



PONTIFICIA UNIVERSIDAD CATOLICA DE CHILE  
ESCUELA DE INGENIERIA

# **DISEÑO Y OPTIMIZACIÓN DE LANDMARKS PARA LOCALIZACIÓN EN TÚNELES**

**FABIAN TOMAS PAREDES LIZAMA**

Tesis presentada a la Dirección de Investigación y Postgrado  
como parte de los requisitos para optar al grado de  
Magister en Ciencias de la Ingeniería

Profesor Supervisor:  
MIGUEL TORRES TORRITI

Santiago de Chile, abril 2017

© MMXVII, FABIAN TOMAS PAREDES LIZAMA



PONTIFICIA UNIVERSIDAD CATOLICA DE CHILE  
ESCUELA DE INGENIERIA

# **DISEÑO Y OPTIMIZACIÓN DE LANDMARKS PARA LOCALIZACIÓN EN TÚNELES**

**FABIAN TOMAS PAREDES LIZAMA**

Miembros del Comité:

MIGUEL TORRES TORRITI

ALVARO VIDELA

FERNANDO AUAT

MIGUEL RÍOS OJEDA

Tesis presentada a la Dirección de Investigación y Postgrado  
como parte de los requisitos para optar al grado de  
Magister en Ciencias de la Ingeniería

Santiago de Chile, abril 2017

© MMXVII, FABIAN TOMAS PAREDES LIZAMA

*A mis seres queridos y a todos  
quienes me han ayudado.*

## **AGRADECIMIENTOS**

Agradezco a mi familia, quienes me han brindado su completo apoyo en cada paso que he dado en este mundo. Me permitieron llegar hasta esta instancia gracias a su esfuerzo y sacrificio por brindarme una mejor vida.

A Andrea por su incomparable ayuda en este trabajo y por acompañarme en mis momentos más oscuros. No lo hubiese podido lograr sin su constante apoyo y amor.

A Miguel por su guía y enseñanza durante gran parte de mi formación profesional. Agradezco las oportunidades que me brindó para desarrollarme y aprender, junto con la confianza que esto implica.

Al grupo RAL, en especial a Daniel, Nicolás y Felipe, y a Joaquín, por colaborar en este desafiante proyecto, ser parte íntegra de mi proceso de aprendizaje y por su gran amistad a lo largo de todos estos años. Espero haber sido de tanta ayuda como lo fueron para mi.

A Jorge, por su gran amistad y su apoyo en no sólo este trabajo, sino que en distintas facetas de la vida. También agradezco su infinita disposición y paciencia al lidiar con cada nuevo desafío.

A los miembros de la comisión por participar de este proceso y su enorme ayuda en mejorar la calidad de este trabajo, junto con el tiempo y voluntad brindados a mí y este trabajo.

Y a esas incontables personas que conocí durante mi vida y que han de alguna forma moldeado quien soy el día de hoy.

## INDICE GENERAL

AGRADECIMIENTOS . . . . .	IV
INDICE DE FIGURAS . . . . .	VIII
INDICE DE TABLAS . . . . .	XI
RESUMEN . . . . .	XII
ABSTRACT . . . . .	XIV
1. INTRODUCCION . . . . .	1
1.1. Motivación . . . . .	1
1.2. Descripción del Problema . . . . .	4
1.3. Objetivos . . . . .	5
1.4. Hipótesis . . . . .	5
1.5. Alcances . . . . .	6
1.6. Contribuciones . . . . .	6
1.7. Contenido . . . . .	7
2. SOLUCIONES EXISTENTES . . . . .	8
2.1. En túneles mineros . . . . .	8
2.2. En ambientes similares . . . . .	10
2.3. Scan-Matching . . . . .	11
3. ESTRUCTURA DE LA SOLUCIÓN PROPUESTA . . . . .	12
3.1. Estimador de Posición . . . . .	13
3.2. Diseño de Landmark . . . . .	15
4. ALGORITMOS DE ESTIMACIÓN DE POSICIÓN . . . . .	16
4.1. RANSAC . . . . .	16
4.2. Asignación del Landmark . . . . .	22

4.3.	ICP . . . . .	24
4.4.	Distancia de Hausdorff . . . . .	25
4.5.	Corrección de Medición . . . . .	31
5.	MODELACIÓN DEL ENTORNO . . . . .	33
5.1.	Características generales . . . . .	33
5.2.	Modelación del Túnel . . . . .	33
5.3.	Modelación de LIDAR . . . . .	33
5.4.	Modelación del cargador frontal . . . . .	35
6.	ALGORITMO DE LANDMARK ÓPTIMO . . . . .	36
6.1.	Algoritmo Genético . . . . .	36
6.1.1.	Genomas . . . . .	37
6.2.	Función Objetivo . . . . .	42
6.2.1.	Error de Posición Promedio . . . . .	42
6.3.	Parámetros Utilizados . . . . .	42
6.4.	Resultados y Análisis . . . . .	43
6.4.1.	Progresión de la Población . . . . .	43
6.4.2.	Resultados Óptimos . . . . .	48
7.	IMPLEMENTACIÓN EXPERIMENTAL . . . . .	59
7.1.	Experimentos a Realizar . . . . .	59
7.2.	Plataforma de Prueba . . . . .	61
7.3.	Construcción del Escenario . . . . .	63
8.	RESULTADOS EXPERIMENTALES . . . . .	66
8.1.	Resultados . . . . .	66
8.2.	Análisis . . . . .	77
8.3.	Reconstrucción de mapa y trayectoria . . . . .	77
8.3.1.	Filtro de Kalman . . . . .	77
8.3.2.	Reconstrucción de mapa . . . . .	78

8.3.3. Reconstrucción del camino . . . . .	81
9. CONCLUSIONES . . . . .	84
9.1. Ventajas y desventajas comparativas . . . . .	84
9.2. Resultados y comentarios finales . . . . .	85
9.3. Temas de investigación futura . . . . .	86
BIBLIOGRAFIA . . . . .	88
ANEXO A. RECURSOS ADICIONALES . . . . .	96
A.1. Notación de Variables . . . . .	96
A.2. Documentación Simulador . . . . .	98
A.3. Documentación Experimentación . . . . .	100
A.4. PCB de control . . . . .	102

## INDICE DE FIGURAS

1.1. Operarios en la división El Teniente de Codelco, Chile. . . . .	1
1.2. Numero de fallecidos de empresas mineras, mandantes o contratistas de categorías A, B, C y D. . . . .	2
1.3. Distribución de trabajadores fallecidos por tipo de instalación, desde 2012 hasta el primer trimestre 2016. . . . .	2
1.4. LHD (Load Haul Dump) manejado mediante control remoto por operario de Atlas Copco. . . . .	3
1.5. Operación remota de bulldozers Caterpillar en minas de carbón. . . . .	4
3.1. Posición real simulada del vehículo y su estimación de posición mediante el algoritmo. . . . .	12
3.2. Diagrama de operación del LIDAR con respecto al vehículo y al túnel. . . . .	13
3.3. Ejemplo de integración del estimador de posición con un filtro de Kalman. . . . .	14
4.1. Flujo del algoritmo de estimación por muestra recibida de LIDAR. . . . .	16
4.2. Iteración de RANSAC incorrecta que no cumple con los parámetros especificados. . . . .	19
4.3. Iteración de RANSAC final que cumple con los parámetros especificados. . . . .	20
4.4. Ejemplo de asignación de puntos a Landmarks. . . . .	22
4.5. Resultado del algoritmo de Hausdorff con un landmark de prueba y selección de distancia al 100% del máximo. . . . .	27
4.6. Resultado del algoritmo de Hausdorff con un landmark de prueba y selección de distancia al 80% del máximo. . . . .	27
4.7. Mapa de distancias mínimas calculado para un landmark de prueba. . . . .	28
4.8. Ejemplo de traslación con y sin corrección por pérdida de landmarks. . . . .	31
5.1. Túnel simulado de muestra. . . . .	34

6.1. Ejemplos de genomas de forma primitiva. . . . .	38
6.2. Ejemplos de formas para figuras primitivas. . . . .	39
6.3. Ejemplos de genomas de forma no determinada no reflejados y no simétricos. . . . .	40
6.4. Ejemplos de <i>landmarks</i> de forma no determinada de igual genoma. . . . .	41
6.5. Progresión del promedio de error de posición promedio por generación. . . . .	44
6.6. Progresión del promedio de error de posición promedio por generación utilizando miembros con desempeño < 1m. . . . .	45
6.7. Progresión de la traza de la matriz de covarianza del genoma por generación. . . . .	46
6.8. Progresión de la traza de la matriz de covarianza del genoma por generación utilizando miembros con desempeño < 1m. . . . .	47
6.9. Histograma de la población según generación y método para el gen n°1 - Factor de escalamiento. . . . .	49
6.10. Histograma de la población según generación y método para el gen n°2 - Ancho de la forma. . . . .	50
6.11. Histograma de la población según generación y método para el gen n°3 - Distancia entre Landmarks. . . . .	51
6.12. Histograma de la población según generación y método para el gen n°7 - Factor de escalamiento. . . . .	52
6.13. Histograma de la población según generación y método para el gen n°8 - Ancho de la forma. . . . .	53
6.14. Histograma de la población según generación y método para el gen n°9 - Distancia entre Landmarks. . . . .	54
6.15. Formas normalizadas de la última generación obtenida con ICP. . . . .	55
6.16. Formas normalizadas de la última generación obtenida con Hausdorff. . . . .	56
6.17. Landmarks de la última generación obtenida con ICP. . . . .	57
6.18. Landmarks de la última generación obtenida con Hausdorff. . . . .	57

7.1. Segmento temporal de toma de muestras para sensores, experimento 2 repetición 1. . . . .	59
7.2. Segmento de estimación de posición de toma de muestras para sensores, experimento 2 repetición 1. . . . .	60
7.3. Formas construidas para validar experimentalmente. . . . .	61
7.4. Plataforma de prueba CAT262C sin modificaciones. . . . .	62
7.5. Plataforma de prueba CAT262C con el sensor SICK LMS511 instalado. . . . .	63
7.6. Instalación de túnel sin landmarks. . . . .	64
7.7. Instalación de landmarks en túnel. . . . .	65
8.1. Muestra de datos obtenidos por LIDAR en una repetición del experimento 2.	66
8.2. Error $E_p$ para las repeticiones del experimento 1. . . . .	70
8.3. Error $E_p$ para las repeticiones del experimento 2. . . . .	71
8.4. Error $E_p$ para las repeticiones del experimento 3. . . . .	72
8.5. Error promedio de la posición X. . . . .	73
8.6. Error promedio de la posición Y. . . . .	74
8.7. Progresión de la covarianza de la posición X. . . . .	75
8.8. Progresión de la covarianza de la posición Y. . . . .	76
8.9. Reconstrucción del túnel, experimento nº2 repetición 4. . . . .	79
8.10. Reconstrucción del túnel, experimento nº3 repetición 4. . . . .	80
8.11. Reconstrucción de la trayectoria, experimento nº1 repetición 3. . . . .	81
8.12. Reconstrucción de la trayectoria, experimento nº2 repetición 3. . . . .	82
8.13. Reconstrucción de la trayectoria, experimento nº3 repetición 3. . . . .	83
A.1. Vista total del diseño de PCB Masterboard. . . . .	103

## INDICE DE TABLAS

6.1. Descripción del genoma para las figuras con forma primitiva. . . . .	37
6.2. Descripción del genoma para las figuras con forma no determinada. . . . .	39
6.3. Rango de valores para genomas de simulación. . . . .	43
6.4. Error de posición promedio para el individuo promedio óptimo en cada caso.	48
7.1. Genomas de Landmarks utilizados para los experimentos. . . . .	61
8.1. Resultados experimentales - Error de posición promedio. . . . .	66
8.2. Resultados Experimento 1 por repetición - Error de posición promedio (m).	67
8.3. Resultados Experimento 2 por repetición - Error de posición promedio (m).	68
8.4. Resultados Experimento 3 por repetición - Error de posición promedio (m).	69
8.5. Tabla de datos promedio para la reconstrucción del túnel. . . . .	80

## RESUMEN

La minería es una de las actividades económicas más importantes del país, y es de interés general automatizar sus distintos procesos para mejorar la seguridad de los trabajadores, aumentar la productividad y reducir los costos de mantención de los vehículos utilizados. Es de especial interés automatizar los procesos de explotación de mineral de galerías subterráneas donde se han implementado distintos procesos parcialmente automatizados mejorando la seguridad de los operadores y reduciendo el desgaste de los vehículos. Uno de los problemas a resolver para lograr la automatización total es localizar al vehículo de forma correcta dentro del escenario de operación para lo cual se han desarrollado numerosos métodos. Sin embargo, no se ha logrado encontrar un buen método de estimación de la localización cuando se recorren túneles rectos homogéneos ya que los métodos convencionales utilizan una infraestructura activa que requiere mantención o requieren de características distintivas del entorno.

En este trabajo se implementa un método para encontrar *landmarks* pasivos volumétricos optimizados y con los parámetros de diseño que mejoran su rendimiento en algoritmos de estimación de posición, junto con dos algoritmos de estimación de la posición relativa del vehículo basados en mediciones de LIDAR de *landmarks* pasivos volumétricos colocados con anterioridad a lo largo de un túnel recto y homogéneo combinado con sus señales de control.

Los algoritmos de estimación se basan en localizar los *landmarks* a lo largo del túnel con técnicas de *matching* de modelos a las mediciones, donde ambos algoritmos se diferencian en la utilización de ICP (*Iterative Closest Point*) con distancia euclidiana punto a punto o una transformaciones rígidas iterativas con distancia de Hausdorff para realizar este calce. Se pudo calcular que los algoritmos exitosamente permiten estimar la posición dependiendo de los *landmarks* utilizados.

Los *landmarks* optimizados se encontraron mediante un proceso de optimización genética para el cual se utilizaron dos genomas, uno que asume una forma básica y otro que define su forma en sus genes. Con los resultados de las simulaciones se pudo obtener que la forma no altera el desempeño del algoritmo, mientras que sus dimensiones y el espacio entre ellos determina el desempeño en cada caso.

El algoritmo de localización relativa fue evaluado en una plataforma *skid-steer* industrial adaptada para una operación semiautónoma en un escenario artificial que replica las condiciones sensoriales encontradas en un túnel minero. De esta manera se pudo validar exitosamente el desempeño de los *landmarks* optimizados.

**Palabras Claves:** localización, estimación de posición, scan matching, túneles subterráneos.

## ABSTRACT

Mining is one of the most important economic activities of the country and there is a generalized interest to automate its various processes to improve workers security, enhance productivity and reduce machineries maintenance costs. It is of special interest to automate the mining processes of underground galleries where numerous partially automated processes have been implemented improving operators security and reducing the wear and tear of the vehicles. One of the problems to solve in order to achieve full automation is to correctly locate the vehicle inside the operating scenario, where various methods have been developed. However a good method of estimating the location in homogeneous straight tunnels has not been found, conventional methods use an active infrastructure that requires maintenance, have a high error accumulating rate or require or rely on the environment's notable features.

In this work a method to find optimized passive volumetric landmarks and the design parameters that improve their performance within position estimation algorithms, along with the development of two vehicle relative position estimation algorithms based on LIDAR scans of passive volumetric landmarks, previously placed along the tunnel, and control signals.

The estimation algorithms are based on locating the landmarks along the tunnel with scan to model matching techniques, where both algorithms differ in their use of ICP (Iterative Closest Point) with euclidean point to point distance or iterative rigid transformations with Hausdorff distance at matching. It was possible to calculate that the algorithms successfully estimate the relative position depending on the landmarks used. Optimized landmarks were found by a genetic optimization process using two genomes, a genome that assumes a basic shape and another one that defines its shape within the genome. With the simulations results it becomes clear that the shape doesn't alter the performance of the estimator while the landmarks dimensions and the distance between them determinates the performance in each case.

The relative localization algorithm was evaluated in an industrial skid-steer platform adapted for semi automated operation in an artificial scenario that replicates the sensor conditions of an underground tunnel, validating the optimized landmarks' performance.

**Keywords:** localization, position estimation, scan matching, underground tunnels.

## 1. INTRODUCCION

### 1.1. Motivación

La minería es una de las actividades económicas más importantes del país, aportando el 9% del producto interno bruto de Chile en el año 2015 y 8,5% del empleo nacional (Consejo Minero, 2015) siendo gran parte de ésta desarrollada de forma subterránea (Area Minera, 2015). Las condiciones intrínsecas de operar maquinaria pesada bajo tierra son de alto riesgo debido a una multitud de factores, incluyendo: caída de rocas, explosiones, golpes y atrapamientos con herramientas, vehículos y maquinarias, lesiones por material particulado, asfixia o intoxicación en lugares mal ventilados, etc (National Institute for Occupational Safety and Health y Centers for Disease Control and Prevention, 2012, 2014, 2015). Éstos factores generan la necesidad de automatizar los procesos in situ para aumentar la calidad de vida de los trabajadores y disminuir el riesgo de accidentes. Durante la última década los fallecimientos promedian 29 accidentes por año, donde cerca del 48% de las muertes ocurridas corresponden a instalaciones subterráneas (Servicio Nacional de Geología y Minería, 2016).



FIGURA 1.1. Operarios en la división El Teniente de Codelco, Chile.

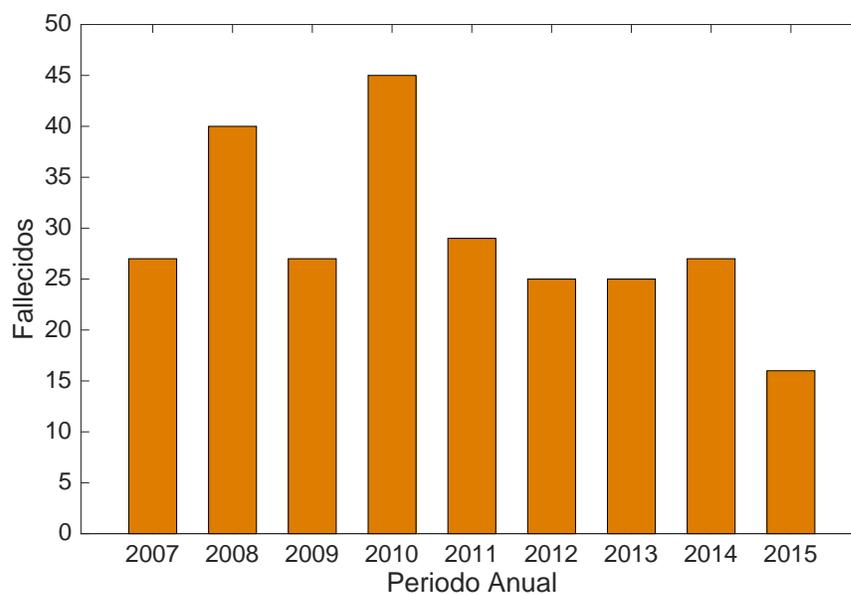


FIGURA 1.2. Numero de fallecidos de empresas mineras, mandantes o contratistas de categorías A, B, C y D.

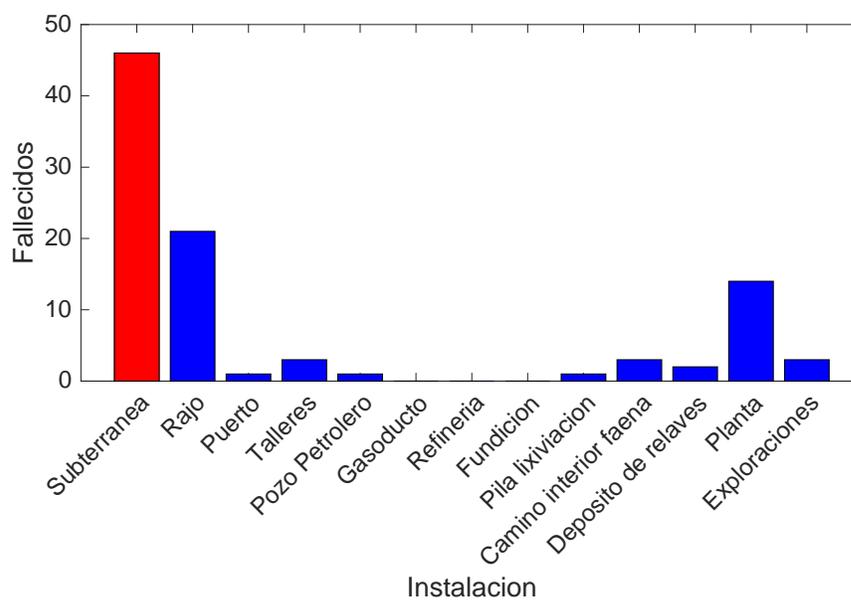


FIGURA 1.3. Distribución de trabajadores fallecidos por tipo de instalación, desde 2012 hasta el primer trimestre 2016.

Parte importante del proceso minero es la extracción de mineral desde las galerías de explotación subterráneas, lo que se hace mediante el uso de vehículos LHD (*Load Haul Dump vehicles*) en operaciones mayores o cargadores frontales en operaciones menores (Scheding, Dissanayake, Nebot, y Durrant-Whyte, 1999; Doherty, 2014), en ambos casos se han realizado varios esfuerzos por crear sistemas autónomos y semi-autónomos por empresas como Caterpillar, Sandvik y Atlas Copco (Gustafson, 2011). Sin embargo la operación autónoma esta lejos de ser alcanzada optándose por soluciones teleoperadas o semiautónomas las cuales son más factibles por la menor complejidad técnica y operacional que representan.

Las soluciones semiautónomas no necesariamente aumentan la tasa de producción



FIGURA 1.4. LHD (Load Haul Dump) manejado mediante control remoto por operario de Atlas Copco.

a menos que un operador maneje varios vehículos al mismo tiempo (Peters, Smith, Medeiros, y Rohrer, 2001), no obstante reducen el desgaste de los vehículos de carga debida a una utilización más óptima de la aceleración y marcha de los mismos, junto con cumplir con sacar al operador de la zona de riesgo (Larsson, 2011). Uno de los

casos más emblemáticos de automatización de operaciones subterráneas corresponde a la mina El Teniente, donde se pudo implementar teleoperación exitosamente, lo cual redujo la producción en un 20% (Díaz Copier y Carranza Meza, 1998).

Los resultados de las implementaciones de automatización en distintas minas son variados y dependen de múltiples factores (Gustafson, 2011) pero comparten los mismos desafíos, incluyendo la reducción de velocidad de movimiento por seguridad y la latencia en las señales de *feedback* (Larsson, Broxvall, y Saffiotti, 2010) evidenciando la necesidad de generar sistemas más confiables y automáticos.



FIGURA 1.5. Operación remota de bulldozers Caterpillar en minas de carbón.

## 1.2. Descripción del Problema

Para lograr una exitosa implementación de cualquier método autónomo o semi-autónomo es necesario resolver el problema de localización y navegación en túneles mineros para llevar el vehículo de extracción de un punto a otro de forma eficiente, segura y rápida. Esto debe hacerse dentro de los túneles subterráneos de extracción

de mineral los cuales tienen condiciones características que condicionan el problema. Gran parte del traslado ocurre en túneles angostos, largos y homogéneos. Esto dificulta la estimación de posición del vehículo debido a que los métodos sin infraestructura típicos para localizarse dentro del túnel (odometría y técnicas de SLAM utilizando LIDAR) tienen una alta tasa de acumulación de errores o dependen de características distintivas del entorno llamadas *features*, lo cual los hace inadecuado como solución a menos que se generen estas características de forma artificial con *landmarks* o estructuras distintivas detectables.

### 1.3. Objetivos

El objetivo de este trabajo es desarrollar e implementar dos métodos, uno contempla generar un diseño optimizado de *landmarks* artificiales, consistentes en perfiles volumétricos adosados a la pared del túnel, a utilizar por algoritmos de estimación de localización relativa, según las condiciones de un túnel sin *features* naturales, junto con un análisis de los parámetros que influyen en su diseño. El segundo método es el desarrollo e implementación de algoritmos capaces de estimar la localización relativa de un vehículo en las condiciones del escenario descritas anteriormente, utilizando las señales de control y sensores LIDAR junto con los *landmarks* mencionados anteriormente; solucionando los problemas presentes en los métodos de estimación de posición que convencionalmente no funcionan de manera correcta en largos pasillos sin *features*.

### 1.4. Hipótesis

Es posible implementar un estimador de posición para la localización de vehículos en ambientes sin características reconocibles como túneles rectos y homogéneos mediante el uso de *landmarks* artificiales y mediciones de LIDAR.

Asimismo es posible optimizar la geometría de los *landmarks* usando técnicas basadas en algoritmos genéticos para encontrar la forma que minimiza el error de localización.

### 1.5. Alcances

Este trabajo no considera dentro de sus alcances el desarrollo de un controlador de trayectorias ni la implementación de nuevas técnicas de SLAM, los cuales son materias ampliamente desarrolladas en la literatura existente, no así el desarrollo de mecanismos que doten al vehículo de una capacidad de percepción del entorno para su correcta localización en lugares como túneles mineros.

### 1.6. Contribuciones

Las contribuciones generadas en este trabajo se pueden resumir en las siguientes:

1. Un simulador implementado en MATLAB que replica las condiciones de un túnel subterráneo con *landmarks* artificiales junto con un vehículo *skid-steer* con sensores LIDAR. El simulador es capaz de calcular la posición y orientación del vehículo dada una trayectoria y velocidad definida, junto con aplicar algoritmos de localización y generar valores de desempeño.
2. Dos algoritmos estimadores de localización que ocupan el modelo de *landmark* utilizado junto con los sensores mencionados anteriormente para estimar la localización a lo largo del túnel.
3. Desarrollo de un método para generar *landmarks* a partir de un genoma y encontrar sistemáticamente soluciones mejores utilizando algoritmos de optimización genética y el simulador de localización implementado en MATLAB.

4. Validación experimental de los algoritmos de localización y de optimización en una plataforma industrial, lo cual muestra la aplicabilidad de este trabajo a condiciones reales.
5. Sistema captador de datos implementado en Python que permite obtener mediciones independientes y coordinadas de los sensores instalados en la plataforma de prueba (LIDAR, encoders, IMU, GPS) junto con controlar y capturar señales de control automatizadas.
6. Diseño e implementación del hardware electrónico para la plataforma de prueba, que permite modificar las señales de control del vehículo permitiendo la operación remota *line of sight* o la operación computarizada, permitiendo realizar experimentos con la repetibilidad requerida.

### **1.7. Contenido**

En la sección 3 se encuentra la estructura del estimador y en la sección 4 se detallan los algoritmos utilizados. En 5 se especifican las características del simulador de prueba, en 6 se muestra el método de optimización utilizado. La validación experimental se muestra en la sección 7 y los resultados junto con su análisis en la sección 8.

