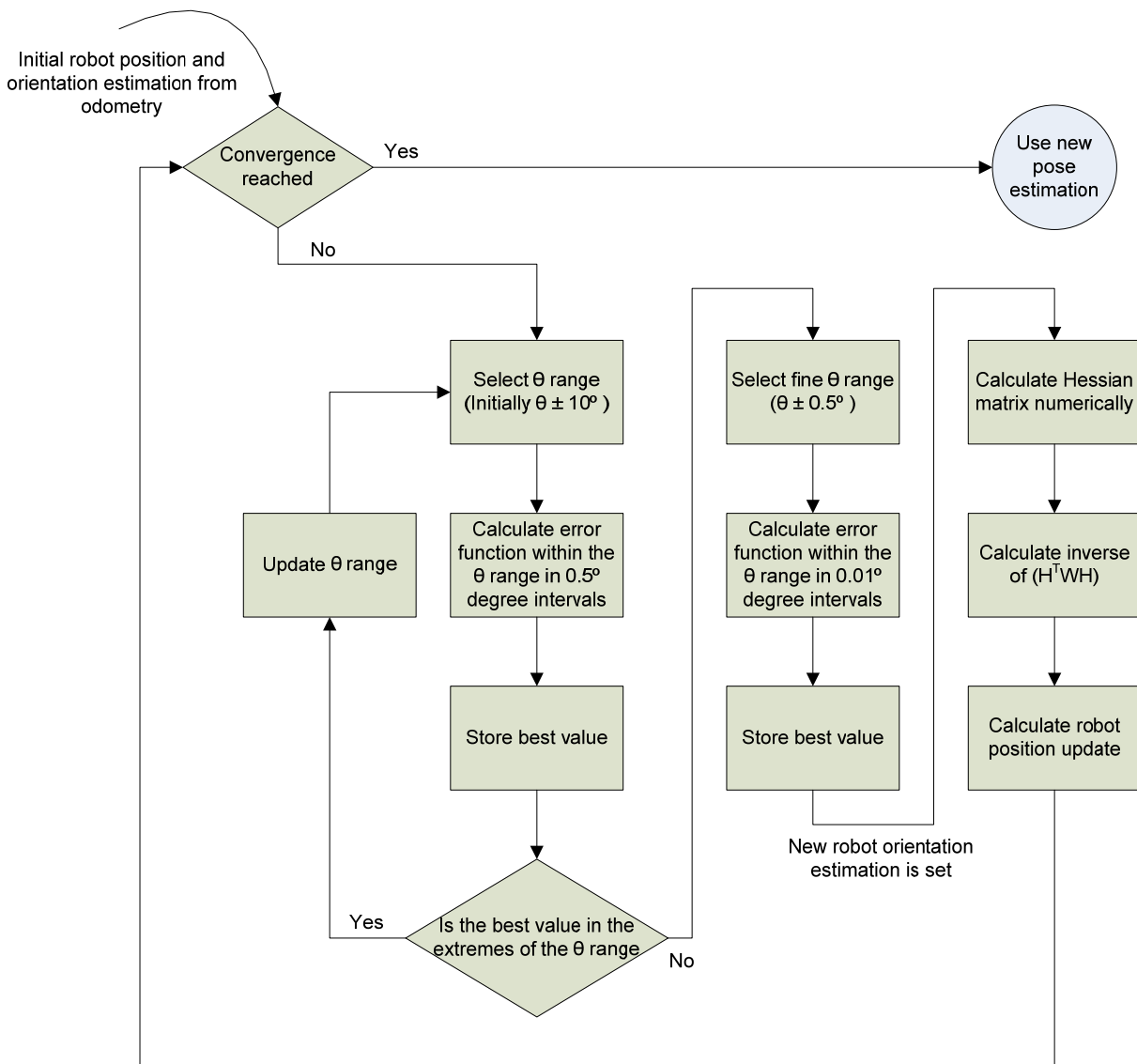


Figura 6: Búsqueda alterna de orientación y posición

$$\overrightarrow{r_{cur}^f} - \overrightarrow{r_{ref}} = H \begin{pmatrix} \delta x_{robot} \\ \delta y_{robot} \end{pmatrix} + \vec{v}$$

(5)

donde:

- El vector $\begin{pmatrix} \delta x_{robot} \\ \delta y_{robot} \end{pmatrix}$ representa el incremento de las coordenadas x, y de la estimación de posición del robot.
- La matriz H representa la matriz hessiana dada por filas del tipo $(\partial r_{cur,i}^f / \partial x_{robot}, \partial r_{cur,i}^f / \partial y_{robot},$
- \vec{v} representa el vector de ruido

La solución al problema lineal de mínimos cuadrados es bien conocida, ver [Nocedal99]. La Ecuación 6 proporciona su solución.

$$\begin{pmatrix} \delta x_{robot} \\ \delta y_{robot} \end{pmatrix} = (H^T H)^{-1} H^T (\overrightarrow{r_{cur}^f} - \overrightarrow{r_{ref}})$$

(6)

En este trabajo, la matriz hessiana se computa numéricamente usando el método de diferencias central. Esta aproximación difiere de la usada por Diosi; [Diosi07a] proporciona una solución explícita para el cálculo de la matriz hessiana que asume de forma implícita que no hay cambios en $\overrightarrow{r_{cur}^f}$ como resultado del proceso de interpolación y descarte. Esta suposición no ha sido usada en este trabajo. La sección 6.1 y la sección 6.2 proporcionan información detallada de los resultados obtenidos con esta aproximación.

Como alternativa a esta forma de computar la actualización de la posición del robot, se desarrolló un método alternativo (ver Figura 7). La motivación para presentar esta alternativa es que la solución dada por la Ecuación 6 es solamente válida en un sistema lineal cuando la variación de la posición es pequeña. Para no depender de la veracidad de esta suposición, se sustituyó la solución basada en mínimos cuadrados (Ecuación 6) por un proceso de búsqueda basado en método de optimización local. Detalles adicionales sobre la teoría de optimización se encuentran disponibles en [Nocedal99] y [Johnson]. El capítulo 6 presenta detalles de los algoritmos empleados para esta parte del proceso de búsqueda de la pose del robot. Como mejora adicional sobre el algoritmo de la Figura 6, se definió un paso adicional una vez que el algoritmo de scan-matching ha convergido a un valor estable. Este paso consiste en comprobar si la nueva pose propuesta por el algoritmo de estimación reduce de hecho el valor de la función objetivo f . Si no

es así, el nuevo procedimiento propuesto rechaza la estimación y actualiza la pose del robot usando únicamente como base los datos obtenidos de los sensores de odometría del robot. El efecto positivo de estas mejoras adicionales podrá ser comprobado en la sección 6.4

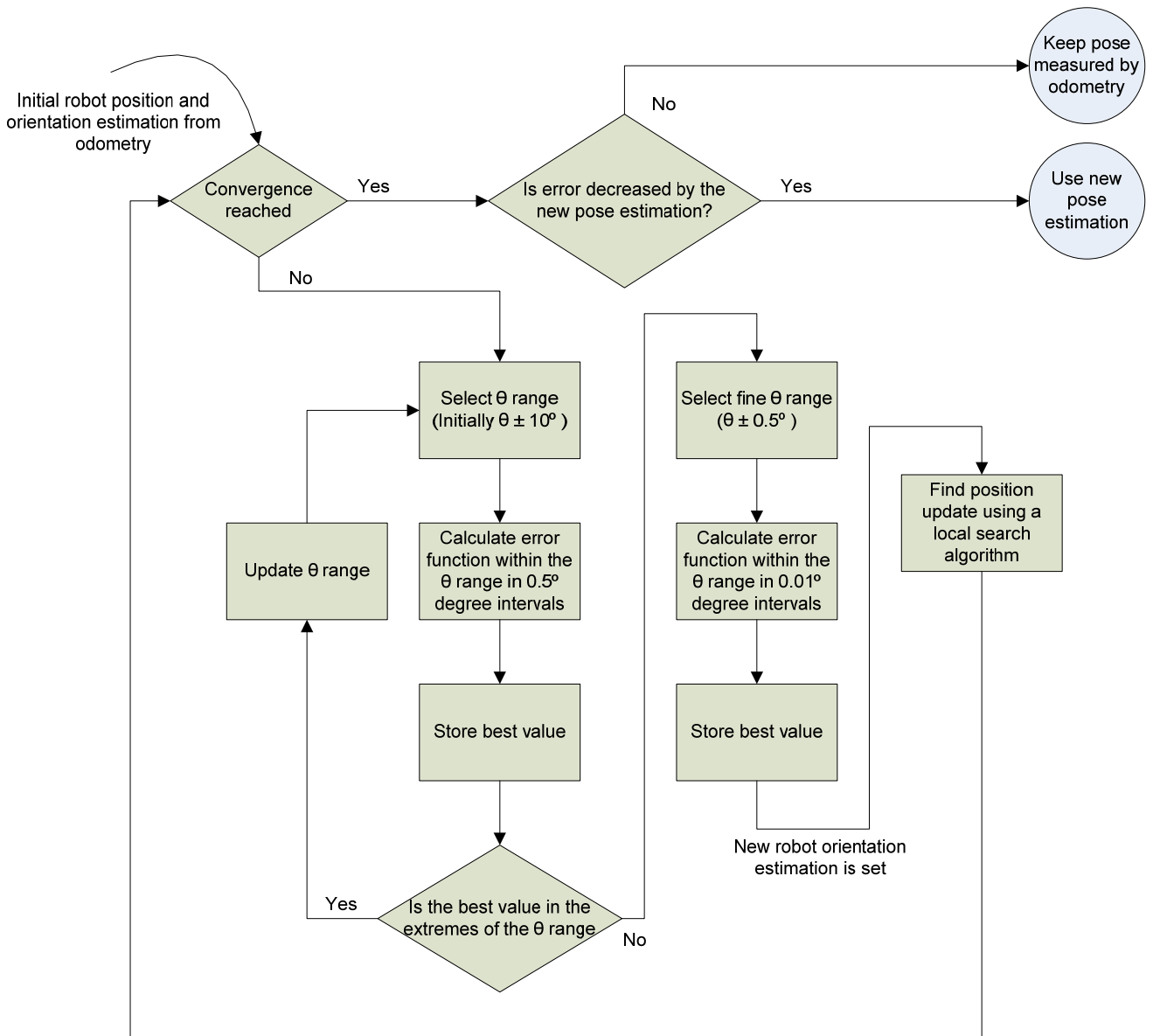


Figura 7: Búsqueda alterna de orientación y posición usando algoritmo de optimización local

5.3.2 Búsqueda simultanea de posición y orientación

Esta aproximación minimiza la función objetivo f (Ecuación 4) usando diferentes tipos de algoritmos de optimización locales y globales (basados en gradientes o no). En este caso, los procedimientos descritos en la Figura 6 y la Figura 7 se ven completamente sustituidos ya que se estima a la vez el valor de δx_{robot} , δy_{robot} , y $\delta \theta_{\text{robot}}$. En todos los casos, se hace uso de la implementación proporcionada por la librería *nlopt*. Detalles adicionales sobre la teoría de general optimización se encuentran disponibles en [Nocedal99]. [Johnson] proporciona detalles sobre todos los algoritmos usados y sus implementaciones. La sección 6.2 y la sección 6.4 proporcionan los resultados obtenidos al aplicar esta aproximación.

6 Resultados

Para evaluar las capacidades del procedimiento de *scan-matching* propuesto en este trabajo, se diseñaron varios casos de prueba para ser ensayados en el simulador denominado *cybersim*. Se debe destacar que *cybersim* es un entorno completo de simulación de robots en dos dimensiones desarrollado por José Manuel Cuadra Troncoso para su tesis doctoral [Cuadra11b]. Está escrito en C++ usando la librería Qt [Nokia11] y proporciona la posibilidad de simular de forma ideal o realista los robots, el entorno, los obstáculos, los estímulos, etc. Además, tiene herramientas como la generación automática de gráficos vectoriales o la de vídeos de simulaciones. Para probar los algoritmos descritos en este trabajo fue necesario modificar el código *cybersim* para incluir los algoritmos aquí definidos. Además se dotó a *cybersim* de la capacidad de ser usado en conjunción de la librería de optimización *nlopt* [Johnson].

Los casos de prueba definidos cubren los diferentes factores que pueden impactar la precisión y prestaciones del algoritmo. En primer lugar, el algoritmo fue probado sin ningún ruido (ni en las medidas láser ni en los datos de odometría). Esto quiere decir que la solución inicial de la pose corresponde en realidad a la pose exacta del robot (ya que no hay ruido odométrico) y que además todas las lecturas de alcance tomadas por el láser del robot son totalmente exactas. Este caso proporciona una base de partida para evaluar las capacidades fundamentales y la estabilidad del algoritmo. En segundo lugar, el nivel de ruido incluido en los sensores odométricos fue probado a varios niveles (1%, 5% y 10%) a la vez que se mantenían lecturas de láser perfectas. En este caso, el ruido en la odometría induce que la estimación inicial del robot sea errónea. Al aumentar el error inicial la búsqueda de la solución exacta se hace más difícil; de este modo, este tipo de prueba nos permite comprobar la capacidad del algoritmo para buscar el espacio de soluciones sin quedarse atascado en un mínimo local. En tercer lugar, se introduce ruido en las medidas del láser (con dos niveles: desviación estándar de 5mm y 20mm para una medida de 1m) manteniendo a cero el ruido en los sensores de odometría. En este caso, la estimación inicial de la pose corresponde a la exacta pero ahora la búsqueda de la pose del robot se va a ver influenciada por la distorsión en el mapa del entorno calculado por el robot a partir de lecturas láser ruidosas. Este tipo de pruebas permite evaluar la eficacia del algoritmo de filtrado y segmentación. Finalmente, se introducen ensayos que combinan los dos tipos de ruido. Este tipo de pruebas pretende representar casos más realistas de uso contra los que el algoritmo de *scan-matching* debe estar diseñado. La *Tabla 1* resume gráficamente la información proporcionada anteriormente.

Tabla 1: Tipos de pruebas realizadas en el simulador

Tipo de ensayo	error _{odom}	σ_{laser} (when measure is 1m)	Propósito del ensayo
Sin ruido	0%	0mm	Estimación inicial de las capacidades fundamentales y la estabilidad del algoritmo.
Ruido en odometría	1%	0mm	Evaluación de la capacidad del algoritmo para buscar el espacio de soluciones sin quedarse atascado en un mínimo local.
	5%	0mm	
	10%	0mm	
Ruido en medidas láser	0%	5mm	Evaluación de la eficacia del algoritmo de filtrado y segmentación.
	0%	20mm	
Ruido combinado	5%	5mm	Estudio realista con dos niveles (uno en un ambiente benigno y otro en un ambiente hostil).
	10%	20mm	

Para el mapa presentado en la Figura 8, la sección 6.1 proporciona los resultados obtenidos al aplicar las pruebas descritas a algoritmo general en el que la búsqueda de la orientación y de la posición se realiza de manera alterna como fue descrito en la sección 5.3.1. La sección 6.2 presenta de igual manera los resultados de aplicar los algoritmos alternativos descritos en la sección 5.3.2 en los que la búsqueda de la orientación y la posición del robot se realiza de manera simultánea. La sección 6.3 analiza los beneficios obtenidos de aplicar el algoritmo de segmentación de línea descrito en el capítulo 4. Finalmente, en la sección 6.4 se complementan los resultados anteriores probando el algoritmo en dos mapas adicionales, ver la Figura 9 y la Figura 10, con el propósito de confirmar los resultados de las secciones anteriores. Además se dota al algoritmo general de las leves mejoras y se prueba la variante del algoritmos general propuesta en la Figura 7.

Finalmente, es importante indicar que los resultados presentados a continuación representan los resultados de dejar correr el simulador durante cien ciclos de *scan-matching*.

