

Robot Path Planning

David Medel - Cesar Quiroz - José Valencia
 Universidad Técnica Federico Santa María

Vicuña Mackenna 3939, San Joaquín, Región Metropolitana, Chile
 david.medel@sansano.usm.cl
 cesar.quirozm@sansano.usm.cl
 jose.valencia@sansano.usm.cl

Resumen—Existen una gran cantidad de problemas de planificación de camino, uno de ellos es el Robot Path Planning que consiste en encontrar la ruta más corta en llegar de un punto a otro en un mapa lleno de obstáculos. Dicha ruta más corta se puede encontrar modelando el problema como un problema de programación lineal y otros métodos tal como se muestra en el presente informe.

Palabras Claves—Ruta óptima, path planning, robot, método potencial, programación lineal.

I. INTRODUCCIÓN

La robótica ya es un área bien establecida[1] y ampliamente adoptada en la industria, aún así hay muchos desafíos que quedan por resolver, entre estos podemos destacar el problema de Robot Path Planning, el cual consiste en encontrar una trayectoria desde un punto inicial "S" hasta un punto final "F", evadiendo los obstáculos "O" que se encuentre en el camino, los cuales poseen una forma y posición conocida.

Por lo tanto el robot necesita tener un completo entendimiento del terreno y sus características, para estar en condiciones de navegar de manera segura. Debido a esto es de gran importancia incorporarle sensores que permitan identificar y evadir obstáculos.

Este artículo se centrará en identificar y explicar algoritmos que permitan avanzar en la optimización del camino que debe seguir un robot para llegar a su objetivo

II. ESTADO DEL ARTE

El Robot Path Planning es un problema NP-completo que tiene como objetivo la obtención de la ruta mínima entre un determinado origen y un destino, al día de hoy se han planteado diversos métodos para resolver este problema, cada método tiene como sus componentes

el espacio de trabajo y el espacio de configuraciones. El espacio de trabajo es el lugar donde se encuentran tanto el robot, obstáculos y objetivo, el espacio de configuraciones es todos los estados que puede tener el robot

Tres de estos métodos tradicionales son:

1. Métodos de descomposición de celdas. [4]
2. Métodos de mapas de camino (road map).[4]
3. Métodos de campo potencial.[4]

Para el proyecto se opta por el método de campo potencial que también es conocido como el método local[4], debido a que solo calcula el movimiento del robot para un estado inicial dado (ej: coordenada de inicio fin y obstáculos), en este método el plano donde se trabaja se asimila a un campo de fuerzas, dando valores diferentes a puntos dentro del plano. El objetivo se considera como una carga con polaridad contraria al robot y el robot tiene la misma polaridad que los obstáculos (recordar que cargas opuestas se atraen y cargas iguales se repelen), Por otro lado, se ha identificado que es posible aterrizar el problema al tratarlo como un Modelo de Programación Lineal, el cual considera una matriz con las posiciones por las cuales podría desplazarse, así como los obstáculos y las celdas por las cuales se desplaza el robot.

Existen una gran cantidad de problemas ligados a la Path Planning, algunos otros ejemplos son trayectorias de aviones, los GPS y las posiciones a seguir de un brazo robotico para llegar a otro punto, estos son solo algunos de los muchos ejemplos que el path planning buscan solucionar.

III. MODELO MATEMÁTICO O LP

El primer modelo a analizar corresponde al método potencial, para el cual primero se debe obtener la lectura de los datos(inicio, fin, las posiciones de los obstáculos y el tamaño del plano).

Una vez obtenido el tamaño del plano se crea una matriz para simular el plano llena de ceros.

Cada elemento de la matriz corresponderá a un área determinada, por ejemplo si el plano tiene un área de $20 \cdot 20cm^2$ cada elemento de la matriz corresponderá a una área de $1 \cdot 1cm^2$. Esta área se puede ajustar si el área del trabajo no es cuadrada.

