UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO

Centro Universitario UAEM Texcoco

Navegación Autónoma de un Vehículo Pequeño en Interiores Empleando Visión Artificial y Diferentes Sensores

Anghello Arturo Buendía Ríos
Maestría en Ciencias en Computación

TESIS

Que para obtener el grado de

Maestro en Ciencias de la Computación

Presenta

Anghello Arturo Buendía Ríos

Tutor Académico	Tutores Adjuntos
Dr. Farid García Lamont	Dr. Jair Cervantes Canales M. en C. Yedid Erandini Niño Membrillo

Texcoco, Estado de México, diciembre de 2017

Universidad Autónoma del Estado de México
Centro Universitario UAEM Texcoco

DICTÁMEN DE AUTORIZACIÓN Y OBTENCIÓN DE GRADO DE MAESTRÍA

Texcoco, Méx. , a 24 de noviembre de 2017

Titulo del proyecto:

Navegación Autónoma de un Vehiculo Pequeño en Interiores empleando Visión Artificial y Diferentes Sensores

Tesista:

Anghello Arturo Buendia Rios

Dictamen:

No. de revisión: 2

- [ ] Rechazado
- [ ] Sujeto a modificaciones
- [ ] Aceptado, condicionado
- [X] Aceptado

POSGRADO
TITULACION
RECIBIDO
Por Adriana Arenas
Texcoco, Méx., 24 de 11 del 2017

Observaciones generales:

Aceptado para la impresión

Aceptado para la defensa de grado

Tutor Adjunto	Tutor Académico	Tutor Adjunto
Dr. en C. Jair Cervantes Canales	Dr. en C. de la Comp. Farid García Lamont	M. en C. C. Yedid Erandini Niño Membrillo

Contenido

Índice de figuras

Resumen

En este trabajo se presenta una propuesta para la navegación en interiores de un robot pequeño, con arquitectura de vehículo con ruedas. Para la navegación autónoma se implementa un algoritmo de planeación de trayectorias en donde se toma información del entorno para reconocimiento de objetos mediante técnicas de visión artificial y sensores de proximidad para medición y cálculo de la distancia entre el robot y los posibles obstáculos encontrados, de esta forma se recalcula la trayectoria necesaria para evitar choques.

Por otra parte, se trabaja la optimización del tiempo y seguridad de navegación, ajustando la velocidad en función de la rugosidad del suelo, a saber: 1) avance rápido en superficies de alta rugosidad, 2) avance lento en superficies de baja rugosidad, posibilitando la disminución del deslizamiento de las ruedas y mejorando el cálculo, por odometría, de la posición del robot dentro de su entorno.

Se presentan las técnicas de extracción de características y las arquitecturas y tipos de redes neuronales artificiales empleados tanto para el reconocimiento de objetos como de los tipos de texturas. Se muestran los resultados obtenidos al realizar pruebas de navegación en diferentes entornos en interiores en donde se presentan tanto los tiempos de recorrido como la distancia entre la posición deseada y la posición final del robot. Las pruebas se realizan en la plataforma robótica LEGO Mindstorm EV3.

1 Introducción

La navegación de vehículos en entornos desconocidos es una tarea fácil para los humanos, pero es muy difícil para los robots. Dotarlos de autonomía es un tema que ha sido estudiado ampliamente, pero existen varios aspectos que faltan por estudiar. Para la navegación autónoma, el vehículo debe ser dotado con la habilidad de interpretar la información adquirida de su entorno para que este pueda planear y seguir las trayectorias desde su posición inicial hasta la posición final.

Los algoritmos de navegación o de exploración de vehículos autónomos se han concentrado en el reconocimiento y evasión de obstáculos tales como piedras, cúmulos de piedra [1] y otros vehículos [2]. En los sistemas de evasión de obstáculos, estos son reconocidos por imágenes GPS [3], o por los sistemas de sensores del vehículos [4], [5].

Por otra parte, el control de la velocidad del robot considerando las características del suelo, ha sido poco trabajado, por lo que la navegación eficiente y segura puede convertirse en una debilidad

En el caso de los vehículos con ruedas, es necesario tener información sobre las características dela superficie tal que la navegación autónoma sea segura. Una de esas características en que este trabajo se concentra es en la rugosidad de la superficie en la cual el vehículo se mueve. La velocidad del vehículo depende de la rugosidad del suelo; además, la integridad del vehículo depende de la velocidad con la que se mueva. Es probable que el vehículo sufra un accidente si se mueve a la misma velocidad en tierra suelta a como si lo hace sobre asfalto, de aquí que es conveniente ajustar la velocidad en función del tipo de suelo [6].

La navegación autónoma de vehículos es particularmente importante para la exploración de planetas. Los terrenos difíciles como aquellos que tienen rocas, pendientes, tierra suelta, similares a los de planetas como Marte, requieren que los robots tengan alto grado de autonomía. De aquí que nuestra propuesta se concentra en la detección de objetos, pero también en el reconocimiento del tipo del suelo utilizando visión artificial.

1.1 Definición del problema

La navegación autónoma de vehículos es altamente compleja. Requiere de la detección de objetos para evitar choques, así como la adquisición de información del terreno para evitar deslizamientos, la cual necesita ser procesada rápidamente y con precisión por el sistema de navegación del

vehículo. Por otra parte, cuando se requiere intervención de los humanos esta no siempre puede estar disponible al momento, por lo que el vehículo debe estar equipado con las reacciones y el comportamiento ante circunstancias poco comunes. Todas estas acciones son tareas que demandan recursos computacionales. Sin embargo, para la navegación autónoma, la combinación de las acciones y de capacidad de cómputo limitada debe ser armonizada; este es el reto a vencer.

Los trabajos relacionados de navegación autónoma se concentran principalmente en la detección y evasión de obstáculos [7], [8], [9]. Hasta el momento no se ha abordado exhaustivamente el problema de navegación autónoma donde la velocidad sea ajustada de acuerdo con las características de la superficie.

El reconocimiento de texturas no es un tema que haya sido estudiado ampliamente para aplicaciones de navegación autónoma. Existen algunos trabajos en los que se presentan métodos de reconocimiento de terrenos, como los métodos basados en rayos laser [10] y vibraciones [11], los cuales han mostrado alto desempeño. Con los métodos basados en vibraciones el vehículo clasifica las texturas hasta que mueve sobre estas, mientras que los métodos con rayos láser son computacionalmente caros debido al intenso procesamiento de la gran cantidad de información obtenida del entorno.

Por otra parte, durante la navegación autónoma del vehículo, este debe determinar su ubicación dentro del entorno, lo que supone el reto de determinar las coordenadas y orientación sobre un plano o sistema de dos dimensiones. Para realizar ésta tarea cuando se trata de los vehículos con ruedas, la odometría es comúnmente empleada [12], pero inevitablemente acumula errores debido a distintas causas como la existencia de diferencias entre los diámetros de las ruedas o una mala alineación de las mismas, deslizamientos, entre otros [13].

En varios trabajos se ha intentado reducir el error odométrico al modelar la desviación obtenida de la lectura de los encoders acoplados a las ruedas, para mejorar el cálculo de la ubicación del vehículo. De acuerdo con los resultados, se aprecian reducciones significativas de dicho error, pero mediante el uso de otros dispositivos auxiliares como marcas [14], rayos laser [15], visión artificial [16], entre otros [17].

Hasta el momento, la reducción del deslizamiento de las ruedas no ha sido atendida con el fin de mejorar el cálculo de la ubicación del vehículo utilizando odometría. Por lo tanto, los problemas a los que se enfrenta este trabajo son:

- El reconocimiento de los objetos que obstaculicen la navegación del vehículo, utilizando visión artificial y sensores de proximidad.
- El ajuste automático de la velocidad durante la navegación.
- La reducción del deslizamiento de las ruedas para disminuir el error odómetrico.

Básicamente las texturas que se deben reconocer son aquellas que pueden encontrarse en interiores.

1.2 Propuesta de solución

Para el ajuste de velocidad del vehículo en función de la rugosidad de la superficie, la propuesta consiste en imitar el comportamiento humano, el cual se vale de estimaciones rápidas e imprecisas sobre las características de la superficie, pero lo suficiente precisas para ajustar la velocidad de los vehículos y conducir sin sufrir derrapes [1]. Los seres humanos clasifican rugosidades tomando como base experiencias pasadas; cuando un conductor humano encuentra una textura nueva, este emplea su experiencia para estimar el grado de rugosidad de la textura y así decidir la velocidad apropiada del vehículo.

La lógica difusa permite modelar heurísticas abstractas definidas por expertos humanos. Sin embargo, las reglas de la lógica difusa carecen de mecanismos de autoajuste en la relación subyacente de los datos, por lo que es difícil decidir los parámetros de las funciones de pertenencia. Por otra parte, la lógica difusa permite incorporar la experiencia de los humanos en forma de reglas lingüísticas. Por lo tanto, lo que se propone es emplear una red neuronal difusa [18] para calcular la velocidad del vehículo, imitando la experiencia de un humano en la conducción de autos.

Para el cálculo de la posición del vehículo, se utilizan comúnmente los métodos basados en odometría, cuya limitación es, como se ha mencionado, el deslizamiento de las ruedas. Al ajustar la velocidad en función de la rugosidad de la superficie, el deslizamiento disminuye y en consecuencia el cálculo de la posición es más exacto. De aquí que la solución propuesta es:

- Adquirir imágenes de diferentes texturas y modelarlas con patrones binarios locales [19] y descriptores de Haralick [20].
- Adquirir imágenes de diferentes objetos en un entorno determinado utilizando visión basada en apariencias [21], [22].

- Una red neuronal supervisada es entrenada para clasificar las texturas, bajo las caracterizaciones mencionadas.
- Una red neuronal difusa meta-clasifica las texturas para calcular la velocidad del robot.
- Reducir el deslizamiento de las ruedas con el ajuste de velocidad para mejorar el cálculo de la ubicación del vehículo.
- Una red neuronal supervisada es entrenada para reconocer los objetos que pueda encontrar en un entorno previamente definido.
- Implementar el algoritmo de *wavelength* para la planeación de trayectorias.

La Figura 1.1 muestra el diagrama de bloques de la propuesta de solución y como la información fluye entre los bloques.

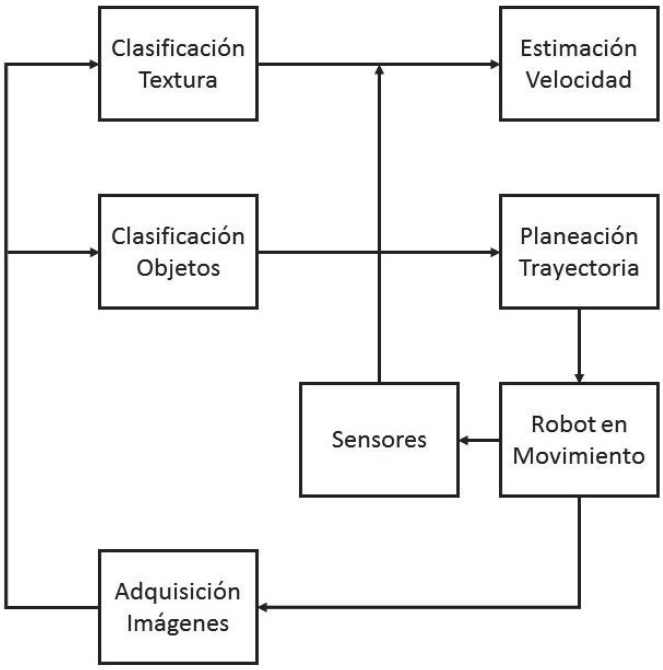

Figura 1.1 Diagrama de bloques de navegación autónoma con ajuste de velocidad.

