

ROBOT NAVEGANTE SOBRE UN MODELO MÉTRICO Y TOPOLÓGICO DEL AMBIENTE

Fernando E. Conde T. Diana C. Mateus L. Daniel Otero F. Carlos Parra
Pontificia Universidad Javeriana
Departamento de Ingeniería Electrónica
Bogotá D.C.

Abstract

Este artículo describe la arquitectura del hardware utilizado por un robot móvil autónomo que planea rutas y se ubica en un ambiente conocido representado con un modelo topológico y métrico. El sistema integra desarrollos en alto y bajo nivel que le permiten al robot navegar por el ambiente desplazándose desde una posición conocida hasta cualquier lugar en la representación indicado por el usuario.

I. INTRODUCCIÓN

MACONDO es un robot móvil autónomo construido con el propósito de reunir en una plataforma simple las tareas concernientes a la navegación: localización, planeación de rutas y control [1].

Para lograr una navegación eficiente, el robot emplea una representación híbrida del ambiente (grilla-grafo) que extrae las mejores características de los modelos métricos y topológicos [2]. Esta representación es construida con ayuda de un programa ejecutado en un PC, y luego almacenada en la memoria del robot.

Con una representación del lugar y contando con que el ambiente es estático, el robot está en capacidad de planear rutas. Para ello utiliza dos algoritmos: A* para buscar el camino en la grilla y Dijkstra [4] para trazar rutas en el grafo.

Cuando la ruta ha sido planeada, el robot se desplaza siguiendo las indicaciones del planeador y haciendo uso del método de localización de Markov para corregir los errores introducidos por el movimiento de la plataforma [1].

Finalmente, cuenta con rutinas que evitan las colisiones en caso de que aparezca algún obstáculo inesperado.

Todas estas funciones se llevan a cabo en la plataforma y sin otra intervención humana que la especificación del destino.

II. MACONDO

MACONDO es un robot de tracción diferencial soportado por dos ruedas frontales acopladas directamente al eje de salida de un par de motores, y una tercera rueda que gira libremente [6]. En el extremo de cada eje se encuentra un encoder óptico que entrega la información de odometría. El sistema sensorial tiene además, seis sensores de ultrasonido [3],[4] distribuidos alrededor de la plataforma (dos al frente y dos a cada lado). Un teclado y un display se alojan sobre la cubierta (ver

Figura 1).

Figura 1. Macondo



El robot cuenta con dos módulos: el principal, encargado del procesamiento de la planeación y la localización (incluyendo el manejo de sensores y el tratamiento de la información sensorial), y el secundario, que tiene como misión el control de los motores, la odometría y la interfaz con el usuario. Entre los dos módulos se establece una comunicación serial que permite coordinar sus acciones (ver Figura 2).

El módulo principal está conformado por un microcontrolador Motorola GP32, una NVRAM, y un circuito encargado de multiplexar las señales de la tarjeta Polaroid 6500 que controla los sensores de ultrasonido.

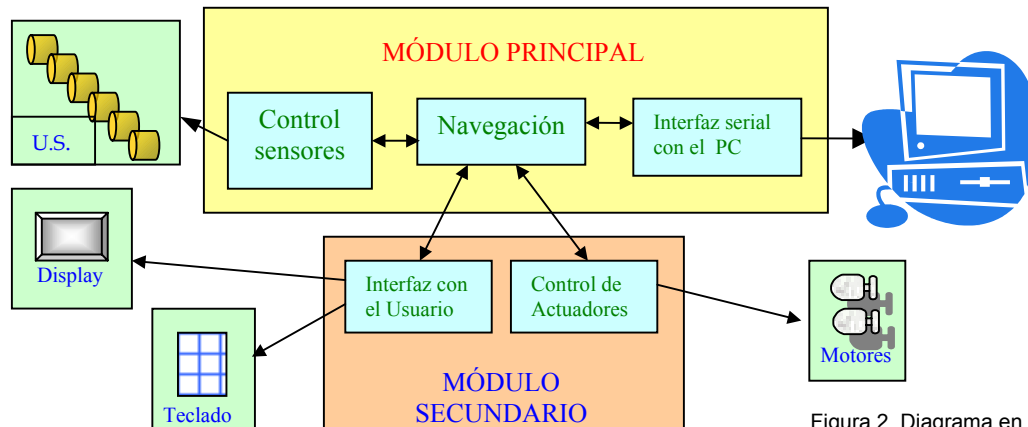


Figura 2. Diagrama en bloques.