Utilizando el método de campos potenciales se pueden encontrar trayectorias desde un punto A a otro B, esto se logra emulando los obstáculos con cargas positivas, el objetivo con una carga negativa y el robot como carga positiva, de esa forma el robot se repele con los obstáculos y es atraído hacia el objetivo.

Este método se procede en base a la siguiente planificación:

- Considerar a cada obstáculo genera una fuerza repulsora, mientras que la posición de objetivo es una fuerza de atracción.
- Calcular el vector de fuerzas resultante.
- Calcular una nueva posición para el robot como resultado de aplicar una fuerza aceleradora.
- Regresar al primer paso

El campo de potencial diferencial se construye sumando el campo de la meta, U_g , y el campo de los obstáculos, U_o :

$$U(q) = U_g(q) + \sum U_o(q) \quad (1)$$

A partir del campo se construye un campo de fuerzas artificiales resultante:

$$F = -\nabla U(q) = \frac{dU}{dx} \quad (2)$$

$$F = -\nabla U(q) = \frac{dU}{dy} \quad (3)$$

Ya construido el campo de fuerzas, el robot se mueve en la dirección de la fuerza local. Obteniendo así un esquema que posee un plan de cualquier punto del espacio al final.

Para obtener las fuerzas hay que modelar las funciones de potencial de la meta y obstáculos calculando el potencial para cada punto del espacio libre:

Meta:

$$U_g(q) = k_1 \cdot \text{dist}(q, \text{meta})^2 \quad (4)$$

Obstáculo:

$$U_o(q) = k_2 \cdot \text{dist}(q, \text{obstaculo})^{-1} \quad (5)$$

Donde la función $\text{dist}(x,y)$ representa la distancia.
 $\text{dist}(x,y) = \sqrt{(x - x_a)^2 + (y - y_a)^2}$

Creación del potencial alrededor del punto de llegada

Se le asigna al punto de llegada un valor numérico igual a "0" y a partir de este se generan valores numéricos ascendentes hasta cubrir el total de la pista.

Realizando esta configuración en la pista, se implementa la fuerza de atracción con lo cual se asegura que desde cualquier punto se converge al punto de llegada.

Creación del potencial alrededor de un obstáculo

El obstáculo posee un valor mayor que todos, generando un potencial descendente alrededor del obstáculo, considerando así zonas de riesgo los alrededores de este evitando que el móvil(robot) sufra alguna colisión.

Creación del potencial en la orilla de la pista

Esta es la forma de asegurarse de que el móvil jamás salga del área de trabajo. Es por eso que los bordes del área de trabajo se representan con un potencial grande.

Considerando lo anterior se presenta el siguiente ejemplo:

Parámetros iniciales:

Posición inicial: $(x,y) = (5,5)$

Posición final: $(x,y) = (15,15)$

Numero de obstáculos = 1

Posición del obstáculo: $(x,y) = (9,9)$

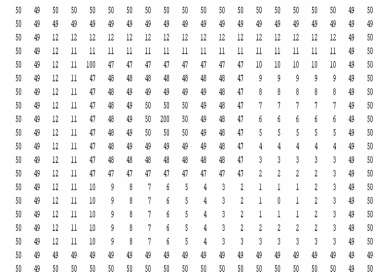


Figura 1. Representación gráfica

De esa forma el robot se moverá de su posición a la que lo deje más cerca de objetivos y que al mismo tiempo tenga un potencial menor o igual al que tiene, y así sucesivamente.

Finalmente, el camino óptimo es el siguiente:

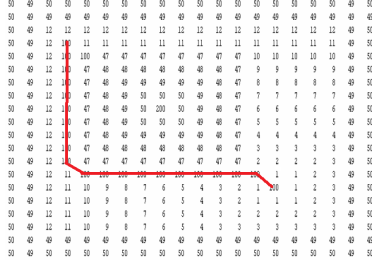


Figura 2. Ruta óptima

El modelo presentado resulta ser un poco confuso y escaparse un poco del concepto de modelamiento matemático, es por esto que se llevo a cabo la elaboración del siguiente modelo de programación lineal.