Para la estimación de la velocidad se utiliza información de la textura de la superficie y de la proximidad de los objetos. El vehículo se mueve siguiendo la trayectoria planeada al tiempo que se capturan imágenes de los objetos que hay en el entorno y de las texturas de la superficie y esta información es procesada para ajustar la velocidad.

Al detectar superficies de baja tracción, el vehículo debe moverse lento, aunque le tome mayor tiempo realizar el recorrido, de esta forma evita que el vehículo resbale. De forma opuesta, si la

superficie tiene alta fricción, entonces el vehículo debe incrementar su velocidad, lo que lleva a reducir el tiempo de navegación, pero sin descuidar las condiciones de seguridad.

Por lo tanto, al reducir el derrape de las ruedas, se reducen los errores de posición del vehículo empleando odometría. Se contempla que el vehículo alcance determinada velocidad tal que, dependiendo del tipo de textura, le permita frenar y detenerse rápidamente. En este trabajo se realizan experimentos utilizando un robot pequeño dotado con ruedas, el cual se hace circular en interiores.

1.3 Objetivos

Objetivo general:

Desarrollar un algoritmo para la navegación de un robot con ruedas en interiores, reconociendo los objetos de su entorno, empleando visión artificial y diferentes sensores de proximidad.

Objetivos específicos

- Desarrollar un algoritmo de visión artificial para el reconocimiento de objetos específicos.
- Reducir los errores de ubicación debido a la odometría al obtener la función de error de la posición de los motores que impulsan las ruedas.
- Incorporar información adquirida por los sensores de proximidad.
- Implementar un algoritmo de navegación autónoma.
- Implementar los algoritmos desarrollados en la plataforma del robot LEGO.

2 Estado del arte

La navegación autónoma de vehículos ha sido un tema ampliamente estudiado ya que tiene gran campo de aplicación tanto para la industria automotriz como en las misiones de exploración de planetas, donde se requiere emplear robots con alto grado de autonomía para superar las dificultades del desplazamiento a través de las superficies con obstáculos e irregularidades. En las misiones de exploración de la tierra, los robots autónomos exploran terrenos que pueden ser peligrosos para los seres humanos.

La navegación autónoma es muy compleja ya que requiere de la detección y evasión de obstáculos, así como la extracción de información de las características del terreno para no sufrir derrapes. La información del entorno debe ser procesada con precisión y rapidez por el sistema de navegación del vehículo. Además, cuando no exista información disponible por parte de conductores humanos, el vehículo autónomo debe estar equipado con reacciones convenientes, particularmente frente a circunstancias poco predecibles.

Por lo tanto, el éxito de la navegación autónoma demanda seleccionar la arquitectura adecuada del robot [5], los métodos para detección y evasión de obstáculos [23], [24]y el control de la velocidad [25], [26]. Más allá de la ubicación y evasión de los obstáculos, el control de la velocidad del robot —considerando las características de los terrenos— ha sido poco atendido, lo que se vuelve en una debilidad para que la navegación autónoma sea eficiente y segura.

En este capítulo se revisan trabajos relacionados a la navegación autónoma de vehículos con ruedas, se presentan las arquitecturas más empleadas y los métodos de reconocimiento de objetos y de texturas.

2.1 Navegación autónoma de vehículos con ruedas

La rueda ha sido por mucho el mecanismo de locomoción más popular en robots y en general por vehículos hechos por el hombre ya que se puede alcanzar gran eficiencia en el transporte con este mecanismo además de ser fácil de implementar [13].

Además, el balance no es un problema usual porque los vehículos con ruedas son casi siempre diseñados para que todas las ruedas estén en contacto con el suelo todo el tiempo, en este sentido, tres ruedas son suficientes para garantizar balance estable, aunque vehículos con dos ruedas

pueden ser también estables [27]. En diseños de más de tres ruedas, es necesario utilizar un sistema de suspensión para que lograr que todas las ruedas hagan contacto con el suelo cuando se encuentra una superficie irregular [28].

A diferencia de los automóviles, los cuales están diseñados para entornos ampliamente estandarizados (como en redes carreteras), los robots móviles están diseñados para aplicaciones en diferentes situaciones. Los automóviles tienen configuraciones de rueda similares porque existe una región en el espacio de diseño que maximiza la manejabilidad, controlabilidad y estabilidad para dicho entorno estandarizado. Sin embargo, no existe una sola configuración de rueda que maximice estos aspectos para la variedad de entornos que enfrentan los diferentes tipos de robots [29].

El modelado y reconocimiento de terrenos es un tema de interés para la navegación autónoma de vehículos ya que es necesario para mantener la trayectoria de navegación en la exploración o recorrido del terreno, pero también para calcular la ubicación del vehículo y las trayectorias de navegación como una función de las características del terreno.

E. Krotkov y R. Hoffman [30] presentan un sistema de mapeo de terrenos para robots que construye modelos cuantitativos de la geometría de la superficie. El sistema adquiere imágenes con un telémetro de rayo láser y las preprocesa y almacena para construir un mapa con elevaciones a una resolución arbitraria.

La geometría del terreno es estudiada por Castelnovi y cols. [31] para controlar la velocidad del robot por medio de operaciones a distancia. La textura de la superficie es considerada como un parámetro de seguridad de la navegación. El objetivo principal es entrenar personas sin experiencia para controlar directamente la plataforma robótica mientras la misión y la integridad del robot se mantienen seguras.

Un método que emplea aprendizaje supervisado y permite al vehículo estimar la rugosidad para navegación a altas velocidades es presentado por Stavens y Thrun [32]. En su trabajo, el vehículo aprende a detectar terrenos de alta rugosidad mientras se encuentra en movimiento. Los datos de entrenamiento se obtienen por el análisis inercial adquirido del vehículo, lo que permite predecir la textura del terreno.

Los algoritmos de detección de obstáculos y de construcción de mapas expuestos por Langer y cols. [10], son empleados para navegación a cross-country. Cuando los algoritmos son utilizados

bajo la arquitectura basada en comportamiento, estos son capaces de controlar al vehículo a velocidades más altas que un sistema que planea un camino óptimo a través de mapas de alta resolución.

En Brooks y Lagnemma [9] utilizan un método para la clasificación de terrenos basado en vibraciones para misiones de exploración de planetas. Un acelerómetro es montado en la estructura del vehículo y las vibraciones detectadas en movimiento son clasificadas de acuerdo con la similitud entre las vibraciones observadas durante el entrenamiento fuera de línea. El algoritmo emplea técnicas de procesamiento de señales, incluyendo componentes principales y análisis discriminante lineal.

Conociendo las características físicas del terreno circundante, el vehículo explorador puede explotar su potencial de movilidad. Halatci y cols. [24] presentan un estudio de clasificación de terrenos para vehículos exploradores de terrenos similares a Marte, que emplea dos algoritmos de clasificación de características de textura y color basados en estimaciones probabilísticas y máquinas de soporte vectorial. Además, se describe otro método de clasificación basado en vibraciones, derivado de interacciones rueda-terreno.

Castelnovi y cols. [31] reportan resultados de navegación de robots y vehículos autónomos. El reconocimiento de texturas se realiza al muestrear el terreno en tiempo real mediante rayos láser; el objetivo es controlar la velocidad del robot dependiendo de las características capturadas de la superficie. La desventaja con este enfoque es el procesamiento intensivo de la gran cantidad de datos adquiridos, lo que vuelve computacionalmente cara esta propuesta.

Un método de comparación estadística de apariencia de imágenes es descrito por Cootes y cols. [33]. Un conjunto de parámetros de forma y variaciones en escala de grises es aprendido del conjunto de entrenamiento. Se emplea un algoritmo iterativo de comparación con el fin de aprender la relación entre los parámetros con ruido del modelo y de los errores inducidos.

Tsai y Tseng [34] proponen un sistema de visión artificial para la clasificación de la textura de superficies de moldes de metal. El método evalúa la calidad de la superficie del molde basado en la transformada discreta de Fourier de dos dimensiones, tanto en escala de grises como en imágenes binarias. Utiliza un clasificador Bayesiano y una red neuronal artificial para la clasificación de acuerdo con la discriminante principal derivado en el dominio del espacio-frecuencia.

Para abordar el reconocimiento de rugosidades de terrenos, algunos enfoques identifican, cuantifican y localizan obstáculos [2], [35]; evaluando si se pueden atravesar o si necesitan ser rodeados. Otros trabajos similares enfatizan el análisis y clasificación de la rugosidad del suelo para integrarlo en la tarea de navegación autónoma del robot, particularmente en localización.

Los métodos de reconocimiento de terrenos con mejor desempeño son aquellos basados en vibraciones y en rayos láser. Los primeros tienen la desventaja de que pueden clasificar las texturas del terreno hasta que el vehículo pasa sobre él y no antes, lo cual es un factor limitante para el ajuste de velocidad en función de las características del terreno.

Los métodos basados en rayos láser son computacionalmente caros debido al intenso procesamiento de la gran cantidad de datos que obtiene del terreno. De aquí que, en los siguientes capitulo se propone un método para reconocer terreno sin hacer contacto y a bajo costo computacional.

2.2 Reconocimiento de objetos por medio de visión basada en apariencias

Existen diversos métodos para el reconocimiento de objetos [36], [37], [33]; aunque ninguno es universal, pues aquellos encontrados en el estado del arte son funcionales si en las imágenes se cumplen determinadas condiciones. Para el propósito de este trabajo, se busca reconocer objetos que se encuentran comúnmente en interiores como son, sillas, mesas, libreros, entre otros.

La visión basada en apariencias (VBA) es un método que ha sido empleado principalmente para el reconocimiento de objetos [38], [22]y rostros [21]; el cual obtiene las componentes principales de una distribución de imágenes, es decir, los vectores propios de la matriz de covarianza de un conjunto de imágenes de objetos [39].

La VBA tiene la ventaja de reducir la dimensionalidad de los datos con una pérdida pequeña de información, lo que lo vuelve adecuado para el procesamiento de imágenes de alta resolución a un costo computacional relativamente bajo. El reconocimiento de objetos utilizando VBA conlleva las siguientes operaciones:

1. Sea $\{I_1, \dots, I_N\} \subset \mathbb{R}^{n \times m}$ el conjunto de entrenamiento de imágenes de objetos, donde n y m son el número de renglones y columnas de las imágenes, respectivamente y N es la cantidad de imágenes de entrenamiento.

