



Instituto Politécnico Nacional

---

---

Centro de Investigación en Ciencia Aplicada y  
Tecnología Avanzada

Unidad Querétaro

Posgrado en Tecnología Avanzada

**Elaboración de un sistema mecatrónico para  
incrementar el ángulo de visión vertical del LIDAR  
HDL-64E**

**Tesis**

Que para obtener el grado de maestro en tecnología avanzada

**Presenta**

Ing. Róger Ernesto Sánchez Alonso

**Directores de tesis**

Dr. Eduardo Castillo Castañeda

Dr. José Joel González Barbosa



Querétaro, Qro. Mayo de 2011



INSTITUTO POLITÉCNICO NACIONAL  
SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

ACTA DE REVISIÓN DE TESIS

En la Ciudad de Querétaro, Qro siendo las 12:00 horas del día 28 del mes de Abril del 2011 se reunieron los miembros de la Comisión Revisora de Tesis, designada por el Colegio de Profesores de Estudios de Posgrado e Investigación de CICATA-QRO para examinar la tesis titulada:

**Elaboración de un sistema mecatrónico para incrementar el ángulo de visión vertical del LIDAR HDL-64E**

Presentada por el alumno:

Sánchez  
Apellido paterno

Alonso  
Apellido materno

Róger Ernesto  
Nombre(s)

Con registro: 

A	0	9	0	5	7	6
---	---	---	---	---	---	---

aspirante de:

Maestría en Tecnología Avanzada

Después de intercambiar opiniones los miembros de la Comisión manifestaron **APROBAR LA TESIS**, en virtud de que satisface los requisitos señalados por las disposiciones reglamentarias vigentes.

LA COMISIÓN REVISORA

Directores de tesis

Dr. Eduardo Castillo Castañeda

Directores de tesis

Dr. José Joel González Barbosa

M en C. Maximiano Francisco Ruiz Torres

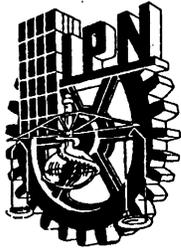
Dr. Francisco Javier Ornelas Rodríguez

Dr. Eduardo Morales Sánchez

PRESIDENTE DEL COLEGIO DE

Dr. Jorge Adalberto Huerta





**INSTITUTO POLITÉCNICO NACIONAL**  
**SECRETARÍA DE INVESTIGACIÓN Y POSGRADO**

**CARTA CESIÓN DE DERECHOS**

En la Ciudad de Santiago de Querétaro el día 28 del mes de Abril del año 2011, el que suscribe Róger Ernesto Sánchez Alonso alumno del Programa de Maestría en Tecnología Avanzada con número de registro A090576, adscrito al Centro de Investigación en Ciencia Aplicada y Tecnología Avanzada - Unidad Querétaro, manifiesta que es autor intelectual del presente trabajo de Tesis bajo la dirección del Dr. Eduardo Castillo Castañeda y cede los derechos del trabajo intitulado **“Elaboración de un sistema mecatrónico para incrementar el ángulo de visión vertical del LIDAR HDL-64E”** , al Instituto Politécnico Nacional para su difusión, con fines académicos y de investigación.

Los usuarios de la información no deben reproducir el contenido textual, gráficas o datos del trabajo sin el permiso expreso del autor y/o director del trabajo. Este puede ser obtenido escribiendo a la siguiente dirección: Cerro Blanco No. 141. Col. Colinas del Cimatario, Santiago de Querétaro, Querétaro. C.P. 76090. Si el permiso se otorga, el usuario deberá dar el agradecimiento correspondiente y citar la fuente del mismo.



Nombre y firma



## **Elaboración de un sistema mecatrónico para incrementar el ángulo de visión vertical del LIDAR HDL-64E**

Róger Ernesto Sánchez Alonso

Centro de Investigación en Ciencia Aplicada y Tecnología Avanzada

Instituto Politécnico Nacional  
Querétaro, México.

---

## Agradecimientos

---

De la forma más atenta quisiera brindar mi más profundo y sincero agradecimiento...

A Dios, pues es por él que estoy aquí. Él me regaló la vida y me ha dado la fortaleza para salir adelante y cumplir cada uno de mis sueños y metas.

A la Universidad Nacional de Ingeniería (UNI), por brindarme la oportunidad de continuar formándome como profesional y como hombre de bien.

Al Centro de Investigación en Ciencia Aplicada y Tecnología Avanzada (CICATA-Qro), por abrirme sus puertas y casi inmediatamente hacerme miembro de la gran familia Politécnica.

A los doctores Joaquín Salas, Eduardo Castillo y Joel González, por confiar en mí y darme la oportunidad de demostrarles a todos que con esfuerzo, dedicación, voluntad y un poco de talento todo se puede alcanzar.

A todos los amigos que encontré en este camino, por ayudarme cuando lo necesité y por brindarme su cariño incondicional.

Al Consejo Nacional de Ciencia y Tecnología (CONAYT), por el apoyo económico suministrado durante el desarrollo de este proyecto.

**A todos infinitas gracias**

---

## Resumen

---

En este trabajo se presenta el desarrollo de un prototipo mecatrónico que permite incrementar de 26.8 a 69.8° el ángulo de visión vertical del LIDAR HDL-64E, el cual es un sensor de alta definición utilizado para la extracción de modelos 3D de cualquier entorno físico.

El sensor LIDAR HDL-64E es un dispositivo elaborado con fines de navegación autónoma de vehículos, sin embargo, a través de la ampliación en su campo de visión provista por el prototipo diseñado y manufacturado en este trabajo de investigación sus horizontes de aplicación se expanden significativamente.

La ampliación del campo de visión del sensor se da al inclinarlo a través de un dispositivo de un grado de libertad, dispositivo que basa su funcionamiento en un mecanismo de transmisión de movimiento conocido como “triángulo con un lado de longitud variable”. El preciso posicionamiento angular del sensor se obtiene gracias a la implementación de un controlador PID, el cual fue complementado con el desarrollo de un interfaz de usuario muy amigable y ajustada a la aplicación.

El hecho de inclinar el HDL-64E para ampliar su campo de visión genera un cambio en el sistemas de referencia de las capturas que se hagan, por esta razón se desarrolló una metodología para corregir las nubes de puntos adquiridas, esta metodología consiste en expresar dichas nubes de puntos en un sistema de referencia global, sistema que corresponde a la referencia horizontal del sensor (0° de inclinación).

Con el sistema mecatrónico de posicionamiento angular y la metodología de corrección de nubes de puntos se pueden realizar capturas con el HDL-64E y obtener nubes de puntos con mucha más información referente a la altura de algún entorno físico. El sistema está listo para ser montado sobre un vehículo y realizar tareas vinculadas a la reconstrucción 3D de ambientes urbanos.

---

## Abstract

---

This work presents the development of a mechatronic prototype that allows to increase from 26.8 to 69.8 ° the vertical viewing angle of the LIDAR HDL-64E, which is a high definition sensor used for the extraction of 3D models of any physical environment.

The LIDAR HDL-64E sensor is a device designed for navigation of autonomous vehicles, however, with the expansion in the field of vision provided by the prototype designed and manufactured in this study, it significantly expands its applicability.

The expansion in the field of vision of sensor is given when it is tilted through a one degree of freedom device, device that bases its operation on a motion transmission mechanism called "variable base triangle". The precise angular positioning of the sensor is achieved through the implementation of a PID controller, which was complemented by the development of a very friendly user interface and tailored to application.

The fact to tilt the HDL-64E to extend its field of view generates a change in the reference systems of all shots made, for this reason a methodology to correct the acquired clouds of points was developed, this methodology is to express these clouds of points in a global reference system, system that corresponds to the horizontal reference of the sensor (0 ° inclination).

With the mechatronic angular positioning system and the methodology for correction of clouds of points, one can perform scans with the sensor, and obtain clouds of points with much more information about the height of a physical environment. The system is ready to be mounted on a vehicle and perform tasks related to the 3D reconstruction of urban environments



---

# Índice de contenido

---

Agradecimientos .....	i
Resumen.....	ii
Abstract.....	iii
Capítulo 1: Introducción .....	1
1.1 Justificación .....	2
1.2 Objetivos.....	2
1.2.1    Objetivo general .....	2
1.2.2    Objetivos específicos .....	3
1.3 Hipótesis .....	3
1.4 Estructura de la tesis .....	3
Capítulo 2: Estado del arte .....	4
Capítulo 3: Marco teórico.....	10
3.1 Sensor LIDAR HDL-64E .....	10
3.1.1    Principio de funcionamiento y especificaciones .....	10
3.1.2    Paquetes de datos .....	12
3.1.3    Factores de corrección a las adquisiciones .....	13
3.2 Mecanismos de tres barras .....	14
3.3 Controladores PID .....	15
3.4 Matrices de transformación homogéneas .....	22
3.4.1    Matrices de rotación.....	23
3.4.2    Vectores de traslación .....	28
3.4.3    Matrices de transformación homogéneas.....	28
Capítulo 4: Descripción de la base experimental .....	30
4.1 Diseño y construcción del prototipo.....	30
4.1.1    Diseño conceptual.....	30
4.1.2    Construcción del prototipo .....	38



---

4.2 Desarrollo del controlador PID del prototipo .....	40
4.2.1 Motor y sensor de posición .....	40
4.2.2 Tarjeta de control .....	40
4.2.3 Resolución del actuador .....	43
4.2.4 Comportamiento cinemático del controlador .....	44
4.2.5 Diseño de la interfaz de usuario .....	47
4.3 Transformación de las nubes de puntos .....	51
4.3.1 Base de cálculo de la metodología .....	52
4.3.2 Selección de los puntos correlacionados .....	54
4.3.3 Cálculo de las matrices de transformación homogéneas .....	59
Capítulo 5: Validación experimental .....	61
Capítulo 6: Conclusiones y trabajo futuro.....	69
Referencias .....	71
Anexos .....	74



---

## Índice de tablas

---

Tabla 3. 1. Descripción de los factores de corrección aplicados a las capturas del DHL-64E. ....	13
Tabla 4. 1. Configuración de pines para el puerto USB del convertidor SSA-485.....	42
Tabla 4. 2. Configuración de pines para la comunicación RS485 entre el convertidor SSA-485 y la tarjeta PIC-SERVO SC. ....	42
Tabla 4. 3. Configuración de pines para el conector DB15 de la tarjeta PIC-SERVO SC. ....	42
Tabla 4. 4. Cálculo de una aproximación al error promedio del sistema de control. ....	46
Tabla 4. 5. Valores de la matriz homogénea que representa la transformación 3D experimentada al inclinar 5° el HDL-64E.....	60

## Índice de figuras

Fig. 3. 1. Ejemplo de nube de puntos adquirida con el HDL-64E en un estacionamiento. ....	11
Fig. 3. 2. Diseño del LIDAR HDL-64E. ....	11
Fig. 3. 3. Ejemplo de la secuencia de captura de datos. ....	12
Fig. 3. 4. Funcionamiento del gato elevador. ....	14
Fig. 3. 5. Algunas aplicaciones de mecanismos de tres barras. ....	15
Fig. 3. 6. Diagrama para la identificación de un motor de corriente directa. ....	17
Fig. 3. 7. Diagrama en lazo cerrado en posición angular de un motor de corriente directa. ....	17
Fig. 3. 8. Respuesta en el tiempo de un sistema de segundo orden. ....	20
Fig. 3. 9. Diagrama esquemático de un controlador PID. ....	21
Fig. 3. 10. Posición de un satélite artificial respecto a dos sistemas de referencia $O$ y $O'$ . ....	23
Fig. 3. 11. Posición de un punto $P$ producto de una rotación pura en el eje $Z$ . ....	24
Fig. 3. 12. Secuencia de rotaciones para la representación de los ángulos Roll-Pitch-Yaw. ....	26
Fig. 3. 13. Secuencia de rotaciones para la representación de los ángulos de Euler. ....	27
Fig. 4. 1. Condición actual de uso del LIDAR-HDL64E. ....	31
Fig. 4. 2. Consideraciones para poder estimar la altura máxima a la que se puede sensor. ....	31
Fig. 4. 3. Bosquejo del aumento del campo de visión vertical del sensor. ....	33
Fig. 4. 4. Composición mecánica básica del prototipo. ....	34
Fig. 4. 5. Transmisión mecánica del prototipo. ....	35
Fig. 4. 6. Mecanismo de transmisión de movimiento. ....	36
Fig. 4. 7. Diagrama cinemático del mecanismo. ....	36
Fig. 4. 8. Prototipo construido. ....	39
Fig. 4. 9. Motor seleccionado como actuador del prototipo. ....	40
Fig. 4. 10. Diagrama eléctrico para la conexión de PIC-SERVO SC. ....	41
Fig. 4. 11. Relación entre la posición del motor y la posición de la plataforma de inclinación. ....	45
Fig. 4. 12. Interfaz de usuario desarrollada en Visual Basic. ....	47
Fig. 4. 13. Selección de parámetros de comunicación. ....	47
Fig. 4. 14. Mensajes de error de la interfaz. ....	48
Fig. 4. 15. Sección "Parámetros de entrada". ....	48
Fig. 4. 16. Máximo posicionamiento angular del HDL-64E. ....	49
Fig. 4. 17. Prueba del sistema de posicionamiento HDL-64E. ....	50



Fig. 4. 18. Vista lateral de escaneo con el HDL-64E posicionado a 30° de inclinación. ....51

Fig. 4. 19. Nubes de puntos aptas para la extracción de sus puntos correlacionados .....55

Fig. 4. 20. Nubes de puntos graficadas en Matlab para la extracción de puntos .....56

Fig. 4. 21. Extracción de primera aproximación para los puntos .....57

Fig. 4. 22. Eliminación de vecindades para la mejor extracción de puntos.....58

Fig. 5. 1. Capturas con el sistema de posicionamiento HDL-64E. ....61

Fig. 5. 2. Transformación de la nube de puntos .....63

Fig. 5. 3. Nube de puntos de un edificio capturada con 5° de inclinación.....64

Fig. 5. 4. Transformación de la nube de puntos adquirida de un edificio a 5° de inclinación .....65

Fig. 5. 5. Corrección de la nube de puntos escaneada a 30° mostrada en la figura 4.18. ....66

Fig. 5. 6. Nube de puntos que representa completamente el escenario escaneado. ....66

Fig. 5. 7. Vista panorámica del escenario escaneado .....67

Fig. 5. 8. Vista superior de la nube de puntos del escenario completo.....67

Fig. 5. 9. Eliminación manual de información ruidosa. ....68

Fig. 5. 10. Vista superior de la nube de puntos del escenario completo después de la supresión  
de ruido.....68

---

## Capítulo 1: Introducción

---

Dentro del gran campo de investigación de la reconstrucción tridimensional existe un área en la que se están concentrando grandes esfuerzos, esta área es la modelización de espacios urbanos. Obtener modelos 3D de ciudades ha sido objeto de creciente interés en los últimos años, esto se debe principalmente a que este tipo de investigaciones poseen horizontes de aplicación muy extensos, por ejemplo en [1] se puntualiza áreas de aplicación vinculadas a la planeación urbana, la arqueología, el turismo virtual, la simulación, la restauración, etc.

Actualmente existe una diversidad de metodologías destinadas a la obtención de modelos 3D de zonas urbanas, muchas de ellas planteadas en [1] y [2], sin embargo, existe una que ha evolucionado ampliamente y que ofrece excelentes resultados, esta metodología está basada en la tecnología de escaneo láser.

El escaneo láser para la generación de modelos 3D de ciudades básicamente se aplica de dos formas; la primera consiste en escaneos aéreos, lo que genera los llamados DEMs (Digital Elevation Models), y la segunda implica escaneos terrestres, lo que genera modelos digitales que a diferencia de los DEMs incluyen detalles a nivel de suelo ideales para la exploración del entorno.

La tecnología de escaneo LIDAR (Light Detection and Ranging o Laser Imaging Detection and Ranging) está disponible tanto de forma aérea como terrestre en diversas marcas y modelos. Cada uno de estos modelos están diseñados para propósitos muy específicos, por lo que incorporan especificaciones muy particulares. Obtener un sensor LIDAR que incorpore especificaciones para múltiples tareas (navegación autónoma, cartografía, reconstrucción 3D de edificios, etc.) es casi imposible, por tal razón los usuarios de esta tecnología tienen que acondicionar sus equipos para ampliar su funcionalidad o bien para desarrollar una aplicación específica.

El presente trabajo está enfocado en el acondicionamiento del LIDAR HDL-64E, el cual es un dispositivo elaborado por la compañía Velodyne con fines orientados a la navegación autónoma [3], para que pueda ser utilizado en tareas de reconstrucción 3D de ambientes urbanos.

El HDL-64E es un sensor que por sus especificaciones de resolución, precisión, alcance y densidad de datos entregados es considerado como de alto desempeño. Sin embargo, éste presenta un ángulo de visión vertical reducido, 2° por encima de la horizontal y 24.8 por debajo, condición que limita su funcionalidad para realizar tareas de reconstrucción de espacios urbanos, pues resulta imposible obtener escaneos completos de estructuras de altura considerable.

Al tomar en cuenta el enorme potencial del sensor para la obtención de modelos 3D es que surge el interés de habilitarlo para el desarrollo de tareas relacionadas al escaneo de espacios urbanos, por lo que en este trabajo se propone el diseño y manufactura de un dispositivo mecatrónico que incremente el campo de visión vertical del sensor.

El prototipo desarrollado en este estudio inclina de forma automática el HDL-64E, fijándolo en la posición angular necesaria para poder, en una serie de barridos, obtener una nube de puntos completa de estructuras que por su altura no podrían ser escaneadas por el sensor en condiciones normales.

## **1.1 Justificación**

Esta investigación permite de una forma práctica y sobre todo económica incrementar el desempeño del LIDAR HDL-64E a través de un aumento considerable en su campo de visión vertical. Ampliar el campo de visión vertical del LIDAR HDL-64E permite acondicionarlo para realizar tareas diferentes a la navegación autónoma de vehículos, entre estas tareas destacan la elaboración de mapas 3D de ciudades que incluyan estructuras de altura considerable, la digitalización del patrimonio histórico y cultural, la extracción de características como el daño o dimensiones de edificios y otras obras civiles verticales, etc. Expandir el ángulo de visión vertical del sensor implica expandir los horizontes de aplicación del usuario sin tener que invertir en otro sensor.

## **1.2 Objetivos**

### **1.2.1 Objetivo general**

Elaborar un sistema mecatrónico que permita incrementar el ángulo de visión vertical del LIDAR HDL-64E para posibilitar la reconstrucción 3D de entornos que contengan estructuras de altura que no podrían ser alcanzadas bajo las condiciones normales de operación del sensor.

### 1.2.2 Objetivos específicos

- Diseñar y construir un sistema mecánico de un grado de libertad que permita aumentar el ángulo de visión vertical del LIDAR HDL-64E de 26.8° a 69.8°.
- Implementar un control PID en posición angular al sistema de mecatrónico de elevación.
- Validar de funcionalidad del sistema para la reconstrucción 3D.

### 1.3 Hipótesis

Es posible incrementar el campo de visión vertical del LIDAR HDL-64E a través de un sistema mecatrónico de posicionamiento angular para habilitarlo en el desarrollo de tareas de reconstrucción 3D de entornos urbanos.

### 1.4 Estructura de la tesis

El presente documento está estructurado por seis capítulos, cuyos contenidos se resumen de forma breve a continuación:

Capítulo 1: en este capítulo se presenta una introducción al trabajo de investigación, en donde se contextualiza el problema a resolver, se define la justificación del trabajo y se plantean los objetivos e hipótesis de investigación.

Capítulo 2: dedicado al planteamiento del Estado del arte.

Capítulo 3: Marco teórico, en este capítulo se explica detalladamente todos los fundamentos teóricos que sustentan la investigación.

Capítulo 4: en este capítulo se presenta el desarrollo de la metodología implementada para el cumplimiento de los objetivos.

Capítulo 5: Validación experimental, contiene los resultados que validan o demuestran la funcionalidad del sistema desarrollado para ampliar el ángulo de visión vertical del LIDAR HDL-64E

Capítulo 6: integrado por las conclusiones finales y propuestas para trabajo futuro.

---

## Capítulo 2: Estado del arte

---

El presente trabajo se enmarca en el contexto de la reconstrucción tridimensional, es decir, el proceso mediante el cual objetos reales son reproducidos en la memoria de una computadora manteniendo sus características físicas esenciales (dimensiones, volumen, forma, etc.).

Dentro de este campo de investigación existe una rama que se ha desarrollado ampliamente en los últimos años, esta rama es la modelización 3D de espacios urbanos. Hoy por hoy todo el mundo tiene claro y son más conscientes de las ventajas, facilidades, horizontes comerciales y la infinidad de oportunidades que están detrás de este tipo de investigaciones, y es por ello que se ha incrementado el interés en la generación de modelos 3D de ciudades y su rápida actualización.

Poder visualizar entornos urbanos, según expertos, ha sido tema del arte occidental desde hace mucho tiempo. Sin embargo, el problema crucial para alcanzar esta meta, como se narra en [4], es resumir en una sola imagen toda la ampliación arquitectónica del espacio deseado e incursionar en zonas remotas sin introducir enormes distorsiones.

El modelado 3D, en general puede ser visto como un proceso, el cual inicia con la adquisición de los datos y termina con un modelo virtual que es posible visualizar interactivamente a través de una computadora. Por ejemplo en [5] se plantea que el modelado 3D se estructura de la siguiente manera:

1. Diseño (del sensor y geometría de la red).
2. Mediciones o captura (obtención de nubes de puntos, líneas, etc.).
3. Estructurado / Modelado (geometría, textura).
4. Visualización / Análisis.

Concentrándose puntualmente en la tecnología, equipo o técnica con la que se extraerán los datos, existe una amplia gama de opciones. Los artículos [1] y [2] constituyen muy buenas referencia de dichos procedimientos para la generación de modelos de ciudades, en ellos se

abordan fuentes de datos como; imágenes satelitales, imágenes aéreas (no satelitales), el escáner laser terrestre y aéreo, entre otras.

A pesar del amplio abanico de posibilidades para la realización de este tipo de estudios existen dos técnicas o metodologías que están evolucionando y ofreciendo buenos resultados en la línea de la automatización de la generación de modelos urbanos. La primera se basa en el uso exclusivo de imágenes ópticas (por ejemplo, fotografías), tanto para la reconstrucción del modelo geométrico o de formas como para el modelo de texturas. La segunda emplea la tecnología LIDAR (Light Detection and Ranging) para la reconstrucción de formas y puede usar tanto la misma tecnología como la fotografía para la reconstrucción de la textura.

## **2.1 Modelado 3D con fotogrametría aérea y terrestre**

La aproximación clásica para la generación de un modelo urbano a partir de fotografías requiere de mucha intervención manual, tanto en la fase de aerotriangulación como en la de modelado o restitución tridimensional. Además es imprescindible combinar los resultados de explotación de imágenes aéreas con imágenes terrestres.

Los algorítmicos descritos en la literatura en los últimos años se ciñen típicamente a la mejora de algún aspecto particular dentro del amplio espectro de problemas que la automatización de la extracción de un modelo tridimensional conlleva: calibración de la cámara, registro, extracción de estructuras, estimación de texturas, iluminación global, entre otros.

Hasta el momento se han desarrollado y descrito diversas soluciones parciales, es decir, soluciones que imponen ciertas restricciones en el uso o en los resultados. Algunos algoritmos asumen configuraciones especiales para las cámaras o las imágenes. Otros, como [6] y [7], han propuesto sistemas que partían de un modelo 3D aproximado y mediante las imágenes que toman con una cámara refinaban dicho modelo. La ventaja reside en que se requiere un número limitado de imágenes.

En [8] presentan un sistema totalmente automático que obtiene un modelo 3D de superficies a partir de imágenes captadas moviendo libremente la cámara alrededor del objeto. El sistema utiliza cámaras de perspectiva y no requiere calibración previa al uso. Muestran algunos resultados de aplicación dentro del ámbito de la arqueología.

En [9] presentan un robot encargado de establecer automáticamente cartografía para áreas urbanas. Las bases de la aproximación metodológica al problema son: adquisición de metadatos

de georreferenciación para cada imagen, adquisición de imágenes omnidireccionales que garantizan una calibración más precisa y robusta de la cámara, el uso de una formulación probabilística del error de proyección y, por último, requisitos computacionales del sistema lineal en relación con el área de adquisición y el tamaño del modelo de salida. Concluye que a medida que crece la magnitud del modelo a establecer, los sistemas automáticos aventajan a los operadores humanos no sólo en cuanto a eficiencia sino también en cuanto a precisión.

## **2.1 Modelado 3D con tecnología LIDAR**

El LIDAR (Light Detection And Ranging) es un instrumento que emite una serie de pulsos láser que son reflejados por los objetos de interés y retornados hacia el instrumento. Los sistemas LIDAR son muy similares a los radares, aunque utilizando longitudes de onda mucho más cortas. El tiempo transcurrido entre emisión-recepción permite calcular la distancia entre objeto y sensor. Simultáneamente se puede registrar la posición y orientación del sensor mediante un sistema integrado GPS/IMU.

Las principales ventajas del LIDAR respecto de la fotografía son dos: determinación directa, precisa y rápida de las distancias (y no indirecta con base a los resultados de una etapa de matching como en la fotografía) y, en segundo lugar, no es necesaria la luz ambiente al tratarse de un sensor activo. Para determinadas aplicaciones, especialmente las que requieran alto grado de camuflaje o seguridad, este último aspecto puede ser el mayor obstáculo ya que un sensor activo, por su naturaleza, delata su posición. También sabemos que un láser no opera perfectamente bajo una intensa radiación solar, es más lento, pesado, de mayor consumo energético y obtiene buenos resultados para un rango de distancias limitado.

Muchos son los defensores acérrimos de esta tecnología para la obtención de modelos digitales de superficie o modelos urbanos 3D, especialmente para zonas de alta densidad urbana [10] [11]. Ahora bien, la elaboración de un modelo urbano requiere la aplicación de costosos preprocesos de filtrado de ruido y suavización de la información previos a la detección y vectorización de las aristas de los edificios, cuya automatización, que esencialmente es un problema de conversión de ráster a vector, es uno de los campos de batalla claves dentro de esta técnica [12].

La fotogrametría y el escaneo láser, a pesar de ser técnicas de naturaleza distinta, pueden trabajar sinérgicamente y proveer grandes resultados. Un ejemplo de la combinación del escaneo láser y la fotogrametría es [13], un trabajo donde se modela tridimensionalmente una bóveda de la catedral de Valencia. Para este proyecto se empleó un escáner láser terrestre de

alta resolución para la adquisición de la nube de puntos y una cámara digital réflex para la toma de imágenes. En primer lugar se procesaron de forma individual las nubes de puntos obtenidas por el láser desde diferentes posiciones. Se aplicaron filtros con restricciones geométricas de acuerdo con las características del monumento para eliminar puntos no deseados. De esta manera se obtuvo un modelo 3D completo sin textura. La cámara se calibró a partir de las imágenes tomadas in situ junto con los datos del escaneado láser mediante un proceso de autocalibración. Finalmente se dotó de textura al modelo 3D mediante dos técnicas diferentes: ZI-Buffer y la proyección de imágenes seleccionadas manualmente.

El escaneo láser terrestre es específicamente ventajoso para el mapeo tridimensional exacto de construcciones con más detalles como las carreteras, puentes, el mobiliario urbano o vegetación. En este contexto la reconstrucción 3D con datos aéreos provee las formas de los techos de los edificios, mientras que la recolección de datos terrestres es útil para el refinamiento geométrico de las fachadas de las construcciones, por ejemplo el trabajo realizado en [14].

En los últimos años se han desarrollado un sin número de trabajos gracias a la tecnología LIDAR. Por ejemplo [15] constituye un proyecto del modelado 3D de una manzana del casco antiguo de la ciudad de Madrid. La forma de trabajar con el equipo láser 3D fue ir barriendo toda la manzana con sucesivos escaneos. Se realizaron 25 tomas a fin de cubrir la manzana entera. En cada toma se emplearon unos 6 minutos, lo cual hizo que, sumado el tiempo de desplazamiento del equipo de un punto a otro y la planificación insitu de las vistas, se permitiera la digitalización de la manzana entera en un único día de trabajo. Cada una de las vistas barre un ángulo vertical de  $90^\circ$  y horizontal de  $180^\circ$ , proporcionándonos una nube de aproximadamente millón y medio de puntos. De esta manera la información que se obtuvo fue muy superior a las de los procedimientos tradicionales. Posteriormente se realizó el procesamiento de los datos obtenidos en campo, iniciando por unir las 25 tomas individuales en una sola toma 3D de la manzana completa (georreferenciar los datos). A continuación se hizo una limpieza del modelo eliminando puntos tomados que no sean propiamente de las fachadas, además de vehículos, viandantes, ruido de las imágenes, etc. Una vez unidas todas las fachadas en un solo archivo y filtrado, se pudo hacer cualquier operación que requiera de posiciones absolutas, como exportar los datos de los puntos obtenidos a un programa de CAD para obtener los alzados de las fachadas, realizar cortes siguiendo cualquier eje, triangular la nube de puntos para obtener una malla y un modelo de superficies, videos de realidad virtual, etc.

Otro trabajo que vale mucho la pena resaltar es [16], el cual consiste en la generación de modelos 3D en la vieja Habana. El objetivo del trabajo citado anteriormente fue obtener un registro digital de alta precisión de inmuebles con valor histórico a través de la aplicación del

escáner láser. El levantamiento con el escáner láser permite la creación de un modelo 3D construido a partir de millones de medidas (nubes de puntos) en tiempos sumamente breves. En este artículo se presenta una aproximación al proceso de levantamiento con escáneres láser para la generación de planos arquitectónicos, a través de orto-imágenes de nubes de puntos con altas densidades. Esta metodología plantea lo innecesario de transformar modelos complejos de nubes de puntos a modelos de mallas texturizados. Como resultado de la investigación se elaboró una metodología que incluye las siguientes etapas: captura de Datos, post proceso (limpieza, registro, optimización del modelo) y explotación 2D y 3D del modelo de nube puntos.

La tecnología LIDAR ha sido ampliamente utilizada alrededor del mundo, incluso para propósitos diferentes a los establecidos en [15] y [16]. Por ejemplo [17] es un trabajo que donde se evaluó las condiciones de la infraestructura ferroviaria que comunica la ciudad de Madrid con Barcelona. Este artículo ilustra, de forma sucinta, las pruebas realizadas por los autores, con el fin de desarrollar una plataforma de monitorización basada en la fusión de datos provenientes de sensores LIDAR, imágenes captadas con cámara de video y datos de posicionamiento GPS. Esta información integrada en un sistema global de monitorización permitiría la obtención de datos métricos y cualitativos del estado de la vía y su entorno hasta una distancia de 100m. Los métodos de procesamiento aplicados sobre estos datos están basados en técnicas de *clustering*, segmentación y reconstrucción utilizadas anteriormente en la inspección “on-line” de líneas de distribución eléctrica, así como en la aplicación de técnicas de localización de sistemas móviles autónomos y fusión sensorial.

Otra de las grandes aplicaciones de la tecnología LIDAR, apunta hacia la generación de mapas para la navegación autónoma. En [18] se propone un sistema de reconstrucción de mapas para un vehículo autónomo equipado con tecnología LIDAR HDL-64E, capaz de obtener más de un millón de puntos por segundo. Dentro del artículo se propone la construcción rápida de Mapas Locales 2D que servirán para la navegación local autónoma, y la construcción de Mapas Globales 2D y 3D que modelizan el ambiente recorrido por el vehículo. Durante el proceso de construcción del Mapa Global se calcula la localización del vehículo con respecto a su posición inicial.

En fin, son un sin número de aplicaciones de esta tecnología, de hecho [19], a través de una revisión bibliográfica, expone la infinidad de aplicaciones del LIDAR, entre las que destacan:

- Planificación y Modelado urbano.
- Arqueología.
- Turismo virtual.



- Simulación.
- Restauración.
- Navegación de automóviles y servicios de navegación.
- Telecomunicaciones.
- Elaboración de mapas de riesgo de inundación.
- Análisis de terreno para exploración de petróleo y gas.
- Arquitectura, Ingeniería y construcción.
- Evaluación del terreno para el estudio de desarrollo inmobiliario.
- Planificación de líneas de transmisión de potencia o corredores de tuberías.
- Cartografía de las zonas costeras.
- Silvicultura.
- Análisis de zonas de acceso restringido.
- Monitoreo atmosférico.
- Modelado Digital por Elevación con LIDAR aerotransportado.
- Aplicaciones oceanográficas. Etc.

---

## Capítulo 3: Marco teórico

---

En este capítulo se presentan los principios teóricos que sustentan la investigación. Estos fundamentos teóricos están constituidos por el estudio de las especificaciones técnicas del sensor HDL-64E, la descripción del principio de funcionamiento del tipo de mecanismo encargado de inclinar el sensor para ampliar su ángulo de visión vertical (mecanismo de tres barras), el estudio de los controladores PID en posición angular, y la descripción de la estructura de las matrices homogéneas y su aplicación para la realización de transformaciones en el espacio 3D.

### 3.1 Sensor LIDAR HDL-64E

Con el propósito de conocer y comprender los conceptos básicos de operación del sensor a continuación se presenta:

- Su principio de funcionamiento y especificaciones.
- El tipo y manejo de los paquetes de datos que provee.
- Los factores de corrección que se aplican a sus adquisiciones.

#### 3.1.1 Principio de funcionamiento y especificaciones

El HDL-64E constituye la segunda generación de sensores láser elaborado por la compañía norteamericana Velodyne. El prototipo de primera generación fue utilizado en el desafío DARPA para navegación autónoma de vehículos en 2005. El desempeño del sensor fue tan exitoso que para 2007 doce equipos habían adquirido e instalado ya el HDL-64E como sistema de navegación.

El principio de funcionamiento del sensor es muy sencillo, obtiene las distancias de los objetos situados a su alrededor utilizando el principio del medidor láser de distancia. Un emisor láser hace incidir un rayo, el cual rebota en la superficie de un objeto y un receptor lo captura calculando la distancia de acuerdo al tiempo que tardó en regresar, si tarda más tiempo del esperado o no regresa entonces no se toma en cuenta. Con los ángulos en que se proyectó el

láser (azimut y colatitud) y la distancia calculada se construye un vector en forma esférica, y así se obtiene un punto en el espacio  $R^3$ . Cuando se acumula una serie de puntos se forma una nube de puntos (ver figura 3.1). Entre más densa sea la nube de puntos mayor resolución tendrá la imagen.

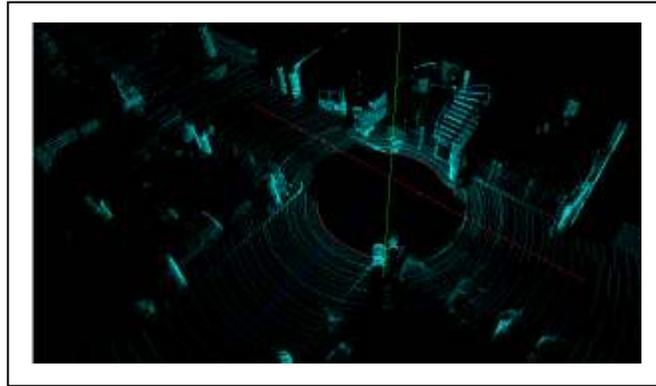


Fig. 3. 1. Ejemplo de nube de puntos adquirida con el HDL-64E en un estacionamiento.