## 2. SOLUCIONES EXISTENTES

### 2.1. En túneles mineros

El estado del arte muestra que no se han implementado soluciones de detección de entorno en cargadores frontales *skid-steer* bajo tierra, sin embargo, se ha trabajado de gran manera en LHDs los cuales presentan problemas y desafíos muy similares a los que se necesitan resolver, por lo que las investigaciones sobre esta plataforma se pueden considerar como base para este trabajo.

Múltiples soluciones para ayudar a la detección y estimación de la posición han sido planteadas, donde la mayoría de ellas usa EKF (*Extended Kalman Filter*) para fusionar sensores de distinto tipo. Un sistema de ejemplo de este tipo para la navegación de LHDs (Scheding et al., 1999) consta de sensores inerciales, odométricos y ultrasónicos; además de un GCS (*Guidance Control System*) láser que permite saber la distancia a retroreflectores colocados en el túnel para comparar con un mapa previo. Sin embargo el GCS es muy lento (1 Hz) para esta aplicación además de requerir retroreflectores que resistan las condiciones dentro de los túneles y un mapa de sus ubicaciones.

En (Marshall, Barfoot, y Larsson, 2008) se muestra un método de navegación para túneles que no requiere *landmarks* artificiales ni un mapa global previo, sino que se basa en recorrer con un operador humano los túneles durante una fase de entrenamiento inicial. Lo anterior tiene como objetivo generar perfiles de navegación grabando los datos de los sensores y replicándolos de forma autónoma, donde los datos de LIDAR son utilizados para realizar *scan-matching* con un atlas de mapas locales; no obstante, esta solución no resuelve el problema de *scan-matching* sin *features* notables como ocurre en túneles largos.

Un problema similar que ha sido investigado extensivamente es guiar a un vehículo dentro del túnel, detectando cruces con otros túneles, galerías o puntos notables. Estos sistemas permiten que el vehículo se traslade desde un punto inicial a uno final a través de una ruta determinada, pero no entregan información sobre la posición en todo momento, sino que corrigen errores una vez alcanzados estos puntos de interés.

Esta estructura al no disponer de información actualizada para la posición o traslación no permite realizar controles predictivos y tienen que ser operados a menor velocidad (Larsson et al., 2010).

Un sistema de este tipo es presentado en (Hurteau, St-Amant, Laperriere, y Chevrette, 1992) donde se utiliza una banda reflectante a lo largo del techo del túnel para ser detectada por cámaras ópticas montadas en el vehículo, mientras que los puntos notables están marcados por modificaciones en la banda. Otros sistemas que operan de forma similar utilizan cables de seguimiento enterrados en concreto o filas de luces (Gustafson, 2011).

En el trabajo de (Larsson y Broxvall, 2008) se muestra un método para el cual, utilizando mediciones LIDAR, se detecta si es que la topología en frente del vehículo corresponde a una intersección o un túnel midiendo las discontinuidades entre los puntos consecutivos, permitiendo detectar cuando el vehículo sale de un sector determinado. De forma similar en (Roberts, Duff, y Corke, 2002) se muestra un sistema de localización oportunista, donde se usa un algoritmo seguidor de paredes con el sensor LIDAR para evitar colisiones, odometría para estimar la posición y, al llegar a una intersección, ambos sensores para determinar cuales son los caminos a seguir. En (J. Bakambu, Polotski, y Cohen, 2004; J. N. Bakambu y Polotski, 2007) se desarrolla un sistema muy similar a lo anterior, pero utilizando técnicas de *scan-matching* basadas en la estructura punto-a-linea para detectar el cruce a nivel global. Todos estos conceptos se refieren a navegación reactiva, donde se toman decisiones en puntos notables resolviendo el problema de la localización global al no tomarlo en cuenta, pero no permiten operaciones a alta velocidad al desconocer el panorama en el que se encuentran.

En (Haehnel, 2004) se muestran diversos ejemplos de mapeo y localización donde resultan relevantes los resultados concernientes a la localización con tags RFID, los que indican que la ubicación de tags a lo largo de un pasillo permite mejorar la localización cuando es usada en conjunto con odometría y detección láser, reduciendo al mismo tiempo la incertidumbre de la odometría y la cantidad de partículas detectadas por el

LIDAR necesarias para alcanzar un buen rendimiento.

Otro trabajo que usa RFID es (Larsson, Broxvall, y Saffiotti, 2005) el cual coloca tags RFID en los cruces importantes de una mina para corregir los errores del algoritmo de detección de cruces; éste no era capaz de distinguir a través de LIDAR áreas abiertas de los cruces, los cuales son corregidos y verificados con los tags.

## 2.2. En ambientes similares

Las aplicaciones de *scan-matching*, definidas como el conjunto de técnicas que busca encontrar la transformación rígida entre un escaneo del medio ambiente y un mapa o modelo, investigadas en la literatura se aplican en diversos escenarios, algunos de los cuales comparten similitudes prácticas con el túnel subterráneo, principalmente escenarios *indoor* o pasillos. Una aproximación útil es utilizar la traslación estimada por la odometría como traslación inicial para reducir el trabajo realizado por el algoritmo de *scan-matching* (Lu y Milios, 1994).

Es natural comparar los múltiples métodos existentes para revisar sus ventajas y desventajas, en (Ouellette y Hirasawa, 2007) se comparan implementaciones de SLAM (*simultaneous localization and mapping*) en distintos robots y ambientes de prueba *indoor* incluyendo pasillos, resultando como mejor método en pasillos DP-SLAM el cual utiliza un filtro de partículas para mantener una distribución de probabilidad sobre los mapas en una grilla actualizable.

En la investigación de (Nguyen, Martinelli, Tomatis, y Siegwart, 2005) se comparan seis métodos distintos para algoritmos de extracción de líneas en ambientes *indoor*, donde los mejores resultados son obtenidos por los métodos Split-Merge y transformada de Hough con respecto a detectar líneas donde deben ser detectadas mientras que RANSAC (Fischler y Bolles, 1981; Derpanis, 2010) obtuvo mayor precisión en la caracterización de las líneas.

El trabajo de Luo, Lin, y Lai en (Luo et al., 2008) explicita el problema de la falta de

*landmarks* naturales detectados por *features* en pasillos largos, donde el filtro extendido de Kalman utilizado para estimar la posición es actualizado sólo cuando se detectan dos *landmarks* al mismo tiempo a través de mediciones de LIDAR. Los *landmarks* son diferenciados utilizando *iterative endpoint fit* (IEPF) y la transformada de Hough para remuestrear las líneas de la pared detectada.

En (Donoso-Aguirre, Bustos-Salas, Torres-Torriti, y Guesalaga, 2007) se explica un escenario de prueba con un robot *skid steer* en una habitación utilizando la distancia de Hausdorff con una variación para tomar en cuenta mediciones parcialmente ofuscadas, situación que puede ocurrir en un túnel al colocar *landmarks* artificiales.

### 2.3. Scan-Matching

Para asignar las mediciones a modelos de datos con el objetivo de realizar *scan-matching* se han desarrollado múltiples métodos, uno de los más utilizados es *iterative closest point* (ICP) el cual permite calcular la rotación y traslación aplicada a una nube de puntos 3D para calzar con una nube de puntos modelo (Besl y McKay, 1992). Numerosas modificaciones se le han propuesto en la literatura que permiten mejorar distintos aspectos del algoritmo; en (Segal, Haehnel, y Thrun, 2009) se muestra ICP combinado con modelos probabilísticos donde se asume que todos los puntos corresponden a distribuciones gaussianas para luego iterar utilizando máxima verosimilitud.

En (Lu y Milios, 1994) se muestran dos algoritmos similares, uno donde se elige el punto más cercano y un punto de igual distancia como correspondencia para converger más rápido que el ICP convencional, mientras que otro utiliza la dirección tangencial de la curva para mejorar la convergencia.

### 3. ESTRUCTURA DE LA SOLUCIÓN PROPUESTA

Este trabajo busca soluciones para dos aspectos: como primer objetivo generar un estimador de posición para túneles rectos sin *features* naturales utilizando *landmarks* artificiales pasivos volumétricos, y como segundo objetivo encontrar parámetros de diseño para construir estos *landmarks* y de esta manera mejorar su desempeño en la estimación. El segundo objetivo utiliza el resultado del estimador logrado en el primero con tal de buscar a través de métodos de optimización *landmarks* con mejor desempeño y así analizar sus características. Este estimador es utilizado por un simulador para encontrar los parámetros de diseño, y luego se validan los resultados de forma experimental. En la figura 3.1 se muestra el resultado del estimador de posición con respecto a su valor real simulado. Para todos los casos se considerará que la trayectoria del vehículo y las paredes del túnel corresponden al eje X.

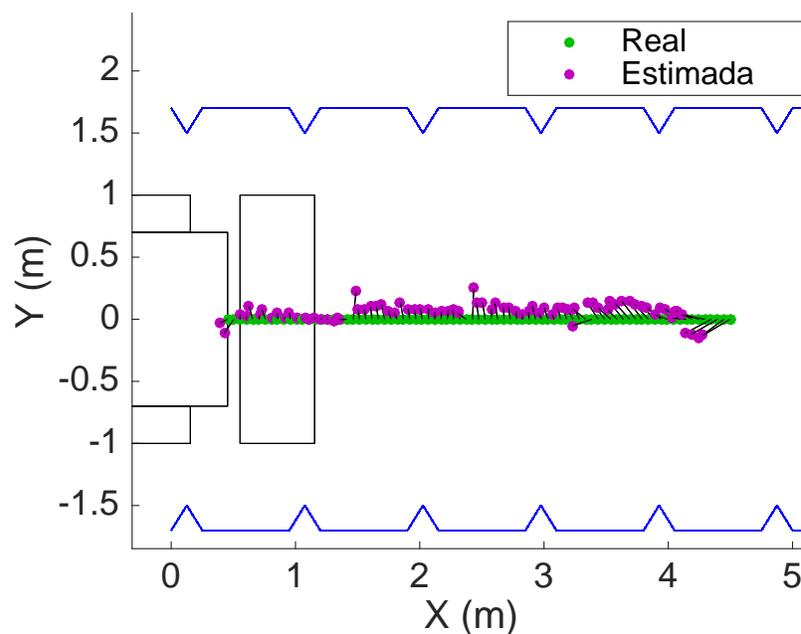


FIGURA 3.1. Posición real simulada del vehículo y su estimación de posición mediante el algoritmo.

### 3.1. Estimador de Posición

El estimador de posición de este trabajo corresponde a un algoritmo que busca calcular la posición relativa del vehículo dentro del túnel descrito en el escenario utilizando como sensor principal LIDAR o LMS, el cual es capaz de detectar la distancia radial a obstáculos en un plano de operación como se muestra en la figura 3.2. Además de utilizar la información generada por las señales de control para detectar discrepancias en la traslación.

Debido a que es de interés para este trabajo el desempeño propio del estimador y no el generar la mejor estimación de posición posible, se deja de lado la fusión con sensores adicionales. En el diagrama 3.3 se muestra un ejemplo conceptual de cómo se puede incorporar el estimador en un sistema que fusiona sensores con un filtro de Kalman en un sistema que ya utiliza un detector de cruces, estimación inercial y *dead reckoning*.

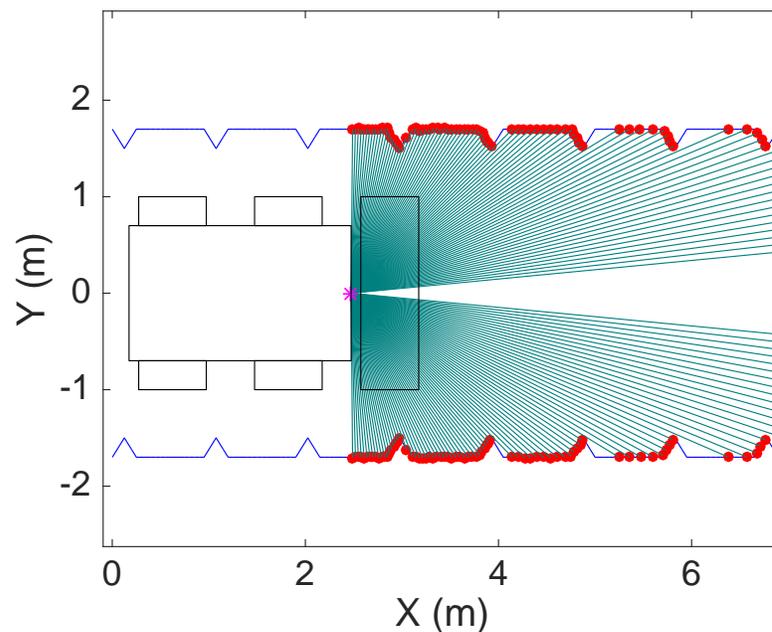


FIGURA 3.2. Diagrama de operación del LIDAR con respecto al vehículo y al túnel.

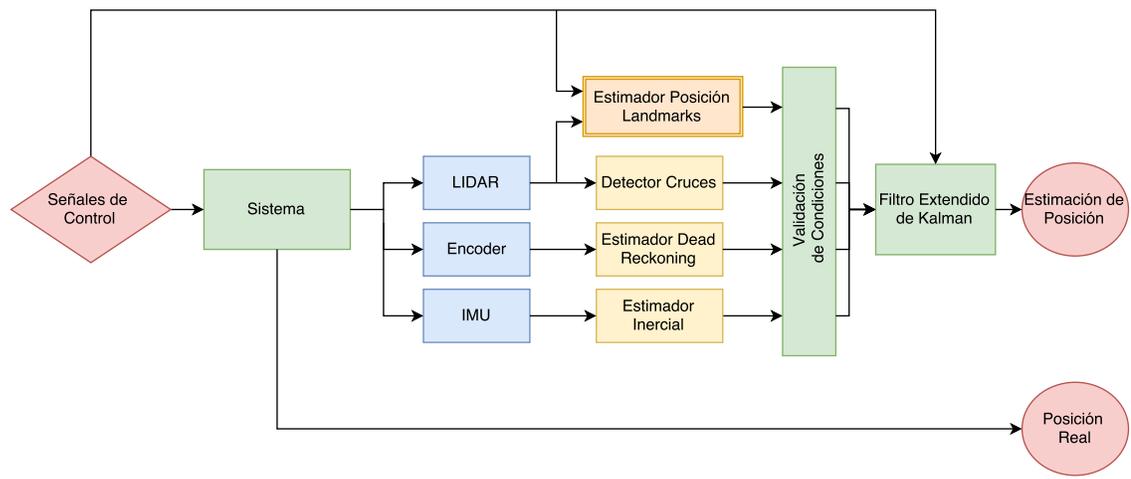


FIGURA 3.3. Ejemplo de integración del estimador de posición con un filtro de Kalman.

La estructura del estimador consiste en ejecutar los siguientes pasos en cada medición de LIDAR:

1. Obtención de mediciones del LIDAR.
2. Transformación de mediciones radiales en mediciones cartesianas.
3. Clasificación de puntos en *wall* o *feature*.
4. Agrupación de puntos de *feature* en *landmark*.
5. Identificación de posición del *landmark* con algoritmo de *matching*.
6. Identificación de mismo o distinto landmark.
7. Cálculo de traslación según el lado derecho del túnel desde la medición pasada.
8. Cálculo de posición según acumulación de traslaciones sucesivas.

Con el objetivo de implementar este algoritmo, se creó un simulador que genera datos representativos del escenario para calcular el desempeño de la implementación del algoritmo de localización. Los detalles del algoritmo se describen en la sección 4 y del simulador en la sección 5.

### **3.2. Diseño de Landmark**

Utilizando el simulador descrito previamente, se implementó una medida de desempeño para evaluar el algoritmo y el *landmark*, sistema el cual es sometido a un método de optimización genética fijando las condiciones de operación del algoritmo, pero variando los parámetros de construcción del *landmark* utilizado. Éste método tiene como objetivo determinar qué parámetros impactan en el desempeño del sistema y a la vez encontrar un diseño de *landmark* en particular que mejore la estimación total. Los detalles de la implementación de la optimización genética y la medida de desempeño se encuentran en la sección 6.

#### 4. ALGORITMOS DE ESTIMACIÓN DE POSICIÓN

En este capítulo se detallan los algoritmos utilizados para estimar la traslación del vehículo según la estructura indicada en el capítulo 3; el algoritmo utilizado para detectar la ubicación de la pared y clasificar los puntos en *pared* o *feature* corresponde a RANSAC (sección 4.1), el algoritmo de asignación de puntos a *landmark* en la sección 4.2 y la identificación de la posición del *landmark* del algoritmo de *matching* corresponden a ICP con distancia euclidiana (sección 4.3, referido como ICP en el resto del documento) y transformaciones rígidas iterativas con distancia dirigida modificada de Hausdorff (sección 4.4, referido como Hausdorff en el resto del documento), el algoritmo de corrección de medición para *landmarks* perdidos de vista se muestra en la sección 4.5.

Un diagrama de la operación del algoritmo se muestra en la figura 4.1.

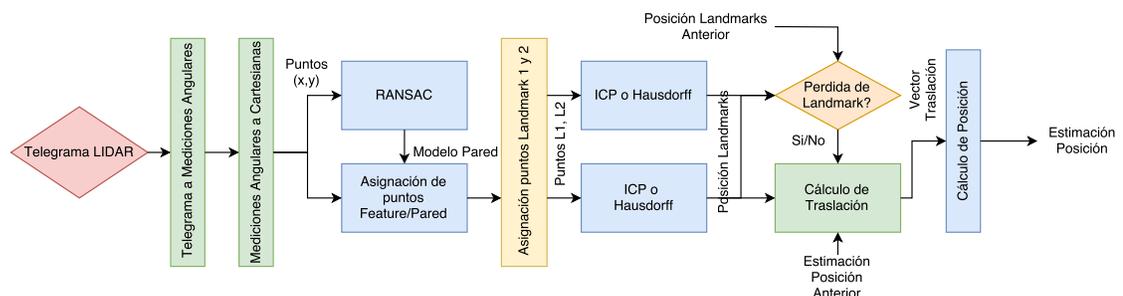


FIGURA 4.1. Flujo del algoritmo de estimación por muestra recibida de LIDAR.

##### 4.1. RANSAC

*Random Sample Consensus* es un algoritmo que permite ajustar un modelo o función a una nube de datos experimentales con la ventaja de que es capaz de rechazar puntos espurios que no se ajustan al modelo calculado sin afectar su desempeño, por lo que no es necesario discriminar y separar de antemano los puntos en la nube que corresponden a errores.

El algoritmo funciona de forma opuesta a las técnicas suavizantes de datos convencionales; en vez de usar la mayor cantidad de datos posibles para lograr una solución

inicial, éste utiliza un set de datos lo más pequeño posible (lo suficiente para definir el modelo) y aumenta este set de datos para incluir los puntos experimentales que pueden ser considerados como parte del modelo (Fischler y Bolles, 1981).

En el caso de disponer del modelo de *landmarks* que se quiere detectar este algoritmo es simple y genérico. Una forma general del algoritmo (Nguyen et al., 2005) corresponde a :

1. Inicialización de un set de  $N$  puntos medidos
2. Iteración de los siguientes pasos hasta alcanzar un modelo lo suficientemente ajustado o un máximo de iteraciones
  - a) Elección de  $M$  puntos uniformemente al azar suficientes para definir un modelo objetivo.
  - b) Ajuste de los parámetros del modelo a los  $M$  puntos elegidos.
  - c) Cálculo de distancia de cada punto de la nube a la curva.
  - d) Separación de puntos en cercanos y lejanos por una distancia umbral
  - e) Si hay suficientes puntos en el grupo cercano y la distancia de estos puntos es la menor encontrada guardar la curva como el mejor modelo encontrado
3. Retornar el mejor modelo y ambos grupos de puntos

En el caso de este trabajo, el algoritmo RANSAC es utilizado para determinar qué puntos detectados por el LIDAR corresponden a la pared del túnel para después rechazar los que calzan con el modelo y utilizar sólo los puntos considerados errores del modelo como los puntos pertenecientes a los *features* buscados. El modelo debe representar la pared del túnel que se asume, por hipótesis, como principalmente recto. Para este trabajo el modelo utilizado corresponde a un polinomio de tercer grado, el cual es capaz de describir rectas y posibles curvaturas suaves dentro del túnel. El algoritmo debe seleccionar 4 puntos para determinar al sistema como se muestra en la ecuación 4.2 donde cada punto  $p_i = (x_i, y_i)$  en coordenadas cartesianas relativas al vehículo para

encontrar el modelo  $Ax^3 + Bx^2 + Cx + D$  de ajuste (Tapia, 2012).

$$\begin{bmatrix} x_1^3 & x_1^2 & x_1 & 1 \\ x_2^3 & x_2^2 & x_2 & 1 \\ x_3^3 & x_3^2 & x_3 & 1 \\ x_4^3 & x_4^2 & x_4 & 1 \end{bmatrix} \begin{bmatrix} A \\ B \\ C \\ D \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} \quad (4.1)$$

$$\begin{bmatrix} A \\ B \\ C \\ D \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} \begin{bmatrix} x_1^3 & x_1^2 & x_1 & 1 \\ x_2^3 & x_2^2 & x_2 & 1 \\ x_3^3 & x_3^2 & x_3 & 1 \\ x_4^3 & x_4^2 & x_4 & 1 \end{bmatrix}^{-1} \quad (4.2)$$

Los 4 puntos elegidos por el algoritmo deben ser, en el caso ideal, pertenecientes a la pared y espaciados para evitar que el modelo se defina cerca de un *landmark* o en un área local respectivamente. Para esto se separa la nube en 4 grupos con la misma cardinalidad ordenados por posición, de los cuales se elige un punto al azar con probabilidad uniforme en cada uno de los grupos como se indica en la ecuación 4.3 donde  $N$  es la cantidad de puntos de la nube.

$$j_{1..4} = \left[ U\left(1, \frac{N}{4}\right), U\left(\frac{N}{4}, \frac{N}{2}\right), U\left(\frac{N}{2}, \frac{3N}{4}\right), U\left(\frac{3N}{4}, N\right) \right] \quad (4.3)$$

Para evaluar la calidad del modelo elegido es necesario calcular la distancia de cada punto al modelo obtenido, para simplificar el cálculo se considera la distancia perpendicular al movimiento del vehículo del punto de la nube. En la figura 4.2 se muestra una iteración del algoritmo RANSAC para un *landmark* y curvatura de prueba donde el modelo no cumple con los criterios de desempeño, mientras que 4.3 es el resultado final del algoritmo que sí cumple con el criterio especificado. En este caso el vehículo se mueve a lo largo del eje X y la distancia utilizada de desempeño se mide a lo largo del eje Y.

La medida de desempeño del modelo corresponde a la razón entre la cantidad de

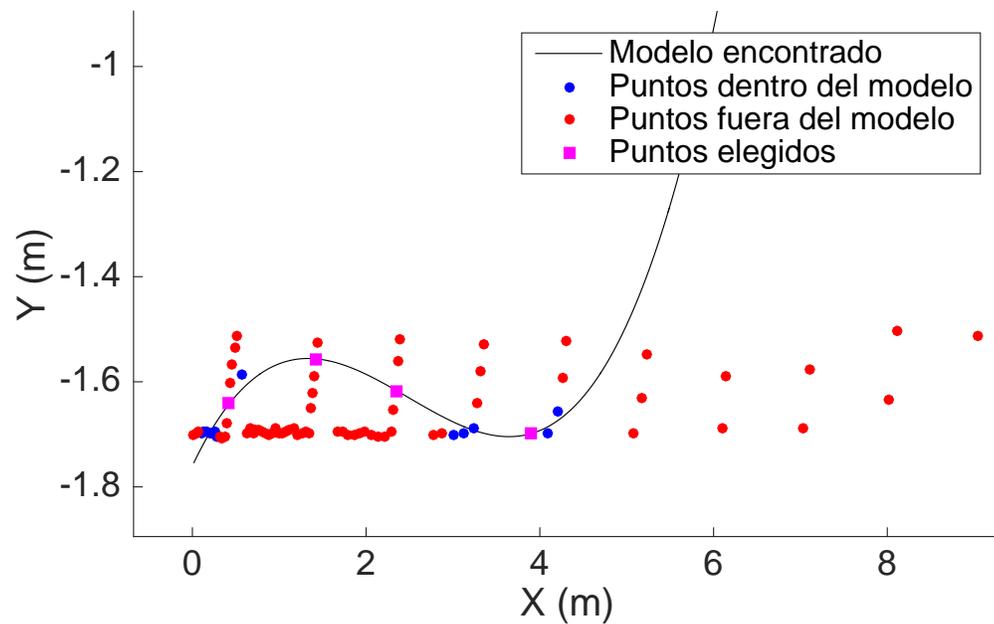


FIGURA 4.2. Iteración de RANSAC incorrecta que no cumple con los parámetros especificados.

puntos considerados dentro del modelo con respecto a la cantidad de puntos totales, la cual se compara con un valor objetivo. El algoritmo itera variando los puntos seleccionados para encontrar el mejor modelo válido después de un número fijo de iteraciones.

En el algoritmo 1 se muestra el RANSAC utilizado.

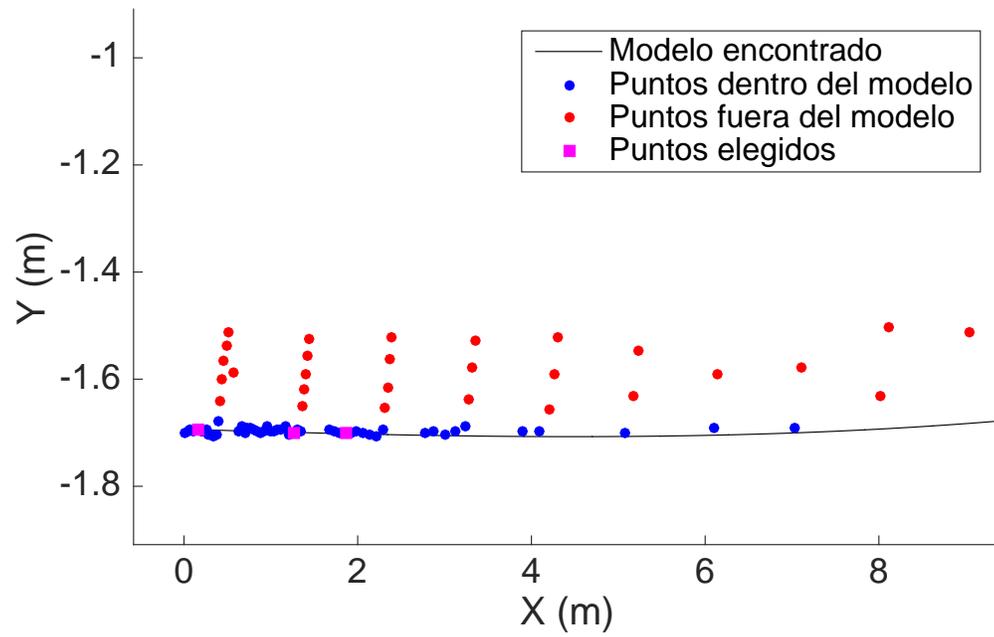


FIGURA 4.3. Iteración de RANSAC final que cumple con los parámetros especificados.