Por su parte, el procesamiento del módulo secundario está a cargo de un PIC16F873 que maneja la interfaz con el usuario (teclado de membrana y un display de cristal líquido) y controla la ejecución de las primitivas que recibe del módulo principal.

III. NAVEGACIÓN

Para que el robot pueda navegar en el ambiente debe conocerlo. En MACONDO ésta tarea es efectuada cargando en la memoria del robot la representación cartográfica creada por el usuario con un programa ejecutado en un PC, a través de una conexión serial entre el computador y el robot.

Teniendo una representación almacenada y con la plataforma situada en el punto de partida indicado por el usuario durante la creación de la representación, la interfaz hombre-máquina permite dar a MACONDO las coordenadas del punto de destino.

El bloque de planeación se encarga entonces de encontrar la ruta hasta el punto especificado y de traducirla en primitivas de movimiento que sean comprendidas por el módulo secundario.

Durante la ejecución de las primitivas, la plataforma puede desviarse de su camino; el bloque de localización es capaz de estimar la posición del robot y corregirla si encuentra algún error respecto a la esperada, llevando a la plataforma de vuelta a la ruta planeada.

Para estimar su posición el robot se vale de la información de odometría que obtiene con dos encoders ópticos acoplados a los ejes de sus motores y de un grupo de seis sensores

de ultrasonido distribuidos alrededor de la plataforma (dos al frente y dos a cada lado).

1. Localización

Para hablar de localización es necesario definir tres conceptos relacionados con la posición del robot:

Posición Real: Es la posición que ocupa el robot en el ambiente.

Posición Esperada: Dadas las primitivas de movimiento ejecutadas por los motores y la información de odometría, el robot espera ocupar una posición en la representación del ambiente.

Posición Estimada: Es la posición que el robot estima a partir de la información sensorial. Este resultado es lo más próximo que tiene el robot a su posición real.

La localización es el método utilizado para hacer coincidir las tres posiciones del robot.

Con el fin de corregir errores en la ejecución de una ruta Macondo utiliza el método de localización de Markov [1] conformado por dos modelos: movimiento y perceptual.

Cada modelo posee una expresión que lo describe. El modelo de movimiento está dado por:

$$P((x',y')|a,(x,y))$$

Indica la probabilidad de encontrar la plataforma en la posición (x',y') dado que ejecutó la primitiva de movimiento a partiendo de la posición (x,y) .

El modelo perceptual está descrito por:

$$P(s|(x,y))$$

Que es la probabilidad de medir s (s es un vector de seis posiciones, una posición por cada sensor) dado que el robot está en la posición (x,y) .

El modelo perceptual se vale de los sensores de ultrasonido de Macondo para dar una ubicación tentativa de la plataforma dentro de una región que la contenga. Este método simple daría resultado, si las medidas de distancia de los sensores fueran siempre diferentes para todos los puntos de la región; como no es posible garantizar esta condición se debe buscar una forma de diferenciar los lugares con medidas similares. Para resolver el inconveniente, se utiliza el modelo de movimiento.

El modelo de movimiento busca precisamente modelar el comportamiento mecánico de la plataforma al generar una atmósfera de incertidumbre alrededor de la posición esperada de Macondo, pero a diferencia del modelo perceptual, solo una posición tiene la mayor probabilidad de estar ocupada por la plataforma; con la información de ambos métodos se determina una posición estimada bastante cercana a la real.