2. Cada imagen del conjunto de entrenamiento es convertido un vector y se coloca en el conjunto $\{\phi_1, \dots, \phi_N\} \subset \mathbb{R}^{n \cdot m}$.
3. Todos los vectores son normalizados con $\tilde{\phi}_i = \phi_i / \|\phi_i\|$ y se colocan en el conjunto $\{\tilde{\phi}_1, \dots, \tilde{\phi}_N\} \subset \mathbb{R}^{n \cdot m}$.
4. Se calcula el vector promedio $\mu = \sum_{i=1}^{N} \tilde{\phi}_i \, / N$
5. Se obtiene la matriz de covarianza $\Omega = \Phi\Phi^T/N$, donde $\Phi = [\tilde{\phi}_1 - \mu, \dots, \tilde{\phi}_N - \mu]$.
6. Se calcula el conjunto de valores $\{\lambda_1, \dots, \lambda_{n \cdot m}\} \subset \mathbb{R}$ y vectores propios $\{\mathbf{v}_1, \dots, \mathbf{v}_{n \cdot m}\} \subset \mathbb{R}^{n \cdot m}$, de la matriz de covarianza.
7. Los vectores propios son ordenados de forma decreciente con respecto a su valor propio asociado y son colocados como columnas en la matriz $\Lambda = [\mathbf{v}_1, \dots, \mathbf{v}_p]$. Nótese que dado que los primeros vectores propios contienen la mayor parte de la información, no es necesario utilizar todos los $n \cdot m$ vectores propios. Es necesario analizar los valores propios, por lo que $p < n \cdot m$.
8. Se construye el espacio propio con $\Psi = \Phi\Lambda$.
9. Se proyecta al espacio propio cada imagen del conjunto de entrenamiento con $\hat{\phi}_i = \Psi^T(\tilde{\phi}_i - \mu)$, para $i = 1, \dots, N$; con lo que se forma el conjunto de entrenamiento $\{\hat{\phi}_1, \dots, \hat{\phi}_N\} \subset \mathbb{R}^p$.

El conjunto de proyecciones $\{\hat{\phi}_i, \dots, \hat{\phi}_N\}$ se utiliza para el entrenamiento de algún clasificador. Para el reconocimiento de un objeto, este primero debe ser proyectado al espacio propio y después procesado por el clasificador. Es decir, sea una imagen I, se convierte a un vector φ, este es normalizado y se obtiene $\tilde{\varphi}$, el vector es proyectado al espacio propio con $\hat{\varphi} = \Psi^T(\tilde{\varphi} - \mu)$. El vector proyectado $\hat{\varphi}$ es procesado por el clasificador previamente entrenado para el reconocimiento del objeto en la imagen.

2.3 Reconocimiento de texturas

El análisis de superficies con texturas ha sido un tema de interés debido al amplio potencial de su aplicación, que incluye clasificación de materiales y objetos que varían en ángulos de perspectiva, clasificación y segmentación de escenas de imágenes de navegación en exteriores, análisis de imágenes aéreas y recuperación de imágenes. Sin embargo, la clasificación de las texturas de interiores no ha sido estudiada ampliamente.

La textura es percibida por los patrones de una amplia variedad de superficies naturales o sintéticas como madera, metal, textiles, etcétera. Si un área de una imagen de textura tiene una gran variación de intensidad, entonces la característica dominante de dicha área seria la textura. Si esta área tiene poca variación en la intensidad entonces la característica dominante dentro del área es el tono.

La textura es una propiedad intrínseca de la superficie de los objetos, sin embargo, a pesar de la cantidad de trabajos de investigación relacionados con este tema, no se tiene una definición precisa de la textura. Existen dos clases principales para el reconocimiento de texturas: análisis estadístico y análisis estructural. El primero es adecuado para el reconocimiento de texturas naturales, mientras que el segundo es adecuado para las texturas con patrones repetitivos. A continuación se presentan los métodos para la caracterización de texturas que se utilizan en este trabajo.

2.3.1 Matrices de co-ocurrencia

Las matrices de co-ocurrencia de niveles de gris (MCNG) es uno de los métodos más empleados para caracterizar la textura, el cual fue presentado por Haralick et al. [20]. Las MCNG se basan en la distancia entre pixeles, el ángulo y número de niveles de gris. Un elemento $P(i,j,d,\theta)$ de la MCNG representa la frecuencia relativa, donde i es el nivel de gris del pixel ubicado en (x,y), y j es el nivel de gris del pixel vecino a una distancia y orientación d y θ, respectivamente.

El cálculo de las MCNG se basa en la distancia entre pixeles, el ángulo de los pixeles (0°, 45°, 90° y 135°) y el número de niveles de gris. Sea $I(x,y)$ una imagen de tamaño $r \times c$ con n_g niveles de gris, las MCNG, que dependen del ángulo y la distancia, se obtienen con:

$$P(i,j,d,\theta) = \#\{(x_1,y_1),(x_2,y_2) \in r \times c | I(x_1,y_1) = i, I(x_2,y_2) = j\} \quad (1)$$

Donde # denota la cantidad de elementos en el conjunto. $P(i,j,d,\theta)$ representa la información contenida entre los pixeles (x_1,y_1) y (x_2,y_2) separados por una distancia d y orientados en un ángulo θ. Siendo los promedios y desviaciones estándar:

$$\mu_x = \sum_i \sum_j i \cdot p(i,j) \quad (2)$$

$$\mu_y = \sum_i \sum_j j \cdot p(i,j) \quad (3)$$

$$\sigma_x = \sum_i \sum_j (i - \mu_x)^2 p(i,j) \tag{4}$$

$$\sigma_y = \sum_i \sum_j (j - \mu_y)^2 p(i,j) \tag{5}$$

Se extraen 18 características de las MCNG como se propone en [1]:

Energía

$$f_1 = \sum_i \sum_j p(i,j)^2 \tag{6}$$

Contraste

$$f_2 = \sum_{k=0}^{n_g} k^2 \left\{ \sum_{i=1}^{n_g} \sum_{j=1}^{n_g} p(i,j) \,\middle|\, |i-j| = k \right\} \tag{7}$$

Correlación

$$f_3 = \frac{\sum_i \sum_j (ij) p(i,j) - \mu_x \mu_y}{\sigma_x \sigma_y} \tag{8}$$

Varianza

$$f_4 = \sum_i \sum_j (i - \mu)^2 p(i,j) \tag{9}$$

Homogeneidad

$$f_5 = \sum_i \sum_j \frac{p(i,j)}{1 + (i-j)^2} \tag{10}$$

Suma promedio

$$f_6 = \sum_{i=2}^{2n_g} i \cdot p_{x+y}(i) \tag{11}$$

Donde $p_{x+y}(k) = \sum_{i=1}^{n_g} \sum_{j=1}^{n_g} p(i,j)$, siendo $i + j = k$ y $k = 2,3,\ldots,2n_g$.

Suma de varianzas

$$f_7 = \sum_{i=2}^{2n_g} (i - f_8)^2 p_{x+y}(i) \tag{12}$$

Suma de entropías

$$f_8 = -\sum_{i=2}^{2n_g} p_{x+y}(i) \log p_{x+y}(i) \tag{13}$$

Entropía

$$f_9 = -\sum_i \sum_j p(i,j) \log p(i,j) \tag{14}$$

Diferencia de varianzas

$$f_{10}\text{: varianza de } p_{x-y} \tag{15}$$

Donde $p_{x-y}(k) = \sum_{i=1}^{n_g} \sum_{j=1}^{n_g} p(i,j)$, siendo $|i-j| = k$ y $k = 0,1,\dots,n_g - 1$.

Diferencia de entropía

$p_{x+y}(k) = \sum_{i=1}^{n_g} \sum_{j=1}^{n_g} p(i,j)$, siendo $i + j = k$ y $k = 2,3,\dots,2n_g$.

$$f_{11} = -\sum_{i=0}^{n_g-1} p_{x-y}(i) \log p_{x-y}(i) \tag{16}$$

Información de métricas de correlación

$$f_{12} = \frac{f_8 - a_1}{\max(b_x, b_y)} \tag{17}$$

$$f_{13} = \sqrt{1 - \exp(-2(a_2 - f_9))} \tag{18}$$

Donde $a_1 = -\sum_i \sum_j p(i,j) \log(p_x(i)p_y(j))$, $a_2 = -\sum_i \sum_j p_x(i)p_y(j) \log(p_x(i)p_y(j))$; siendo $p_x(j) = \sum_i p(i,j)$ y $p_y(i) = \sum_j p(i,j)$; b_x y b_y son las entropías de p_x y p_y, respectivamente.

Autocorrelación

$$f_{14} = \sum_i \sum_j (ij) p(i,j) \tag{19}$$

Prominencia de cúmulo

$$f_{15} = \sum_i \sum_j \left(i + j - \mu_x - \mu_y\right)^4 p(i,j) \tag{20}$$

Forma de cúmulo

$$f_{16} = \sum_i \sum_j \left(i + j - \mu_x - \mu_y\right)^3 p(i,j) \tag{21}$$

Disimilitud

$$f_{17} = \sum_i \sum_j |i-j| p(i,j) \tag{22}$$

Máxima probabilidad

$$f_{18} = \max_{i,j} p(i,j) \tag{23}$$

Los valores de las características extraídos de las cuatro orientaciones son promediados, como se sugiere en [1].

2.3.2 Patrones binarios locales

Los patrones binarios locales (PBL) son una primitiva estadística de textura invariante en escala de grises [19]. Para cada pixel de una imagen, se produce un código binario al comparar los pixeles vecinos con el del centro. Un histograma es creado para recoger las frecuencias de diferentes patrones binarios. Los PBL se pueden considerar como un operador micro-texton, tal que en cada pixel se detecta el mejor PBL que representa un borde, un punto, etc. Después de procesar toda la imagen a ser analizada, cada pixel tiene una etiqueta que corresponde a una de las etiquetas del *vocabulario*. El histograma de etiquetas es calculado y después empleado para describir la textura.

Cada pixel de la imagen primero es etiquetado al comparar la diferencia entre el pixel central y los pixeles vecinos utilizando la función escalón:

$$u(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases} \tag{24}$$

La concatenación de las etiquetas vecinas es usada como un descriptor de cada patrón. Los patrones son uniformes si las transiciones entre “0” y “1” sin menores o igual a dos. Por ejemplo, “01100000” y “11011111” son patrones uniformes. El histograma de los patrones uniformes en toda la imagen es utilizado como el vector de características [19]. Cada patrón en la imagen es asignado a una etiqueta única por la siguiente ecuación:

$$PBL(p,R) = \sum_{i=0}^{p-1} u(t_i - t_c) 2^i \tag{25}$$

Donde p denota el número de pixeles vecino con respecto al pixel del centro, R representa la distancia entre el pixel central y cada uno de los pixeles vecinos, t_c y t_i son las intensidades de los pixeles centrales y del pixel vecino i, respectivamente, y $u(x)$ es la función escalón.