**Input:** Puntos detectados por el LIDAR  $P$ , Distancia límite  $d_{th}$ , Cantidad de iteraciones  $QI$

**Output:** Parámetros del Modelo  $A, B, C, D$ , conjunto de puntos  $P_p$  ajustado a la curva (pared), conjunto de puntos de error  $P_e$  (*landmark*)

$P_p, P_e \leftarrow \emptyset$

$E_{best} \leftarrow 0$

**para**  $i = \{1..QI\}$  **hacer**

$j_{1,4} \leftarrow [U(1, \frac{N}{4}), U(\frac{N}{4}, \frac{N}{2}), U(\frac{N}{2}, \frac{3N}{4}), U(\frac{3N}{4}, N)]$

$M_a \leftarrow \begin{bmatrix} x_{j1}^3 & x_{j1}^2 & x_{j1} & 1 \\ x_{j2}^3 & x_{j2}^2 & x_{j2} & 1 \\ x_{j3}^3 & x_{j3}^2 & x_{j3} & 1 \\ x_{j4}^3 & x_{j4}^2 & x_{j4} & 1 \end{bmatrix}$

$w \leftarrow \begin{bmatrix} y_{j1} \\ y_{j2} \\ y_{j3} \\ y_{j4} \end{bmatrix}$

$v \leftarrow M_a^{-1} \cdot w$

$E_d \leftarrow \frac{\sum_j^N (y_j - (v_1 \cdot x_j^3 + v_2 \cdot x_j^2 + v_3 \cdot x_j + v_4))^2}{N}$

**Si**  $E_d < E_{best}$  **entonces**

$E_{best} \leftarrow E_d$

$[A, B, C, D] \leftarrow v$

$T \leftarrow |Y - (A \cdot X^3 + B \cdot X^2 + C \cdot X + D)|$

$P_p \leftarrow P_{T < d_{th}}$

$P_e \leftarrow P_{T \geq d_{th}}$

**fin**

**fin**

**Retorna**  $P_p, P_e, A, B, C, D$

**Algoritmo 1:** RANSAC Calcula una curva cúbica ajustada a los datos detectados.

## 4.2. Asignación del Landmark

Una vez ejecutado el algoritmo de detección de puntos de *feature*, se deben agrupar estos puntos para identificar cuáles corresponden a un determinado *landmark*. Esto se logra midiendo la distancia entre cada par de puntos consecutivos y separando los grupos si se detecta una distancia mayor a un valor límite, ya que en una detección correcta los puntos van a estar separados con una distancia similar al espaciamiento entre *landmarks* utilizado en el diseño preliminar.

Si no se logran detectar 2 o más grupos se reduce el valor iterativamente hasta encontrarlos. En la figura 4.4 se muestra un ejemplo de las muestras de LIDAR clasificadas según pared-*feature* y asignadas a sus respectivos *landmarks*.

El algoritmo de asignación se muestra es el 2.

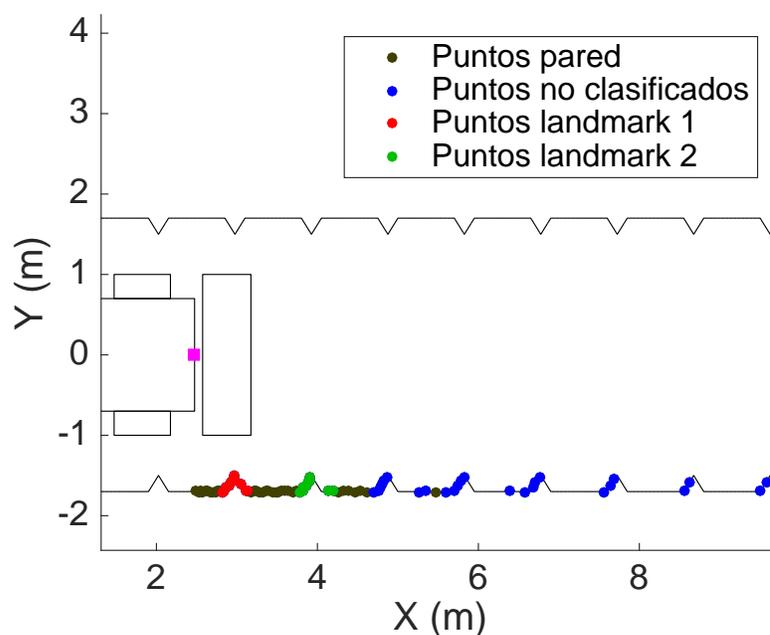


FIGURA 4.4. Ejemplo de asignación de puntos a Landmarks.

Para no perder la referencia de traslación es necesario tener detectado e identificado anteriormente un *landmark*; por lo que se busca detectar los 2 *landmarks* más próximos al vehículo, utilizando el más cercano al vehículo para medir el desplazamiento

**Input:** Puntos detectados por el LIDAR  $P$ , Puntos considerados como *feature*  $P_e$ , Distancia límite entre landmarks  $L_{th}$ , Factor de escalamiento  $g_{assign}$

**Output:** Índices de los puntos pertenecientes a cada *landmark*  $AS_{(c,j)}$

$AS_{(c,j)} \leftarrow \emptyset$

$C, J \leftarrow 1$

**mientras**  $C < 2$  **hacer**

**para**  $i = \{1..N\}$  **hacer**

**Si**  $P_i \in P_e$  **entonces**

**Si**  $\|P_{i-1} - P_i\| < L_{th}$  **entonces**

$AS_{(C,J)} \leftarrow i$

$J++$

**sinó**

$C++, J \leftarrow 1$

$AS_{(C,J)} \leftarrow i$

$J++$

**fin**

**fin**

**fin**

$L_{th} \leftarrow L_{th} \cdot g_{assign}$

$AS_{(c,j)} \leftarrow \emptyset$

$C, J \leftarrow 1$

**fin**

**para**  $i = \{1..C\}$  **hacer**

$k \leftarrow \text{largo}(AS_{(i,:)})$

$AS_{(i,\{k+1,..,k+4\})} \leftarrow$

$\{\min(AS_{(i,:)} - 1), \min(AS_{(i,:)} - 2), \max(AS_{(i,:)} + 1), \max(AS_{(i,:)} + 2)\}$

**fin**

**Retorna**  $AS_{(c,j)}$

**Algoritmo 2:** ASIGNACIÓN Determina que conjuntos de puntos pertenecen a cada *landmark*.

y registrando la posición del segundo para utilizarlo en caso de perder de vista al primero. Éste método se explica con mayor detalle en la sección 4.5.

La nube de puntos de un *landmark* asignado no incluye la base de la pared que fue eliminada por RANSAC, lo que permite que al detectar sólo un borde del *landmark*, éste tenga varias soluciones para el algoritmo de *matching*. Es por esto que artificialmente se añaden los 2 puntos anteriores y posteriores de la medición del LIDAR para

asegurar de que la pared está representada y reduzca la variabilidad de la solución.

### 4.3. ICP

*Iterative Closest Point* es un algoritmo de propósito general que permite encontrar la transformación de cuerpo rígido entre dos nubes de puntos e iterativamente refinar la transformación inicial utilizada como hipótesis, generando pares de puntos correspondientes entre ambas nubes con el objetivo de minimizar un criterio de error (Rusinkiewicz y Levoy, 2001). El ICP converge de forma monótona a mínimos locales y tiende a converger rápidamente para figuras con un cierto nivel de complejidad, por lo que es posible utilizarlo para relacionar la figura sensada con la figura objetivo o ideal (Besl y McKay, 1992).

La taxonomía básica del algoritmo ICP contiene los siguientes pasos (Rusinkiewicz y Levoy, 2001):

1. Selección de puntos en una o ambas nubes de puntos.
2. Pareo de puntos seleccionados, relacionar cada punto en una nube con su correspondiente en la nube contraria.
3. Pesaje de los pares de puntos de forma apropiada, asignando importancia distinta a cada par de puntos según criterio de la implementación.
4. Rechazar ciertos pares según criterio de la implementación revisando cada par de forma individual o en conjunto.
5. Asignar una métrica de error de matching basada en los pares de puntos no rechazados.
6. Minimizar el criterio de error.

El algoritmo ICP utilizado en este trabajo corresponde al método implementado y publicado por Kjer y Wilm para superficies de tres dimensiones (Kjer y Wilm, 2010). Debido a que la asignación de la superficie del *landmark* y el modelo es un problema en

sólo dos dimensiones, a cada punto de modelo y medición se le asigna una coordenada de valor 0 en la tercera dimensión, utilizando el algoritmo sin perder generalidad.

La implementación del algoritmo utilizado corresponde a la siguiente:

1. Selección de puntos: Se recorre toda la nube de medición, tomando cada punto y comparándolo con cada punto en la nube modelo.
2. Pareo de puntos: Se calcula la distancia euclidiana entre cada punto de medición con los puntos de la nube modelo, eligiendo el punto con la menor distancia con repetición.
3. Pesaje de los pares: Se aplica un pesaje uniforme para cada par.
4. Rechazo de pares: No se realiza rechazo de pares, todos son considerados.
5. Métrica de error: Error RMS de las distancias euclidianas de cada punto de la nube de datos con su punto asignado en la nube modelo.
6. Minimización: Utilizando descomposición de valores singulares se encuentra una solución de forma cerrada.

Este algoritmo se aplica sobre los puntos detectados y separados en la etapa de asignación como nube de medición, mientras que la nube modelo es el vector de muestra utilizado para definir el *landmark*.

#### 4.4. Distancia de Hausdorff

La distancia de Hausdorff corresponde a un cálculo de similitud entre dos imágenes o nubes de puntos, que mide la cercanía de cada punto de un conjunto modelo con un conjunto medido. La distancia esta definida por la ecuación 4.4 donde  $H$  es la distancia de Hausdorff y  $h$  la distancia dirigida de Hausdorff, ambos sets de puntos finitos se definen como  $PL = \{pl_1, \dots, pl_i\}$  para el conjunto medido y  $PM = \{pm_1, \dots, pm_i\}$  para el conjunto de referencia (Knauer, Löffler, Scherfenberg, y Wolle, 2011). La distancia dirigida de Hausdorff es la distancia corresponde a la mayor distancia euclidiana entre cada punto y el punto más cercano del set contrario, mientras

que la distancia de Hausdorff es la mayor de las distancias dirigidas intercambiando los conjuntos.

$$H(PL, PM) = \text{máx}(h(PM, PL), h(PL, PM)) \quad (4.4)$$

$$h(PL, PM) = \text{máx}_{pl \in PL} \left( \text{mín}_{pm \in PM} (||pl - pm||) \right) \quad (4.5)$$

Se utiliza este cálculo junto con un método de transformación rígida iterativa como alternativa a ICP para realizar *matching*, ya que tiene una relativa insensibilidad a perturbaciones pequeñas y permite comparar formas incompletas como las que son obtenidas por el ángulo de detección del LIDAR (Huttenlocher, Klanderman, y Rucklidge, 1993).

Para esta implementación se utiliza la distancia dirigida de Hausdorff (ecuación 4.5) en vez de la distancia completa para evitar que los puntos del modelo que no están representados en la nube objetivo afecten la medición. Adicionalmente, se considera la distancia modificada de Hausdorff que en vez de utilizar la distancia máxima de las mínimas usa la n-ésima medición de las distancias ordenadas de forma creciente (Torres-Torriti y Guesalaga, 2008).

En la figura 4.5 y 4.6 se muestra el resultado del algoritmo implementado para un *landmark* de prueba cuyos puntos están sujetos a un ruido gaussiano de 0.01 m mientras que el 10% de los puntos fue sometido a un ruido gaussiano de 0.1 m para representar mediciones espurias. En la figura 4.5 se utilizó la distancia dirigida de Hausdorff mientras que en 4.6 se utilizó la distancia dirigida modificada al 80% del total.

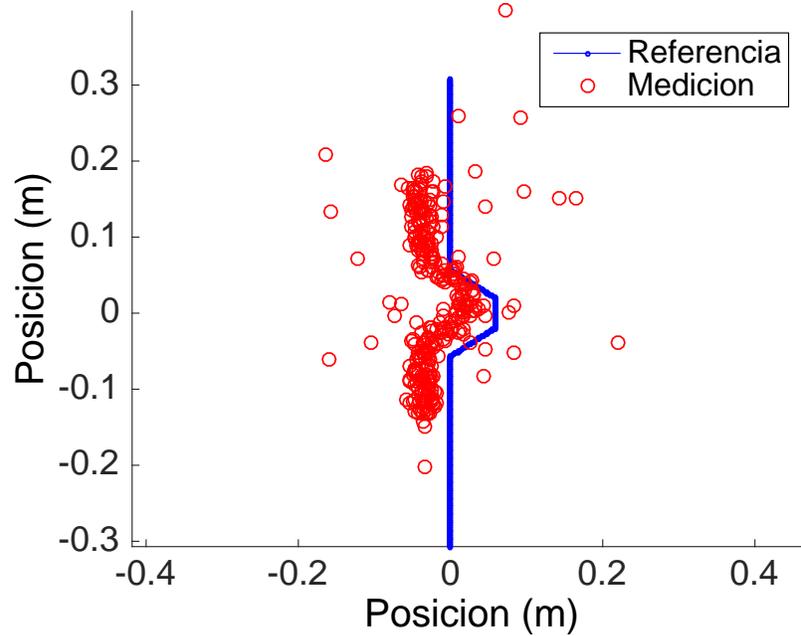


FIGURA 4.5. Resultado del algoritmo de Hausdorff con un landmark de prueba y selección de distancia al 100% del máximo.

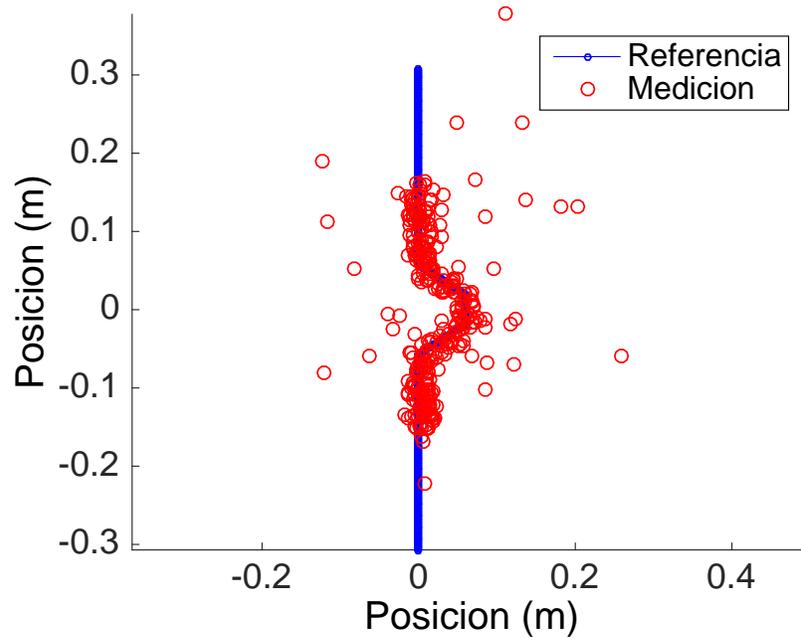


FIGURA 4.6. Resultado del algoritmo de Hausdorff con un landmark de prueba y selección de distancia al 80% del máximo.

Para acelerar el cálculo de la distancia se calcula previamente un mapa de distancias mínimas mediante un mapa de Voronoi de la nube modelo, para ser utilizado en cada cálculo de *matching*. Este mapa consiste en un plano donde el valor de cada punto corresponde a la distancia mínima desde esa ubicación a todos los puntos de la nube, como se muestra en la figura 4.7. Lo anterior permite calcular la distancia dirigida de Hausdorff sin iterar sobre la nube modelo.

La distancia se reduce inicialmente trasladando el centro de masa de la nube me-

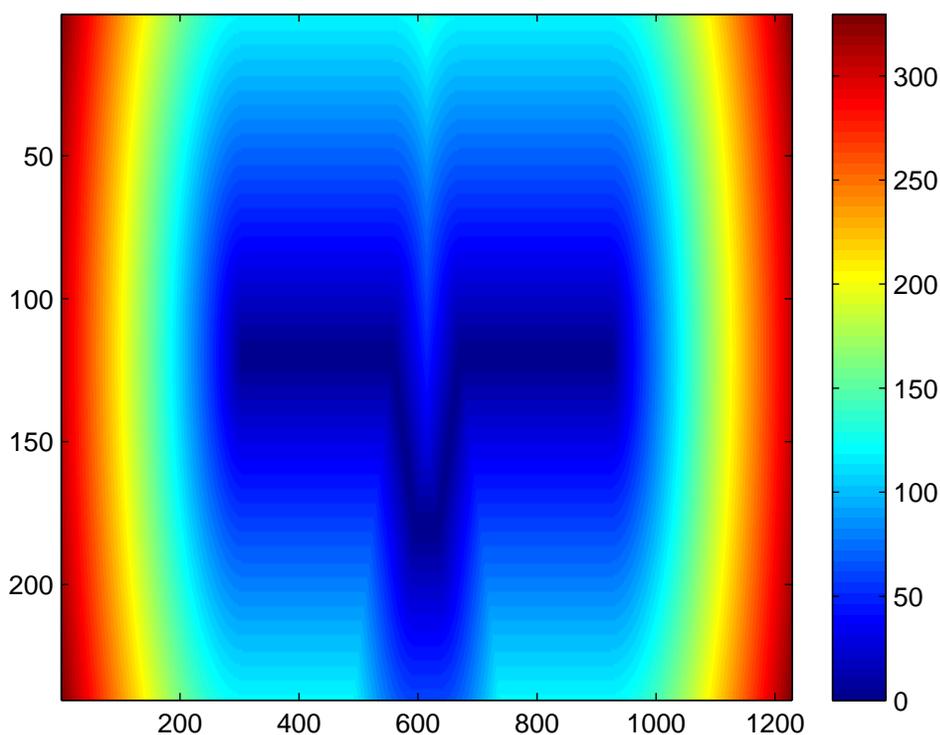


FIGURA 4.7. Mapa de distancias mínimas calculado para un landmark de prueba.

didada al centro de masa de la nube modelo, para luego iterar trasladando la nube en  $(\pm x_{step}; \pm y_{step})$  en y rotando en un rango  $(-\theta_{range}, \theta_{range})$  en pasos de  $\theta_{step}$  para encontrar un mínimo local de la distancia de Hausdorff y reducir estos valores en cada iteración hasta un número fijo. En el algoritmo 3 se muestra el algoritmo de cálculo

para la distancia de Hausdorff y en el algoritmo 4 el de *matching* que utiliza la distancia.

**Input:** Matriz de distancias mínimas correspondiente al modelo del *landmark*  $V$ , Resolución del mapa  $r_v$ , Nube de puntos  $PL$ , Fracción de modificación para la distancia  $r_m$

**Output:** Distancia de Hausdorff dirigida  $h$ , Punto más lejano de la nube  $h_p$

$$Lims \leftarrow \frac{\text{largo}(V) \cdot r_v}{2}$$

**para**  $i = 1..NL$  **hacer**

$$out_x \leftarrow xl_i > Lims_x \vee xl_i < 0$$

$$out_y \leftarrow yl_i > Lims_y \vee yl_i < 0$$

**Si**  $out_x \wedge xl_i > Lims_x$  **entonces**

$$| \quad bx \leftarrow \frac{\text{largo}(V_x)}{2}$$

**sinó, si**  $out_x \wedge xl_i < -Lims_x$  **entonces**

$$| \quad bx \leftarrow 1$$

**sinó**

$$| \quad bx \leftarrow \text{redondear} \left( \frac{xl_i}{r_v} \right)$$

**fin**

**Si**  $out_y \wedge yl_i > Lims_y$  **entonces**

$$| \quad by \leftarrow \frac{\text{largo}(V_y)}{2}$$

**sinó, si**  $out_y \wedge yl_i < -Lims_y$  **entonces**

$$| \quad by \leftarrow 1$$

**sinó**

$$| \quad by \leftarrow \text{redondear} \left( \frac{yl_i}{r_v} \right)$$

**fin**

$$d_i \leftarrow \left[ V_{(bx,by)+\left|pl-\text{redondear}\left(\frac{pl}{r_v}\right)\right|}, i \right]$$

**fin**

$$d \leftarrow \text{ordenar}(d)$$

$$id \leftarrow r_m \cdot \text{largo}(d)$$

$$h \leftarrow d_{id}$$

$$h_p \leftarrow pl_i$$

**Retorna**  $h, h_p$

**Algoritmo 3:** DISTANCIA DE HAUSDORFF Calcula la distancia dirigida modificada entre una nube de puntos y un mapa de distancias mínimas.

**Input:** Matriz de distancias mínimas correspondiente al modelo del *landmark*  $V$ , Resolución del mapa  $r_v$ , Nube de puntos  $PL$ , Fracción de modificación para la distancia  $(x_{step}, y_{step})$  y divisor por nivel  $t_d$ , Fracción de modificación angular  $\theta_{step}$  y divisor por nivel  $\theta_d$ , Cantidad de niveles de búsqueda  $l_m$ , nivel de mejora mínimo  $f_{th}$

**Output:** Rotación óptima  $\theta$ , Translación óptima  $T$

$T \leftarrow -[\text{promedio}(XL), \text{promedio}(YL)]$

**para**  $i = 1.. \frac{360}{\theta_{step}}$  **hacer**

$\alpha \leftarrow i \cdot \theta_{step}$

$R_t \leftarrow \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix}$

$R_p \leftarrow R_t \cdot (PL + T)$ ,  $E_i \leftarrow h \leftarrow \text{DistanciaDeHausdorff}(V, r_v, R_p, r_m)$

**fin**

$\theta \leftarrow \theta_{step} \cdot \text{IndiceDe}(\text{Min}(E))$ ,  $G_{min} \leftarrow \text{Min}(E)$

$R_p \leftarrow \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \cdot (PL + T)$

**mientras**  $level < l_m$  **hacer**

$Taux_{1,A} \leftarrow [(x_{step}, 0); (-x_{step}, 0); (0, y_{step}); (0, -y_{step})]$

$PT_{1,A} \leftarrow R_p + Taux$ ,  $F_{1,A} \leftarrow \text{DistanciaDeHausdorff}(V, r_v, PT_{1,A}, r_m)$

$b_t \leftarrow \text{Min}(F) - G_{min} > f_{th}$

**Si**  $b_t$  **entonces**

$T \leftarrow T + Taux_{\text{IndiceDe}(\text{Min}(F))}$

**fin**

$\beta_{aux} \leftarrow -\theta_{step}.. \theta_{step}$

**para**  $i = \text{largo}(\beta_{aux})$  **hacer**

$\beta \leftarrow \beta_{aux}_i$

$R_t \leftarrow \begin{bmatrix} \cos(\beta + \theta) & -\sin(\beta + \theta) \\ \sin(\beta + \theta) & \cos(\beta + \theta) \end{bmatrix}$

$P\beta_i \leftarrow R_t \cdot R_p$

$N_i \leftarrow \text{DistanciaDeHausdorff}(V, r_v, P\beta_i, r_m)$

**fin**

$b_\theta \leftarrow \text{Min}(N, G_{min}) \neq G_{min}$

**Si**  $b_\theta$  **entonces**

$\theta \leftarrow \theta + \beta_{aux}_{\text{IndiceDe}(\text{Min}(N))}$

**fin**

**Si**  $\overline{b_\theta} \wedge \overline{b_t}$  **entonces**

$\theta_{step} \leftarrow \frac{\theta_{step}}{\theta_d}$ ,  $t_m \leftarrow \frac{t_m}{t_d}$ ,  $level ++$

**fin**

**fin**

**Retorna**  $\theta, T$

**Algoritmo 4:** *Matching* DE HAUSDORFF Reduce la distancia dirigida modificada entre una nube de puntos y el modelo de *landmark*.

#### 4.5. Corrección de Medición

La información de las señales de control permite estimar el desplazamiento de forma referencial para cada paso, y si se detecta una discrepancia con lo calculado mediante *landmarks* sobre un valor límite se puede asumir que se perdió de vista el de referencia y es necesario corregir la pérdida de visión. El efecto de perder de vista un *landmark* sin corrección corresponde a desplazarse en sentido contrario al movimiento del vehículo una distancia similar a la distancia entre *landmarks*. En la figura 4.8 se muestra la progresión de la traslación a lo largo del túnel para un caso de ejemplo con y sin corrección.

El algoritmo de corrección corresponde al pseudocódigo 5.

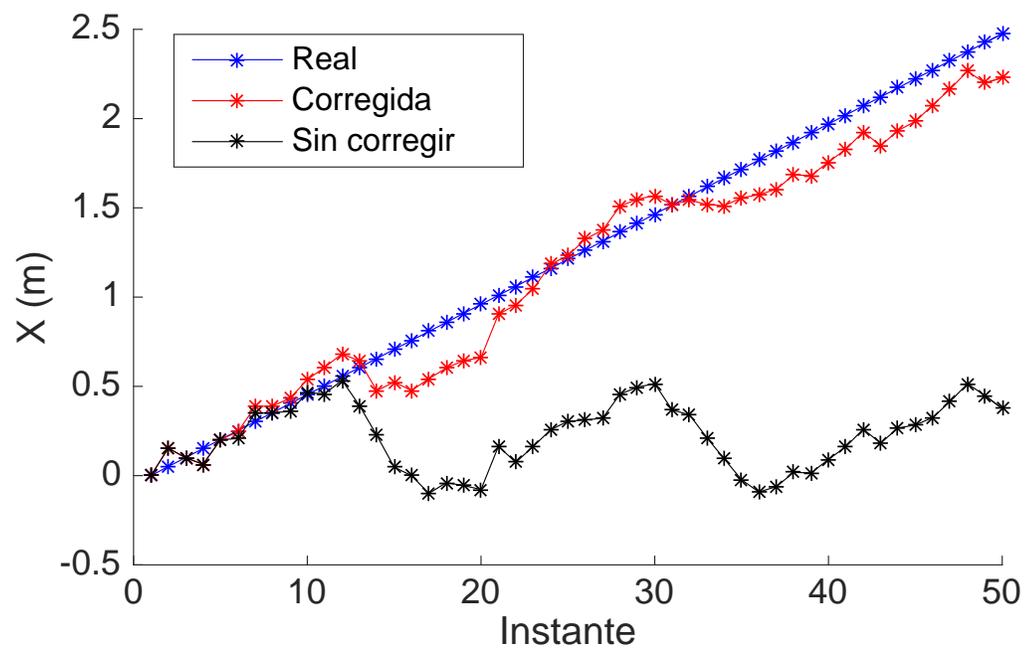


FIGURA 4.8. Ejemplo de traslación con y sin corrección por pérdida de landmarks.