El HDL-64E tiene montado un arreglo de 64 láseres (ver figura 3.2), divididos en un bloque superior y otro inferior de 32 cada uno. Toda la unidad gira alrededor del eje vertical ofreciendo un campo de visión horizontal de 360°, en donde se entrega cada 0.09° el ángulo de proyección  $\theta$  (azimut). Por otro lado, el sensor tiene un campo de visión vertical de 26.8°, el cual va desde 2 hasta -24.8° debido al acomodo descendente de los láseres según el ángulo de proyección  $\varphi$  (colatitud). Cada láser tiene una longitud de onda de 905 nm con pulsos de 10 ns, lo que provee una gran cantidad de datos y en consecuencia una nube de puntos mucho más robusta que los diseños convencionales. El HDL-64E es de clase 1M, seguro para la vista, cuenta con un alcance de hasta 120 m y una precisión de 5 cm.

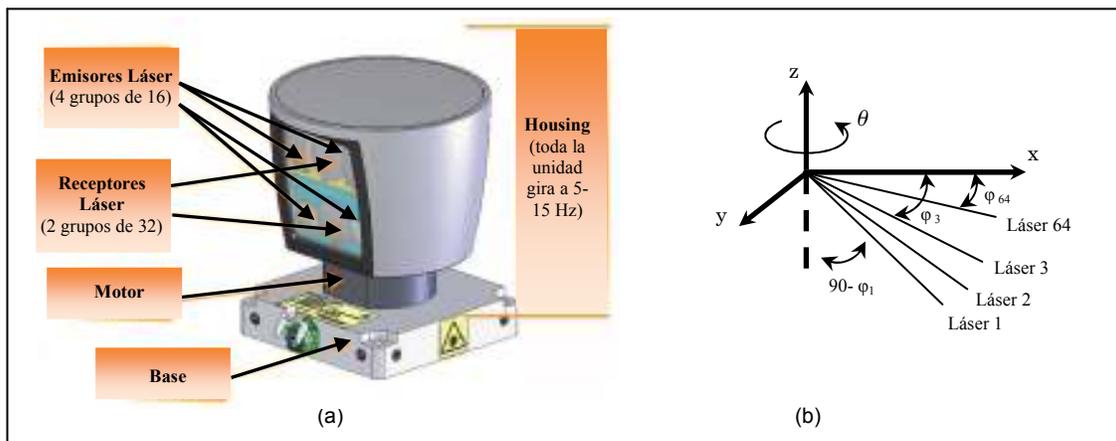


Fig. 3. 2. Diseño del LIDAR HDL-64E. (a) Localización de los emisores y receptores. (b) Ilustración de los ángulos  $\theta$  (azimut) y  $\varphi$  (colatitud).

El dispositivo cuenta con su cable de alimentación, el cual debe ser conectado a 12 Volts (4-6 Amperes), cualquier voltaje mayor a 16 Volts puede dañar la unidad. Adicionalmente se tienen 2 interfaces: Ethernet y Serial, con conectores RJ-45 y RS-232 DB9 respectivamente. El conector Ethernet está diseñado para ser conectado a una computadora estándar y permitir la transferencia de datos (en la sección 3.1.2 se aborda lo referente al tipo y formato de los paquetes de datos). La interfaz serial permite realizar actualizaciones al firmware del sensor, además se utiliza para hacer modificaciones en la velocidad de giro de la unidad.

La unidad rota horizontalmente y dispara los láseres por grupos ya sea superior o inferior. Para obtener un mayor desempeño a grandes distancias se emiten y capturan tres veces el grupo de láseres superiores por sólo uno del inferior (ver figura 3.3).

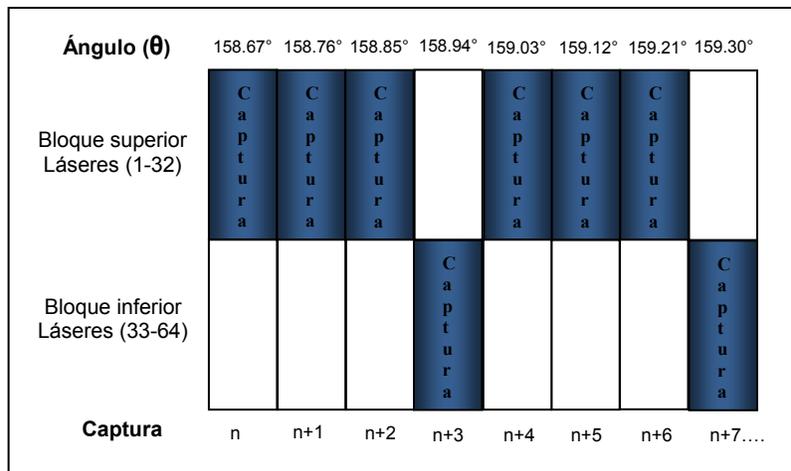


Fig. 3. 3. Ejemplo de la secuencia de captura de datos.

### 3.1.2 Paquetes de datos

El HDL-64E entrega paquetes UDP Ethernet. Cada paquete contiene 1206 bytes que consisten en 12 bloques de 100 bytes seguidos de 6 bytes al final de cada paquete que incluyen un contador de giro e información de la versión del firmware o bien la temperatura de la unidad. Cada paquete puede ser tanto para el bloque superior de láseres o como para el inferior. El formato de los paquetes se presenta a continuación.

Encabezado: contiene 2 bytes que indican cuando indican cuando un paquete pertenece al bloque superior o inferior. El bloque superior se representa con 0xEEFF y el inferior con 0xDDFF.

Información rotacional: incluye 2 bytes que representan un entero entre 0 y 35999. Es necesario dividir este número entre 100 para obtener el ángulo real  $\theta$  (azimut).



Información de los láseres: cada láser (del bloque superior o inferior) proporciona en 2 bytes información de la distancia con incrementos de 0.2 centímetros y en 1 byte información de la intensidad (0-255, donde 255 representa la mayor intensidad). Cuando un láser no proporciona esta información significa que el objeto está fuera del alcance del sensor o bien está muy cerca de él (la mínima distancia para sensar es aproximadamente 3 pies).

Estado: 6 bytes al final de cada paquete muestran el número de versión del firmware en ASCII con el formato "Vn.n" o la temperatura interna de la unidad, la cual se muestra con 2 bytes seguidos por la cadena "DegC" en ASCII.

Velodyne incluye un software visualizador llamado DSR (Digital Sensor Recorder), cuyo instalador está en el CD que viene con la unidad. DSR lee los paquetes arrojados por el HDL-64E y realiza los cálculos necesarios para graficar los puntos en el espacio 3D y poder visualizarlos.

### 3.1.3 Factores de corrección a las adquisiciones

Cada HDL-64E viene con un archivo único en formato .XML, el cual fue generado como resultado de la calibración realizada en la fábrica de Velodyne. DSR usa este archivo para corregir la información suministrada por el sensor y desplegar las nubes de puntos con precisión. El archivo .XML contiene un valor determinado de cada uno de los siguientes 5 factores de corrección para cada uno de los 64 láseres.

Tabla 3. 1. Descripción de los factores de corrección aplicados a las capturas del DHL-64E.

<b>Factor de corrección</b>	<b>Descripción</b>
<b>rotCorrection</b>	Es el ángulo de corrección rotacional para cada láser visto desde atrás de la unidad. Valores positivos rotan hacia la izquierda y negativos hacia la derecha.
<b>vertCorrection</b>	Este ángulo es una corrección vertical para cada láser visto desde atrás de la unidad. Valores positivos apuntan hacia arriba y valores negativos apuntan hacia abajo.
<b>distCorrection</b>	Distancia de corrección en centímetros para cada láser debido al diseño. Este valor debe ser sumado directamente a las distancias leídas de los paquetes.
<b>vertoffsetCorrection</b>	Distancia en centímetros que representa la altura de cada láser medida desde la parte inferior de la base.
<b>horizOffsetCorrection</b>	Distancia de corrección horizontal en centímetros para cada laser. Es una constante positiva o negativa.

El archivo .XML único para cada HDL-64E puede ser utilizado por usuarios que deseen crear su propia software de visualización.

Las adquisiciones arrojadas en los paquetes UDP incluyen el ángulo  $\theta$  (azimut) y la distancia sensada por los láseres, al aplicar los factores de corrección se obtienen coordenadas esféricas, las cuales serán transformadas a cartesianas para su visualización. Para mayor información acerca de la calibración del HDL-64E y la forma en que DSR aplica los factores de corrección para estimar las coordenadas cartesianas de los puntos consultar [20].

Para examinar a fondo las especificaciones técnicas del HDL-64E revisar [21], el cual puede ser descargado de la página web del fabricante ([www.velodyne.com](http://www.velodyne.com)).

### 3.2 Mecanismos de tres barras

Para incrementar el ángulo de visión vertical del sensor en este trabajo se propone construir un dispositivo mecánico que permita con un solo actuador inclinar el HDL-64E y fijarlo en una posición angular deseada. El dispositivo diseñado constituye un mecanismo de tres barras. El funcionamiento de este tipo de mecanismos se presenta a continuación.

Los mecanismos de tres barras o también conocidos como de “triángulo articulado con un lado de longitud variable” [22], son aquellos que basan su funcionamiento en la variación de longitud de una de sus barras mientras las dos restantes permanecen constantes.

Un ejemplo muy clásico de este tipo de mecanismos lo constituye el gato elevador. En la figura 3.4 se muestran dos tipos de gatos elevadores que a pesar de aparentar ser diferentes funcionan exactamente igual. En ambos tipos de gato elevador la distancia CD varía al girar el husillo roscado que conforma el segmento variable AB.

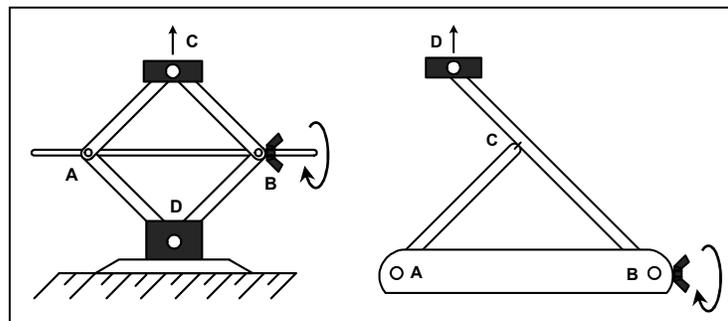


Fig. 3. 4. Funcionamiento del gato elevador

Existen muchos dispositivos de uso cotidiano que funcionan gracias a este simple mecanismo e independientemente de su configuración y aplicación el principio es el mismo. En la figura 3.5 se presenta algunos ejemplos de la aplicación de este mecanismo acompañado de un bosquejo diagrama cinemático.

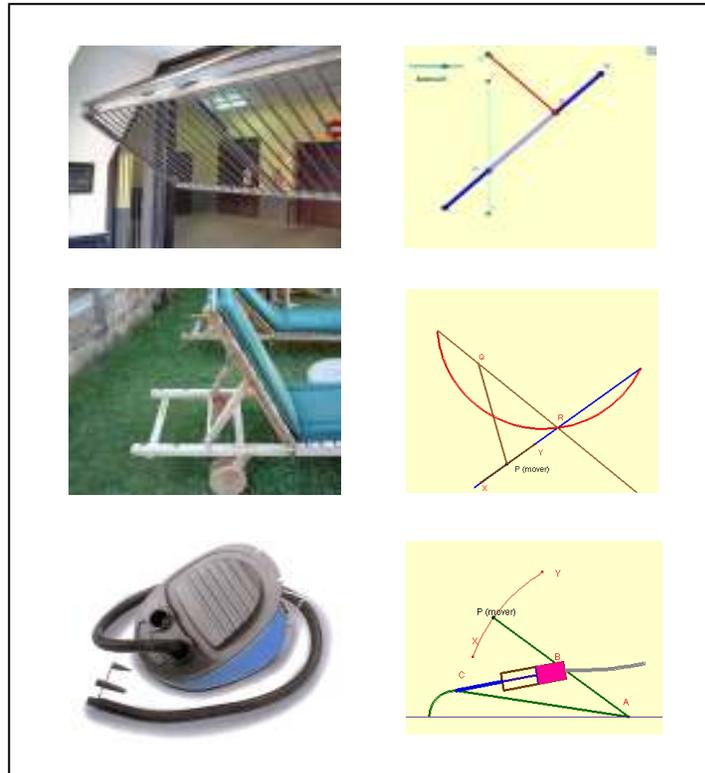


Fig. 3. 5. Algunas aplicaciones de mecanismos de tres barras.

El análisis cinemático de este tipo de mecanismos depende de lo que se quiera considerar como variable de entrada y salida (impulsor y seguidor), y estriba básicamente en la aplicación de leyes de senos y cosenos. Para conocer más a fondo particularidades de este tipo de mecanismos se puede consultar [23], el cual es un estudio clásico para el análisis de mecanismos desde la óptica del estudio de polígonos.

### 3.3 Controladores PID

Para controlar automáticamente el posicionamiento angular del LIDAR HDL-64E a través de un mecanismo de tres barras se plantea la utilización de un control PID, cuyo principio de funcionamiento se resume a continuación.

Los motores de corriente directa son los tipos de actuadores más comunes para servomecanismos. Los más utilizados son los motores de corriente directa de imán permanente con escobillas y sin escobillas. La función de transferencia que representa la relación entre la posición angular de salida,  $\theta(s)$ , de un motor de corriente directa y su voltaje de alimentación,  $E_a(s)$ , es la que se presenta a continuación:

$$\frac{\theta(s)}{E_a(s)} = \frac{K_m}{S(T_m S + 1)} \quad (1)$$

Con:

$$K_m = \frac{K_t}{R_a B + K_t K_e} \quad T_m = \frac{R_a J}{R_a B + K_t K_e}$$

Donde:

$K_t$ : Constante de par del motor.

$R_a$ : Resistencia de la armadura.

$B$ : Coeficiente de fricción viscosa.

$K_e$ : Constante de velocidad del motor.

$J$ : Momento de inercia.

La función de transferencia de la ecuación 1 se deriva del modelado electromecánico de los motores de corriente directa [24].

La determinación de los coeficientes de la función de transferencia,  $K_m$  y  $T_m$ , o identificación del sistema puede realizarse mediante varios métodos. Los más comunes son a partir de los datos de la placa del motor o bien utilizando técnicas de identificación experimental como la "identificación gráfica de parámetros como respuesta a una señal escalón para procesos de primer orden" [25] y el ajuste de curvas mediante polinomios de interpolación de Lagrange [26] o mínimos cuadrados [27]. Para cualquiera de las técnicas de identificación experimental es necesario aplicar una señal de excitación,  $E_x(s)$ , al sistema y analizar su respuesta, usualmente en velocidad,  $\Omega(s)$  (ver figura 3.6).

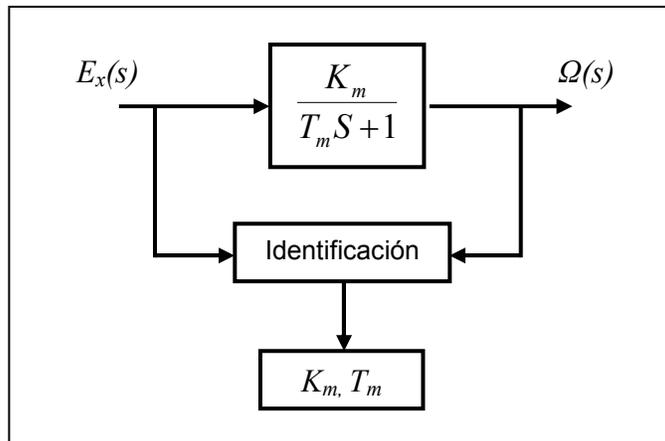


Fig. 3. 6. Diagrama para la identificación de un motor de corriente directa.

El comportamiento en posición angular de un motor de corriente directa es inestable, esto se debe a que una de las raíces del polinomio característico de su función de transferencia (polos) es igual a cero (ver ecuación 1). Esta situación tiene una explicación lógica, pues cuando alimentamos un motor en lazo abierto con un voltaje determinado éste inicia a girar ubicándose en infinitas posiciones angulares.

Para estabilizar el sistema se incorpora una retroalimentación, a esto se le conoce como cerrar el lazo de control. En un lazo de control cerrado la posición angular de salida,  $\theta(s)$ , es restada iterativamente de una posición de referencia,  $\theta_r(s)$ , para formar una señal de control tipo error,  $E(s)$ , que permita alcanzar un “Set Point” (ver figura 3.7).

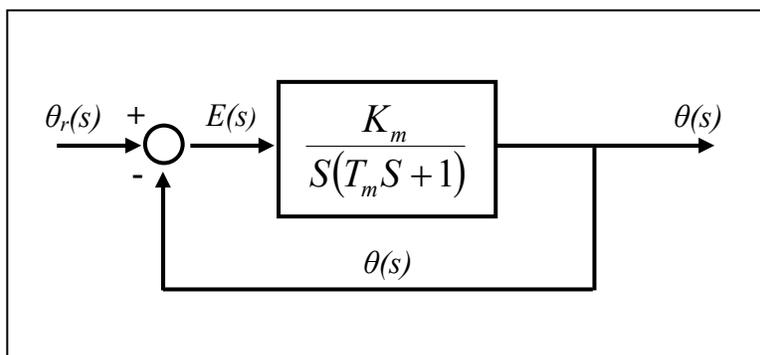


Fig. 3. 7. Diagrama en lazo cerrado en posición angular de un motor de corriente directa.

Al cerrar el lazo de control la función de transferencia expresada en la ecuación 1 se puede redefinir de la siguiente manera:



$$\frac{\theta(s)}{\theta_r(s)} = \frac{K_m}{S(T_m S + 1) + K_m} \quad (2)$$

O bien:

$$\frac{\theta(s)}{\theta_r(s)} = \frac{K_m}{T_m S^2 + S + K_m} \quad (3)$$

En la ecuación 3 debido a que los valores de  $K_m$  y  $T_m$  serán siempre mayores que cero es imposible que las raíces del polinomio característico (polos) adquieran valores de cero o valores reales positivos, por lo que el sistema se considera como estable.

La ecuación 3 podría definirse de la siguiente forma:

$$\frac{\theta(s)}{\theta_r(s)} = \frac{\frac{K_m}{T_m}}{S^2 + \frac{1}{T_m} S + \frac{K_m}{T_m}} \quad (4)$$

La ecuación 4 corresponde a un sistema de segundo orden de la forma:

$$\frac{C(s)}{R(s)} = \frac{\omega_n^2}{S^2 + 2\xi\omega_n S + \omega_n^2} \quad (5)$$

Con:

$$\omega_n = \sqrt{\frac{K_m}{T_m}} \qquad \xi = \frac{1}{2T_m \omega_n}$$

Donde:

$C(s)$ : Salida del sistema.

$R(s)$ : Entrada del sistema.

$\omega_n$  : Frecuencia natural no amortiguada del sistema.

$\xi$  : Factor de amortiguamiento relativo.

Al aplicar la transformada inversa de Laplace a la ecuación 5 se puede obtener la siguiente respuesta en el tiempo (ver figura 3.8).

$$C(t) = 1 - e^{-\xi\omega_n t} \left( \cos \omega_d t + \frac{\xi}{\sqrt{1-\xi^2}} \sin \omega_d t \right) \quad (6)$$

Con:

$$\omega_d = \omega_n \sqrt{1-\xi^2}$$

La respuesta de un sistema de control práctico muestra con frecuencia oscilaciones amortiguadas antes de alcanzar el estado estacionario. Al especificar las características de la respuesta transitoria de un sistema de control es común especificar los siguientes índices de desempeño.

Tiempo de retardo,  $t_d$ : es el tiempo requerido para que la respuesta alcance la primera vez la mitad del valor final.

Tiempo de subida,  $t_r$ : es el tiempo requerido para que la respuesta alcance el 100 % de su valor final. La ecuación para calcularlo es:

$$t_r = \frac{1}{\omega_d} \left( \pi - \tan^{-1} \left( \frac{\sqrt{1-\xi^2}}{\xi} \right) \right) \quad (7)$$

Tiempo pico,  $t_p$ : es el tiempo requerido para que la respuesta alcance el primer pico de sobreelongación. La ecuación para calcularlo es:

$$t_p = \frac{\pi}{\omega_d} \quad (8)$$

Sobreelongación máxima (porcentaje),  $M_p$ : es el máximo valor del pico de la curva de respuesta. La ecuación para calcularla es:

$$M_p = e^{-\left( \frac{\xi}{\sqrt{1-\xi^2}} \right) \pi} \quad (9)$$

Tiempo de asentamiento,  $t_s$ : es el tiempo que se requiere para que la curva de respuesta alcance un rango alrededor del valor final del tamaño especificado por el porcentaje absoluto del valor final (por lo general de 2 o 5%). La ecuación para calcularlo es:

$$t_s = \frac{4}{\xi\omega_n} \quad (\text{criterio del 2\%}) \quad (10)$$

O bien:

$$t_s = \frac{3}{\xi\omega_n} \quad (\text{criterio del 5\%}) \quad (11)$$

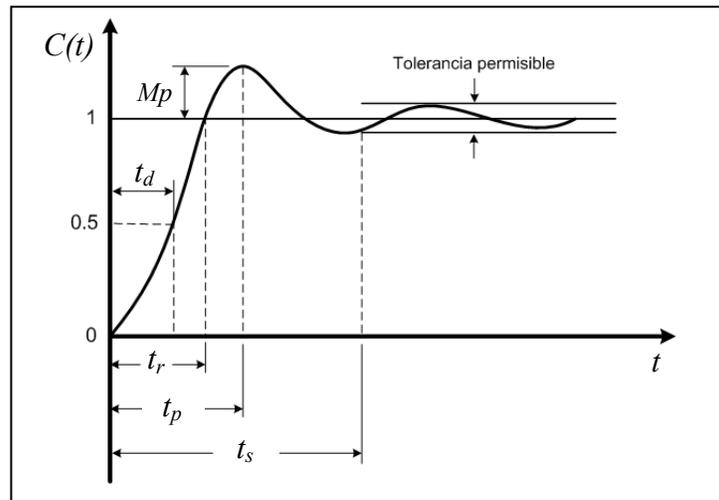


Fig. 3. 8. Respuesta en el tiempo de un sistema de segundo orden.

En las ecuaciones 7, 8, 9, 10 y 11, es posible observar que los índices de desempeño dependen de  $\omega_n$  y de  $\xi$ , de manera que es posible alcanzar cualquier índice de desempeño ajustando la frecuencia natural no amortiguada y/o el factor de amortiguamiento relativo.

La frecuencia natural no amortiguada y el factor de amortiguamiento relativo se ajustan a través de una acción de control. Esto quiere decir que sólo a través de una acción de control se pueden alcanzar los índices de desempeño deseados (especificaciones para un controlador).

Es importante mencionar que los índices de desempeño deben ser definidos antes de diseñar el controlador para imponer el comportamiento al sistema según las necesidades.

El tipo de controlador más utilizado en la industria es el PID (Proporcional Integral Derivativo), su diagrama esquemático se presenta en la figura 3.9.

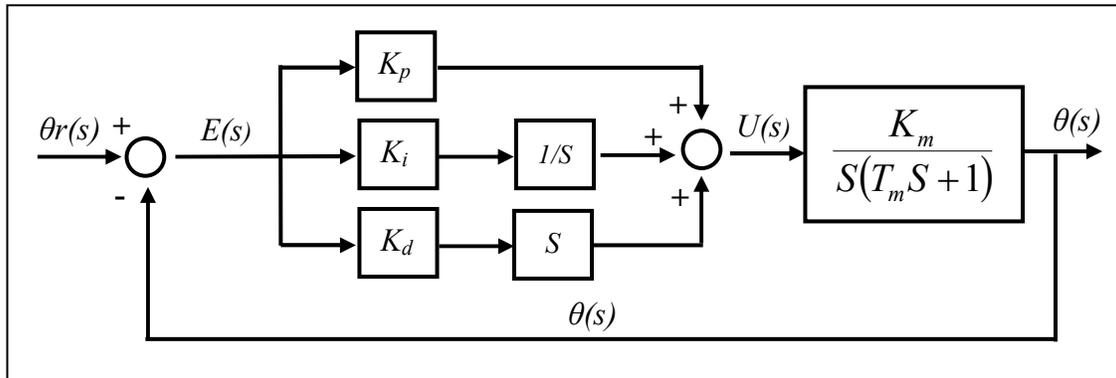


Fig. 3. 9. Diagrama esquemático de un controlador PID.

La función de transferencia de un controlador PID básico como el mostrado anteriormente es la siguiente:

$$\frac{U(s)}{E(s)} = K_p + K_i \frac{1}{S} + K_d S \quad (12)$$

Donde:

$K_p$  : Ganancia proporcional.

$K_i$  : Ganancia integral.

$K_d$  : Ganancia derivativa

Un controlador tipo PID incorpora una acción proporcional, una acción integral ( $1/S$ ) y una derivativa ( $S$ ). La acción proporcional, muchas veces visualizada como una amplificación, envía una señal de control proporcional al error, haciendo que el sistema alcance más rápido el “Set Point”, sin embargo, cuando la acción proporcional es muy grande el sistema se puede volver inestable produciendo sobreoscilación. Por otro lado la acción integral envía una señal de control proporcional al acumulado de los errores a lo largo del tiempo. La sumatoria de los errores instantáneos a lo largo del tiempo provee el offset acumulado que debió haber sido corregido previamente. La acción integral cuando se combina con la acción proporcional acelera el comportamiento del sistema hacia el “Set Point” y elimina el error residual en estado estacionario, sin embargo, dado que la acción integral responde a errores acumulados del

pasado usualmente genera sobreelongaciones inusuales. Finalmente la acción derivativa viene a reducir la magnitud de las sobreelongaciones producidas por la acción integral mejorando la respuesta transitoria del sistema. La acción derivativa tiene este efecto pues considera la tasa de cambio del error.

Anteriormente se había planteado que cuando se cierra el lazo de control la función de transferencia expresada en la ecuación 1 adquiere la forma de la ecuación 5, en donde los valores de la frecuencia natural no amortiguada,  $\omega_n$ , y del factor de amortiguamiento relativo,  $\zeta$ , dependen únicamente de los valores de,  $K_m$  y  $T_m$ . Sin embargo, cuando se introduce un controlador PID (ver figura 3.9) con una función de transferencia como la formulada en la ecuación 12, estos elementos dependerán ahora también de las ganancias  $K_p$ ,  $K_i$  y  $K_d$ . De tal manera que si la función de transferencia de un sistema es conocida, al definir los valores apropiados de las ganancias  $K_p$ ,  $K_i$  y  $K_d$ , es posible encontrar los valores de  $\omega_n$  y  $\zeta$  que permitan alcanzar cualquier índice de desempeño deseado, es decir cualquier especificación para un controlador.

Al proceso de estimación de las ganancias de un controlador PID de acuerdo a las especificaciones que se tenga se le conoce como “Sintonización”.

Existen muchos métodos para la sintonización de un PID. A continuación se menciona los más utilizados.

- Método manual.
- Método de Ziegler-Nichols.
- Vía software especializados de sintonización.
- Método de Cohen-Coon.

Para mayor información acerca controladores PID y su sintonización se puede consultar [28].

### **3.4 Matrices de transformación homogéneas**

La acción de inclinar el HDL-64E para fijarlo en una posición angular deseada y poder así aumentar su ángulo de visión vertical genera un efecto de transformación 3D sobre las adquisiciones del sensor. Estas transformaciones obedecen al cambio de sistema de referencia desde una posición horizontal (condición normal del sensor sin inclinación) hasta una posición

angular deseada. Cualquier transformación en el espacio 3D puede ser representada a través de matrices homogéneas, por tal razón su estudio es de suma importancia para este trabajo.

Para poder entender las matrices homogéneas y su utilidad es necesario primero estudiar dos de sus principales componentes; las matrices de rotación y los vectores de traslación.

### 3.4.1 Matrices de rotación

A menudo es necesario convertir de un sistema de referencia a otro mediante rotaciones alrededor de los ejes de coordenadas. Un ejemplo de esto es cuando en un momento dado se representa la posición de un satélite artificial en un sistema de referencia  $O$ . Un tiempo después, la posición del satélite habrá variado debido al movimiento de su órbita, pero también la Tierra habrá rotado, de modo que se tiene un nuevo sistema de referencias  $O'$  (ver figura 3.10).

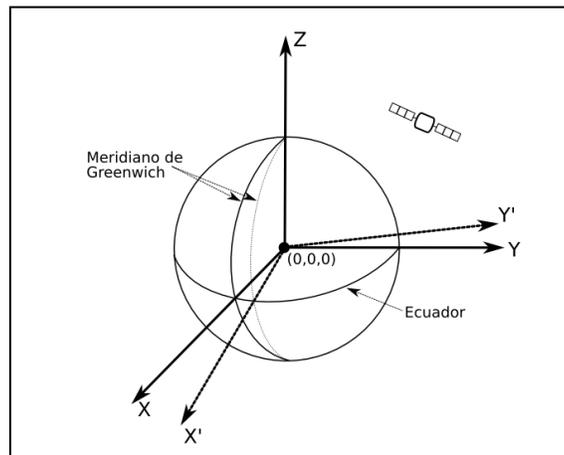


Fig. 3. 10. Posición de un satélite artificial respecto a dos sistemas de referencia  $O$  y  $O'$ .

En el ejemplo anterior es importante notar que la rotación únicamente fue experimentada alrededor del eje  $Z$ . A continuación se analiza este tipo de rotación.

En la figura 3.11 se ilustra la rotación alrededor del eje  $Z$  del sistema  $XY$  en  $X'Y'$ . Nótese que la distancia  $r$  desde el origen al punto  $P$  es igual en ambos sistemas de modo que las coordenadas  $(x, y)$  y  $(x', y')$  se pueden escribir de la siguiente forma:

$$x = r \cos \alpha \tag{13}$$

$$y = r \sin \alpha \tag{14}$$

$$x' = r \cos(\alpha - \theta) \tag{15}$$

$$y' = r \sin(\alpha - \theta) \tag{16}$$

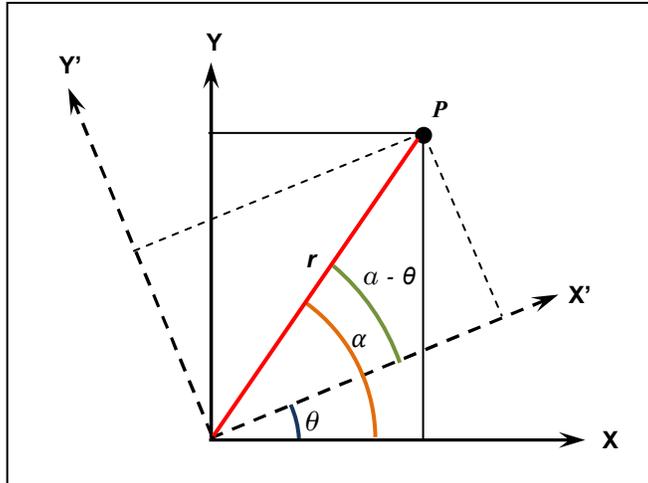


Fig. 3. 11. Posición de un punto  $P$  producto de una rotación pura en el eje  $Z$ .

Si aplicamos las leyes de cosenos para la diferencia de ángulos podemos expresar la ecuación 15 de la siguiente forma:

$$\begin{aligned}
 x' &= r(\cos \alpha \cos \theta + \sin \alpha \sin \theta) \\
 x' &= r \cos \alpha \cos \theta + r \sin \alpha \sin \theta \\
 x' &= x \cos \theta + y \sin \theta
 \end{aligned}
 \tag{17}$$

De igual manera si aplicamos la ley de los senos para la diferencia de ángulos sobre la ecuación 16, es posible obtener:

$$\begin{aligned}
 y' &= r(\sin \alpha \cos \theta - \cos \alpha \sin \theta) \\
 y' &= r \sin \alpha \cos \theta - r \cos \alpha \sin \theta \\
 y' &= -x \sin \theta + y \cos \theta
 \end{aligned}
 \tag{18}$$

Las ecuaciones 17 y 18 se pueden escribir de forma matricial como se muestra a continuación:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}
 \tag{19}$$

La ecuación 19 determina la rotación en dos dimensiones. Para obtener la expresión tridimensional basta saber que las coordenadas en el eje  $Z$  se mantienen constantes. De esta forma se obtiene:



$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (20)$$

La convención para el signo del ángulo  $\theta$  es muy importante. El ángulo será positivo cuando la rotación alrededor del eje se realice en el sentido de la regla de la mano derecha. La ecuación 20 es para  $\theta$  positivo (ver figura 3.11). La ecuación 21 representa rotaciones contrarias al sentido de la regla de la mano derecha.

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (21)$$

Si aplicamos el mismo razonamiento matemático planteado anteriormente, es posible expresar una rotación de  $\psi$  grados alrededor del eje X, y una rotación de  $\phi$  grados alrededor del eje Y. Las ecuaciones 22, 23 y 24 describen las matrices de rotación pura sobre los ejes X, Y y Z respectivamente.

$$R(X, \psi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \psi & -\sin \psi \\ 0 & \sin \psi & \cos \psi \end{bmatrix} \quad (22)$$

$$R(Y, \phi) = \begin{bmatrix} \cos \phi & 0 & \sin \phi \\ 0 & 1 & 0 \\ -\sin \phi & 0 & \cos \phi \end{bmatrix} \quad (23)$$

$$R(Z, \theta) = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (24)$$

La rotación de un sistema de coordenadas  $P$  a otro sistema de coordenadas  $Q$  se puede expresar como:

$$X_Q = R_P^Q X_P \tag{25}$$

Donde  $X_Q$  y  $X_P$  son vectores definidos en los sistemas  $Q$  y  $P$  respectivamente, y  $R_P^Q$  es la matriz de rotación que convierte de  $P$  a  $Q$ .

Esta matriz  $R_P^Q$  puede ser cualquiera de las matrices de las ecuaciones 22, 23 y 24 o bien podría estar formada por la composición sucesiva de éstas. Por ejemplo la ecuación 26 representa la composición de una rotación primero en el eje X, luego en el eje Y, y finalmente en Z.

$$R_P^Q = R(Z, \theta)R(Y, \phi)R(X, \psi) \tag{26}$$

Es importante acotar que el producto de las matrices de rotación no es conmutativo, por lo que el orden en que se aplican dichas rotaciones es un dato que hay que tomar en cuenta.

Existen muchas expresiones para la matriz  $R_P^Q$ , y por consiguiente muchas formas de definir la orientación de un vector, sin embargo las más utilizadas son: los ángulos Roll-Pitch-Yaw, y los ángulos de Euler.

Ángulos Roll-Pitch-Yaw: Considera tres rotaciones sucesivas alrededor de los ejes de coordenadas de un sistema fijo. Inicia con una rotación alrededor del eje X en un ángulo  $\psi$ , resultando el sistema  $(u', v', w')$ . A continuación una segunda rotación en  $\theta$  alrededor del eje Y, resultando el sistema  $(u'', v'', w'')$ . Finalmente una tercera rotación  $\phi$  alrededor del eje Z resultando en el sistema final  $(u, v, w)$  (Ver figura 3.12).