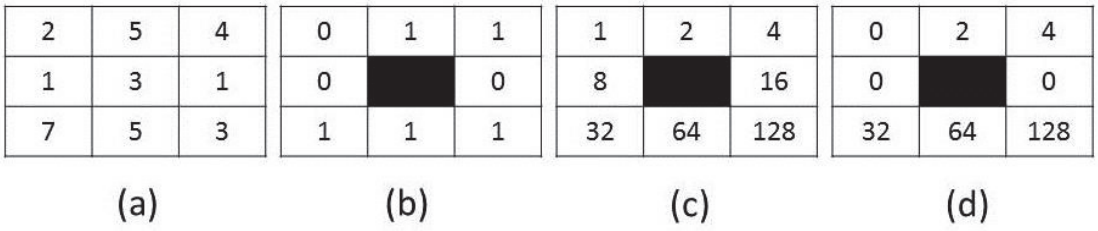

Figura 2.1 Patrones Binarios Locales

En la Figura 2.1 se muestra un ejemplo de PBL, donde $p = 8$ y $R = 1$: (a) es l vecindario original de tamaño 3×3, donde $t_c = 3$ y $t_i = \{2,5,4,1,1,7,4,3\}$, $i = 0, \dots, 7$; (c) son los pesos PBL; (d) son los valores obtenidos de la comparación entre los pixeles vecinos y el pixel central que son multiplicados por los pesos PBL (b), dado su correspondiente pixel. Los elementos son sumados para formar el valor de esta unidad de textura: $PLB(8,1) = \sum_{i=0}^{7} u(t_i - 3)2^i = 230$.

2.3.3 Patrones binarios locales avanzados

Los PBL convencionales solo consideran los patrones uniformes de las imágenes, descartando información importante de varios patrones para imágenes cuyos patrones dominantes no son uniformes [19]. Los patrones binarios avanzados (PBLA) [40] al extender el enfoque tradicional de los PBL, proponen un nuevo método de clasificación de textura invariante a la rotación que refleja la información del patrón dominante contenida en la textura de la imagen y que captura la información de la distribución espacial de los patrones dominantes.

Los patrones uniformes son formalmente definidos utilizando la métrica de uniformidad $U(\cdot)$, el cual corresponde al número de transiciones “0/1” en el patrón. Por ejemplo, los patrones “00000000” y “11111111” tienen un valor U de 0, mientras que patrones “01111111” y “00000011” tienen un valor U de 2. Los PBLA definen como patrones uniformes a aquellos cuyo valor U es 2, con lo que se propone el siguiente operador para la descripción de textura invariante a la rotación:

$$PBLA(p,R) = \begin{cases} \sum_{i=0}^{p-1} u(t_i - t_c), & U(PBL(p,R)) \leq 2 \\ p+1, & U(PBL(p,R)) > 2 \end{cases} \tag{26}$$

Donde:

$$U(PBL(p,r)) = |u(t_{p-1} - t_c) - u(t_0 - t_c)| + \sum_{i=1}^{p-1} |u(t_i - t_c) - u(t_{i-1} - t_c)| \tag{27}$$

Por definición, exactamente $p + 1$ patrones binarios uniformes pueden darse en un vecindario circularmente simétrico de un conjunto de p pixeles. Con la ecuación (27) se asigna una única etiqueta a cada uno de ellos que corresponden al número "1", mientras que los patrones no uniformes son agrupados bajo la etiqueta $p + 1$ [40].

3 Propuesta de navegación

Para el propósito de navegación autónoma con ajuste de velocidad, el reconocimiento de texturas con alta precisión no es un requisito estricto. Aquí se propone imitar la percepción humana para el ajuste de velocidad, por lo que se utiliza una red neuronal difusa para estimar la velocidad del vehículo considerando tanto las texturas como la distancia que hay entre el vehículo y los objetos reconocidos o detectados.

La navegación comienza con la planeación de trayectorias. Para el ajuste de velocidad se adquieren imágenes del entorno del vehículo y se modelan las texturas con los descriptores de Haralick y los patrones binarios locales; por otra parte, los objetos son modelados con la visión basada en apariencias. Posteriormente, las texturas y los objetos son clasificados, cada uno con su respectiva red neuronal supervisada, entrenada previamente con imágenes representativas tanto de texturas como de objetos. Una red neuronal difusa recibe la clase de textura y la distancia del objeto detectado, dando como salida la velocidad del vehículo quien ajusta su velocidad y conforme avanza adquiere nuevas imágenes y calcula con los sensores la distancia de los objetos, manteniendo el ciclo. La planeación de trayectorias se actualiza cuando el vehículo no puede superar algún obstáculo, ver Figura 3.1.

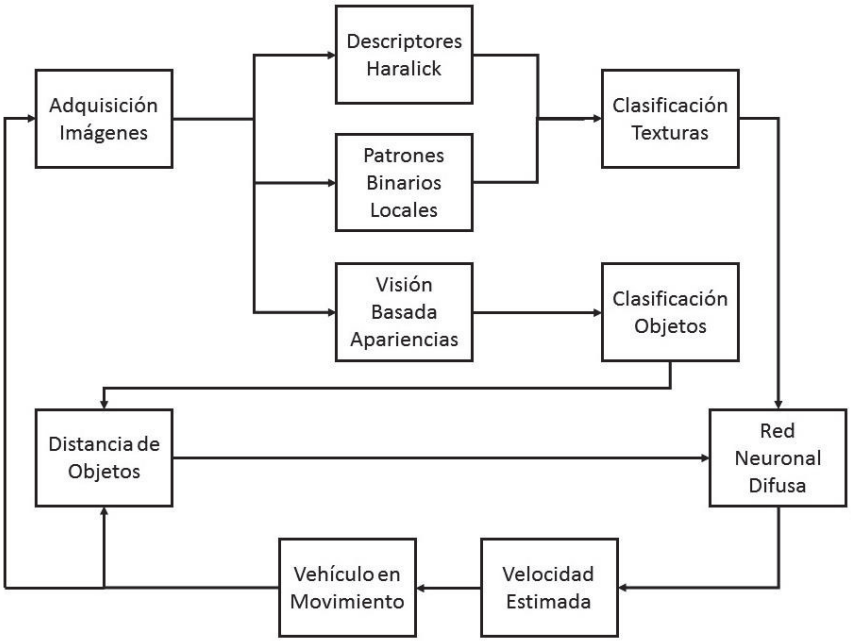

Figura 3.1 Diagrama de bloques del enfoque propuesto para la navegación del vehículo

En esta sección se describen las arquitecturas de las redes neuronales necesarias para los reconocimientos, así como el algoritmo de planeación de trayectorias seleccionado para implementarse en este trabajo.

3.1 Algoritmo de planeación de trayectorias

Existen varias teorías detrás de la navegación de robots. Uno de los algoritmos de planeación de trayectorias que se utilizan para la navegación autónoma de robots es el *wavefront*, que consta principalmente de cuatro pasos:

1. **Crear una versión discretizada del mapa**: crear un plano cartesiano discreto que se representa por una matriz en donde se marcan los espacios vacíos, las ubicaciones del robot, el destino deseado y los obstáculos.
2. **Definir las ubicaciones del robot y de la posición final**: normalmente las posiciones del robot, destino y obstáculos son representadas como R, G y W, respectivamente.
3. **Llenar los demás espacios con la propagación de ola (wavefront)**: los espacios son *llenados* con *pesos* que representan la distancia entre el robot y la posición final, utilizando el algoritmo de wavefront.
4. **Trazar la trayectoria del robot hacia la posición final**: se planea la trayectoria tal que la suma de los pesos de los cuadros entre el robot y la posición final sea mínima.

En el paso 3, la propagación de la ola se hace con el siguiente seudocódigo:

Entrada: Mapa discretizado, posición inicial y final, posición de obstáculos

Salida: Trayectoria de navegación

```
1:  si el espacio es un obstáculo entonces ignorar este espacio, ir al siguiente espacio B;
2:  en caso contrario, si el espacio frontera es la ubicación del robot y tiene un peso
    entonces
3:      encontrar el espacio frontera con el peso más pequeño;
4:      devolver tal dirección al control del robot;
5:      el robot se mueve a ese nuevo espacio;
6:  en caso contrario, si el espacio frontera es la posición final entonces
7:      marcar el espacio con el numero 3;
8:  en caso contrario, si el espacio está marcado con un número entonces
9:      encontrar el espacio con el peso más pequeño;
10:     marcar el espacio el número más pequeño + 1;
11: si no se encuentra un camino entonces
12:     ir al siguiente espacio, ordenar toda la matriz;
```

13: *si* aún no hay camino después de la búsqueda *entonces*
14: ir al espacio inicial y comenzar de nuevo, sin limpiar el mapa;
15: ordenar toda la matriz;
16: repetir hasta encontrar un camino;
17: *si* aún no se encuentra un camino y la matriz está llena *entonces*
18: no existe solución;
20: limpiar toda la matriz de obstáculos y comenzar de nuevo;
21: *fin*

Algoritmo 3.1 Seudocódigo del algoritmo de wavelength para planeación de trayectorias

3.2 Arquitectura de la red neuronal difusa

La red neuronal propuesta tiene cinco capas, la red neuronal difusa tiene dos entradas: r y d que son la clase de textura y distancia del objeto, respectivamente, y una salida f que es el valor de la velocidad. La Figura 3.2 muestra la correspondiente arquitectura de la red neuronal difusa, la cual es un modelo Sugeno de primer orden. La salida del i-ésimo nodo de la capa l es denotado como $O_{i,l}$.

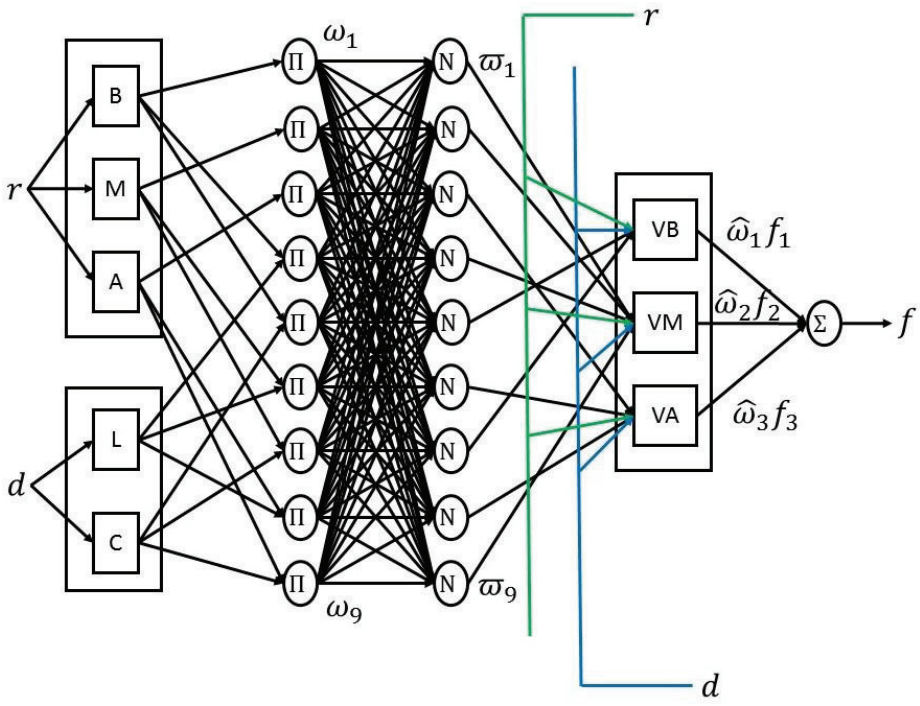

Figura 3.2 Arquitectura de la red neuronal difusa

Las texturas son meta-clasificadas en las categorías de rugosidades Baja (B), Media (M) y Alta (A), ver Figura 3.3. La distancia está dividida en dos conjuntos difusos: Lejos (L) y Cerca (C), ver Figura 3.4. El proceso de *fusificación* mapea los valores *crisp* en términos lingüísticos difusos con valores en el intervalo [0,1]. Los valores difusos de salida para la velocidad son: Velocidad Baja (VB), Velocidad Media (VM) y Velocidad Alta (VA), ver Figura 3.5.