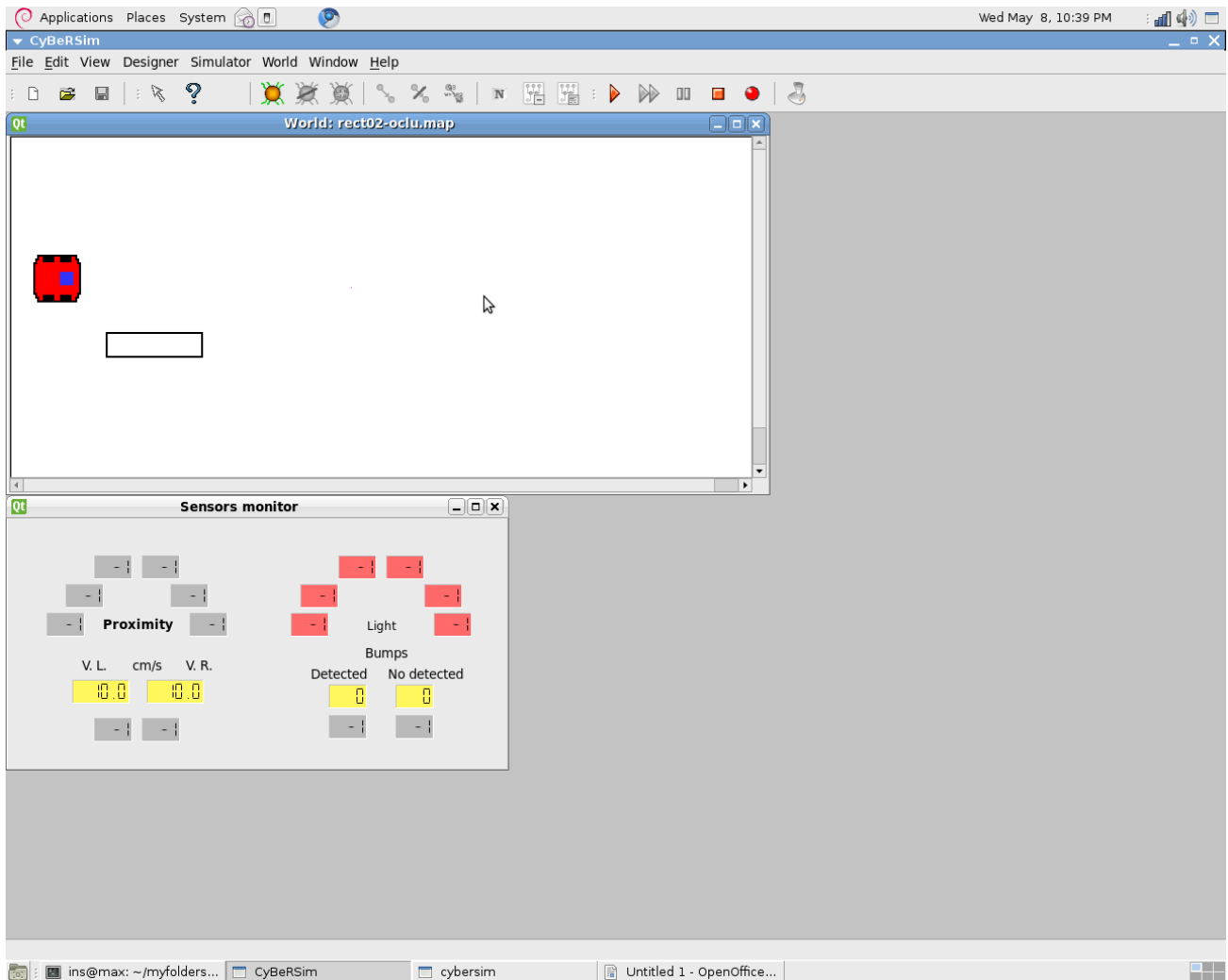


Figura 8: Mapa rectangular con una columna

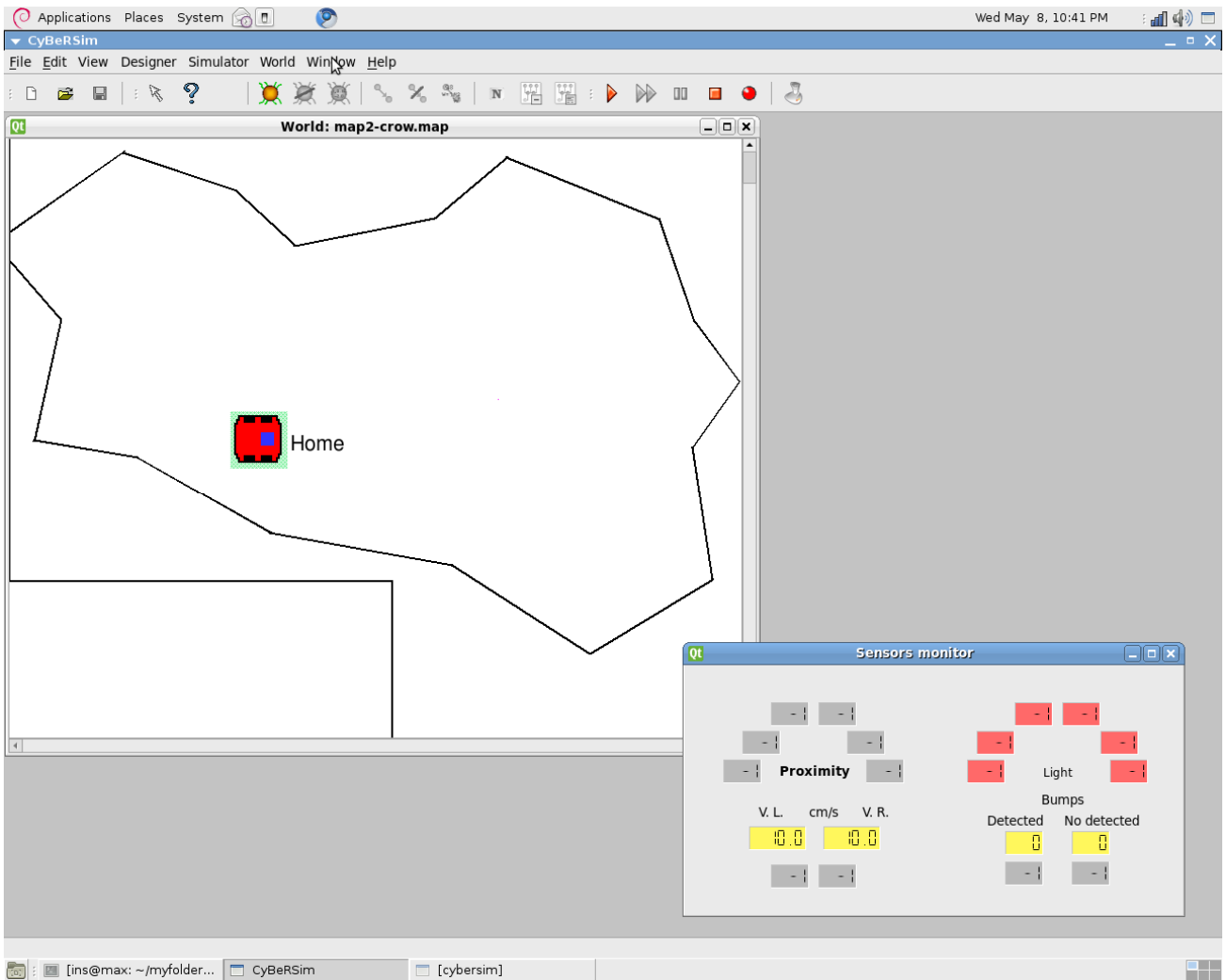


Figura 9: Mapa poligonal

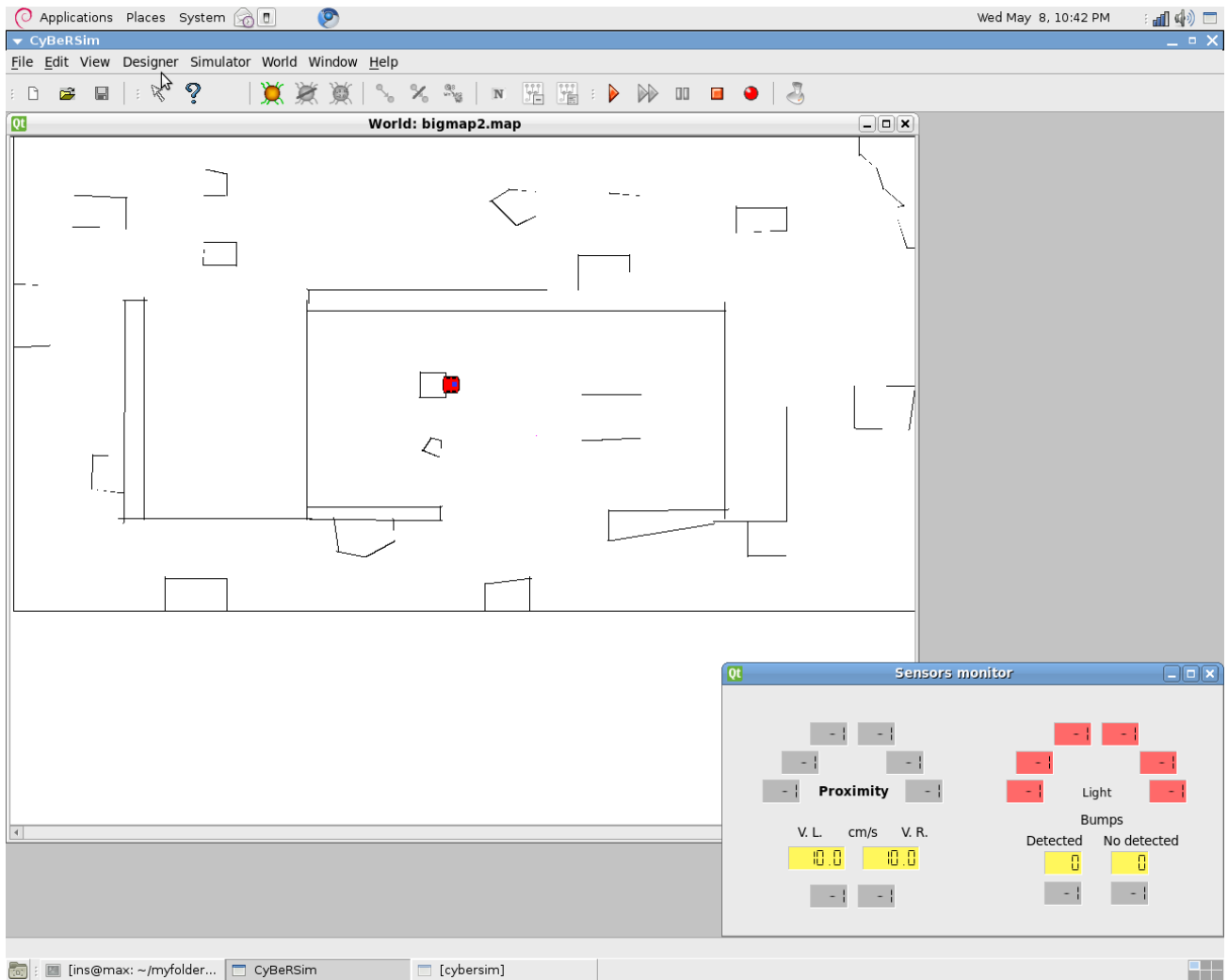


Figura 10: Mapa de gran tamaño

6.1 Búsqueda alterna de posición y orientación

Se prueban los siguientes tres algoritmos de *scan-matching* que emplean la técnica de búsqueda alterna de orientación y posición:

- El algoritmo original presentado en [Diosi07a] con la peculiaridad de que no se aplican la segmentación heurística original sino la segmentación según presentada en el capítulo 4
- El algoritmo E-PSM propuesto en la sección 5.3.1 (Figura 6)
- Una aproximación de fuerza bruta. En este caso, la búsqueda de la orientación es idéntica a la propuesta en el E-PSM. Sin embargo, la búsqueda de la posición se realiza barriendo completamente cierta área en torno al robot (800mm por 800mm en intervalo de 10mm) que corresponde a los cambios posibles de la posición del robot en un intervalo de tiempo dado. Esta aproximación se incluye como una ayuda para comparar la precisión de los resultados proporcionados por los otros algoritmos pero no como una solución práctica.

Antes de discutir los resultados resulta útil repasar como se han tratado los datos obtenidos del simulador *cybersim*. En general, todos los datos del simulador han sido procesados con la herramienta MATLAB por su capacidad de tratar un gran número de datos de una manera rápida y simple. Así se han generado las gráficas que se presentan a continuación. Para el mapa rectangular y en el caso de tener un 10% de ruido en los sensores de odometría, las Figura 11, Figura 12, y Figura 13 muestran la evolución en el tiempo de las coordenadas x, y además de su orientación. La línea negra muestra el valor exacto, la línea roja la solución proporcionada por el algoritmo E-PSM, y la línea azul discontinua el valor obtenido a partir de los datos odométricos únicamente. Se puede observar como la localización ofrecida por la solución odométrica se desvía rápidamente de la solución exacta y cómo la solución calculada por el algoritmo E-PSM consigue rectificar buena parte del error introducido por el ruido en los sensores odométricos. Dada la dificultad para interpretar la Figura 13, en la Figura 14 se proporciona el desarrollo del error acumulado en la estimación de la orientación del error (de nuevo, línea roja da la solución proporcionada por el algoritmo E-PSM, y la línea azul discontinua la solución basada sólo en odometría). Se puede observar que el error en la orientación se acumula en puntos concretos (que corresponden a giros del robot sobre sí mismo) y que la solución propuesta elimina con éxito la mayor parte del ruido introducido. A modo de resumen, la Figura 15 proporciona una vista en el plano XY de la trayectoria seguida por el robot (línea negra) y la estimación de la trayectoria obtenida del E-PSM (línea roja) siempre en el caso de tener un 10% de error en los sensores de odometría.

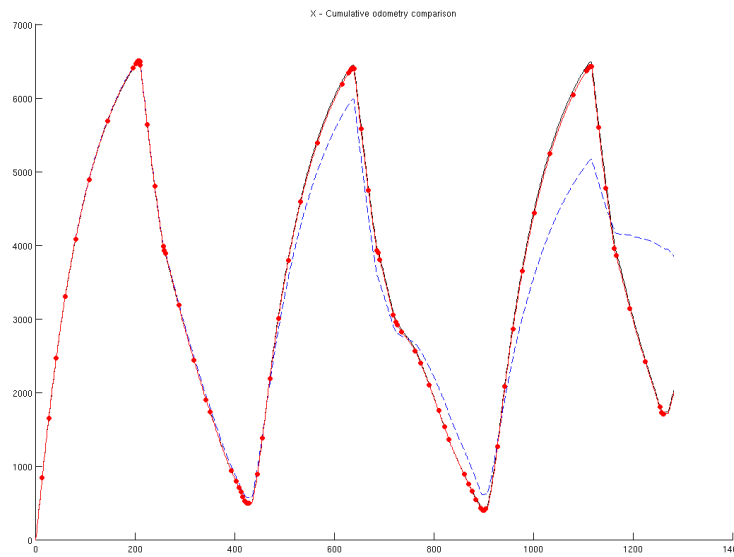


Figura 11: Posición absoluta del robot (x-coord) en función del tiempo.

Para el mapa rectangular y con un $error_{odom}$ de 10%, esta figura muestra la posición absoluta del robot (en el eje de abscisas) computada por el algoritmo E-PSM (línea roja), la posición exacta (línea negra), y la posición computada a partir de datos odométricos únicamente (línea azul discontinua). Los puntos rojos muestran el momento en el que se realiza el proceso de *scan-matching* con el fin de corregir la desviación introducida en la computación de la posición por el ruido presente en la odometría. Se puede observar como la localización ofrecida por la solución odométrica se desvía rápidamente de la solución exacta y cómo la solución calculada por el algoritmo E-PSM consigue rectificar buena parte del error introducido por el ruido en los sensores odométricos.

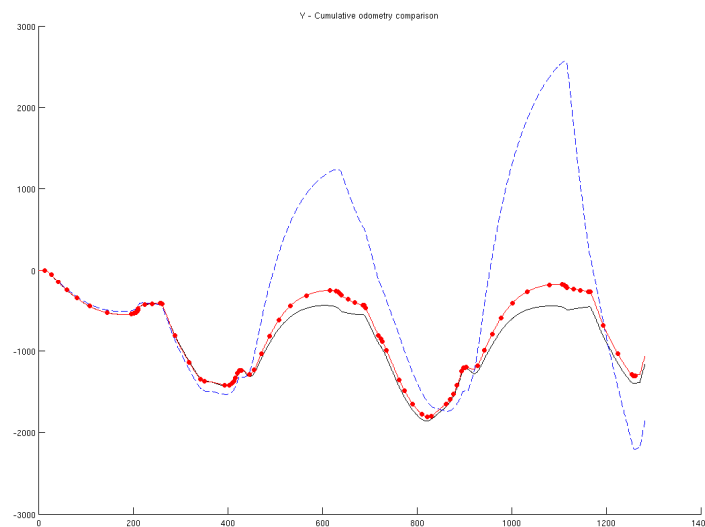


Figura 12: Posición absoluta del robot (y-coord) en función del tiempo

Para el mismo caso de la figura anterior, se presenta las diferentes posiciones computadas del robot en el eje de ordenadas. Se observa un error más pronunciado en la posición computada por odometría debido a los errores en la computación de la orientación del robot.

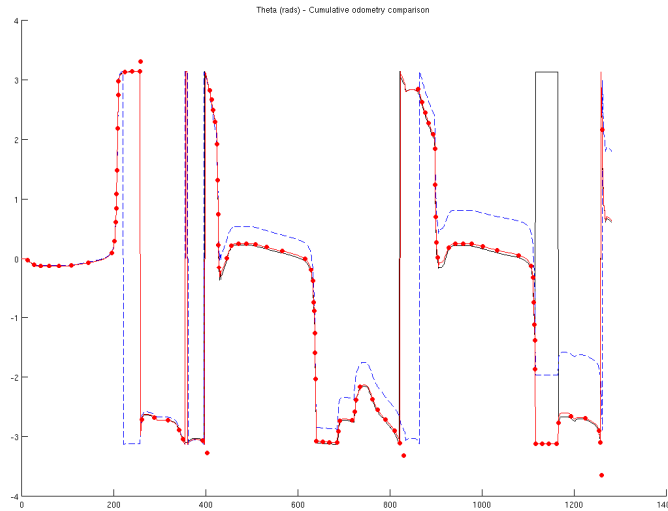


Figura 13: Orientación del robot (en rad) en función del tiempo

Para el mapa rectangular y con un error_{odom} de 10%, esta figura muestra la orientación robot (en radianes) computada por el algoritmo E-PSM (línea roja), la orientación exacta (línea negra), y la orientación computada a partir de datos odométricos únicamente (línea azul discontinua). Los puntos rojos muestran el momento en el que se realiza el proceso de *scan-matching*.

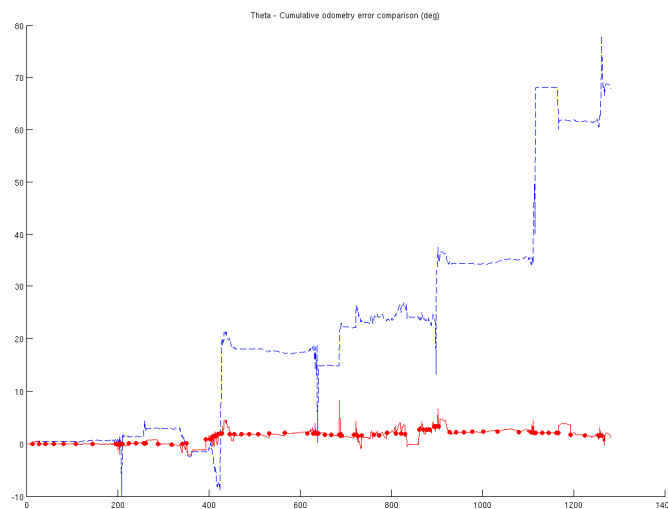


Figura 14: Error acumulado en la estimación de la orientación del robot (grados)

Para el mismo caso de la figura anterior, esta figura presenta el error (en grados) que se acumula en la estimación de la orientación del robot según éste se desplaza. Se observa que el error en la estimación de la odometría (línea azul discontinua) sufre de grandes saltos que coinciden con los momentos en los que el robot de grandes giros sobre sí mismo a gran velocidad. Por otro lado, el error de la estimación del algoritmo E-PSM permanece en un valor pequeño durante toda la simulación.

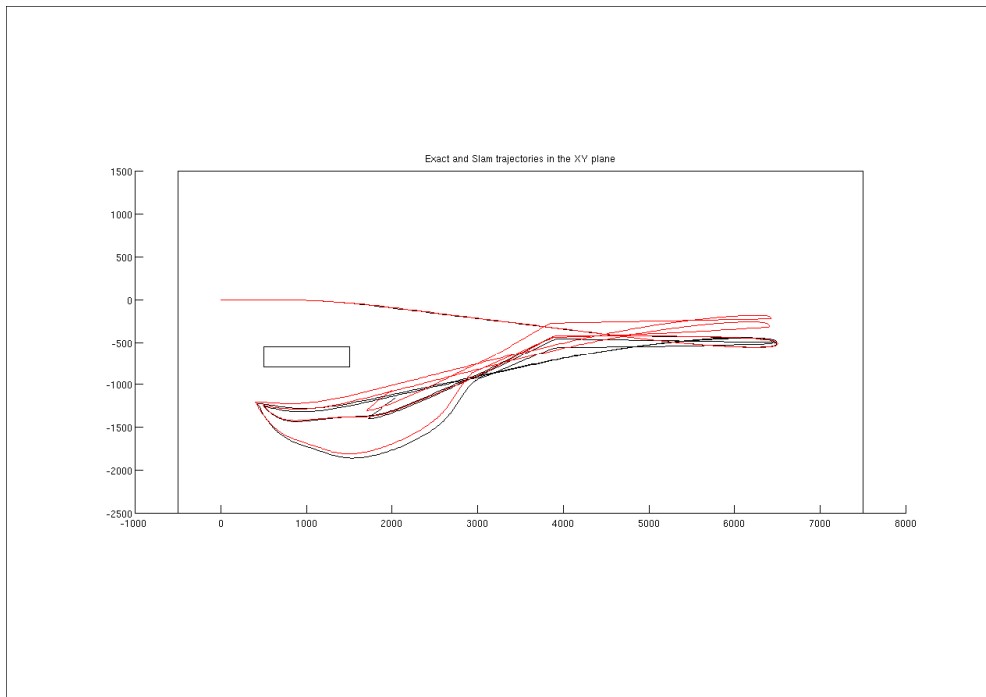


Figura 15: Trayectoria del robot en el plano XY

Para el mapa rectangular y con un $\text{error}_{\text{odom}}$ de 10%, esta figura muestra la trayectoria exacta (línea negra) y la trayectoria calculada por el algoritmo E-PSM (línea roja). La trayectoria computada únicamente por odometría no se incluye por salirse fuera de la escala rápidamente. Para referencia, se incluye en negro el perfil del mapa rectangular de la Figura 8. Se observa cómo, a pesar del alto nivel de ruido, el algoritmo E-PSM consigue trazar con fidelidad la trayectoria del robot.

Para profundizar en el estudio del comportamiento de la solución propuesta es necesario analizar que ocurre en cada *scan-matching* realizado por el algoritmo E-PSM. En la Figura 16, Figura 17, y Figura 18 se presentan la evolución de los errores cometidos en el cálculo del cambio de la pose del robot (a_x , a_y y a_{theta}) en las coordenadas de referencia para cada segmento recorrido. De manera similar, la Figura 19 proporciona el valor de la función objetivo obtenido al final del proceso de búsqueda de la pose del robot. Recuérdese que el objetivo del proceso es minimizar dicha función y que un valor de cero (o un valor ligeramente positivo) representa un alineamiento perfecto entre el *current scan* y el *reference scan*. Se asume que la pose correspondiente a este alineamiento proporciona la mejor estimación posible del cambio en la pose del robot obtenible por el método del *scan-matching*. Usando estas gráficas de referencia se procede a presentar en detalle que ocurre en dos casos particulares (paso número 2 como ejemplo de proceso de *scan-matching* con error muy bajo y paso número 78 como ejemplo de proceso de *scan-matching* con error medio –en relación con lo obtenido en otros casos-).

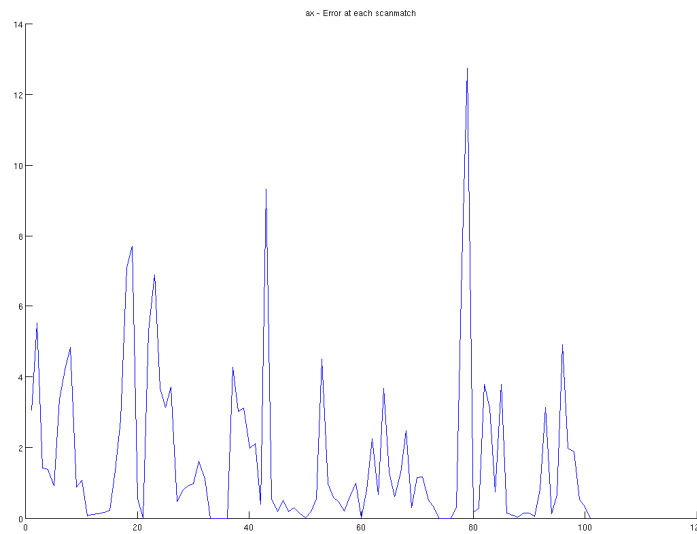


Figura 16: Error en a_x en cada *scan-matching* realizado

Para el mapa rectangular y con un $\text{error}_{\text{odom}}$ de 10%, esta figura muestra el error incurrido en la estimación cambio de la posición del robot a lo largo del eje x_{ref} cada vez que se usa el *scan-matching*. Así, cada punto de esta gráfica corresponde a un punto rojo marcado en la Figura 11. Se observa un error máximo de 13mm y un error medio de 2.8mm aproximadamente.