**Input:** Nubes de puntos correspondientes a los 3 *landmarks* más cercanos al vehículo  $PL1, PL2, PL3$ , *landmarks* identificados en la iteración anterior  $PV1, PV2$ , distancia entre *landmarks*  $L_L$ , factor límite para detectar perdida  $k_{th}$ , Desplazamiento estimado por señales de control  $DR$

**Output:** *landmarks* asignados a la primera y segunda posición  $PN1, PN2$ , Desplazamiento corregido  $d_{cor}$

$M1 \leftarrow$  Promedio ( $PL1$ ),  $M2 \leftarrow$  Promedio ( $PL2$ )

$MV1 \leftarrow$  Promedio ( $PV1$ ),  $MV2 \leftarrow$  Promedio ( $PV2$ )

**Si**  $\text{abs}((M1_x - MV1_x) - DR_x) > L_L \cdot k_{th}$  **entonces**

$PN1 \leftarrow PV2$

$PN2 \leftarrow PV3$

$d_{cor} \leftarrow PN1 - PV2$

**sinó**

$PN1 \leftarrow PV1$

$PN2 \leftarrow PV2$

$d_{cor} \leftarrow PN1 - PV1$

**fin**

**Retorna**  $PN1, PN2, d_{cor}$

**Algoritmo 5:** CORRECCIÓN DE *landmarks* corrige la asignación de *landmarks* cuando se pierde de vista el más cercano.

## 5. MODELACIÓN DEL ENTORNO

Para encontrar el modelo de *landmark* mejorado, se simula el ambiente del túnel y se aplica el algoritmo de localización en distintas condiciones detalladas en la sección 6. En esta sección se detallan las características generales del simulador a utilizar y la modelación de distintos aspectos del escenario a considerar.

### 5.1. Características generales

El simulador está implementado en MATLAB donde se programaron numerosas funciones que representan distintas secciones del simulador. La lista de funciones y sus características se detallan en el anexo.

### 5.2. Modelación del Túnel

Se considera una topología de un túnel completamente recto para representar el escenario descrito en la sección 1. Debido a que el LIDAR detecta distancias en un plano horizontal se utilizan 2 paredes rectas equidistantes.

La ruta a seguir por el vehículo es la recta equidistante a ambas paredes, se asume que el vehículo se mueve en dirección positiva por el eje X del túnel. En la figura 5.1 se muestra una vista aérea de un túnel de muestra.

### 5.3. Modelación de LIDAR

El LIDAR a utilizar en los experimentos es SICK LMS511-20100, cuyo funcionamiento es replicado en el simulador. Se instaló éste sensor en la parte frontal del vehículo apuntando en el sentido de movimiento como se muestra en la figura 3.2. Se utiliza un ángulo de visión total de  $180^\circ$  con mediciones cada  $1^\circ$  de la distancia entre el sensor y la primera colisión con la pared, con un máximo de 80 m.

Cada medición es sujeta a un error sistemático y estadístico (SICK, 2015), para luego

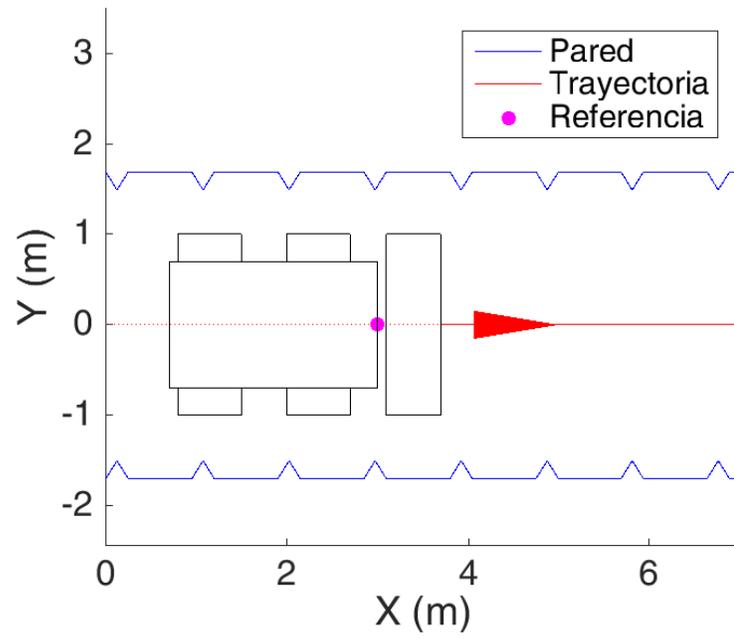


FIGURA 5.1. Túnel simulado de muestra.

ser transformada a coordenadas cartesianas relativas al vehículo.

#### 5.4. Modelación del cargador frontal

Para representar el vehículo se utiliza el modelo cinemático simplificado para una plataforma *skid-steer* con respecto al sistema de coordenadas locales (Oonk, Maldonado, Li, Reichard, y Pentzer, 2014) de las ecuaciones 5.1.

$$V_x = \frac{V_r + V_l}{2} \cos \theta \quad (5.1)$$

$$V_y = \frac{V_r + V_l}{2} \sin \theta \quad (5.2)$$

$$\dot{\gamma} = \frac{V_r - V_l}{b} \quad (5.3)$$

Se considera un resbalamiento proporcional al giro de cada rueda aplicado a cada rueda de forma separada en su sentido de rotación, y no se considera un resbalamiento lateral.

## 6. ALGORITMO DE LANDMARK ÓPTIMO

Con el objetivo de obtener características optimizadas del diseño de *landmarks* para la estimación de posición dentro del túnel, se simula el escenario de prueba conservando sus características, pero utilizando distintos diseños de *landmarks* seleccionados a través de un algoritmo genético de optimización con tal de elegir sistemáticamente *landmarks* de mejor desempeño.

### 6.1. Algoritmo Genético

La optimización por algoritmos genéticos es un método de búsqueda de soluciones inspirado en la selección natural, que asume que la solución de un problema potencial puede ser representada por un set de parámetros denominados genes (Man, Tang, y Kwong, 1996). En este caso, el algoritmo genético tiene como objetivo mejorar las características de un *landmark*, asumido utilizando una función de desempeño relacionada con la estimación de posición indicada en 6.2.

El algoritmo genético fue utilizado por su facilidad de implementación dentro del simulador, por no necesitar un buen punto de inicio del algoritmo y además por poseer una relativa inmunidad al ruido (Dahiya, Saini, y Gopal, 2015; Goldberg, 1994). Estos aspectos son ventajosos, ya que la simulación utiliza métodos basados en el azar que generan ruido en sus realizaciones, lo cual no afectaría la capacidad del algoritmo de acercarse a una solución optimizada en general.

El genoma fue diseñado para describir de forma numérica el dibujo de los *landmarks* a lo largo del túnel incluyendo distancias y elevaciones, mientras que se considera que el *landmark* es constante con respecto a la altura y es capaz de ser visto sin problemas por el plano de operación.

El genoma fue separado en dos casos distintos: un genoma con formas primitivas o predeterminadas (sección 6.1.1.1), y un genoma con una forma no determinada conformado por una cadena de elevaciones (6.1.1.2).

### 6.1.1. Genomas

#### 6.1.1.1. Forma Primitiva

El genoma para figuras primitivas asume de manera predeterminada una forma específica que puede variar entre una forma triangular, una rectangular, una forma de cúpula y una cuña, tal como se muestra en la figura 6.1. El genoma contiene 3 valores de dimensionalidad que modifican la profundidad, ancho y distancia de las formas en la pared del túnel, los cuales se detallan en la tabla 6.1. En la figura 6.2 se muestran ejemplos de distintos genomas para el caso triangular.

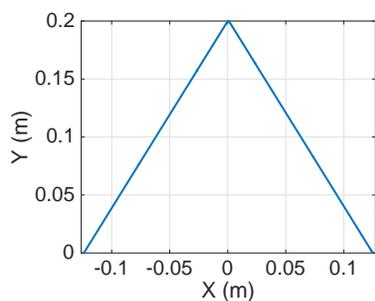
TABLA 6.1. Descripción del genoma para las figuras con forma primitiva.

Gen N°	Tipo	Nombre	Descripción
1	Racional	<i>Scale Factor</i>	Multiplicador para la profundidad
2	Entero	<i>Form Length</i>	Medida del ancho del <i>landmark</i>
3	Entero	<i>Blank Spacing</i>	Medida de distancia entre <i>landmarks</i>

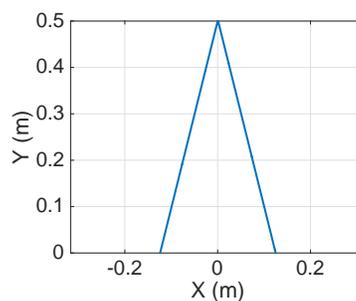
#### 6.1.1.2. Forma No Determinada

El genoma para figuras no determinadas no asume una forma específica como el caso primitivo, sino que asume una forma con una estructura continua, que ubica 6 puntos equidistantes a lo largo del ancho del *landmark* a diferentes alturas según el valor del genoma correspondiente, uniéndolos con líneas rectas entre sí o entre el último punto y la pared según corresponda, generando una estructura poligonal como se puede observar en la figura 6.3. Esta estructura le permite al algoritmo genético encontrar formas mejoradas que no se ajustan a las formas primitivas al igual que el caso primitivo.

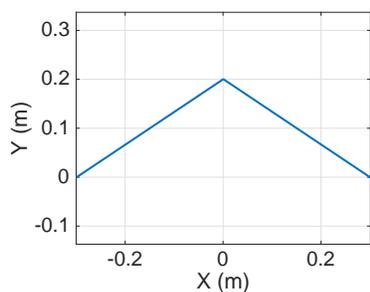
El genoma también contiene 3 valores de dimensionalidad que modifican la profundidad, ancho y distancia de las formas en la pared del túnel; el detalle del genoma se puede ver en la tabla 6.2.



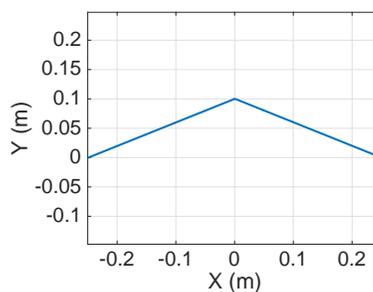
(A) Forma 1, Genoma [0.2,50,700].



(B) Forma 1, Genoma [0.5,50,700].



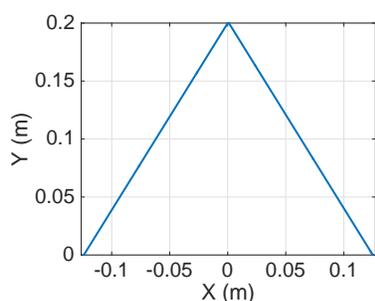
(C) Forma 1, Genoma [0.2,120,700].



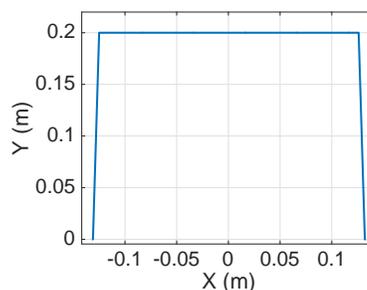
(D) Forma 1, Genoma [0.1,100,700].

FIGURA 6.1. Ejemplos de genomas de forma primitiva.

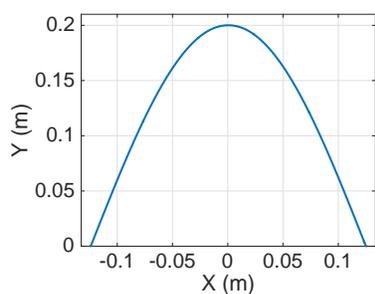
Para añadir variabilidad a la estructura, se consideran casos reflejados donde la forma *landmark* se invierte a lo largo del túnel, casos reflejados en donde se genera un *landmark* simétrico, utilizando como punto de reflexión el punto más alejado; y casos simétricos reflejados donde se utilizan ambas modificaciones. En la figura 6.4 se muestran ejemplos del mismo genoma bajo estas condiciones.



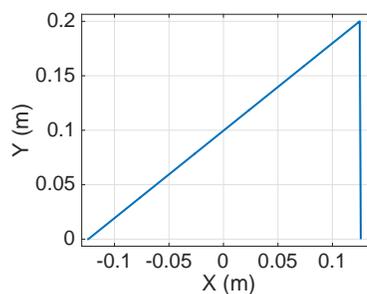
(A) Forma 1, Genoma [0.2,50,700].



(B) Forma 2, Genoma [0.2,50,700].



(C) Forma 3, Genoma [0.2,50,700].

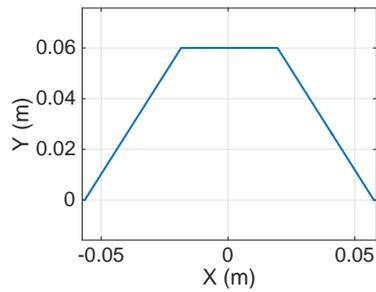


(D) Forma 4, Genoma [0.2,50,700].

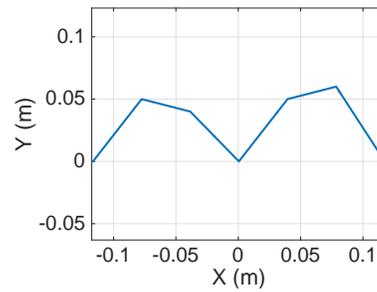
FIGURA 6.2. Ejemplos de formas para figuras primitivas.

TABLA 6.2. Descripción del genoma para las figuras con forma no determinada.

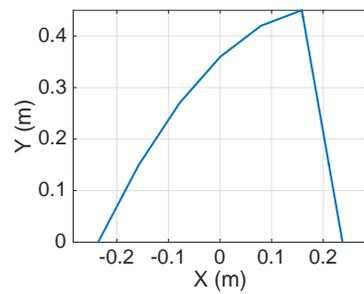
Gen N°	Tipo	Nombre	Descripción
1	Racional	<i>Shape Point 1</i>	Diferencia de profundidad del punto 1 con respecto a la pared
2	Racional	<i>Shape Point 2</i>	Diferencia de profundidad del punto 2 con respecto al punto 1
3	Racional	<i>Shape Point 3</i>	Diferencia de profundidad del punto 3 con respecto al punto 2
4	Racional	<i>Shape Point 4</i>	Diferencia de profundidad del punto 4 con respecto al punto 3
5	Racional	<i>Shape Point 5</i>	Diferencia de profundidad del punto 5 con respecto al punto 4
6	Racional	<i>Shape Point 6</i>	Diferencia de profundidad del punto 6 con respecto al punto 5
7	Racional	<i>Scale Factor</i>	Multiplicador para la profundidad
8	Entero	<i>Form Length</i>	Medida del ancho del <i>landmark</i>
9	Entero	<i>Blank Spacing</i>	Medida de distancia entre <i>landmarks</i>



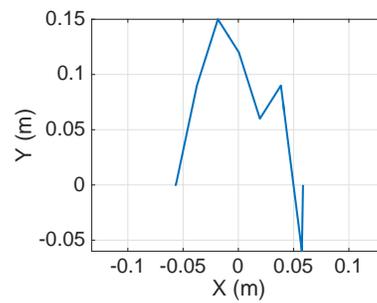
(A) Genoma [1,1,0,0,-1,-1,0.03,20,500].



(B) Genoma [5,-1,-4,5,1,-6,0.01,40,200].

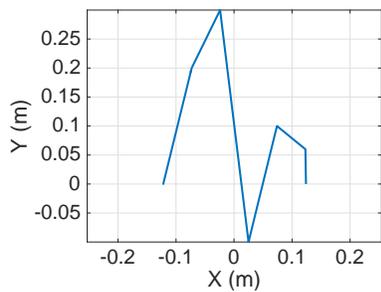


(C) Genoma [5,4,3,2,1,-15,0.03,80,1000].

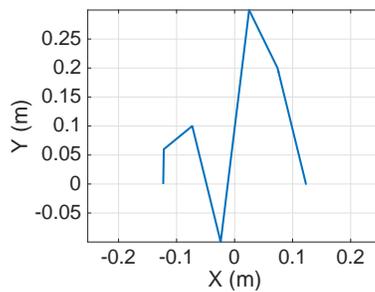


(D) Genoma [3,2,-1,-2,1,-5,0.03,20,1200].

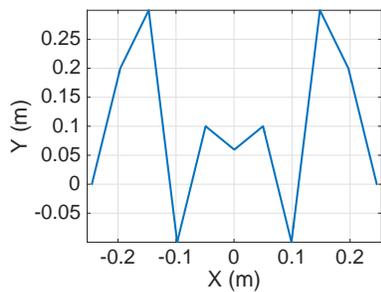
FIGURA 6.3. Ejemplos de genomas de forma no determinada no reflejados y no simétricos.



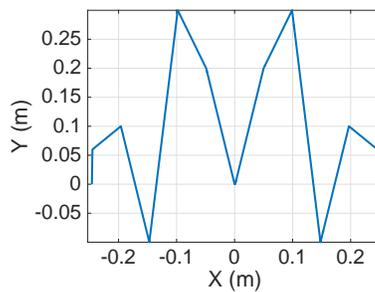
(A) No reflejado, no simétrico.



(B) Reflejado, no simétrico.



(C) No reflejado, simétrico.



(D) Reflejado, simétrico.

FIGURA 6.4. Ejemplos de *landmarks* de forma no determinada de igual genoma.

## 6.2. Función Objetivo

Para determinar qué *landmarks* obtienen una mejor estimación, es necesario establecer una función de desempeño, la cual debe ser evaluable al simular y de forma experimental. La medida seleccionada corresponde al error de posición promedio.

### 6.2.1. Error de Posición Promedio

El error de posición promedio se define como el promedio de la distancia entre la posición estimada y la posición real en cada instante de medición como se define en la ecuación 6.1. Esta medida es utilizada para evaluar el desempeño de los *landmarks* dentro del algoritmo de estimación de posición.

$$E_{pp} = \frac{1}{n} \sum_{i=1}^n \|\hat{p}_i - \bar{p}_i\| \quad (6.1)$$

Para calcular esta distancia es necesario saber para el mismo instante de tiempo  $i$ , la posición real  $\bar{p}$  y la posición estimada  $\hat{p}$ ; el simulador es capaz de generar ambos datos en cualquier instante de tiempo, sin embargo para la realización de experimentos es necesario corregir la estimación de posición real  $\bar{p}$  como se indica en la sección 7.

## 6.3. Parámetros Utilizados

Las simulaciones se ejecutaron de forma independiente para cada caso del genoma de forma primitiva y para cada caso del genoma con figura no determinada, los casos son numerados de la siguiente forma:

- P1 - Forma primitiva triangular.
- P2 - Forma primitiva rectangular.
- P3 - Forma primitiva cúpula.
- P4 - Forma primitiva cuña.
- F1 - Forma no determinada.
- F2 - Forma no determinada inversa.
- F3 - Forma no determinada reflejada.

F4 - Forma no determinada inversa reflejada.

Cada simulación se ejecuta utilizando como algoritmo de *matching* ICP y Hausdorff de forma independiente en cada caso, generando los resultados de error de posición promedio indicados en la tabla 6.4.

El rango de parámetros utilizado fue elegido para permitir una gran variabilidad de figuras con una profundidad desde la pared no mayor a 0,3 m. para prevenir interferir el paso del vehículo. Los rangos se indican en la tabla 6.3

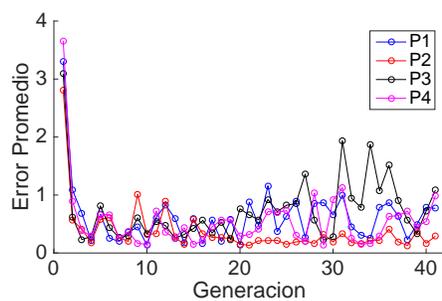
TABLA 6.3. Rango de valores para genomas de simulación.

		G1	G2	G3	G4	G5	G6	G7	G8	G9
Forma Primitiva	Mínimo	0.01	1	100	-	-	-	-	-	-
	Máximo	0.3	60	1000	-	-	-	-	-	-
Forma No Determinada	Mínimo	-2	-2	-2	-2	-2	-2	0.01	5	600
	Máximo	2	2	2	2	2	2	0.07	15	750

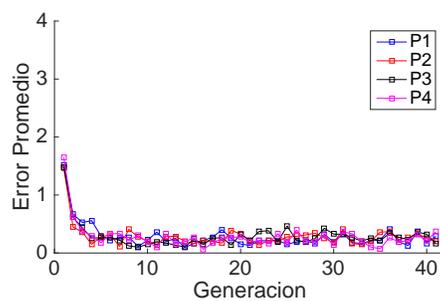
## 6.4. Resultados y Análisis

### 6.4.1. Progresión de la Población

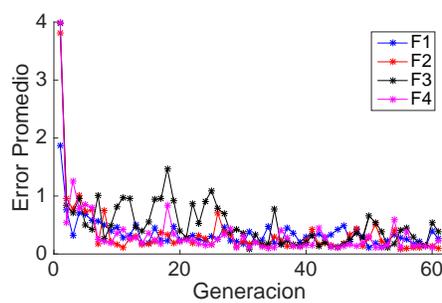
El error de posición promedio de la población a través de las generaciones se muestra en los gráficos 6.5 y 6.6, donde se observa que el desempeño de los individuos mejora a medida que avanzan las generaciones para terminar oscilando cerca de un valor estable. Los gráficos 6.5 y 6.6 indican que el resultado mejora a través de las generaciones del algoritmo genético hasta llegar a un nivel de estabilidad donde oscila y no tiende a mejorar, mientras que los gráficos 6.7 y 6.8 indican que la varianza de los genomas de los individuos disminuyen con la progresión de generaciones y que el espacio de soluciones se optimiza con el algoritmo genético.



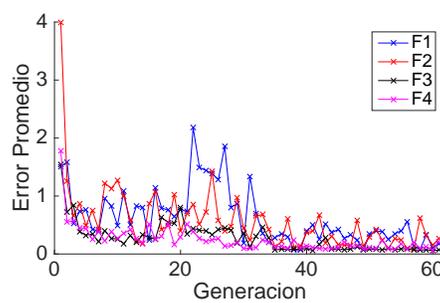
(A) ICP - Forma primitiva.



(B) Hausdorff - Forma primitiva.

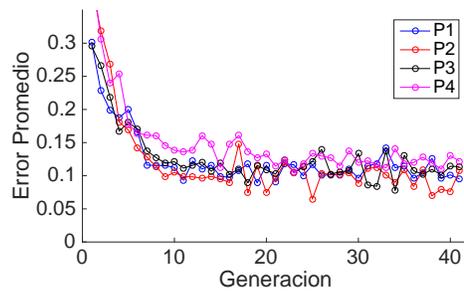


(C) ICP - Forma no determinada.

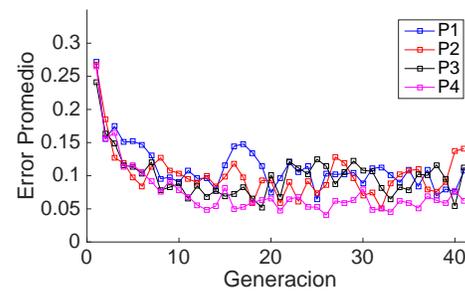


(D) Hausdorff - Forma no determinada.

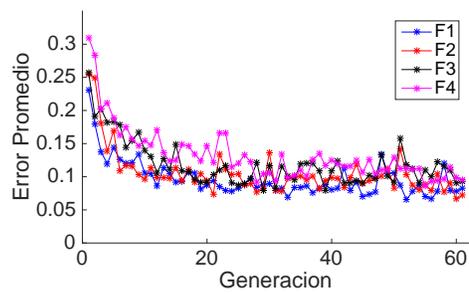
FIGURA 6.5. Progresión del promedio de error de posición promedio por generación.



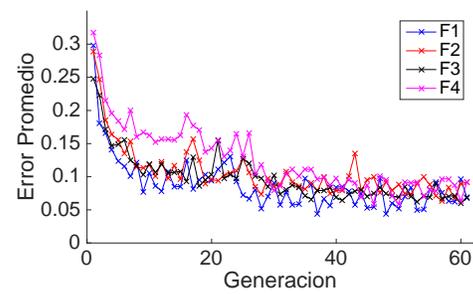
(A) ICP - Forma primitiva.



(B) Hausdorff - Forma primitiva.

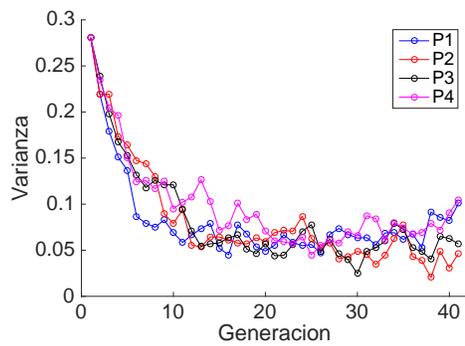


(C) ICP - Forma no determinada.

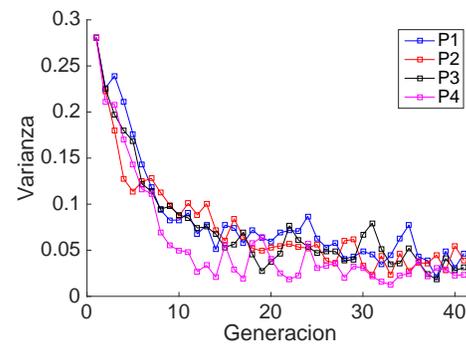


(D) Hausdorff - Forma no determinada.

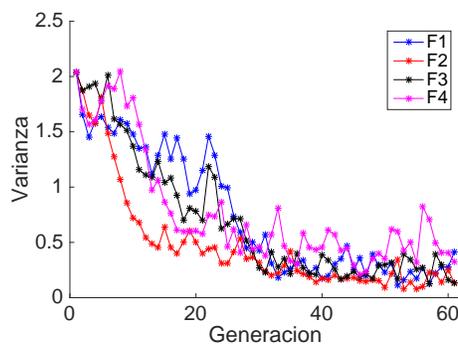
FIGURA 6.6. Progresión del promedio de error de posición promedio por generación utilizando miembros con desempeño  $< 1m$ .



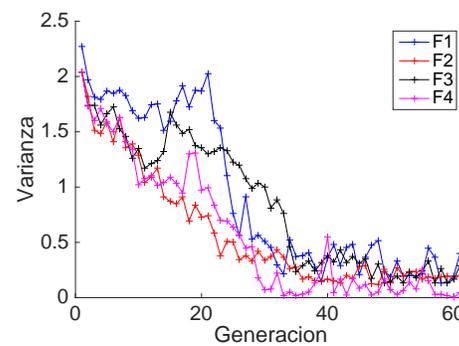
(A) ICP - Forma primitiva.



(B) Hausdorff - Forma primitiva.