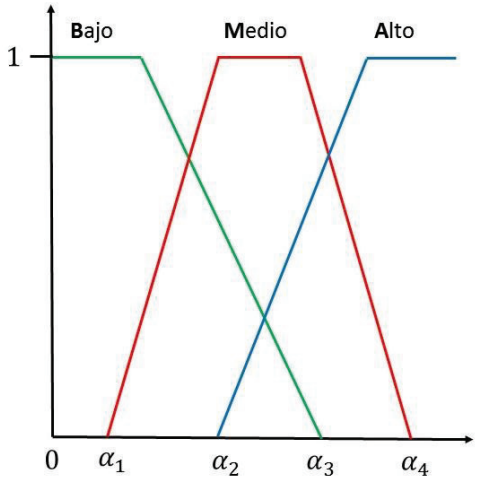

Figura 3.3 Función de pertenencia para la rugosidad de textura

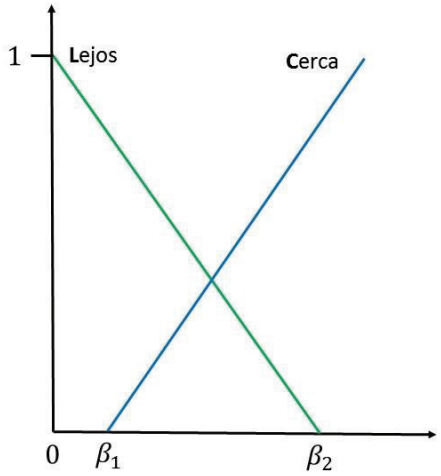

Figura 3.4 Función de pertenencia para la proximidad de objetos

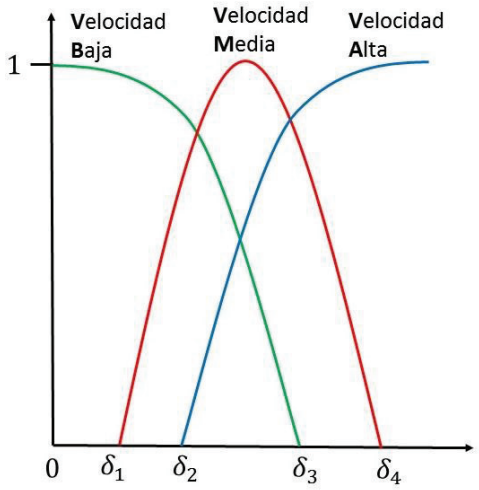

Figura 3.5 Función de pertenencia para la velocidad

Las funciones de pertenencia para la rugosidad de las texturas, proximidad de objetos y velocidad son trapecio, pirámide y campana, respectivamente; ver ecuaciones (28), (29) y (30):

$$\mu(x;a,b,c,d) = \begin{cases} 0, & x \leq a \\ \dfrac{x-a}{b-a}, & a < x \leq b \\ 1, & b < x \leq c \\ \dfrac{d-x}{d-c}, & c < x \leq d \\ 0, & d < d \end{cases} \tag{28}$$

$$\mu(x;a,b,c) = \begin{cases} 0, & x \leq a \\ \dfrac{x-a}{b-a}, & a < x \leq b \\ \dfrac{c-x}{c-b}, & b < x \leq c \\ 0, & c < x \end{cases} \tag{29}$$

$$\mu(x;a,b,c) = \frac{1}{1+\left|\frac{x-c}{a}\right|^{2b}} \tag{30}$$

La base de reglas modela la relación entrada-salida; tomando como A, B y C como variables de los predicados respectivos, la forma de las reglas de inferencia es:

SI *rugosidad* es A **Y** *distancia* es B **ENTONCES** *velocidad* es C

La reglas de inferencia de la red neuronal difusa están enlistadas en la Tabla 3.1.

Tabla 3.1 Reglas si-entonces del sistema de inferencia de la red neuronal difusa

Regla No.	Entrada		Salida
	Rugosidad	*Distancia*	*Velocidad*
1	B	-	VM
2	M	-	VM
3	A	-	VA
4	B	L	VM
5	B	C	VB
6	M	L	VA
7	M	C	VB
8	A	L	VA
9	A	C	VM

A continuación, se explica cada capa de la red neuronal propuesta.

Capa 1: Cada nodo i de esta capa es un nodo adaptado con una función nodo:

$$O_{1,1} = \mu_r(B) \quad O_{2,1} = \mu_r(M) \quad O_{3,1} = \mu_r(A) \quad O_{4,1} = \mu_d(L) \quad O_{5,1} = \mu_d(C)$$

Donde r y d son los nodos de entrada hacia los nodos de etiquetas lingüísticas {B,M,A} y {L,C} respectivamente; μ_r y μ_d son las funciones de pertenencia de la rugosidad y distancia, respectivamente. Los parámetros en esta capa son conocidos como parámetros premisa.

Capa 2: Cada nodo en esta capa esta fijo a los nodos etiquetados como Π, cuya salida es el producto de todas las señales de salida, $O_{i,2} = \omega_i$, $i = 1, \dots, 9$, donde:

$$\omega_1 = \mu_r(B) \qquad \omega_4 = \mu_r(B)\mu_d(L) \qquad \omega_7 = \mu_r(M)\mu_d(C)$$
$$\omega_2 = \mu_r(M) \qquad \omega_5 = \mu_r(B)\mu_d(C) \qquad \omega_8 = \mu_r(A)\mu_d(L)$$
$$\omega_3 = \mu_r(A) \qquad \omega_6 = \mu_r(M)\mu_d(L) \qquad \omega_9 = \mu_r(A)\mu_d(C)$$

Cada nodo de salida representa la fortaleza de disparo de cada regla. El operador norma T empleado es el producto algebraico.

Capa 3: Cada nodo de esta capa esta fijo y es etiquetado como N. El i-esimo nodo calcula el radio de fuerza de la i-esima regla en la suma de todas las fuerzas de disparo de las reglas:

$$O_{i,3} = \varpi_i = \frac{\omega_i}{\sum_{j=1}^{9} \omega_j}$$

Para $i = 1, \dots, 9$, las salidas de esta capa se les conoce como la fortaleza de disparo normalizado.

Capa 4: Cada nodo k en esta capa es un nodo adaptado con el nodo función:

$$O_{k,4} = \widehat{\omega}_k(p_k r + q_k d + s_k)$$

Para $k = 1,2,3$; donde $\{p_k, q_k, s_k | k = 1,2,3\}$ es el conjunto de parámetros en este nodo. Los parámetros en esta capa son conocidos como los parámetros de consecuencia. De aquí que, los parámetros de consecuencia son:

$$\widehat{\omega}_1 = \varpi_5 + \varpi_7$$
$$\widehat{\omega}_2 = \varpi_1 + \varpi_2 + \varpi_4 + \varpi_9$$
$$\widehat{\omega}_3 = \varpi_3 + \varpi_6 + \varpi_8$$

Capa 5: El único nodo de esta capa es un nodo fijo etiquetado con Σ, con el cual se calcula la salida total como la suma de todas las señales procesadas, para la desfusificación se utiliza el método de centro de área:

$$O_{1,5} = \sum_{k=1}^{3} \widehat{\omega}_k f_k$$

El procedimiento de desfusificación mapea la salida difusa desde el mecanismo de inferencia hacia la señal crisp. La fórmula para calcular la velocidad del vehículo es $v = v_{max} \cdot v_n$, donde v_{max} es la máxima velocidad que puede alcanzar el vehículo y v_n es el valor de salida de la red neuronal difusa, un valor en el intervalo [0,1].

Para el entrenamiento de la red neuronal difusa es utilizado el algoritmo de aprendizaje híbrido. Se observa que cuando los valores de los parámetros de la premisa son fijos, la salida puede expresarse como una combinación lineal de los parámetros de consecuencia [18]. El algoritmo de aprendizaje híbrido, el cual combina descenso empinado y el estimador de mínimos cuadrados para la rápida identificación de parámetros, identifica estos parámetros lineales por el método de mínimos cuadrados.

La salida f en la Figura 3.2 puede escribirse como:

$$f = \widehat{\omega}_1 f_1 + \widehat{\omega}_2 f_2 + \widehat{\omega}_3 f_3$$
$$f = \widehat{\omega}_1(p_1 r + q_1 d + s_1) + \widehat{\omega}_2(p_2 r + q_2 d + s_2) + \widehat{\omega}_3(p_3 r + q_3 d + s_3)$$

De aquí que:

$$\begin{aligned} f \quad = \quad & (\widehat{\omega}_1 r)p_1 + (\widehat{\omega}_1 d)q_1 + (\widehat{\omega}_1)s_1 + (\widehat{\omega}_2 r)p_2 + (\widehat{\omega}_2 d)q_2 + (\widehat{\omega}_2)s_2 \\ + \quad & (\widehat{\omega}_3 r)p_3 + (\widehat{\omega}_3 d)q_3 + (\widehat{\omega}_3)s_3 \end{aligned}$$

El cual es una combinación lineal de parámetros $\{p_k, q_k, s_k | k = 1,2,3\}$. Por lo que, el conjunto de entrenamiento son pares ordenados de la forma de una matriz $\mathbf{A}$ de tamaño $m \times 9$, conocida como la matriz de diseño, y el vector de salida $\mathbf{y}$ de tamaño $m \times 1$; entonces se obtiene la ecuación matricial:

$$\mathbf{A}\vec{\theta} = \mathbf{y}$$

Donde $\vec{\theta}$ es un vector de tamaño 9×1 que contiene las variables o parámetros desconocidos $\{p_k, q_k, s_k | k = 1,2,3\}$. Este es un problema de mínimos cuadrados estándar, y la mejor solución para $\vec{\theta}$, el cual minimiza $\left\|\mathbf{A}\vec{\theta} - \mathbf{y}\right\|^2$, es el estimador de mínimos cuadrados $\vec{\theta}^*$:

$$\vec{\theta}^* = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{y}$$

Donde $(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T$ es la pseudoinversa de $\mathbf{A}$. Puede incluso ser utilizada la fórmula del estimador de mínimos cuadrados para calcular$\vec{\theta}^*$. Específicamente, sea $\mathbf{a}_i^T$ el i-esimo vector renglón de la matriz $\mathbf{A}$, y sea y_i^T el i-esimo elemento de $\mathbf{y}$; entonces $\vec{\theta}$ puede calcularse iterativamente como sigue:

$$\vec{\theta}_{i+1} = \vec{\theta}_i + \mathbf{P}_{i+1}\mathbf{a}_{+1}(y_{i+1}^T - \mathbf{a}_{i+1}^T\vec{\theta}_i)$$

$$\mathbf{P}_{i+1} = \mathbf{P}_i - \frac{\mathbf{P}_i\mathbf{a}_{+1}\mathbf{a}_{i+1}^T\mathbf{P}_i}{1 + \mathbf{a}_{i+1}^T\mathbf{P}_i\mathbf{a}_{+1}}$$

Las condiciones iniciales necesarias son $\vec{\theta}_0 = \vec{0}$ y $\mathbf{P}_0 = \gamma\mathbf{I}$, donde γ es un número positivo grande y $\mathbf{I}$ es la matriz identidad.