III-A. De forma general:

Comenzamos creando una matriz de orden $N \times N$ la cual representará el espacio donde se desplazará el robot, indicando el inicio de donde comienza su trayecto, los obstáculos y el final.

$$\begin{pmatrix} x_{11} & \cdots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{nn} \end{pmatrix}$$

Figura 3. Matriz general

Esta construcción tendrá las siguientes condiciones iniciales.

- Se debe conocer las posiciones de los obstáculos.
- El final (meta) siempre será en x_{nn} .
- El comienzo será en x_{11} .

Variables:

$$x_{ij} = \begin{cases} 1 : & \text{robot pasa por } ij \\ 0 : & \text{robot no pasa por } ij \end{cases}$$

Función Objetivo:

$$\text{Min } z = \sum_{i=1}^n \sum_{j=1}^n x_{ij}$$

Restricciones

- Condiciones para obstáculos

$$x_{ij} = 0 \text{ Para cada obstaculo}$$

$$x_{11} = 1 \text{ Inicio}$$

$$x_{nn} = 1 \text{ Final}$$

$$x_{12} + x_{21} = 1 \cdot \text{Extras}$$

$$x_{n-1,n} + x_{n,n-1} = 1 \cdot \text{Extras}$$

- Condiciones para posición

$$2x_{ij} \leq x_{i-1,j} + x_{i+1,j} + x_{i,j-1} + x_{i,j+1}$$

Dominio

$$0 < i \leq n$$

$$0 < j \leq n$$

$$\forall x_{ij} \in \{0, 1\}$$

III-B. Un caso en específico

En el presente documento resolveremos el problema de *Robot Path Planning* para un caso en específico que se describirá a continuación.

	1	2	3	4	5
1	O		x	x	x
2					
3	x	x			
4	x	x	x		
5	x	x	x		Δ

Donde:

O : Robot

Δ : Meta

X : Obstaculos

Además el robot solo se puede mover en dirección arriba, abajo, izquierda, derecha.

Variables: Descritas en el caso general.

Función Objetivo

$$\text{Min } z = \sum_{i=1}^5 \sum_{j=1}^5 x_{ij}$$

Restricciones

$$\text{Obstáculos: } x_{31} = x_{41} = x_{51} = x_{13} = x_{23} = x_{14} = x_{24} = x_{34} = x_{15} = x_{25} = x_{35} = 0$$

Inicio:

$$x_{11} = 1$$

Final:

$$x_{55} = 1$$

Posición:

$$\begin{array}{ll}
2x_{12} \leq x_{11} + x_{22} & x_{11} + x_{22} \geq 1 \\
2x_{21} \leq x_{11} + x_{22} & x_{11} + x_{22} \geq 1 \\
2x_{22} \leq x_{12} + x_{21} + x_{32} & x_{12} + x_{21} + x_{32} \geq 1 \\
2x_{32} \leq x_{42} + x_{22} + x_{33} & x_{42} + x_{22} + x_{33} \geq 1 \\
2x_{42} \leq x_{52} + x_{43} + x_{32} & x_{52} + x_{43} + x_{32} \geq 1 \\
2x_{52} \leq x_{42} + x_{53} & x_{42} + x_{53} \geq 1 \\
2x_{33} \leq x_{32} + x_{43} & x_{32} + x_{43} \geq 1 \\
2x_{43} \leq x_{33} + x_{42} + x_{53} + x_{44} & x_{33} + x_{42} + x_{53} + x_{44} \geq 1 \\
2x_{53} \leq x_{52} + x_{43} + x_{54} & x_{52} + x_{43} + x_{54} \geq 1 \\
2x_{44} \leq x_{43} + x_{54} + x_{45} & x_{43} + x_{54} + x_{45} \geq 1 \\
2x_{54} \leq x_{53} + x_{55} + x_{44} & x_{53} + x_{55} + x_{44} \geq 1 \\
2x_{45} \leq x_{44} + x_{55} & x_{44} + x_{55} \geq 1
\end{array}$$