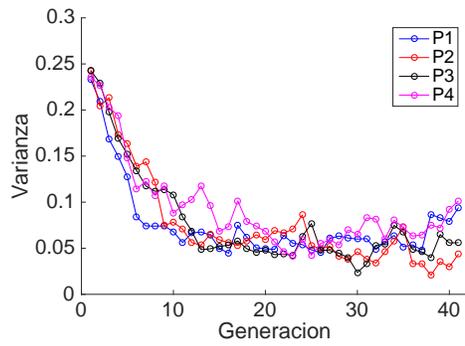


(C) ICP - Forma no determinada.

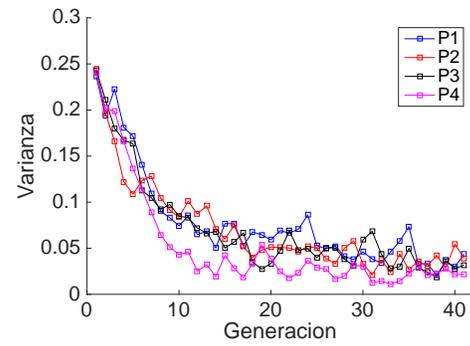


(D) Hausdorff - Forma no determinada.

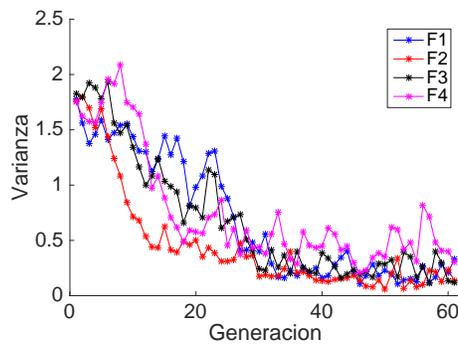
FIGURA 6.7. Progresión de la traza de la matriz de covarianza del genoma por generación.



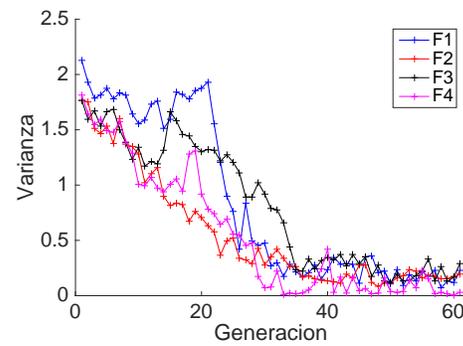
(A) ICP - Forma primitiva.



(B) Hausdorff - Forma primitiva.



(C) ICP - Forma no determinada.



(D) Hausdorff - Forma no determinada.

FIGURA 6.8. Progresión de la traza de la matriz de covarianza del genoma por generación utilizando miembros con desempeño  $< 1m$ .

## 6.4.2. Resultados Óptimos

El individuo óptimo para cada caso es calculado promediando el valor del genoma de cada individuo de la última generación que está bajo un valor umbral de 1 m en su error de posición promedio, su desempeño es calculado al ser sometido a la simulación se obtiene su valor de desempeño indicado en la tabla 6.4.

El desempeño en cada caso se ubica entre 1,1 % y 0,1 % del desplazamiento total si-

TABLA 6.4. Error de posición promedio para el individuo promedio óptimo en cada caso.

	ICP	Hausdorff		ICP	Hausdorff
Triangular	0,0424	0,0216	No determinada	0,0731	0,0522
Rectangular	0,0905	0,0260	Reflejada	0,0434	0,0642
Cúpula	0,0627	0,0281	Inversa	0,0332	0,0675
Cuña	0,0645	0,0255	Inversa y Reflejada	0,0348	0,0236

mulado. El espacio de solución de cada variable de la última generación nos permite analizar la influencia que tienen los distintos factores del diseño del *landmark* en su desempeño:

### 6.4.2.1. Forma Primitiva

En las figuras 6.9, 6.10 y 6.11 se muestra histogramas de los genes correspondientes a los resultados de forma primitiva, comparando la generación final utilizando ICP y Hausdorff como técnicas de *matching* y como referencia la generación inicial.

El espacio de soluciones de las formas primitivas tienden a tener profundidad, ancho y distancia acotados dentro del rango de prueba como se muestra en los histogramas 6.9, 6.10 y 6.11: del rango completo posible la profundidad tiende a ser cercana a 0,15 m, el ancho tiende a tener un ancho de 20 unidades (0,1 m) y la distancia entre *landmarks* tiende a ser la mayor posible cercana a 1 m.

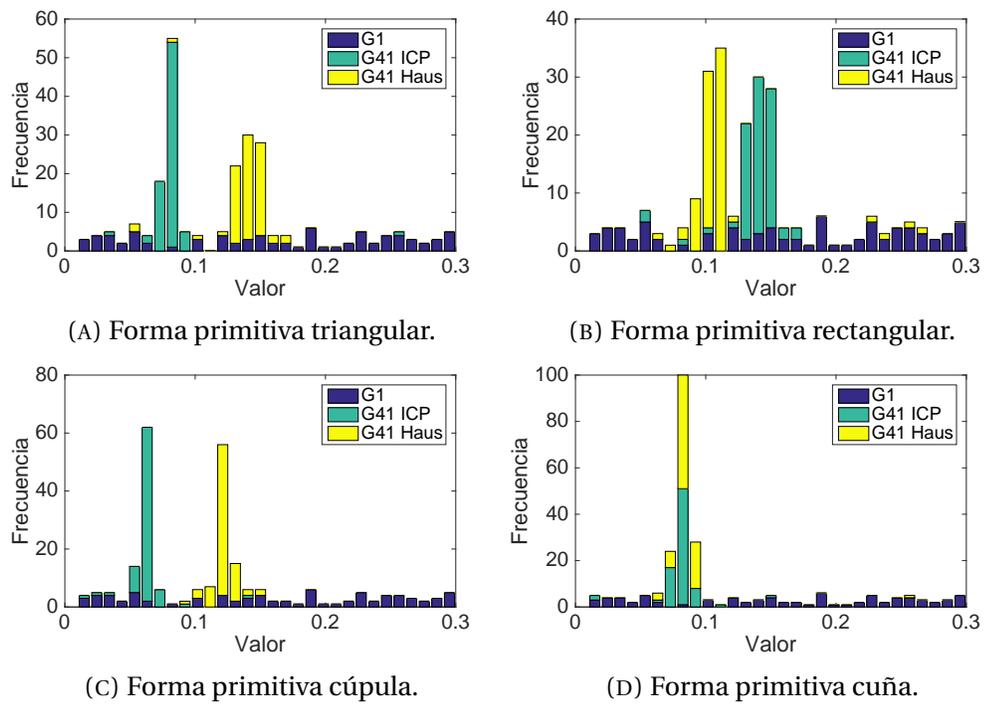
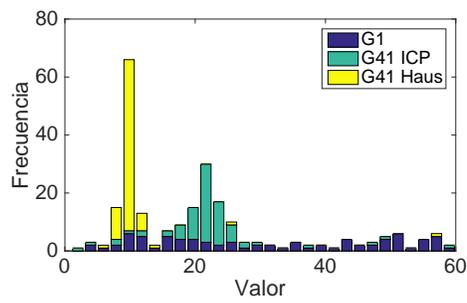
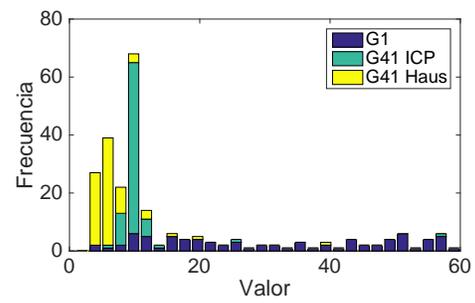


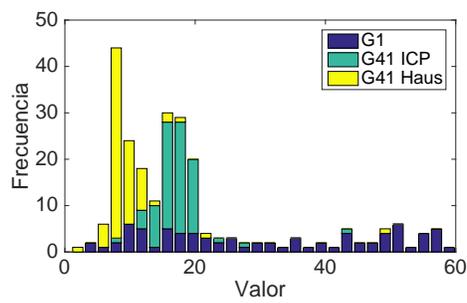
FIGURA 6.9. Histograma de la población según generación y método para el gen n°1 - Factor de escalamiento.



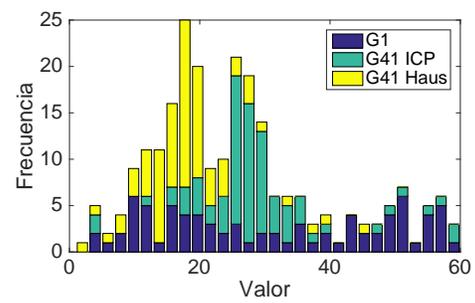
(A) Forma primitiva triangular.



(B) Forma primitiva rectangular.

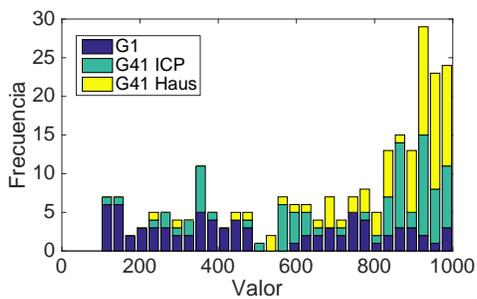


(C) Forma primitiva cúpula.

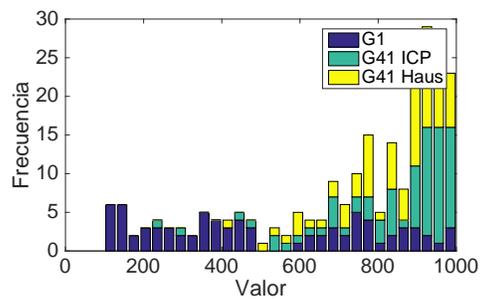


(D) Forma primitiva cuña.

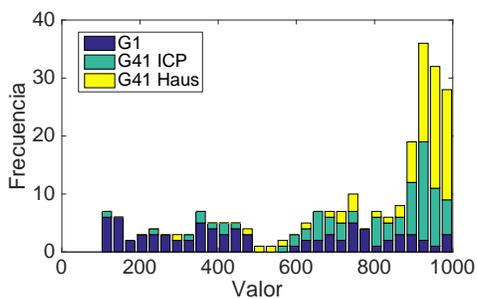
FIGURA 6.10. Histograma de la población según generación y método para el gen  $n^{\circ}2$  - Ancho de la forma.



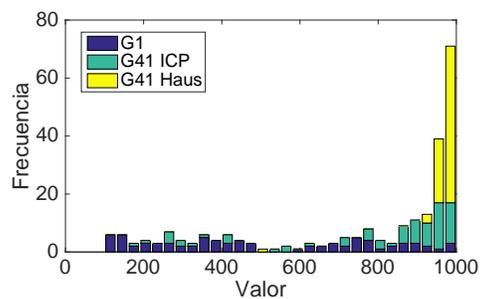
(A) Forma primitiva triangular.



(B) Forma primitiva rectangular.



(C) Forma primitiva cúpula.



(D) Forma primitiva cuña.

FIGURA 6.11. Histograma de la población según generación y método para el gen n°3 - Distancia entre Landmarks.

### 6.4.2.2. Forma No Determinada

En las figuras 6.12, 6.13 y 6.14 se indican histogramas de los genes correspondientes a la dimensión para la forma no determinada, comparando la generación final utilizando ICP y Hausdorff como técnicas de *matching* y como referencia la generación inicial. En las figuras 6.15 y 6.16 se muestran las figuras sobrepuestas por las generaciones finales normalizadas en profundidad y ancho, para comparar visualmente el resultado de los genes 1 al 6. Finalmente en las figuras 6.17 y 6.18 se muestra la superposición de los *landmarks* de la última generación de cada simulación sin normalizar incluyendo información sobre la dimensión de la respuesta.

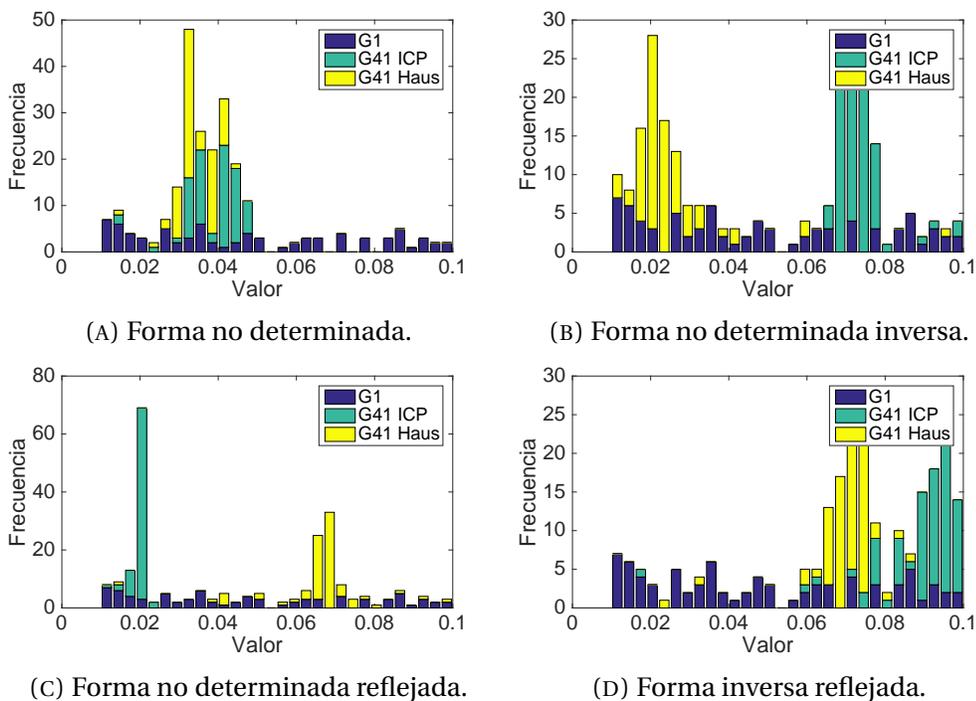
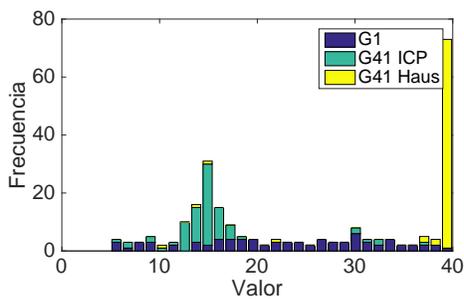
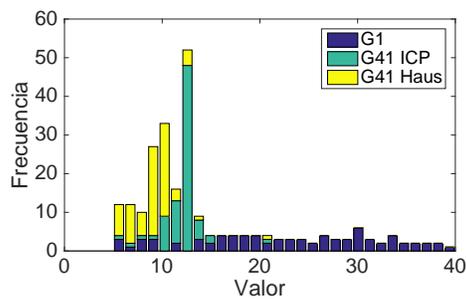


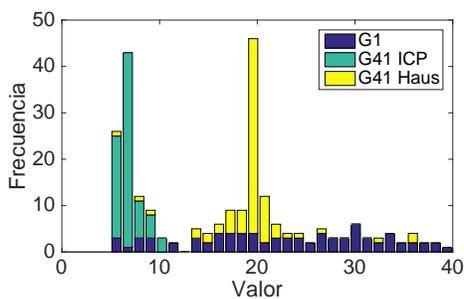
FIGURA 6.12. Histograma de la población según generación y método para el gen n°7 - Factor de escalamiento.



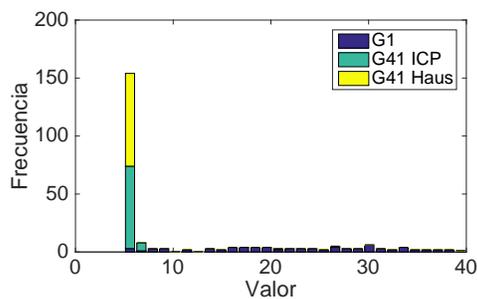
(A) Forma no determinada.



(B) Forma no determinada inversa.

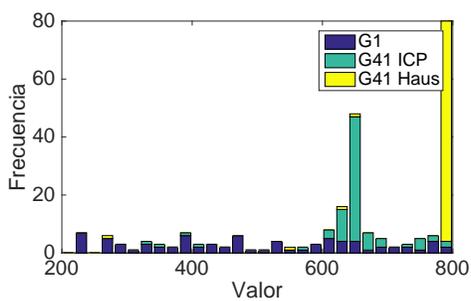


(C) Forma no determinada reflejada.

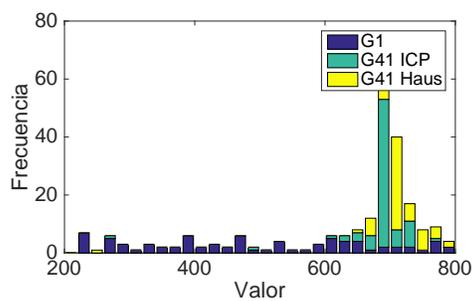


(D) Forma inversa reflejada.

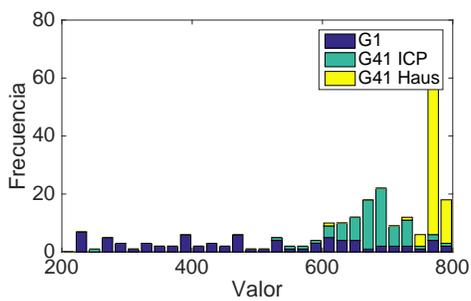
FIGURA 6.13. Histograma de la población según generación y método para el gen n°8 - Ancho de la forma.



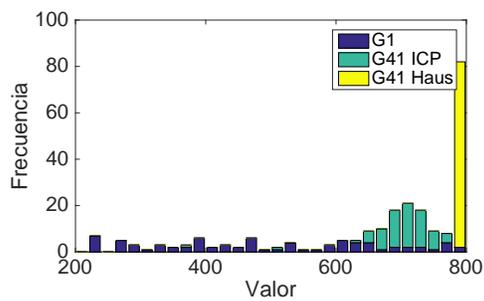
(A) Forma no determinada.



(B) Forma no determinada inversa.

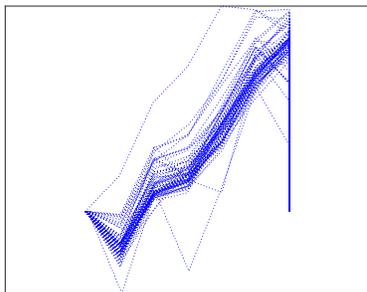


(C) Forma no determinada reflejada.

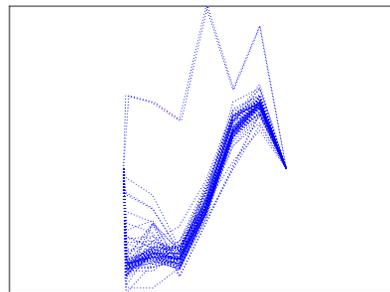


(D) Forma inversa reflejada.

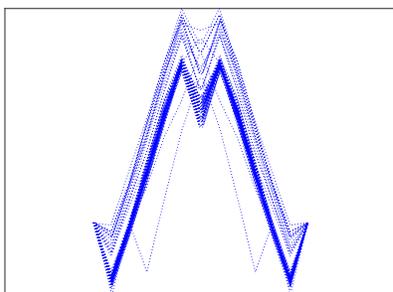
FIGURA 6.14. Histograma de la población según generación y método para el gen n°9 - Distancia entre Landmarks.



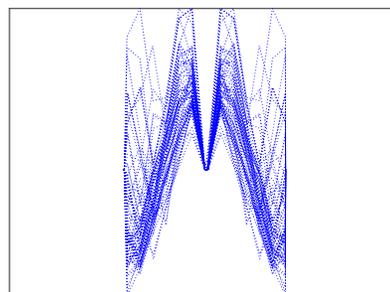
(A) Forma no determinada.



(B) Forma no determinada inversa.

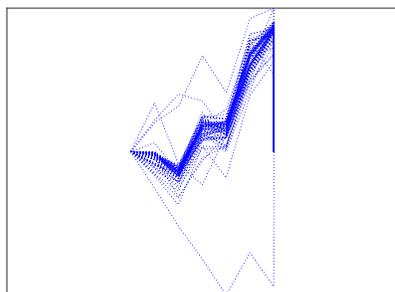


(C) Forma no determinada reflejada.

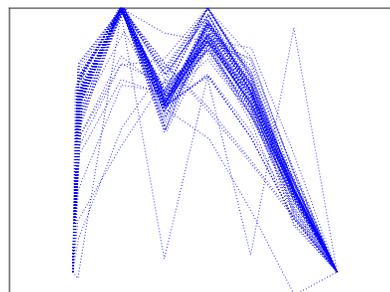


(D) Forma inversa reflejada.

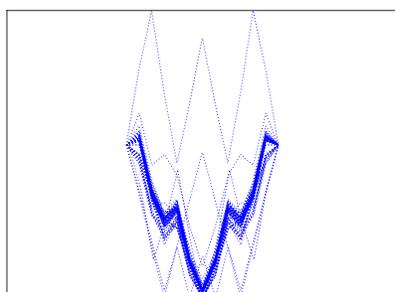
FIGURA 6.15. Formas normalizadas de la última generación obtenida con ICP.



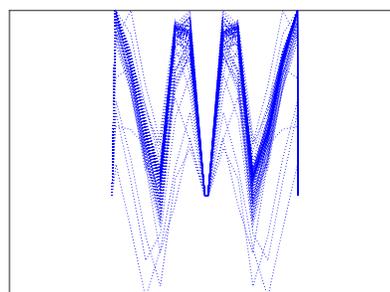
(A) Forma no determinada.



(B) Forma no determinada inversa.

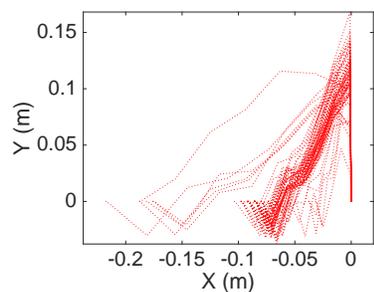


(C) Forma no determinada reflejada.

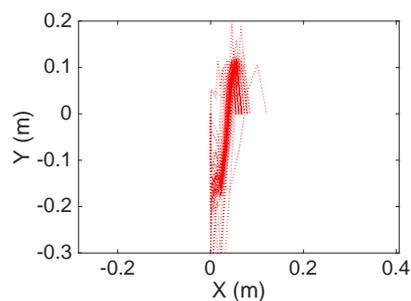


(D) Forma inversa reflejada.

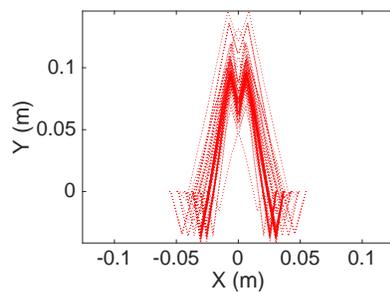
FIGURA 6.16. Formas normalizadas de la última generación obtenida con Hausdorff.



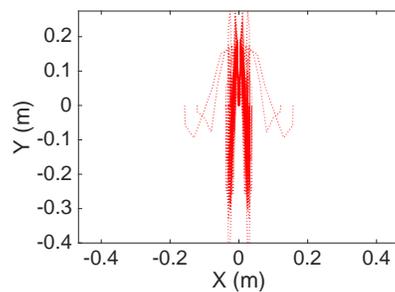
(A) Forma no determinada.



(B) Forma no determinada inversa.

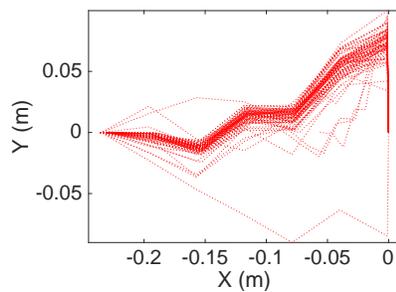


(C) Forma no determinada reflejada.

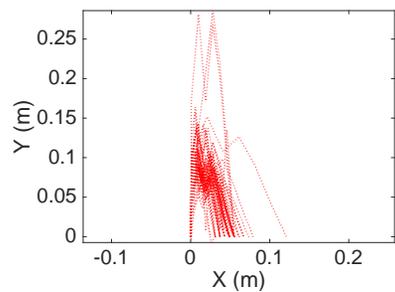


(D) Forma inversa reflejada.

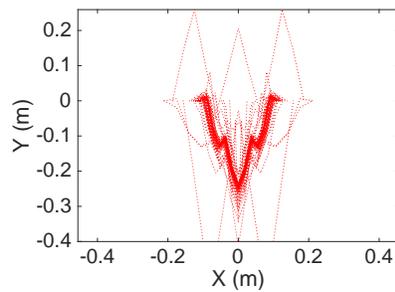
FIGURA 6.17. Landmarks de la última generación obtenida con ICP.



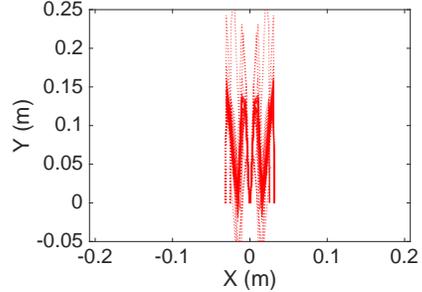
(A) Forma no determinada.



(B) Forma no determinada inversa.



(C) Forma no determinada reflejada.



(D) Forma inversa reflejada.

FIGURA 6.18. Landmarks de la última generación obtenida con Hausdorff.

Con respecto a las formas no determinadas se observa en los histogramas 6.13 y 6.14 que los individuos se centran en un espacio de soluciones reducido, con el ancho y la distancia entre *landmarks* cercano a 0,1 m y 1 m, exceptuando el caso de forma no determinada sin variaciones utilizando Hausdorff (figura 6.13a).

Mientras que en los histogramas 6.12 no se puede determinar un patrón para el genoma de profundidad, los *landmarks* acumulados en las figuras 6.17 y 6.18 indican que la mayoría de las soluciones tienden a estar entre 0,1 m y 0,15 m. Esta discrepancia es debido a que la profundidad es definida por los genomas de la forma y por el genoma de profundidad de forma conjunta.

En las figuras 6.15 y 6.16 se muestra la acumulación de formas óptimas sin considerar sus dimensiones, lo que muestra que no hay una relación entre las formas optimizadas encontradas en cada iteración de la simulación. La mayoría de los resultados corresponden a *landmarks* con una mayor longitud en profundidad que en ancho y considerando los desempeños similares entregados en la tabla 6.4.

Junto con la baja relación entre las figuras resultantes se puede inferir que los parámetros que definen la dimensionalidad tienen relevancia para mejorar el desempeño, mientras que no hay relevancia en la forma del *landmark*.

El método de *matching* utilizado tampoco genera diferencias significativas en el desempeño, sin embargo Hausdorff obtiene mejores resultados en la mayoría de los casos.