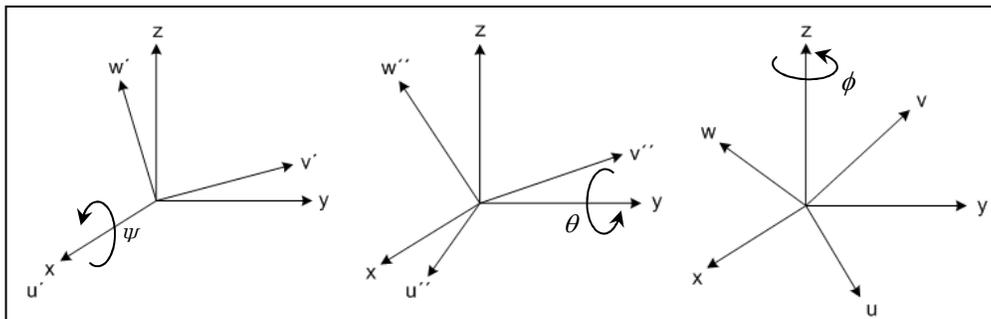


Fig. 3. 12. Secuencia de rotaciones para la representación de los ángulos Roll-Pitch-Yaw.

La matriz de rotación resultante expresada en la ecuación 27 es obtenida de la premultiplicación de las tres matrices básicas de rotación.

$$R(\psi, \theta, \phi) = R(Z, \phi)R(Y, \theta)R(X, \psi)$$

$$R(\psi, \theta, \phi) = \begin{bmatrix} \cos \phi \cos \theta & \cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi & \cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi \\ \sin \phi \cos \theta & \sin \phi \sin \theta \sin \psi + \cos \phi \cos \psi & \sin \phi \sin \theta \cos \psi - \cos \phi \sin \psi \\ -\sin \theta & \cos \theta \sin \psi & \cos \theta \cos \psi \end{bmatrix} \quad (27)$$

La rotación sobre el eje X es llamada Roll, la rotación sobre el eje Y es llamada Pitch, y la rotación sobre el eje Z es llamada Yaw.

Los ángulos de Euler funcionan de manera similar, la única variación la constituyen los ejes donde se realizan las rotaciones (ver figura 3.13).

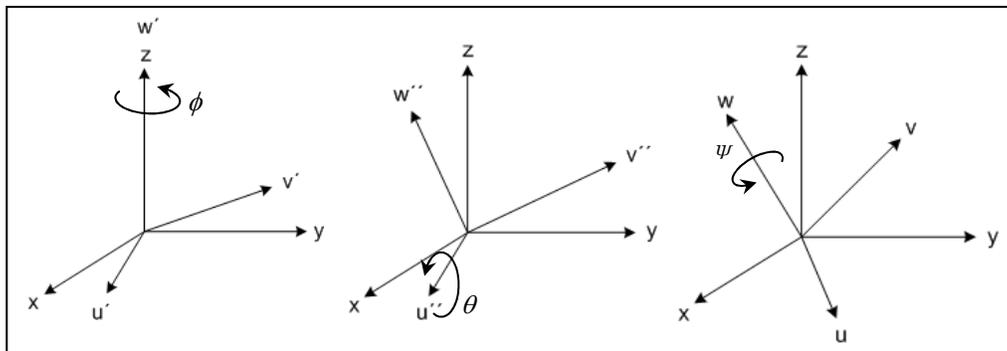


Fig. 3. 13. Secuencia de rotaciones para la representación de los ángulos de Euler.

La matriz de rotación resultante al aplicar las rotaciones según los ángulos de Euler se presenta a continuación:

$$R(w, \phi)R(u, \theta)R(w, \psi) = \begin{bmatrix} \cos \phi \cos \psi - \sin \phi \cos \theta \sin \psi & -\cos \phi \sin \psi - \sin \phi \cos \theta \cos \psi & \sin \phi \sin \theta \\ \sin \phi \cos \psi + \cos \phi \cos \theta \sin \psi & -\sin \phi \sin \psi + \cos \phi \cos \theta \cos \psi & -\cos \psi \sin \theta \\ \sin \theta \sin \psi & \sin \theta \cos \psi & \cos \theta \end{bmatrix} \quad (28)$$

Resulta importante destacar que la matriz de la ecuación 28, a diferencia de la de la ecuación 27, es obtenida a través de una postmultiplicación.

Para mayor información acerca de los ángulos Roll-Pitch-Yaw y los ángulos Euler y algunas de sus variantes consultar [29].

### 3.4.2 Vectores de traslación

Los vectores de traslación representan el desplazamiento de un punto o conjuntos de puntos  $X$  según un vector fijo  $T$  no nulo. Los desplazamientos de forma paralela se pueden aplicar en dos y tres dimensiones. La ecuación 29 describe una traslación en tres dimensiones.

$$X_Q = T_P^Q + X_P$$

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} + \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (29)$$

### 3.4.3 Matrices de transformación homogéneas

La matriz de transformación homogénea es una matriz de 4X4 que está definida con el propósito de transformar un vector en coordenadas homogénea desde un sistema de coordenadas a otro [29]. Un vector en  $R^3$  expresado en coordenadas homogéneas incluye una cuarta coordenada que representa un factor de escala que para efectos prácticos se define como la unidad.

Una matriz homogénea  $H_P^Q$  está conformada por cuatro componentes tal y como se muestra a continuación:

$$H_P^Q = \begin{bmatrix} R_{3x3} & T_{3x1} \\ F_{1x3} & E_{1x1} \end{bmatrix} \quad (30)$$

La matriz superior izquierda ( $R_{3x3}$ ) representa la rotación existente entre dos sistemas de referencia, el vector superior derecho ( $T_{3x1}$ ) la traslación, el vector inferior izquierdo ( $F_{1x3}$ ) la perspectiva y el elemento inferior derecho ( $E_{1x1}$ ) un factor de escalamiento. Para muchas aplicaciones prácticas el vector de perspectiva se considera nulo y el factor de escalamiento igual a la unidad.

Una matriz homogénea relaciona fácilmente dos sistemas de referencia que han experimentado rotaciones y traslaciones uno con respecto al otro. Esto permite transformar cualquier vector  $X_P$



, expresado en un sistema de referencia, en su equivalente  $X_Q$ , bajo otro sistema de referencia, de la siguiente manera:

$$X_Q = H_P^Q X_P \quad (31)$$

Si se expresa los vectores  $X_Q$  y  $X_P$  en sus respectivas coordenadas rectangulares y la matriz  $H_P^Q$  según la ecuación 30 se obtiene:

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} R_{3 \times 3} & T_{3 \times 1} \\ F_{1 \times 3} & E_{1 \times 1} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (32)$$

---

## Capítulo 4: Descripción de la base experimental

---

En este capítulo se detalla cada una de las fases llevadas a cabo para el cumplimiento de los objetivos y la validación de la hipótesis.

Estas fases son las siguientes:

- Diseño y construcción del prototipo mecánico que se encargará de aumentar el ángulo de visión vertical del LIDAR HDL-64E.
- Desarrollo de un controlador PID en posición angular para automatizar el funcionamiento de prototipo mecánico.
- Desarrollo de una metodología para la estimación de las matrices homogéneas que rigen las transformaciones que experimentan las capturas del HDL-64E al momento de ser inclinado para ampliar su campo de visión vertical.

### 4.1 Diseño y construcción del prototipo

#### 4.1.1 Diseño conceptual

La función principal del prototipo diseñado es inclinar el LIDAR HDL-64E para ampliar su campo de visión vertical. La idea básica es montar el sensor sobre una plataforma de inclinación y a través de una acción mecánica llevarlo hasta una posición angular deseada.

En cuanto a las especificaciones iniciales para el diseño del prototipo se tomó en cuenta lo siguiente:

- El prototipo debe permitir el fácil montaje y desmontaje del HDL-64E.
- Debe estar acondicionado para montarse en la parte superior de un vehículo con canastilla porta equipaje estándar.
- Debe ser ligero, pero suficientemente rígido para reducir las vibraciones del vehículo en movimiento.
- Debe proporcionar un aumento en el campo de visión del sensor que permita realizar escaneos de estructuras de aproximadamente 10 metros de altura.

Todas las especificaciones presentadas anteriormente son muy importantes, sin embargo, la que respecta a la ampliación del campo de visión para escanear hasta 10 metros de altura es la más crítica. Para poder satisfacer esta especificación, fue necesario primero conocer la altura máxima que actualmente es posible escanear con el sensor.

Actualmente los escaneos con el LIDAR HDL-64E se realizan montándolo en la parte superior de un automóvil, el cual debe contar con una canastilla porta equipaje estándar y una base rígida para colocar el sensor (ver figura 4.1).



Fig. 4. 1. Condición actual de uso del LIDAR-HDL64E.

Bajo esta condición de uso y tomando en cuenta que los escaneos se realizan transitando sobre la calle, para poder estimar la altura máxima que se puede sensar, es necesario hacer algunas consideraciones dimensionales tales como las que se muestran en la figura 4.2.

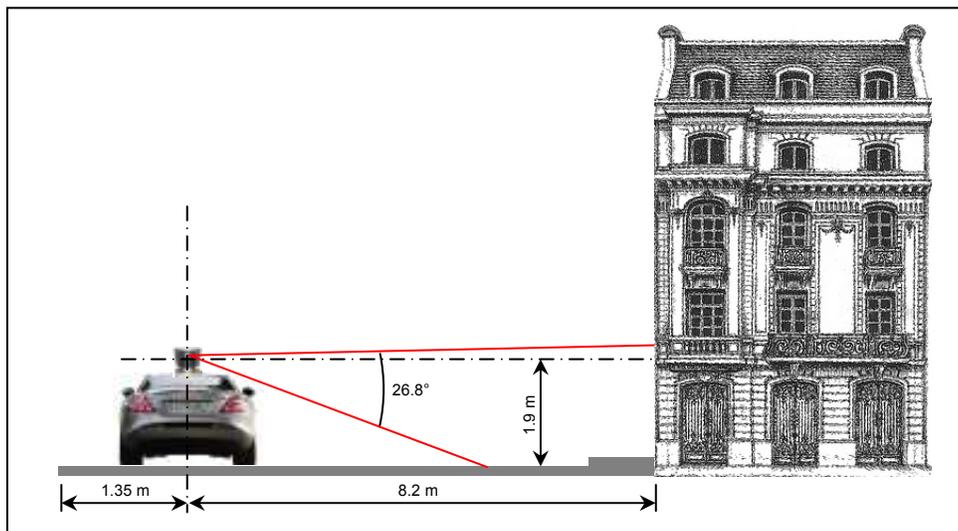


Fig. 4. 2. Consideraciones para poder estimar la altura máxima a la que se puede sensar.

En la figura 4.2 las dimensiones de la calle fueron tomadas con base al promedio de algunas calles del centro de la ciudad de Querétaro. Evidentemente estas dimensiones están sujetas a cambio, sin embargo, es necesario definir un estándar. Otro dato útil a tomar en cuenta es la altura a la que se encuentra el sensor, ésta se estimó en aproximadamente 1.9 m.

Si el campo de visión vertical del sensor es de aproximadamente  $26.8^\circ$ , de los cuales únicamente  $2^\circ$  están por encima de la horizontal, la máxima altura que se puede sensar ( $H_{max}$ ) se puede aproximar de la siguiente forma:

$$H_{max} = D_h \tan(2^\circ) + D_v \quad (33)$$

Donde:

$D_h$  : Es una aproximación de la distancia horizontal desde el HDL-64E hasta la estructura a escanear.

$D_v$  : Es una aproximación de la altura a la que se encuentra el sensor.

Sustituyendo se obtiene:

$$(8.2) \tan(2^\circ) + (1.9) = 2.18$$

De esta manera la altura máxima que se puede sensar actualmente es de aproximadamente 2.18 metros.

Esta altura máxima está muy lejos de la especificación de 10 metros, por lo que el prototipo propuesto en este trabajo debe lograr que la altura máxima de escaneo se incremente en aproximadamente 7.82 metros.

El diseño propuesto sustituye la base rígida sobre la cual se está montando actualmente el HDL-64E (ver figura 4.1). Esto quiere decir que ahora el sensor será montado sobre el prototipo, el cual incluye una plataforma móvil que inclina al sensor para aumentar su ángulo de visión vertical (en la sección 4.1.1.1 se presenta la estructura geométrica del mecanismo).

La inclinación necesaria de la plataforma, expresada en un ángulo  $\theta$ , para poder alcanzar la especificación de 10 metros de altura máxima, se puede estimar de la siguiente forma:

$$\theta = \tan^{-1}\left(\frac{H_{\max} - D_v}{D_h}\right) - 2 \quad (34)$$

Sustituyendo se obtiene:

$$\tan^{-1}\left(\frac{10 - 1.9}{8.2}\right) - 2 = 42.64$$

De esta manera se deduce que el prototipo debe inclinar 43° (redondeando) el sensor para que, permaneciendo constantes las consideraciones dimensionales establecidas en la figura 4.2, se pueda escanear a una altura máxima de 10 metros.

Al inclinar 43° el HDL-64E su campo de visión aumenta de 26.8° a 69.8°, tal y como se muestra en la figura 4.3 (c).

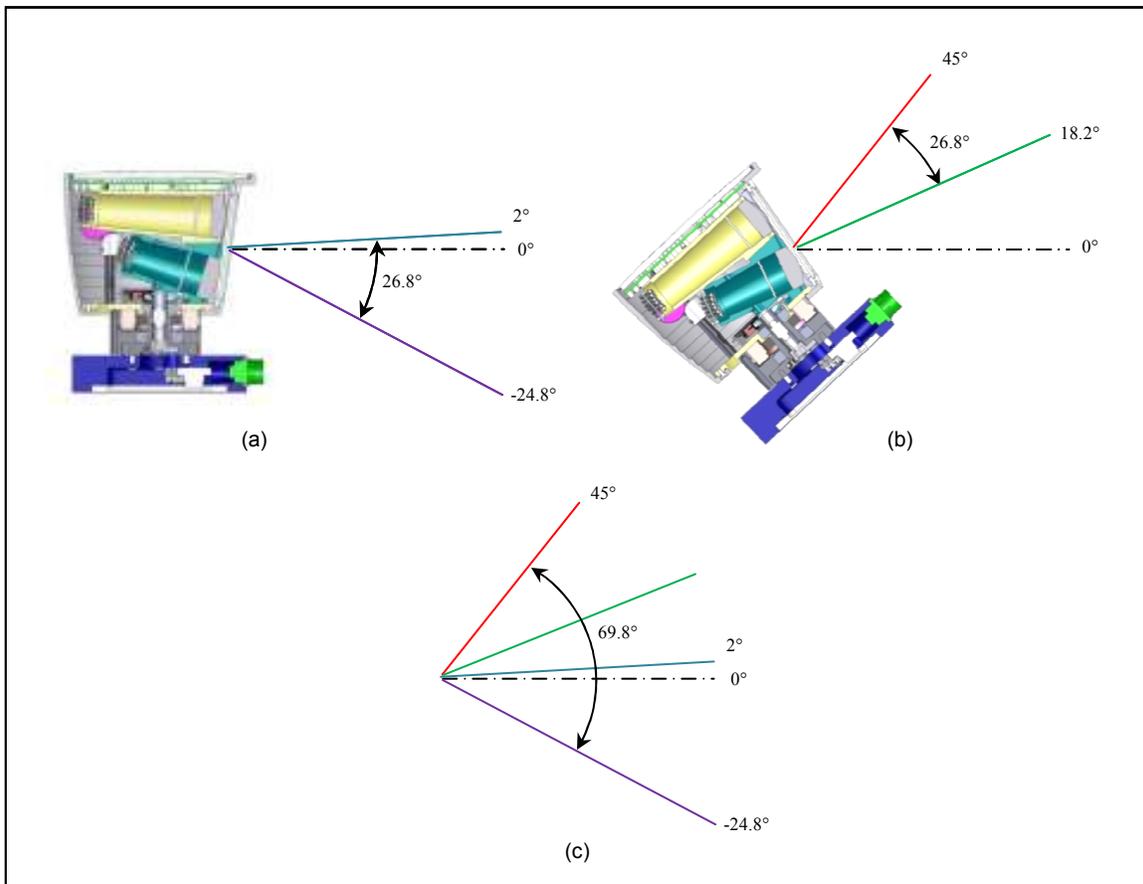


Fig. 4. 3. Bosquejo del aumento del campo de visión vertical del sensor. (a) Campo de visión vertical en la condición inicial. (b) Inclinación del HDL-64E 43°. (c) Campo de visión obtenido al inclinar el HDL-64E 43°.

La generación del nuevo campo de visión del HDL-64E tiene la siguiente explicación. El campo de visión vertical del sensor en condiciones normales va de  $2^\circ$  por encima de la horizontal a  $24.8^\circ$  por debajo de ésta (ver figura 4.3 (a)). Cuando el sensor es inclinado a través de la realización de una serie de barridos consecutivos (según un paso angular necesario) hasta alcanzar una posición máxima a  $43^\circ$  (ver figura 4.3 (b)) su campo de visión ya no será medido a partir de  $2^\circ$  por encima de la horizontal, sino que será medido a partir de  $45^\circ$  (ver figura 4.3 (c)). De ahí que su nuevo campo de visión se considera de aproximadamente  $69.8^\circ$ .

#### 4.1.1.1 Estructura geométrica del mecanismo

En la figura 4.4 se presenta la composición mecánica básica del prototipo. La estructura tubular "ET" mostrada en la figura representa el soporte o base de todo el sistema. Esta estructura tubular está diseñada para poder sujetarse firmemente a un vehículo equipado con una canastilla porta equipaje estándar. La plataforma de inclinación "PI" es donde se monta el LIDAR HDL-64E, por lo que está diseñada con base a las dimensiones de la base del sensor. El sensor se sujeta a la plataforma de inclinación a través de tornillería.

La plataforma de inclinación rota con mínima fricción alrededor de un eje "O" gracias a la incorporación de dos rodamientos idénticos "R". El eje de rotación está sujeto firmemente al soporte tubular a través de un par de abrazaderas ajustables idénticas "A".

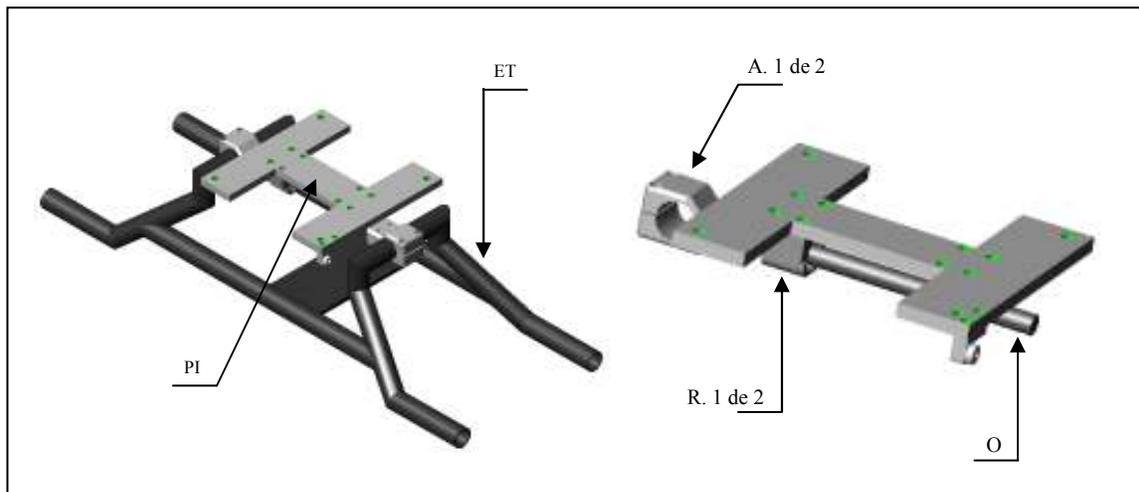


Fig. 4. 4. Composición mecánica básica del prototipo.

El movimiento rotacional de la plataforma es provisto por un único actuador, en este caso un motor de corriente directa "M". Entre el motor y la plataforma de inclinación existe una

transmisión mecánica “TM” (ver figura 4.5) cuyas funciones son: amplificar la potencia del motor para poder manipular los más de 13 kg del HDL-64E, y reducir su velocidad para realizar movimientos angulares suaves. La transmisión mecánica está formada por dos sistemas de reducción: el primer sistema “S<sub>1</sub>” lo constituye un reductor planetario acoplado directamente al motor con una relación de 1:100, y el segundo sistema “S<sub>2</sub>” es un arreglo de dos engranes rectos con una relación de 1:4.

A la salida del segundo sistema de reducción hay una corredera “C”, la cual está conectada con la plataforma de inclinación a través de un pequeño rodamiento. Cuando la plataforma se encuentra en la condición horizontal (ver figura 4.5) y el motor es accionado, esta corredera gira en sentido contrario a las manecillas del reloj, haciendo que la plataforma se incline para realizar el posicionamiento angular del HDL-64E.

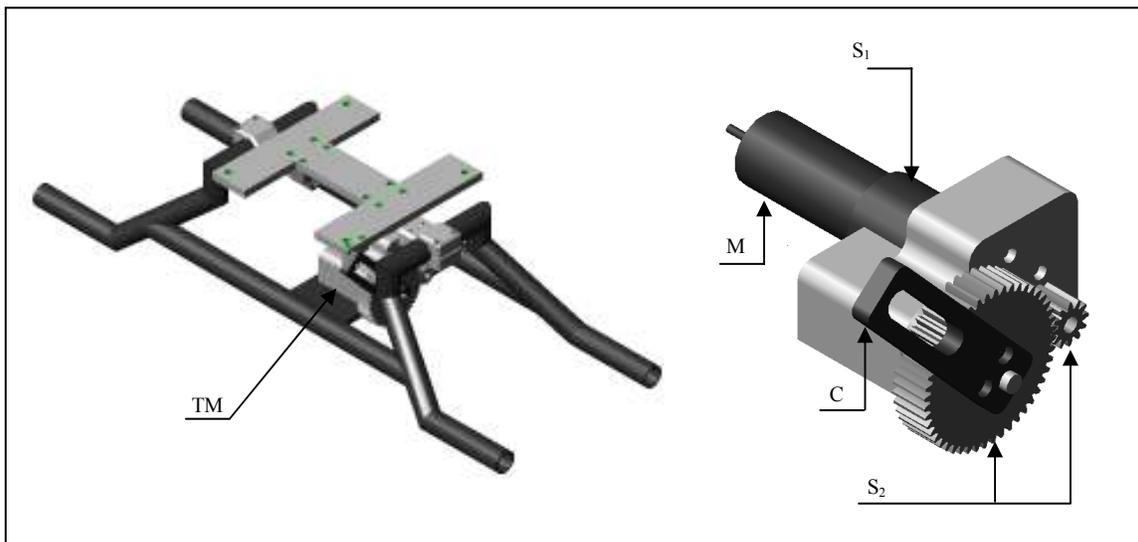


Fig. 4. 5. Transmisión mecánica del prototipo.

#### 4.1.1.2 Análisis de movilidad

La movilidad del prototipo desde una posición inicial de 0° hasta una posición final de 43° se da gracias a un mecanismo de trasmisión de movimiento de tres barras conocido como “triángulo articulado con un lado de longitud variable”. En la figura 4.6 se observa los vértices T<sub>1</sub>, T<sub>2</sub> y T<sub>3</sub> que forman el triángulo, y se evidencia que el movimiento se genera cuando la barra T<sub>1</sub>T<sub>3</sub> varía en longitud en la medida que se quiera inclinar la plataforma, mientras las barras T<sub>1</sub>T<sub>2</sub> y T<sub>2</sub>T<sub>3</sub> permanecen constantes.

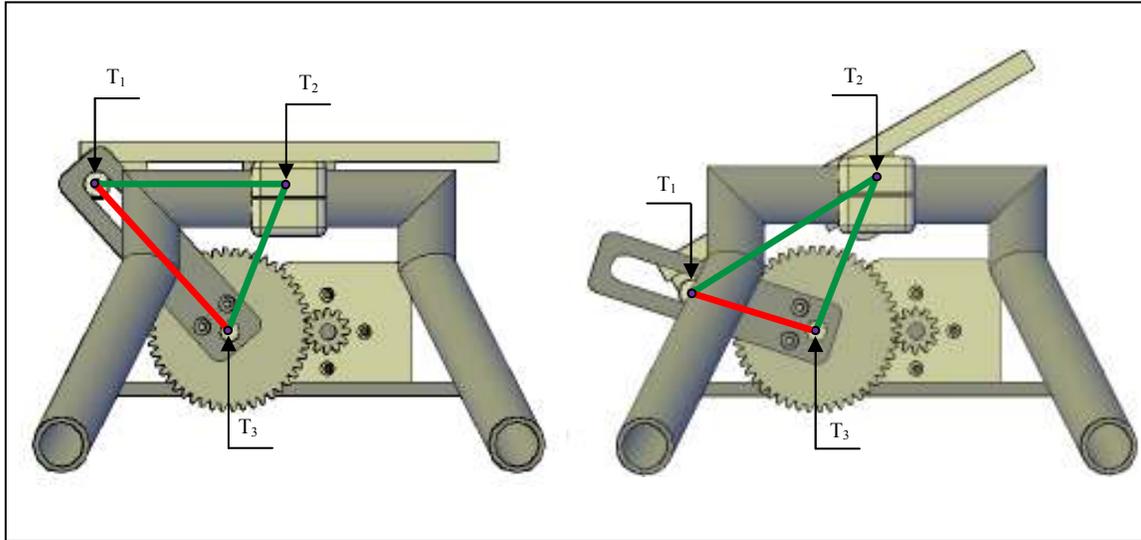


Fig. 4. 6. Mecanismo de transmisión de movimiento.

Una vez identificado el mecanismo de transmisión de movimiento el siguiente paso es estudiar su cinemática, esto significa establecer la relación que existe entre las posiciones angulares de entrada y las posiciones angulares de salida. Para el prototipo desarrollado en este trabajo se entiende como entrada el movimiento rotacional del motor transmitido en el eje del engranaje más grande (vértice  $T_3$  de la figura 4.6), y como salida la posición angular  $\theta$  de la plataforma de inclinación. En la figura 4.7 (a) y (b) se presentan todos los elementos que intervienen en el análisis cinemático.

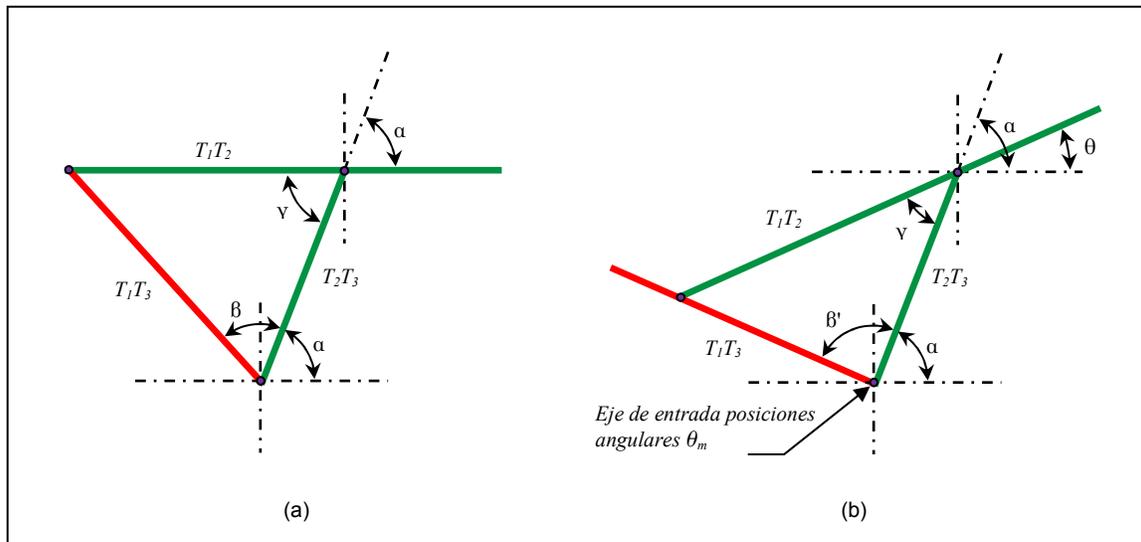


Fig. 4. 7. Diagrama cinemático del mecanismo. (a) Posición inicial  $0^\circ$ . (b) Posición  $\theta$ .

Cuando la plataforma se inclina  $\theta$  grados debido a una rotación en el vértice  $T_3$ , el triángulo  $T_1T_2T_3$  sufre una serie de modificaciones aparte de la longitud de la barra  $T_1T_3$  (ver figura 4.7). Estas modificaciones se pueden expresar de la siguiente forma:

$$Y = \alpha - \theta \quad (35)$$

$$\beta' = \beta + \theta_m \quad (36)$$

Donde:

$Y$  : Ángulo variable cuyo máximo valor se presenta cuando  $\theta = 0^\circ$ .

$\alpha$  : Ángulo constante.

$\beta$  : Ángulo constante, definido cuando  $\theta = 0^\circ$ .

$\beta'$  : Ángulo variable cuyo mínimo valor se presenta cuando  $\theta = 0^\circ$  y máximo cuando  $\theta = 43^\circ$ .

$\theta_m$  : Ángulo en que se mueve la barra  $T_1T_3$  debido a la acción del motor.

Aplicando la ley de cosenos es posible establecer la longitud de la barra  $T_1T_3$  de la siguiente manera:

$$T_1T_3 = \sqrt{T_1T_2^2 + T_2T_3^2 - 2(T_1T_2)(T_2T_3)\cos(Y)}$$

Sustituyendo la ecuación 35 se obtiene:

$$T_1T_3 = \sqrt{T_1T_2^2 + T_2T_3^2 - 2(T_1T_2)(T_2T_3)\cos(\alpha - \theta)} \quad (37)$$

Por otro lado aplicando la ley de senos se obtiene:

$$\frac{\sin Y}{T_1T_3} = \frac{\sin \beta'}{T_1T_2}$$

$$\sin \beta' = \frac{T_1T_2 \sin Y}{T_1T_3} \quad (38)$$

Sustituyendo las ecuaciones 35, 36 y 37 se obtiene:

$$\sin(\beta + \theta_m) = \frac{T_1 T_2 \sin(\alpha - \theta)}{\sqrt{T_1 T_2^2 + T_2 T_3^2 - 2(T_1 T_2)(T_2 T_3)\cos(\alpha - \theta)}}$$

$$\beta + \theta_m = \sin^{-1} \frac{T_1 T_2 \sin(\alpha - \theta)}{\sqrt{T_1 T_2^2 + T_2 T_3^2 - 2(T_1 T_2)(T_2 T_3)\cos(\alpha - \theta)}}$$

$$\theta_m = \sin^{-1} \frac{T_1 T_2 \sin(\alpha - \theta)}{\sqrt{T_1 T_2^2 + T_2 T_3^2 - 2(T_1 T_2)(T_2 T_3)\cos(\alpha - \theta)}} - \beta \quad (39)$$

La ecuación 39 es el resultado del estudio cinemático del mecanismo. La ecuación se utiliza para conocer cuántos grados tiene que girar la barra  $T_1 T_3$  para obtener una posición angular deseada  $\theta$  de la plataforma de inclinación. Para conocer con precisión los desplazamientos angulares que tiene que dar el motor de corriente directa es necesario multiplicar la ecuación 39 por 400, cuatrocientos es la razón de reducción suministrada por la transmisión “TM” de la figura 4.5.

#### 4.1.1.3 Consideraciones importantes de diseño

- La plataforma de inclinación “PI” debe ser ligera pero suficientemente rígida para soportar la masa del HDL-64E, la cual es de 13.1541 kg.
- El eje de rotación “O” debe ser de acero inoxidable, pues como en toda articulación mecánica es necesario evitar el desgaste y la corrosión entre las partes en contacto.
- El rodamiento que conecta la plataforma de inclinación con la corredera “C” debe soportar las fuerzas que se aplican sobre él durante la tarea de posicionamiento angular del HDL-64E.
- El ángulo  $\beta'$  jamás debe ser tan grande como para posicionar la barra  $T_1 T_3$  sobre la horizontal o por debajo de ella (ver figura 4.7). De esta manera se elimina la posibilidad de existencia de posiciones singulares en el prototipo.
- La carrera de la corredera “C” debe ser calculada para imposibilitar mecánicamente posicionamientos angulares mayores a 43°.

#### 4.1.2 Construcción del prototipo

En esta sección se presentan todos los elementos necesarios para la obtención de un prototipo físico. En el diseño a detalle se toman en cuenta todas las consideraciones definidas en el diseño conceptual.

El prototipo está conformado por una serie de elementos comerciales y otros que necesitaron ser manufacturados.

Los elementos comerciales son los siguientes:

- Motor de corriente directa MAXON 118778.
- Motoreductor DUNKERMOTOREN PLG 42S (sistema de reducción "S<sub>1</sub>").
- Engrane Martin TS1612 20 (entrada del sistema de reducción "S<sub>2</sub>").
- Engrane Martin TS1648 20 (salida del sistema de reducción "S<sub>2</sub>").
- Rodamiento de contacto Fag 624Z (plataforma de inclinación-corredera).
- Barra de acero inoxidable (eje de rotación "O").
- Rodamientos de aguja IKO ("R")
- Tornillos y tuercas.

Los elementos a manufacturar son:

- Estructura tubular ("ET").
- Plataforma de inclinación ("PI").
- Abrazaderas ajustables ("A").
- Soporte del motor y transmisión.
- Corredera "C".
- Chumaceras.

En la figura 4.8 se presenta el prototipo manufacturado.



Fig. 4. 8. Prototipo construido. (a) Posicionado en  $\theta = 0^\circ$  (b) Posicionado en  $\theta = 43^\circ$ .

## 4.2 Desarrollo del controlador PID del prototipo

El posicionamiento preciso del HDL-64E a través del prototipo diseñado es gracias a la implementación de un controlador PID. A continuación se presentan cada uno de los elementos necesarios para desarrollar el control automático del prototipo.

### 4.2.1 Motor y sensor de posición

El motor seleccionado para dar movilidad al sistema es un MAXON 118778 (ver figura 4.9). El torque máximo de este servomotor, dentro de un rango de operación recomendable, es de 0.113 Nm. Con el máximo brazo de palanca aplicado (longitud de la corredera "C" mostrada en la figura 4.5) el motor por sí solo no podría soportar los 13.1541 kg del HDL-64E, por esta razón, como se mencionó en la sección 4.1.1.1, se incorporó una transmisión mecánica que permite aumentar 400 veces el torque del motor.



Fig. 4. 9. Motor seleccionado como actuador del prototipo.

Con la transmisión mecánica de la figura 4.5 el torque del motor aumenta a 45.2 Nm, con ese torque según cálculos nominales es posible cargar una masa de hasta 46.12 kg, por lo que manipular los 13.1541 kg del sensor no constituye ninguna dificultad.

A diferencia de otros motores el MAXON 118778 no incluye encoder, por lo que se le adaptó uno de tipo incremental de la serie 102 de AMT. Dicho encoder está configurado para una resolución de 1,000 ppr (pulsos por revolución) por canal.