Cada vez que Macondo termina de ejecutar una primitiva, el modelo de movimiento actualiza la matriz de Markov teniendo en cuenta cuatro factores: la matriz de estado anterior, el incremento proporcional de la incertidumbre respecto al tiempo, el movimiento ejecutado después de la última actualización y la información cartográfica del lugar. El modelo perceptual tiene en cuenta la información cartográfica y las medidas de localización; actúa cuando la posición esperada es un nodo o el destino de una ruta.

Ambos modelos calculan la probabilidad de encontrar a Macondo en cualquier posición de la matriz de Markov, la cual posee dimensiones de grilla y tiene un área de 21×21 celdas ($2,1 \text{m} \times 2,1 \text{m}$).

La matriz está centrada alrededor del punto donde debería encontrarse la plataforma, por lo tanto cada posición de la matriz equivale a una posición (x,y) en el mapa, que cambia de acuerdo con el desplazamiento del robot sobre el ambiente. Se puede entender la matriz como una sombra cuadrada que sigue a Macondo, pero manteniendo su centro en la posición esperada.

Modelo De Movimiento

El modelo de movimiento cuenta con tres matrices de transición para realizar la actualización. Estas contienen las probabilidades de transición del sistema.

Las matrices de transición son modelos estadísticos de las primitivas de movimiento que indican que tan probable es que la plataforma ocupe determinada posición de grilla después de haber ejecutado cierta acción dado que partió de un punto específico.

Ya que mecánicamente no es lo mismo ejecutar un avance que un giro hacia la izquierda o a la derecha, es necesario el uso de distintas matrices de transición que modelen este comportamiento.

El procedimiento que se realiza para la actualización de la matriz de Markov no cambia de acuerdo a la acción ejecutada, se mantiene y es el siguiente:

Se revisa cada posición de la matriz, si la probabilidad encontrada es cero o la posición corresponde a un obstáculo no se realiza cálculo alguno y se asigna cero, sino, se multiplica cada valor de la matriz de transición por el valor encontrado en la posición revisada y se suma a la matriz que está siendo actualizada el resultado obtenido en la posición correspondiente.

El ejemplo ilustrativo de la siguiente página muestra la actualización de la matriz de Markov en el caso de un avance.

Modelo Perceptual

El modelo perceptual utiliza dos vectores de medidas para trabajar: un vector de medidas virtuales y uno de medidas reales. El vector de medidas reales se actualiza cada vez que el modelo perceptual actúa y es el vector s de la regla de actualización de este método.

A diferencia de las medidas reales, que se obtienen por hardware, las medidas virtuales se calculan por software. En este caso se ubica virtualmente al robot en cualquier posición válida dentro del mapa almacenado en memoria y teniendo en cuenta el tamaño de la plataforma y la ubicación de los sensores en ésta, se cuenta el número de posiciones de grilla desde cada sensor hasta el obstáculo o pared más cercano. De esta forma se puede obtener un vector de seis

posiciones que contiene lo que mediría Macondo desde cualquier ubicación válida en el ambiente.

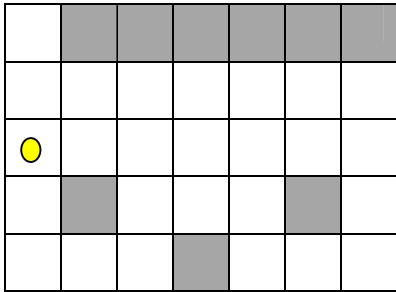


Figura 3. Grilla de un sector, el robot está simbolizado por el punto amarillo.

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

Figura 4. Matriz de estado equivalente centrada alrededor de la posición esperada.

0	0	0	0	0	0	0
0	0	0	.0625	.0078	0	0
0	0	0	.7929	.0625	0	0
0	0	0	.0625	.0078	0	0
0	0	0	0	0	0	0

Figura 7. Matriz de estado equivalente actualizada centrada alrededor del destino esperado. Es ya apreciable el aumento de la incertidumbre con el tiempo a partir de la matriz de estado inicial.

En el ejemplo de la Figura 8 se obtendría el vector de medidas virtuales (3,3,4,3,2,2).