## 7. IMPLEMENTACIÓN EXPERIMENTAL

Los experimentos realizados consisten en calcular la estimación de la localización de la plataforma  $\bar{X}$  utilizando los algoritmos explicados anteriormente, mediciones reales del sistema LIDAR y el registro de las señales de control. Esta medición es comparada con la estimación de la posición real  $\bar{\bar{X}}$  obtenida con un sistema GPS RTK. Para medir el desempeño de los experimentos se requiere calcular el error de posición promedio para el mismo instante, pero el instante de muestra difiere como se indica en la figura 7.1. Para resolver este problema se utiliza una regresión lineal para interpolar y extrapolar la posición real de la toma de datos del LIDAR. Un ejemplo de esta estimación se muestra en la figura 7.2.

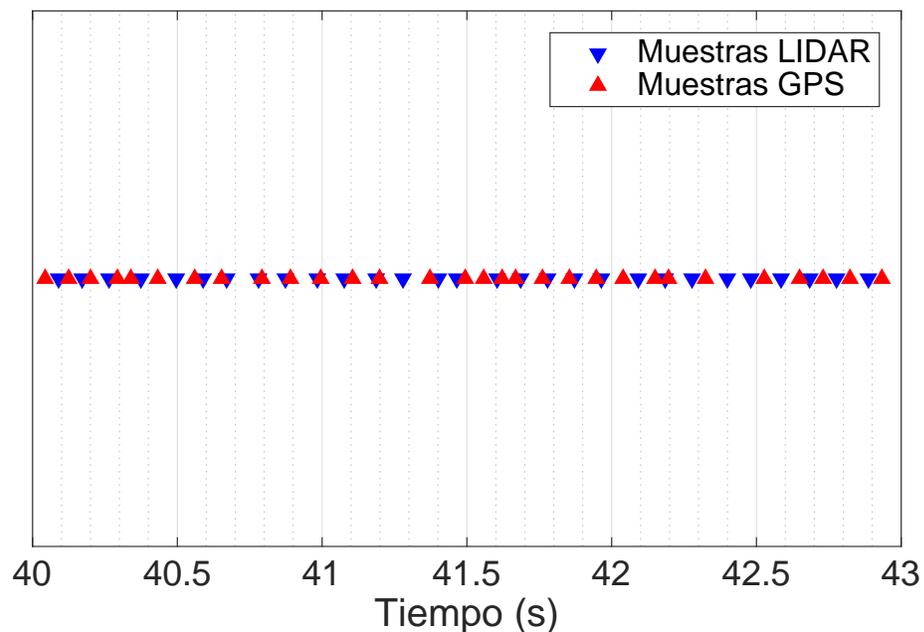


FIGURA 7.1. Segmento temporal de toma de muestras para sensores, experimento 2 repetición 1.

### 7.1. Experimentos a Realizar

Los casos a evaluar se limitaron a los siguientes:

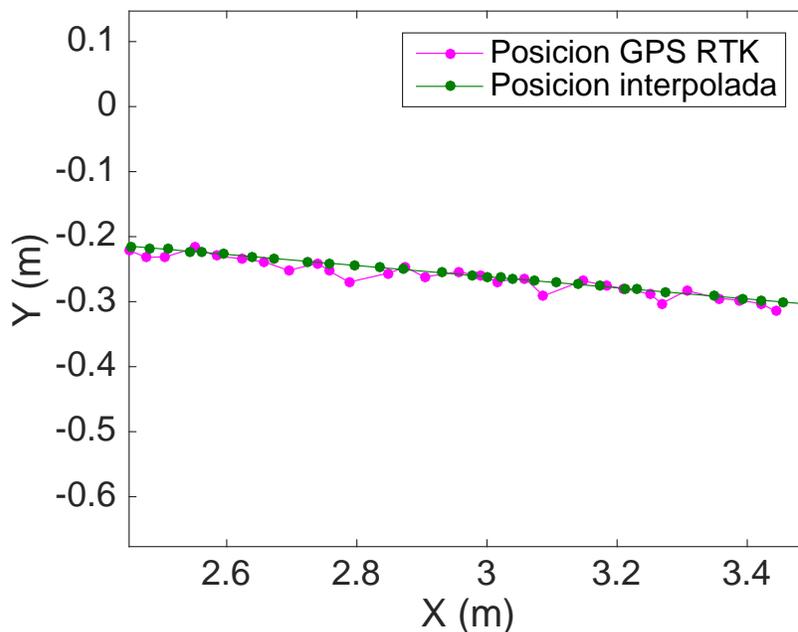


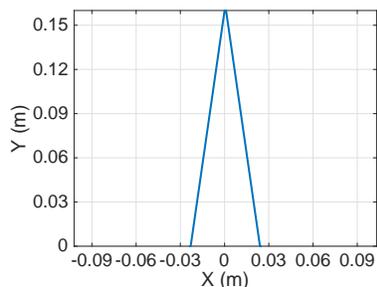
FIGURA 7.2. Segmento de estimación de posición de toma de muestras para sensores, experimento 2 repetición 1.

1. Túnel recto sin *landmarks*.
2. Túnel recto con *landmarks* definidos como el mejor resultado del algoritmo genético para genomas de formas primitivas.
3. Túnel recto con *landmarks* definidos como el mejor resultado del algoritmo genético para genoma de forma no determinada.

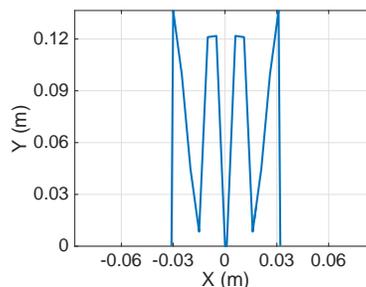
Los experimentos buscan validar el desempeño de las figuras encontradas para comprobar que se diseñó un sistema optimizado. Se numeran del 1 al 3 dependiendo de los *landmarks* utilizados, los cuales corresponden a los indicados en la figura 7.3 y sus genomas se indican en la tabla 7.1. Para efectos prácticos se denomina experimento a cada uno de estas tres combinaciones del escenario y *landmark*, y se denomina repetición a cada realización del experimento, el cual consiste en recorrerlo en un sólo sentido una vez de principio a fin.

TABLA 7.1. Genomas de Landmarks utilizados para los experimentos.

N°	Tipo	G1	G2	G3	G4	G5	G6	G7	G8	G9
1	Sin <i>Landmark</i>	-	-	-	-	-	-	-	-	-
2	Primitivo	0,16	9,43	893	-	-	-	-	-	-
3	No determinado	1,74	-0,01	-1,61	0,52	0,78	0,53	0,07	6	797



(A) Experimento 2.



(B) Experimento 3.

FIGURA 7.3. Formas construidas para validar experimentalmente.

## 7.2. Plataforma de Prueba

Los experimentos fueron realizados utilizando como vehículo un cargador frontal CAT262 (figura 7.4), el cual es un vehículo industrial utilizado en diversos ámbitos productivos, incluyendo labores de acarreo de material al interior de túneles subterráneos, otorgándole mayor validez a los experimentos en comparación al utilizar una plataforma de laboratorio.

Este cargador no fue diseñado para integrar los sensores, actuadores y componentes necesarios para realizar los experimentos por lo que fue acondicionado por el equipo de investigación RAL en varios aspectos como los siguientes:

- Instalación de puntos de montaje para sensores LIDAR, GPS RTK e IMU GPS.
- Interrupción de señales de movimiento internas para controlar el movimiento por control remoto *line of sight* y de forma programada.
- Lectura de encoders internos para obtener mediciones de *dead reckoning*
- Diseño de módulos y programas para unificación de lecturas y de control del vehículo.



FIGURA 7.4. Plataforma de prueba CAT262C sin modificaciones.

Para obtener mediciones se utilizó un sensor LIDAR SICK LMS511, ubicado en la parte frontal del vehículo, además de utilizar las señales de control enviadas al sistema de control interno del CAT262. Como referencia de posición se utilizó un GPS RTK Swift-Nav Piksi, colocado en el techo del vehículo, mientras que la estación de referencia correspondiente fue colocada en una posición estática fuera del escenario de prueba durante los experimentos.

El GPS RTK tiene una precisión de 1 cm a 5 cm con respecto a la base utilizada (Swift Navigation, 2015), mientras que el sensor LIDAR LMS511 tiene un error sistemático de 35 mm y un error estadístico de 8 mm (Sick AG, s.f.).



FIGURA 7.5. Plataforma de prueba CAT262C con el sensor SICK LMS511 instalado.

### 7.3. Construcción del Escenario

Para la realización de los experimentos se construyó una estructura capaz de representar el escenario descrito en el problema, un túnel sin *features* naturales para los sensores a utilizar, Éste consiste en un pasillo recto de 9 m de largo y 3,6 m de ancho, suficiente espacio para permitir el máximo largo de *feature* permitido, el ancho del vehículo y una distancia adicional de seguridad. Por otra parte se dejó el techo del pasillo sin cubrir ya que no afecta el escenario de sensado y permite utilizar GPS RTK para estimar la posición de referencia.

Este pasillo fue construido con malla opaca al haz de luz del sensor LIDAR con una altura entre 0,4 y 1 m del suelo, para alcanzar a cubrir el rango del sensor y cubrir variaciones angulares. Además se instalaron *landmarks* removibles contruidos de cartón corrugado, también opacos al sensor de la longitud de la malla. El lado del túnel a la derecha del vehículo es el lado utilizado para estimar la posición, evitando la perdida de visión por posibles vehículos u obstáculos que pueden bloquear la línea de visión

al *landmark*.



FIGURA 7.6. Instalación de túnel sin landmarks.



FIGURA 7.7. Instalación de landmarks en túnel.

## 8. RESULTADOS EXPERIMENTALES

### 8.1. Resultados

En la imagen 8.1 se muestra una representación de una muestra capturada durante una repetición del experimento 2.

Los resultados estadísticos de los experimentos se muestran en las tablas 8.1, 8.2, 8.3

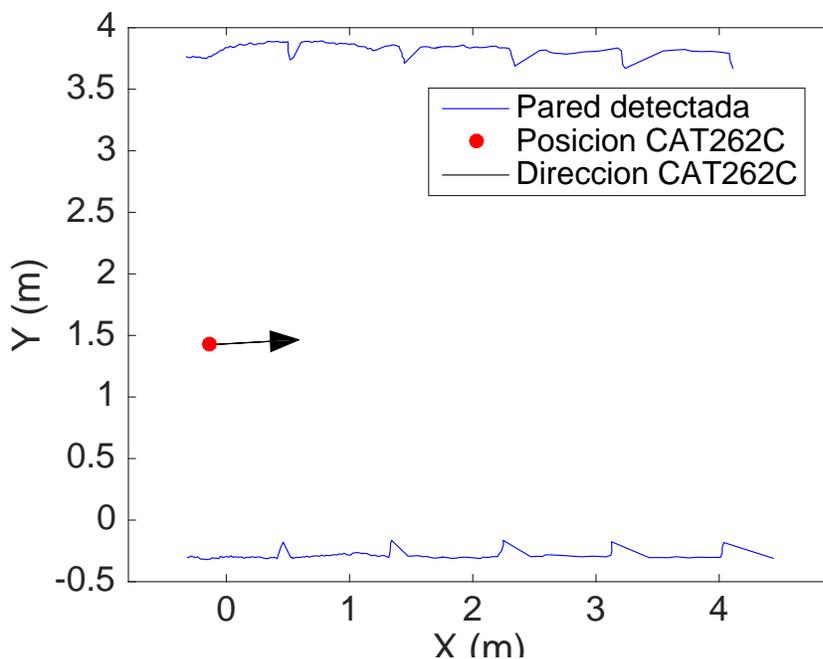


FIGURA 8.1. Muestra de datos obtenidos por LIDAR en una repetición del experimento 2.

y 8.4.

TABLA 8.1. Resultados experimentales - Error de posición promedio.

	Experimento 1		Experimento 2		Experimento 3	
	ICP	Hausdorff	ICP	Hausdorff	ICP	Hausdorff
Repeticiones	10	10	22	22	15	15
Media Error (m)	20,765	19,748	0,258	0,235	0,206	0,219
Desviación estándar (m)	7,596	5,2134	0,111	0,083	0,189	0,184
Error estándar media (m)	2,402	1,649	0,0236	0,017	0,048	0,047

De los resultados se puede observar que para el caso del experimento 1, la estimación de posición falla de forma notable generando un error de posición promedio mayor al 200% de la traslación total. En el experimento 2 y 3 la estimación de posición tiene un error entre 2,5% y 2,8% del desplazamiento total. La desviación obtenida se ubica en un rango de 0,9% y 2,1% del desplazamiento total.

Comparando los casos de estudio del experimento 2 y 3, se observa que la diferencia de desempeño de la media entre ambos casos es de 0,052 m para ICP y 0,016 m para Hausdorff, lo cual corresponde a 0,58% y 0,18% del desplazamiento total.

TABLA 8.2. Resultados Experimento 1 por repetición - Error de posición promedio (m).

Repetición	Experimento 1	
	ICP	Hausdorff
1	31,2174	26,9502
2	20,2106	21,5888
3	33,1035	16,3646
4	13,3947	12,7168
5	20,0499	20,98
6	20,3869	26,9231
7	16,2143	15,6725
8	10,2477	12,938
9	27,3472	23,2665
10	15,4819	20,0746

TABLA 8.3. Resultados Experimento 2 por repetición - Error de posición promedio (m).

Repetición	Experimento 2	
	ICP	Hausdorff
1	0,1126	0,1803
2	0,1685	0,3116
3	0,1681	0,1688
4	0,1229	0,1033
5	0,2798	0,1338
6	0,2181	0,2013
7	0,2529	0,1743
8	0,2885	0,1301
9	0,3744	0,294
10	0,4255	0,2574
11	0,1243	0,2152
12	0,1805	0,1384
13	0,2119	0,2577
14	0,4485	0,3057
15	0,2511	0,3584
16	0,1157	0,243
17	0,3219	0,2083
18	0,4721	0,281
19	0,2916	0,3439
20	0,2414	0,2069
21	0,4075	0,4304
22	0,2056	0,2335

TABLA 8.4. Resultados Experimento 3 por repetición - Error de posición promedio (m).

Repetición	Experimento 3	
	ICP	Hausdorff
1	0,2303	0,3357
2	0,0953	0,146
3	0,1347	0,1312
4	0,6178	0,6871
5	0,1374	0,1211
6	0,0585	0,1129
7	0,1281	0,0835
8	0,6809	0,6022
9	0,1252	0,1426
10	0,2018	0,1917
11	0,1056	0,1477
12	0,1365	0,1568
13	0,2592	0,1516
14	0,0415	0,0688
15	0,1403	0,218

En los gráficos 8.2, 8.3 y 8.4; se pueden observar las curvas de error absoluto de posición de cada experimento con el algoritmo basado en la distancia de Hausdorff:

$$E_p = |X_{GPS} - X_{algoritmo}| \quad (8.1)$$

$$E_p = |\overline{\overline{X}} - \overline{X}| \quad (8.2)$$

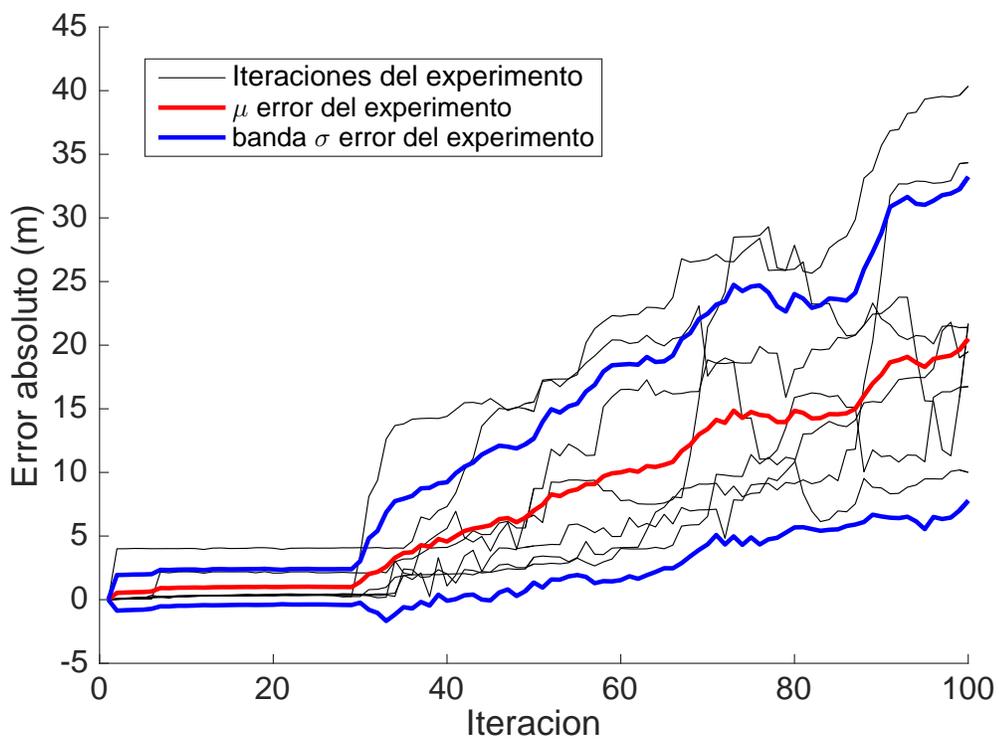


FIGURA 8.2. Error  $E_p$  para las repeticiones del experimento 1.

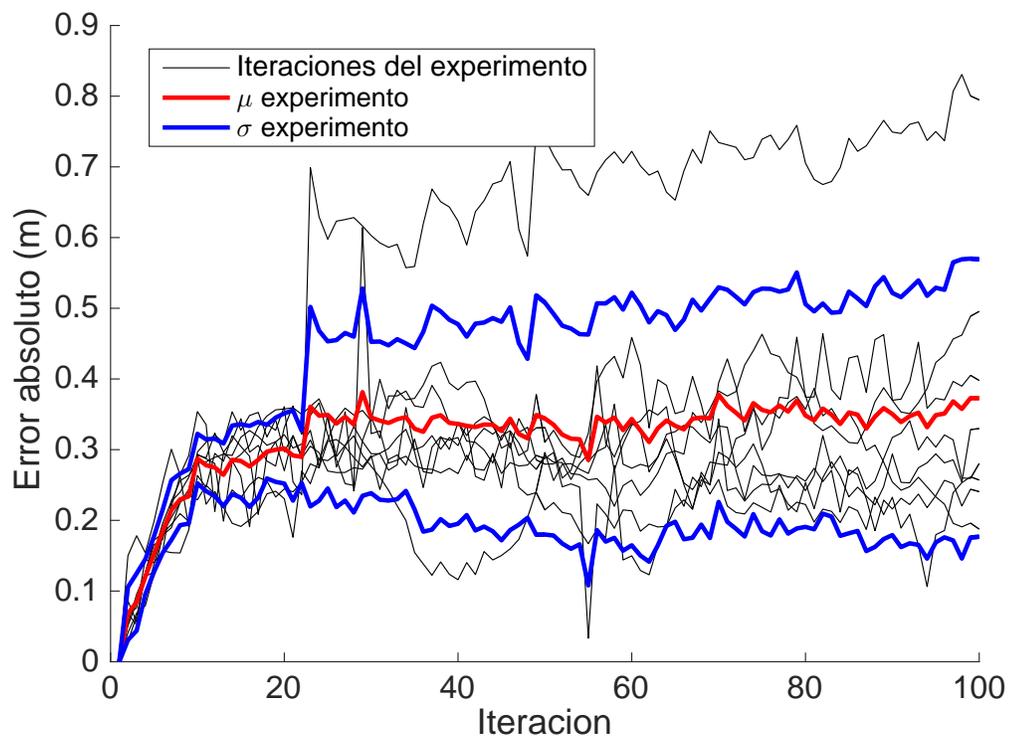


FIGURA 8.3. Error  $E_p$  para las repeticiones del experimento 2.

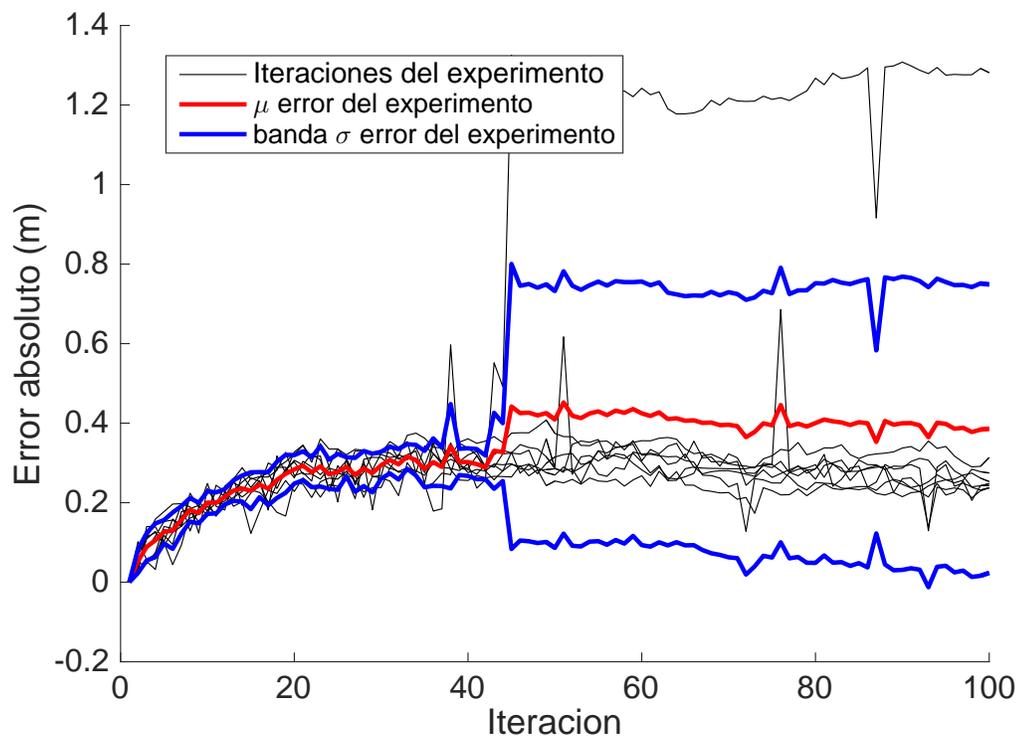


FIGURA 8.4. Error  $E_p$  para las repeticiones del experimento 3.

El promedio del error del vector de estado del vehículo, comparado con el GPS RTK como baseline se muestra en los gráficos 8.5 y 8.6. Siendo  $N$  la cantidad de repeticiones del experimento y  $k$  la iteración en al cual se mide el promedio la formula es la siguiente, siendo análoga para el caso del filtro de Kalman:

$$\mu_k = \frac{1}{N} \sum_{n=1}^N (|\overline{X_{nk}} - \overline{\overline{X_{nk}}}|) \quad (8.3)$$

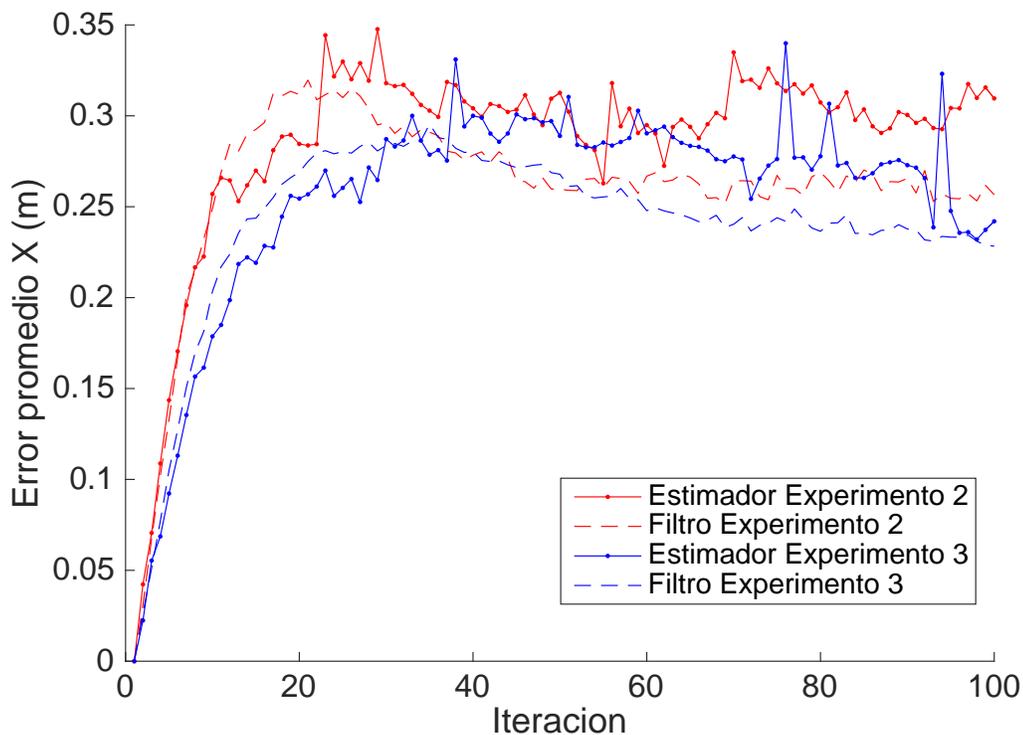


FIGURA 8.5. Error promedio de la posición X.

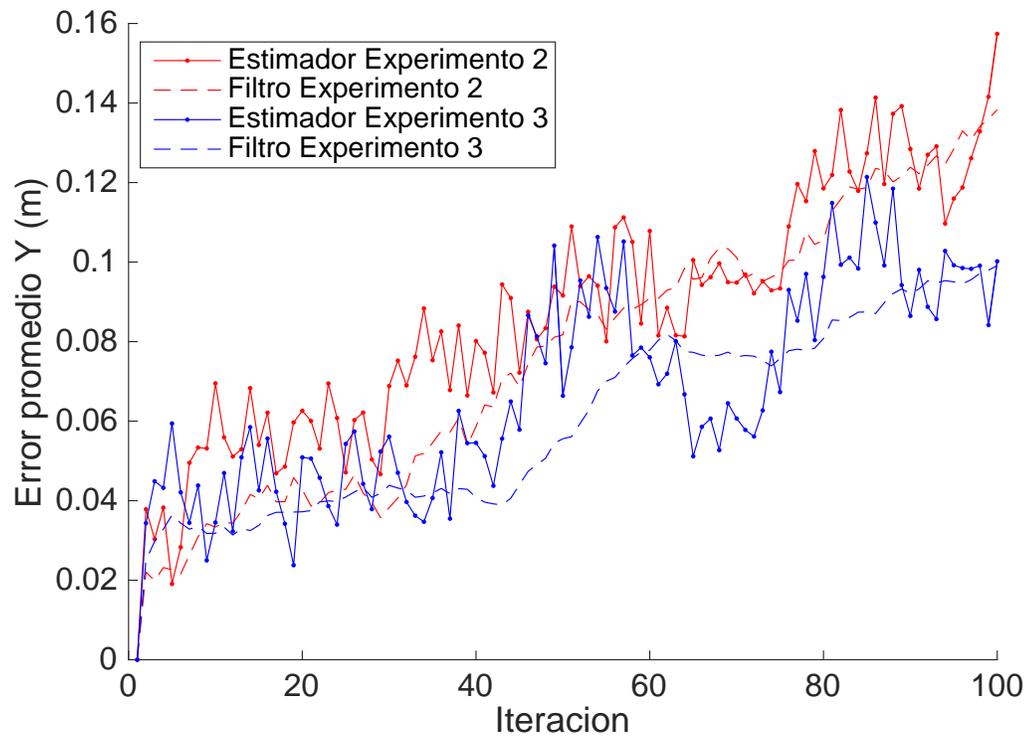


FIGURA 8.6. Error promedio de la posición Y.