### 4.2.2 Tarjeta de control

La tarjeta de control que utiliza el prototipo es PIC-SERVO SC, la cual es un sistema completo de control para servomecanismos con las siguientes características:

- Incorpora un microcontrolador PIC18F2331 para el control de motores de corriente directa con encoder incremental que incluye perfiles trapezoidales, perfiles de velocidad y soporte para movimientos coordinados de múltiples ejes.
- Cuenta con un amplificador puente H LMD18200 capaz de manejar 3 A continuamente. Construido con protección térmica, de sobrecorriente y de mínima tensión.
- Protección para sobrecorriente y sobrevoltaje.
- Su interfaz RS485 permite que hasta 32 PIC-SERVO sean controladas desde un único puerto serial.
- Incluye dos entradas limit switch para protección de sobrecarrera.
- Es pequeña lo que permite montarla cerca de los motores reduciendo ruido y simplificando el cableado.
- Incluye interfaz de prueba y software de sintonización de ganancias PID.

En la figura 4.10 se presenta el diagrama eléctrico para la conexión de la tarjeta PIC-SERVO SC.

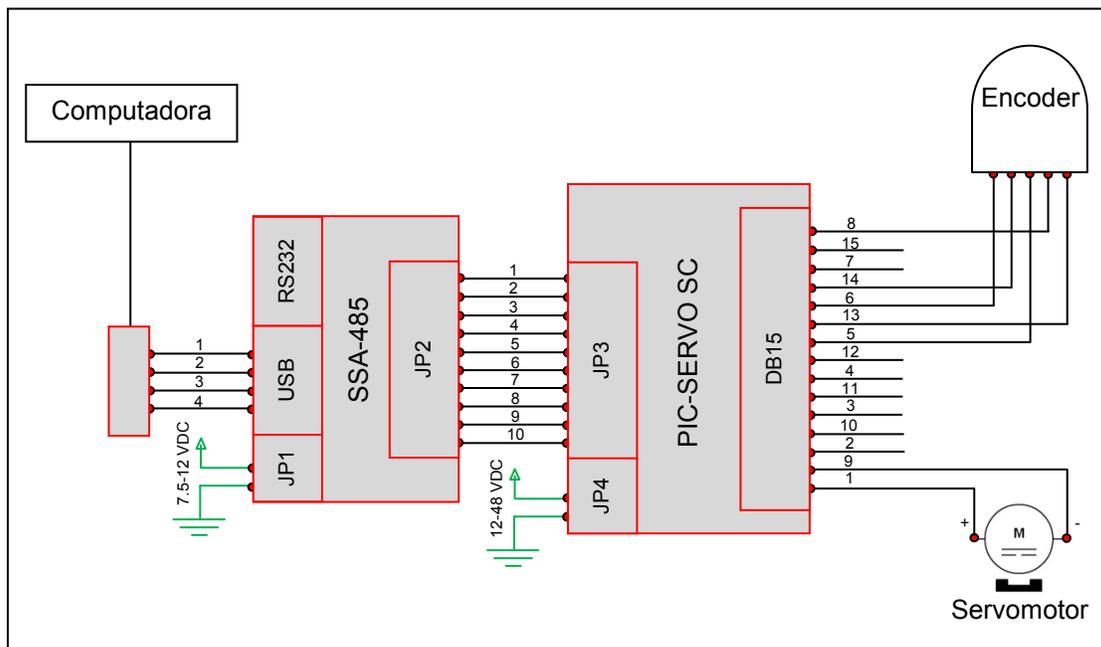


Fig. 4. 10. Diagrama eléctrico para la conexión de PIC-SERVO SC.

La comunicación entre la tarjeta PIC-SERVO SC y la computadora es a través del dispositivo SSA-485 (Smart Serial Adapter), el cual es un convertidor de USB/RS232 a RS485, lo que significa que este dispositivo puede convertir a RS485 una entrada USB o bien RS232. En la figura 4.10 se evidencia que el convertidor fue configurado para trabajar con entradas bajo el protocolo USB. En la tabla 4.1 se muestra la definición de cada uno de los pines para el puerto USB.

Tabla 4. 1. Configuración de pines para el puerto USB del convertidor SSA-485.

Pin	Definición
1	5 vdc
2	-Data
3	+Data
4	Tierra

El convertidor SSA-485 y la PIC-SERVO SC se comunican por medio del protocolo RS485 al conectar “JP2” con “JP3” (ver figura 4.10). La configuración de pines se presenta a continuación.

Tabla 4. 2. Configuración de pines para la comunicación RS485 entre el convertidor SSA-485 y la tarjeta PIC-SERVO SC.

Pin	Definición
1	PIC-SERVO RCV+
2	PIC-SERVO RCV-
3	PIC-SERVO XMT+
4	PIC-SERVO XMT-
5*	No utilizado
6	Tierra
7	Voltaje lógico (7.5 – 12VCD)
8	Tierra
9	Voltaje lógico (7.5 – 12VCD)
10	Tierra

\* Utilizado cuando se conecta más de 1 PIC SERVO SC.

La tarjeta de control PIC-SERVO SC se comunica con el actuador y el sensor de posición del prototipo a través de un conector DB15 (ver figura 4.10) cuya configuración se presenta en la tabla 4.3.

Tabla 4. 3. Configuración de pines para el conector DB15 de la tarjeta PIC-SERVO SC.

Pin	Definición
1	Salida a motor (M+)
2	Salida a motor (M+)
3	No utilizado
4	No utilizado
5	Encoder canal A
6	Encoder canal B
7	No utilizado
8	Encoder Índice
9	Salida a motor (M-)
10	Salida a motor (M-)
11	Tierra
12	Tierra
13	Tierra (encoder)
14	+5V (encoder)
15	Tierra

Durante todo el proceso de operación del controlador se debe alimentar el convertidor SSA-485 en JP1 con una fuente de poder lógica (ver figura 4.10), este voltaje debe estar en el orden de 7.5-12 VCD. Además se debe alimentar la tarjeta PIC-SERVO SC en JP4 con una fuente de poder para el motor, este voltaje puede estar en el orden de 12-48 VCD, sin embargo, se alimentó con 24 VDC pues es el voltaje nominal del motor MAXON 118778. Tanto la fuente de poder lógica como la fuente de poder para en motor son módulos independientes.

En lo que respecta al cableado el controlador únicamente necesita:

- Cable para la conexión USB entre la computadora y el convertidor SSA-485.
- Cable plano para 10 pines con socket IDC estándar en los dos extremos. Intercomunica el convertidor SSA-485 y la tarjeta PIC-SERVO SC.
- Cable con conector DB15 (Motor-encoder). Se conecta a la tarjeta PIC-SERVO SC para llevar la señal de control hacia el motor y captar la retroalimentación en posición angular del encoder.

Para mayor información acerca de las especificaciones y conexión del la tarjeta PIC-SERVO SC y el convertidor SSA-485 se puede consultar [30] y [31] respectivamente, los cuales pueden ser descargados de la página web del fabricante ([www.picservo.com](http://www.picservo.com)).

### 4.2.3 Resolución del actuador

La resolución de un mecanismo según [32] es el paso mecánico más pequeño que el mecanismo puede hacer durante el movimiento de punto a punto. En esta sección se estima el mínimo movimiento angular que se puede generar por el motor a la salida de la transmisión mecánica instalada “TM”, es decir en el punto  $T_3$  de la figura 4.6.

La resolución del actuador ( $R_{act}$ ) en el punto  $T_3$  es función de la resolución del sensor de posición ( $R_{enc}$ ) y de la de la relación de reducción ( $R_{ratio}$ ) suministrada por la transmisión “TM”.

De lo anterior se puede plantear:

$$R_{act} = R_{enc} R_{ratio} \quad (40)$$

El encoder, tal y como se planteo en la sección 2.2.1, está configurado para tener una resolución de 1,000 ppr por canal. Si dicho encoder posee dos canales con señales en cuadratura

desfasados  $90^\circ$ , entonces su resolución global es de 4,000 cpr (cuentas de encoder por revolución).

Por otro lado en la sección 4.1.1.1 se definió que la relación de reducción suministrada por la transmisión "TM" es de 1:400.

Sustituyendo se obtiene:

$$(4,000)(400) = 1,600,000$$

La resolución del actuador es de 1, 600,000 cpr, lo que equivale a 4,444.4444 cuentas de encoder por cada grado. Visto de otra forma cada cuenta de encoder equivale a  $0.000225^\circ$  en el actuador (punto  $T_3$  de la figura 4.6).

Esta capacidad del actuador para realizar movimientos angulares tan finos evidentemente redundante en la capacidad del prototipo para realizar posicionamientos angulares muy finos del HDL-64E.

#### 4.2.4 Comportamiento cinemático del controlador

En la sección 4.1.1.2 se realizó un estudio cinemático cuyo resultado fue la ecuación 39, la cual se define como la expresión matemática que relaciona las posiciones angulares de entrada (movimientos  $\theta_m$  del actuador transmitidos en el punto  $T_3$  de la figura 4.6) con las posiciones angulares de salida (posición angular  $\theta$  de la plataforma de inclinación).

El control en posición a través de la tarjeta PIC-SERVO SC se realiza por medio de cuentas de encoder, lo que quiere decir que para posicionar la plataforma de inclinación en un ángulo  $\theta$  es necesario conocer el equivalente en cuentas de encoder del ángulo  $\theta_m$ , que calculado a través de la ecuación 39, permita alcanzar dicho ángulo de inclinación.

Transformar cualquier ángulo  $\theta_m$  a cuentas de encoder es sumamente fácil, simplemente se utiliza la relación obtenida en la sección 4.2.3, la cual dice que cada cuenta de encoder equivale a  $0.000225^\circ$ . Sin embargo, calcular con precisión el ángulo  $\theta_m$  no es fácil, pues según la ecuación 39 este ángulo depende de algunas dimensiones (angulares y longitudinales) cuya medición implica incertidumbres que impactan en el ángulo  $\theta_m$ .

Frente a esta dificultad lo que se hizo fue encontrar experimentalmente las cuentas de encoder que permitieran alcanzar cada 1° los 43° de máximo posicionamiento de la plataforma de inclinación. El resultado de la aproximación experimental se presenta en la figura 4.11. El signo negativo de las cuentas de encoder es debido a que los movimientos en el punto T<sub>3</sub> de la figura 4.6 se realizan en sentido contrario a las manecillas del reloj.

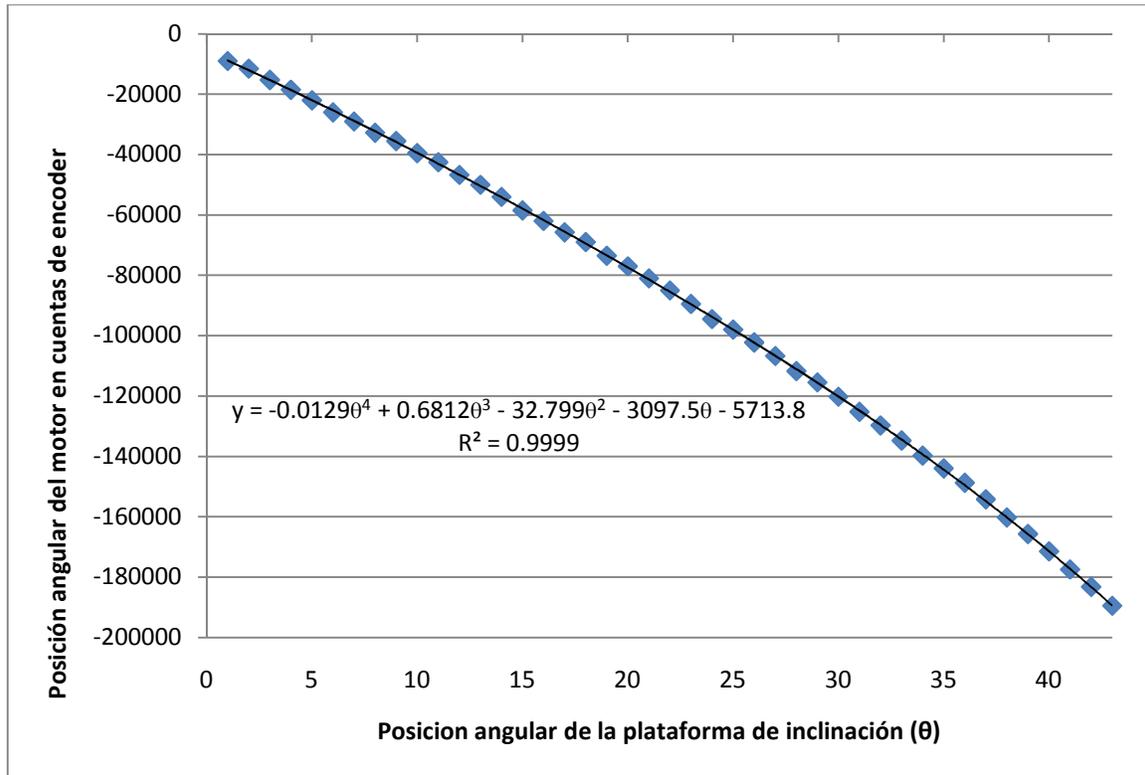


Fig. 4. 11. Relación entre la posición del motor y la posición de la plataforma de inclinación.

En la figura 4.11 se puede evidenciar que la relación entre la posición angular del motor en cuentas de encoder y la posición angular de la plataforma de inclinación, dentro del rango de 0 a 43°, sigue una tendencia que se ajusta casi perfectamente a un polinomio de grado 4. Este polinomio es el que rige el comportamiento cinemático del controlador.

En la tabla 4.4 se presenta: en la primera columna; la posición angular de la plataforma, en la segunda; la posición del motor en cuentas de encoder que resultó de la aproximación experimental presentada en la figura 4.11, en la tercera; la posición de motor en cuentas de encoder al utilizar el polinomio de grado 4 para controlar el sistema, en la cuarta; el error absoluto en cuentas de encoder que se genera al utilizar el polinomio, y en la última columna; el error absoluto expresado en grados. El error angular promedio al utilizar el polinomio de grado 4 como la ecuación que describe la cinemática del prototipo está en el orden de 0.062360138°.

Tabla 4. 4. Cálculo de una aproximación al error promedio del sistema de control.

Posición angular de la plataforma ( $\theta$ )	Posición angular del motor (Cnt. de Enc.)	Posición angular del motor estimada (Cnt. de Enc.) <sup>1</sup>	Error (Cnt. de Enc.) <sup>2</sup>	Error (grados) <sup>3</sup>
1	-9000	-8843.4307	156.5693	0.035228092
2	-11500	-12034.7528	534.7528	0.12031938
3	-15250	-15284.1435	34.1435	0.007682287
4	-18500	-18588.2896	88.2896	0.01986516
5	-22000	-21944.1875	55.8125	0.012557813
6	-26000	-25349.1432	650.8568	0.14644278
7	-29000	-28800.7723	199.2277	0.044826232
8	-32750	-32297	453	0.101925
9	-35500	-35836.0611	336.0611	0.075613747
10	-39500	-39416.5	83.5	0.0187875
11	-42500	-43037.1707	537.1707	0.120863408
12	-46750	-46697.2368	52.7632	0.01187172
13	-50000	-50396.1715	396.1715	0.089138588
14	-54000	-54133.7576	133.7576	0.03009546
15	-58500	-57910.0875	589.9125	0.132730313
16	-62000	-61725.5632	274.4368	0.06174828
17	-65750	-65580.8963	169.1037	0.038048333
18	-69000	-69477.108	477.108	0.1073493
19	-73500	-73415.5291	84.4709	0.019005953
20	-77000	-77397.8	397.8	0.089505
21	-81000	-81425.8707	425.8707	0.095820907
22	-85000	-85502.0008	502.0008	0.11295018
23	-89500	-89628.7595	128.7595	0.028970888
24	-94500	-93809.0256	690.9744	0.15546924
25	-98000	-98045.9875	45.9875	0.010347188
26	-102250	-102343.1432	93.1432	0.02095722
27	-106750	-106704.3003	45.6997	0.010282432
28	-111750	-111133.576	616.424	0.1386954
29	-115500	-115635.3971	135.3971	0.030464348
30	-120250	-120214.5	35.5	0.0079875
31	-125250	-124875.9307	374.0693	0.084165593
32	-129750	-129625.0448	124.9552	0.02811492
33	-134750	-134467.5075	282.4925	0.063560813
34	-139750	-139409.2936	340.7064	0.07665894
35	-144000	-144456.6875	456.6875	0.102754687
36	-148750	-149616.2832	866.2832	0.19491372
37	-154250	-154894.9843	644.9843	0.145121467
38	-160250	-160300.004	50.004	0.0112509
39	-165750	-165838.8651	88.8651	0.019994647
40	-171500	-171519.4	19.4	0.004365
41	-177500	-177349.7507	150.2493	0.033806093
42	-183250	-183338.3688	88.3688	0.01988298
43	-189500	-189494.0155	5.9845	0.001346513
<b>Error promedio</b>				<b>0.062360138</b>

<sup>1</sup> Posición angular del motor estimada =  $-0.0129\theta^4 + 0.6812\theta^3 - 32.799\theta^2 - 3097.5\theta - 5713.8$ .

<sup>2</sup> Error en cuentas de encoder = ABS (Posición angular del motor – Posición angular del motor estimada).

<sup>3</sup> Error en grados = (Error en cuentas de encoder)( 0.000225).

#### 4.2.5 Diseño de la interfaz de usuario

En la sección 4.2.2 se mencionó que la tarjeta PIC-SERVO SC incluye una interfaz de usuario para pruebas, sin embargo, esta interfaz es poco amigable y evidentemente no está ajustada a la aplicación objeto de estudio en este trabajo. Por esta razón se decidió desarrollar una interfaz propia en el lenguaje de Visual Basic aprovechando la disponibilidad de las librerías de PIC-SERVO en este lenguaje de programación. Esta interfaz llamada “Interfaz de Control LIDAR” es exclusiva para la tarea de aumentar el ángulo de visión vertical del HDL-64E y su formulario gráfico se presenta en la figura 4.12.



Fig. 4. 12. Interfaz de usuario desarrollada en Visual Basic.

Esta interfaz consta de tres secciones y cada una de ellas se describe a continuación.

La primera sección se titula “Comunicación” e incluye la posibilidad de seleccionar el puerto de comunicación y la velocidad de transmisión de datos tal y como se muestra en la figura 4.13.

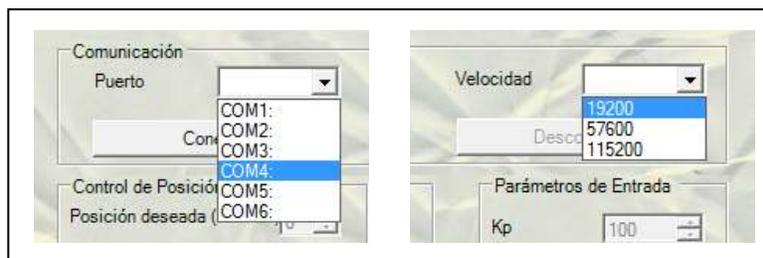


Fig. 4. 13. Selección de parámetros de comunicación.

Esta sección incluye además dos botones; uno llamado “conectar” y otro llamado “desconectar”. El botón “conectar” habilita el uso de la interfaz siempre y cuando se haya seleccionado correctamente el puerto de comunicación, la velocidad de transmisión de datos y se encuentre conectada correctamente la tarjeta PIC-SERVO SC tal y como se muestra en la figura 4.10, en caso contrario aparecerán mensajes de error como los presentados en la figura 4.14.

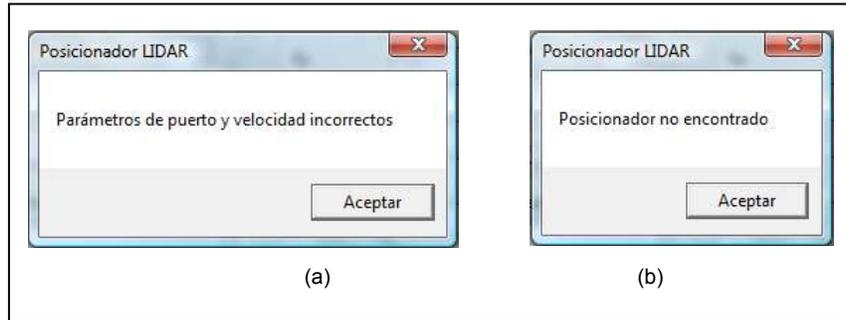


Fig. 4. 14. Mensajes de error de la interfaz. (a) Al seleccionar incorrectamente el puerto de comunicación y/o velocidad. (b) Al haber un problema de conexión de hardware.

El siguiente botón es “desconectar” y sirve para cerrar la sesión de la “Interfaz de Control LIDAR”.

La segunda sección se titula “Parámetros de entrada” e incorpora la posibilidad de modificar las ganancias de control (proporcional, derivativa e integral) así como la oportunidad de seleccionar la velocidad y aceleración del desplazamiento de la plataforma de inclinación. Cuando se inicia sesión esta sección aparece con los valores por defecto que se muestran en la figura 4.15, dichos valores se fijaron como aceptables para la aplicación de posicionamiento angular del HDL-64E con ayuda del software de sintonización “Stepresponse” que incluye la tarjeta PIC-SERVO SC.

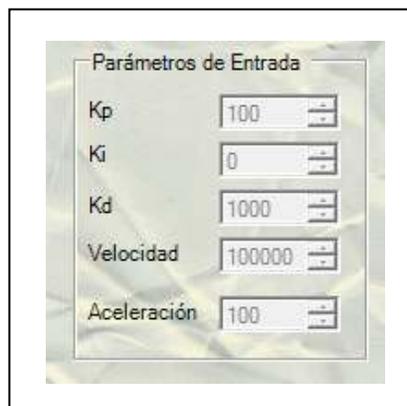


Fig. 4. 15. Sección “Parámetros de entrada”.

La tercera sección se denomina “Control de posición”. En esta sección se selecciona el ángulo en grados al que se quiere posicionar el LIDAR HDL-64E. La interfaz no permite seleccionar un ángulo superior a  $43^\circ$  ni inferior a  $0^\circ$ .

Una vez que se selecciona el ángulo deseado, éste es enviado como “*Set Point*” en cuentas de encoder hacia la tarjeta de control al pulsar el botón “ir a posición”. Resulta importante mencionar que para traducir la posición angular deseada de grados a cuentas de encoder se utiliza el polinomio de grado 4 que se estimó en la sección 4.2.4, como se señaló con anteriormente este polinomio rige la cinemática del prototipo.

Esta tercera sección contiene además dos indicadores de barra. Estos indicadores de barra tienen el propósito de mostrar la manera en que se alcanza el “*Set Point*” en tiempo real. El código fuente en Visual Basic se presentan en el anexo 1.

Con la interfaz desarrollada es posible fijar el HDL-64E en la posición angular que el usuario considere necesaria, para poder así, escanear estructuras que usualmente no podrían ser alcanzadas por el sensor en condiciones normales. En la figura 4.16 se presenta cómo a través de una acción de control es posible inclinar el sensor hasta una posición máxima de  $43^\circ$ .

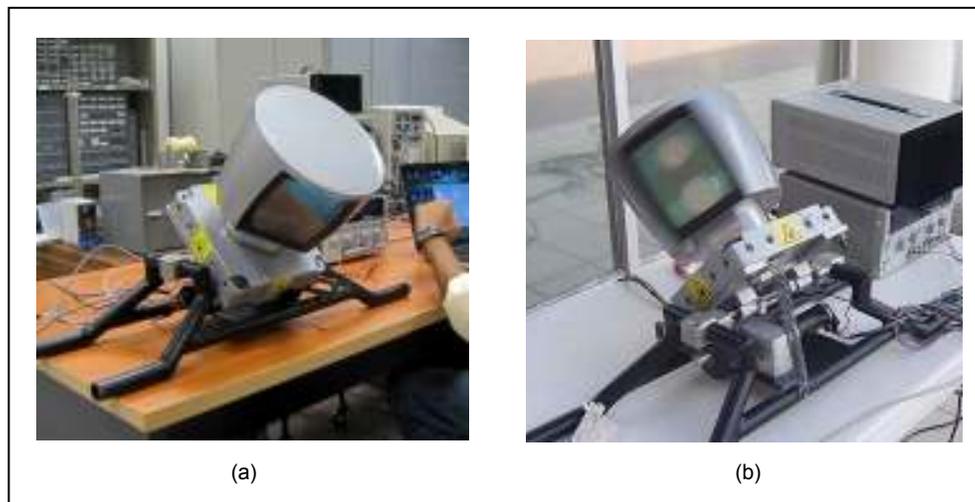


Fig. 4. 16. Máximo posicionamiento angular del HDL-64E. (a) Sin operación del sensor. (b) Bajo operación del sensor.

A partir de esta sección al resultado de la incorporación del controlador en posición angular al prototipo mecánico descrito en toda la sección 4.1 se le llamará “sistema de posicionamiento HDL-64E”.

Para poner a prueba la funcionalidad del sistema de posicionamiento HDL-64E se escaneó la parte frontal del edificio del Centro de Investigación en Ciencia Aplicada y Tecnología Avanzada (CICATA-Qro). Es importante mencionar que para la adquisición y visualización de los paquetes de datos arrojados por el sensor en este trabajo se utiliza un software desarrollado en el lenguaje C++ por los autores de [19]. Los resultados de la prueba de posicionamiento y escaneo se presentan a continuación.

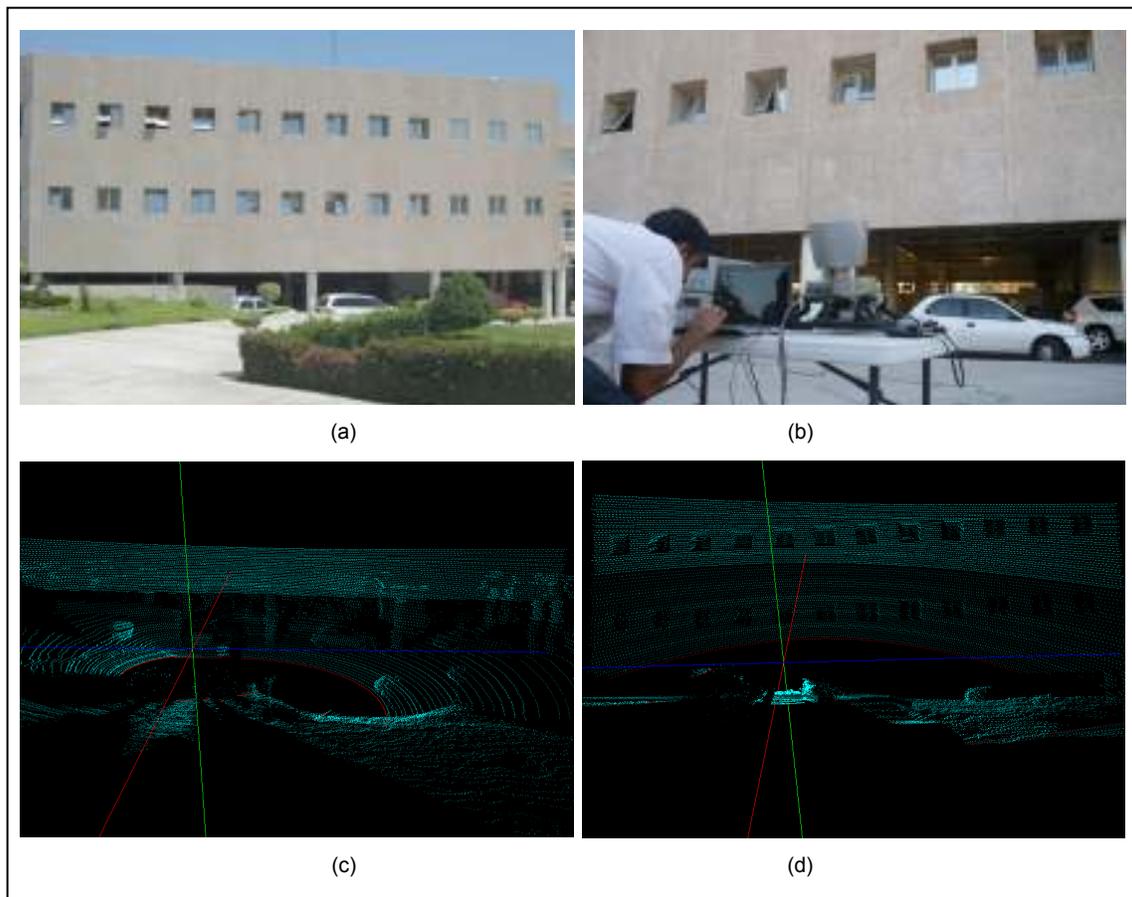


Fig. 4. 17. Prueba del sistema de posicionamiento HDL-64E. (a) Edificio escaneado. (b) Sensor montado sobre el sistema de posicionamiento. (c) Barrido con el HDL-64E a 5° de inclinación. (d) Barrido a 30° de inclinación.

En la figura 4.7 (a) se muestra el edificio escaneado, este edificio tiene aproximadamente 14 metros de altura. Bajo condiciones normales de uso del sensor sería imposible obtener información completa del inmueble, sin embargo, a través del sistema de posicionamiento HDL-64E es posible por medio de dos barridos, uno a 5° y otro a 30° (figura 4.7 (c) y (d)), obtener una nube de puntos completa del edificio.

### 4.3 Transformación de las nubes de puntos

El hecho de inclinar el HDL-64E para aumentar su campo de visión no significa que las adquisiciones del sensor sean ya útiles para realizar tareas de reconstrucción 3D. Esto se debe a que todas las adquisiciones que se hagan con el sensor inclinado experimentarían una rotación y una traslación con respecto a la referencia horizontal del sensor (condición normal de uso sin inclinación). La figura 4.17 (d) representa el escaneo de un edificio a 30° de inclinación, y aunque no sea muy notorio esta captura ha sufrido una drástica transformación en cuanto a orientación y posición (ver figura 4.18).

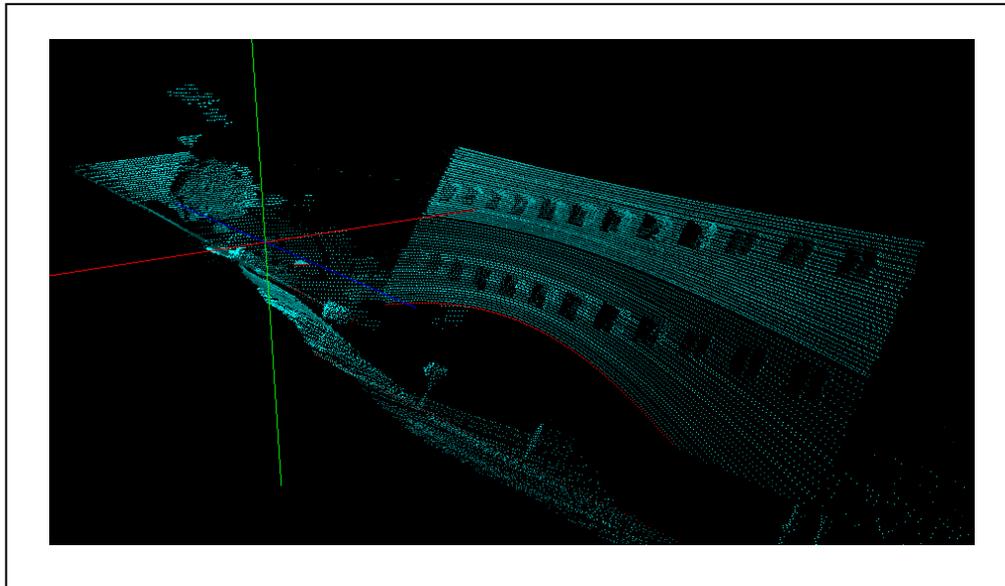


Fig. 4. 18. Vista lateral de escaneo con el HDL-64E posicionado a 30° de inclinación.

Para considerar como útiles las nubes de puntos adquiridas con el escáner inclinado es necesario referenciar dichas nubes de puntos con la referencia horizontal del sensor, es decir, considerar el posicionamiento horizontal del sensor como un sistema de referencia global y expresar las nubes de puntos adquiridas al inclinar  $\theta$  grados el sensor bajo dicho sistema de referencia.

Existen muchos enfoques para resolver la problemática de transformar nubes de puntos, por ejemplo en [33] se utiliza el algoritmo "Iterative Closest Point" para encontrar la matriz de rotación y traslación que aplicada sobre una nube de puntos lleve al alineamiento con otra.

En el presente trabajo para transformar las nubes de puntos y llevarlas a la referencia horizontal del sensor y poder así corregir el efecto de transformación generado al inclinar el HDL-64E, se plantea una metodología que se basa en la estimación de las matrices homogéneas que rigen dichas transformaciones.

#### 4.3.1 Base de cálculo de la metodología

Según la ecuación 31, dos vectores pertenecientes a diferentes sistemas de referencia están relacionados por una matriz de transformación homogénea. Esta expresión se puede hacer extensiva a la relación de nubes de puntos tal y como se muestra en la ecuación 41.

$$N_{\theta} = H_{\theta}N_0 \quad (41)$$

Donde:

$N_{\theta}$  : Nube de puntos adquirida al inclinar  $\theta$  grados el escáner laser.

$$N_{\theta} = \{P_{\theta}^1, P_{\theta}^2, \dots, P_{\theta}^n\}; \text{ con } P_{\theta}^i = [x_{\theta}^i \quad y_{\theta}^i \quad z_{\theta}^i \quad 1]^T$$

$N_0$  : Nube de puntos adquirida con la referencia horizontal del escáner láser.

$$N_0 = \{P_0^1, P_0^2, \dots, P_0^n\}; \text{ con } P_0^i = [x_0^i \quad y_0^i \quad z_0^i \quad 1]^T$$

La matriz homogénea  $H_{\theta}$  de la ecuación 41 rige la transformación experimentada por una nube de puntos al inclinar  $\theta$  grados el escáner, por lo que calcularla permitiría utilizarla para transformar cualquier nube de puntos de cualquier escenario y llevarla a la referencia horizontal del sensor siempre y cuando ésta haya sido capturada con los mismos  $\theta$  grados de inclinación.

En la ecuación 32 se plantea que la matriz homogénea que relaciona dos vectores pertenecientes a diferentes sistemas de referencia está integrada por cuatro componentes; una matriz de rotación ( $R_{3x3}$ ), un vector de traslación ( $T_{3x1}$ ), un vector de perspectiva ( $F_{1x3}$ ), y un factor de escalamiento ( $E_{1x1}$ ). Para la aplicación objeto de estudio en este trabajo el vector de perspectiva y el factor de escalamiento se consideran como nulo y unitario respectivamente. Considerando lo anterior, un punto perteneciente a  $N_{\theta}$  y otro punto perteneciente a  $N_0$  están relacionados de la manera siguiente:

$$P_{\theta}^i = \begin{bmatrix} R_{3x3} & T_{3x1} \\ 0 & 1 \end{bmatrix} P_0^i \quad (42)$$

La ecuación 42 se puede expresar representando a la matriz de transformación homogénea como un conjunto de 12 variables, tal y como se muestra a continuación:

$$P_{\theta}^i = \begin{bmatrix} a & b & c & d \\ e & f & g & h \\ j & k & l & m \\ 0 & 0 & 0 & 1 \end{bmatrix} P_0^i \quad (43)$$

En la ecuación 42 las variables  $a, b, c, e, f, g, j, k$  y  $l$  representan la rotación entre las nubes de puntos  $N_{\theta}$  y  $N_0$  (mostradas en el recuadro rojo), mientras que las variables  $d, h$  y  $m$  representan la traslación (mostradas en el recuadro verde).