Para determinar la probabilidad de encontrar a Macondo en alguna posición de la matriz de Markov, se obtiene el vector de medidas virtuales correspondiente y se calcula el valor absoluto de la diferencia con el vector de medidas reales. Para actualizar toda la matriz, se repite el proceso con cada posición; siempre y cuando sea una ubicación válida (que no corresponda a un

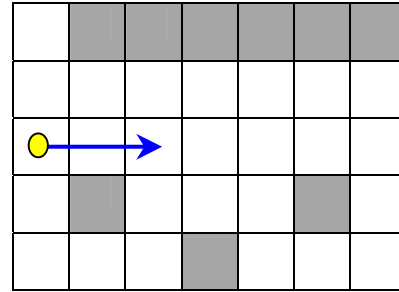


Figura 5. Desplazamiento del robot.

0	.0625	.0078
0	.7929	.0625
0	.0625	.0078

Figura 6. Matriz de transición correspondiente a un avance. La flecha apunta en la dirección del avance.

obstáculo y esté dentro del mapa), en caso contrario se asigna cero. La magnitud de esta diferencia y la probabilidad de estar en esa ubicación se comportan de manera contraria, conforme aumenta la diferencia la probabilidad disminuye y viceversa.

Es conveniente anotar que para los cálculos de probabilidades se cuenta con dos matrices, una para el estado anterior y otra para el estado actual. Ambos modelos (perceptual y movimiento) hacen uso de ellas

para la alteración de la matriz que en ese momento se considere la del estado actual.

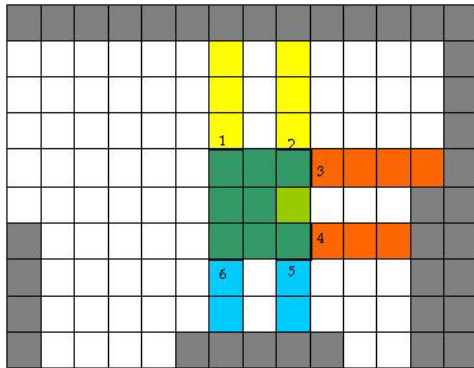


Figura 8. La región verde mar del mapa representa a Macondo, la verde lima la posición esperada, las líneas resaltadas muestran la posición de los sensores; la región gris simboliza los obstáculos, los colores restantes son las posiciones de grilla entre cada sensor y el obstáculo más cercano. Los sensores están numerados según su posición en el vector de medidas virtuales.

Confiabilidad

La precisión de los sensores no se mantiene en todo el rango de operación, disminuye conforme aumenta la distancia que se va a medir. Este comportamiento exige de un criterio en la toma de medidas si se quiere tener una información confiable.

Una forma de llegar a esto es por medio de la estadística. Se midieron distancias a intervalos de 50cm, en cada etapa se realizaron cincuenta medidas.

Los resultados mostraron que hasta los 2,5m las medidas eran exactas en su totalidad; entre los 2,5m y los 5m, el 75% de las medidas presentaban errores de ± 30 cm; más allá, por lo general los resultados eran erróneos.

Gracias a este período de observación se estableció un índice de confiabilidad en la toma de medidas. Una medida que se encuentre en el intervalo de 0m a 2,5m se le da una confiabilidad del 100%; entre los 2,5m y los 5m, 25%; mayor a 5m, 0%.

Como a partir de la medida obtenida con un sensor (medida real) no siempre se puede determinar en qué rango se encuentra, se debe conocer de antemano que distancia medirá el transductor para asignar la confiabilidad a la medida; para lograr esto se hace uso de las medidas virtuales.

Cabe anotar que en el caso de un vector de medidas reales se puede conocer su confiabilidad al promediar los índices de cada medida.

Ponderación

Para que el método de localización esté completo, hace falta un procedimiento que utilice la información obtenida por ambos modelos y logre de la manera más aproximada su fin: la unión de las tres posiciones que posee un robot.

El método utilizado en este trabajo es la ponderación o suma ponderada. Su objetivo es darle un peso a cada modelo en el resultado final.