Los valores correspondientes a la traza de la matriz de covarianza y su progresión según se muestran en los gráficos 8.8 y 8.7. Los resultados corresponden al promedio de las covarianzas de los errores de cada repetición del experimento, desde el inicio hasta la iteración indicada.

$$S_k = \frac{1}{N} \sum_{n=1}^N \left( \frac{1}{k} \sum_{i=1}^k (\bar{X}_{ni} - \overline{\bar{X}}_{ni})^2 \right) \quad (8.4)$$

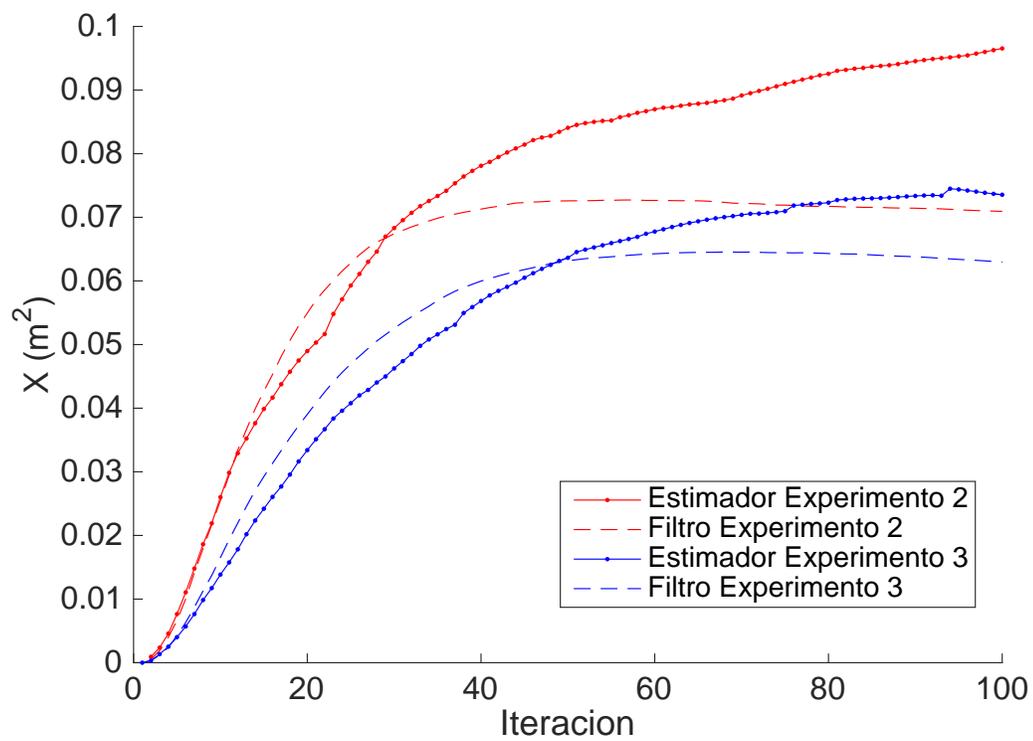


FIGURA 8.7. Progresión de la covarianza de la posición X.

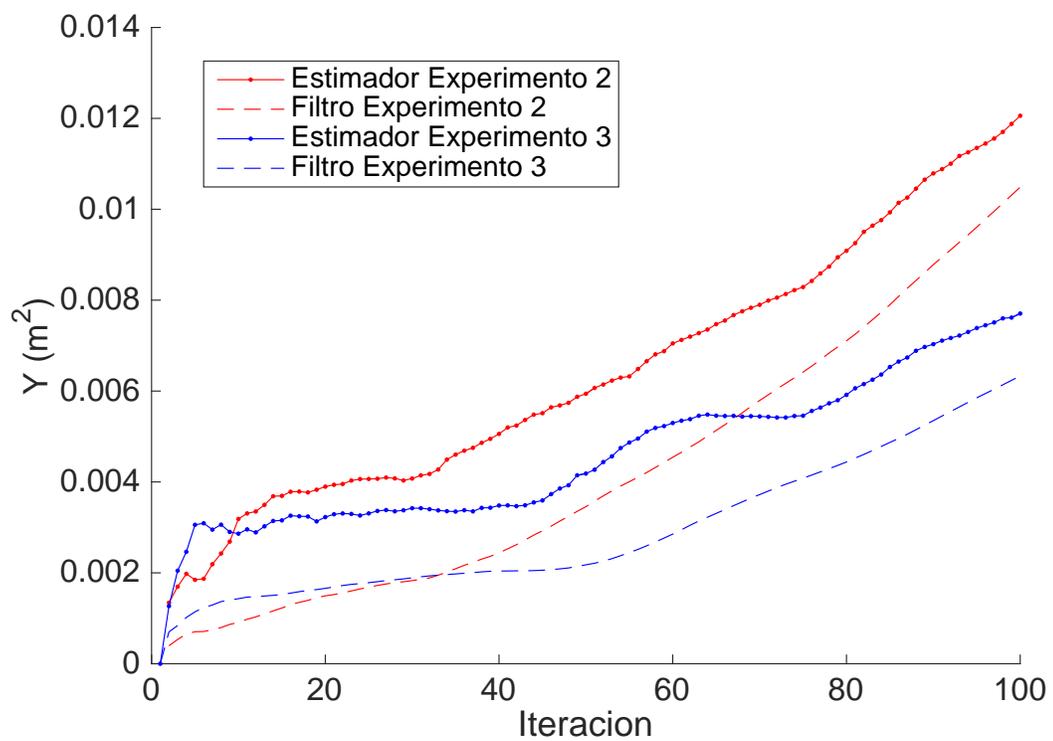


FIGURA 8.8. Progresión de la covarianza de la posición Y.

## 8.2. Análisis

De los resultados obtenidos es posible deducir que el método no converge al ser utilizado en un túnel sin *landmarks*, el algoritmo intenta agrupar puntos de *features* para determinar la ubicación general del *landmark*, sin embargo, al no existir detecta puntos arbitrarios a lo largo de la curva del túnel detectada.

El error de posición promedio obtenido experimentalmente es mayor que en los casos simulados respectivos manteniéndose en un rango porcentual menor a 3%.

Se infiere que los resultados de ámbos métodos (ICP y Hausdorff) de *matching* tienen un desempeño similar y dentro del mismo orden de magnitud por lo que son suficientes para lograr un *matching* adecuado para el funcionamiento del algoritmo, la elección no es relevante desde el punto de vista del desempeño.

Se detecta analizando el error absoluto que el sistema se ve afectado por fallas en el sistema de asignación de *landmarks*, el cual genera errores acumulados al asignar erróneamente la correspondencia similares a la distancia entre *landmarks*.

Ambos *landmarks* tienen un desempeño similar al ser comparados utilizando el mismo método de *matching*, por lo que la forma no es relevante desde el punto de vista del desempeño. Con respecto a los parámetros dimensionales ambos *landmarks* elegidos son muy similares por lo que no se puede concluir condiciones de diseño en este ámbito.

## 8.3. Reconstrucción de mapa y trayectoria

Para reconstruir ambas se desarrollo un filtro de Kalman para fusionar el estimador con el modelo del vehículo.

### 8.3.1. Filtro de Kalman

El modelo de movimiento utilizado es un modelo cinemático de movimiento para un vehículo skid-steer.

El modelo de medición de estado es el algoritmo el cual entrega directamente el vector de estado del sistema.

$$u = \begin{bmatrix} v_i \\ \omega_i \end{bmatrix} \quad (8.5)$$

$$X = \begin{bmatrix} x_i \\ y_i \\ \theta_i \end{bmatrix} = \begin{bmatrix} x_{i-1} + v_i \cdot \Delta t \cdot \cos(\theta_{i-1}) \\ y_{i-1} + v_i \cdot \Delta t \cdot \sin(\theta_{i-1}) \\ \theta_{i-1} + \Delta t \cdot \omega_i \end{bmatrix} \quad (8.6)$$

El modelo de medición de datos de LIDAR entrega datos en coordenadas polares, las cuales son transformadas en una nube de puntos cartesianas relativas al vehículo según el ángulo fijo que corresponde a las características del LIDAR.

$$\begin{bmatrix} x_m^r \\ y_m^r \end{bmatrix} = \begin{bmatrix} r_m \cdot \cos(\gamma_m) \\ r_m \cdot \sin(\gamma_m) \end{bmatrix} \quad (8.7)$$

Las mediciones relativas son llevadas a mediciones globales considerando la posición y orientación estimada

$$\begin{bmatrix} x_m^g \\ y_m^g \end{bmatrix} = R_\theta \cdot \begin{bmatrix} x_m^r \\ y_m^r \end{bmatrix} + \begin{bmatrix} x \\ y \end{bmatrix} \quad (8.8)$$

### 8.3.2. Reconstrucción de mapa

La siguiente reconstrucción del mapa (figuras 8.9 y 8.10) es realizada utilizando las mediciones de LIDAR corregidas con la posición del algoritmo de localización. Se utilizó sólo el algoritmo basado en la distancia de Hausdorff para este análisis ya que ambos métodos presentan resultados similares.

En rojo se muestra la media de las mediciones acumuladas a lo largo del eje Y, siendo la zona celeste corresponde a una desviación estándar de distancia de la medición.

Para evitar problemas de alineación en las repeticiones, en las figuras 8.9 y 8.10 se muestran 2 reconstrucciones de ejemplo de repeticiones específicas. En la tabla 8.5

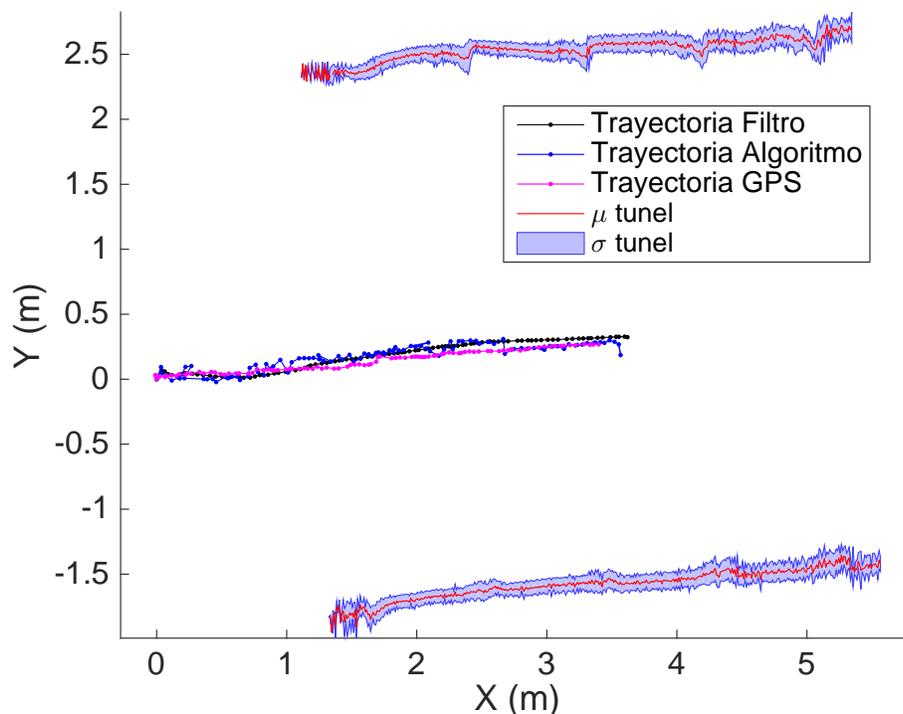


FIGURA 8.9. Reconstrucción del túnel, experimento n°2 repetición 4.

se muestra el promedio de las desviaciones estándar en metros del borde del mapa reconstruido según el experimento y el algoritmo utilizado. Para calcular la desviación estándar se usaron los  $N$  puntos detectados acumulados  $X_r = [X_1 \dots X_i]$  en una sección  $r$  de 1cm en el eje X con la siguiente ecuación:

$$S_r = \sqrt{\frac{1}{N_r - 1} \sum_{i=1}^{N_r} |\overline{X}_i - \mu|^2} \quad (8.9)$$

La diferencia entre la desviación del estimador y del filtro es la procedencia del vector de estado utilizado para convertir los puntos detectados de local a global.

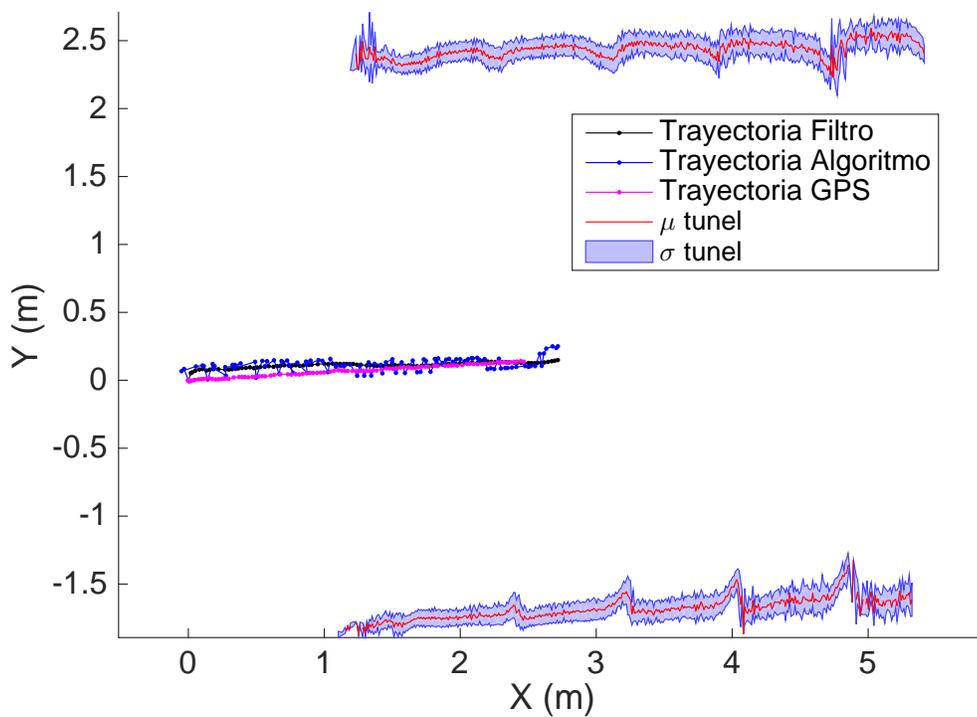


FIGURA 8.10. Reconstrucción del túnel, experimento n°3 repetición 4.

Debido a que en el experimento 1 no se puede reconstruir el mapa por la falta de *landmarks* por lo que la desviación estándar no representa correctamente una medida de error

TABLA 8.5. Tabla de datos promedio para la reconstrucción del túnel.

Experimento	Desviación Estimador (m)	Desviación Filtro Kalman (m)	Distancia recorrida (m)	Cantidad experimentos
N°1	0,6915	0,2059	3,3841	10
N°2	0,0700	0,0444	2,7155	22
N°3	0,0875	0,0429	2,5136	15

### 8.3.3. Reconstrucción del camino

Para reconstruir el camino se utilizó como baseline las mediciones de GPS RTK obtenidas en los experimentos, y como corrección se implementó un filtro de Kalman que combina la posición obtenida por el algoritmo y el modelo cinemático del vehículo. Para evitar problemas de visualización al traslapar todas las trayectorias, la posición según el filtro, el algoritmo y el GPS se muestran a modo de ejemplo en las siguientes figuras 8.11, 8.12 y 8.13 según el experimento.

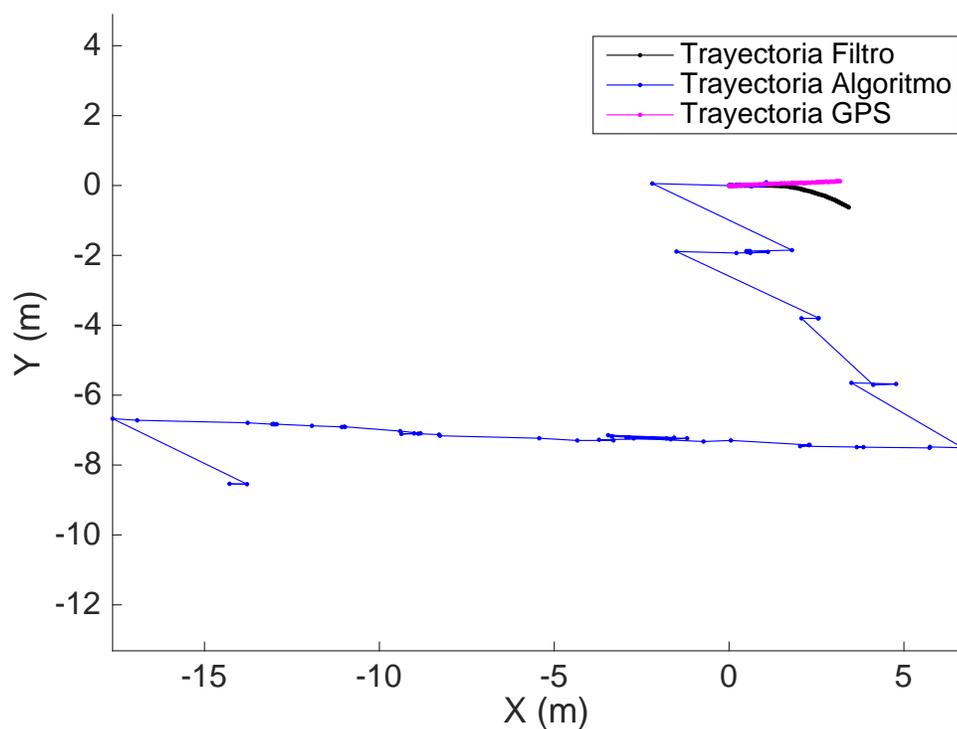


FIGURA 8.11. Reconstrucción de la trayectoria, experimento n°1 repetición 3.

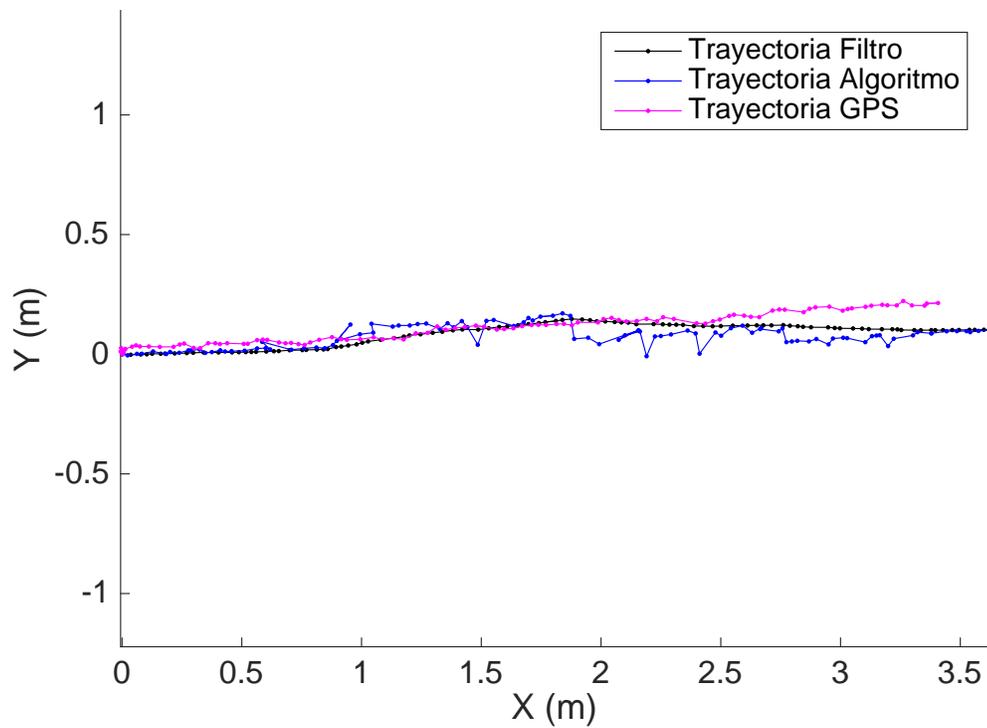


FIGURA 8.12. Reconstrucción de la trayectoria, experimento n°2 repetición 3.

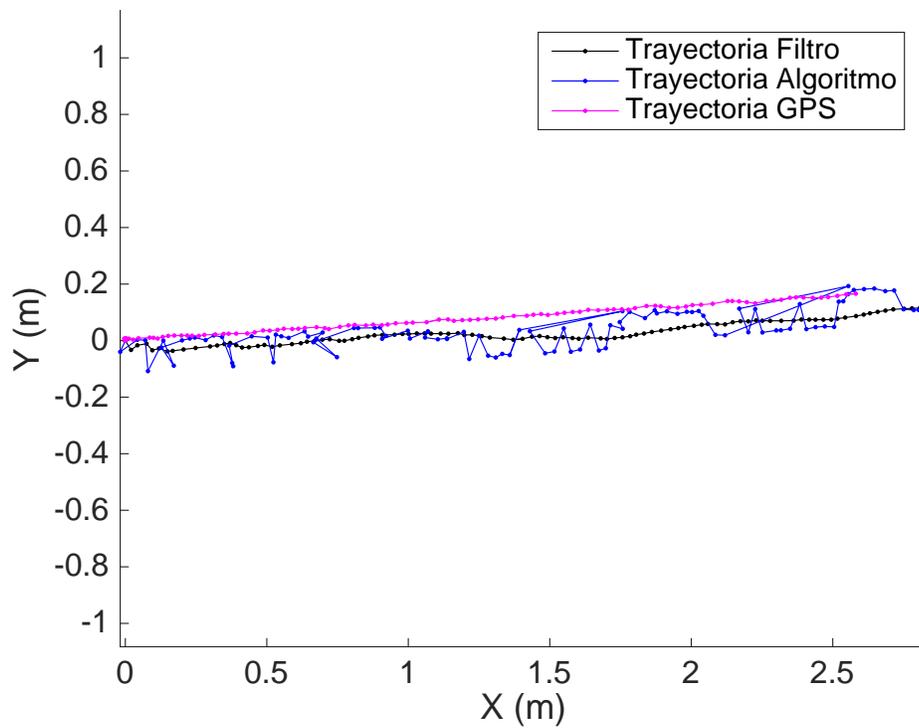


FIGURA 8.13. Reconstrucción de la trayectoria, experimento n°3 repetición 3.

## 9. CONCLUSIONES

### 9.1. Ventajas y desventajas comparativas

A diferencia de gran parte de los trabajos existentes en la literatura, los métodos de estimación de posición desarrollados en este trabajo funcionan en ambientes donde no se disponen de características naturales del terreno. En los trabajos existentes que utilizan características o *landmarks* artificiales se utiliza una infraestructura activa que requiere constante mantención.

En este trabajo se utilizan indicadores que aportan pasivamente con su volumen por lo que no requieren mantención constante, además de no requerir la correlación con un mapa previamente adquirido del túnel.

Una desventaja comparada con otros métodos de localización es que el sistema depende de que el modelo de RANSAC describa correctamente la curva de la pared, por lo que es conveniente tener un algoritmo que utilice el método de forma situacional para la fusión de sensores.

Además es importante considerar que es necesario siempre estar en un rango de visión de al menos dos *landmarks*, lo cual puede ser una limitante en algunos casos. Este algoritmo de localización asume que no hay obstrucciones ni secuestro del vehículo en su funcionamiento.

Debido a que la optimización de la forma del *landmark* se realiza con algoritmos genéticos, modificando repetidamente la población para acercarse a un óptimo no necesariamente global. A pesar de que las soluciones dentro del algoritmo se estancan en un área de optimización (no pueden seguir mejorando) es posible que existan otros puntos de operación o vecindades mejores que la población final.

Otro aspecto a considerar es que actualmente no se considera una corrección del método de asignación, por lo que una asignación incorrecta genera un error acumulado que reduce el rendimiento del sistema.

## 9.2. Resultados y comentarios finales

Dos algoritmos de estimación de posición que utilizan *landmarks* pasivos volumétricos fueron desarrollados para resolver efectivamente el problema de la localización relativa en túneles rectos y homogéneos. Las soluciones actuales desarrolladas para la localización no son aplicables fácilmente a la situación descrita sin requerir condiciones especiales, de alta complejidad de implementación o consisten en métodos reactivos que localizan al vehículo sólo al llegar a puntos notables. Los algoritmos desarrollados en este trabajo como solución a la localización consisten en utilizar mediciones de un sensor LIDAR frontal para detectar las paredes del túnel junto con el volumen de los *landmarks* utilizados para estimar la traslación entre mediciones del sensor. Ambos algoritmos, basados en ICP con distancia euclidiana y transformaciones rígidas iterativas con distancia dirigida y modificada de Hausdorff respectivamente, difieren en el algoritmo de *matching* utilizado para asignar las mediciones detectadas a los *landmarks* correctamente, sin embargo los resultados de las simulaciones indican que ambos métodos logran errores de magnitudes similares en los casos exitosos sin establecer si uno es superior al otro.

Para crear el *landmark* utilizado se estableció un método de optimización novedoso basado en algoritmos genéticos, el cual buscó establecer los parámetros de diseño determinantes a la hora de diseñar y construir el *landmark*. Para esto se consideraron dos genomas, uno basado en formas predefinidas simples llamado primitivo y uno basado en formas poligonales llamado no determinado. En ambos casos las simulaciones pudieron encontrar espacios de solución acotados por lo que en cada población los genes tendieron a ser homogéneos, sin embargo comparando entre las distintas simulaciones y sus condiciones podemos concluir que la forma no determina el desempeño del algoritmo, pero la altura, ancho y distancia entre la posición de los *landmarks* genera un efecto constante y determinante en el resultado.

La experimentación con la plataforma industrial CAT262C muestra que el algoritmo

de estimación de posición relativa es aplicable en un escenario real ya que para la experimentación fueron replicadas las limitaciones sensoriales de un túnel homogéneo con los *landmarks* instalados logrando un desempeño acotado y aceptable según el diseño del túnel utilizado, al ser comparada la estimación con un sensor GPS-RTK de alta precisión.