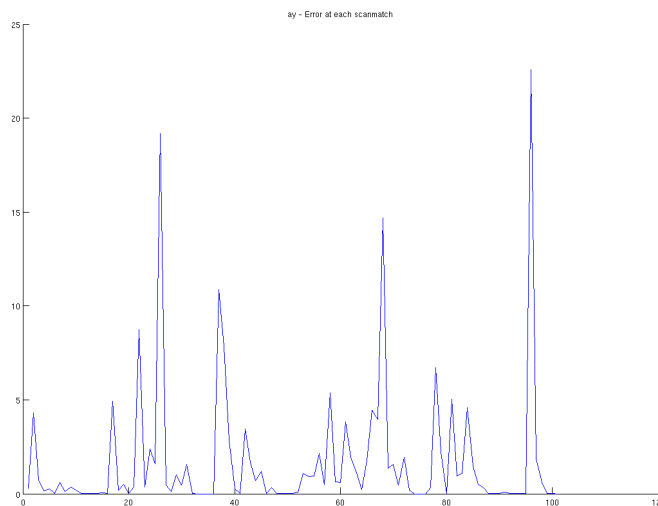


Figura 17: Error en a_y en cada *scan-matching* realizado

Para el mapa rectangular y con un $\text{error}_{\text{odom}}$ de 10%, esta figura muestra el error incurrido en la estimación cambio de la posición del robot a lo largo del eje y_{ref} cada vez que se usa el *scan-matching*. Así, cada punto de esta gráfica corresponde a un punto rojo marcado en la Figura 12. Se observa un error máximo de 23mm y un error medio de 3.7mm

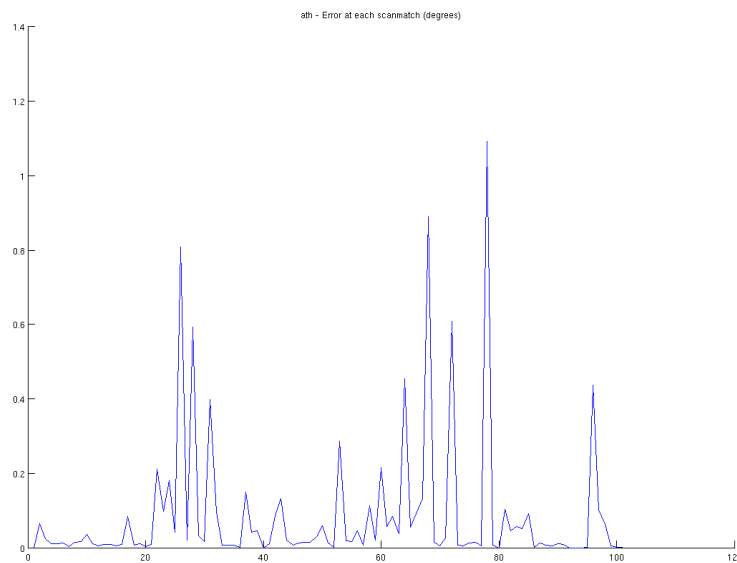


Figura 18: Error en a_{θ} en cada *scan-matching* realizado (grados)

Para el mapa rectangular y con un $error_{odom}$ de 10%, esta figura muestra el error incurrido en la estimación cambio de la orientación del robot cada vez que se usa el *scan-matching*. Así, cada punto de esta gráfica corresponde a un punto rojo marcado en la Figura 13. Se observa un error máximo de 1.1 grados y un error medio de 0.1 grados aproximadamente.

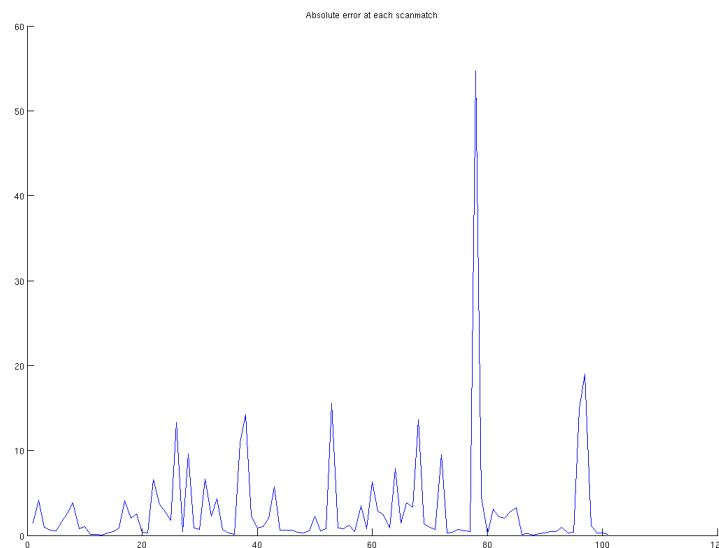


Figura 19: Valor de la función objetivo f (error en comparación) en cada *scan-matching* realizado

Para el mapa rectangular y con un $error_{odom}$ de 10% y para cada *scan-matching* realizado, esta figura muestra el error incurrido en el alineamiento de los escáneres de referencia y actual según se define en la Ecuación 4 (definición de la función objetivo). Es importante notar cómo el error máximo en f coincide con los errores máximos en las estimaciones de la posición u orientación. Así, para el paso 78, se observa el valor de f máximo (unos 54) que coincide para el mismo paso con errores máximos en los valores de a_x y a_{θ} según se mostró en la Figura 16 y la Figura 18 respectivamente.

Ejemplo de proceso de alineamiento: Paso No. 2

La Figura 20 representa el recorrido del robot de izquierda a derecha (la posición del robot se representa por un punto verde) y los escáneres obtenidos en cada paso (marcados como líneas azules) y las paredes del entorno (líneas verdes discontinuas). Se puede apreciar la precisión del escáner obtenido ya que en este caso se ha ajustado la desviación estándar del error en los sensores láser a 0mm. La Figura 21 proporciona el equivalente en coordenadas polares de la información de los escáneres. Nótese que el objetivo básico del proceso de scan-matching es encontrar la pose que nos permita alinear el *reference scan* y *actual scan* aquí representados en coordenadas polares. La Figura 22 y la Figura 23 proporcionan los resultados del proceso de segmentación aplicado a las medidas láser; se representa cada segmento en un color diferente en el plano XY (Figura 22) y en coordenadas polares (Figura 23).

El resultado del proceso de scan-matching se ve representado en la Figura 24; aquí se muestra el alineamiento obtenido entre el *current scan* y el *actual scan* una vez se ha hallado el valor óptimo de la pose del robot. Es importante resaltar la gran precisión que se consigue en este proceso de alineamiento. Sólo los puntos representados por cruces azules han sido definidos como válidos para el proceso de scan-matching. Aquellos puntos marcados con un círculo verde corresponden a los datos descartados por: (1) No tener un punto de correspondencia en el *current scan*, o (2) por haber sido identificado como no-válido por el algoritmo de segmentación de líneas. Finalmente, los puntos marcados con un triángulo invertido de color cian corresponden a aquellos descartados por el nuevo criterio de oclusión definido en el capítulo 5.2.

La Figura 25 y Figura 26 proporcionan una representación gráfica de las búsquedas de la orientación y posición respectivamente según son realizadas por el algoritmo E-PSM. En la Figura 25 se puede observar las variaciones en la función objetivo en función de θ_{robot} para una misma posición dada por (x_{robot}, y_{robot}) : la gráfica izquierda muestra la búsqueda inicial en intervalos de un grado y la gráfica de la derecha la búsqueda refinada realizada en intervalos de 0.01 grados. La línea cian representa la aproximación parabólica propuesta de Diosi que, en este caso, proporciona buenos resultados. La Figura 26 muestra los contornos de la función objetivo en el plano XY (para un mismo θ_{robot}) en el área alrededor de la posición estimada a partir de datos odométricos. La solución obtenida y la posición exacta se representan con un cuadrado verde y rojo respectivamente. En este caso se observa que la función presenta contornos muy regulares y que la solución obtenida por el algoritmo E-PSM se superpone a la posición exacta.

Finalmente, en la Figura 27 se presenta la evolución de la función objetivo con el paso de las iteraciones del algoritmo E-PSM. Se observa que se obtiene el valor óptimo muy rápidamente y que la solución se estabiliza sin ningún problema lo que permite, en este caso, que el algoritmo se ejecute sólo el número mínimo de veces definido.

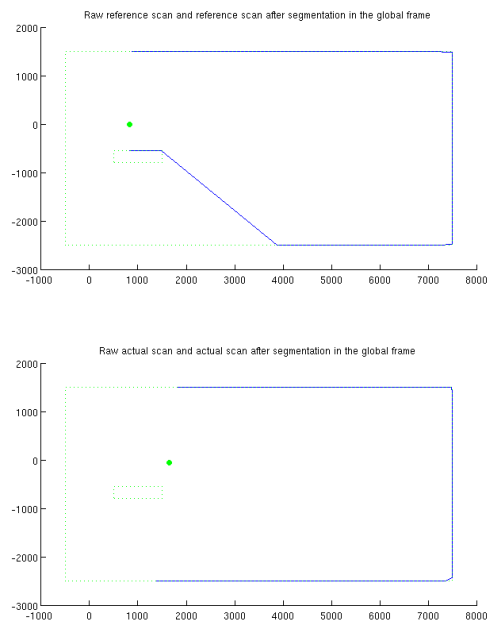


Figura 20: Escáneres de referencia y actual en el plano XY (No. Paso = 2)

Para el mapa rectangular y con un error_{odom} de 10%, esta figura representa el recorrido del robot de izquierda a derecha (la posición del robot se representa por un punto verde) y los escáneres obtenidos en cada paso (marcados como líneas azules) y las paredes del entorno (líneas verdes discontinuas). Concretamente, se presenta las posiciones del robot entre la segunda y tercera vez que se utiliza el invoca el proceso de *scan-matching* (es decir, el segundo paso). Nótese también que, en este caso, el escáner obtenido es preciso pues se impuso un σ_{laser} de 0mm.

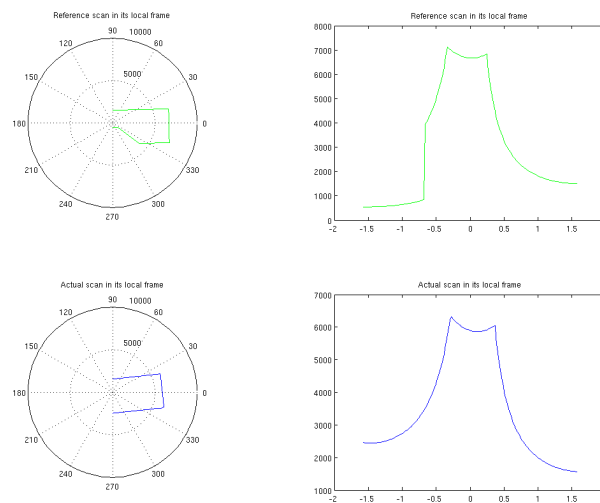


Figura 21: Escáneres de referencia y actual en coordenadas polares (No. Paso = 2)

Esta figura proporciona la representación en coordenadas polares de los escáneres mostrados en la figura anterior. En este caso, la línea verde representa el escáner de referencia y la línea azul el escáner actual.

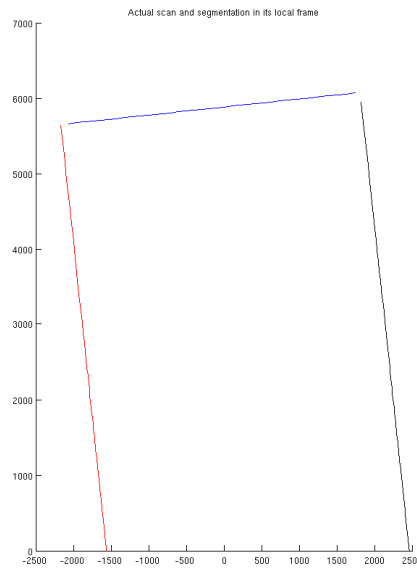


Figura 22: Segmentación del escáner actual (No. Paso = 2)

Esta figura representa el proceso de segmentación aplicada a los escáneres obtenidos anteriormente. En este caso, se observan en el plano XY tres segmentos diferentes (presentados con diferentes colores asignados aleatoriamente) que corresponden con exactitud a las tres paredes que se encuentran en frente del robot para este paso.

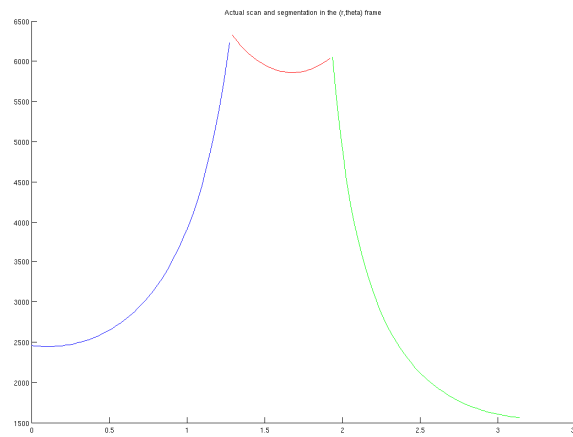


Figura 23: Segmentación del escáner actual en coordenadas polares (No. Paso = 2)

Esta figura presenta la versión polar de los tres segmentos identificados en la figura anterior.

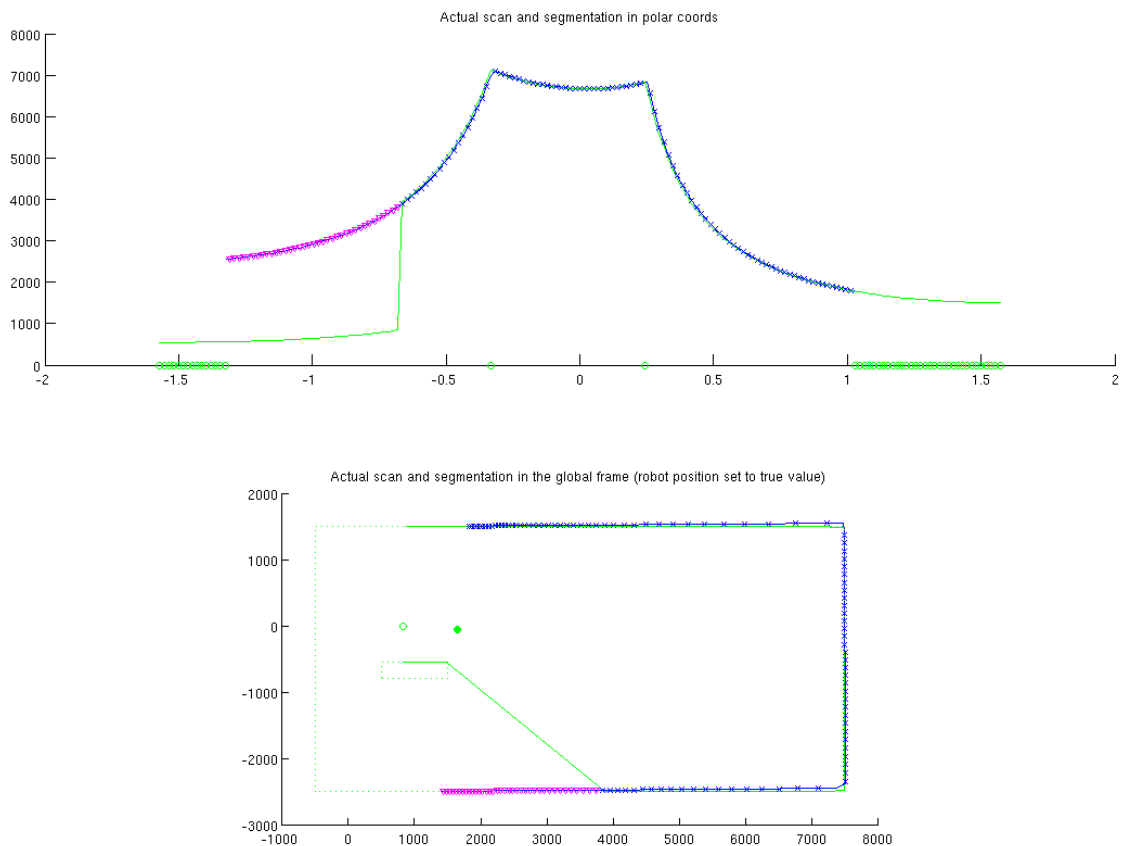


Figura 24: Alineamiento de los escáneres de referencia y actual (No. Paso = 2)

Esta figura muestra, para el segundo paso, el alineamiento obtenido entre el *current scan* y el *actual scan* una vez se ha hallado el valor óptimo de la pose del robot. Los puntos representados por cruces azules corresponden a los definidos como válidos para el proceso de *scan-matching*. Aquellos puntos marcados con un círculo verde corresponden a los datos descartados por: (1) No tener un punto de correspondencia en el *current scan*, o (2) por haber sido identificados como no-válidos por el algoritmo de segmentación de líneas. Los puntos marcados con un triángulo invertido de color cian corresponden a aquellos descartados por el nuevo criterio de oclusión definido en el capítulo 5.2. Se puede observar que el alineamiento de los puntos válidos es casi perfecto; esto corresponde a un valor de la función objetivo menor a 1.0 y a errores de posición y orientación del orden de 1mm y 0.1 grados respectivamente.

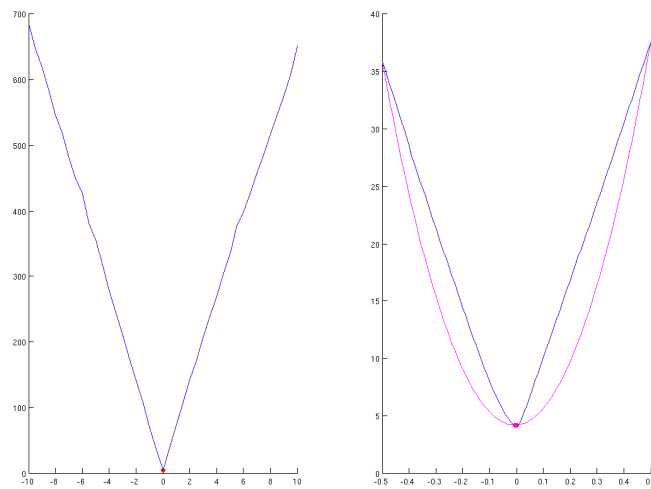


Figura 25: Búsqueda de la orientación (No. Paso = 2)

En el contexto de la búsqueda de la orientación durante el segundo paso del *scan-matching*, esta figura presenta las variaciones en la función objetivo en función de θ_{robot} para una misma posición. La gráfica izquierda muestra la búsqueda inicial en intervalos de un grado y la gráfica de la derecha la búsqueda refinada realizada en intervalos de 0.01 grados. La línea cian representa la aproximación parabólica propuesta de Diosi. Se puede observar que ambas gráficas muestran un buen comportamiento con mínimos globales claramente localizados y la ausencia de mínimos locales. Además se observa que es este caso la aproximación parabólica de Diosi coincide exactamente con el mínimo local.

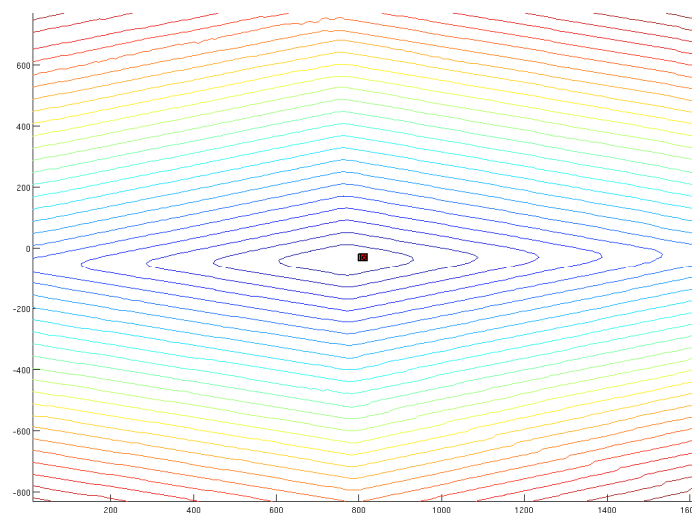


Figura 26: Contorno de la función objetivo en el plano XY (No. Paso = 2)

En el contexto de la búsqueda de la posición durante el segundo paso del *scan-matching*, esta figura muestra los contornos de la función objetivo en el plano XY (para un mismo θ_{robot}) en el área alrededor de la posición estimada a partir de datos odométricos. La solución obtenida y la posición exacta se representan con un cuadrado verde y rojo respectivamente. En este caso se observa que la función presenta contornos muy regulares y que la solución obtenida por el algoritmo E-PSM se superpone a la posición exacta.