4 Experimentos y resultados

En este capítulo se presentan los experimentos y sus respectivos resultados empleando los algoritmos descritos en los capítulos anteriores. En los experimentos se lleva registro de las velocidades que se calculan durante el recorrido del vehículo. Las texturas donde el vehículo es probado son diferentes tipos de mosaicos.

4.1 Plataforma robótica

Para los experimentos de navegación autónoma se utiliza un robot dotado de ruedas para impulsarse, la plataforma robótica es un Lego Mindstorm EV3 [41]. Lego es un kit de robótica con el cual se puede ensamblar diferentes arquitecturas de robots, tal que cumpla los requerimientos del usuario. El usuario no solo controla la apariencia del robot, sino también puede programar los patrones de comportamiento de este.

El robot emplea una unidad de procesamiento, dos servomotores para la transmisión de potencia a las ruedas y un sensor infrarrojo que está localizado en el frente para detectar objetos y medir la distancia hasta estos. El kit no está dotado de algún sistema de visión, por lo que se le instala un teléfono móvil para capturar y enviar de forma inalámbrica las imágenes a una computadora personal (CP) en donde son procesadas. En la Figura 4.1 se puede ver el robot y la ubicación de cada uno de los elementos que se han descrito anteriormente.

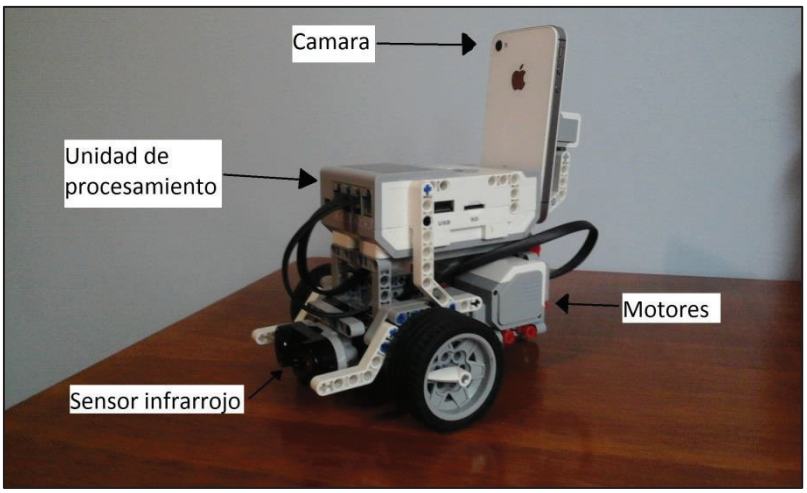

Figura 4.1 Robot empleado para los experimentos

En esta plataforma se utiliza una CP y el procesador del robot, para formar una arquitectura maestro-esclavo, los cuales se comunican de forma inalámbrica. En la CP se implementan y corren los algoritmos de estimación de velocidad y de planeación de trayectorias. El robot, por un lado, reporta a la computadora las lecturas del sensor y transmite las imágenes capturadas por el teléfono móvil; por otra parte, realiza los movimientos de acuerdo con las instrucciones que le comunica la CP.

4.2 Navegación en interiores

En esta sección se presentan los experimentos y resultados obtenidos de las pruebas de navegación del robot mostrado en la Figura 4.1. Los experimentos se realizan en un entorno que se asume desconocido y estático, cuya área es de 9 m^2, en donde el suelo está cubierto con diferentes clases de mosaicos. La ubicación objetivo está localizada a 4.24 metros en línea recta de la ubicación inicial del robot. Se colocan dos obstáculos en medio del trayecto entre la posición inicial y la posición final, como se observa en la Figura 4.2.

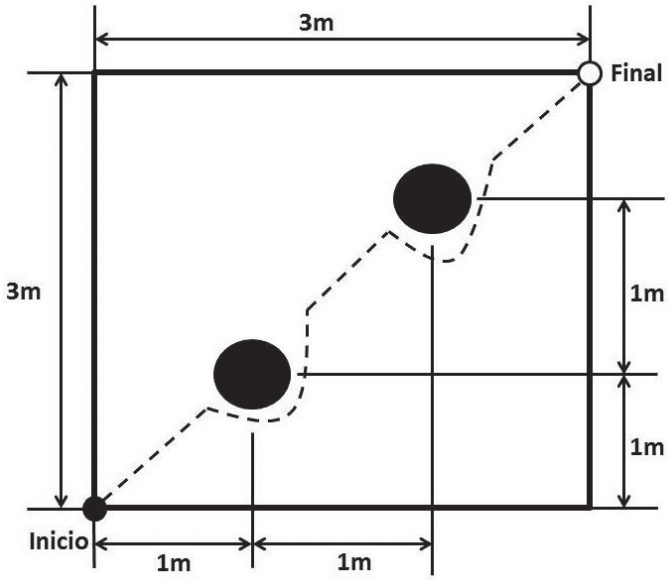

Figura 4.2 Dimensiones de la superficie de prueba, posiciones de los obstáculos y trayectoria típica del robot durante los experimentos

La línea punteada de la Figura 4.2 indica la trayectoria típica de la navegación del robot durante los experimentos. Se realizaron 30 experimentos los cuales se dividieron en tres partes:

1. Diez pruebas a velocidad media constante (7.51 cm/s).

2. Diez pruebas a máxima velocidad constante (15.02 cm/s).
3. Diez pruebas con ajuste de velocidad.

Para probar y evaluar la actual propuesta se buscaron modelos de superficies en trabajos relacionados de navegación autónoma que sean empleados como referencias para evaluar el desempeño de los algoritmos. Pero no hubo entornos específicos para emplearse como marco de referencia; las superficies empleados para los experimentos de los trabajos [1], [2], [17], [42]y [43] no siguen un modelo estándar porque no existe uno que sirva como referencia. De aquí que cada trabajo proponga su propio entorno.

A pesar de que no existen características específicas para una superficie, se han establecido criterios para evaluar el desempeño de la navegación del robot [4], [44]:

- Precisión en la estimación de la posición del robot,
- Detección rápida y precisa del entorno del robot,
- Planeación de trayectorias para desplazarse de un punto a otro sin chocar contra obstáculos en entornos desconocidos,

Esto se cumple en la actual propuesta y se agrega lo siguiente:

- El tiempo total que toma al robot en realizar el trayecto,
- La comparación entre las ubicaciones inicial y final.

La detección del entorno del robot y la planeación de trayectorias son reforzados con el ajuste de velocidad. Al ajustar la velocidad en función de las características de la superficie, la seguridad es mayor y el tiempo de recorrido del robot es reducido. Esto es, si el robot detecta que la rugosidad del suelo es baja entonces se moverá despacio, de modo que aunque al robot le tome más tiempo para alcanzar su destino la probabilidad de que sufra un accidente disminuye.

En contraste, si se detecta que la rugosidad de la superficie es alta entonces el robot se mueve rápido y por lo tanto el tiempo del recorrido disminuye. Por lo tanto, al comparar el tiempo de recorrido con y sin ajuste de velocidad, el desempeño del robot es mejorado al usar información sobre las características de la superficie.

En la Tabla 4.1 se observa que a velocidad media constante el robot alcanza la posición final, pero está localizada a 32.76 cm de la posición deseada. Para velocidad máxima constante, el robot prácticamente se queda a la mitad del camino, ya que termina su recorrido en un punto que se

encuentra a 201.71 cm del punto deseado. Con ajuste de velocidad, el robot concluye la trayectoria a 16.34 del punto destino.

Tabla 4.1 Resultados obtenidos del conjunto de experimentos realizados en la superficie con las características mostradas en la Figura 4.2

	Velocidad constante		**Ajuste de velocidad**
	Media	*Máxima*	
Velocidad de navegación	7.51 cm/s	15.02 cm/s	9.36 cm/s (promedio)
Distancia recorrida	450.89 cm	238.53 cm	440.56 cm
Error de ubicación final	32.76 cm	201.72 cm	16.34 cm
Tiempo de navegación	86.18 s	40.90 s	65.09 s

Cuando el robot se mueve a máxima velocidad, sus ruedas no hacen el contacto apropiado con las superficies de baja tracción y en consecuencia sufren muchos derrapes, generando errores de odometría.

A velocidad constante media el robot presenta un mejor desempeño que el obtenido a velocidad constante máxima pues mejora su posición final respecto de la posición objetivo. Sin embargo, el deslizamiento de las ruedas aún se produce en superficies de baja tracción, por lo que debe moverse lentamente en donde se requiera.

Con ajuste de velocidad, el robot tiene menos errores de odometría porque sufre menos derrapes en sus ruedas ya que disminuye su velocidad en superficies de baja tracción, dando mayor tiempo a las ruedas para hacer mejor contacto con el suelo. Aunque el tiempo del trayecto no disminuyó, el cálculo de la ubicación del robot fue más exacto.

Al ajustar la velocidad de acuerdo con las características de la superficie, la seguridad es incrementada y/o el tiempo del recorrido es reducido. Si se detecta que la rugosidad de la superficie es baja entonces el robot se mueve despacio y aunque emplea más tiempo para alcanzar la ubicación objetivo, la probabilidad de que el robot sufra un accidente se reduce.

Considerando como 100% la distancia real entre la posición final y la posición inicial del robot, cuando éste se mueve a velocidad media constante (7.51 cm/s) recorre 106.28% de la ruta. La

distancia del viaje incrementa debido a la evasión de obstáculos y los errores odómetricos que se acumulan por el deslizamiento de las ruedas.

Por otra parte, cuando el robot se mueve a velocidad máxima constante (15.02 cm/s), este recorre, en promedio, 56.23% de la trayectoria. La explicación plausible es que cuando el robot se mueve máxima velocidad las ruedas sufren deslizamiento con mayor frecuencia y por lo tanto la estimación de la ubicación del robot es más imprecisa.

Con ajuste de velocidad el robot recorre el 103.85% de la distancia entre las posiciones inicial y final. En este caso la distancia es menor que la recorrida a velocidad media constante porque el derrape de las ruedas es menos frecuente.

Al adaptar la velocidad, el robot se mueve más despacio en áreas que provocan el deslizamiento de las ruedas y gracias a que se dan menos deslizamientos de estas, la estimación de la posición del robot es más exacta, por lo tanto, mejora el cálculo de su posición respecto de la posición objetivo.

El tiempo de navegación con ajuste de velocidad es 24.48% menor y 37.17% mayor que a velocidad constante medio y máximo, respectivamente. Nótese que a velocidad constante media el robot recorre la trayectoria con buena precisión, pero en un tiempo de recorrido mayor.

A velocidad máxima constante el recorrido toma menos tiempo, pero la precisión para seguir la trayectoria es muy mala. Con ajuste de velocidad el desempeño mejora, porque la precisión del viaje del robot sobre la trayectoria es buena y es realizado en menos tiempo, es decir, el robot se mueve a velocidad óptima en función de las características de la superficie para evitar el derrape de las ruedas. Con el enfoque propuesto la velocidad promedio representa el 62.32% de la máxima velocidad que puede alcanzar el robot.