El estimador logrado tiene como objetivo ser utilizado sólo en las condiciones descritas y puede ser fusionado con otros métodos de medición y estimación.

### 9.3. Temas de investigación futura

Varias consideraciones se proponen como aristas de investigación futura, siendo una de éstas la fusión de las estimaciones con otros sensores o métodos a través de métodos como el filtro de Kalman extendido.

Se consideró para este trabajo utilizar sólo un sensor LIDAR frontal por simplicidad, sin embargo la detección de *landmarks* puede mejorar al utilizar sensores que cubren áreas con traslape u otros sectores del campo de visión del vehículo. También se utilizó un indicador de error global de localización, sin embargo otros indicadores como seguridad, economía o errores críticos pueden ser de utilidad en algunos casos.

Sólo se utilizaron *landmarks* con un diseño único a lo largo del túnel, sin embargo se pueden utilizar diseños distintos con el objetivo de evitar la corrección de la medición o indicar sectores globales al vehículo. Esto se puede lograr comparando los parámetros de desempeño de los métodos de *matching* junto con distintos modelos de *landmarks* instalados.

Para una validación mayor de la experimentación se puede considerar la implementación de este trabajo en una ambiente real de producción subterránea y analizar su impacto en la localización con los métodos actualmente utilizados, descartando posibles problemas no detectados al recrear el ambiente de prueba como interacción con partículas de polvo con el LIDAR, posibles diferencias en el diseño de los *landmarks* y

heterogeneidades propias del túnel recto como también su comportamiento en túneles de gran largo.

**BIBLIOGRAFIA**

Area Minera. (2015). *La minería subterránea es una tendencia mundial*. Descargado 2015-12-11, de <http://www.aminera.com/index.php/mineria-nacional/item/10736-la-miner%C3%ADa-subterr%C3%A1nea-es-una-tendencia-mundial.html>

Bakambu, J., Polotski, V., y Cohen, P. (2004). Autonomous system for exploration and navigation in drift networks. En *Ieee intelligent vehicles symposium, 2004* (pp. 212–217). IEEE. Descargado de <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1336383> doi: 10.1109/IVS.2004.1336383

Bakambu, J. N., y Polotski, V. (2007, oct). Autonomous system for navigation and surveying in underground mines. *Journal of Field Robotics*, 24(10), 829–847. Descargado de <http://doi.wiley.com/10.1002/rob.20213> doi: 10.1002/rob.20213

Besl, P. J., y McKay, H. D. (1992). *A method for registration of 3-D shapes* (Vol. 14) (n.º 2). Descargado de <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=121791> doi: 10.1109/34.121791

Consejo Minero. (2015). Reporte Anual 2015. , 57.

Dahiya, J., Saini, K., y Gopal, G. (2015). Hybridization In Genetic Algorithm. , 33–36.

Derpanis, K. G. (2010). Overview of the RANSAC Algorithm. *Image Rochester NY*, 4(1), 2–3. doi: 10.1002/cne.901000107

Díaz Copier, G., y Carranza Meza, E. (1998). Robotization at El Teniente Mine. *Latin American Perspectives: Exploration, Mining and Processing*, 113 – 122.

Doherty, P. (2014). *Howard Mining System's mini-loader has big impact*. Descargado 2015-12-11, de <http://www.theherald.com.au/story/2589671/mini-loader-has-big-impact/>

Donoso-Aguirre, E, Bustos-Salas, J.-P., Torres-Torriti, M., y Guesalaga, a. (2007). Mobile robot localization using the Hausdorff distance. *Robotica*, 26(02), 129–141. Descargado de [http://www.journals.cambridge.org/abstract{\\_}S0263574707003657](http://www.journals.cambridge.org/abstract{_}S0263574707003657) doi: 10.1017/S0263574707003657

Fischler, M. a., y Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6), 381–395. doi: 10.1145/358669.358692

Goldberg, D. (1994). Genetic and evolutionary algorithms come of age. *Communications of the ACM*, 37(3), 113–119. Descargado de <http://dl.acm.org/citation.cfm?id=175259>

Gustafson, A. (2011). *Dependability Assurance for Automatic Load Haul Dump Machines*.

Haehnel, D. (2004). Mapping with Mobile Robots. (December), 1–169.

Hurteau, R., St-Amant, M., Laperriere, Y., y Chevette, G. (1992). Optical guidance system for underground mine vehicles. En *Proceedings 1992 IEEE international conference on robotics and automation* (pp. 639–644). IEEE Comput. Soc. Press. Descargado de <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=220220> doi: 10.1109/ROBOT.1992.220220

Huttenlocher, D., Klanderman, G., y Rucklidge, W. (1993). Comparing images using the Hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9), 850–863. Descargado de <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=232073> doi: 10.1109/34.232073

Kjer, H. M., y Wilm, J. (2010). Evaluation of surface registration algorithms for PET motion correction Bachelor thesis. *Orbitdtudk*. Descargado de <http://orbit.dtu.dk/getResource?recordId=262855&objectId=1&versionId=1>

Kjer, H. M., y Wilm, J. (2013). *Iterative Closest Point - FileExchange - MATLAB Central*. Descargado de <http://www.mathworks.com/matlabcentral/fileexchange/27804-iterative-closest-point>

Knauer, C., Löffler, M., Scherfenberg, M., y Wolle, T. (2011). The directed Hausdorff distance between imprecise point sets. *Theoretical Computer Science*, 412(32), 4173–4186. doi: 10.1016/j.tcs.2011.01.039

Larsson, J. (2011). *Unmanned Operation of Load-Haul-Dump Vehicles in Mining Environments*. Descargado de <http://www.diva-portal.org/smash/record.jsf?pid=diva2:512493>

Larsson, J., y Broxvall, M. (2008). Laser based intersection detection for reactive navigation in an underground mine. *Intelligent Robots and Systems (IROS), 2008*, 2222–2227. Descargado de [http://ieeexplore.ieee.org/xpls/abs/\\_all.jsp?arnumber=4650911](http://ieeexplore.ieee.org/xpls/abs/_all.jsp?arnumber=4650911) doi: 10.1109/IROS.2008.4650911

Larsson, J., Broxvall, M., y Saffiotti, A. (2005). A Navigation System for Automated Loaders in Underground Mines. *International Conference on Field and Service Robotics*(July 2005).

Larsson, J., Broxvall, M., y Saffiotti, A. (2010). An evaluation of local autonomy applied to teleoperated vehicles in underground mines. *Proceedings - IEEE International Conference on Robotics and Automation*, 1745–1752. doi: 10.1109/ROBOT.2010.5509276

Lu, F. L. F., y Milios, E. (1994). Robot pose estimation in unknown environments by matching 2D range\ncans. *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94., 1994 IEEE Computer Society Conference on*, 3–6. doi: 10.1109/CVPR.1994.323928

Luo, R., Lin, S.-C. L. S.-C., y Lai, C. C. L. C. C. (2008). Indoor autonomous mobile robot localization using natural landmark. *2008 34th Annual Conference of IEEE Industrial Electronics*. doi: 10.1109/IECON.2008.4758197

Man, K. F., Tang, K. S., y Kwong, S. (1996). Genetic algorithms: Concepts and applications. *IEEE Transactions on Industrial Electronics*, 43(5), 519–534. doi: 10.1109/41.538609

Marshall, J., Barfoot, T., y Larsson, J. (2008, jun). Autonomous underground tramming for center-articulated vehicles. *Journal of Field Robotics*, 25(6-7), 400–421. Descargado de <http://doi.wiley.com/10.1002/rob.20242> doi: 10.1002/rob.20242

National Institute for Occupational Safety and Health, y Centers for Disease Control and Prevention. (2012). *CDC - Mining Topic - Respirable Dust*. Descargado 2005-12-18, de <http://www.cdc.gov/niosh/mining/topics/RespirableDust.html>

National Institute for Occupational Safety and Health, y Centers for Disease Control and Prevention. (2014). *CDC - Mining Topic - Rock Falls*. Descargado 2005-12-18, de <http://www.cdc.gov/niosh/mining/topics/rockfalls.html>

National Institute for Occupational Safety and Health, y Centers for Disease Control and Prevention. (2015). *CDC - Mining Topic - Blasting and Explosives*. Descargado 2005-12-18, de <http://www.cdc.gov/niosh/mining/topics/Explosives.html>

Nguyen, V., Martinelli, A., Tomatis, N., y Siegwart, R. (2005). A comparison of line extraction algorithms using 2D laser rangefinder for indoor mobile robotics. *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, 1768–1773. doi: 10.1109/IROS.2005.1545234

Oonk, S., Maldonado, F. J., Li, Z., Reichard, K., y Pentzer, J. (2014). Extended Kalman Filter for Improved Navigation with Fault Awareness. *{IEEE} International Conference on Systems, Man, and Cybernetics*, 2681–2686.

Ouellette, R., y Hirasawa, K. (2007). A comparison of SLAM implementations for indoor mobile robots. *IEEE International Conference on Intelligent Robots and Systems*, 1479–1484. doi: 10.1109/IROS.2007.4399575

Peters, B., Smith, J., Medeiros, D., y Rohrer, M. (2001). a Preliminary Study of Trammig Speeds in Multiple Tele-Operated Load-Haul-Dump Scenarios Using Quest. En *Proceedings of the 2001 winter simulation conference* (pp. 1084–1089). Washington, DC, USA: IEEE Computer Society. Descargado de <http://ukpmc.ac.uk/abstract/CIT/700989>

Roberts, J. M., Duff, E. S., y Corke, P. I. (2002). Reactive navigation and opportunistic localization for autonomous underground mining vehicles. *Information Sciences*, 145, 127–146. doi: 10.1016/S0020-0255(02)00227-X

Rusinkiewicz, S., y Levoy, M. (2001). Efficient variants of the ICP algorithm. *Proceedings Third International Conference on 3-D Digital Imaging and Modeling*, 145–152. Descargado de <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=924423> doi: 10.1109/IM.2001.924423

Scheduling, S., Dissanayake, G., Nebot, E. M., y Durrant-Whyte, H. (1999). An experiment in autonomous navigation of an underground mining vehicle. *IEEE Transactions on Robotics and Automation*, 15(1), 85–95. doi: 10.1109/70.744605

Segal, a., Haehnel, D., y Thrun, S. (2009). Generalized-ICP. *Robotics: Science and Systems*. doi: 10.1.1.149.3870

Servicio Nacional de Geología y Minería. (2016). *Accidentabilidad Minera Primer Trimestre* (Inf. Téc.). Sernageomin. Descargado de <http://www.sernageomin.cl/pdf/mineria/estadisticas/accidentabilidad{ }Minera/Estadistica-de-accidentabilidad-2016.pdf>

SICK. (2015). *LMS511-20100 PRO*. Descargado de <https://www.sick.com/media/pdf/0/40/240/dataSheet{ }LMS511-20100{ }PRO{ }1047782{ }en.pdf>

Sick AG. (s.f.). *2D laser scanners LMS5xx / LMS511 / Outdoor*. Descargado 2016-09-14, de <https://www.sick.com/de/en/detection-and-ranging-solutions/2d-laser-scanners/lms5xx/lms511-10100-pro/p/p215941>

Swift Navigation. (2015). *Understanding Piksi RTK GPS technology*. Descargado 2016-09-14, de <http://docs.swiftnav.com/wiki/Understanding{ }Piksi{ }RTK{ }GPS{ }technology>

Tapia, R. (2012). A Robust Lane Geometry Estimation and Tracking Approach for Driver Alert Using Color and Texture Segmentation. (January).

Torres-Torriti, M., y Guesalaga, A. (2008, may). Scan-to-map matching using the Hausdorff distance for robust mobile robot localization. En *2008*

*ieee international conference on robotics and automation* (Vol. 26, pp. 455–460). IEEE. Descargado de <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4543249> doi: 10.1109/ROBOT.2008.4543249

## ANEXOS

## ANEXO A. RECURSOS ADICIONALES

### A.1. Notación de Variables

$P$ : Nube de puntos detectada por LIDAR.

$p_i$ : Punto en particular de la nube.

$X$ : Coordenadas X de la nube de puntos P.

$x_i$ : Coordenada X de un punto en particular de la nube P.

$Y$ : Coordenadas Y de la nube de puntos P.

$y_i$ : Coordenada Y de un punto en particular de la nube P.

$N$ : Cantidad de puntos en una nube.

$P_p$ : Nube de puntos ajustados a la pared por RANSAC.

$P_e$ : Nube de puntos fuera de la pared por RANSAC.

$PL$ : Nube de puntos correspondiente a un *landmark* medido.

$PM$ : Nube de puntos correspondiente al *landmark* modelo.

$PL1, PL2, PL3$ : *Landmarks* identificados según cercanía al vehículo.

$PV1, PV2$ : *Landmarks* identificados de la iteración anterior.

$PN1, PN2$ : *Landmarks* identificados según cercanía al vehículo y corregidos.

$\bar{p}$ : Posición real del vehículo en el sistema de referencia del túnel.

$\hat{p}$ : Posición estimada del vehículo como resultado del sistema estimador.

$\overline{\bar{p}}$ : Posición real medida del vehículo con GPS.

$A, B, C, D$ : Parámetros del polinomio solución de RANSAC.

$j_i$ : Índice seleccionado con una distribución normal.

$d_{th}$ : Distancia límite para considerar puntos de RANSAC como dentro del modelo.

$QI$ : Cantidad de iteraciones de RANSAC.

$L_{th}$ : Distancia límite entre *landmarks* para asignación.

$g_{assign}$ : Factor de escalamiento para asignaciones fallidas.

$AS$ : Índices identificación para asignación de puntos.

$x_{step}, y_{step}$ : Coordenadas de modificación cartesiana para *matching* de Hausdorff.

$\theta_{step}$ : Resolución de modificación angular para *matching* de Hausdorff.

$\theta_{range}$ : Rango de modificación angular para *matching* de Hausdorff.

$V$ : Mapa de distancias mínimas del *landmark* modelo.

$r_v$ : Resolución del mapa de distancias mínimas

$r_m$ : Fracción de modificación para la distancia de Hausdorff.

$h$ : Distancia dirigida modificada de Hausdorff.

$h_p$ : Punto más alejado de la nube por Hausdorff.

$t_d$ : División por nivel de  $x_{step}$  e  $y_{step}$ .

$\theta_d$ : División por nivel de  $\theta_{step}$ .

$l_m$ : Cantidad de niveles de búsqueda en *matching* de Hausdorff.

$f_{th}$ : Nivel de mejora mínimo considerado como útil

$\theta$ : Rotación óptima del *matching* de Hausdorff.

$T$ : Traslación óptima del *matching* de Hausdorff.

$L_L$ : Distancia entre *landmarks*.

$k_{th}$ : Factor límite para detectar pérdida de vista de *landmarks*.

$DR$ : Desplazamiento estimado por las señales de control en la iteración actual.

$d_{cor}$ : Desplazamiento por *landmarks* corregido.

$V_x, V_y$ : Velocidades en el sistema de coordenadas del vehículo.

$V_r, V_l$ : Velocidad en cada rueda del vehículo.

$\gamma$ : Ángulo de orientación del vehículo en coordenadas locales.

$b$ : Distancia entre ruedas.

$E_{pp}$ : Error de posición promedio.

$E_p$ : Error de posición absoluto.

## A.2. Documentación Simulador

A continuación se resumen las funcionalidades de los métodos implementados en MATLAB para simular los escenarios de prueba, la optimización genética y el procesamiento de datos. Información detallada sobre su uso se puede encontrar dentro del código fuente.

**AnalyzeFeature:** Método auxiliar que permite analizar si un set de puntos auxiliar pertenece a la pared del túnel o a un *landmark*. Usado sólo en simulación para depurar errores.

**CubicModelRANSAC:** Dado un set o nube de puntos busca un modelo de tercer orden a través de RANSAC que se ajuste lo mejor posible al set, entregando dos grupos de puntos, los que se consideran como calces del modelo y los que se consideran como error.

**EncoderCounter:** A partir del movimiento del vehículo calcula la medición del encoder de cada rueda. Usado sólo en simulación.

**FeatureAssignment:** Compara la posición del *landmark* del paso anterior con la posición del mismo *landmark* en la iteración actual para determinar si se ha perdido de vista y es necesario corregir la medición.

**FeatureBuilder:** Construye el *landmark* de forma no determinada a ser adosado a la pared con la información proveniente de un genoma particular.

**FeatureClustering:** Agrupa los puntos detectados como *features* en *landmarks*.

**FitnessFeatureCheck:** Función que genera las mediciones y estimación para una simulación en particular iterando hasta llegar al final del túnel, recibe un genoma y un set de parámetros que definen la simulación para entregar una estructura con los resultados indexados según el paso de la iteración. Normalmente utilizado dentro del algoritmo genético para cada genoma de cada generación.

**GenerateVoronoiMap:** Genera el mapa de distancias mínimas basado en un mapa de Voronoi para un *landmark* particular con el objetivo de acelerar el cálculo de la distancia de Hausdorff.

**GeneticStateSaver:** Función de *callback* para almacenar la información de la generación de la optimización genética.

**HausdorffCalculation:** Usando el mapa de distancias mínimas y el set de puntos, calcula la distancia parcial modificada de Hausdorff.

**HausdorffMatching:** Utilizando el cálculo de Hausdorff busca de forma iterativa una transformación que reduce la distancia entre el set de puntos y el mapa de distancias mínimas.

**icp\_1:** Función de ICP creada por Kjer y Wilm utilizada para realizar el *matching* entre lo sensado y el modelo.

**InterX:** Función externa que calcula la intersección de dos curvas compuestas por puntos discretos.

**LaserRanging:** Dadas las condiciones determinadas por el simulador, genera mediciones del LIDAR para el paso de la iteración correspondiente.

**LMStoXY:** Transforma y filtra las mediciones polares del LIDAR en mediciones cartesianas relativas a la orientación y posición del vehículo.

**MainTunnel:** Genera los datos considerados como reales y constantes para el simulador tales como las paredes del túnel, posición del vehículo en el tiempo y señales de control.

**MoveVehicle2:** Genera información sobre el movimiento y la posición dado un esquema de control para un paso de tiempo determinado.

**ObjectiveFitnessFunction:** Función que es ejecutada en cada generación del algoritmo genético al utilizar genomas de forma no determinada y configura los parámetros de funcionamiento de la simulación. Recibe el arreglo de genomas de la generación de individuos y entrega un arreglo de parámetros de desempeño para cada uno, ejecutando las funciones de cálculo de mediciones, estimaciones y desempeños.

- ParalellObjectiveFitnessFunction:** Similar a ObjectiveFitnessFunction pero permite utilizar múltiples núcleos del procesador para calcular paralelamente la simulación de cada individuo, disminuyendo los tiempos de cómputo total.
- ParalellPrimitiveObjectiveFitnessFunction:** Similar a ParalellObjectiveFitnessFunction pero utiliza genomas de forma primitiva.
- PerformanceCheck2:** Toma los resultados de cada simulación generados por FitnessFeatureCheck y calcula parámetros de desempeño.
- PerformanceCheckReal:** Similar a PerformanceCheck2 pero acondicionada para leer las variables de los experimentos reales en vez de las simuladas.
- PrimitiveBuilder:** Construye el *landmark* de forma primitiva a ser adosado a la pared con la información proveniente de un genoma particular.
- PrimitiveObjectiveFitnessFunction:** Similar a ObjectiveFitnessFunction pero utiliza genomas de forma primitiva.
- RawToLMS:** Filtra y configura mediciones del LIDAR reales para ser utilizadas por la función LMStoXY.
- RealDataFitnessFeatureCheck:** Similar a FitnessFeatureCheck pero acondicionada para procesar las mediciones reales en vez de generarlas mediante simulaciones y ejecutar las estimaciones correspondientes.
- RealParalellObjectiveFitnessFunction:** Similar a ParalellObjectiveFitnessFunction pero utiliza datos reales adquiridos en los experimentos en vez de generar datos simulados.
- VelocityControl2:** Dado un camino objetivo y una posición del vehículo simulado, genera las señales de control necesarias para simular el movimiento del vehículo y mantener la trayectoria dentro del túnel.

### A.3. Documentación Experimentación

En esta sección se describen los *scripts* utilizados durante los experimentos para capturar los datos de los sensores y controlar el vehículo de prueba. Éstos programas

fueron implementados utilizando Python y tienen como objetivo grabar la información pertinente de manera sincronizada para su posterior procesamiento en el simulador antes descrito.

Este sistema fue diseñado para ser multipropósito y recibe información adicional que no necesariamente es utilizada en este trabajo.

**DuplexSerialCom:** Crea un objeto auxiliar capaz de recibir y enviar mensajes de a través de los puertos seriales o COM, éstos puertos son utilizados para interactuar con el sistema de control del vehículo, los sensores de movimiento inercial y el sistema GPS.

**ClientSideSocket:** Crea un objeto auxiliar capaz de recibir y enviar información a través de un Socket TCP definido a la aplicación InfoHub\_GUI.

**IMU\_InfoHub:** Programa que captura datos de la unidad de medición inercial (IMU) instalada en el vehículo, para ser enviados al programa InfoHub\_GUI. Debe ser ejecutado en paralelo con el resto de las aplicaciones.

**GPS\_InfoHub:** Programa que captura datos del GPS RTK instalado en el vehículo y en el escenario de prueba, para ser enviados al programa InfoHub\_GUI. Debe ser ejecutado en paralelo con el resto de las aplicaciones.

**MBD\_InfoHub:** Programa que captura datos de los encoders a través de la placa de control instalada en el vehículo y maneja las señales de control enviadas al vehículo de prueba. Posee una interfaz que permite controlar el vehículo en tiempo real y ejecutar ordenes de forma predefinida. Debe ser ejecutado en paralelo con el resto de las aplicaciones.

**LMS\_InfoHub:** Programa que captura datos del LIDAR instalado en el vehículo, para ser enviados al programa InfoHub\_GUI. Debe ser ejecutado en paralelo con el resto de las aplicaciones.

**InfoHub\_GUI:** Programa central para la adquisición de datos, recibe los datos de los programas antes mencionados y los guarda en la memoria en un formato común junto con su registro de tiempo. También posee una interfaz que

permite visualizar las mediciones del LIDAR junto con el registro del camino recorrido según las órdenes de control enviadas al vehículo.

#### A.4. PCB de control

Como parte del trabajo de adaptación del CAT262C para la realización de experimentos, se realizaron varias modificaciones al hardware del vehículo, incluyendo la modificación de señales y sensores y la adición de módulos con distintos objetivos. Uno de los módulos más importantes es el *Masterboard* que sirve como punto central para la realización de diversos experimentos y fue diseñada por el equipo RAL con una tarjeta de control Arduino Due como microcontrolador central.

Entre sus características se encuentran:

- 4 conexiones seriales RS232.
- Conexión serial-USB incluido en la placa de desarrollo.
- Bus I2C de 3.3V o 5V.
- 2 sets de 6 pines GPIO multivoltaje, capaces de funcionar a un voltaje arbitrario por set mayor al del microcontrolador utilizado.
- 1 set de 6 pines GPIO multivoltaje, capaces de funcionar a un voltaje arbitrario por set menor al del microcontrolador utilizado.
- Puertos de lectura de encoders.
- Sistema de intervención de señales para la llave de ignición y el botón de *parking*, permitiendo controlar externamente el encendido, apagado del vehículo junto con la activación y desactivación del freno de emergencia.
- Conversor de señales PWM para controlar directamente al vehículo.
- Estructura con microcontrolador removible para permitir su cambio si es requerido.
- Selector optoaislado de relays externos para seleccionar el modo de operación entre normal, teleoperación, y automatizado.

En la imagen A.1 se muestra la vista total de la PCB diseñada en CAD.

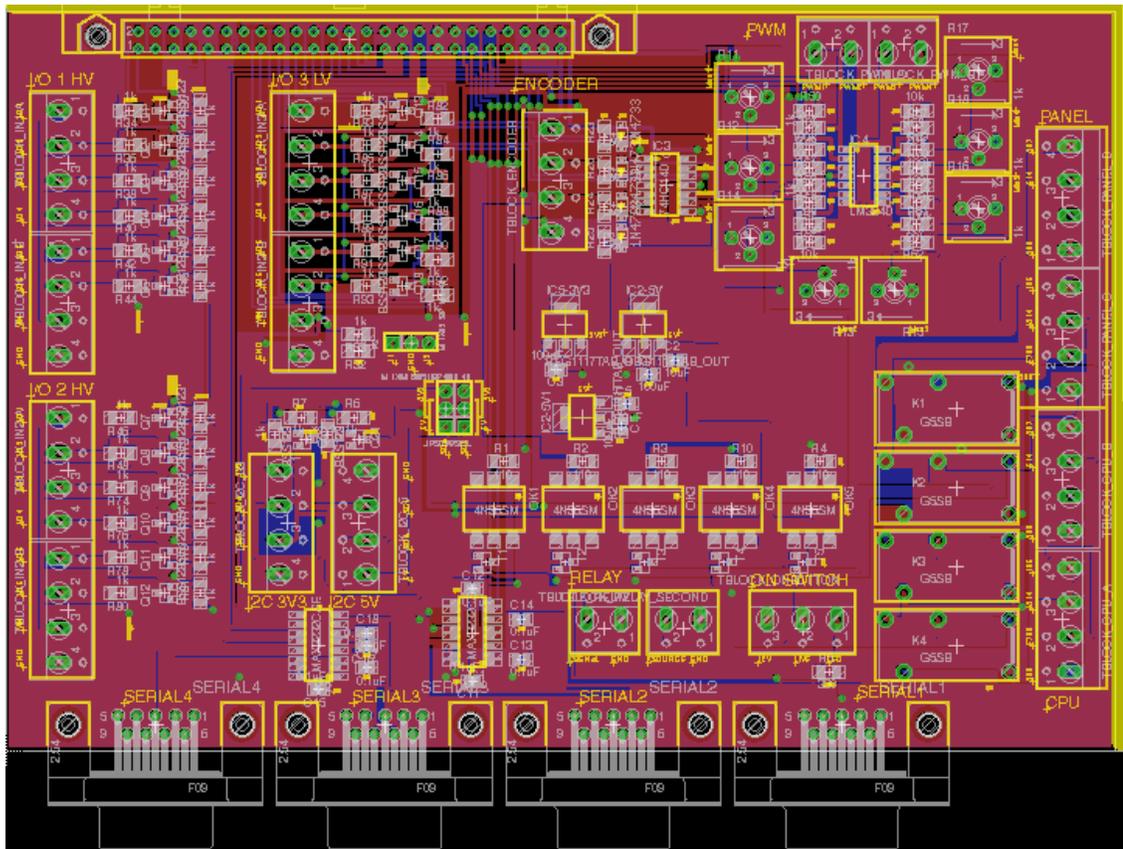


FIGURA A.1. Vista total del diseño de PCB Masterboard.