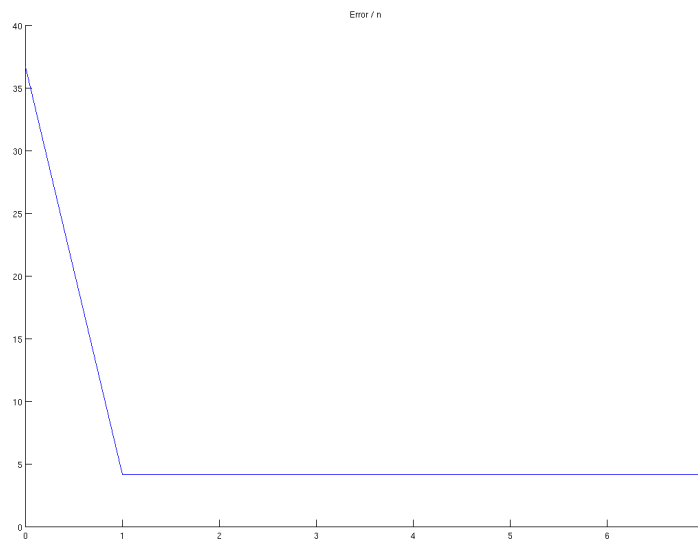


Figura 27: Evolución del valor de la función objetivo en función del número -de iteraciones (No. Paso = 2)

En el contexto de la búsqueda de la orientación y posición durante el segundo paso del *scan-matching*, esta figura presenta la evolución de la función objetivo con el paso de las iteraciones del algoritmo E-PSM. Partiendo de un valor de f muy alto (cerca de 37), se consigue rápidamente reducir el error a valores inferiores a 5 y mantenerlo en esos niveles constantes. Esto permite que sólo sea necesario ejecutar el algoritmo el número o mínimo de veces definido. Nótese también que las iteraciones pares (0, 2, 4, 6) corresponden a las búsquedas de orientación y las pares (1,3, 5, 7) a las búsquedas de la posición.

Ejemplo de proceso de alineamiento: Paso No. 78

En las figuras siguientes se presentan resultados análogos a los anteriores para un caso en el que error obtenido es más alto. La Figura 28, Figura 29, Figura 30, y Figura 31 muestran el mismo proceso de movimiento y recogida y segmentación de los escáneres. Es importante observar el círculo rojo marcado en la gráfica superior de la Figura 32; un punto de *actual scan* que no es apto para su comparación no es descartado lo que introduce una cierta cantidad de error en el proceso que no puede ser eliminada. Hay que recordar que el criterio para descartar puntos se basa en determinar si la lectura de referencia está fuera de la visión del robot desde la posición actual. En la gráfica inferior de la Figura 32 se ha insertado también un círculo negro para indicar como cualquier error en la estimación de la posición del robot puede hacer que dejen de descartar los puntos que están justo en la frontera de decisión. De todos modos, es necesario destacar que esto representa un problema menor. La Figura 32 nos muestra que el proceso se *scan-matching* consigue llegar a un alineamiento casi perfecto a nivel global. Más tarde también se verá que los problemas más graves surgen al introducir ruido en las medidas de los sensores láser. La Figura 33 y Figura 34 muestran el proceso de búsqueda de la orientación y posición respectivamente. Se observa la gran abundancia de mínimos locales que advierte de los posibles errores en que se puede incurrir si se usan métodos lineales o puramente locales.

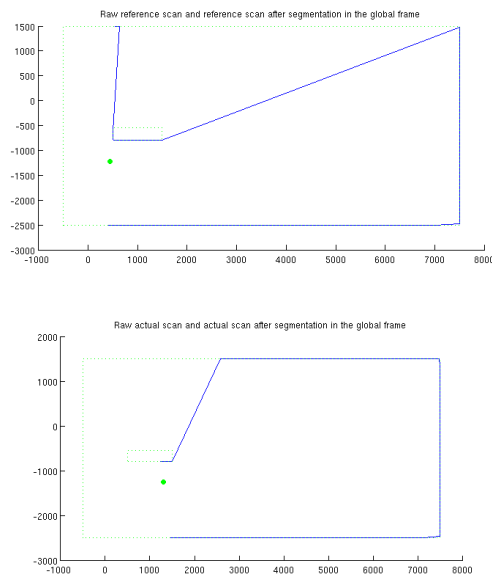


Figura 28: Escáneres de referencia y actual en el plano XY (No. Paso = 78)

Para el mapa rectangular y con un error_{odom} de 10%, esta figura representa el recorrido del robot de izquierda a derecha (la posición del robot se representa por un punto verde) y los escáneres obtenidos en cada paso (marcados como líneas azules) y las paredes del entorno (líneas verdes discontinuas). Concretamente, se presenta las posiciones del robot entre la 78ª y 79ª vez que se utiliza el invoca el proceso de *scan-matching*. Nótese también que, en este caso, el escáner obtenido es preciso pues se impuso un σ_{laser} de 0mm.

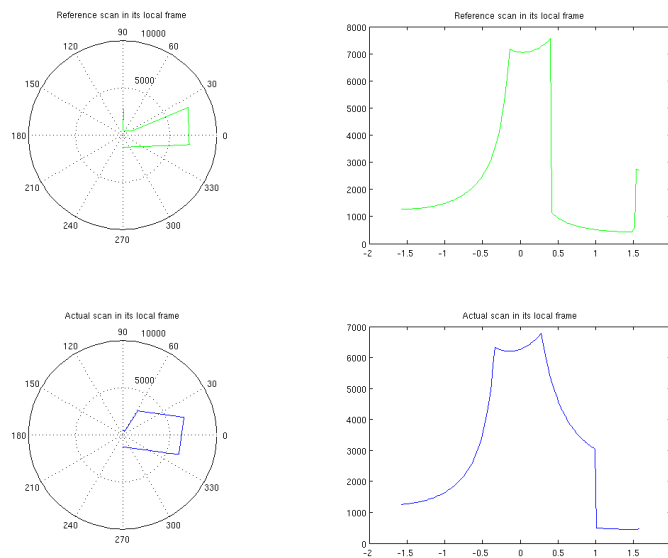


Figura 29: Escáneres de referencia y actual en coordenadas polares (No. Paso = 78)

Esta figura proporciona la representación en coordenadas polares de los escáneres mostrados en la figura anterior. En este caso, la línea verde representa el escáner de referencia y la línea azul el escáner actual.

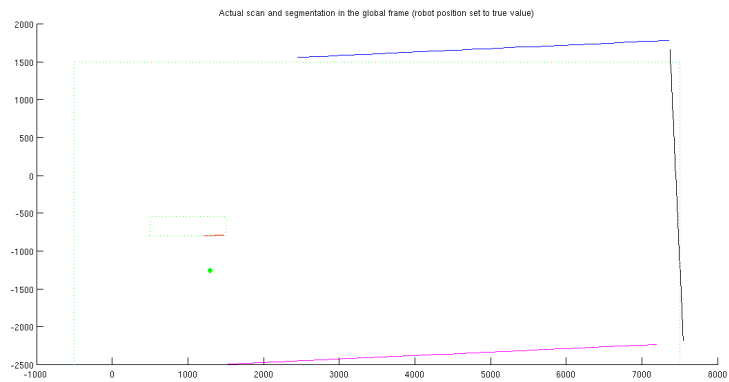


Figura 30: Segmentación del escáner actual (No. Paso = 78)

Esta figura representa el proceso de segmentación aplicado a los escáneres obtenidos anteriormente. En este caso, se observan en el plano XY cuatro segmentos diferentes que corresponden con exactitud a las tres paredes y a parte de la columna que se encuentran en frente del robot para este paso. Se puede observar cierta desviación en la orientación de los segmentos representados con respecto a las paredes representadas por líneas verdes discontinuas. Esto es debido a que la pose de robot que se usa como origen para la representación incluye los errores acumulados en la estimación de la misma y no presupone ningún error en el proceso de filtrado y segmentación de las medidas láser.

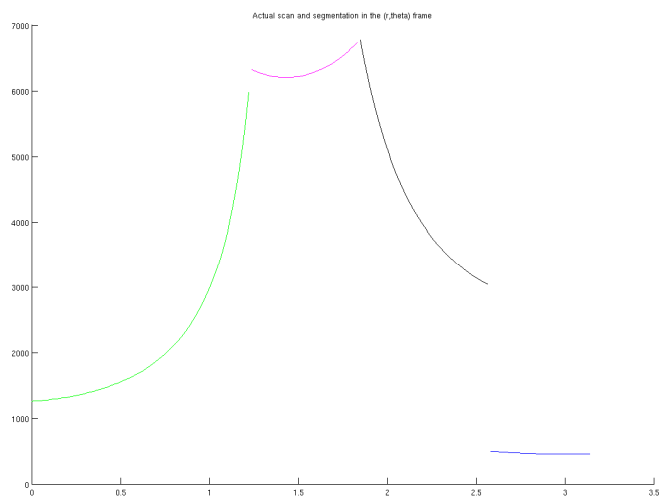


Figura 31: Segmentación del escáner actual en coordenadas polares (No. Paso = 78)

Esta figura presenta la versión polar de los segmentos identificados en la figura anterior.

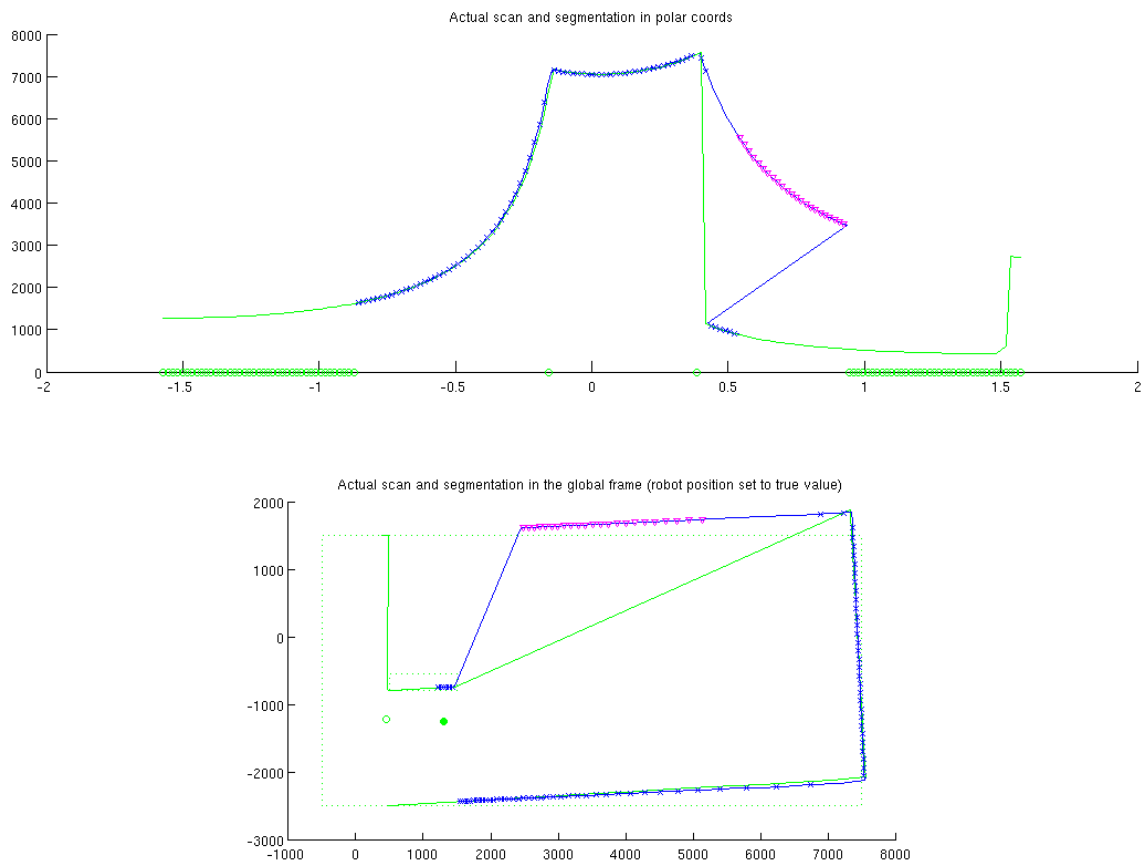


Figura 32: Alineamiento de los escáneres de referencia y actual (No. Paso = 78)

Esta figura muestra, para el 78º paso, el alineamiento obtenido entre el *current scan* y el *actual scan* una vez se ha hallado el valor óptimo de la pose del robot. Los puntos representados por cruces azules corresponden a los definidos como válidos para el proceso de *scan-matching*. Aquellos puntos marcados con un círculo verde corresponden a los datos descartados por: (1) No tener un punto de correspondencia en el *current scan*, o (2) por haber sido identificados como no-válidos por el algoritmo de segmentación de líneas. Los puntos marcados con un triángulo invertido de color cian corresponden a aquellos descartados por el nuevo criterio de oclusión definido en el capítulo 5.2. Es importante observar el círculo rojo marcado en la gráfica superior: un punto del *actual scan* que no es apto para su comparación no es descartado lo que introduce una cierta cantidad de error en el proceso que no puede ser eliminada. Hay que recordar que el criterio para descartar puntos se basa en determinar si la lectura de referencia está fuera de la visión del robot desde la posición actual. En la gráfica inferior se ha insertado también un círculo negro para indicar como cualquier error en la estimación de la posición del robot puede hacer que dejen de descartar los puntos que están justo en la frontera de decisión. De todos modos, es necesario destacar que esto representa un problema menor. Esta figura nos muestra que el proceso de *scan-matching* consigue llegar a un alineamiento casi perfecto a nivel global.

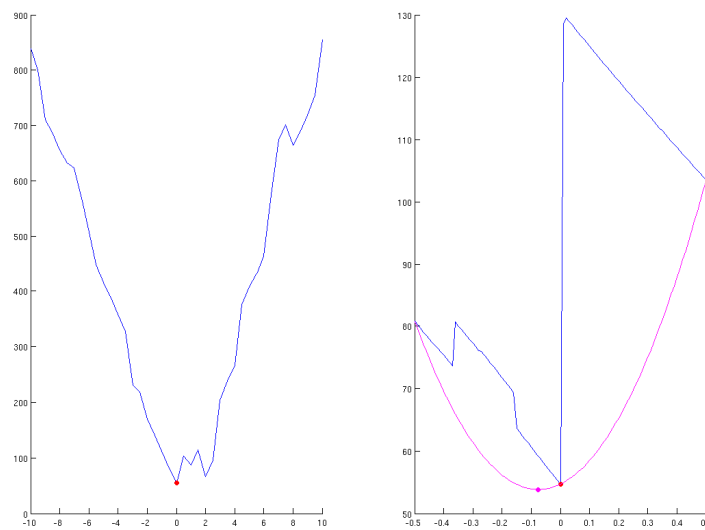


Figura 33: Búsqueda de la orientación (No. Paso = 78)

En el contexto de la búsqueda de la orientación durante el 78º paso del *scan-matching*, esta figura presenta las variaciones en la función objetivo en función de θ_{robot} para una misma posición. La gráfica izquierda muestra la búsqueda inicial en intervalos de un grado y la gráfica de la derecha la búsqueda refinada realizada en intervalos de 0.01 grados. La línea cian representa la aproximación parabólica propuesta de Diosi. Se puede observar que ambas gráficas muestran un mal comportamiento con múltiples mínimos locales. Además se observa que es este caso la aproximación parabólica de Diosi resulta en un error entorno a los 0.1 grados.

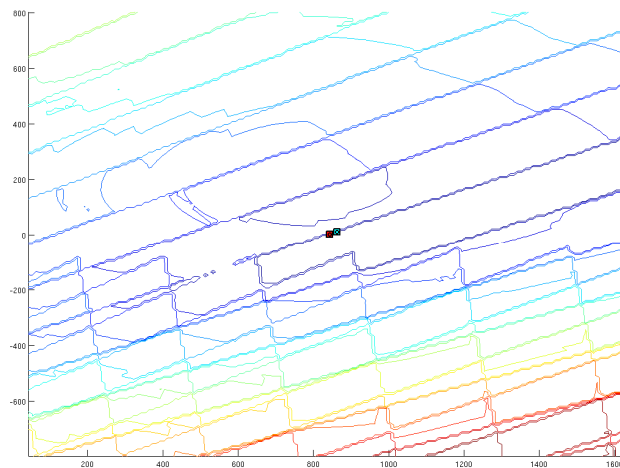


Figura 34: Contorno de la función objetivo en el plano XY (No. Paso = 78)

En el contexto de la búsqueda de la posición durante el 78º paso del *scan-matching*, esta figura muestra los contornos de la función objetivo en el plano XY (para un mismo θ_{robot}) en el área alrededor de la posición estimada a partir de datos odométricos. La solución obtenida y la posición exacta se representan con un cuadrado verde y rojo respectivamente. En este caso se observa que la función presenta contornos muy irregulares aunque aún así la solución obtenida por el algoritmo E-PSM resulte muy cercana a la posición exacta.

Para cada uno de los casos de prueba discutidos en la Tabla 1, se proporcionan en la Tabla 2 los resultados (valores de media y máximos) obtenidos para las siguientes cinco variables:

- δx_{robot} (también designado a_x): error en mm de la posición calculada del robot a lo largo del eje x_{ref}
- δy_{robot} (también designado a_y): error en mm de la posición calculada del robot a lo largo del eje y_{ref}
- $\delta \theta_{robot}$ (también designado a_{theta}): error en grados de la orientación calculada del robot
- f : función objetivo según fue definida en la Ecuación 4
- *Time*: tiempo computacional usado por la función de *scan-matching* en cada segmento recorrido por el robot (medido en segundos en un ordenador equipado de un microprocesador Intel Core2 Duo)

Tomando como referencia los valores de $\delta \theta_{robot}$ se observa claramente que la aplicación del algoritmo original de Diosi resulta en errores muy altos en la estimación de la orientación del robot (errores medios de entre 0.18 y 0.67 grados y errores máximos de entre 1.3 y 10 grados). Tendencias similares se pueden contrastar para δx_{robot} y δy_{robot} . El algoritmo E-PSM propuesto en este trabajo así como la aproximación de fuerza bruta obtienen mucha mejor precisión (errores medios de entre 0.03 grados y 0.28 grados y errores máximos de entre 0.32 y 3.2 grados). Los recursos computacionales requeridos muestran una gran diferencia entre las tres aproximaciones probadas. El algoritmo de Diosi requiere de media unos 0.003 segundos para obtener la estimación del cambio de la pose del robot. El algoritmo E-PSM requiere de media 0.07 segundos. Finalmente, la aproximación de fuerza bruta requiere de no menos de diez segundos de computación para obtener la estimación.

Estos resultados muestran preliminarmente que el algoritmo E-PSM representa una excelente alternativa. Con E-PSM se obtienen resultados con una precisión sólo ligeramente inferior a la de la aproximación por fuerza bruta pero con requerimientos computacionales tres órdenes de magnitud inferiores.

Antes de continuar con el refinamiento de estos resultados, resulta útil revisar los resultados obtenidos con el proceso de *scan-matching* con ruido presente en los sensores láser. Con un error de 20mm (desviación estándar de las medidas láser), la Figura 35 muestra las medidas obtenidas (paso 30) y cómo el ruido lleva al algoritmo de filtrado y segmentación a cometer ligeros errores en la identificación de los segmentos presentes en el mapa. La Figura 36, Figura 37, Figura 38 y Figura 39 muestran el subsiguiente desarrollo del proceso de proyección y alineamiento. Aunque el proceso de alineamiento sea exitoso, esta última figura da un claro ejemplo de la dificultad del proceso de *scan-matching* en presencia de ruido y la inevitabilidad de introducir pequeños errores en la estimación de la pose del robot.