Para la ubicación del robot con ajuste de velocidad se obtiene un error del 3.85%; similar al reportado en trabajos relacionados aunque significativo considerando que solo se utilizó odometría, lo que es poco común al abordar el problema de la reducción del deslizamiento de las ruedas para mejorar la ubicación del robot. Los resultados muestran que, con ajuste de velocidad, el cálculo de la posición del robot puede mejorar, ya que el robot se mueve lento en superficies de baja tracción, lo que propicia que las ruedas derrapen menos y por lo tanto se acumulan menos errores odómetricos.

5 Discusión

La navegación autónoma utiliza con frecuencia métodos basados en odometría para calcular la distancia entre las posiciones actual y final de los vehículos; sin embargo, estos métodos presentan la desventaja de que acumulan errores conforme avanza el vehículo debidos principalmente a los derrapes de las ruedas.

La mayoría de los trabajos que utilizan odometría, son auxiliados o apoyados por marcas u otros dispositivos para mejorar el cálculo de la ubicación de los vehículos [12], [14]. La desventaja de estos enfoques es el alto costo computacional. La propuesta actual previene y reduce el derrape de las ruedas al ajustar la velocidad del vehículo en función de las características de la superficie, lo cual es una ventaja relevante para los métodos de localización basados en odometría.

Los resultados de este capítulo muestran la mejora en el cálculo de la ubicación de un vehículo con ruedas al aplicar el ajuste de velocidad considerando la rugosidad de la superficie.

5.1 Navegación autónoma

Para la navegación autónoma el vehículo debe determinar su propia localización dentro del entorno; el reto es determinar las coordenadas y orientación sobre un plano de dos dimensiones. La navegación del robot puede identificarse como una combinación de las siguientes competencias, ver Figura 5.1:

1. Auto-localización
2. Planeación de trayectorias
3. Construcción del mapa
4. Interpretación del mapa

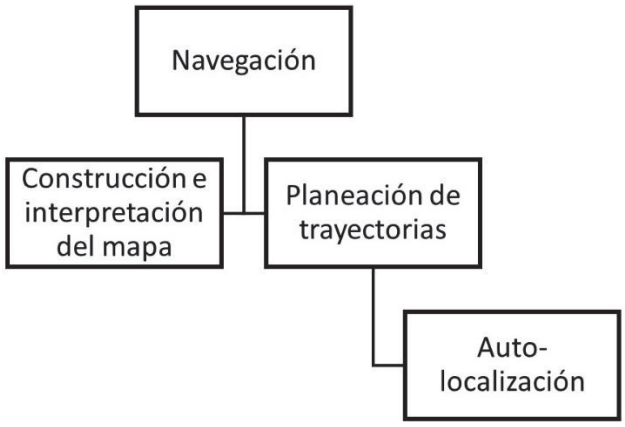

Figura 5.1 Competencias para navegación

La auto-localización denota la competencia del vehículo para establecer su propia posición sin un marco de referencia. La planeación de trayectorias como una extensión de la localización requiere determinar la posición actual y final del vehículo, ambos dentro del mismo marco de referencia. El mapa denota cualquier posición del mundo real dentro de la representación interna y su construcción cubre cualquier anotación que describa ubicaciones dentro del marco de referencia. En el mapa se marcan las partes del territorio exploradas dentro del marco de referencia y finalmente, para utilizarlo es necesario tener la habilidad para interpretarlo.

En los vehículos que usan ruedas, es frecuente encontrar el uso de odometría para el cálculo de posición [12] que inevitablemente acumula errores debido a causas como la disparidad de los diámetros de las ruedas, su mala alineación, el derrape, entre otros [13]. El derrape es producido principalmente por [6], [45]:

- Terrenos de baja tracción
- Rotación extremadamente rápida de las ruedas
- Terrenos deformables o pendientes muy inclinadas

Muchos de los trabajos sobre navegación autónoma han acotado las características que tienen los terrenos [9], [35], [31], [46]. Una de las razones es el alto costo computacional del procesamiento de los datos adquiridos del entorno. La falta de información de la rugosidad del terreno hace difícil calcular de forma precisa la ubicación del vehículo, así como la localización de la posición deseada mientras el robot se encuentra en movimiento.

Otros trabajos han abordado la reducción del error odométrico modelando el error de las lecturas de los encoders acoplados a las ruedas del vehículo, con lo que se ha obtenido cierta mejora. Sin embargo, en todos estos trabajos se auxilian de otros dispositivos o técnicas como marcas de tierra [14], rayos laser [15], visión artificial [16], entre otros [17].

El problema de la ubicación de vehículos con ruedas es un tema ampliamente estudiado debido su importancia práctica. En vez de construir un problema de control donde el vehículo requiere la trayectoria a priori, Nakamura y cols. [23] formulan un problema de campo de seguimiento de velocidades. Específicamente, un campo de velocidad es desarrollado por la trayectoria restringida del vehículo, y un control diferencial es formulado con el fin de probar la velocidad asintótica global del campo. El controlador cinemático desarrollado es agregado en el control adaptivo que fomenta el seguimiento asintótico global.

Los vehículos con ruedas han sido empleados para navegar en entornos peligrosos tales como la superficie lunar. La estimación de la velocidad es importante para los vehículos lunares que navegan sobre terreno rugoso. Las ruedas y sus efectos dinámicos, como el derrape, complican la estimación en tiempo real la velocidad del vehículo. Además, es difícil de observar la información de la aceleración porque los vehículos lunares se mueven a velocidades muy bajas. Song et al. [25] proponen un método basado en cinemática para el cálculo de la velocidad de vehículos lunares, utilizando encoders y girómetros. El modelo cinemático es determinado por la teoría de cadena de velocidad cerrada. La estimación de la velocidad es calculada al resolver la cinemática del vehículo.

Ward y Lagnemma [6] proponen un modelo basado en el deslizamiento longitudinal de la rueda y detección de condiciones de inmovilidad en robots móviles autónomos, presentando un nuevo modelo de rueda con el fin de calcular las fuerzas dinámicas del vehículo. Las fuerzas externas del vehículo y la estimación de la velocidad se obtienen utilizando encoders, mediciones inerciales y con GPS. Los resultados experimentales muestran que la técnica detecta condiciones de inmovilidad mientras se calcula la velocidad del robot durante excursiones normales. El vehículo adapta su velocidad tal que alcance su valor máximo pero que pueda detenerse rápidamente sin sufrir derrapes y moverse lentamente en superficies resbalosas. Al reducir el derrape de las ruedas del vehículo y el deslizamiento del vehículo, el error odométrico disminuye, con lo que se obtiene un cálculo más preciso de la ubicación del vehículo dentro de su entorno.

5.2 Error odométrico

La navegación autónoma es un reto para los robots móviles que requieren la competencia de determinar su propia posición en el entorno. Una forma muy conocida para calcular la ubicación de robots que utilizan ruedas es por medio de la odometría, al emplear las lecturas de los encoders acoplado a las ruedas del robot. La ubicación del robot es determinada al calcular la distancia recorrida por este desde su posición inicial hasta la posición deseada. Sin embargo, la odometría es susceptible en acumular errores durante el avance del robot: entre más largo sea el recorrido mayor es el error odómetrico. Los robots móviles con ruedas emplean diferentes técnicas para reducir los errores odómetricos.

Para superar esta dificultad, todos los trabajos relacionados a robots móviles son asistidos por otros dispositivos, tales como compases electrónicos [17], sensores sonares [4], GPS [44], etc. Por lo tanto, las propuestas de robots autónomos se han concentrado en obtener mayor información del entorno, pero el problema de reducir el deslizamiento de las ruedas no ha sido atendido completamente [45]. En los trabajos que emplean odometría junto con otros dispositivos para calcular la posición de los robots, se reportan las tasas de error con respecto a su posición final. A continuación se citan algunos trabajos relacionados que abordan métodos de localización basados en odometría.

Martinelli y cols. [15] presentan un método para permitir simultáneamente la estimación de la ubicación del robot y del error odómetro. La estimación de los componentes sintéticos se realiza por medio de un filtro de Kalman aumentado, el cual estima un estado que contiene la configuración del robot y los parámetros caracterizando los componentes del sistema del error odómetrico. Este utiliza las lecturas de los encoders y de las observaciones de un rayo láser como entradas. Los filtros son integrados en un marco coherente de auto-calibración del sistema de localización por odometría mientras el robot navega.

Jüngel [16] calcula la posición de un robot utilizando odometría y marcas de tierra (líneas en el suelo). El método no requiere representación interna de la posición del robot, el cual alterna el movimiento y la actualización del sensor, y no depende de las medidas de distancia, lo que es con frecuencia inexacto cuando se obtiene por el tamaño de marcas distantes. El enfoque funciona sobre un conjunto de observaciones y el historial de las mediciones odométricas. La posición del robot es directamente estimada por esta información alternando actualizaciones de movimiento.

O'Kane [47] presenta una técnica de localización con odómetros lineales y angulares, cuya configuración está compuesta por la posición y orientación en un entorno totalmente conocido al alternar rotaciones y traslaciones hacia enfrente, lo que se vuelve un problema de planeación en tiempo discreto en la información espacial del robot. La localización por odometría está limitado a conexiones simples, en entornos acotados por polígonos simétricos.

Santana et al. [48] proponen un sistema de localización donde el robot navega en un entorno desconocido, utilizando marcas de tierra. En la fase de predicción se utiliza un filtro de Kalman extendido, empleando el modelo odómetro del robot. Durante la fase de actualización, el enfoque propuesto usa parámetros de las líneas detectadas por la transformada de Hough para corregir el espacio del robot.

Seraji y Werger [17] presentan varios algoritmos para calcular la ubicación de un robot móvil. Los métodos *weight average with local priority* (WAL) y *peak with high neighbors* (PHN) muestran los mejores resultados en trayectorias de 13 metros (medidos en línea recta entre los puntos de inicio y fin), con tasas de error del 4% y 3.92% para PHN y WAL, respectivamente. El enfoque combina información de odometría y de compases electrónicos montados en el robot. Este esquema continuamente descompone todos los comandos de movimiento en segmentos de líneas cortos. El compás es empleado para orientar al robot hacia la propia dirección en el inicio de cada segmento, y corregir la desviación desde el inicio hasta el final. La distancia recorrida es calculada con odometría y la información del segmento es continuamente acumulada para proveer información de la posición. El error de ubicación es muy similar al obtenido con dicho enfoque.

Los trabajos mencionados se concentran en reducir los errores odométricos subyacentes al emplear métodos matemáticos o utilizando otros dispositivos para calcular la posición del robot; pero estos necesitan mayor poder de cómputo para procesar toda la información que adquieren.

Ningún otro trabajo aborda el ajuste de velocidad de la forma planteada en el presente trabajo, y tampoco existe algún escenario estándar que sirva de referencia para comparar las mejoras obtenidas en el cálculo de posición, sin embargo es importante resaltar que al aplicar la propuesta de ajuste de velocidad, dicho cálculo mejora respecto de las pruebas realizadas a velocidad constante.

6 Conclusiones

El resumen de conclusiones y contribuciones se enlistan a continuación.