Considerando que las coordenadas rectangulares de los puntos expresados en la ecuación 43 son datos conocidos (provistos por el LIDAR HDL-64E), dicha ecuación se puede definir como un sistema de ecuaciones de la siguiente manera:

$$x_{\theta}^i = ax_0^i + by_0^i + cz_0^i + d \quad (44)$$

$$y_{\theta}^i = ex_0^i + fy_0^i + gz_0^i + h \quad (45)$$

$$z_{\theta}^i = jx_0^i + ky_0^i + lz_0^i + m \quad (46)$$

El sistema formado por las ecuaciones 44, 45 y 46 contiene las doce incógnitas que integran la matriz homogénea que representa la relación entre las nubes de puntos  $N_{\theta}$  y  $N_0$ . Sin embargo, debido a la disparidad entre el número de ecuaciones e incógnitas es imposible solucionar el sistema. Este problema de disparidad se resuelve fácilmente si en lugar de considerar un punto en cada nube de puntos se consideran cuatro (puntos correlacionados explicados en la sección 4.3.2), de manera que se forme un sistema de doce ecuaciones y doce incógnitas.

Una vez planteado y solucionado el sistema de ecuaciones se obtiene una primera aproximación para las doce variables que representan la matriz transformación homogénea  $H_{\theta}$ . Sin embargo, dicha aproximación podría ser incorrecta pues puede limitarse únicamente a satisfacer el sistema de doce ecuaciones y no necesariamente representar una matriz de transformación homogénea. Para condicionar a que los resultados obligatoriamente representen una matriz de transformación homogénea, al sistema de doce ecuaciones se agrega la ecuación 47, la cual es la representación matemática de una de las propiedades más conocidas de las matrices homogéneas, esta propiedad es la que establece que el determinante de una matriz de transformación homogénea es igual a 1.

$$afl + bgj + cek - bel - agk - cjj = 1 \quad (47)$$

Una vez que se soluciona el sistema de trece ecuaciones, el cual se define como un sistema no lineal, se obtiene una excelente aproximación para la matriz de transformación  $H_\theta$ , la cual, como se ha mencionado anteriormente, representa la relación entre las nubes de puntos  $N_\theta$  y  $N_0$ , o si se quiere ver de otra manera, define la transformación 3D que experimenta una nube de puntos al inclinar  $\theta$  grados el HDL-64E.

Si esta matriz de transformación homogénea  $H_\theta$  representa la relación entre una nube de puntos  $N_\theta$  de algún escenario adquirida al inclinar  $\theta$  grados el HDL-64E y otra nube de puntos  $N_0$  obtenida del mismo escenario con el sensor bajo una referencia horizontal, la inversa de dicha matriz podría utilizarse para llevar a la referencia horizontal cualquier nube de puntos de cualquier escenario adquirida al inclinar  $\theta$  grados el HDL-64E. La ecuación 48 resume lo planteado anteriormente:

$$N_\theta^0 = H_\theta^{-1} N_\theta \quad (48)$$

Donde:

$N_\theta^0$ : Representación de la nube de puntos  $N_\theta$  bajo el sistema de referencia horizontal del sensor.

El resultado de aplicar la ecuación 48 es la nube de puntos  $N_\theta^0$ , la cual es una nube de puntos lista para realizar tareas de reconstrucción 3D de espacios urbanos pues su sistema de referencia corresponde con la referencia horizontal del sensor.

#### 4.3.2 Selección de los puntos correlacionados

La precisión en el cálculo de las matrices homogéneas, y en consecuencia la precisión del método propuesto de transformación de nubes de puntos, dependen en gran medida de la adecuada selección de los cuatro puntos pertenecientes a  $N_\theta$  y sus respectivos puntos correlacionados pertenecientes a  $N_0$ .

Una nube de puntos  $N_\theta$  y otra nube de puntos  $N_0$  son útiles para la selección de sus respectivos puntos correlacionados si comparten suficiente información para la extracción de las coordenadas cartesianas de dichos puntos. Por ejemplo en la figura 4.19 se muestra un escenario que fue escaneado a 0 (referencia horizontal del sensor) y a 5° de inclinación. En esta

figura se evidencia que ambas nubes de puntos comparten suficiente información para la selección de los cuatro pares de puntos correlacionados.

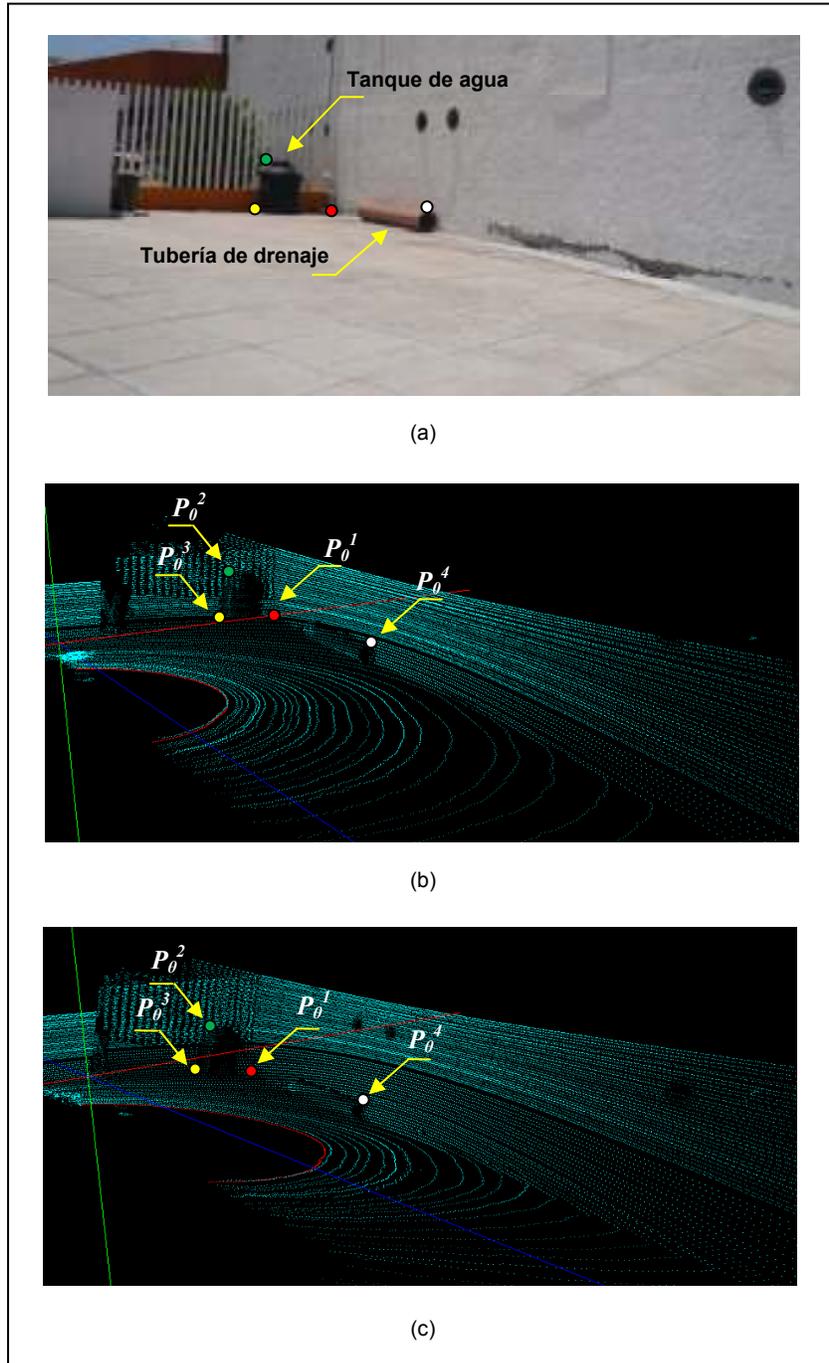


Fig. 4. 19. Nubes de puntos aptas para la extracción de sus puntos correlacionados. (a) Escenario escaneado para el cálculo de la matriz de transformación homogénea  $H_5$ . (b) Nube de puntos  $N_0$ . (c) Nube de puntos  $N_5$ .

Es importante mencionar que dentro de la información que compartan las nubes de puntos  $N_\theta$  y  $N_0$  es necesario que existan características geométricas o bien elementos fáciles de distinguir y de correlacionar manualmente. Por ejemplo en la figura 4.19 (a) se evidencia la presencia de dos elementos (un tanque de agua y un segmento de tubería de drenaje) que son completamente distinguibles en la nube de puntos  $N_5$  y en la nube de puntos  $N_0$  (ver figura 4.19 (b) y (c)).

Tomando en cuenta las consideraciones anteriores, para el caso particular de las nubes de puntos de la figura 4.19, los puntos seleccionados como correlacionados se presentan con los colores rojo ( $P_0^1$  y  $P_\theta^1$ ), verde ( $P_0^2$  y  $P_\theta^2$ ), amarillo ( $P_0^3$  y  $P_\theta^3$ ) y blanco ( $P_0^4$  y  $P_\theta^4$ ). Estos puntos sirven como base para calcular la matriz de transformación homogénea que permite, a través de la aplicación de la ecuación 48, llevar a la referencia horizontal cualquier nube de puntos de cualquier escenario adquirida con el HDL-64E posicionado a  $5^\circ$ . Para conocer las coordenadas de estos puntos se aplica un procedimiento que consta de los siguientes pasos:

1. Graficar en Matlab las nubes de puntos: las nubes de puntos, en primera instancia  $N_0$  y posteriormente  $N_5$ , son graficadas en Matlab puesto que este software, a diferencia del desarrollado en [19] para adquisición y visualización, permite manipular gráficos en 2 y 3 dimensiones y extraer fácilmente de éstos las coordenadas de cualquier punto. El código para graficar una nube de puntos 3D en Matlab se presenta en anexo 2. En la figura 4.20 se muestran las nubes de puntos  $N_0$  y  $N_5$  graficadas en Matlab, en estas vistas se aprecian todos los detalles presentes en el escenario escaneado, el cual corresponde a la figura 4.19 (a).

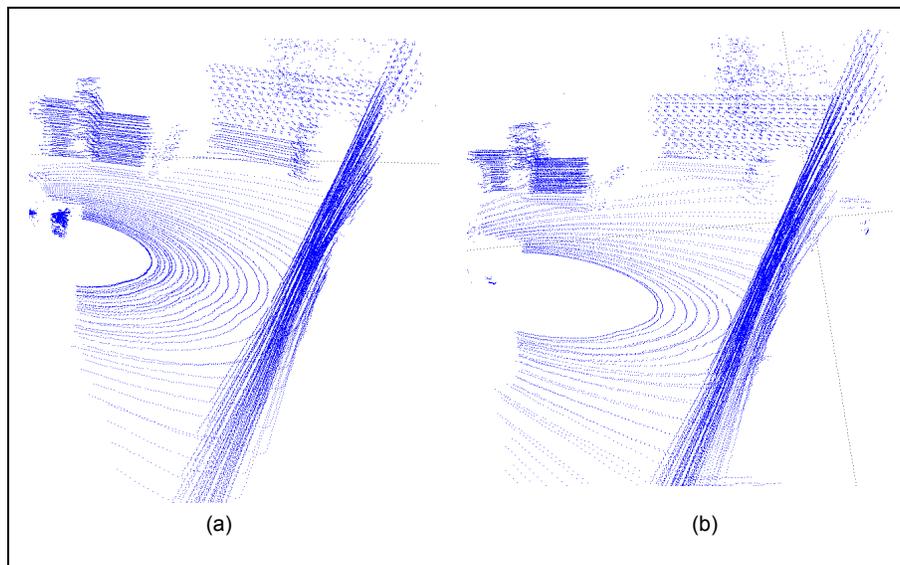


Fig. 4. 20. Nubes de puntos graficadas en Matlab para la extracción de puntos. (a)  $N_0$ . (b)  $N_5$ .

2. Seleccionar manualmente una primera aproximación para las coordenadas de los puntos: con las nubes de puntos graficadas en Matlab es posible extraer manualmente una aproximación rápida para las coordenadas de cada uno de los puntos pertenecientes a  $N_0$  y sus respectivos puntos correlacionados pertenecientes a  $N_\theta$ .

A título de ejemplo, la figura 4.21 muestra la primera aproximación para el punto  $P_0^1$  y su respectivo punto correlacionado  $P_\theta^1$ , dichas aproximaciones corresponden a las coordenadas (-860, -160.6, 719.4) y (-862.5, -275.2, 679.4). Recuérdese que estos puntos están representados en color rojo en la figura 4.19.

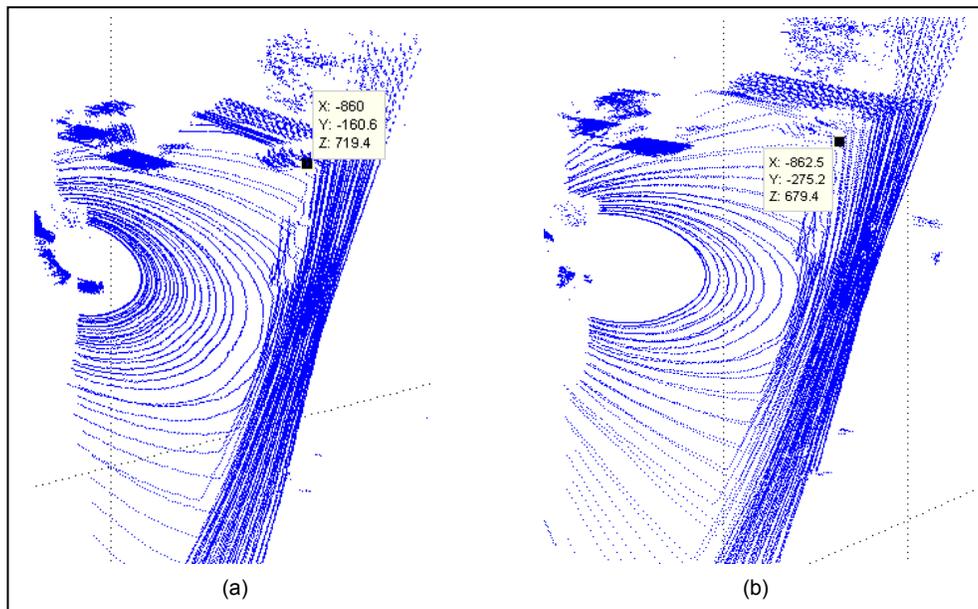


Fig. 4. 21. Extracción de primera aproximación para los puntos. (a) Punto  $P_0^1$ . (b) Punto  $P_\theta^1$ .

3. Eliminar todos los puntos que no pertenecen a la vecindad de cada aproximación: este paso permite concentrar la búsqueda de cada uno de los cuatro pares de puntos correlacionados tomando en cuenta únicamente sus vecinos. La vecindad se calcula a partir de la primera aproximación de cada punto encontrada en el paso 2, y se extiende sobre los ejes de coordenadas X, Y y Z según un tamaño de ventana ajustable “v”. El código en Matlab para visualizar la vecindad de cada uno de los puntos se presenta en anexo 2.

4. Buscar en la vecindad de cada punto una mejor aproximación: al delimitar la búsqueda de los cuatro pares de puntos correlacionados a sus vecinos más cercanos se puede seleccionar una mejor aproximación, puesto que, aparte de que las nubes de puntos se vuelven más ligeras y

fáciles de manipular, se eliminan oclusiones y algunas fuentes de ruido que interferirían con una búsqueda exitosa. Anteriormente se había mencionado que la primera aproximación para los puntos  $P_0^1$  y  $P_\theta^1$  eran las coordenadas (-860, -160.6, 719.4) y (-862.5, -275.2, 679.4) respectivamente, sin embargo, al aplicar este procedimiento se encuentra aproximaciones más precisas, las cuales corresponden a (-897.9, -157.6, 722) y a (-850.8, -254.4, 764.5), este resultado se puede apreciar en la figura 4.22.

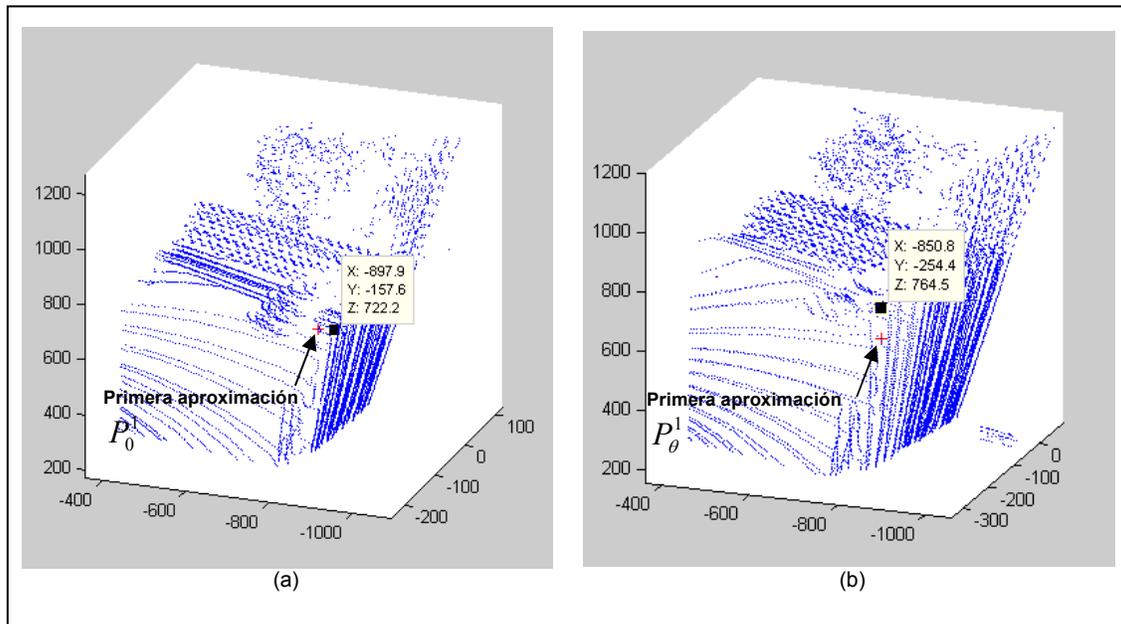


Fig. 4. 22. Eliminación de vecindades para la mejor extracción de puntos. (a) Punto  $P_0^1$  . (b) Punto  $P_\theta^1$  .

Una vez que se aplica completamente el procedimiento de extracción de puntos planteado anteriormente se obtienen las coordenadas para los cuatro pares de puntos correlacionados que se muestran en la figura 4.19. Las coordenadas de dichos puntos se presentan a continuación:

$$\begin{aligned}
 P_0^1 &= \begin{bmatrix} -897.9 \\ -157.6 \\ 722.2 \\ 1 \end{bmatrix} & P_0^2 &= \begin{bmatrix} -749 \\ -16.65 \\ 694 \\ 1 \end{bmatrix} & P_0^3 &= \begin{bmatrix} -673.5 \\ -153.8 \\ 707.5 \\ 1 \end{bmatrix} & P_0^4 &= \begin{bmatrix} -738.2 \\ -157.9 \\ 71.29 \\ 1 \end{bmatrix} \\
 P_\theta^1 &= \begin{bmatrix} -850.8 \\ -254.4 \\ 764.5 \\ 1 \end{bmatrix} & P_\theta^2 &= \begin{bmatrix} -729.2 \\ -99.92 \\ 689.7 \\ 1 \end{bmatrix} & P_\theta^3 &= \begin{bmatrix} -641.9 \\ -232.1 \\ 716.5 \\ 1 \end{bmatrix} & P_\theta^4 &= \begin{bmatrix} -718.2 \\ -236.5 \\ 74.2 \\ 1 \end{bmatrix}
 \end{aligned}$$

El procedimiento de extracción de puntos que se plantea, y que se ejemplificó con las nubes de puntos de las figuras 4.19 (b) y 4.19 (c), es aplicable para cualquier par de nubes de puntos a las que se les quiera estimar la matriz de transformación homogénea que las relaciona, siempre y cuando se cumpla la condición planteada de que ambas nubes compartan suficiente información para la fácil extracción de las coordenadas cartesianas de sus puntos correlacionados.

### 4.3.3 Cálculo de las matrices de transformación homogéneas

En la sección 4.3.2 se ejemplificó el procedimiento de extracción de puntos correlacionados tomando en cuenta una nube de puntos con la referencia horizontal del sensor y otra nube de puntos con el HDL-64E posicionado a 5°. Con el resultado de dicho procedimiento de extracción se puede formar el sistema de trece ecuaciones y doce incógnitas planteado en la sección 4.3.1, el sistema de ecuaciones se muestra a continuación:

$$\begin{aligned}
 -850.8 &= -897.9a - 157.6b + 722.2c + d \\
 -254.4 &= -897.9e - 157.6f + 722.2g + h \\
 764.5 &= -897.9j - 157.6k + 722.2l + m \\
 -729.2 &= -749a - 16.65b + 694c + d \\
 -99.92 &= -749e - 16.65f + 694g + h \\
 689.7 &= -749j - 16.65k + 694l + m \\
 -641.9 &= -673.5a - 153.8b + 707.5c + d \\
 -232.1 &= -673.5e - 153.8f + 707.5g + h \\
 716.5 &= -673.5j - 153.8k + 707.5l + m \\
 -718.2 &= -738.2a - 157.9b + 71.29c + d \\
 -236.5 &= -738.2e - 157.9f + 71.29g + h \\
 74.2 &= -738.2j - 157.9k + 71.29l + m \\
 afl + bgj + cek - bel - agk - cfj &= 1
 \end{aligned}$$

Recuérdese que para este sistema de ecuaciones la rotación entre las nubes de puntos de análisis está representada por las variables  $a, b, c, e, f, g, j, k, y l$ , mientras que la traslación está representada por las variables  $d, h$  y  $m$ .

El sistema de ecuaciones planteado anteriormente se resuelve fácilmente con ayuda de la herramienta “fsolve” del software Matlab. El código en Matlab para solucionar este tipo de sistemas de ecuaciones se presenta en anexo 2.

Al solucionar el sistema de ecuaciones se obtienen los valores para sus doce incógnitas, dichos valores se muestran en la tabla 4.5.

Tabla 4. 5. Valores de la matriz homogénea que representa la transformación 3D experimentada al inclinar 5° el HDL-64E.

Representación de la rotación						Representación de la traslación	
<b>A</b>	0.93463178340	<b>e</b>	0.08178894450	<b>j</b>	-0.14379689569	<b>d</b>	-48.95220797592
<b>B</b>	-0.11949796179	<b>f</b>	1.00801079049	<b>k</b>	-0.17364150820	<b>h</b>	-16.39547394830
<b>C</b>	0.02565094605	<b>g</b>	-0.00789780131	<b>l</b>	1.02531497840	<b>m</b>	-1.32463600138e+002

Con los valores presentados en la tabla anterior y tomando como base la ecuación 43, se puede expresar la matriz de transformación homogénea que representa la transformación 3D experimentada al inclinar 5° el HDL-64E de la siguiente manera:

$$H = \begin{bmatrix} 0.9346317 & -0.1194979 & 0.0256509 & -48.9522079 \\ 0.0817889 & 1.0080107 & -0.0078978 & -16.3954739 \\ -0.1437968 & -0.1736415 & 1.0253149 & -1.324636e+002 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

La matriz homogénea expuesta anteriormente, permite a través de la ecuación 48, llevar a la referencia horizontal cualquier nube de puntos de cualquier escenario adquirida con el HDL-64E posicionado a 5° de inclinación. Para referenciar nubes de puntos capturadas con una inclinación  $\theta$  distinta a 5° evidentemente es necesario utilizar otra matriz de transformación, esto significa repetir toda la metodología de cálculo de las matrices de transformación homogéneas, desde la selección de las nubes de puntos  $N_\theta$  y  $N_0$  y la extracción de sus respectivos puntos correlacionados hasta la formación y solución del sistema de ecuaciones que permita obtener la nueva matriz de transformación homogénea.

Como es sabido, la inclinación del sensor para ampliar su campo de visión vertical se da a través del sistema de posicionamiento HDL-64E (descrito en la sección 4.1), el cual tiene un espacio de movilidad de 43° y una resolución angular muy pequeña, esto significa que existe una cantidad enorme de posiciones angulares  $\theta$  con las que se podrían obtener capturas del sensor. Calcular las matrices homogéneas para todas las posiciones angulares que permita la resolución del sistema de posicionamiento sería extremadamente tedioso y carente de sentido, por esta razón se decidió calcular las matrices de transformación homogéneas para poder realizar una serie de barridos con el HDL-64E cada 5° de posicionamiento, lo que quiere decir que para alcanzar los 43° de máximo posicionamiento es necesario calcular 9 matrices de transformación homogéneas. Dichas matrices de transformación se presentan en anexo 3.

## Capítulo 5: Validación experimental

En este capítulo se presentan los resultados finales del trabajo de investigación, dichos resultados se enmarcan en la aplicación del sistema de posicionamiento HDL-64E para aumentar el campo de visión vertical del sensor.

En la figura 5.1 (a) se presenta un escenario el cual fue escaneado con el sensor montado sobre el sistema de posicionamiento HDL-64E. Los escaneos se realizaron con la referencia horizontal del sensor y a 5° de inclinación (figura 5.1 (b) y (c) respectivamente). Las nubes resultantes son las mismas que las presentadas en la figura 4.19, sin embargo, estas se muestran ahora graficadas en el software Geomagic Studio.

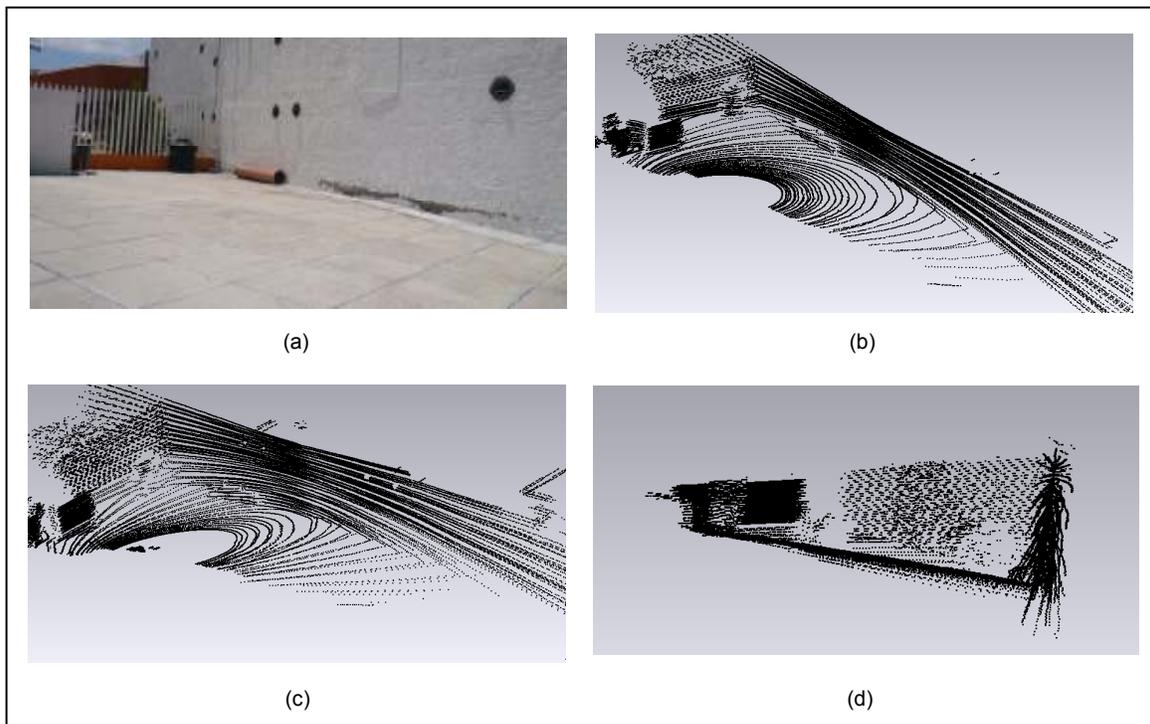


Fig. 5. 1. Capturas con el sistema de posicionamiento HDL-64E. (a) Escenario escaneado. (b) Nube de puntos capturada a 0°. (c) Nube de puntos capturada a 5°. (d) Vista lateral de la nube de puntos capturada a 5°.

La nube de puntos presentada en la figura 5.1 (c) contiene más información de la altura del escenario escaneado que la nube de puntos presentada en la figura 5.1 (b), evidentemente esto

se debe a que ésta fue obtenida inclinando  $5^\circ$  el HDL-64E. A pesar de lo anterior, como se mencionó en la sección 4.3, esta nube de puntos ha sufrido una drástica transformación en cuanto a orientación y posición debido a la inclinación del sensor. La figura 5.1 (d) es una vista lateral donde se evidencia el efecto experimentado por la nube de puntos al inclinar  $5^\circ$  el HDL-64E.

En la sección 4.3 se desarrolló toda una metodología para transformar nubes de puntos capturadas con el HDL-64E inclinado  $\theta$  grados. Dicha metodología consiste en estimar las matrices homogéneas que definen las transformaciones 3D generadas al inclinar el sensor, para posteriormente utilizar la inversa de dichas matrices para revertir el efecto de transformación.

La nube de puntos de la figura 5.1 (c) fue obtenida al inclinar  $5^\circ$  el HDL-64E, de tal manera que para corregir la transformación 3D experimentada es necesario utilizar la matriz  $H_5$  que se presenta en anexo 3. Las correcciones de las nubes de puntos se hacen al aplicar la ecuación 48, la cual para el caso de la corrección de la nube de puntos de la figura 5.1 (c) se puede reescribir de la siguiente manera:

$$N_5^0 = H_5^{-1} N_5$$

Donde:

$N_5$  : Nube de puntos adquirida con el sistema de posicionamiento al inclinar  $5^\circ$  grados el HDL-64E.

$H_5$ : Matriz homogénea que define la transformación 3D que experimenta una nube de puntos al inclinar  $5^\circ$  grados el HDL-64E.

$N_5^0$  : Representación de la nube de puntos  $N_5$  bajo el sistema de referencia horizontal del sensor.

Al aplicar la ecuación anterior se obtiene la expresión corregida de la nube de puntos de la figura 5.1 (c). La nube de puntos resultante  $N_5^0$  incluye la misma información y tiene el mismo tamaño que la nube de puntos de la figura 5.1 (c), pero ha sido llevada a la referencia horizontal del sensor de manera que el efecto de transformación en posición y orientación generado por la inclinación del escáner ha desaparecido. El código en Matlab que permite realizar la corrección de las nubes de puntos se presenta en anexo 2 y el resultado de la corrección se muestra en la figura 5.2.

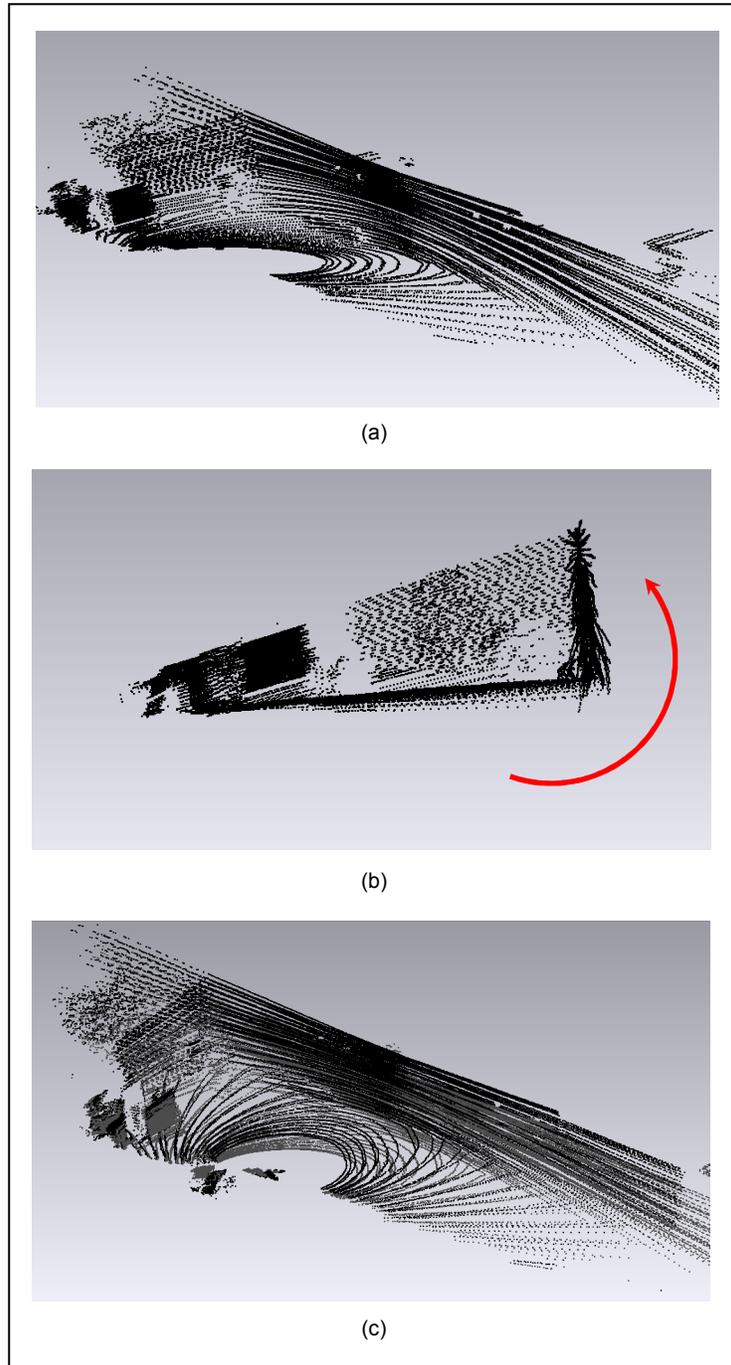


Fig. 5. 2. Transformación de la nube de puntos. (a) Vista global de la nube de puntos  $N_s^0$ . (b) Vista lateral de la nube de puntos  $N_s^0$ . (c) Gráfica simultánea de la nube de puntos  $N_s^0$  y la nube de puntos  $N_s$ .

Si se comparan las figuras 5.1 (d) y 5.2 (b), las cuales corresponden a vistas laterales de la nube de puntos capturada a  $5^\circ$  y su respectiva nube de puntos llevada a referencia horizontal del sensor, se evidencia como el efecto de transformación efectivamente ha desaparecido. La

transformación de la nube de puntos se da prácticamente siguiendo la trayectoria de la flecha roja que se muestra en la figura 5.2 (b). Si se grafican simultáneamente las nubes de puntos (la capturada a  $5^\circ$  y su transformación), tal y como se presenta en la figura 5.2 (c), se evidencia que la nube de puntos que está bajo la referencia horizontal del escáner (representada en negro) se encuentra ligeramente por encima de la nube de puntos capturada con  $5^\circ$  de inclinación (representada en gris), lo que se puede traducir en una transformación exitosa. El proceso de llevar las nubes de puntos a la referencia horizontal del sensor permite obtener una nube como la presentada en la figura 5.2 (a), la cual representa una nube de puntos capturada con el HDL-64E con un aumento de  $5^\circ$  en su ángulo de visión vertical

En la sección 4.3.3 se menciona que una vez que se estima la matriz homogénea que representa la transformación 3D experimentada al inclinar el HDL-64E  $\theta$  grados, dicha matriz puede utilizarse para referenciar cualquier nube de puntos de cualquier escenario siempre y cuando ésta sea capturada con los mismos  $\theta$  grados de inclinación. Lo anterior quiere decir que la matriz  $H_5$  utilizada para transformar la nube de puntos mostrada en la figura 5.1 (c) podría utilizarse para cualquier nube de puntos capturada con  $5^\circ$  de inclinación.