Tabla 2: Resultados de búsqueda alterna de orientación y posición (valores medio y máximo)

		No noise	Odometry 1	Odometry 5	Odometry 10	Scan 5	Scan 20	Scan 5 Odo 5%	Scan 20 Odo 10%
δx_{robot}	Diosi	3.0937	3.3203	7.3799	8.6384	10.5081	33.162	10.3399	29.2531
		33.9534	35.253	132.2094	231.8395	269.3919	337.9772	169.6942	262.6518
	E-PSM	0.5063	0.1967	0.7072	2.8446	2.9504	15.6797	3.0183	16.1739
		15.9846	2.4038	30.6976	13.0562	14.6592	122.8687	16.2117	128.6824
	F. bruta	0.8923	0.8152	1.5725	2.1447	2.421	14.0421	2.8316	15.0018
		10.2146	9.9943	10.7619	33.7543	10.9659	120.0791	21.6522	110
δy_{robot}	Diosi	4.5134	4.697	5.1987	5.1261	7.3843	20.3421	6.7216	18.1529
		40.1491	40.5981	79.5054	89.488	134.6917	324.5208	53.6533	205.7982
	E-PSM	0.4439	0.2715	0.8568	3.7107	2.0399	8.4649	2.5053	11.7983
		24.3551	7.0441	8.9925	23.63	16.2678	96.3736	27.748	181.6468
	F. bruta	0.7975	0.6143	1.2624	1.4261	0.9214	5.4918	1.8133	6.3702
		9.8902	10.2576	19.555	9.8882	10.2198	69.923	29.8773	40
$\delta \theta_{robot}$	Diosi	0.1766	0.2218	0.2808	0.3774	0.4066	0.6699	0.3891	0.602
		1.3279	2.2729	3.6457	6.3483	7.7022	10.1063	5.0007	7.703
	E-PSM	0.0342	0.0268	0.0413	0.0906	0.0693	0.2645	0.0943	0.2855
		0.58	0.5865	0.4525	1.1215	0.68	1.12	1.3613	3.1994
	F. bruta	0.0259	0.0253	0.0331	0.0623	0.0681	0.219	0.0828	0.2109
		0.32	0.4022	0.3244	0.8785	0.59	1.17	0.5744	1.49
f	Diosi	8.4208	9.9313	14.3579	19.1583	20.8199	45.0015	19.365	41.8228
		106.373	126.2762	241.1459	593.0042	391.501	476.5114	321.8906	500.6415
	E-PSM	2.0844	1.3389	1.3907	3.7652	6.749	23.3636	6.0804	22.491
		34.0923	30.7056	15.4602	54.265	153.2249	144.9759	56.2004	226.4303
	F. bruta	1.6453	1.3117	1.7231	2.4663	6.1342	20.9788	4.6705	20.4701
		17.3516	16.0953	15.0326	20.7511	153.5585	144.1796	22.2036	130.603
Time	Diosi	0.002	0.0021	0.0025	0.0025	0.0026	0.0034	0.0028	0.0148
		0.0065	0.0061	0.0082	0.0085	0.0094	0.0099	0.0085	0.0342
	E-PSM	0.0689	0.0676	0.0739	0.087	0.0665	0.0808	0.0752	0.1007
		0.2902	0.2745	0.094	0.3439	0.2762	0.1642	0.1949	0.3623
	F. bruta	10.7171	10.4781	10.2197	11.1113	10.9715	12.5948	10.6354	15.0517
		15.5487	15.1178	13.5278	16.2456	16.3084	22.96	16.2407	36.1591

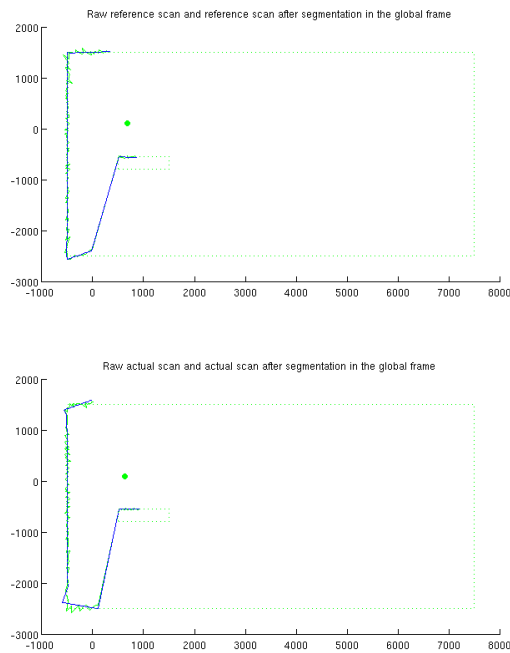


Figura 35: Escáneres de referencia y actual en el plano XY (No. Paso = 30)

Para el mapa rectangular, con un $\text{error}_{\text{odom}}$ de 10% y un σ_{laser} de 20mm., esta figura representa el recorrido del robot de izquierda a derecha (la posición del robot se representa por un punto verde) y los escáneres obtenidos en cada paso (marcados como líneas azules) y las paredes del entorno (líneas verdes discontinuas). Concretamente, se presenta las posiciones del robot entre la 29ª y 30ª vez que se utiliza el invoca el proceso de *scan-matching*. Nótese como el ruido presente en las lecturas láser hace que las esquinas de las paredes sean representas con cierto nivel de distorsión.

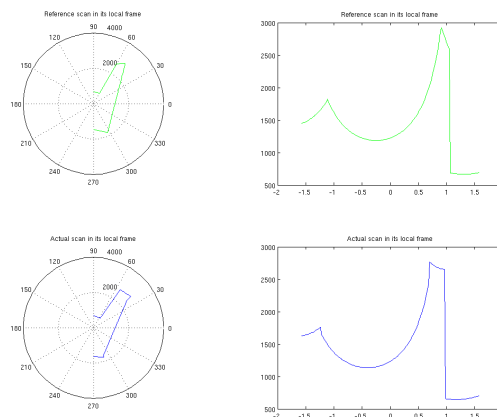


Figura 36; Escáneres de referencia y actual en coordenadas polares (No. Paso = 30)

Esta figura proporciona la representación en coordenadas polares de los escáneres mostrados en la figura anterior. En este caso, la línea verde representa el escáner de referencia y la línea azul el escáner actual. Nótese que esta representación ya no es ruidosa aunque ciertamente sufra de un cierto nivel de distorsión que no pudo ser corregido por el algoritmo de filtrado y segmentación.

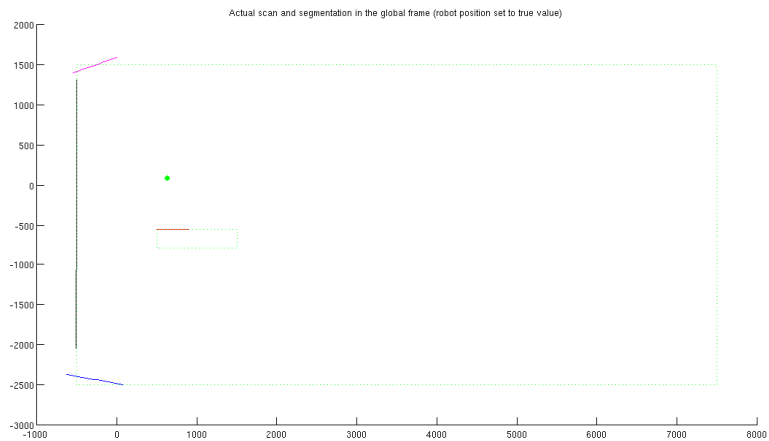


Figura 37: Segmentación del escáner actual (No. Paso = 30)

Esta figura representa el proceso de segmentación aplicado a los escáneres obtenidos anteriormente. En este caso, se observan en el plano XY cuatro segmentos diferentes que corresponden a las tres paredes y a parte de la columna que se encuentran en frente del robot para este paso. Se puede observar que las paredes superior e inferior presentar cierto nivel de error en la estimación de su orientación debido al nivel de ruido introducido en las medidas láser.

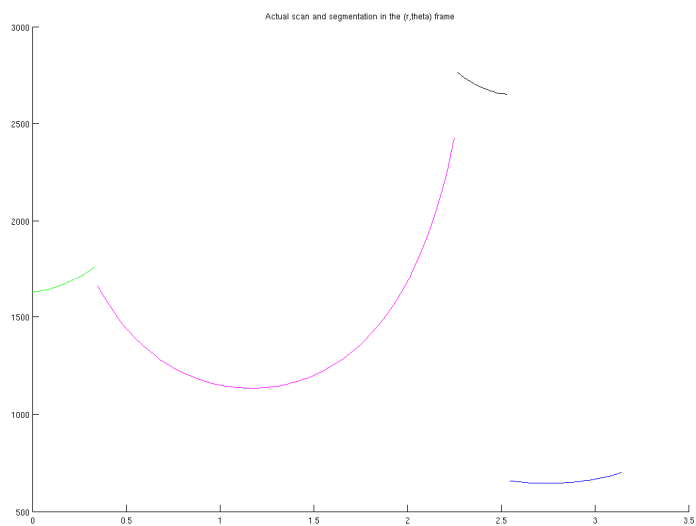


Figura 38: Segmentación del escáner actual en coordenadas polares (No. Paso = 30)

Esta figura presenta la versión polar de los segmentos identificados en la figura anterior.

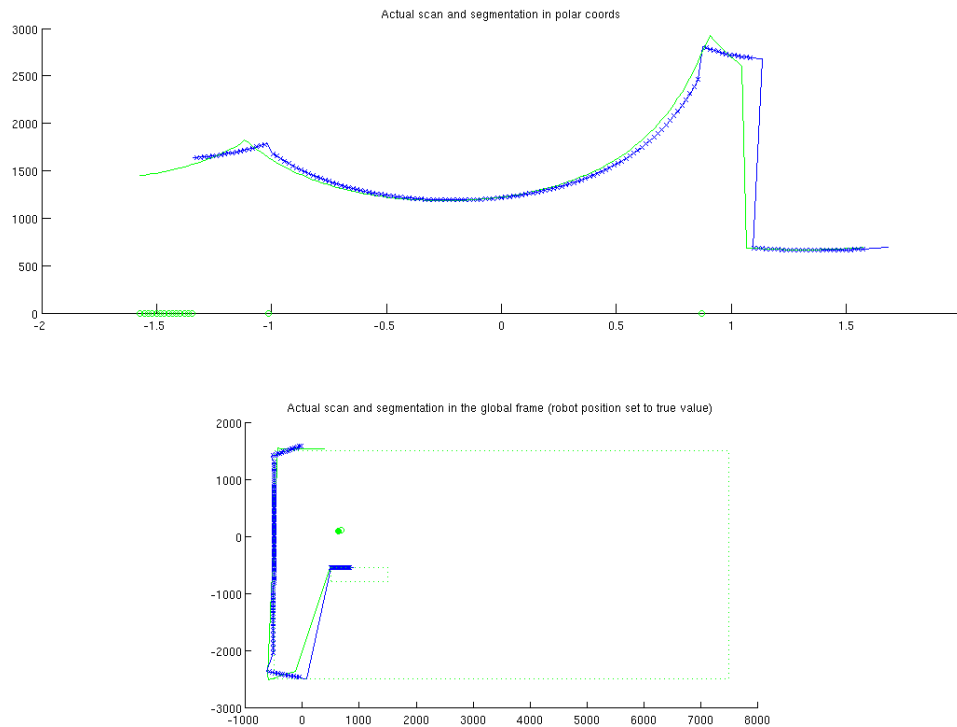


Figura 39: Alineamiento de los escáneres de referencia y actual (No. Paso = 30)

Esta figura muestra, para el 30º paso, el alineamiento obtenido entre el *current scan* y el *actual scan* una vez se ha hallado el valor óptimo de la pose del robot. Los puntos representados por cruces azules corresponden a los definidos como válidos para el proceso de *scan-matching*. Aquellos puntos marcados con un círculo verde corresponden a los datos descartados por: (1) No tener un punto de correspondencia en el *current scan*, o (2) por haber sido identificados como no-válidos por el algoritmo de segmentación de líneas. Los puntos marcados con un triángulo invertido de color cian corresponden a aquellos descartados por el nuevo criterio de oclusión definido en el capítulo 5.2. Aunque el proceso de alineamiento sea exitoso resulta evidente que no es posible compensar con exactitud las distorsiones introducidas a cada paso por el ruido en las medidas láser. Así, el alineamiento entre escáneres nunca será exacto aunque si se asegura un buen comportamiento global la estimación dada por el proceso de *scan-matching* será de alta calidad-

6.2 Búsqueda simultánea de posición y orientación

El desarrollo de esta alternativa fue motivada por la presencia de errores en el algoritmo de fuerza bruta presentado anteriormente. La magnitud de los errores presentes era mayor de lo esperado teniendo en cuenta el tamaño de los intervalos en los que la función objetivo fue evaluada (se hicieron evaluaciones cada 0.01 grados y 10mm mientras que el error de localización del robot tenía errores de un orden de magnitud que estos valores). Así se consideró que la aproximación de alternas las búsquedas de orientación y posición pudiera ser la fuente de error al inducir posiblemente que el algoritmo de búsqueda se quedara bloqueado en un mínimo

local. De hecho, esta suposición fue el origen de implementar en el algoritmo de búsqueda de la orientación la regla por la cual la evaluación no continua a la fase de búsqueda fina hasta que se haya encontrado un mínimo relativo; esta modificación por sí sola mejoró la precisión del algoritmo E-PSM en algunos casos de prueba.

Para analizar sistemáticamente la propuesta de realizar simultáneamente la búsqueda de la posición y orientación el robot, se optó por probar diferentes grupos de algoritmos de los disponibles en la librería *nlopt* [Johnson]:

- Algoritmos de optimización global: DIRECT-L, CRS2, ISRES and MLSL-LDS
- Algoritmos de optimización local no basados en gradiente: COBYLA, BOBYQA, and SBPLX
- Algoritmos de optimización local basados en gradiente: SLSQP, BFGS, NEWTON, and VAR2

Información detallada acerca de cada uno e estos algoritmos se encuentra disponible en [Johnson].

La Tabla 3 detalla, para cada uno de los casos de prueba definidos anteriormente, los valores medios obtenidos para dos diferentes variables: f (función objetivo) y $Time$ (tiempo computacional usado por la función *scan-matching* por cada segmento recorrido). Los datos obtenidos muestran que en lo respectivo a la minimización de la función objetivo el algoritmo de Diosi original (y en segundo lugar los algoritmos no basados en gradiente COBYLA, BOBYQA y SBPLX) proporciona los peores resultados. Los algoritmos locales basados en gradiente BFGS, NEWTON, and VAR2 tampoco se muestran capaces de proporciona resultaos precisos de forma consistente. En algunos casos son capaces de obtener resultados comparables a los de las estrategias globales pero los resultados llegan a niveles inaceptables en los casos de ruido combinado (σ_{laser} de 5mm y $error_{odom}$ de 5%). En conjunto, resulta evidente que ni el algoritmo original de Diosi ni ninguna de las aproximaciones de optimización locales es capaz de proporcionar consistentemente estimaciones de la pose del robot precisas.

Tabla 3: Resultados de búsqueda simultanea de orientación y posición

		No noise	Odometry 1	Odometry 5	Odometry 10	Scan 5	Scan 20	Scan 5 Odo 5%
<i>f</i>	Diosi	8.42	9.93	14.36	19.16	20.82	45.00	19.37
	E-PSM	2.08	1.34	1.39	3.77	6.75	23.36	6.08
	Brute Force	1.65	1.31	1.72	2.47	6.13	20.98	4.67
	DIRECT-L	1.19	1.64	1.89	2.06	3.73	19.50	3.85
	CRS2	1.37	1.46	1.29	1.49	3.61	19.73	3.53
	ISRES	1.15	1.21	1.31	1.59	3.52	19.47	3.57
	GD_MLSL_LDS	1.16	0.98	0.99	1.36	3.47	19.39	3.50
	COBYLA	2.31	6.29	11.75	18.21	9.58	23.96	15.85
	BOBYQA	2.49	7.00	13.07	14.79	9.58	26.72	14.00
	SBPLX	1.43	1.86	3.65	5.90	3.77	19.92	6.81
	SLSQP	1.82	1.63	7.21	12.06	4.75	22.24	12.99
	BFGS	1.88	1.68	3.84	4.11	5.94	24.38	12.46
	NEWTON	1.88	1.68	2.03	3.85	6.09	24.39	12.27
	VAR2	1.88	1.67	3.85	3.12	5.99	24.35	12.50
	Time	Diosi	0.002	0.0021	0.0025	0.0025	0.0026	0.0034
E-PSM		0.0689	0.0676	0.0739	0.087	0.0665	0.0808	0.0752
Brute Force		10.72	10.48	10.22	11.11	10.97	12.59	10.64
DIRECT-L		2.11	3.90	4.66	4.79	11.60	15.44	11.60
CRS2		0.0576	0.0946	0.0918	0.0838	0.1127	0.1194	0.1252
ISRES		0.21	0.77	0.90	0.87	1.74	1.91	1.51
GD_MLSL_LDS		9.29	6.16	6.31	9.34	53.00	60.66	45.43
COBYLA		0.0048	0.0067	0.0078	0.008	0.0083	0.0078	0.0083
BOBYQA		0.0052	0.0068	0.0087	0.0089	0.0077	0.0079	0.0086
SBPLX		0.0076	0.0177	0.0266	0.0319	0.0265	0.0354	0.0378
SLSQP		0.0056	0.0075	0.0102	0.0121	0.0127	0.0172	0.0173
BFGS		0.0088	0.0115	0.0143	0.018	0.0176	0.0222	0.0197
NEWTON		0.0091	0.0131	0.0199	0.0245	0.0195	0.0217	0.0242
VAR2		0.0086	0.0119	0.0153	0.0162	0.0181	0.0199	0.0199

Como era de esperar, los cuatro algoritmos de optimización global obtienen los mejores resultados en lo que respecta a la precisión en todos los casos de prueba (de entre ellos, el algoritmo MLSL_LDS obtiene el mejor resultado en la mayoría de los casos). En lo que respecta al algoritmo E-PSM propuesto en este trabajo, los resultados muestran que obtiene resultados sólo ligeramente inferiores que sus contrincantes globales (exceptuando el caso de un error_{odom} de 10% donde E-PSM obtiene resultados notablemente peores).

En lo que respecta al uso de tiempo computacional, se pueden identificar los siguientes grupos en orden creciente de requerimientos computacionales: PSM/Diosi (en torno a 0.003 segundos por cada paso), algoritmos de optimización locales (entre 0.004 y 0.005 segundos), E-PSM y CRS2 (entre 0.07 y 0.1 segundos) y finalmente el resto de algoritmos de optimización global y la aproximación de fuerza bruta (varios segundos o más).

Estos resultados muestran que la aplicación de un algoritmo de búsqueda simultánea de la posición y orientación puede resultar ser ventajosa pero solo cuando se emplean métodos globales. De éstos, el algoritmo CRS2 [Kaelo06, Price83] se presenta como el más prometedor ya que obtiene una precisión ligeramente mejor que la del E-PSM con requerimientos computacionales ligeramente más altos.

6.3 Efecto del uso del algoritmo de segmentación

En las secciones anteriores se aplicó el algoritmo de filtrado y segmentación introducido en el capítulo 4 sin mostrar que estaba realmente introduciendo mejoras en el proceso de *scan-matching* global. En esta sección se clarifica si el efecto de este algoritmo de filtrado y segmentación es beneficioso. Para ello, usando el mapa rectangular de la Figura 8 para todos los casos de prueba de la Tabla 1, se simula el comportamiento global del algoritmo de *scan-matching* en dos configuraciones:

- Diosi: Uso del algoritmo de [Diosi07a] pero con sus entradas sustituidas por la salida del algoritmo del capítulo 4 en vez de ser el resultado de aplicar las reglas heurísticas propuestas en [Diosi07a]
- Diosi Original: Uso del Algoritmo de Diosi según lo descrito en [Diosi07a]

La Tabla 4 muestra los resultados obtenidos en la optimización de la función objetivo f y el tiempo computacional requerido por cada proceso de *scan-matching*. Los resultados muestran que en ausencia de ruido en los retornos de las señales láser los dos algoritmos proporcionan una precisión similar. Sin embargo, en los casos de prueba en los que se introduce ruido en las medidas de alcance el método que usa el algoritmo de segmentación y filtrado del capítulo 4 obtiene significativamente mejores resultados (error de 7.3 en vez de 14 con sigma de 5mm, error de 22.1 en vez de 40.7 con sigma de 20mm) tanto si se introduce ruido en los sensores de odometría como si no. Además se puede observar que los requerimientos computacionales de las dos aproximaciones son equivalentes. Estos resultados validan que la capacidad del algoritmo de

scan-matching se ve mejorada substancialmente al introducir en el método un proceso avanzado de tratamiento de datos ruidosos como el presentado en el capítulo 4 .

Tabla 4: Evaluación del uso del algoritmo de segmentación

AVERAGE ERRORS		No noise	Odometry 1	Odometry 5	Odometry 10	Scan 5	Scan 20	Scan 5 Odo 5%	Scan 20 Odo 10%
Error	Diosi	6,172	4,4555	5,2674	6,5863	7,3347	21,2265	8,2647	23,2304
	Diosi Original	4,463	5,7642	5,9116	6,0475	14,0716	40,6741	14,816	38,3966
Time	Diosi	0,0106	0,0105	0,011	0,0111	0,0115	0,0114	0,0118	0,0123
	Diosi Original	0,0102	0,0104	0,0115	0,0113	0,0107	0,0117	0,0112	0,0125
No. evals	Diosi	14,32	14,2	14,85	15,18	14,4	15,92	15,51	17,43
	Diosi Original	14,04	14,18	15,26	15,18	14,52	16,4	15,32	17,21

6.4 Ensayos en otros dos mapas adicionales con cambios adicionales en el algoritmo de E-PSM

En las secciones 6.1 y 6.2 se mostró como los dos algoritmos más prometedores eran el E-PSM y su variante basada en el algoritmo de optimización global CRS2 [Kaelo06, Price83]. En esta sección se propone ampliar los resultados anteriores realizando pruebas de simulación en dos mapas adicionales que corresponden a la Figura 9 y la Figura 10. Además, durante la evaluación del algoritmo E-PSM detectaron dos posibles mejoras cuya implementación también se evalúa: (1) la comprobación de si el valor de f realmente disminuye con la pose estimada por *scan-matching*; de lo contrario se rechaza la actualización y se usa la posición computado a partir de datos odométricos, (2) la sustitución del método de búsqueda de la posición basada mínimos cuadrados de Diosi [Diosi07a] (debido su limitación a cambios pequeños) por una búsqueda de la posición basada en un algoritmo de optimización local no basado en gradiente. Para ello se eligió el algoritmo BOBYQA al haber obtenido valores representativos de este tipo de aproximación.

Los resultados obtenidos para el algoritmo E-PSM y su variante con CRS2 se presentan en la Tabla 5. Se observa que los datos se presentan divididos en tres partes (cada una

correspondiendo a uno de los tres mapas usados). Los datos representan los errores medios obtenidos en la estimación de la pose y de la función objetivo así como el tiempo computacional consumido por el algoritmo de scan-matching en cada paso. Al igual que se hizo en los casos anteriores, de la Figura 40 a la Figura 51 se presentan gráficos que permiten entender cómo el proceso propuesto se aplica en un mapa más complejo como el de la Figura 10.