Extras

$$\begin{array}{l}
x_{54} + x_{45} = 1 \\
x_{12} + x_{21} = 1
\end{array}$$

IV. REPRESENTACIÓN Y ENTORNO DE EJECUCIÓN

Debido a la dificultad presente en el modelo de método potencial y a las herramientas que se utilizan para llevar a cabo su implementación, se decidió elegir otro camino para resolver el presente problema de seleccionar la ruta óptima de un robot.

El modelo de programación lineal resultó ser el más practico debido a que existen herramientas digitales accesibles a todo público(como Lingo) que permiten resolver un modelo lineal con una entendible interpretación de la solución.

El modelo presentado en la sección anterior se desarrolló y ejecutó en el software *Lingo* en su versión 17.0.79 (12 Junio 2018) para Mac64.

El ordenador que compiló dicho código corresponde a un Macbook Air 2017 con sistema operativo macOS Mojave.

V. RESULTADOS

Según lo expuesto en la sección de entorno de ejecución, Lingo nos facilitó el desarrollo y solución del modelo planteado, entregando la información que se puede observar en la Figura presente más adelante. De esta figura se puede apreciar que las variables por la cual deberá pasar el robot corresponden a:

$x_{11}, x_{12}, x_{22}, x_{32}, x_{33}, x_{43}, x_{53}, x_{54}, x_{55}$

Las cuales otorgan un valor a la función objetivo correspondiente a 9, que en este caso sería la mínima cantidad de movimientos que tiene que realizar el robot para llegar su destino sin pasar por los obstáculos,

considerando el inicio y el término.

Variable	Value	Reduced Cost
X11	1.000000	0.000000
X12	1.000000	1.000000
X13	0.000000	0.000000
X14	0.000000	0.000000
X15	0.000000	0.000000
X21	0.000000	1.000000
X22	1.000000	1.000000
X23	0.000000	0.000000
X24	0.000000	0.000000
X25	0.000000	0.000000
X31	0.000000	0.000000
X32	1.000000	1.000000
X33	1.000000	1.000000
X34	0.000000	0.000000
X35	0.000000	0.000000
X41	0.000000	0.000000
X42	0.000000	1.000000
X43	1.000000	0.000000
X44	0.000000	1.000000
X45	0.000000	1.000000
X51	0.000000	0.000000
X52	0.000000	1.000000
X53	1.000000	0.000000
X54	1.000000	1.000000
X55	1.000000	0.000000

Figura 4. Solución entregada por Software:Lingo

Dicha solución representada gráficamente en el mapa queda de la siguiente forma.

	1	2	3	4	5
1	O		x	x	x
2					
3	x	x			
4	x	x	x		
5	x	x	x		Δ

Tabla I
REPRESENTACIÓN DE CAMINO ÓPTIMO

VI. BENCHMARK O COMPARACIÓN CON RESULTADOS EN LITERATURA

El modelo presentado y analizado en el presente documento se basó indicando que el robot solo puede desplazarse horizontalmente y verticalmente, por lo cual un modelo que considere movimientos en diagonales, o no tan segmentados en casillas, pueden obtener un mejor óptimo. Lo anterior presentaba una dificultad mayor al momento de plantear nuestro modelo de programación lineal, el cual buscando en internet o similares resulta muy difícil de hallar, debido a que asume estos supuestos que encapsulan a una representación que en la vida real no será la más óptima, sin embargo, existen varias formas de enfrentarse al problema de *Robot Path Planning* entre ellas y la que

más se asemeja a los resultados presentados, es una solución con programación dinámica. Tal solución posee varios algoritmos sobre como encontrar un óptimo, entre ellos Bellman-Ford y Kwok-Driessen[2].