En vista de la complejidad de la navegación de robots con ruedas, la red neuronal difusa utiliza un conjunto pequeño de reglas de inferencia si-entonces.

Los algoritmos de reconocimiento de texturas clasifican exitosamente las texturas del terreno. Los algoritmos reconocen exitosamente los patrones de líneas, puntos o bordes que se encuentren en su trayecto.

Los algoritmos implementados tanto de visión artificial como de navegación son computacionalmente de bajo costo y fácil de implementar.

El enfoque propuesto puede ser escalado para emplearse en vehículos comerciales.

El bajo costo computacional para procesar la información adquirida del entorno permite establecer que un carro, a determinada velocidad, pueda tener suficiente tiempo para reaccionar a los cambios de rugosidad en el terreno.

La red neuronal difusa permite hacer una meta-clasificación de las texturas usadas para reconocer la rugosidad de la superficie; esta permite ajustar la velocidad durante la navegación.

La experiencia humana para el reconocimiento de rugosidades y ajuste de velocidad son modelados al aplicar la red neuronal difusa, la cual es implementada para la navegación de robots.

Al aplicar el ajuste de velocidad considerando las características del terreno, se obtiene mayor precisión en el cálculo de la ubicación del robot utilizando odometría. Con el ajuste de velocidad, el derrape de las ruedas es reducida significativamente, de aquí que se mejora la precisión en la posición final; además, el riesgo de que robot sufra un accidente también disminuye.

7 Referencias

[1] H. Seraji y A. Howard, «Behaviour-based robot navigation on challenging terrain: a fuzzy logic approach",» *IEE Transactions on Robotics and Automation,* vol. 18, nº 3, pp. 308-321, 2002.

[2] Z. Sun, G. Bebis y R. Miller, «On-road vehicle detection: a review,» *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 28, nº 5, pp. 694-711, 2006.

[3] K. Konolige, M. Agrawal, M. Blas, B. Gerkey, J. Sola y A. Sundaresan, «Mapping, navigation and learning off-road traversal,» *Journal of Field Robotics,* vol. 26, nº 1, pp. 88-113, 2009.

[4] X. Dai, H. Zhang y Y. Shi, «Autonomous navigation for wheeled mobile robots - a survey,» *International Conference on Innovative Computing, Information and Control,* pp. 2207-2210, 2007.

[5] A. Seeni, B. Schäfer, B. Rebele y N. Tolyarenko, «Robot mobility concepts for extraterrestrial surface exploration,» *IEEE Aerospace Conference,* pp. 1-14, 2008.

[6] C. Ward y K. Lagnemma, «A dynamic-model-based wheel slip detector for mobile robots on outdoor terrain,» *IEEE Transactions on Robotics ,* vol. 24, nº 4, pp. 821-831, 2008.

[7] A. Lambert, D. Gruyer, G. Pierre y A. Ndjeng, «Collision probability assessment for speed control,» *International IEEE Conference on Intelligent Transportation Systems,* pp. 1043-1048, 2008.

[8] B. Browning, J. Bruce, M. Bowling y M. Veloso, «Skills, tactics and plays for multi-robot control in adversarial environements,» *Journal of Systems and Control Engineering,* vol. 219, nº 11, pp. 33-52, 2005.

[9] A. Brooks y K. Lagnemma, «Vibration-based terrain classification for planetary exploration rovers,» *IEEE Transactions on Robotics,* vol. 21, nº 6, pp. 1185-1191, 2005.

[10] D. Langer, J. Rosenblatt y M. Herbert, «A behavior-based system for off-road navigation,» *IEEE Transactions on Robotics and Automation,* vol. 10, nº 6, pp. 776-783, 1994.

[11] C. Ward y K. Lagnemma, «Speed-indeendent vibration-based terrain classification for passenger vehicles,» *Vehicle System Dynamics,* vol. 47, nº 9, pp. 1095-1113, 2009.

[12] U. Nehmzow, «Mobile robotics: A practical Introduction,» *Springer-Verlag,* 2000.

[13] R. Siegwart y I. Nourbaksh, «Introduction to Autonomous Mobile robots,» *MIT Press,* 2004.

[14] J. Zhou, J. Shi y X. Qu, «Statistical characteristics of landmark-based localization performance,» *International Journal of Advanced Manufacturing Technology,* vol. 46, nº 9-12, pp. 1215-1227, 2010.

[15] A. Martinelli, N. Tomatis y R. Siegwart, «Simultaneous localization and odometry self calibration for mobile robot,» *Autonomous robots,* vol. 22, nº 1, pp. 75-85, 2007.

[16] M. Jüngel, «Self-localization based on a short-term memory of bearings and odometry,» *IEEE/RSJ International Conference of Intelligent Robots and Systems,* pp. 2494-2499, 2007.

[17] H. Seraji y B. Werger, «Theory and experiments in smartnav rover navigation,» *Autonomous Robots,* vol. 22, nº 2, pp. 165-182, 2007.

[18] J. Jang, C. Sun y E. Mituzani, «Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence,» *Prentice Hall,* 1997.

[19] T. Ojala, M. Pietikäinen y T. Mäenpää, «Multiresolution gray-scale and rotation invariant texture classification with local binary patterns,» *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 24, nº 7, pp. 971-987, 2002.

[20] R. Haralick, K. Shanmugam y I. Dinstein, «Textural features for image classification,» *IEEE Transactions on Systems, Man and Cybernetics,* Vols. %1 de %2SMC-3, nº 6, pp. 610-621, 1973.

[21] M. Turk y A. Pentalnd, «Eigenfaces for recognition,» *Journal of Cognitive Neuroscience,* vol. 1, nº 3, pp. 71-86, 1991.

[22] H. Murase y S. Nayar, «Visual learning and recognitionof #-D object from appearance,» *International Journal of Compuer Vision,* vol. 14, nº 1, pp. 5-24, 1995.

[23] S. Nakamura, M. Faragalli, N. Mizukami, I. Nakatani, Y. Kunni y T. Kubota, «Wheeled robot with movable center of mass of traversing over rough terrain,» *International Conference on Intelligent Robots and Systems,* pp. 1228-1233, 2007.

[24] I. Halatci, C. Brooks y K. Lagnemma, «Terrain classification and classifier fusion for planetary exploration rovers,» *IEEE Aerospace Conference,* pp. 1-11, 2007.

[25] X. Song, Y. Wang, Z. Wu, C. Bu y Y. Chang, «Kinematics-based velocity estimation of lunar rovers,» *IEEE International Conference n Robotics and Biomimetics,* pp. 1568-1573, 2007.

[26] W. Dixon, T. Galluzo, G. Hu y C. Crane, «Adaptative velocity field motion of a wheeled mobile robot,» *International Workshop on Robot Motion and Control ,* pp. 145-150, 2005.

[27] K. M. D. Seiguchi y A. Inoue, «Obstacle avoidance and two wheeled mobile robot control using potential function,» *IEEE International Conference on Information Technology ,* pp. 2314-2319, 2007.

[28] W. Wang, L. Zhou, Z. Du y L. Sun, «Track-terrain interaction analysis for tracked mobile robot,» *IEEE/ASME International Conference on Advanced Intelligent Mechatronics,* pp. 126-131, 2008.

[29] K. Seluga, L. Baker y I. Ojalvo, «A parametric study of golf car and personal transport vehicle braking stability and their deficiencies,» *Accidental Analysis and Prevention,* vol. 41, nº 4, pp. 839-848, 2009.

[30] E. Krotkov y R. Hoffman, «Terrain mapping for a walking planetary rover,» *IEEE Transactions on Robotics and Automation ,* vol. 10, nº 6, pp. 728-739, 1994.

[31] M. Castelnovi, R. Arkin y T. Collins, «Reactive speed control system based on terrain roughness detection,» *IEEE International conference on Robotics and Automation ,* pp. 891-896, 2005.

[32] D. Stavens y S. Thrun, «A self-supervised terrain roughness estimator for off-road autonomous driving,» *conference on Uncertainly in Artificial Intelligence,* 2006.

[33] T. Cootes, G. Edwards y C. Taylor, «Active appearance models,» *IEEE Transactions on Pattern Analysis and Machine Intelligence ,* vol. 23, nº 6, pp. 681-685, 2001.

[34] D. Tsai y C. Tseng, «Surface roughness classification for castings,» *Pattern Recognition,* vol. 32, nº 3, pp. 389-405, 1999.

[35] M. Selewka, D. Dunlap, D. Shi y E. Collins, «Robot navigation in very cluttered environnements by preference-based fuzzy behaviors,» *Robotics and Autonomous Systems,* vol. 53, nº 3, pp. 231-246, 2008.

[36] A. Leonardis y H. Bischof, «Robust recognition using eigenimages,» *Computer Vision and Image Understanding,* vol. 78, nº 1, pp. 99-118, 2000.

[37] S. Nayar, S. Nene y H. Murase, «Subspace methods for robot vision,» *IEEE Transactions on Robotics and Automation ,* vol. 12, nº 5, pp. 750-758, 1996.

[38] L. Altamirano y M. Alvarado, «Non-uniform sampling for improved appearance-based models,» *Pattern Recognition Letters,* vol. 24, nº 1-3, pp. 521-535, 2002.

[39] J. Abonyi, «Cluster Analysis for Data Mining and System Identification,» *Springer-Verlag,* 2007.

[40] S. Liao y A. Chung, «Texture classification by using advanced local binary patterns and spatial distribution of dominant patterns,» *IEEE International Conference on Acoustics, Speech and Signal Processing ,* pp. 1221-1224, 2007.

[41] LEGO. [En línea]. Available: https://www.lego.com/es-es/mindstorms. [Último acceso: 2017].

[42] R. Arkin, «Integrating behavioral, perceptual and world knowledge in reactive navigation,» *Robotics and Autonomous Systems,* vol. 6, nº 1-2, pp. 105-122, 1990.

[43] I. Rekleitis, J. Bedwani y E. Dupuis, «Over-the-horizon, autonomous navigation for planetary exploration,» *IEEE/RSJ International Conference on Intelligent Robots and Systems,* pp. 2248-2255, 2007.

[44] L. Matthes, E. Gat, R. Harrison, B. Wilcox, R. Volpe y T. Litwin, «Mars microrover navigation: performance evaluation and enhancement,» *Autonomous robots,* vol. 2, nº 4, pp. 291-311, 1995.

[45] G. Ishigami, K. Nagatahi y K. Yoshida, «Path planning for planetary exploration rovers and its evaluation based on wheel slip dynamics,» *IEEE International Conference on robotics and Automation,* pp. 2361-2366, 2007.

[46] P. Kim, C. Park, Y. Jong, J. Yun, E. Mo, C. Kim, M. Jie, S. Hwang y K. Lee, «Obstacle avoidance of a mobile robot using vision system and ultrasonic sensor,» *International Conference on Intelligent Computing: LNCS 4681,* pp. 545-533, 2007.

[47] J. O'Kane, «Global localization using odometry,» *Conference of International Robotics and Automation,* pp. 37-42, 2006.

[48] A. Santana, A. Sousa, R. Brito, P. Alsina y A. Medeiros, «Localization of a mobile robot based in odometry and natural landmarks using extended kalman filter,» *International Conference on Informatics in Control, Automation and Robotics,* vol. 2, pp. 187-193, 2008.