La casi totalidad de los casos de prueba muestran que la variante basada en CRS2 obtiene mejores resultados en lo que se refiere a precisión. Sin embargo, las diferencias no son significativas excepto en el mapa grande para el caso de tener 10% de ruido en la odometría y en el mapa poligonal para el caso de tener 1% de ruido en la odometría. Por otra parte, se observa que la formulación de E-PSM original consume menos tiempo computacional en la mayoría de las pruebas, especialmente en las más exigentes con ahorro de hasta el 50% sobre un tiempo de 0.1 segundos. Por tanto, se deduce que las dos aproximaciones proporciona una buena combinación de precisión y eficiencia siendo la elección de entre las dos dependiente del caso de aplicación particular.

Más allá del problema de la localización, el trabajo aquí presentado permite también generar mapas del entorno siguiendo la aproximación incremental presentada en la sección 2.4.1 . A modo de ejemplo, en el apéndice, de la Figura 52 a la Figura 59 se proporcionan los mapas generados en el caso del entorno poligonal (Figura 9). Se constata que se obtiene un contorno con gran precisión incluso en presencia de un alto nivel de ruido odométrico. Cuando también se incluye ruido en el láser, se observa cómo el contorno registrado por el algoritmo se convierte en ruidoso de forma proporcional al ruido presente en el retorno de las señales láser.

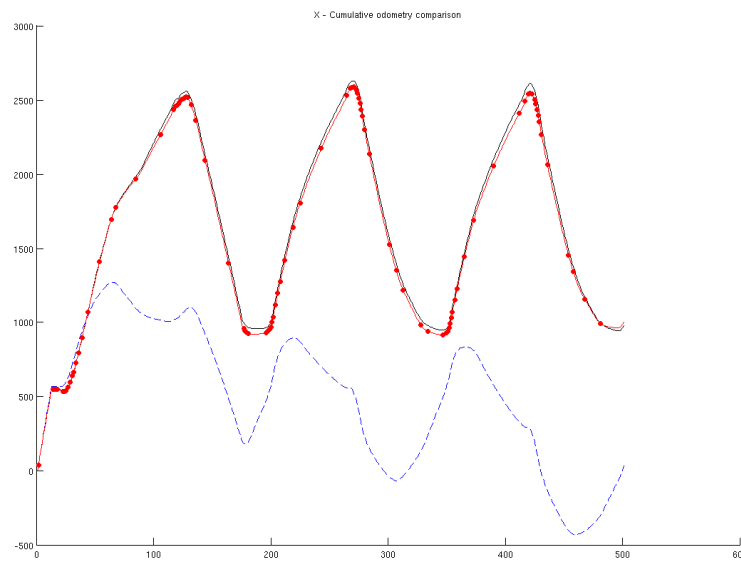


Figura 40: Posición absoluta del robot (x-coord) en función del tiempo

Para el mapa de gran tamaño y con un $\text{error}_{\text{odom}}$ de 10%, esta figura muestra la posición absoluta del robot (en el eje de abscisas) computada por el algoritmo E-PSM (línea roja), la posición exacta (línea negra), y la posición computada a partir de datos odométricos únicamente (línea azul discontinua). Los puntos rojos muestran el momento en el que se realiza el proceso de *scan-matching* con el fin de corregir la desviación introducida en la computación de la posición por el ruido presente en la odometría. Al igual que en casos anteriores, se puede observar como la localización ofrecida por la solución odométrica se desvía rápidamente de la solución exacta y cómo la solución calculada por el algoritmo E-PSM consigue rectificar buena parte del error introducido por el ruido en los sensores odométricos.

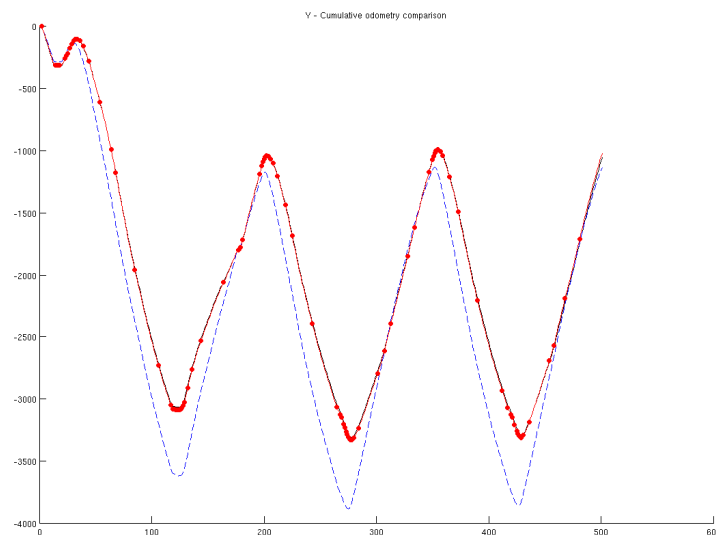


Figura 41: Posición absoluta del robot (y-coord) en función del tiempo

Para el mismo caso de la figura anterior, se presenta las diferentes posiciones computadas del robot en el eje de ordenadas.

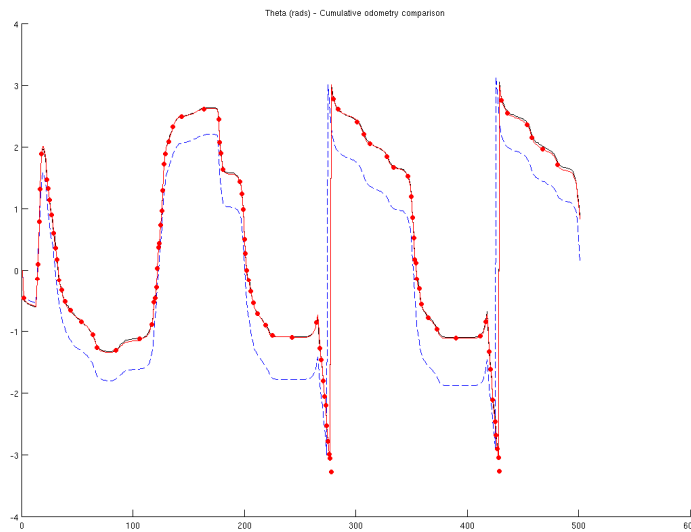


Figura 42: Orientación del robot (en rad) en función del tiempo

Para el mapa de gran tamaño y con un $\text{error}_{\text{odom}}$ de 10%, esta figura muestra la orientación robot (en radianes) computada por el algoritmo E-PSM (línea roja), la orientación exacta (línea negra), y la orientación computada a partir de datos odométricos únicamente (línea azul discontinua). Los puntos rojos muestran el momento en el que se realiza el proceso de *scan-matching*.

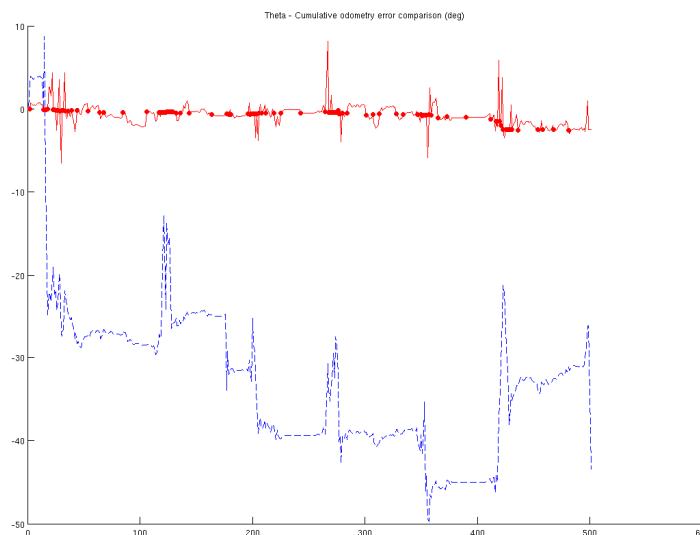


Figura 43: Error acumulado en la estimación de la orientación del robot (deg)

Para el mismo caso de la figura anterior, esta figura presenta el error (en grados) que se acumula en la estimación de la orientación del robot según éste se desplaza. Se observa que el error en la estimación de la odometría (línea azul discontinua) sufre de grandes saltos que coinciden con los momentos en los que el robot de grandes giros sobre sí mismo a gran velocidad. Es importante notar que los picos en la línea roja (E-PSM) corresponden al intervalo antes de que el algoritmo de *scan-matching* es llamado. Así, se observa que después de aplicar la corrección dada por el algoritmo E-PSM (puntos rojos) el valor de la línea roja (error acumulado) retorna a valores pequeños.

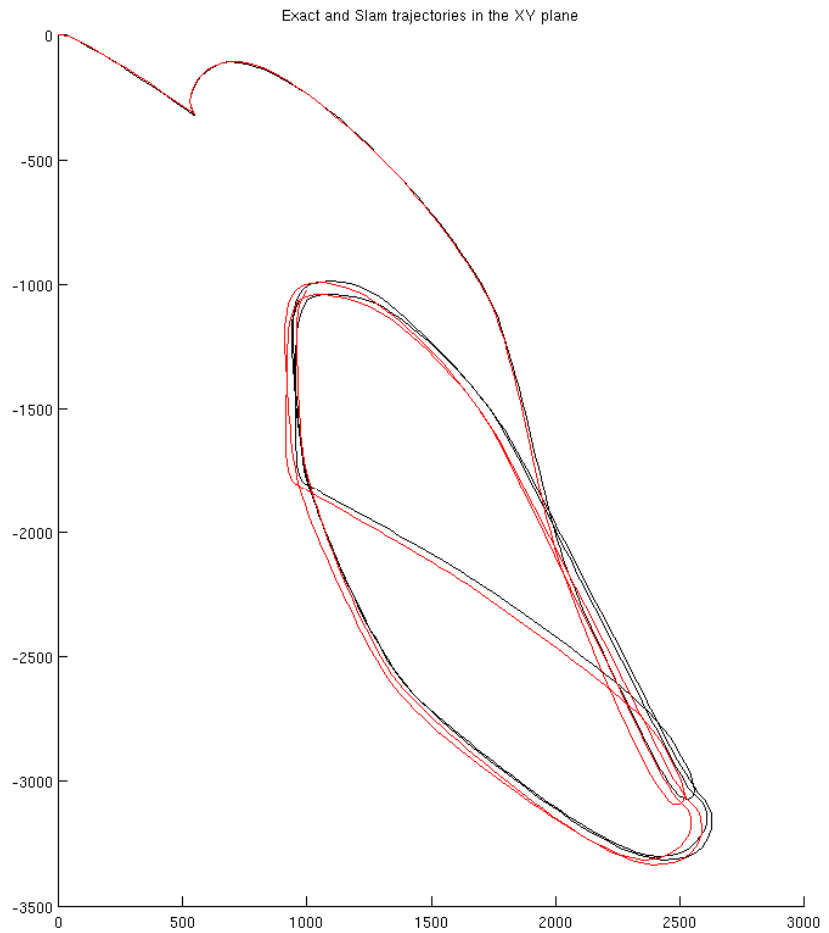


Figura 44: Trayectoria del robot en el plano XY

Para el mapa de gran tamaño y con un $\text{error}_{\text{odom}}$ de 10%, esta figura muestra la trayectoria exacta (línea negra) y la trayectoria calculada por el algoritmo E-PSM (línea roja). La trayectoria computada únicamente por odometría no se incluye por salirse fuera de la escala rápidamente. Se observa cómo, a pesar del alto nivel de ruido, el algoritmo E-PSM consigue trazar con fidelidad la trayectoria del robot.

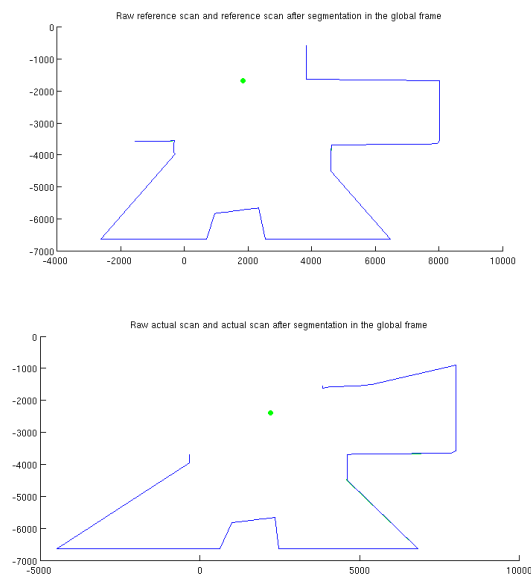


Figura 45: Escáneres de referencia y actual en el plano XY (No. Paso = 55)

Para el mapa de gran tamaño y con un $\text{error}_{\text{odom}}$ de 10%, esta figura representa el recorrido del robot (la posición del robot se representa por un punto verde) y los escáneres obtenidos en cada paso (marcados como líneas azules) y las paredes del entorno (líneas verdes discontinuas). Concretamente, se presenta las posiciones del robot entre la 54ª y 55ª vez que se utiliza el invoca el proceso de *scan-matching*. Nótese también que, en este caso, el escáner obtenido es preciso pues se impuso un σ_{laser} de 0mm.

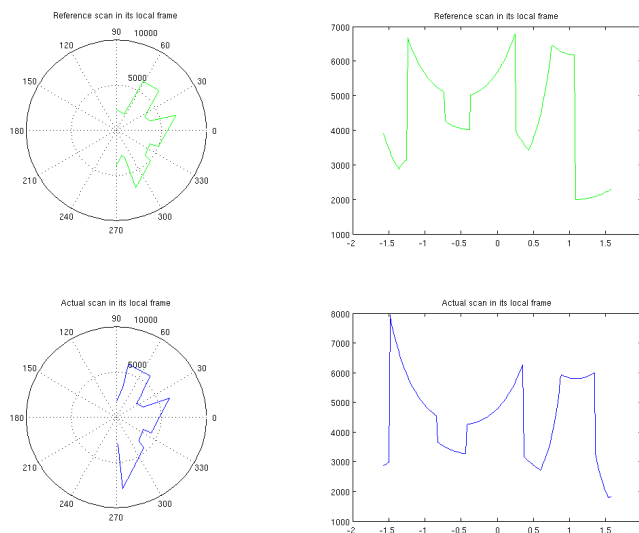


Figura 46: Escáneres de referencia y actual en coordenadas polares (No. Paso = 55)

Esta figura proporciona la representación en coordenadas polares de los escáneres mostrados en la figura anterior. En este caso, la línea verde representa el escáner de referencia y la línea azul el escáner actual.

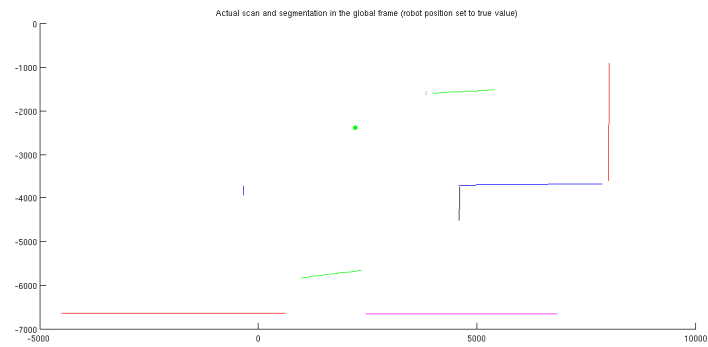


Figura 47: Segmentación del escáner actual (No. Paso = 55)

Esta figura representa el proceso de segmentación aplicado a los escáneres obtenidos anteriormente. En este caso, se observan en el plano XY numerosos segmentos diferentes que corresponden al entorno complejo en el que se mueve el robot. Los buenos resultados obtenidos para un mapa de gran tamaño y relativa complejidad confirman la robustez del método propuesto.

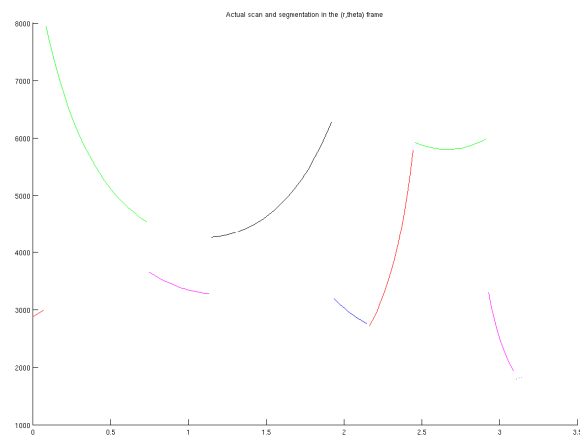


Figura 48: Segmentación del escáner actual en coordenadas polares (No. Paso = 55)

Esta figura presenta la versión polar de los segmentos identificados en la figura anterior.

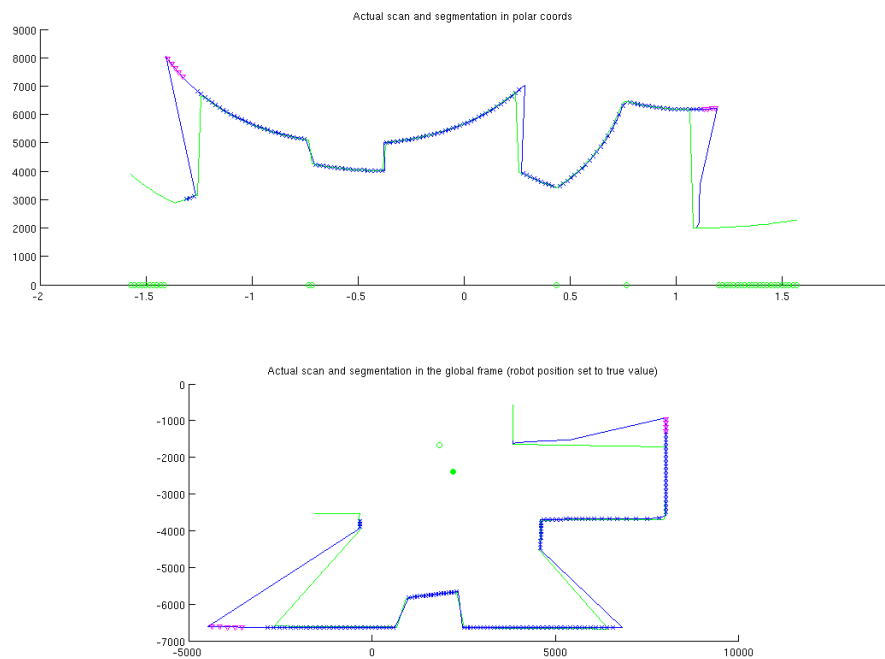


Figura 49: Alineamiento de los escáneres de referencia y actual (No. Paso = 55)

Esta figura muestra, para el 55º paso, el alineamiento obtenido entre el *current scan* y el *actual scan* una vez se ha hallado el valor óptimo de la pose del robot. Los puntos representados por cruces azules corresponden a los definidos como válidos para el proceso de *scan-matching*. Aquellos puntos marcados con un círculo verde corresponden a los datos descartados por: (1) No tener un punto de correspondencia en el *current scan*, o (2) por haber sido identificados como no-válidos por el algoritmo de segmentación de líneas. Los puntos marcados con un triángulo invertido de color cian corresponden a aquellos descartados por el nuevo criterio de oclusión definido en el capítulo 5.2. Se observa que el alineamiento entre los dos escáneres es óptimo. De nuevo, este ejemplo demuestra la robustez del procedimiento propuesto y su capacidad de adaptarse a entornos complejos.

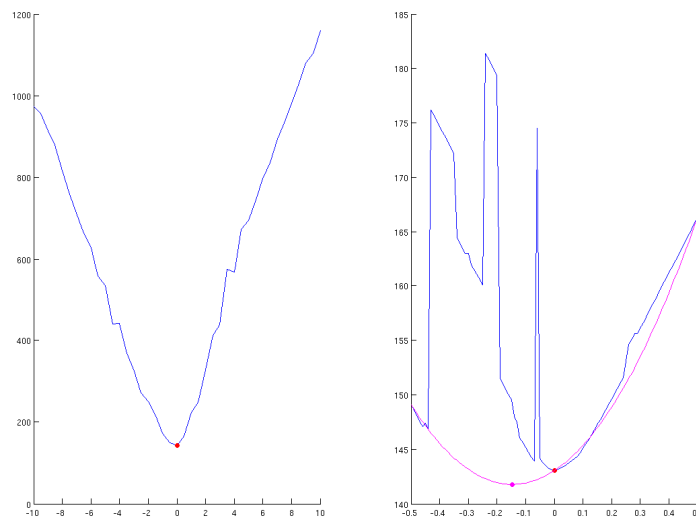


Figura 50: Búsqueda de la orientación (No. Paso = 55)

En el contexto de la búsqueda de la orientación durante el 55º paso del *scan-matching*, esta figura presenta las variaciones en la función objetivo en función de θ_{robot} para una misma posición. La gráfica izquierda muestra la búsqueda inicial en intervalos de un grado y la gráfica de la derecha la búsqueda refinada realizada en intervalos de 0.01 grados. La línea cian representa la aproximación parabólica propuesta de Diosi. Se puede observar que ambas gráficas muestran un mal comportamiento con múltiples mínimos locales. Además se observa que es este caso la aproximación parabólica de Diosi resulta en un error mayor a los 0.1 grados.