En la figura 4.17 se muestra un edificio de aproximadamente 14 metros de altura, el cual para poder ser visualizado por completo requiere de la realización dos barridos con el sensor montado sobre el sistema de posicionamiento HDL-64E, uno de los barridos es a  $5^\circ$  y la nube de puntos resultante, que lógicamente también experimenta una transformación en posición y orientación, se muestra en la figura 5.3.

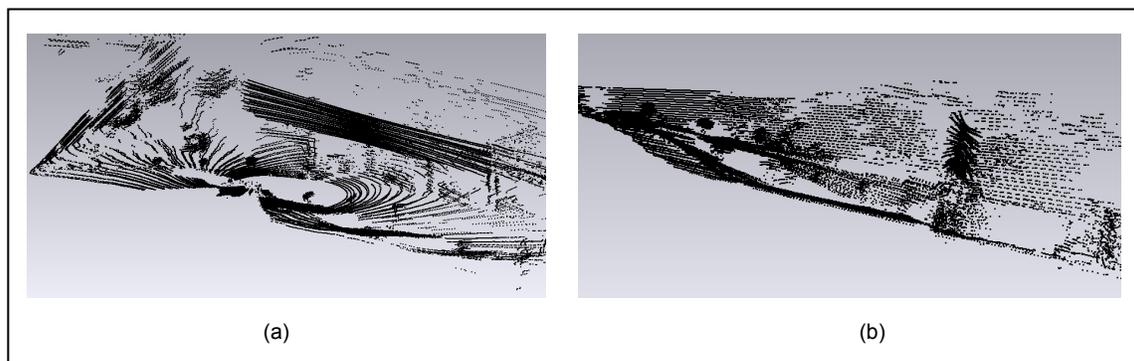


Fig. 5. 3. Nube de puntos de un edificio capturada con  $5^\circ$  de inclinación. (a) Vista global de la nube de puntos. (b) Vista lateral de la nube de puntos.

Al aplicar la misma matriz homogénea  $H_5$  utilizada para transformar la nube de puntos de la figura 5.1 (c), se puede obtener también una expresión equivalente para la nube de puntos de la

figura 5.3 (a). El resultado del proceso de transformación de dicha nube de puntos se presenta en la figura 5.4.

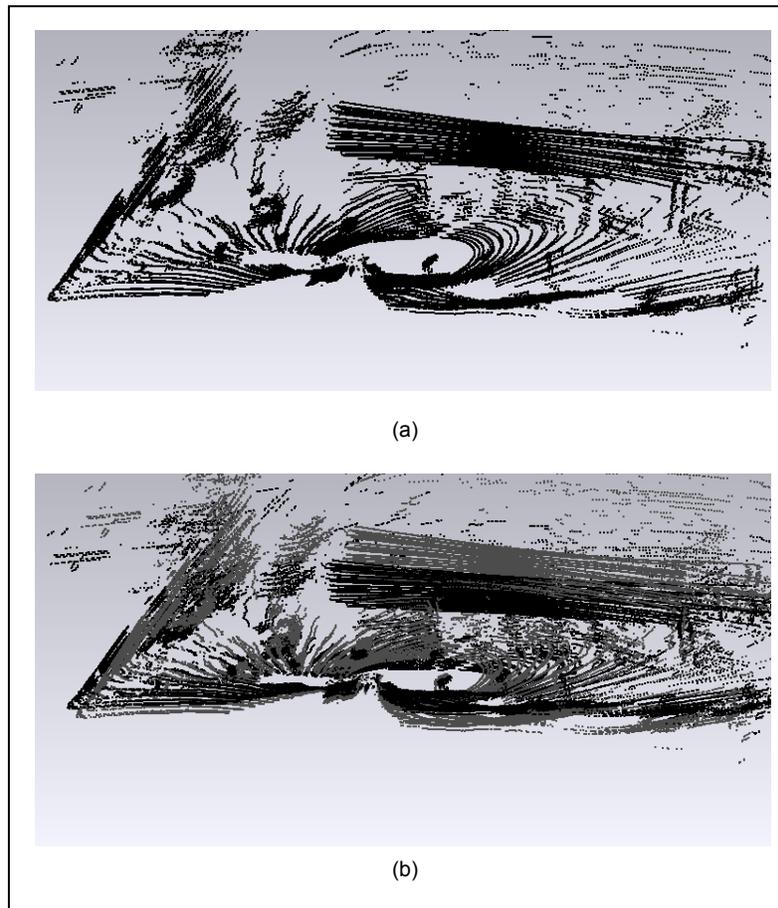


Fig. 5.4. Transformación de la nube de puntos adquirida de un edificio a 5° de inclinación. (a) Vista global de la nube de puntos transformada. (b) Gráfica simultánea de la nube de puntos transformada y la nube de puntos capturada 5°.

Los resultados mostrados en la figura 5.4 son similares a los presentados anteriormente. La transformación experimentada por la nube de puntos obtenida al inclinar el sensor ha desaparecido. Si se grafican en simultaneo ambas nubes de puntos (la capturada a 5° y su transformación) se percibe como la nube de puntos que ahora está bajo el sistema de referencia horizontal del sensor (representada en negro) se encuentra posicionada por encima de la nube de puntos capturada a 5° de inclinación, esto se puede notar fácilmente en la figura 5.4 (b).

Con lo anterior se comprueba el planteamiento de una misma matriz homogénea  $H_{\theta}$ , que representa la transformación 3D experimentada al inclinar  $\theta$  grados el HDL-64E, puede ser utilizada para transformar cualquier nube de puntos de cualquier escenario siempre y cuando ésta sea capturada con los mismos  $\theta$  grados de inclinación.

Inclinar unos cuantos grados el HDL-64E con el sistema de posicionamiento podría en ciertos casos con una única captura abarcar toda la estructura que interesa escanear, sin embargo, como se mencionó anteriormente, para el caso de estructuras muy altas como el edificio de la figura 4.17, es necesario realizar más de una captura a diferentes niveles de inclinación. Esto implica corregir todas las capturas que se hagan y aprovechando que se encuentran bajo un mismo sistema de referencia graficarlas en simultáneo para obtener una visión completa del escenario.

Para el caso del edificio mostrado en la figura 4.17 es necesario realizar dos escaneos, uno a  $5^\circ$  y otro a  $30^\circ$ . La transformación para la nube de puntos a  $5^\circ$  se presenta en la figura 5.4 (a), sin embargo la transformación para la nube de puntos a  $30^\circ$  (mostrada en la figura 4.18) se presenta en la figura 5.5.

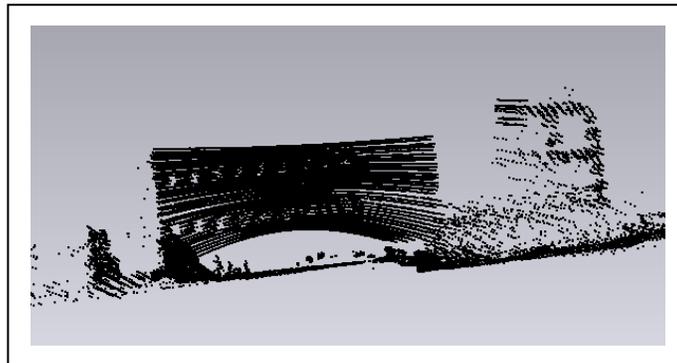


Fig. 5. 5. Corrección de la nube de puntos escaneada a  $30^\circ$  mostrada en la figura 4.18.

Ya con las dos nubes de puntos que representan el escenario llevadas a la referencia horizontal del sensor es posible graficarlas simultáneamente para obtener una nube de puntos completa, la cual se presenta en la figura 5.6.

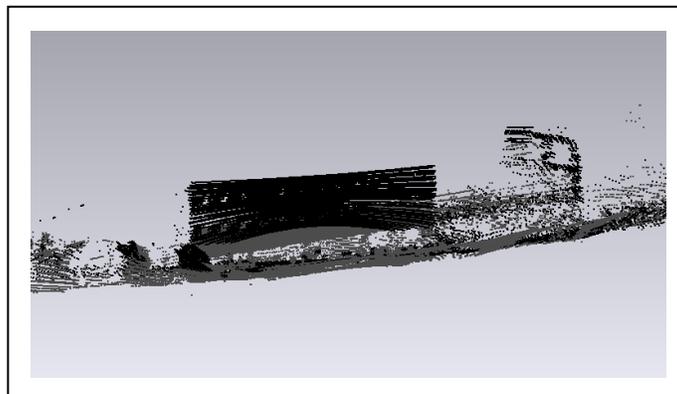


Fig. 5. 6. Nube de puntos que representa completamente el escenario escaneado.

En la figura 5.7 se presenta una vista panorámica del escenario escaneado, en donde se evidencia que la nube de puntos de la figura 5.6 representa bastante bien el escenario escaneado.



Fig. 5. 7. Vista panorámica del escenario escaneado

La figura 5.8 muestra una vista superior de la nube de puntos del escenario completo. En dicha figura se evidencia la presencia de líneas ruidosas que pertenecen a la nube de puntos adquirida con el HDL-64E posicionado a 30°.

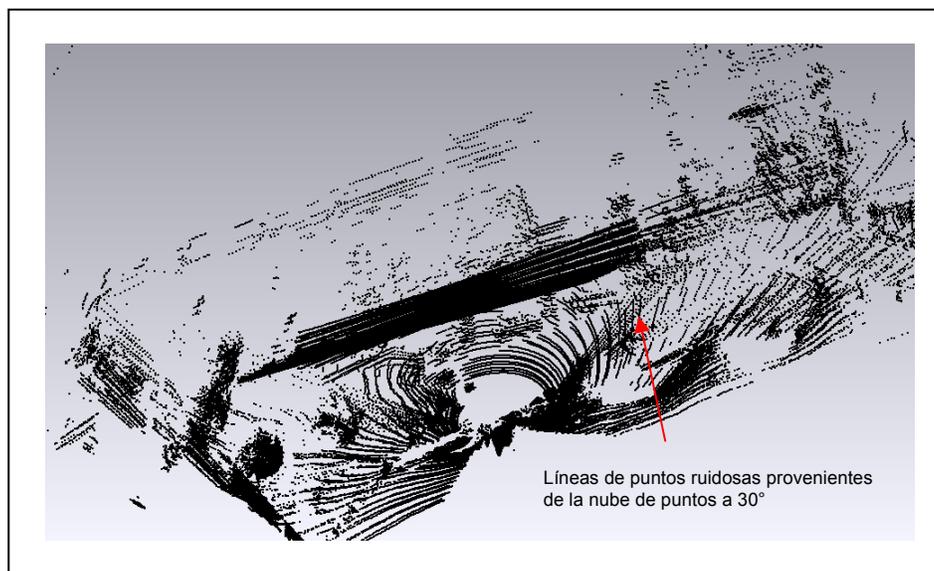


Fig. 5. 8. Vista superior de la nube de puntos del escenario completo.

Dichas líneas de puntos ruidosas se pueden eliminar manualmente de la nube de puntos capturada a 30° antes o después de ser transformadas, tal y como se muestra en la figura 5.9.

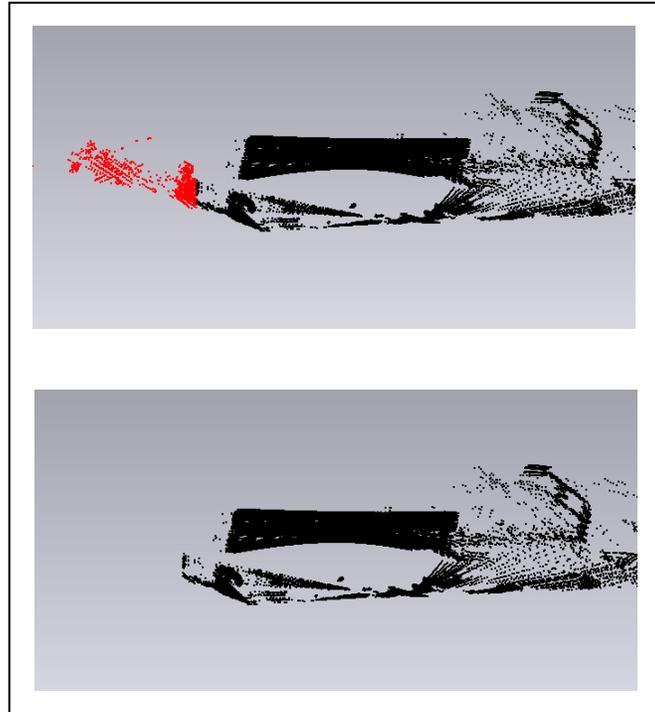


Fig. 5. 9. Eliminación manual de información ruidosa.

Una vez que se elimina manualmente la información que no es de interés perteneciente a la nube de puntos capturada a 30°, se puede volver a graficar simultáneamente las nubes de puntos corregidas que representan el escenario. En la figura 5.10 se presenta nuevamente una vista superior de la nube de puntos del escenario completo, sin embargo ahora se aprecia un mejor resultado debido a la eliminación de la información ruidosa.

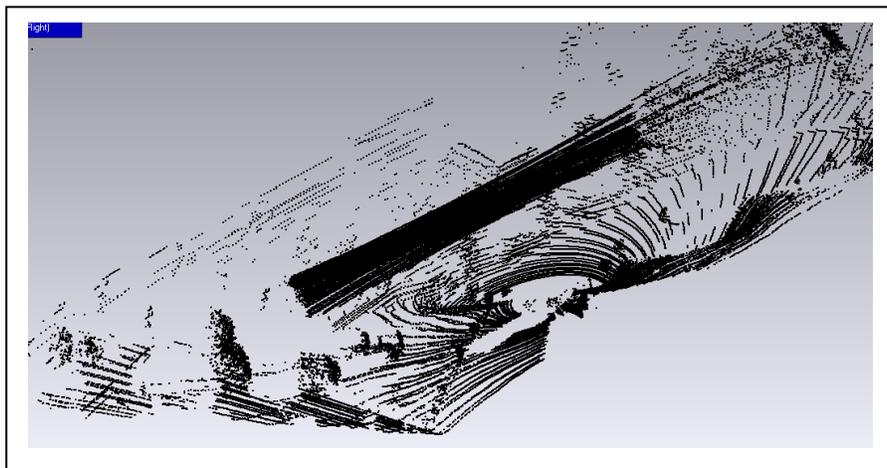


Fig. 5. 10. Vista superior de la nube de puntos del escenario completo después de la supresión de ruido.

---

## Capítulo 6: Conclusiones y trabajo futuro

---

Se diseñó y manufacturó una estructura mecánica de un grado de libertad muy funcional: es ligera, con poca fricción durante operación, de gran estabilidad y de alta confiabilidad para el montaje y manipulación de un equipo de alta tecnología como el LIDAR HDL-64E. Esta estructura, con un único actuador, es capaz de posicionar desde 0 a 43° el HDL-64E, lo que permite que el sensor amplíe su campo de visión vertical de 26.8° a 69.8°.

El sistema de posicionamiento del HDL-64E cuenta con un controlador PID en posición angular, que permite un preciso posicionamiento del sensor con una elevada resolución. El usuario a través de la interfaz desarrollada en un lenguaje de alto desempeño puede fácilmente seleccionar el ángulo en el que se desea fijar el sensor, además de que presenta la posibilidad de modificar las ganancias PID para obtener una mejor respuesta del controlador según las necesidades del usuario.

Se desarrolló una metodología para transformar las nubes de puntos adquiridas con el sensor a un determinado nivel de inclinación. Esta corrección es necesaria debido a que las nubes de puntos obtenidas con el sensor cuando está inclinado experimentan un cambio en su sistema de referencia, concretamente lo que sucede es que estos sistemas de referencia experimentan una rotación y una pequeña traslación con respecto a la referencia horizontal o inicial de sensor. El proceso de transformación de las nubes de puntos consiste en considerar la referencia horizontal del sensor como un sistema de referencia global, y a través de la estimación y aplicación de matrices de transformación homogéneas referenciar las adquisiciones hechas con el sensor inclinado a dicho sistema de referencia global.

Con el prototipo mecatrónico propuesto en este trabajo y la metodología de corrección de nubes de puntos, se obtiene finalmente todo un sistema muy robusto que permite maximizar el ángulo de visión vertical del sensor y en consecuencia potencializar sus perspectivas de aplicación. Con el sistema propuesto se alcanza el objetivo trazado de habilitar al sensor para obtener escaneos de estructuras cuyas alturas estén en el orden de los 10 metros de altura.

El sistema está listo para ser enviado al campo e iniciar a realizar tareas de reconstrucción 3D de espacios urbanos, o para realizar cualquier otra actividad que requiera la utilización de un escáner con un elevado ángulo de visión vertical.



**Trabajo futuro:**

1. Diseñar patrones especiales para que el cálculo de las matrices homogéneas de transformación no se tengan que hacer con escenarios físicos, esto permitiría obtener matrices mucho más precisas.
2. Utilizar más de cuatro puntos correlacionados para aumentar la precisión en el cálculo de las matrices homogéneas de corrección.
3. Realizar trabajos más profundos relacionados al pegado de nubes de puntos capturados con diferentes niveles de inclinación del HDL-64E.

---

## Referencias

---

- [1] E. Tunc, F. Karsli, E. Ayhan. 3D city reconstruction by different technologies to manage and reorganize the current situation. 2004.
- [2] C. Brenner. Interactive modelling tools for 3D building reconstruction. Photogrammetric Week. 1999.
- [3] R. Halterman, M. Bruch. Velodyne HDL-64E LIDAR for unmanned surface vehicle obstacle detection. 2010.
- [4] A. Román, G. Garg, M. Levoy. Interactive design of multi-perspective images for visualizing urban landscapes. 2004.
- [5] E. Baltsavias, F. Remondino, R. Weckerle. Image-based virtual reconstruction of complex architectures using sophisticated automated procedures: the artemis temple in jerash, jordan. 2006.
- [6] Debevec, Taylor, Malik. Modelling and rendering architecture from photographs: a hybrid geometry- and image-based approach. 1996.
- [7] L. Dorffner, G. Forkert. Generation and visualization of 3D photo-models using hybrid block adjustment with assumptions on the object shape. 1998.
- [8] M. Pollefeys, R. Koch, M. Vergauwen, L. Van Gool. Automated reconstruction of 3D scenes from sequences of images. 2000.
- [9] S. Scalable. Controlled imagery capture in urban environments. 2001.
- [10] C. Hug. Extracting artificial surface objects from airborne laser scanner data. 1997.

- [11] N. Haala, C. Brenner. Generation of 3-D city models from airborne laser scanning data. 3rd EARSL Workshop on LIDAR Remote Sensing on Land and Sea. 1997.
- [12] J. Shan, T. Palmer. Effective urban modeling and visualization using Lidar Data in GIS. 2001.
- [13] J. Biosaca, S. Navarro, J. Lerma. Modelado tridimensional de una bóveda barroca mediante la combinación de láser escáner y fotogrametría. 7 Setmana Geomática. 2007.
- [14] A white paper on lidar mapping. Ambergcore terrapoint division offices.. 2008.
- [15] Instituto de Rehabilitación Urbana y el Instituto de Ciencias Virtuales de Madrid. Modelado 3D de una manzana en el casco antiguo de la ciudad de Madrid.
- [16] J. Roca, A. Marambio. Modelos digitales de nubes de puntos de la Habana Vieja, Cuba.
- [17] F. Gómez Bravo, A. Vale, Gomes Mota J. Prototipo para la monitorización de la infraestructura ferroviaria utilizando sensores LIDAR, imagen de video y GPS.
- [18] J. Rico, J. González, J. Hurtado, J. Salas, F. Ornelas, M. García, J. Cortes. Construcción rápida de mapas para la navegación autónoma. Centro de Investigación en Ciencia Aplicada y Tecnología Avanzada, unidad Querétaro. 2009.
- [19] J. Rico. Construcción de mapas 3D a partir de la extracción de primitivas geométricas obtenidas de datos de un LIDAR. 2009.
- [20] C. Glennie, D. Lichti. Static calibration and analysis of the Velodyne HDL-64E S2 for High Accuracy Mobile Scanning. 2010.
- [21] Velodyne Acoustics, Inc. HDL-64E user's manual. 2007.
- [22] S. Cardona, D. Clos. Teoría de maquinas. Universidad Politécnica de Cataluña. Primera edición 2001.
- [23] B. Bolt, J. Hiscocks. Machines, mechanisms and mathematics. Chatto & Windus, 1970.



[24] J. Pichardo. Compensación de zona muerta y variación de carga en el control de velocidad de motores de corriente directa. Centro de Investigación en Ciencia Aplicada y Tecnología Avanzada, unidad Querétaro. 2008.

[25] D. Graupe. Identification of systems. 1972

[26] R. Burden, D. Faires. Numerical analysis. 1997.

[27] I. Markovsky, J. Willems. Application of structured total least squares for system identification and model reduction. IEEE Transactions on Automatic Control. 2005.

[28] K. Ogata. Ingeniería de control moderna. Pearson Educación, S.A. Madrid. 2003

[29] L. Tsai. "Robot Analysis. The Mechanics of serial and Parallel Manipulators". John Wiley & sons. EEUU. 1999.

[30] J. Kerr. PIC-SERVO Motion Control Board. 2005

[31] J. Kerr. SSA-Smart Serial Adapter. 2005.

[32] R. Yañes. Resolución de mecanismo paralelo planar 3RRR impulsado por actuadores eléctricos. Centro de Investigación en Ciencia Aplicada y Tecnología Avanzada, unidad Querétaro. 2007.

[33] R. Alonso, J. Gómez. Alineamiento automático de nubes densas de puntos a partir de información geométrica y cromática. XXVIII Jornadas de Automática. 2006.



# Anexos



Anexo 1  
Código fuente de la “Interfaz de control LIDAR”



```
Public Class Form1
    Inherits System.Windows.Forms.Form

#Region " Windows Form Designer generated code "

    Public Sub New()
        MyBase.New()

        'This call is required by the Windows Form Designer.
        InitializeComponent()

        'Add any initialization after the InitializeComponent() call

    End Sub

    'Form overrides dispose to clean up the component list.
    Protected Overloads Overrides Sub Dispose(ByVal disposing As
Boolean)
        If disposing Then
            If Not (components Is Nothing) Then
                components.Dispose()
            End If
        End If
        MyBase.Dispose(disposing)
    End Sub

    'Required by the Windows Form Designer
    Private components As System.ComponentModel.IContainer

    'NOTE: The following procedure is required by the Windows Form
Designer
    'It can be modified using the Windows Form Designer.
    'Do not modify it using the code editor.
    Friend WithEvents calibrar As System.Windows.Forms.Button
    Friend WithEvents MoveButton As System.Windows.Forms.Button
    Friend WithEvents BarraPV As System.Windows.Forms.ProgressBar
    Friend WithEvents BarraSP As System.Windows.Forms.ProgressBar
    Friend WithEvents Label1 As System.Windows.Forms.Label
    Friend WithEvents NumPosdes As System.Windows.Forms.NumericUpDown
    Friend WithEvents GroupBox1 As System.Windows.Forms.GroupBox
    Friend WithEvents Label3 As System.Windows.Forms.Label
    Friend WithEvents Label2 As System.Windows.Forms.Label
    Friend WithEvents Label6 As System.Windows.Forms.Label
    Friend WithEvents Label4 As System.Windows.Forms.Label
    Friend WithEvents Label5 As System.Windows.Forms.Label
    Friend WithEvents GroupBox2 As System.Windows.Forms.GroupBox
    Friend WithEvents Label9 As System.Windows.Forms.Label
    Friend WithEvents Label8 As System.Windows.Forms.Label
    Friend WithEvents Label7 As System.Windows.Forms.Label
    Friend WithEvents Kd As System.Windows.Forms.NumericUpDown
    Friend WithEvents Ki As System.Windows.Forms.NumericUpDown
    Friend WithEvents Kp As System.Windows.Forms.NumericUpDown
    Friend WithEvents Label10 As System.Windows.Forms.Label
    Friend WithEvents Vel As System.Windows.Forms.NumericUpDown
    Friend WithEvents Label11 As System.Windows.Forms.Label
    Friend WithEvents aceleracion As System.Windows.Forms.NumericUpDown
```



```
Friend WithEvents GroupBox3 As System.Windows.Forms.GroupBox
Friend WithEvents velocidad As System.Windows.Forms.ComboBox
Friend WithEvents puerto As System.Windows.Forms.ComboBox
Friend WithEvents Label13 As System.Windows.Forms.Label
Friend WithEvents ExitButton As System.Windows.Forms.Button
Friend WithEvents InitButton As System.Windows.Forms.Button
Friend WithEvents PictureBox1 As System.Windows.Forms.PictureBox
Friend WithEvents Label12 As System.Windows.Forms.Label
<System.Diagnostics.DebuggerStepThrough()> Private Sub
InitializeComponent()
    Dim resources As System.ComponentModel.ComponentResourceManager
= New System.ComponentModel.ComponentResourceManager(GetType(Form1))
    Me.calibrar = New System.Windows.Forms.Button
    Me.MoveButton = New System.Windows.Forms.Button
    Me.BarraPV = New System.Windows.Forms.ProgressBar
    Me.BarraSP = New System.Windows.Forms.ProgressBar
    Me.Label1 = New System.Windows.Forms.Label
    Me.NumPosdes = New System.Windows.Forms.NumericUpDown
    Me.GroupBox1 = New System.Windows.Forms.GroupBox
    Me.Label6 = New System.Windows.Forms.Label
    Me.Label4 = New System.Windows.Forms.Label
    Me.Label5 = New System.Windows.Forms.Label
    Me.Label3 = New System.Windows.Forms.Label
    Me.Label2 = New System.Windows.Forms.Label
    Me.GroupBox2 = New System.Windows.Forms.GroupBox
    Me.Label11 = New System.Windows.Forms.Label
    Me.aceleracion = New System.Windows.Forms.NumericUpDown
    Me.Label10 = New System.Windows.Forms.Label
    Me.Vel = New System.Windows.Forms.NumericUpDown
    Me.Label9 = New System.Windows.Forms.Label
    Me.Label8 = New System.Windows.Forms.Label
    Me.Label7 = New System.Windows.Forms.Label
    Me.Kd = New System.Windows.Forms.NumericUpDown
    Me.Ki = New System.Windows.Forms.NumericUpDown
    Me.Kp = New System.Windows.Forms.NumericUpDown
    Me.GroupBox3 = New System.Windows.Forms.GroupBox
    Me.ExitButton = New System.Windows.Forms.Button
    Me.velocidad = New System.Windows.Forms.ComboBox
    Me.InitButton = New System.Windows.Forms.Button
    Me.puerto = New System.Windows.Forms.ComboBox
    Me.Label13 = New System.Windows.Forms.Label
    Me.Label12 = New System.Windows.Forms.Label
    Me.PictureBox1 = New System.Windows.Forms.PictureBox
    CType(Me.NumPosdes,
System.ComponentModel.ISupportInitialize).BeginInit()
    Me.GroupBox1.SuspendLayout()
    Me.GroupBox2.SuspendLayout()
    CType(Me.aceleracion,
System.ComponentModel.ISupportInitialize).BeginInit()
    CType(Me.Vel,
System.ComponentModel.ISupportInitialize).BeginInit()
    CType(Me.Kd,
System.ComponentModel.ISupportInitialize).BeginInit()
    CType(Me.Ki,
System.ComponentModel.ISupportInitialize).BeginInit()
```

```
        CType (Me.Kp,
System.ComponentModel.ISupportInitialize).BeginInit()
        Me.GroupBox3.SuspendLayout()
        CType (Me.PictureBox1,
System.ComponentModel.ISupportInitialize).BeginInit()
        Me.SuspendLayout()
        |
        'calibrar
        |
        Me.calibrar.Location = New System.Drawing.Point(9, 196)
        Me.calibrar.Name = "calibrar"
        Me.calibrar.Size = New System.Drawing.Size(186, 24)
        Me.calibrar.TabIndex = 2
        Me.calibrar.Text = "Calibrar Posición Cero"
        |
        'MoveButton
        |
        Me.MoveButton.Location = New System.Drawing.Point(6, 84)
        Me.MoveButton.Name = "MoveButton"
        Me.MoveButton.Size = New System.Drawing.Size(186, 24)
        Me.MoveButton.TabIndex = 3
        Me.MoveButton.Text = "Ir a posición"
        |
        'BarraPV
        |
        Me.BarraPV.BackColor =
System.Drawing.SystemColors.ButtonHighlight
        Me.BarraPV.ForeColor = System.Drawing.Color.SkyBlue
        Me.BarraPV.Location = New System.Drawing.Point(23, 150)
        Me.BarraPV.Name = "BarraPV"
        Me.BarraPV.Size = New System.Drawing.Size(161, 20)
        Me.BarraPV.Style =
System.Windows.Forms.ProgressBarStyle.Continuous
        Me.BarraPV.TabIndex = 6
        |
        'BarraSP
        |
        Me.BarraSP.BackColor =
System.Drawing.SystemColors.ButtonHighlight
        Me.BarraSP.ForeColor = System.Drawing.Color.DeepSkyBlue
        Me.BarraSP.Location = New System.Drawing.Point(23, 45)
        Me.BarraSP.Name = "BarraSP"
        Me.BarraSP.Size = New System.Drawing.Size(161, 21)
        Me.BarraSP.Style =
System.Windows.Forms.ProgressBarStyle.Continuous
        Me.BarraSP.TabIndex = 7
        |
        'Labell
        |
        Me.Labell.AutoSize = True
        Me.Labell.BackColor = System.Drawing.Color.Transparent
        Me.Labell.Location = New System.Drawing.Point(6, 19)
        Me.Labell.Name = "Labell"
        Me.Labell.Size = New System.Drawing.Size(134, 13)
        Me.Labell.TabIndex = 9
        Me.Labell.Text = "Posición deseada (Grados)"
```



```

'
'NumPosdes
'
Me.NumPosdes.Increment = New Decimal(New Integer() {1, 0, 0,
65536})
Me.NumPosdes.Location = New System.Drawing.Point(143, 19)
Me.NumPosdes.Maximum = New Decimal(New Integer() {43, 0, 0, 0})
Me.NumPosdes.Name = "NumPosdes"
Me.NumPosdes.Size = New System.Drawing.Size(40, 20)
Me.NumPosdes.TabIndex = 10
'
'GroupBox1
'
Me.GroupBox1.BackgroundImage =
Global.PICSERVO_VB_Example.My.Resources.Resources.hoja_papel_14
Me.GroupBox1.Controls.Add(Me.Label6)
Me.GroupBox1.Controls.Add(Me.Label4)
Me.GroupBox1.Controls.Add(Me.Label5)
Me.GroupBox1.Controls.Add(Me.calibrar)
Me.GroupBox1.Controls.Add(Me.MoveButton)
Me.GroupBox1.Controls.Add(Me.Label3)
Me.GroupBox1.Controls.Add(Me.Label2)
Me.GroupBox1.Controls.Add(Me.NumPosdes)
Me.GroupBox1.Controls.Add(Me.Label1)
Me.GroupBox1.Controls.Add(Me.BarraSP)
Me.GroupBox1.Controls.Add(Me.BarraPV)
Me.GroupBox1.Location = New System.Drawing.Point(18, 99)
Me.GroupBox1.Name = "GroupBox1"
Me.GroupBox1.Size = New System.Drawing.Size(201, 234)
Me.GroupBox1.TabIndex = 11
Me.GroupBox1.TabStop = False
Me.GroupBox1.Text = "Control de Posición"
'
'Label6
'
Me.Label6.AutoSize = True
Me.Label6.BackColor = System.Drawing.Color.Transparent
Me.Label6.Location = New System.Drawing.Point(6, 128)
Me.Label6.Name = "Label6"
Me.Label6.Size = New System.Drawing.Size(110, 13)
Me.Label6.TabIndex = 16
Me.Label6.Text = "Posición real (Grados)"
'
'Label4
'
Me.Label4.AutoSize = True
Me.Label4.BackColor = System.Drawing.Color.Transparent
Me.Label4.Location = New System.Drawing.Point(166, 179)
Me.Label4.Name = "Label4"
Me.Label4.Size = New System.Drawing.Size(19, 13)
Me.Label4.TabIndex = 15
Me.Label4.Text = "43"
'
'Label5
'
Me.Label5.AutoSize = True