Cuando el modelo perceptual actualiza la matriz actual, la matriz de estado anterior contiene la información del modelo de movimiento. La ponderación no es más que una suma de matrices, donde cada una está multiplicada por su peso respectivo.

Los pesos dependen de la confiabilidad del vector de medidas reales y están dados por las siguientes ecuaciones:

$$\text{Peso modelo perceptual} = \text{Confiabilidad} \times 0.6$$

$$\text{Peso modelo de movimiento} = 1 - \text{Confiabilidad} \times 0.6$$

Inicialización de la matriz de Markov

Antes de comenzar a navegar, Macondo debe ser ubicado en su estación base. Como no se puede garantizar que se encuentre exactamente en esa posición, la matriz de Markov debe considerar esta incertidumbre. Para esto se hace uso de los modelos que conforman la localización.

Una matriz, inicializada en ceros excepto la posición que corresponde a la estación base (a la cual se le asigna probabilidad uno), es la información que aporta el modelo de movimiento; la otra matriz (previamente inicializada con ceros en todos sus puntos) es actualizada por el modelo perceptual. Después de realizar la ponderación, se tiene una matriz que considera la incertidumbre inicial sobre la ubicación de Macondo antes de comenzar su navegación.

Relocalización

El resultado de la localización es la posición estimada, la cual es la ubicación dentro de la

matriz de Markov con la mayor probabilidad, y por lo general, la más cercana a la real. Si la posición esperada es diferente a la posición estimada una rutina para corregir la posición de la plataforma es necesaria.

A diferencia de la posición esperada, que se mantiene fija en el centro de la matriz, la posición estimada puede estar en cualquier punto de esta. Conociendo la posición de ambas se puede determinar la secuencia de movimientos que debe ejecutar la plataforma para volver a la posición esperada.

2. Representación Híbrida

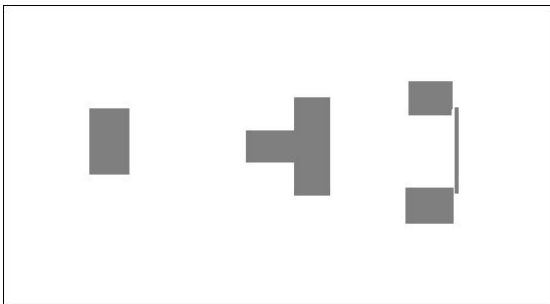


Figura 9. Mapa del lugar

Para la construcción de la representación cartográfica que utiliza Macondo se recurrió a los métodos más simples de las tendencias métrica y topológica, a saber: la grilla uniforme y el grafo. Su elaboración está a cargo de un programa desarrollado en Builder C++ y ejecutado en un PC.

El método híbrido utilizado construye una grilla uniforme a partir de un mapa elaborado por el usuario y luego, divide todo el ambiente en zonas que conecta a través de un grafo.

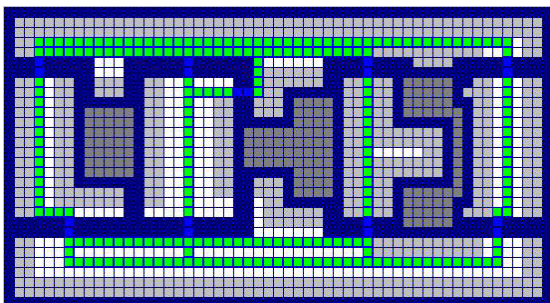


Figura 10. Representación del ambiente: grilla y grafo.

En la representación métrica el robot (cuya plataforma mide 3 celdas X 3 celdas) es simbolizado como una celda. Los inconvenientes que ésta simplificación conlleva se eliminan incluyendo en la

representación *celdas prohibidas*: celdas originalmente libres que se evitan en la planeación porque presentan algún riesgo para la navegación. Las celdas prohibidas se extienden 3 celdas más allá de los obstáculos y las paredes del ambiente.