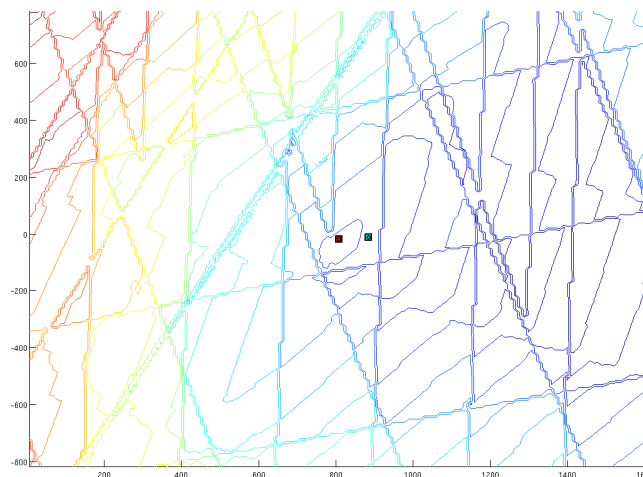


Figura 51: Contorno de la función objetivo en el plano XY (No. Paso = 55)

En el contexto de la búsqueda de la posición durante el 55º paso del *scan-matching*, esta figura muestra los contornos de la función objetivo en el plano XY (para un mismo θ_{robot}) en el área alrededor de la posición estimada a partir de datos odométricos. La solución obtenida y la posición exacta se representan con un cuadrado verde y rojo respectivamente. En este caso se observa que la función presenta contornos muy irregulares aunque aún así la solución obtenida por el algoritmo E-PSM resulte muy cercana a la posición exacta.

Tabla 5: Resultados para varios mapas con el algoritmo E-PSM y CRS2

Rect02-oclu									
AVERAGE ERRORS		No noise	Odometry 1	Odometry 5	Odometry 10	Scan 5	Scan 20	Scan 5 Odo 5%	Scan 20 Odo 10%
ax	E-PSM	4.15E-06	0.1968	1.1264	1.7383	0.3743	1.638	1.2197	3.9116
	CRS2	3.83E-06	0.242	1.2178	2.2727	3.59E-06	4.16E-06	1.1187	2.2517
ay	E-PSM	8.22E-05	0.1667	0.9004	1.7642	0.1166	0.6192	1.0292	1.6539
	CRS2	7.42E-05	0.1578	0.8962	1.7064	9.39E-05	9.52E-05	0.9915	1.2393
atheta	E-PSM	0.024	0.0239	0.0429	0.088	0.0743	0.2021	0.0954	0.2121
	CRS2	0.0161	0.0253	0.034	0.0355	0.0732	0.2717	0.0668	0.2603
Error	E-PSM	1.4906	1.0596	1.6707	3.1891	4.8249	23.3575	5.2144	20.724
	CRS2	1.3676	1.4551	1.2901	1.4865	3.6111	19.731	3.5305	17.1499
Time	E-PSM	0.0837	0.0875	0.0985	0.1028	0.0834	0.0882	0.0971	0.1039
	CRS2	0.0576	0.0946	0.0918	0.0838	0.1127	0.1194	0.1252	0.1275
map2-crow									
AVERAGE ERRORS		No noise	Odometry 1	Odometry 5	Odometry 10	Scan 5	Scan 20	Scan 5 Odo 5%	Scan 20 Odo 10%
ax	E-PSM	0.0031	0.1703	0.8433	1.1607	0.2836	1.012	1.2799	2.3452
	CRS2	2.49E-06	0.2405	1.0815	1.9699	2.08E-06	1.93E-06	1.2414	1.6439
ay	E-PSM	0.0011	0.1219	0.5187	1.0378	0.0743	0.6699	0.7841	1.8083
	CRS2	7.21E-05	0.0842	0.4938	1.1513	7.17E-05	8.47E-05	0.4476	1.379
atheta	E-PSM	0.0076	0.0167	0.0184	0.0576	0.0847	0.3269	0.0758	0.3249
	CRS2	0.0209	0.0233	0.0307	0.0352	0.0768	0.4778	0.0914	0.55
Error	E-PSM	3.6136	4.2407	2.8564	6.2484	10.3567	40.405	11.2151	39.8342
	CRS2	5.6193	3.3113	2.7579	2.8123	9.6741	37.948	9.241	40.4912
Time	E-PSM	0.0822	0.0868	0.091	0.0925	0.0803	0.0999	0.0999	0.1106
	CRS2	0.0513	0.0759	0.0897	0.1037	0.1336	0.1246	0.1361	0.1283
bigmap2									
AVERAGE ERRORS		No noise	Odometry 1	Odometry 5	Odometry 10	Scan 5	Scan 20	Scan 5 Odo 5%	Scan 20 Odo 10%
ax	E-PSM	3.40E-06	5.149	3.5773	1.7661	4.07E-06	0.0161	3.3702	1.8804
	CRS2	3.40E-06	0.1583	0.7268	1.7629	4.07E-06	3.54E-06	1.5858	1.953
ay	E-PSM	5.88E-05	5.0815	2.405	1.649	6.09E-05	0.3031	2.6335	1.7677
	CRS2	5.88E-05	0.1647	0.9955	1.6528	6.09E-05	5.12E-05	0.9004	1.8363
atheta	E-PSM	0.0343	0.2936	0.1434	0.0564	0.0753	0.1682	0.1311	0.1784
	CRS2	0.0414	0.1852	0.0594	0.0637	0.0943	0.2664	0.2018	0.2173
Error	E-PSM	11.9286	54.4099	17.8341	14.5006	17.2527	63.6684	33.7155	54.8992
	CRS2	9.0545	15.1918	8.5343	9.5872	13.1004	54.5207	27.0051	50.9141
Time	E-PSM	0.084	0.093	0.0988	0.1037	0.084	0.0844	0.1004	0.1005
	CRS2	0.0938	0.1095	0.1042	0.1076	0.1301	0.1495	0.1178	0.1269

7 Conclusiones y trabajos futuros

7.1 Conclusiones

En este trabajo se ha desarrollado un procedimiento mejorado de PSM [Diosi07a] que se ha denominado E-PSM (*Enhanced Polar Scan Matching*). La metodología consiste de un algoritmo de *scan-matching* relativo, de punto a punto, en coordenadas polares. Se definieron varias mejoras sobre el PSM original con el objeto de mejorar su precisión pero manteniendo en lo posible su rapidez y, por tanto, su capacidad para ser usado en tiempo real. En primer lugar, un avanzado algoritmo de segmentación de línea se aplicó a las lecturas de los sensores láser en vez de las normas heurísticas usadas en [Diosi07a]. En segundo lugar, se añadió un nuevo criterio para seleccionar los puntos del escáner aptos para el proceso de comparación. Finalmente, se definieron nuevas mejoras en los métodos usados para mejorar los procesos de búsqueda de la orientación y la posición del robot.

El procedimiento en conjunto consta de cuatro pasos principales: (1) Filtrado y segmentación de los escáneres de referencia y actuales (*reference scan & actual scan*) usando los algoritmos diseñados por Cuadra [Cuadra11b] según son descritos en el Capítulo 4. (2) Proyección e interpolación del *current scan* al eje de coordenadas de referencia siguiendo la aproximación de Diosi [Diosi07a]. Ver sección 5.1. (3) Descarte de los datos de alcance del escáner que no deban ser usados por no ser adecuados para el proceso de comparación. Ver sección 5.2. (4) Búsqueda de la pose del robot que permita alinear de manera óptima el *current scan* y al *actual scan*. Se establecieron dos opciones con un excelente balance entre precisión y prestaciones (ver sección 6.4): una de búsqueda alterna de orientación y posición y una de búsqueda simultánea de orientación y posición usando el algoritmo de optimización global CRS2 [Kaelo06, Price83].

Para probar el algoritmo E-PSM propuesto se han realizado numerosas pruebas de simulación en tres mapas diferentes usando la herramienta *cybersim*. El primer resultado a destacar es la constatación que el uso del algoritmo de filtrado y segmentación desarrollado por Cuadra [Cuadra11b] (que reemplaza las reglas heurísticas usadas en [Diosi07a]) permite reducir significativamente el error obtenido en el proceso de *scan-matching*.

En el mapa rectangular de la Figura 8 y considerando los diferentes niveles de ruido en los sensores de odometría y en las medidas de alcance láser del robot, los resultados de las pruebas de simulación obtenidos muestran que tanto el algoritmo E-PSM propuesto originalmente en este trabajo (Figura 6) como la variante definida que usa el método de optimización global CRS2 obtienen buena precisión en la estimación de la pose del robot a la vez que se mantienen los costes computacionales en un nivel razonable. Por otro lado, se comprobó que el algoritmo original de Diosi [Diosi07a] y los basados en métodos locales de optimización no son capaces de

dar resultados precisos consistentemente. Las variantes del E-PSM que usan métodos de optimización global obtienen los resultados más precisos; desafortunadamente, excepto en el caso del algoritmo CRS2, los costes computacionales asociados a su uso no los hacen prácticos para muchas aplicaciones. Por tanto, se puede concluir que el algoritmo E-PSM y su variante basada en CRS2 son los que prestan el comportamiento mejor equilibrado entre precisión y coste computacional.

Estos resultados fueron validados mediante pruebas en dos mapas adicionales usando además una versión mejorada del algoritmo E-PSM (Figura 7). Se observa que la variante basada en CRS2 obtiene los mejores resultados en cuanto a precisión se refiere aunque requiera un coste ligeramente mayor al del E-PSM original.

Todos los ensayos realizados aportan gran confianza en la robustez y eficacia de los algoritmos propuestos en este trabajo y en la mejora que introducen respecto al trabajo original de Diosi.

7.2 Trabajos futuros

Más allá de las comparaciones hechas con el trabajo de Diosi, existe una literatura técnica muy extensa que proporciona una multitud de resultados obtenidos en simulaciones y en ensayos reales. Ver, por ejemplo, el proyecto *Rawseeds* presentado en [Burgard01]. En este sentido, será necesario profundizar el estudio presentado este trabajo mediante la realización de simulaciones que puedan ser comparadas con aproximaciones al SLAM totalmente diferentes a las usadas aquí o en [Diosi07a].

Adicionalmente, sería muy beneficioso realizar pruebas en robot físicos en entornos interiores de diversa índole para obtener otros conjuntos de datos y comprobar si las suposiciones realizadas acerca del ruido presente son suficientemente conservadores. También será necesario comprobar cómo afectan al proceso de *scan-matching* las irregularidades que surjan de un algoritmo de filtrado y segmentación sometido a escenario no controlados de forma precisa como ocurre en las simulaciones realizadas hasta ahora.

Considerando los objetivos alcanzados en este trabajo y las limitaciones observadas en el mismo, varias líneas de trabajo para el futuro se han identificado:

- Optimización de la implementación del algoritmo para optimizar su uso de recursos computacionales
- Revisión de las técnicas de interpolación empleadas
- Adición de otras técnicas para mejorar el comportamiento del algoritmo de proyección al estar sometido a oclusiones
- Creación de mapas con técnicas avanzadas a partir de las estimaciones de la pose del robot obtenidas por *scan-matching*
- Extensión del algoritmo E-PSM a ambientes dinámicos y entornos no poligonales

Con estas mejoras, se pretendería extender la utilidad del algoritmo propuesto a entornos más complejos y facilitar a las tareas de un robot móvil con la creación de mapas complejos manteniendo a la vez un nivel de prestaciones que permitan realizar estas tareas en tiempo real.

8 Referencias

- [Bekey05] G. Bekey, *Autonomous Robots: From Biological Inspiration to Implementation and Control*. The MIT Press, 2005.
- [Besl92] P.J. Besl and H.D. McKay. A method for registration of 3-D shapes. *IEEE Transactions on pattern analysis and machine intelligence*, 14(2):239-256, 1992.
- [Biber06] P. Biber and W. Strasser. nScan-matching: simultaneous matching of multiple scans and application to slam. In *IEEE International Conference on Robotics and Automation*, 2006.
- [Bishop01] G. Bishop and G. Welch, "An Introduction to the Kalman Filter," in *Proc of SIGGRAPH*, Course 8, no. NC 27599-3175, Chapel Hill, NC, USA, 2001.
- [Borthwick94] Borthwick, S., Durrant-Whyte, H.: Simultaneous localisation and map building for autonomous guided vehicles. In: *Intelligent Robots and Systems '94. 'Advanced Robotic Systems and the Real World', IROS '94. Proceedings of the IEEE/RSJ/GI International Conference on*. Volume 2. (sep 1994) 761-768.
- [Bosse03] M. Bosse, P. Newman, J. Leonard, M. Soika, W. Feiten, and S. Teller. An atlas framework for scalable mapping. In *IEEE International Conference on Robotics and Automation*, volume 2, pages 1899-1906, 2003.
- [Bosse04] M. Bosse, P. Newman, J. Leonard, and S. Teller. Simultaneous localization and map building in large-scale cyclic environments using the Atlas framework. *The International Journal of Robotics Research*, 23(12):1113, 2004.
- [Bosse07] M. Bosse and J. Roberts. Histogram matching and global initialization for laser-only SLAM in large unstructured environments. In *2007 IEEE International Conference on Robotics and Automation*, pages 4820-4826, 2007.
- [Burgard01] Burgard, W., Ruhnke, M., Stachniss, C., and Grisetti, G. (2001). The rawseeds project. <http://www.rawseeds.org/home/>
- [Censi08] A. Censi. An ICP variant using a point-to-line metric. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Pasadena, CA, May 2008.
- [Chen07] Z. Chen, J. Samarabandu, and R. Rodrigo, "Recent advances in simultaneous localization and map-building using computer vision", *Advanced Robotics*, Vol. 21, No. 3-4, pp. 233-265 (2007).
- [Chetverikov02] D. Chetverikov, D. Svirko, D. Stepanov, and P. Krsek. The trimmed iterative closest point algorithm. In *International Conference on Pattern Recognition*, volume 16, pages

545-548. Citeseer, 2002.

[Chown95] E. Chown, S. Kaplan and D. Kortenkamp, Prototypes, location, and associative networks (plan): Towards a unified theory of cognitive mapping, *Cognitive Sci.* 19, 1–51 (1995).

[Cox01] Cox, I.: Blanche: an experiment in guidance and navigation of an autonomous robot vehicle. *Robotics and Automation, IEEE Transactions on* 7(2) (April 1991)193-204

[Cuadra11a] Cuadra Troncoso, J.M., Alvarez-Sanchez, J.R., Paz Lopez, F., Fernandez-Caballero, A.: Improving area center robot navigation using a novel range scan segmentation method. In Fernandez, J.M., Alvarez Sanchez, J.R., Paz, F., Toledo, F., eds.: *Foundations on Natural and Artificial Computation*. Volume 6686 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg (2011) 233-245.

[Cuadra11b] Cuadra Troncoso, José Manuel, *Modelado adaptativo del medio para la navegación de robot autónomos utilizando algoritmos basados en el centro de áreas*, Tesis doctoral, UNED, 2011.

[Davison03] A. Davison, Real-time simultaneous localisation and mapping with a single camera, in: *Proc. Int. Conf. on Computer Vision, Nice*, pp. 1403–1416 (2003).

[DiMarco01] M. Di-Marco, A. Garulli, S. Lacroix and A. Vicino, Set membership localization and mapping for autonomous navigation, *Int. J. Robust Nonlinear Control* 11, 709–734 (2001).

[DiMarco03] M. Di-Marco, A. Garulli, A. Giannitrapani and A. Vicino, Simultaneous localization and map building for a team of cooperating robots: a set membership approach, *IEEE Trans. Robotics Automat.* 19, 238–249 (2003).

[Diosi05] A. Diosi and L. Kleeman. Laser scan matching in polar coordinates with application to SLAM. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3317-3322, 2005.

[Diosi07a] Diosi, A., Kleeman, L.: Fast laser scan matching using polar coordinates. *Int. J. Rob. Res.* 26(10) (oct 2007) 1125-1153.

[Diosi07b] A. Diosi and L. Kleeman. Fast laser scan matching using polar coordinates. *The International Journal of Robotics Research*, 26(10):1125, 2007.

[Fox99] D. Fox, W. Burgard, and S. Thrun, “Markov localization for mobile robots in dynamic environments,” *Journal of Artificial Intelligence Research*, vol. 11, pp. 391–427, 1999.

[Gutmann00] J.S. Gutmann and K. Konolige. Incremental mapping of large cyclic environments. In *Proceedings of the IEEE international symposium on computational intelligence in robotics and automation (CIRA)*. Citeseer, 2000.

- [Gutmann02] J.-S. Gutmann and D. Fox, "An experimental comparison of localization methods continued," in IEEE International conference on Intelligent Robots and Systems, vol. 1, 2002, pp. 454–459.
- [Grisetti08a] G. Grisetti, D.L. Rizzini, C. Stachniss, E. Olson, and W. Burgard. Online constraint network optimization for efficient maximum likelihood map learning. In Proceedings of the IEEE International Conference on Robotics and Automation, volume 1, 2008.
- [Grisetti08b] G. Grisetti, C. Stachniss, and W. Burgard. Non-linear constraint network optimization for efficient map learning. IEEE Transaction on Intelligent Transportation Systems, 2008.
- [Hahnel03] D. Hähnel, W. Burgard, B. Wegbreit and S. Thrun, Towards lazy data association in SLAM, in: Proc. 11th Int. Symp. of Robotics Research, Sienna, pp. 421–431 (2003).
- [Heinemann06] P. Heinemann, J. Haase, and A. Zell, "A combined monte-carlo localization and tracking algorithm for robocup," in IEEE International conference on Intelligent Robots and Systems, October 2006, pp. 1535–1540.
- [Howard02] A. Howard, M. Mataric, and G. Sukhatme, "Localization for mobile robot teams using maximum likelihood estimation," in IEEE International Conference on Intelligent Robots and Systems, 2002, pp. 434–459.
- [Hygounenc04] E. Hygounenc, I. K. Jung, P. Soueres and S. Lacroix, The autonomous blimp project of LAAS-CNRS: achievements in flight control and terrain mapping, Int. J. Robotics Res. 23 00–00 (2004).
- [Iser08] R. Iser and F.M. Wahl. Building local metrical and global topological maps using efficient scan matching approaches. In IEEE/RSJ International Conference on Intelligent Robots and Systems, 2008. IROS 2008, pages 1023-1030, 2008.
- [Johnson] Johnson, S.G.: The nlopt nonlinear-optimization package, <http://ab-initio.mit.edu/nlopt>
- [Julier03] S. Julier and J. Uhlmann, "Using multiple SLAM algorithms," in IEEE International Conference on Intelligent Robots and Systems, vol. 1, October 2003, pp. 200–205.
- [Kaelo06] P. Kaelo and M. M. Ali, "Some variants of the controlled random search algorithm for global optimization," J. Optim. Theory Appl. 130 (2), 253-264 (2006).
- [Kaess05] M. Kaess and F. Dellaert, A Markov chain Monte Carlo approach to closing the loop in: SLAM, in: Proc. IEEE Int. Conf. on Robotics and Automation, Barcelona, pp. 643–648 (2005).
- [Kim04] J. Kim and S. Sukkarieh, Autonomous airborne navigation in unknown terrain

environments, *IEEE Trans. Aerospace Electron. Syst.* 40, 1031–1045, July 2004.

[Kleeman95] L. Kleeman and R. Kuc, Mobile robot sonar for target localization and classification, *Int. J. Robotics Res.* 14, 295–318 (1995).

[Kleeman99] L. Kleeman, Real time mobile robot sonar with interference rejection, *Sonar Rev.* 19, 214–221 (1999).

[Konolige99] K. Konolige and K. Chou. Markov localization using correlation. In *International Joint Conference on Artificial Intelligence*, volume 16, pages 1154-1159. Citeseer, 1999.

[Kuipers91] B. J. Kuipers and Y. T. Byun, A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations, *Int. J. Robotics Res.* 8, 47–63 (1991).

[Kwok04] N.W. Kwok and G. Dissanayake, An efficient multiple hypothesis filter for bearing-only SLAM, in: *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, Sendai*, pp. 736–741 (2004).

[Leonard91] Leonard, J.J., Durrant-Whyte, H.F.: Simultaneous map building and localization for an autonomous mobile robot. In: *Proc. IEEE Int. Workshop on Intelligent Robots and Systems, Osaka, Japan* (1991) 1442-1447.

[Lingemann04] Lingemann, K., Surmann, H., Nuchter, A., Hertzberg, J.: Indoor and outdoor localization for fast mobile robots. In: *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on. Volume 3.* pp. 2185-2190.

[Lu94] Lu, F., Milios, E.: Robot pose estimation in unknown environments by matching 2d range scans. In: *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94., 1994 IEEE Computer Society Conference on. (jun 1994)* 935-938.

[Lu97] F. Lu and E. Milios. Robot pose estimation in unknown environments by matching 2d range scans. *Journal of Intelligent and Robotic Systems*, 18(3):249-275, 1997.

[Madhavan04] R. Madhavan, K. Fregene, and L. Parker, “Distributed cooperative outdoor multirobot localization and mapping,” *Autonomous Robots*, vol. 17, no. 1, pp. 23–39, July 2004.