```



```
Me.Label5.BackColor = System.Drawing.Color.Transparent
Me.Label5.Location = New System.Drawing.Point(18, 177)
Me.Label5.Name = "Label5"
Me.Label5.Size = New System.Drawing.Size(13, 13)
Me.Label5.TabIndex = 14
Me.Label5.Text = "0"
'
'Label3
'
Me.Label3.AutoSize = True
Me.Label3.BackColor = System.Drawing.Color.Transparent
Me.Label3.Location = New System.Drawing.Point(169, 65)
Me.Label3.Name = "Label3"
Me.Label3.Size = New System.Drawing.Size(19, 13)
Me.Label3.TabIndex = 13
Me.Label3.Text = "43"
'
'Label2
'
Me.Label2.AutoSize = True
Me.Label2.BackColor = System.Drawing.Color.Transparent
Me.Label2.Location = New System.Drawing.Point(21, 63)
Me.Label2.Name = "Label2"
Me.Label2.Size = New System.Drawing.Size(13, 13)
Me.Label2.TabIndex = 12
Me.Label2.Text = "0"
'
'GroupBox2
'
Me.GroupBox2.BackgroundImage =
Global.PICSERVO_VB_Example.My.Resources.Resources.hoja_papel_13
Me.GroupBox2.Controls.Add(Me.Label11)
Me.GroupBox2.Controls.Add(Me.aceleracion)
Me.GroupBox2.Controls.Add(Me.Label10)
Me.GroupBox2.Controls.Add(Me.Vel)
Me.GroupBox2.Controls.Add(Me.Label9)
Me.GroupBox2.Controls.Add(Me.Label8)
Me.GroupBox2.Controls.Add(Me.Label7)
Me.GroupBox2.Controls.Add(Me.Kd)
Me.GroupBox2.Controls.Add(Me.Ki)
Me.GroupBox2.Controls.Add(Me.Kp)
Me.GroupBox2.Location = New System.Drawing.Point(247, 99)
Me.GroupBox2.Name = "GroupBox2"
Me.GroupBox2.Size = New System.Drawing.Size(160, 170)
Me.GroupBox2.TabIndex = 12
Me.GroupBox2.TabStop = False
Me.GroupBox2.Text = "Parámetros de Entrada"
'
'Label11
'
Me.Label11.AutoSize = True
Me.Label11.BackColor = System.Drawing.Color.Transparent
Me.Label11.Location = New System.Drawing.Point(6, 136)
Me.Label11.Name = "Label11"
Me.Label11.Size = New System.Drawing.Size(63, 13)
Me.Label11.TabIndex = 20
```

```
Me.Label11.Text = "Aceleración"
'
'aceleracion
'
Me.aceleracion.Increment = New Decimal(New Integer() {5, 0, 0,
65536})
Me.aceleracion.Location = New System.Drawing.Point(79, 134)
Me.aceleracion.Maximum = New Decimal(New Integer() {1000000, 0,
0, 0})
Me.aceleracion.Name = "aceleracion"
Me.aceleracion.Size = New System.Drawing.Size(66, 20)
Me.aceleracion.TabIndex = 19
Me.aceleracion.Value = New Decimal(New Integer() {100, 0, 0, 0})
'
'Label10
'
Me.Label10.AutoSize = True
Me.Label10.BackColor = System.Drawing.Color.Transparent
Me.Label10.Location = New System.Drawing.Point(6, 104)
Me.Label10.Name = "Label10"
Me.Label10.Size = New System.Drawing.Size(54, 13)
Me.Label10.TabIndex = 18
Me.Label10.Text = "Velocidad"
'
'Vel
'
Me.Vel.Increment = New Decimal(New Integer() {5, 0, 0, 65536})
Me.Vel.Location = New System.Drawing.Point(79, 102)
Me.Vel.Maximum = New Decimal(New Integer() {1000000, 0, 0, 0})
Me.Vel.Name = "Vel"
Me.Vel.Size = New System.Drawing.Size(66, 20)
Me.Vel.TabIndex = 17
Me.Vel.Value = New Decimal(New Integer() {100000, 0, 0, 0})
'
'Label9
'
Me.Label9.AutoSize = True
Me.Label9.BackColor = System.Drawing.Color.Transparent
Me.Label9.Location = New System.Drawing.Point(6, 78)
Me.Label9.Name = "Label9"
Me.Label9.Size = New System.Drawing.Size(20, 13)
Me.Label9.TabIndex = 16
Me.Label9.Text = "Kd"
'
'Label8
'
Me.Label8.AutoSize = True
Me.Label8.BackColor = System.Drawing.Color.Transparent
Me.Label8.Location = New System.Drawing.Point(6, 50)
Me.Label8.Name = "Label8"
Me.Label8.Size = New System.Drawing.Size(16, 13)
Me.Label8.TabIndex = 15
Me.Label8.Text = "Ki"
'
'Label7
'
```

```
Me.Label7.AutoSize = True
Me.Label7.BackColor = System.Drawing.Color.Transparent
Me.Label7.Location = New System.Drawing.Point(6, 26)
Me.Label7.Name = "Label7"
Me.Label7.Size = New System.Drawing.Size(20, 13)
Me.Label7.TabIndex = 14
Me.Label7.Text = "Kp"
'
'Kd
'
Me.Kd.Increment = New Decimal(New Integer() {5, 0, 0, 65536})
Me.Kd.Location = New System.Drawing.Point(79, 76)
Me.Kd.Maximum = New Decimal(New Integer() {10000, 0, 0, 0})
Me.Kd.Name = "Kd"
Me.Kd.Size = New System.Drawing.Size(66, 20)
Me.Kd.TabIndex = 13
Me.Kd.Value = New Decimal(New Integer() {1000, 0, 0, 0})
'
'Ki
'
Me.Ki.Increment = New Decimal(New Integer() {5, 0, 0, 65536})
Me.Ki.Location = New System.Drawing.Point(79, 50)
Me.Ki.Maximum = New Decimal(New Integer() {10000, 0, 0, 0})
Me.Ki.Name = "Ki"
Me.Ki.Size = New System.Drawing.Size(66, 20)
Me.Ki.TabIndex = 12
'
'Kp
'
Me.Kp.Increment = New Decimal(New Integer() {5, 0, 0, 65536})
Me.Kp.Location = New System.Drawing.Point(79, 24)
Me.Kp.Maximum = New Decimal(New Integer() {10000, 0, 0, 0})
Me.Kp.Name = "Kp"
Me.Kp.Size = New System.Drawing.Size(66, 20)
Me.Kp.TabIndex = 11
Me.Kp.Value = New Decimal(New Integer() {100, 0, 0, 0})
'
'GroupBox3
'
Me.GroupBox3.BackgroundImage =
Global.PICSERVO_VB_Example.My.Resources.Resources.hoja_papel_15
Me.GroupBox3.Controls.Add(Me.ExitButton)
Me.GroupBox3.Controls.Add(Me.velocidad)
Me.GroupBox3.Controls.Add(Me.InitButton)
Me.GroupBox3.Controls.Add(Me.puerto)
Me.GroupBox3.Controls.Add(Me.Label13)
Me.GroupBox3.Controls.Add(Me.Label12)
Me.GroupBox3.Location = New System.Drawing.Point(18, 12)
Me.GroupBox3.Name = "GroupBox3"
Me.GroupBox3.Size = New System.Drawing.Size(389, 81)
Me.GroupBox3.TabIndex = 13
Me.GroupBox3.TabStop = False
Me.GroupBox3.Text = "Comunicación"
'
'ExitButton
'
```



```
Me.ExitButton.Location = New System.Drawing.Point(215, 51)
Me.ExitButton.Name = "ExitButton"
Me.ExitButton.Size = New System.Drawing.Size(160, 24)
Me.ExitButton.TabIndex = 15
Me.ExitButton.Text = "Desconectar"
'
' velocidad
'
Me.velocidad.FormattingEnabled = True
Me.velocidad.Location = New System.Drawing.Point(295, 16)
Me.velocidad.Name = "velocidad"
Me.velocidad.Size = New System.Drawing.Size(80, 21)
Me.velocidad.TabIndex = 5
'
' InitButton
'
Me.InitButton.Location = New System.Drawing.Point(23, 51)
Me.InitButton.Name = "InitButton"
Me.InitButton.Size = New System.Drawing.Size(160, 24)
Me.InitButton.TabIndex = 14
Me.InitButton.Text = "Conectar"
'
' puerto
'
Me.puerto.FormattingEnabled = True
Me.puerto.Location = New System.Drawing.Point(103, 16)
Me.puerto.Name = "puerto"
Me.puerto.Size = New System.Drawing.Size(80, 21)
Me.puerto.TabIndex = 4
'
' Label13
'
Me.Label13.AutoSize = True
Me.Label13.BackColor = System.Drawing.Color.Transparent
Me.Label13.Location = New System.Drawing.Point(212, 19)
Me.Label13.Name = "Label13"
Me.Label13.Size = New System.Drawing.Size(54, 13)
Me.Label13.TabIndex = 2
Me.Label13.Text = "Velocidad"
'
' Label12
'
Me.Label12.AutoSize = True
Me.Label12.BackColor = System.Drawing.Color.Transparent
Me.Label12.Location = New System.Drawing.Point(22, 19)
Me.Label12.Name = "Label12"
Me.Label12.Size = New System.Drawing.Size(38, 13)
Me.Label12.TabIndex = 0
Me.Label12.Text = "Puerto"
'
' PictureBox1
'
Me.PictureBox1.Image =
CType(resources.GetObject("PictureBox1.Image"), System.Drawing.Image)
Me.PictureBox1.Location = New System.Drawing.Point(266, 290)
Me.PictureBox1.Name = "PictureBox1"
```

```
        Me.PictureBox1.Size = New System.Drawing.Size(114, 29)
        Me.PictureBox1.SizeMode =
System.Windows.Forms.PictureBoxSizeMode.StretchImage
        Me.PictureBox1.TabIndex = 14
        Me.PictureBox1.TabStop = False
    '
    'Form1
    '
        Me.AutoScaleBaseSize = New System.Drawing.Size(5, 13)
        Me.BackgroundImage =
CType(resources.GetObject("$this.BackgroundImage"),
System.Drawing.Image)
        Me.ClientSize = New System.Drawing.Size(435, 349)
        Me.Controls.Add(Me.PictureBox1)
        Me.Controls.Add(Me.GroupBox3)
        Me.Controls.Add(Me.GroupBox2)
        Me.Controls.Add(Me.GroupBox1)
        Me.Icon = CType(resources.GetObject("$this.Icon"),
System.Drawing.Icon)
        Me.Name = "Form1"
        Me.StartPosition =
System.Windows.Forms.FormStartPosition.CenterScreen
        Me.Text = "Interfaz de Control LIDAR"
        CType(Me.NumPosdes,
System.ComponentModel.ISupportInitialize).EndInit()
        Me.GroupBox1.ResumeLayout(False)
        Me.GroupBox1.PerformLayout()
        Me.GroupBox2.ResumeLayout(False)
        Me.GroupBox2.PerformLayout()
        CType(Me.aceleracion,
System.ComponentModel.ISupportInitialize).EndInit()
        CType(Me.Vel,
System.ComponentModel.ISupportInitialize).EndInit()
        CType(Me.Kd, System.ComponentModel.ISupportInitialize).EndInit()
        CType(Me.Ki, System.ComponentModel.ISupportInitialize).EndInit()
        CType(Me.Kp, System.ComponentModel.ISupportInitialize).EndInit()
        Me.GroupBox3.ResumeLayout(False)
        Me.GroupBox3.PerformLayout()
        CType(Me.PictureBox1,
System.ComponentModel.ISupportInitialize).EndInit()
        Me.ResumeLayout(False)

    End Sub

#End Region

    Dim Nummod As Integer 'asi se declara variable entera llamada Nummod

    Private Sub calibrar_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles calibrar.Click

        BarraPV.Value = 0
        ServoResetPos(1) 'instruccion para resetear la posicion del
servo a 0
```



```
End Sub

Private Sub MoveButton_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles MoveButton.Click
    Dim statbyte As Byte
    Dim pos As Int32
    Dim msg As String
    Dim angulo As Double

    If (NumPosdes.Value = 0) Then
        angulo = 0
    Else
        angulo = (NumPosdes.Value ^ 4) * (-0.0129) +
        (NumPosdes.Value ^ 3) * (0.6812) - (NumPosdes.Value ^ 2) * (32.799) -
        (NumPosdes.Value) * (3097.5) - 5713.8
    End If

    ServoStopMotor(1, AMP_ENABLE Or STOP_ABRUPT)
    ServoSetGain(1, Kp.Value, Kd.Value, Ki.Value, 0, 255, 0, 2000,
1, 0)

    ServoLoadTraj(1, LOAD_POS Or LOAD_VEL Or LOAD_ACC Or
ENABLE_SERVO Or START_NOW, angulo, Vel.Value, aceleración.Value, 0)

    Do
        NmcNoOp(1)                'NoOp command to read current
module status
        statbyte = NmcGetStat(1) 'Fetch the module status byte

        NmcReadStatus(1, SEND_POS) 'Read the current position data
from the cotnroller
        pos = ServoGetPos(1)        'Fetch the position data

        If (pos > 0) Then
            ServoResetPos(1)
        Else
            BarraPV.Value = pos * (-1)
        End If

        Loop While (statbyte And MOVE_DONE) = 0

    End Sub

Private Sub ExitButton_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs)
    Close()
End Sub

Private Sub Form1_Closing(ByVal sender As Object, ByVal e As
System.ComponentModel.CancelEventArgs) Handles MyBase.Closing
    If (Nummod > 0) Then NmcShutdown()
End Sub
```



'aquí se incorporan todos los parámetros cuando se carga por primera vez el programa

```
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
    puerto.Items.Add("COM1:")
    puerto.Items.Add("COM2:")
    puerto.Items.Add("COM3:")
    puerto.Items.Add("COM4:")
    puerto.Items.Add("COM5:")
    puerto.Items.Add("COM6:")
    velocidad.Items.Add("19200")
    velocidad.Items.Add("57600")
    velocidad.Items.Add("115200")
    BarraPV.Minimum = 0
    BarraPV.Maximum = 189500
    BarraSP.Minimum = 0
    BarraSP.Maximum = 189500
    ExitButton.Enabled = False
    NumPosdes.Enabled = False

    calibrar.Enabled = False
    MoveButton.Enabled = False
    Kp.Enabled = False
    Ki.Enabled = False
    Kd.Enabled = False
    Vel.Enabled = False
    aceleracion.Enabled = False

End Sub

Private Sub InitButton_Click_1(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles InitButton.Click
    Dim modtype As Integer

    Try
        Nummod = NmcInit(puerto.Text, velocidad.Text)

    Catch
        MsgBox("Parámetros de puerto y velocidad incorrectos")

    End Try

    If (Nummod < 1) Then
        MsgBox("Posicionador no encontrado")
        Exit Sub
    End If

    modtype = NmcGetModType(1)

    If (modtype <> SERVOMODTYPE) Then
        MsgBox("Posicionador no encontrado")
        Exit Sub
    End If

    MsgBox("Posicionador conectado")
```



```
ExitButton.Enabled = True
NumPosdes.Enabled = True

calibrar.Enabled = True
MoveButton.Enabled = True
Kp.Enabled = True
Ki.Enabled = True
Kd.Enabled = True
Vel.Enabled = True
aceleracion.Enabled = True
InitButton.Enabled = False
velocidad.Enabled = False
puerto.Enabled = False
End Sub

Private Sub NumPosdes_ValueChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles NumPosdes.ValueChanged

    If (NumPosdes.Value = 0) Then
        BarraSP.Value = 0
    Else
        BarraSP.Value = -1 * ((NumPosdes.Value ^ 4) * (-0.0129) +
        (NumPosdes.Value ^ 3) * (0.6812) - (NumPosdes.Value ^ 2) * (32.799) -
        (NumPosdes.Value) * (3097.5) - 5713.8)
    End If
End Sub

Private Sub ExitButton_Click_1(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles ExitButton.Click
    Me.Close()
    ExitButton.Enabled = False
    InitButton.Enabled = True
    BarraPV.Value = 0

End Sub

End Class
```



Anexo 2  
Programas desarrollados en Matlab

**Código de Matlab para graficar una nube de puntos.**

```
close all
clear all
clc
a=load ('C:\Users\Róger Sánchez\Desktop\pared0grados.txt'); % Cargar
nube de puntos
plot3 (a(:,1), a(:,2), a(:,3), '.', 'MarkerSize',2); % Graficar nube de
puntos
grid on
```

**Código de Matlab para encontrar la vecindad de los puntos correlacionados pertenecientes a  $N_0$  y a  $N_\theta$ .**

```
close all
clear all
clc

%% Mostrando toda la nube de puntos
a=load ('C:\Users\Róger Sánchez\Desktop\pared0grados.txt'); % Cargar
nube de puntos
figure(1);
plot3 (a(:,1), a(:,2), a(:,3), '.', 'MarkerSize',2); % Graficar nube de
puntos
grid on
hold on
p1=[-860;-160.6;719.4]; %Primera aproximación
plot3(p1(1,1),p1(2,1),p1(3,1), 'r+') % Graficar punto preseleccionado en
toda la nube

%% Buscando vecinos
v=500; % tamaño de la ventana de búsqueda
A=a(:,1);
B=a(:,2);
C=a(:,3);
i=1;
for j=1:size(a,1)
    if (A(j)<=p1(1,1)+v) && (A(j)>=p1(1,1)-v) && (B(j)<=p1(2,1)+v) &&
(B(j)>=p1(2,1)-v) && (C(j)<=p1(3,1)+v) && (C(j)>=p1(3,1)-v)
        b(i,:)= [A(j),B(j),C(j)];
        i=i+1;
    end
end

%% Mostrando vecindad
figure(2); hold on
plot3 (b(:,1),b(:,2),b(:,3), '.', 'MarkerSize',2) % Graficando los vecinos
del punto preseleccionado
plot3(p1(1,1),p1(2,1),p1(3,1), 'r+') % Graficar punto preseleccionado con
sus vecinos
```

**Código de Matlab para solucionar los sistemas de ecuaciones no lineales que permiten encontrar la matriz de transformación homogénea.**

```
function [sisecu]= sol(x)
%% Definición de puntos correlacionados

%Nube P'

A1p=[-850.8, -254.4, 764.5];
A2p=[-729.2, -99.92, 689.7];
A3p=[-641.9, -232.1, 716.5];
A4p=[-718.2, -236.5, 74.2];

%Nube P

A1=[-897.9, -157.6, 722.2];
A2=[-749, -16.65, 694];
A3=[-673.5, -153.8, 707.5];
A4=[-738.2, -157.9, 71.29];

%% Planteamiento del sistema de ecuaciones

sisecu(1)=A1(1)*x(1)+A1(2)*x(2)+A1(3)*x(3)+x(4)-A1p(1);
sisecu(2)=A1(1)*x(5)+A1(2)*x(6)+A1(3)*x(7)+x(8)-A1p(2);
sisecu(3)=A1(1)*x(9)+A1(2)*x(10)+A1(3)*x(11)+x(12)-A1p(3);
sisecu(4)=A2(1)*x(1)+A2(2)*x(2)+A2(3)*x(3)+x(4)-A2p(1);
sisecu(5)=A2(1)*x(5)+A2(2)*x(6)+A2(3)*x(7)+x(8)-A2p(2);
sisecu(6)=A2(1)*x(9)+A2(2)*x(10)+A2(3)*x(11)+x(12)-A2p(3);
sisecu(7)=A3(1)*x(1)+A3(2)*x(2)+A3(3)*x(3)+x(4)-A3p(1);
sisecu(8)=A3(1)*x(5)+A3(2)*x(6)+A3(3)*x(7)+x(8)-A3p(2);
sisecu(9)=A3(1)*x(9)+A3(2)*x(10)+A3(3)*x(11)+x(12)-A3p(3);
sisecu(10)=A4(1)*x(1)+A4(2)*x(2)+A4(3)*x(3)+x(4)-A4p(1);
sisecu(11)=A4(1)*x(5)+A4(2)*x(6)+A4(3)*x(7)+x(8)-A4p(2);
sisecu(12)=A4(1)*x(9)+A4(2)*x(10)+A4(3)*x(11)+x(12)-A4p(3);
sisecu(13)=x(1)*x(6)*x(11)+x(2)*x(7)*x(9)+x(3)*x(5)*x(10)-
x(2)*x(5)*x(11)-x(1)*x(7)*x(10)-x(3)*x(6)*x(9)-1;
```

**Código de Matlab para corregir las nubes de puntos capturadas con teta grados de inclinación del HDL-64E a la referencia horizontal del sensor.**

```
clear all
close all
clc
%% Matriz de transformación homogénea de 0 a teta grados calculada con
fsolve
x0=[0 0 0 0 0 0 0 0 0 0 0 0];
[x]=fsolve('sol',x0);

Mat=[x(1) x(2) x(3) x(4); x(5) x(6) x(7) x(8); x(9) x(10) x(11) x(12); 0
0 0 1]
```



```
% Transformación de nube de puntos de teta grados a 0° (referencia
horizontal)

Nubeteta=load ('C:\Users\Róger Sánchez\Desktop\cicata\cica30.txt'); %
cargar nube de puntos
Nubeteta(1:size(Nube5),4)=1; % Agregamos una cuarta columna de 1 a la
nube de puntos

% Transformando cada punto de la nube de teta grados a 0°

r= size(Nubeteta);
r= r(1,1);
for i=1:1:r
r2=Nubeteta(i,:);
r3=r2';

Nube0(:,i)= inv(Mat)*r3;

end

Nube0=Nube0';

plot3 (Nube0(:,1), Nube0(:,2), Nube0(:,3), '.', 'MarkerSize',2);
grid on
```



# Anexo 3

## Matrices de transformación Homogéneas

Matices homogéneas que representan las transformaciones 3D experimentadas al posicionar el HDL-64E a diferentes niveles de inclinación.

Matriz de transformación	Estimación de la matriz de transformación
$H_5$	$\begin{bmatrix} 0.93463178340 & -0.11949796179 & 0.02565094605 & -48.95220797592 \\ 0.08178894450 & 1.00801079049 & -0.00789780131 & -16.39547394830 \\ -0.14379689569 & -0.17364150820 & 1.02531497840 & -1.32463600138e+002 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
$H_{10}$	$\begin{bmatrix} 0.58473118350 & -0.11849894177 & 0.05575797605 & -35.55620397192 \\ 0.04176896467 & 1.03804019743 & -0.00484760161 & -12.45547374840 \\ -0.14879689669 & -0.17764550420 & 1.02221597740 & -1.894656090138e+002 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
$H_{15}$	$\begin{bmatrix} 0.78463138350 & -0.11645874177 & 0.25275394605 & -51.65220717552 \\ 0.04674896477 & 1.43507089773 & -0.60477463161 & -10.31527334860 \\ -0.5437988959 & -0.13644550230 & 1.22266887740 & -2.32567607118e+002 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
$H_{20}$	$\begin{bmatrix} 0.73465178150 & -0.51635844296 & 0.85442334835 & -62.97222793592 \\ 0.01676846277 & 1.13264086739 & -0.10373475961 & -19.19517354630 \\ -0.1469786911 & -0.23156560970 & 1.42365886647 & -2.67483928437e+002 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
$H_{25}$	$\begin{bmatrix} 0.53365678950 & -0.51665864276 & 0.65472354805 & -30.94222791572 \\ 0.31666876287 & 1.53204086779 & -0.10376473961 & -21.31517392868 \\ -0.5467788919 & -0.13146550930 & 1.42363887640 & -2.89469409131e+002 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
$H_{30}$	$\begin{bmatrix} 0.13355638920 & -0.11625834246 & 0.75482394805 & -40.35240767512 \\ 0.611236876287 & 1.53264076789 & -0.32036463971 & -15.49447597830 \\ -0.2465787912 & -0.23136540950 & 1.52343837620 & -3.00263206214e+002 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
$H_{35}$	$\begin{bmatrix} 0.14355636970 & -0.32167583247 & 0.74484394807 & -28.65220717542 \\ 0.631246876287 & 1.55266072719 & -0.36036466977 & -17.49537334840 \\ -0.2565687712 & -0.26137541959 & 1.53343834626 & -3.4467737137e+002 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
$H_{40}$	$\begin{bmatrix} 0.33356637970 & -0.52157553267 & 0.243554694876 & -66.96226792592 \\ 0.631266873285 & 1.78264074714 & -0.12034456977 & -16.36541394830 \\ -0.6575687712 & -0.34537741857 & 1.33323632677 & -3.7943700338e+002 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
$H_{43}$	$\begin{bmatrix} 0.36366667371 & -0.22167523163 & 0.443454496866 & -50.65220727522 \\ 0.098265873225 & 1.31364374314 & -0.14031451979 & -11.69547324810 \\ -0.6777687711 & -0.35536761849 & 1.13126636676 & -3.99468630538e+002 \\ 0 & 0 & 0 & 1 \end{bmatrix}$



## Anexo 4

### Publicaciones derivadas de la investigación

9º Congreso Nacional de Mecatrónica  
Octubre 13-15, 2010. Puebla, Puebla.

## Sistema mecatrónico para aumentar el ángulo de visión vertical del LIDAR HDL-64E

Sánchez Alonso Róger Ernesto, Castillo Castañeda Eduardo, González Barbosa José Joel.

Centro de Investigación en Ciencia Aplicada y Tecnología Avanzada, CICATA-IPN.  
Cerro Blanco No 141, Colinas del Cimatarío, Querétaro, Qro, C.P. 76090

### Resumen

*En este trabajo se presenta un prototipo mecatrónico que incrementa de 26.8 a 69.8° el ángulo de visión vertical del LIDAR HDL-64E, el cual es un sensor de última generación utilizado para la modelización 3D de ambientes urbanos. Para ampliar el campo de visión vertical del sensor se diseñó y manufacturó un dispositivo de un grado de libertad que permite inclinar el HDL-64E según sea necesario, inclinación que es posible a través de un mecanismo de transmisión de movimiento conocido como "triángulo con un lado de longitud variable". El preciso posicionamiento angular del sensor se obtiene gracias a la implementación de un controlador PID, el cual fue complementado con el desarrollo de una interfaz de usuario ajustada a la aplicación. Finalmente se desarrolla una metodología muy práctica para estimar las matrices homogéneas que permiten transformar una nube de puntos capturada con una determinada inclinación en otra con la referencia inicial del sensor, lo que significa obtener nubes de puntos con mayor información referente a la altura de algún escenario y lo que a su vez se traduce en un aumento del desempeño en visión vertical del LIDAR HDL-64E.*

**Palabras clave:** LIDAR HDL-64E, ángulo de visión vertical, controlador PID de posición angular, matrices homogéneas.

### 1. Introducción

Obtener modelos 3D de ciudades ha sido objeto de creciente interés en los últimos años, esto se debe principalmente a que este tipo de investigaciones poseen horizontes de aplicación muy extensos, por ejemplo en [1] se puntualiza áreas de aplicación vinculadas a la planeación urbana, la arqueología, el turismo virtual, la simulación, la restauración, etc.

Actualmente existe una diversidad de metodologías destinadas a la obtención de modelos 3D de zonas urbanas, muchas de ellas planteadas en [1] y [2], sin embargo, existe una que ha evolucionado ampliamente y que ofrece excelentes resultados, esta metodología está basada en la tecnología LIDAR (Light Detection and Ranging).

El presente trabajo está enfocado hacia el modelo LIDAR HDL-64E, el cual es un dispositivo elaborado por la compañía Velodyne y cuyo principio de funcionamiento se basa en el cálculo de la distancia entre un emisor láser y un objeto en su entorno.

El HDL-64E es un sensor que por sus especificaciones de resolución, precisión, alcance y densidad de datos entregados es considerado como de alto desempeño. Sin embargo, éste presenta un ángulo de visión vertical reducido, 2° por encima de la horizontal y 24.8 por debajo, condición que limita su funcionalidad para realizar tareas de reconstrucción de espacios urbanos, pues resulta imposible obtener escaneos completos de estructuras de altura considerable.

Al tomar en cuenta el enorme potencial del sensor para la obtención de modelos 3D es que surge el interés de habilitarlo para la realización de tareas relacionadas al escaneo de espacios urbanos, y es en ese sentido que en este trabajo se propone el diseño y manufactura de un dispositivo mecatrónico que incremente el campo de visión vertical del sensor.

El prototipo desarrollado en este estudio inclina de forma automática el HDL-64E, fijándolo en la posición angular necesaria para poder, en una serie de barridos, obtener una nube de puntos completa de estructuras que por su altura no podrían ser escaneadas por el sensor en condiciones normales.

Asociación Mexicana de Mecatrónica A.C.

Benemérita Universidad Autónoma de Puebla

8º Congreso Nacional de Mecatrónica  
Octubre 13-15, 2010. Puebla, Puebla.

## 2. Tecnología LIDAR HDL-64E

El LIDAR HDL-64E es un sensor láser de segunda generación y su principio de operación es bastante simple, obtiene las distancias de los objetos situados a su alrededor utilizando el principio del medidor láser de distancia. Un emisor láser hace incidir un rayo, el cual rebota en la superficie de un objeto y un receptor lo captura calculando la distancia de acuerdo al tiempo que tarda en regresar, si tarda más tiempo del esperado o no regresa entonces no se toma en cuenta. Con los ángulos en que se proyectó el láser y la distancia calculada se construye un vector en forma esférica, y así se obtiene un punto en el espacio  $R^3$ , acumulando cada uno de estos puntos se forma una nube de puntos (ver figura 1). Cuando esta colección de puntos es renderizada se crea una imagen tridimensional del escenario. Entre más densa sea la nube de puntos mayor resolución tendrá la imagen.

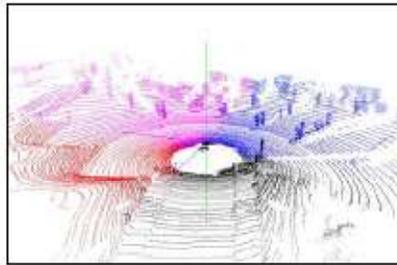


Fig. 1. Ejemplo de nube de puntos LIDAR HDL-64E.

El HDL-64E tiene montado un arreglo de 64 láseres (ver figura 2), divididos en un bloque superior y otro inferior de 32 cada uno. Toda la unidad gira alrededor del eje vertical ofreciendo un campo de visión horizontal de  $360^\circ$ , en donde se entrega cada  $0.09^\circ$  el ángulo de proyección  $\theta$  (azimut). Por otro lado, el sensor tiene un campo de visión vertical de  $26.8^\circ$ , el cual va desde  $2^\circ$  hasta  $-24.8^\circ$  debido al acomodo descendente de los láseres según el ángulo de proyección  $\phi$  (colatitud). Cada láser es disparado miles de veces por segundo, provyendo una gran cantidad de datos y en consecuencia una nube de puntos mucho más robusta que los diseños convencionales. El HDL-64E cuenta con un alcance de hasta 120 m y una precisión de 5 cm. El sensor es de clase IM, seguro para la vista, con una longitud de onda de 905 nm y con pulsos de 10 ns. Para mayor información acerca del sensor ver [3], el cual puede ser descargado de la página web del fabricante ([www.velodyne.com](http://www.velodyne.com)).

Asociación Mexicana de Mecatrónica A.C.

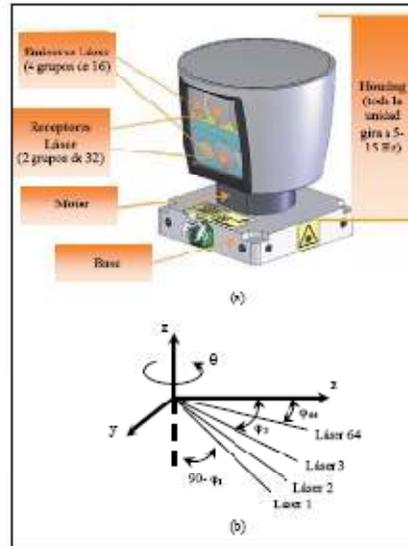


Fig. 2. (a) Posición de los emisores y receptores. (b) Ilustración de los ángulos  $\theta$  (azimut) y  $\phi$  (colatitud).

## 3. Diseño mecánico del sistema de posicionamiento

La función del prototipo diseñado es inclinar el HDL-64E hasta una posición angular capaz de visualizar una estructura de aproximadamente 10 metros de altura. Según cálculos realizados, en donde se considera que para la realización de escaneos en espacios urbanos es necesario montar el sensor en la parte superior de un automóvil y transitar sobre la calle, se estima que para poder realizar escaneos hasta 10 metros de altura es necesario inclinar el sensor aproximadamente  $43^\circ$ . Tomando en cuenta lo anterior se ha diseñado un soporte tubular que puede ser instalado sobre un vehículo con canastera y que incluye una plataforma móvil para el montaje y posicionamiento del sensor, tal y como se ilustra en la figura 3.

La plataforma móvil está elaborada en aluminio, lo que la hace muy ligera, sin embargo es lo suficientemente rígida para soportar la masa de aproximadamente 13 kg del HDL-64E. Esta plataforma se sujeta firmemente al soporte tubular a través de un par de abrazaderas ajustables y gira con mínima fricción alrededor de un eje de acero inoxidable gracias a la incorporación de dos rodamientos de agujas.

Benemérita Universidad Autónoma de Puebla

9° Congreso Nacional de Mecatrónica  
 Octubre 13-15, 2010. Puebla, Puebla.



Fig. 3. Estructura tubular y plataforma móvil.

La movilidad del sistema desde una posición inicial a 0° hasta una posición final a 43° es provista gracias a un mecanismo conocido como "triángulo con un lado de longitud variable". Como se observa en la figura 4, el movimiento se genera cuando la barra ilustrada en rojo varía en longitud en la medida que se quiere inclinar la plataforma móvil mientras las barras ilustradas en verde permanecen constantes.

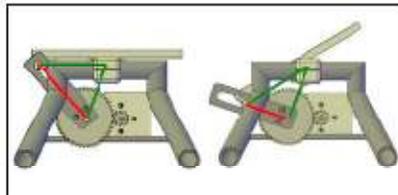


Fig. 4. Movilidad del sistema.

Si observamos con detenimiento la figura 4 es posible detectar que la plataforma móvil incluye un pequeño rodamiento que se desliza a través de una corredera para lograr un desplazamiento de mínima fricción. Otro detalle que se puede observar es la presencia de dos engranes rectos. Estos engranes forman parte de la transmisión mecánica del dispositivo (ver figura 5).



Fig. 5. Transmisión mecánica del sistema.

Asociación Mexicana de Mecatrónica A.C.

La transmisión mecánica del sistema está integrada por dos engranes MARTIN, el TS1648 20 y el TS1612 20, con una relación 4:1 y un reductor DINKERMOTOREN PI.G 425 acoplado directamente al motor con una relación de 100:1. La presencia de esta transmisión se justifica por dos razones; la primera, es que el HDL-64E es muy pesado y se necesita aumentar el torque del motor para poder manipular sus más de 13 kg, y la segunda, es que necesitamos realizar movimientos de posicionamiento lentos y suaves.

Una vez identificado el mecanismo de transmisión de movimiento el siguiente paso es estudiar su cinemática, esto significa establecer la relación que existe entre las posiciones angulares de entrada y las posiciones angulares de salida. Para el mecanismo desarrollado en este trabajo se entiende como entrada el movimiento rotacional del motor transmitido al eje del engrane más grande (TS1648 20), y como salida la posición angular de la plataforma de inclinación. En la figura 6 se ilustran todos los elementos que intervienen en el análisis cinemático.

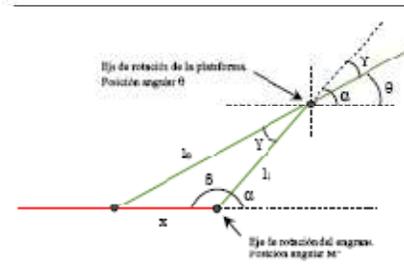


Fig. 6. Elementos para el análisis cinemático del sistema.

La ecuación que describe la relación entre las rotaciones del motor trasladadas al eje del engrane (M°) y la posición angular de la plataforma (θ) es la siguiente.

$$M^\circ = \beta - \sin^{-1} \left( \frac{\sin(\alpha - \theta) l_2}{\sqrt{l_1^2 + l_2^2 - 2l_1 l_2 \cos(\alpha - \theta)}} \right) \quad (1)$$

Donde:

- x: barra móvil de longitud variable.
- l<sub>1</sub>: barra fija de longitud constante.
- l<sub>2</sub>: barra móvil de longitud constante.
- α: ángulo constante.
- β: ángulo constante de condición inicial.

Benemérita Universidad Autónoma de Puebla

9<sup>o</sup> Congreso Nacional de Mecatrónica  
Octubre 13-15, 2010. Puebla, Puebla.

En la figura 7 se presenta el prototipo ya manufacturado, el cual se puede describir como un sistema mecánico de un grado de libertad capaz de posicionar el HDL-64E hasta un ángulo máximo de 43°, lo que unido al ángulo de visión vertical del sensor (26.8°) permite obtener un campo de visión de 69.8°.



Fig. 7. (a) Mecanismo en posición inicial a 0°. (b) Mecanismo en posición máxima a 43°.

#### 4. Implementación de control PID

En esta sección del trabajo se presenta un control PID para automatizar el posicionamiento del LIDAR.

##### 4.1 Motor y sensor de posición

El motor seleccionado para dar movilidad al sistema es un MAXON 118778. El torque máximo de este servomotor, dentro de un rango de operación recomendable, es de 0.113 Nm. Con el brazo de palanca aplicado el motor por sí solo no podría soportar los 13.1541 kg del HDL-64E, por esta razón, como se mencionó en la sección 3, se incorporó una transmisión mecánica que permite aumentar 400 veces el torque del motor (ver figura 5).

Este motor no incluye encoder, por lo que se adaptó uno de tipo incremental de la serie 102 de AMT. Dicho encoder fue configurado para una resolución de 1000 PPR.

##### 4.2 Tarjeta de control

El control del sistema se realizó a través de PIC-SERVO SC, la cual es una tarjeta muy robusta para el control PID de servomecanismos. Para mayor información acerca de la tarjeta ver [4], el cual puede ser descargado de la página web del fabricante ([www.picservo.com](http://www.picservo.com)).

##### 4.3 Hardware necesario para el control

El hardware utilizado para la implementación del control PID se detalla a continuación.