Las características métricas de la grilla y las topológicas del grafo se combinan y complementan en la representación utilizada para dar lugar a un modelo híbrido que trae grandes ventajas para la navegación:

- La grilla permite distinguir dos lugares topológicamente parecidos en el grafo por sus coordenadas, sin necesidad de faros externos, favoreciendo la tarea de localización.
- En la planeación es la grilla y no el grafo quien determina el número de destinos posibles de forma que se conserva una alta resolución.
- El grafo permite planear rutas con gran velocidad y poca memoria pues el espacio de búsqueda es reducido.
- La ubicación de los nodos del grafo en los puntos centrales de las uniones entre zonas es estratégica tanto en la planeación como para la localización. En la planeación, permiten acercarse a las rutas óptimas con costo porque en estos puntos convergen muchas de las rutas óptimas cuando no hay división en zonas. Por otra parte, como se mencionó en la sección anterior, los límites de las zonas portan información topológica relevante que colabora con la localización.

3. Planeación de Rutas

La planeación de rutas exige además de una representación, la especificación de un origen y un destino. Como se mencionó previamente, el primer origen del robot es siempre la Estación Base, una posición indicada por el usuario dentro de la representación. El destino se ingresa mediante un pequeño teclado ubicado en la plataforma.

Se requieren tres coordenadas para ubicar un punto dado en la representación. Las dos primeras son las coordenadas cartesianas (x,y) que sirven para ubicar el punto en la grilla; la tercera coordenada corresponde a la zona en la que se encuentra esta posición y además, permite ubicar el punto respecto al grafo.

Dados un origen y un destino es posible que sus celdas pertenezcan: a la misma zona, zonas adyacentes o a zonas diferentes no adyacentes. Cada caso posible es tratado de forma diferente.

Cuando origen y destino pertenecen a la misma zona, el trazado de rutas se hace con el algoritmo A* y se ignora por completo cualquier información del grafo.

En los otros dos casos, la planeación es un poco más compleja. En primer lugar, se buscan todos los nodos que conecten la zona del punto origen con el grafo y se traza una ruta con A* desde la celda origen hasta cada uno de estos nodos. Luego, se hace un tratamiento similar con los nodos de la zona destino y la celda destino. Los resultados, se guardan con el mismo formato de las rutas al origen. Finalmente, se traza la ruta entre cada posible par de nodos origen-destino con Dijkstra, se evalúa el costo total (incluyendo el de las rutas trazadas con A* hasta y desde el grafo) y se escoge la de menor costo.

3.1. Simulador

El programa de alto nivel utilizado para la creación de la representación topológica fue desarrollado para que después de haber creado la grilla y el grafo, se pudiera simular la planeación de rutas.

IV. CONCLUSIONES Y RESULTADOS

Se ha desarrollado un sistema que planea y traza rutas de manera eficiente en ambientes complejos, aprovechando las ventajas de una representación híbrida.

Es posible darle a un robot la autonomía suficiente para planear rutas y la capacidad de ubicarse en un ambiente conocido con poco hardware y un sistema sensorial sencillo.

Gracias a la localización el robot llega a su destino en el 90% de los casos. Para evaluar esto, se consideró aceptable un error de 1 celda a la redonda.

V. REFERENCIAS

- [1] BURGARD, Wolfram; CREMERS, Armin; FOX, Dieter; HÄHNEL, Dirk; LAKEMEYER, Gerhard ; SCHULZ, Dirk; STEINER, Walter; THRUN, Sebastian.

The Interactive Museum Tour-Guide Robot.

- [2] THRUN, Sebastian; BÜCKEN, Arthur. Integrating grid-based and topological maps for mobile robot navigation. En Proceedings of the AAAI Thirteenth National Conference on Artificial Intelligence, Portland, Oregon, 1996.
- [3] POLAROID. Technical Specifications for 600 Series Instrument Grade Electrostatic Transducer.
- [4] POLAROID. Technical Specifications for 6500 Series Sonar Ranging Module.
- [5] BECERRA, César. Estructuras de Datos en C. Editorial Kimpres Ltda. Santafé de Bogotá, Colombia. 1995.
- [6] Borenstein, J; Everett, H.R.; Feng, L. Where am I? Systems and Methods for Mobile Robot Positioning. March 1996