[Metropolis49] N. Metropolis and S. Ulam, “The Monte Carlo method,” *Journal of American Statistical Association*, vol. 44, no. 247, pp. 335–341, 1949.

[Milanese96] M. Milanese, J. P. Norton, H. Piet-Lahanier and E. Walter, *Bounding Approaches to System Identification*. Plenum Press, New York (1996).

[Minguez05] J. Minguez, F. Lamiroux, and L. Montesano. Metric-based scan matching algorithms for mobile robot displacement estimation. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 4, page 3557. Citeseer, 2005.

- [Minguez06] J. Minguez, L. Montesano, and F. Lamiroux. Metric-based iterative closest point scan matching for sensor displacement estimation. *IEEE Transactions on Robotics*, 22(5):1047-1054, 2006.
- [Montemerlo02] M. Montemerlo, S. Thrun, D. Koller and B. Wegbreit, FastSLAM: A factored solution to the simultaneous localization and mapping problem, in: *Proc. AAAI Nat. Conf. on Artificial Intelligence*, Edmonton, pp. 593–598 (2002).
- [Montesano05] L. Montesano, J. Gaspar, J. Santos-Victor, and L. Montano, “Cooperative localization by fusing vision-based bearing measurements and motion,” in *IEEE International Conference on Intelligent Robots and Systems*, August 2005, pp. 2333–2338.
- [Moravec85] H. Moravec and A. Elfes, High resolution maps from wide angle sonar, in: *Proc. IEEE Int. Conf. on Robotics and Automation*, St. Louis, MO, pp. 116–121 (1985).
- [Mourikis06] A. Mourikis and S. Roumeliotis, “Performance analysis of multirobot cooperative localization,” *IEEE Transaction on Robotics*, vol. 22, no. 4, pp. 666–681, August 2006.
- [Neira01] J. Neira and J. D. Tardos, Data association in stochastic mapping using the joint compatibility test, *IEEE Trans. Robotics Automat.* 17, 890–897 (2001).
- [Nieto05] J. Nieto, T. Bailey, and E. Nebot. Scan-slam: Combining ekf-slam and scan correlation. In *International Conference on Field and Service Robotics (FSR)*. Springer, 2005.
- [Nieto07] J. Nieto, T. Bailey, and E. Nebot. Recursive scan-matching SLAM. *Robotics and Autonomous Systems*, 55(1):39-49, 2007.
- [Nocedal99] Nocedal, J., Wright, S.J.: *Numerical Optimization*. Springer (1999).
- [Nokia11] Nokia (2011). Qt. <http://qt.nokia.com/>
- [Olson09] E.B. Olson. Real-time correlative scan matching, *IEEE International Conference on Robotics and Automation, ICRA '09*, 4387 - 4393.
- [Peña85] Peña, D., Sobre la robustificación interna del algoritmo de Plackett-Kalman para la estimación recursiva del modelo de regresión lineal. *Trabajos de Estadística y de Investigación Operativa*, 36(1), 93-106, 1985.
- [Price83] W. L. Price, "Global optimization by controlled random search," *J. Optim. Theory Appl.* 40 (3), p. 333-348 (1983).
- [Rekleitis02] I. M. Rekleitis, G. Dudek, and E. E. Milios, “Multi-robot cooperative localization: A study of trade-offs between efficiency and accuracy,” in *IEEE International Conference on Intelligent Robots and Systems*, 2002, pp. 2690–2695.

- [Roumeliotis00] S. Roumeliotis and G. Bekey, "Bayesian estimation and Kalman filtering: A unified framework for mobile robot localization," in IEEE International Conference on Robotics and Automation, vol. 3, 2000, pp. 2985–2992.
- [Roumeliotis02] S. Roumeliotis and G. Bekey, "Distributed multirobot localization," IEEE Transactions on Robotics and Automation, vol. 18, no. 5, pp. 781–795, 2002.
- [Rusinkiewicz01] S. Rusinkiewicz and M. Levoy. Efficient variants of the ICP algorithm. In Proceedings of the Third Intl. Conf. on 3D Digital Imaging and Modeling, pages 145-152, 2001.
- [Se05] S. Se, D. G. Lowe and J. J. Little, Vision-based global localization and mapping for mobile robots, IEEE Trans. Robotics 21, 364–375 (2005).
- [Siciliano08] B. Siciliano and O. Khatib, Eds., Springer Handbook of Robotics. Springer, 2008.
- [Smith86] R. Smith and P. Cheeseman, On the representation of spatial uncertainty, Int. J. Robotics Res. 5, 56–68 (1986). Advances in SLAM using computer vision 261.
- [Smith90] R. Smith, M. Self and P. Cheeseman, Estimating uncertain spatial relationships in robotics, in: Autonomous Robot Vehicles (I. J. Cox and G. T. Wilfong, Eds), pp. 167–198. Springer, New York, NY (1990).
- [Solà05] J. Solà, A. Monin, M. Devy and T. Lemaire, Undelayed initialization in bearing only SLAM, in: Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, Edmonton, pp. 2499–2504 (2005).
- [Stroupe00] A. W. Stroupe, M. C. Martin, and T. Balch, "Merging gaussian distributions for object localization in multi-robot systems," in Seventh International Symposium on Experimental Robotics. Springer-Verlag, 2000.
- [Sujan04] V. A. Sujan, S. Dubowsky, T. Huntsberger, H. Aghazarian, Y. Cheng and P. Schenker, An architecture for distributed environment sensing with application to robotic cliff exploration, Autonomous Robots 16, 287–311 (2004).
- [Sujan05] V. A. Sujan and S. Dubowsky, Efficient information-based visual robotic mapping in unstructured environments, Int. J. Robotics Res. 24, 275–293 (2005).
- [Thrun00] S. Thrun, W. Burgard, and D. Fox. A real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping. In IEEE international conference on robotics and automation (ICRA), volume 1, pages 321-328, 2000.
- [Thrun01] S. Thrun, "A probabilistic online mapping algorithm for teams of mobile robots," International Journal of Robotics Research, vol. 20, no. 5, pp. 335–378, 2001.
- [Thrun02] S. Thrun, Robotic mapping: a survey, in: Exploring Artificial Intelligence in the New Millenium (G. Lakemeyer and B. Nebel, Eds), pp. 1–35. Morgan Kaufmann, San Mateo, CA (2002).

- [Thrun05] S. Thrun, W. Burgard, and D. Fox, Probabilistic Robotics. The MIT Press, 2005.
- [Tomono04] Tomono, M.: A scan matching method using euclidean invariant signature for global localization and map building. In: Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on. Volume 1. (2004) 866-871
- [Tungadi07] F. Tungadi and L. Kleeman. Multiple laser polar scan matching with application to SLAM. In Proceedings of the Australasian Conference on Robotics and Automation, Brisbane, Australia, 2007.
- [Wan00] E. Wan and R. V. D. Merwe, "The unscented Kalman filter for nonlinear estimation," in IEEE Adaptive Systems for Signal Processing, Communications, and Control Symposium, 2000, pp. 153-158.
- [Wang05] Z. Wang, S. Huang and G. Dissanayake, Decoupling localization and mapping in SLAM using compact relative maps, in: Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2005), Edmonton, AB, August 2–6, pp. 3336–3341 (2005).
- [Weiss95] Weiss, G., Puttkamer, E.: A map based on laserscans without geometric interpretation. In: In proceeding of: Intelligent Autonomous Systems 4 (IAS-4), IOS Press(1995) 403-407.
- [Witkin84] Witkin, A. Scale-space filtering: A new approach to multi-scale description. In Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '84., volume 9, pages 150-153.
- [Wolter04] D. Wolter, L. Latecki, R. Lakamper and X. Sun, Shape-based robot mapping, Adv. Artif. Intell. (LNCS) 3238, 439–452 (2004).
- [ZeSu07] C. Ze-Su, H. Bing-Rong, and L. Hong. An Improved Polar Scan Matching Using Genetic Algorithm. Information Technology Journal, 6(1):89-95, 2007.
- [Zhang94] Z. Zhang. Iterative point matching for registration of free-form curves and surfaces. International Journal of Computer Vision, 13(2):152, 1994.

9 Apéndice: Ejemplos de mapas generados

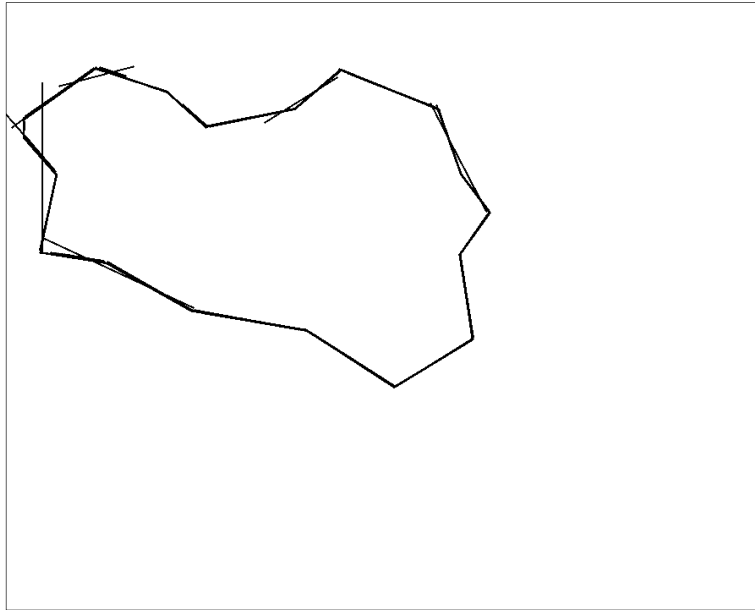


Figura 52: Mapa poligonal generado (sin ruido)

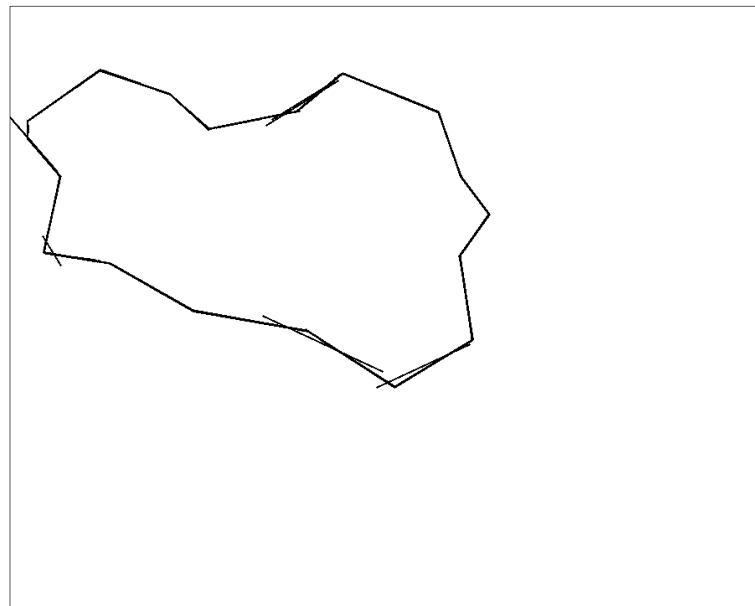


Figura 53: Mapa poligonal generado (ruido odométrico de 1%)

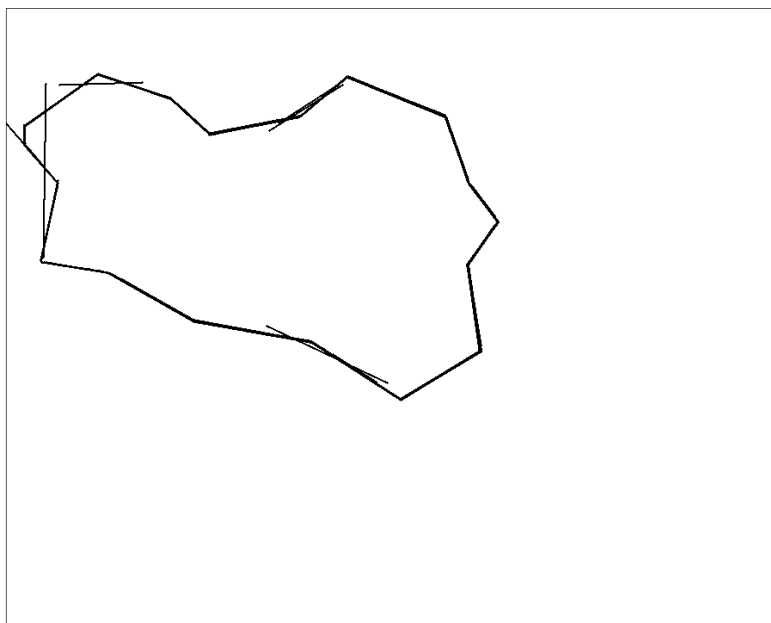


Figura 54: Mapa generado (ruido odométrico de 5%)

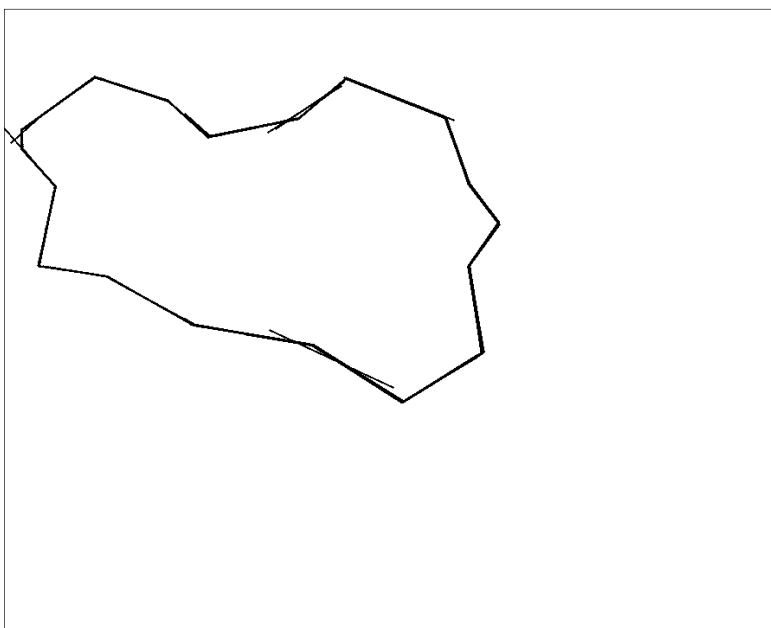


Figura 55: Mapa generado (ruido odométrico de 10%)

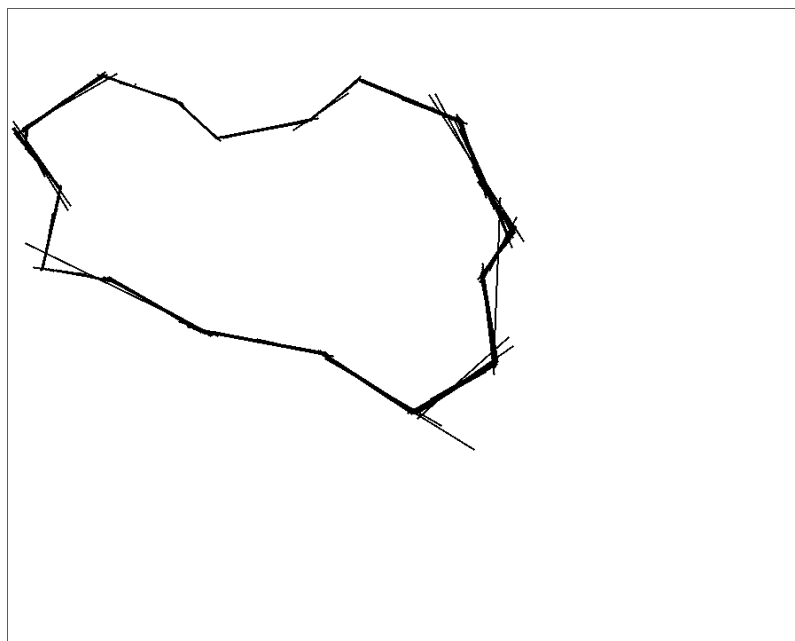


Figura 56: Mapa generado (ruido en lecturas láser con desviación estándar de 5mm)

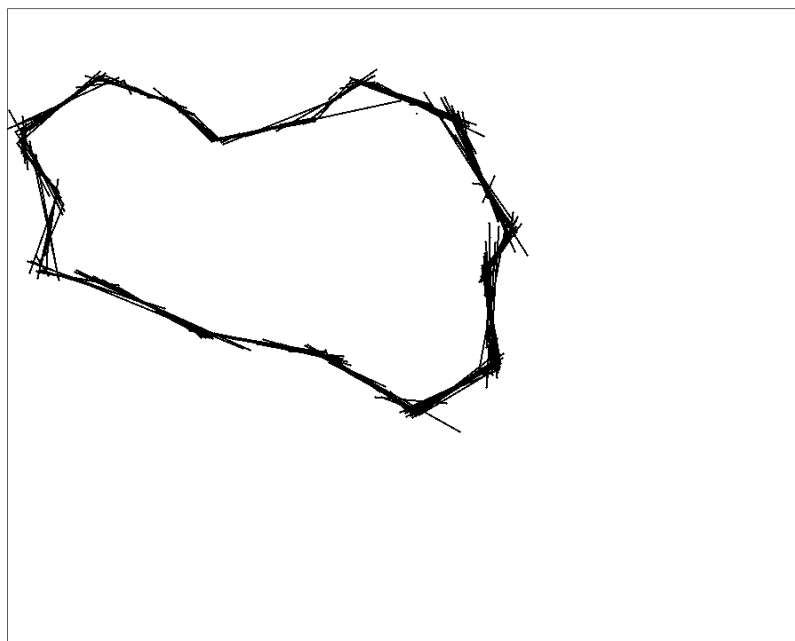


Figura 57: Mapa generado (ruido en lecturas láser con desviación estándar de 20mm)

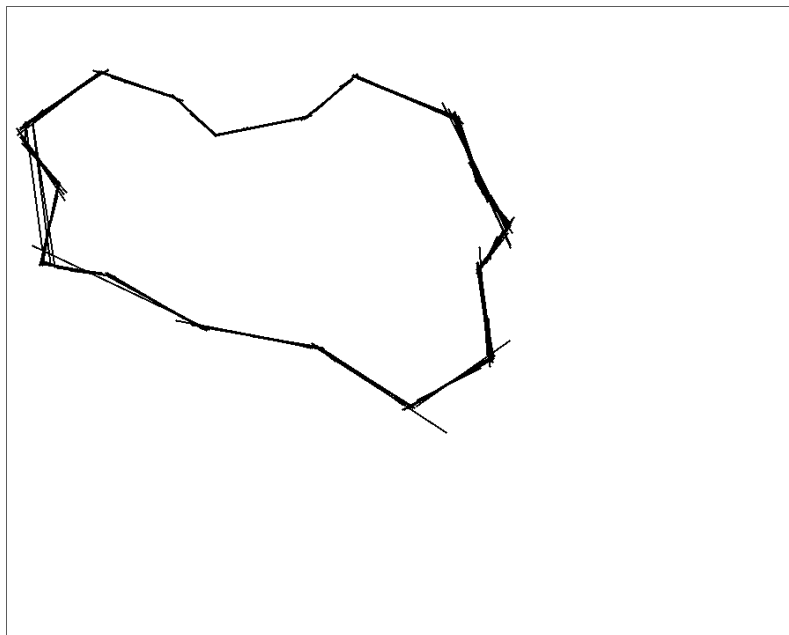


Figura 58: Mapa generado (ruido 5% + 5mm)

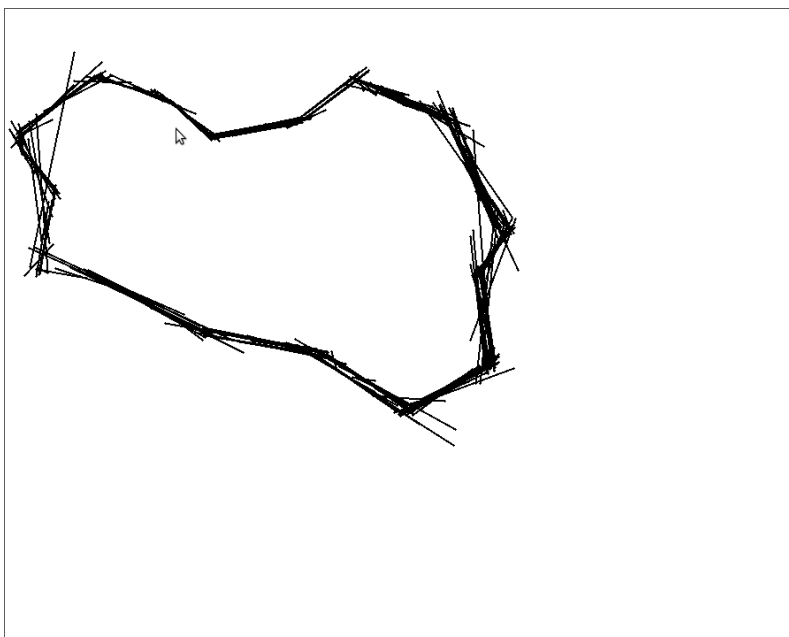


Figura 59: Mapa generado (ruido 10% + 20mm)