- a. Tarjeta PIC-SERVO SC.
- b. Convertidor SSA-485. Permite comunicarse con la PIC-SERVO SC a través del puerto USB.
- c. Sistema mecánico de posicionamiento HDL-64E con motor acoplado y encoder.
- d. Fuente de poder para alimentar el convertidor SSA-485 (7.5-12 VDC) y al motor (24 VDC).
- e. Cable DB15 (Motor/encoder). Se conecta a la tarjeta para llevar la señal de control hacia el motor y captar la retroalimentación del encoder.
- f. Cable plano para 10 pines con socket IDC estándar en los dos extremos. Interconecta el convertidor SSA-485 y la PIC-SERVO SC.
- g. Cable para conexión USB.
- h. Computadores con puerto USB.

#### 4.4 Interfaz de usuario

La interfaz realizada (ver figura 8) fue desarrollada en Visual Basic para aprovechar la disponibilidad de las librerías de PIC-SERVO en este lenguaje de programación. La interfaz llamada "Interfaz de control LIDAR" es fácil de utilizar y está ajustada a la aplicación de posicionamiento angular del HDL-64E.



Fig. 8. Interfaz de usuario.

La interfaz de usuario está constituida por tres secciones. La primera sección es la encargada de iniciar y finalizar la comunicación con el sistema de posicionamiento. La segunda sección sirve para ajustar las ganancias de control (proporcional, derivativa e integral) y seleccionar la velocidad y aceleración del desplazamiento de la plataforma de inclinación. Finalmente la última sección es utilizada para seleccionar el ángulo al que queremos posicionar el LIDAR HDL-64E, visualizar como se alcanza este "Set Point" y ordenar al mecanismo ir a su posición de reposo "home".

9º Congreso Nacional de Mecatrónica  
Octubre 13-15, 2010. Puebla, Puebla.

Después de afinar todos los detalles referentes al control PID, se procedió a validar la funcionalidad del sistema, para ello se realizaron una serie de pruebas de posicionamiento y escaneo. En la figura 9 (a) se presenta un edificio de 3 plantas que no podría ser escaneado por el HDL-64E bajo condiciones normales, sin embargo, utilizando el sistema de posicionamiento propuesto con dos barridos, uno a 15° y otro a 35°, podemos obtener una nube de puntos completa del edificio.

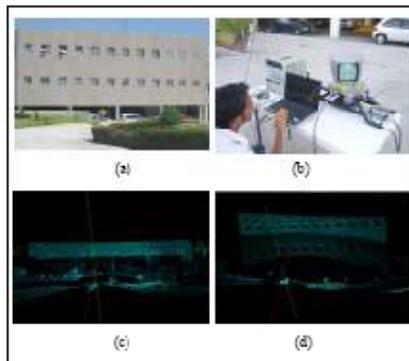


Fig. 9. (a) Edificio a escanear. (b) HDL-64E montado sobre el sistema de posicionamiento. (c) Primer barrido a 15°. (d) Segundo barrido a 35°.

### 5. Transformación de las nubes de puntos

El hecho de inclinar el HDL-64E para aumentar su campo de visión no significa que las adquisiciones del sensor sean ya útiles para realizar reconstrucción 3D. Esto se debe a que todas las adquisiciones que se hagan con el sensor inclinado experimentaran una rotación y una traslación con respecto a la posición inicial del sensor. La figura 9(d) representa el escaneo de un edificio a 35° de inclinación, y aunque no sea muy notorio esta captura ha sufrido una drástica transformación en cuanto a orientación y posición (ver figura 10).



Fig. 10. Nube de puntos capturada a 35°.

Asociación Mexicana de Mecatrónica A.C.

Existen muchos enfoques para resolver la problemática de alinear dos nubes de puntos con diferente sistema de referencia, por ejemplo [5] utiliza el algoritmo "Iterative Closest Point" para encontrar la matriz de rotación y traslación que aplicada sobre una de las dos nubes lleve al alineamiento con la otra.

En el presente trabajo para referenciar las nubes de puntos a la condición inicial (0°) se utiliza exclusivamente información geométrica.

Des nubes de puntos; P' (nube de puntos que sufrió una transformación por inclinar el sensor) y P (nube de puntos que no ha experimentado transformación) están relacionadas por una matriz homogénea (H) de la siguiente manera.

$$P' = HP \quad (2)$$

De lo anterior, un punto A' perteneciente a P' está relacionado con un punto A perteneciente a P de esta manera.

$$A' = \begin{bmatrix} R_{3 \times 3} & T_{3 \times 1} \\ 0 & 1 \end{bmatrix} * A \quad (3)$$

Esta matriz homogénea, como se menciona en [6] está compuesta por una sub-matriz de 3x3 (rotación), un vector columna de 3x1 (traslación), un vector fila nulo de 1x3 (perspectiva) y un componente unitario que representa el escalamiento.

La ecuación (3) se puede expresar como la relación de A' (X', Y', Z') y A(X, Y, Z) a través de una matriz homogénea integrada por 12 variables.

$$\begin{bmatrix} X' \\ Y' \\ Z' \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (4)$$

Si se considera que las coordenadas de los puntos A y A' son datos conocidos se puede obtener el siguiente conjunto de ecuaciones.

$$X' = aX + bY + cZ + d \quad (5)$$

$$Y' = eX + fY + gZ + h \quad (6)$$

$$Z' = iX + jY + kZ + l \quad (7)$$

Lo que se tiene ahora es un sistema de tres ecuaciones y doce incógnitas. Esto significa que con 4 pares de puntos correlacionados (A<sub>1</sub>, A'<sub>1</sub>, A<sub>2</sub>, A'<sub>2</sub>, A<sub>3</sub>, A'<sub>3</sub>, A<sub>4</sub>, A'<sub>4</sub>) se obtendría un sistema de doce ecuaciones y doce incógnitas.

Benemérita Universidad Autónoma de Puebla

9<sup>o</sup> Congreso Nacional de Mecatrónica  
 Octubre 13-15, 2010. Puebla, Puebla.

Es bien sabido que el determinante de una matriz homogénea es igual a 1, por lo que al sistema de doce ecuaciones hay que agregar otra.

$$afic + bgi + cej - bek - agj - cfi = 1 \quad (8)$$

Cuando se solucione el sistema de ecuaciones (Matlab) y se tenga la matriz homogénea toda la nube de puntos  $P'$  puede ser alineada a la referencia  $0^\circ$  de la siguiente forma.

$$\text{inv}(H)P' = P \quad (9)$$

La nube de puntos  $P$  significará ahora la nube  $P'$  pero con una referencia horizontal, ya lista para desarrollar tareas de modelado 3D.

Es importante destacar que la matriz homogénea  $H$  es válida para referenciar a  $0^\circ$  cualquier nube de puntos siempre y cuando dicha nube haya sido adquirida con el mismo ángulo de inclinación con el que se calculó la matriz. Por esta razón se calcularon las matrices homogéneas para una serie de barridos cada  $5^\circ$ , lo que significa que para alcanzar los  $43^\circ$  de máximo posicionamiento del sistema fue necesario calcular 9 matrices homogéneas.

Para garantizar el cálculo de las 9 matrices homogéneas que rigen los barridos cada  $5^\circ$  se utilizó un patrón de capturas. Este patrón fue un cubo de 1.5 m de arista. Los puntos correlacionados para formar los sistemas de ecuaciones y encontrar las matrices fueron los vértices del cubo.

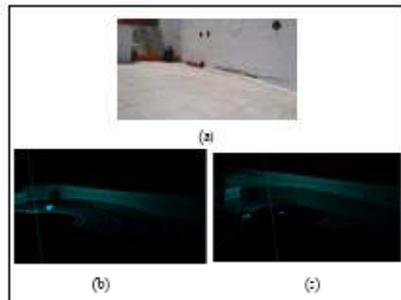


Fig. 11. (a) Pared a escanear. (b) Escaneo normal del sensor. (c) Captura a  $5^\circ$  transformada a la referencia  $0^\circ$

Una vez calculadas las nueve matrices homogéneas se puede utilizar el sistema de

posicionamiento para, a través de un único barrido o una serie de ellos, obtener escaneos con el HDL-64E y conseguir datos útiles para realizar actividades de reconstrucción tridimensional. Por ejemplo en la figura 11 (a) se muestra una pared cuya altura no podría ser visualizada por el sensor en condiciones normales, caso contrario cuando aplicamos el sistema propuesto en este trabajo (ver figura 11 (c)).

## 6. Conclusiones

Se ha desarrollado un sistema mecatrónico para ampliar de  $26.8$  a  $69.8^\circ$  el ángulo de visión vertical del LIDAR HDL-64E y en consecuencia incrementar significativamente su potencial para la realización de tareas vinculadas a la reconstrucción tridimensional de espacios urbanos.

El sistema es preciso y robusto, y está listo para ser montado sobre un vehículo y ser utilizado cuando se pretenda escanear estructuras de alturas no superiores a 10 metros.

Como trabajo futuro se contempla su aplicación para el modelado tridimensional de ciudades, así como también para el registro digital de su patrimonio arquitectónico.

## Referencias

- [1] Tunc E, Karli F, Aykan E. "3D city reconstruction by different technologies to manage and reorganize the current situation". 2004.
- [2] Bsenner C. "Interactive modeling tools for 3D building reconstruction". 1999.
- [3] Velodyne Acoustics, Inc. HDL-64E User's Manual. 2007.
- [4] Ken J. PK-SERVO Motion Control Board. 2005.
- [5] Alonso R, Gómez J, Zalama E. "Alineamiento automático de nubes densas de puntos a partir de información geométrica y cromática". 2006.
- [6] Tzai L. "Robot Analysis. The Mechanics of serial and Parallel Manipulators". John Wiley & sons. EEUU. 1999.



CIINDET 2010

VIII Congreso Internacional sobre Innovación y Desarrollo Tecnológico,  
24 al 26 de noviembre de 2010, Cuernavaca Morelos, México.

## Elaboración de mallado tridimensional a partir de datos de un LIDAR HDL-64E

R. Sánchez Alonso, E. Castillo Castañeda

**Resumen:** Este trabajo presenta un procedimiento para la elaboración de mallado tridimensional sobre nubes de puntos de escenarios físicos extraídas a través de un LIDAR HDL-64E. El procedimiento inicia con la adquisición de datos, en donde se utiliza un dispositivo mecatrónico que inclina el sensor para ampliar su campo de visión vertical, continua con la representación cartesiana de la nube de puntos, la corrección de ésta debido a la transformación experimentada al inclinar el sensor, su visualización y la supresión manual de ruido, para finalmente dar paso a la aplicación automática en el software Geomagic Studio de una malla triangular que represente la nube de puntos del escenario escaneado.

**Palabras Clave:** LIDAR, HDL-64E, nube de puntos, mallado tridimensional.

**Abstract:** This paper presents a procedure for three dimensional meshing of clouds of points that represent real scenarios obtained through an LIDAR HDL-64E. The process begins with data acquisition, where a mechatronic device is used to tilt the sensor and extend its vertical field of view, continues with the cartesian representation of the cloud of points, the correction of the latter due to the transformation experienced by tilting the sensor, display and manual noise suppression, to finally give way to the automatic application in Geomagic Studio software of a triangular mesh that represent the cloud of points of the scanned scenario.

**Keywords:** LIDAR, HDL-64E, cloud of points, three-dimensional meshing.

**Key:** Ernesto Sanchez Alonso,  
Eduardo Castillo Castañeda,  
Centro de Investigación en Ciencia Aplicada y Tecnología Avanzada,  
CICATA-IPN,  
Cerro Blanco No. 141, Colinas del Cimataro, Querétaro, Qro,  
México. C.P. 76090.  
E-mail: rogerian1984@hotmail.es

### Introducción

Dentro del gran campo de investigación de la reconstrucción tridimensional existe un área en la que se están concentrando grandes esfuerzos, esta área es la modelización de espacios urbanos. Obtener modelos 3D de ciudades ha sido objeto de creciente interés en los últimos años, esto se debe principalmente a que este tipo de investigaciones poseen horizontes de aplicación muy extensos, por ejemplo en [1] se puntualiza áreas de aplicación vinculadas a la planeación urbana, la arqueología, el turismo virtual, la simulación, la restauración, etc.

En la actualidad se dispone de numerosas fuentes de datos para la realización de modelos 3D de ciudades [1-2], sin embargo, existe una que ha evolucionado ampliamente y que ofrece excelentes resultados, esta fuente está basada en la tecnología LIDAR (Light Detection and Ranging).

La nueva tecnología LIDAR de alta definición facilita enormemente la tarea de reconstrucción tridimensional de ambientes urbanos, pues prácticamente resuelve la problemática de obtener una nube de puntos que represente el entorno, dejando básicamente pendiente la realización de actividades vinculadas a la recuperación de la continuidad de la superficie original (mallado tridimensional y texturización).

Actualmente se están desarrollando muchas investigaciones vinculadas al mallado 3D proveniente de datos escaneados, por ejemplo en [3] a través de técnicas de regularización de mallas es posible juntar nubes de puntos de diferentes objetos, y en [4] se desarrolla un sistema para editar nubes de puntos a través de subdivisión y suavizado de mallas 3D.

Este trabajo está enfocado hacia la elaboración de mallado tridimensional a partir de información de escenarios físicos escaneados con un LIDAR HDL-64E, el cual es un sensor diseñado por la compañía Velodyne para fines de navegación autónoma.



CIINDET 2010

VIII Congreso Internacional sobre Innovación y Desarrollo Tecnológico,  
24 al 26 de noviembre de 2010, Cuernavaca Morelos, México

Tecnología LIDAR HDL-64E

El LIDAR HDL-64E es un sensor laser de segunda generación y su principio de operación es bastante simple, obtiene las distancias de los objetos situados a su alrededor utilizando el principio del medidor laser de distancia. Un emisor laser hace incidir un rayo, el cual rebota en la superficie de un objeto y un receptor lo captura calculando la distancia de acuerdo al tiempo que tarda en regresar, si tarda más tiempo del esperado o no regresa entonces no se toma en cuenta. Con los ángulos en que se proyectó el laser y la distancia calculada se construye un vector en forma esférica, y así se obtiene un punto en el espacio  $R^3$ , acumulando cada uno de estos puntos se forma una nube de puntos (ver figura 1). Cuando esta colección de puntos es renderizada se crea una imagen tridimensional del escenario. Entre más densa sea la nube de puntos mayor resolución tendrá la imagen.

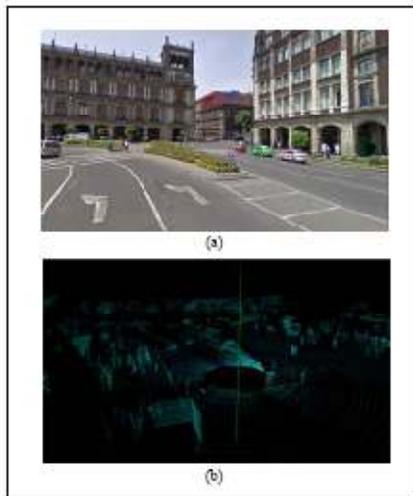


Fig. 1 (a) Escenario capturado "Plaza de la constitución ciudad de México" (aquina toroeste) (b) Nube de puntos del escenario.

El HDL-64E tiene montado un arreglo de 64 láseres (ver figura 2), divididos en un bloque superior y otro inferior de 32 cada uno. Toda la unidad gira alrededor

del eje vertical ofreciendo un campo de visión horizontal de  $360^\circ$ , en donde se entrega cada  $0.39^\circ$  el ángulo de proyección  $\theta$  (azimut). Por otro lado, el sensor tiene un campo de visión vertical de  $26.8^\circ$ , el cual va desde  $0$  hasta  $-24.8^\circ$  debido al arcamiento descendente de los láseres según el ángulo de proyección  $\phi$  (colatitud). Cada láser tiene una longitud de onda de  $905 \text{ nm}$  con pulsos de  $10 \text{ ns}$ , lo que provee una gran cantidad de datos y en consecuencia una nube de puntos mucho más robusta que los diseños convencionales. El HDL-64E es de clase 1M, seguro para la vista, cuenta con un alcance de hasta  $120 \text{ m}$  y una precisión de  $5 \text{ cm}$ .

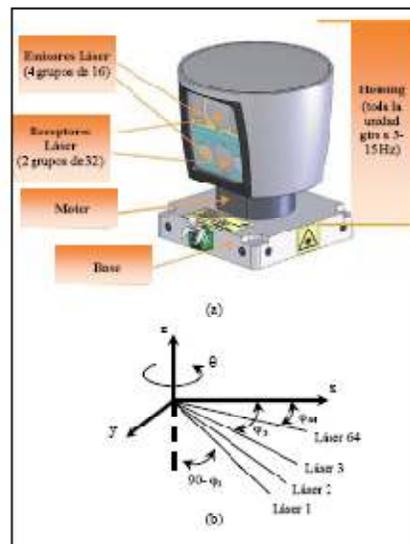


Fig. 2. (a) Posición de los emisores y receptores. (b) Ilustración de los ángulos  $\theta$  (azimut) y  $\phi$  (colatitud).

En lo que respecta al cableado, el dispositivo cuenta con su cable de alimentación, al cual debe ser conectado a  $12 \text{ Volts}$  ( $4\text{-}6 \text{ Amperes}$ ), cualquier voltaje mayor a  $16 \text{ Volts}$  puede dañar la unidad. Adicionalmente se tienen 2 interfaces: Ethernet y



CIINDET 2010

VIII Congreso Internacional sobre Innovación y Desarrollo Tecnológico,  
24 al 26 de noviembre de 2010, Cuernavaca Morelos, México

Serial, con conectores RJ-45 (transmisión de datos) y RS-232 DB9 (programación de tasa de giro) respectivamente. La cabeza rota horizontalmente y dispara los láseres por grupos ya sea superior o inferior. Para obtener un mayor desempeño a grandes distancias se emiten y capturan tres veces el grupo de láseres superiores por sólo uno del inferior (ver figura 3). Para mayor información acerca del sensor ver [5], el cual puede ser descargado de la página web del fabricante ([www.velodyne.com](http://www.velodyne.com)).

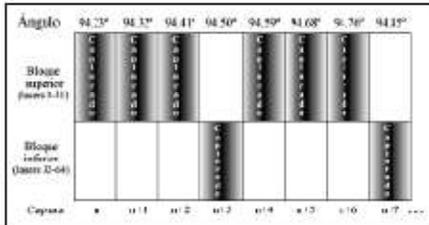


Fig. 3. Ejemplo de captura de datos.

**Metodología**

En la figura 4 se presenta un diagrama que resume la metodología seguida para la realización de mado tridimensional a partir del LIDAR HDL-64E.

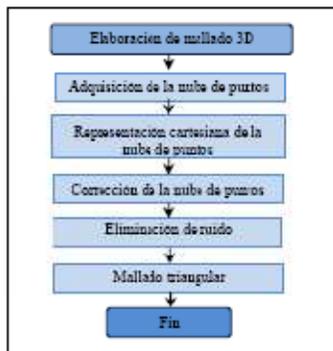


Fig.4. Metodología implementada para el mado 3D.

**Adquisición de datos para realizar el mado 3D**  
Para realizar un adecuado mado tridimensional se necesita nubes de puntos lo más regular posible, esto se debe a que los algoritmos más usados para este fin, Delaunay y Voronoi [6], ofrecen resultados poco eficientes para nubes de puntos de topología muy arbitraria.

El tipo de superficie que se busca para realizar el mado normalmente se encuentra en el suelo o, por ejemplo, en un muro. En este trabajo se decidió realizar el escaneo y mado a una pared (ver figura 5).



Fig. 5. Imagen de la pared a reconstruir.

Anteriormente se mencionó que el HDL-64E tiene un campo visión vertical de 26.3°, los cuales están 2° por encima de la horizontal y 24.8 por abajo. Esta característica muy particular de este modelo LIDAR dificultaría el escaneo completo de una pared. Para resolver este problema se utilizó un sistema mecánico (ver figura 6), el cual fue diseñado específicamente para aumentar el ángulo de visión vertical del LIDAR HDL-64E a través de la inclinación del sensor según sea necesario [7].



Fig. 6. Sistema mecánico para aumentar el ángulo de visión vertical del LIDAR HDL-64E.



CIINDET 2010

VIII Congreso Internacional sobre Innovación y Desarrollo Tecnológico,  
24 al 26 de noviembre de 2010, Cuernavaca Morelos, México

El sistema mecánico es de un grado de libertad y a través de un control PID es capaz de posicionar el sensor hasta un ángulo máximo de 43°, lo que aunado a su ángulo de visión vertical (26.8°) permite obtener un campo de visión de 69.8°. Este sistema es además muy ligero, de poca fricción durante su operación, de gran estabilidad y elevada confiabilidad para el montaje de un equipo de alta tecnología como el HDL-64E.

En la figura 7(b) se evidencia que el sensor tiene dificultades para escanear a determinada altura, sin embargo, gracias al sistema mecánico podemos obtener una simplificación significativa del desempeño del sensor (ver figura 7(c)), lo que permite escanear exitosamente una sección significativa de la pared mostrada en la figura 5.

Para adquirir la nube de puntos basta con montar el sensor sobre el sistema mecánico y ponerlo a girar. Cuando el sensor adquiere su velocidad de estabilización iniciará automáticamente a enviar paquetes UDP a través de la interfaz Ethernet. Cada uno de estos paquetes contiene la siguiente información: encabezado (indicando si es del bloque superior o inferior), Ángulo (azimut), distancia, intensidad y estado (temperatura de la unidad).

En este trabajo se utilizó un software capturador desarrollado en [8], el cual de toda la información suministrada en los paquetes UDP únicamente extrae: el Ángulo (azimut), la distancia y la intensidad, para generar archivos en un formato VLCD (propio del autor) cuyo tamaño estará en dependencia de la cantidad de puntos que se logre escanear.

Visualización y conversión de datos.

En [9] se desarrolló un visualizador de nubes de puntos (ver figura 7(b) y 7(c)) que toma en cada captura (n) el ángulo  $\theta$  (azimut), las distancias (r) y los ángulos de proyección de cada laser  $\phi$  (colatitud) para graficarlos en coordenadas esféricas y obtener una visión panorámica de todo el escenario.

Para aplicaciones de reconstrucción 3D trabajar en coordenadas esféricas resulta algo complicado, además el software utilizado para elaborar el mallado 3D, Geomagic Studio, no utiliza este formato de datos, por estas razones es necesario transformar las coordenadas esféricas a coordenadas cartesianas.

Convertir las coordenadas esféricas entregadas por el HDL-64E a coordenadas cartesianas (ver figura 8) es posible a través de las siguientes ecuaciones:

$$r_x = r \times \cos \theta \times \sin \phi \quad (1)$$

$$r_y = r \times \sin \theta \times \sin \phi \quad (2)$$

$$r_z = r \times \cos \phi \quad (3)$$

Se desarrolló un programa en Matlab que aplica las ecuaciones anteriores y entrega la posición en X, Y y Z de cada uno de los puntos de la nube.

Con la nube de puntos en forma cartesiana es posible desplegarla en Matlab o importarla desde Geomagic Studio para analizar su geometría (ver figura 9).

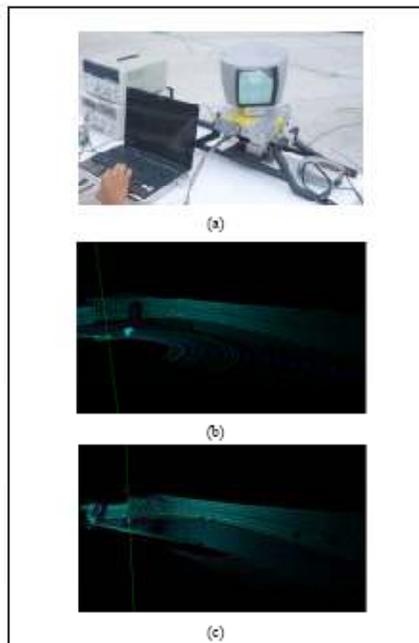


Fig. 7. (a) HDL-64E montado sobre el sistema mecánico. (b) Nube de puntos a 0°. (c) Nube de puntos con HDL-64E inclinado 15°.

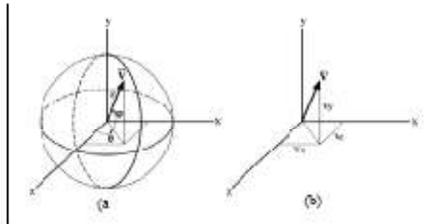


Fig. 8. Representación del vector  $V$  en (a) forma polar y (b) forma cartesiana.

**Corrección de las nubes de puntos**

El hecho de inclinar el HDL-64E para aumentar su campo de visión no significa que las adquisiciones del sensor sean ya útiles para realizar mapeo 3D. Esto se debe a que todas las adquisiciones que se hagan con el sensor inclinado experimentarán una rotación y una traslación con respecto a la posición inicial del sensor. La figura 7(c) representa el escaneo de una pared a 15° de inclinación, y aunque no sea muy notorio esta captura la sufrió una distinta transformación en cuanto a posición y orientación.

En el presente trabajo para corregir las nubes de puntos a la condición inicial (0°) se utiliza exclusivamente información geométrica [7].

Dos nubes de puntos;  $P'$  (nube de puntos que sufrió una transformación por inclinar el sensor) y  $P$  (nube de puntos que no la experimentó transformación) están relacionadas por una matriz homogénea ( $H$ ) de la siguiente manera.

$$P' = HP \quad (4)$$

De lo anterior, un punto  $A'$  perteneciente a  $P'$  está relacionado con un punto  $A$  perteneciente a  $P$  de esta forma.

$$A' = \begin{bmatrix} R_{3 \times 3} & T_{3 \times 1} \\ 0 & 1 \end{bmatrix} * A \quad (5)$$

Esta matriz homogénea está compuesta por una submatriz de 3x3 (rotación), un vector columna de 3x1 (traslación), un vector fila nulo de 1x3 (perspectiva) y un componente unitario que representa el escalamiento.

La ecuación (5) se puede expresar como la relación de  $A'$  ( $X', Y', Z'$ ) y  $A$  ( $X, Y, Z$ ) a través de una matriz homogénea integrada por 12 variables.

$$\begin{bmatrix} X' \\ Y' \\ Z' \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (6)$$

Si se considera que las coordenadas de los puntos  $A$  y  $A'$  son datos conocidos se puede obtener el siguiente conjunto de ecuaciones.

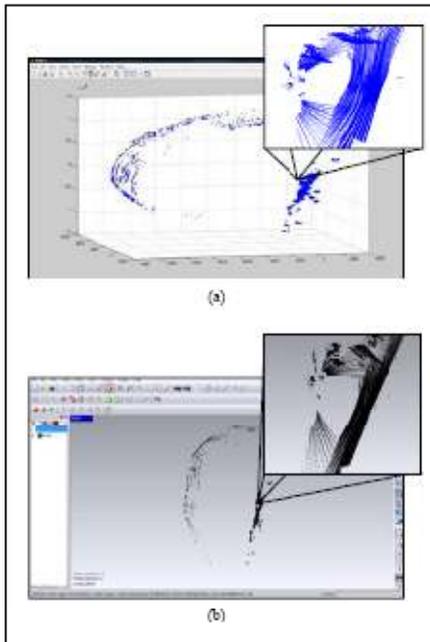


Fig. 9. Visualización de la pared con zoom en (a) Matlab y (b) Geographic Studio.



CIINDET 2010

VIII Congreso Internacional sobre Innovación y Desarrollo Tecnológico,  
24 al 26 de noviembre de 2010, Cuernavaca Morelos, México

$$X' = aX + bY + cZ - d \quad (7)$$

$$Y' = eX + fY + gZ + h \quad (8)$$

$$Z' = iX + jY + kZ + l \quad (9)$$

Lo que se tiene ahora es un sistema de tres ecuaciones y doce incógnitas. Esto significa que con 4 pares de puntos correlacionados ( $A^1_1, A^1_2, A^1_3, A^1_4$  y  $A_1, A_2, A_3, A_4$ ) se obtendría un sistema de doce ecuaciones y doce incógnitas.

Es bien sabido que el determinante de una matriz homogénea es igual a 1, por lo que al sistema de doce ecuaciones hay que agregar otra.

$$afk + bgi + cej - bek - agj - cfi = 1 \quad (10)$$

Cuando se solucione el sistema de ecuaciones (Matlab) y se tenga la matriz homogénea toda la nube de puntos  $P'$  puede ser alineada a la referencia  $0^\circ$  de la siguiente forma.

$$\text{inv}(H)P' - P \quad (11)$$

La nube de puntos  $P$  significará ahora la nube  $P'$  pero con una referencia horizontal, ya lista para desarrollar tareas de mado 3D (Ver figura 10).



Fig. 10. Nube de puntos capturada a 15° corregida.

**Elaboración de la malla**

Antes de realizar el mado tridimensional a la superficie deseada es necesario llevar a cabo una etapa de eliminación de ruido. Se considera ruido todos los puntos que no forman parte de la pared, es decir suelo, árboles, etc.

La supresión del ruido se realizó de forma manual en el software Geomagic Studio, de manera que al final se obtuviera únicamente la información de interés, la cual corresponde a los puntos que forman la pared (ver figura 11).

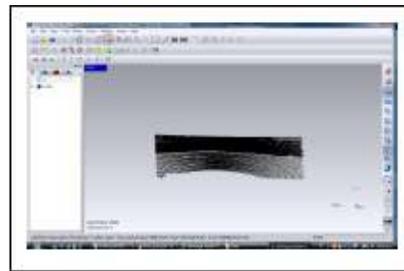


Fig. 11. Nube de puntos que representa la pared a malar.

Una vez que se tiene la nube de puntos de interés se utiliza una de las muchas herramientas de Geomagic Studio; "WRAP". Esta herramienta permite aplicar un mado a la nube de puntos para transformarla en una superficie con una textura determinada (ver figura 12).

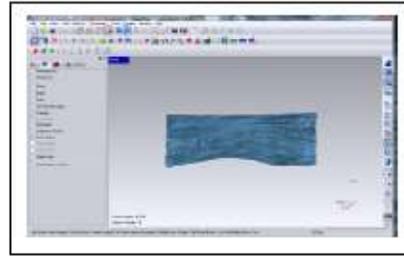


Fig. 12. Mado de la nube de puntos con textura simple.

A simple vista este mado es completo a lo largo de toda la nube de puntos, sin embargo, existen pequeños espacios que no son tomados en cuenta, esto se debe a que la nube de puntos no es tan regular como aparenta. Esos pequeños espacios se hacen notorios si se habilita la herramienta "HOLES", la cual evidencia en color



rojo la presencia de estas pequeñas figuras (ver figura13).

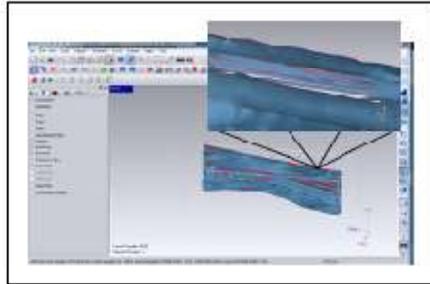


Fig. 13. Presencia de espacios no mallaados en el modelo 3D.

Es importante destacar que estas zonas del mallaado pueden ser reparadas manualmente utilizando la herramienta "FILL HOLES".

El mallaado que se realizó sobre la nube de puntos corresponde a una triangulación no óptima (ver figura 14), la cual está compuesta por 60,166 triángulos.

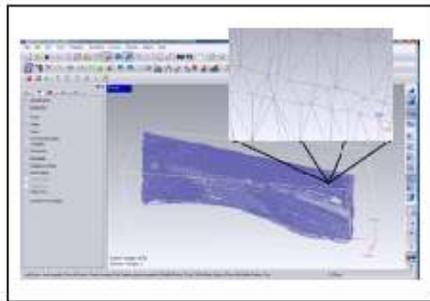


Fig. 14. Malla de 60,166 triángulos

Las tomas que se ha presentado hasta ahora del modelo 3D generado dan la impresión de una superficie algo rugosa; pero relativamente plana sin embargo, esto no es totalmente cierto. El modelo generado no es completamente plano, de hecho desde ciertos ángulos se evidencia una especie de grosor en la pared.

Este ligero grosor en el modelo no es debido a fallas en el algoritmo de triangulación que implementa el software, en realidad es debido a pequeñas deficiencias en la precisión del HDL-64E. Estas imprecisiones hacen que en lugar de tener una nube de puntos que represente una superficie plana y uniforme, como la pared, se tenga una nube de puntos de grosor zigzagante, es decir una nube de puntos con información por delante y por detrás del plano real que representa la pared. Una nube de puntos tan ruidosa genera un modelo 3D en donde se aprecia en ciertas zonas un volumen en lugar de una superficie.

#### Conclusiones

Se ha desarrollado un procedimiento para acondicionar el sensor LIDAR HDL-64E para la realización de tareas relacionadas a la reconstrucción tridimensional de escenarios físicos.

El sistema está listo para ser montado sobre un vehículo e iniciar a escanear entornos urbanos para obtener un modelo 3D de los mismos.

Como trabajo futuro se plantea desarrollar esfuerzos para el acondicionamiento de filtros que permitan mejorar las capturas del HDL-64E, lo que a su vez redundará en modelos 3D de mayor calidad.

#### Referencias

- [1] F. Tunc, F. Kardi, F. Ayhan. "3D city reconstruction by different technologies to manage and reorganize the current situation". 2004.
- [2] C. Beuzner. "Interactive modeling tools for 3D building reconstruction". 1999.
- [3] W. Sawdon. "MAPS: Multiresolution Adaptive Parameterization of Surfaces". 2001.
- [4] D. Zorn. "Interactive Multiresolution Mesh Editing". 2011.
- [5] Velodyne Acoustics, Inc. *HDL-64E User's Manual*. 2007.
- [6] B. Delaunay. "Sur la sphere vide". Bulletin de L'Academie des Sciences de L'URSS. 1934.
- [7] R. Sánchez, E. Castillo, J. González. "Sistema mecatrónico para orientar el ángulo de visión vertical del LIDAR HDL-64E". Centro de Investigación en Ciencia Aplicada y Tecnología Avanzada, CICATA-IPN Unidad Querétaro. 2010.
- [8] J. Rico. "Construcción de mapas 3D a partir de la extracción de primitivas geométricas obtenidas de datos de un LIDAR". Centro de Investigación en Ciencia Aplicada y Tecnología Avanzada, CICATA-IPN Unidad Querétaro. 2005.



**CIINDET 2010**

*VIII Congreso Internacional sobre Innovación y Desarrollo Tecnológico,  
24 al 26 de noviembre de 2010, Cuernavaca Morelos, México*

---

**Curriculo corto de los autores**

Eduardo Castillo Castañeda: es Ingeniero Mecánico Electricista egresado de la Universidad Nacional Autónoma de México en 1987. Otruvo el grado de Doctor en Control Automático en el Institut National Polytechnique de Grenoble, Francia en 1994. Actualmente es profesor-investigador de tiempo completo en el Centro de Investigación en Ciencia Aplicada y Tecnología Avanzada del Instituto Politécnico Nacional.

Róger Ernesto Sánchez Alonso: es Ingeniero Industrial graduado con honores de la Universidad Nacional de Ingeniería, Nicaragua en 2007. Actualmente es estudiante de maestría en el Centro de Investigación en Ciencia Aplicada y Tecnología Avanzada del Instituto Politécnico Nacional bajo autorización de su universidad de formación en donde labora como docente de tiempo completo.