Al observar este método de programación dinámica se encuentran similitudes, como por ejemplo, en el algoritmo de Kwok-Driessen se parte creando una matriz de $N \times N$ donde a cada elemento de la matriz se le asocia un estado del entorno según su localización dentro del mismo, correspondiente a 0 si el estado es prohibido o 1 si el estado es libre [3], cuya interpretación es similar a la presentada en el modelo de programación lineal elegido para solucionar el problema de *Robot Path Planning* dispuesto en este escrito.

Otro método y tal como se señaló en secciones anteriores corresponde al método potencial, el cual puede resultar más preciso obteniendo trayectorias en tiempo real a partir del plano (campo de fuerza), ayudando al control y planificación de caminos, por ejemplo, útil cuando los obstáculos pueden estar en movimiento. Dicho modelo también detalla una matriz inicial, pero modificable con el tiempo, debido a que representa numéricamente el campo potencial desde el robot hacia el punto donde puede o no haber un obstáculo.

Sin embargo, el método tiene una gran desventaja, la aparición de mínimos locales, es decir, este método funciona dando al punto objetivo un valor "0", pero que ocurre cuando el campo de fuerzas (plano donde el robot se mueve) existen otros puntos donde la fuerza se anulen, pueden llevar al robot a confundirse y no llegar a el objetivo.

En cuanto a calidad de solución esta depende del algoritmo que se utilice para solucionar el problema de *Robot Path Planning*, si se consideran diversos factores, entre otros. De los métodos analizados y vistos el más efectivo cuanto a solución óptima y que considera movimientos no solamente rectilíneo es el de programación lineal, utilizando el algoritmo de Bellman-Ford Optimizado [3]. Como un problema de programación dinámica que considera varios factores y movimientos genera más restricciones, en cuanto al tiempo de ejecución del modelo de programación lineal resulta más conveniente, con los supuestos descritos, se evitan gran porcentaje de iteraciones.

VII. CONCLUSIONES

Como ya se ha mencionado antes Robot Path Planning Problem es NP-Completo, por lo que no existe manera óptima de resolverlo, aún así, se ha logrado identificar un método que consigue presentar un buen algoritmo para la resolución del problema, el método del campo potencial, permite identificar mediante cargas y polaridades el camino a seguir por el robot, si bien presenta grandes ventajas, como generar trayectorias continuas y en tiempo real, facilitando el control y la planificación de caminos, se presenta una gran desventaja, la aparición de mínimos locales, que podría impedir que el robot alcance su objetivo y existen puntos en los que se anulen las fuerzas.

Utilizar un buen método para la planificación de caminos del robot puede hacer que un robot se desplace de una mejor manera al reducir su camino recorrido, lo cual permite simular que el robot analiza los caminos y selecciona el mejor para desplazarse. Los constantes avances de investigación en inteligencia artificial podrían ayudar a resolver este tipo de problemas de manera más dinámica.

En base a esto, queda como trabajo futuro, mejorar el algoritmo, o identificar e implementar un algoritmo mediante el cual se asegure que de haber un camino el robot siempre sea capaz de llegar desde el punto inicial al punto final.

REFERENCIAS

- [1] Flor Ángela B. and Alejandro F. La robótica como un recurso para facilitar el aprendizaje y desarrollo de competencias generales. *Teoría de la Educación. Educación y Cultura en la Sociedad de la Información*, 13:120–136, 2012.
- [2] R. BELLMAN. *Dynamic Programming*. Princeton University Press, 1957.
- [3] Maykel Orlando Torres Piñeiro and Moreno Vega V. Planificación de trayectorias mediante programación dinámica. *Revista Ingeniería Electrónica, Automática y Comunicaciones*, 31, 01 2011.
- [4] Diego Alexander Tibaduiza. *Planteamiento de trayectorias de un robot movi*. Tesis, 2006.