

# Enfoque de ventana dinámica para evitar colisiones

Luis Servín

Taller TRA, Universidad Nacional Autónoma de México

April 22, 2016

Este enfoque fue diseñado para robot móviles equipados con un equipo con sistema de locomoción síncrono, ya que la implementación se deriva directamente de la dinámica del robot.

Uno de los objetivos finales de la investigación en sistemas robóticos para espacios internos es construir robots que sean capaces de llevar a cabo una misión en ambientes habitados y/o hostiles. Para poder construir robots móviles autónomos se debe buscar que estos perciban su ambiente, sean capaces de reacciones a situaciones imprevistas, además de planear/replantear sus acciones de manera dinámica con el objetivo de llevar a termino sus misiones.

Este artículo tiene como punto central el buscar un comportamiento reactivo para buscar evitar colisiones con obstáculos imprevistos. *dynamic windows approach* es un enfoque especialmente diseñado para lidiar con las restricciones debidas a limitaciones en velocidad y aceleración. Nuestro enfoque considera de manera periódica solo un pequeño intervalo de tiempo al momento de calcular el próximo comando de movimiento. Por lo tanto el robot solo considerará aquellas velocidades que pueden ser alcanzadas dentro de dicho intervalo de tiempo.

Se escoge una combinación de velocidades de traslación y rotación de tal manera que se maximice una función objetivo. Esta función objetivo incluye una medida del progreso actual hacia la ubicación objetivo, la velocidades actual del robot, además de la distancia al siguiente obstáculo encontrado en la trayectoria.

Se ha implementado y probado este enfoque en la plataforma *B21* fabricada por *Real World Interface Inc.* El método ha sido igualmente exitoso si las entradas al sistema provienen de cámaras y sensores infrarrojos.

## 1 Trabajo relacionado

Las estrategias para evitar colisiones pueden ser clasificadas en dos categorías

- **Global:** Tales como *road-map*, *cell-descomposition*, *potential fields* suelen asumir que un modelo completo del ambiente se encuentra disponible. Su principal ventaja radica en que se puede calcular una trayectoria completa desde el punto inicial hacia el objetivo de manera previa (*offline*). Pero no son apropiadas para generar una respuesta rápida al momento de evadir un obstáculo imprevisto.

- **Local o enfoque reactivo:** Hace uso solamente de una fracción del ambiente para con ello generar el control del robot. Tiene como desventaja el hecho de no producir una solución óptima, además de que pueden estancarse en mínimos locales. Su principal ventaja radica en que tiene una baja complejidad de cálculos, lo cual es importante si el modelo del ambiente se calcula de manera frecuente. Borestein y Koren identificaron que dichos métodos pueden fallar si se trata de trayectorias que se encuentran entre espacios muy cercanos, además de causar un comportamiento oscilatorio en corredores angostos.

## 2 Ecuaciones de movimiento para un robot de locomoción síncrona

Se asume que las velocidad traslacional y rotacional del robot pueden ser controladas de manera independiente. Las trayectorias del robot consisten en un secuencia finita de muchos segmentos de círculos. Esta representación en pedazos de círculos forma la base principal del enfoque *dynamic window approach to collision avoidance*.

### 2.1 Ecuaciones de Movimiento Generales

Sean  $x(t), y(t)$  las coordenadas del robot en el tiempo  $t$ , con respecto a un sistema de coordenadas global. Sea  $\theta(t)$  la orientación del robot. Por lo tanto la tripleta  $\langle x, y, \theta \rangle$  describe la configuración cinemática del robot. Además se tiene la condición que la velocidad de traslación  $v$  sigue el sentido de giro de  $\theta$ . Sean  $x(t_0), x(t_1)$  las coordenadas del robot en el eje  $x$  en el momento  $t_0, t_1$  respectivamente. Definimos  $v(t)$  como la velocidad de traslación en el tiempo  $t$ , y  $\omega(t)$  como la velocidad de rotación. Entonces:

$$x(t_n) = x(t_0) + \int_{t_0}^{t_n} v(t) \cdot \cos\theta(t) dt \quad (1)$$

$$y(t_n) = y(t_0) + \int_{t_0}^{t_n} v(t) \cdot \sin\theta(t) dt \quad (2)$$

$v(t)$  depende de la velocidad de traslación inicial  $v(t_0)$ , además de la aceleración de traslación  $\dot{v}(t)$  en el intervalo  $t \in [t_0, t]$ .  $\theta(t)$  es una función de la orientación inicial  $\theta(t_0)$  y la aceleración rotacional  $\dot{\omega}(t)$  en el intervalo  $t \in [t_0, t]$ . Dadas estas relaciones y sustituyendo en  $x(t_n)$ :

$$x(t_n) = x(t_0) + \int_{t_0}^{t_n} (v(t_0) + \int_{t_0}^t \dot{v}(\hat{t}) d\hat{t} \cdot \cos(\theta(t_0) + \int_{t_0}^t (\omega(t_0) + \int_{t_0}^{\hat{t}} \dot{\omega}(\check{t}) d\check{t}) d\hat{t}) dt \quad (3)$$

Se observa por lo tanto que la ecuación que describe la trayectoria del robot depende únicamente de la configuración dinámica inicial en el momento  $t_0$  y las aceleraciones, las cuales asumimos controlables (para la mayoría de las plataformas móviles una limitación en la corriente aplicada corresponde directamente con una limitación en la aceleración).

Eq. (3) puede ser simplificada si se asume que entre dos puntos arbitrarios  $t_0, t_n$  el robot puede ser controlado únicamente por un número finito de comando de aceleración. Definiendo a  $n$  como el número de pasos. Las aceleraciones  $\dot{v}$  y  $\dot{\omega}$  para  $i = 1 \dots n$  se mantiene constantes en los intervalos  $[t_i, t_{i+1}]$  ( $i = 1 \dots n$ ). Además definiendo  $\Delta t$  como  $t - t_1$ . Por lo tanto:

$$x(t_n) = x(t_0) + \sum_{i=0}^{n-1} \int_{t_i}^{t_{i+1}} (v(t_i) + \dot{v}_i \cdot \Delta_t^i) \cdot \cos(\theta(t_i) + \omega(t_i) \cdot \Delta_t^i + 1/2 \dot{\omega}_i \cdot (\Delta_t^i)^2) dt \quad (4)$$

La Eq. 4 describe el caso general de control de un robot móvil

Ahora para simplificar la Eq. 4 aproximaremos las velocidades del robot dentro de un intervalo  $[t_i, t_{i+1}]$  por un valor constante. Lo cual nos permitirá aproximar la trayectoria de un robot a por una serie de arcos circulares.

Si suponemos que el intervalo  $[t_i, t_{i+1}]$  es suficientemente pequeño, entonces el término  $v(t_i) + \dot{v}_i \cdot \Delta_t^i$  puede ser aproximado por una velocidad de traslación  $v_i \in [v_{t_i}, v_{t_{i+1}}]$  gracias al movimiento continuo y suave del robot. De la misma manera con el término  $\theta(t_i) + \omega(t_i) \cdot \Delta_t^i + 1/2 \dot{\omega}_i \cdot (\Delta_t^i)^2$  puede ser aproximado por  $\theta(t_i) + \omega_i \cdot \Delta_t^i$  donde  $\omega_i \in [\omega_{t_i}, \omega_{t_{i+1}}]$ . Esto nos lleva a lo siguiente:

$$x(t_n) = x(t_0) + \sum_{i=0}^{n-1} \int_{t_i}^{t_{i+1}} \cdot \cos(\theta(t_i) + \omega_i \cdot (\hat{t} - t_i)) d\hat{t} \quad (5)$$

Resolviendo la integral la ecuación puede ser simplificada a

$$x(t_n) = x(t_0) + \sum_{i=0}^{n-1} (F_x^i(t_{i+1})) \quad (6)$$

donde

$$F_x^i(t) = \begin{cases} \frac{v_i}{\omega_i} (\sin \theta(t_i) - \sin(\theta(t_i) + \omega_i \cdot (t - t_i))), \omega \neq 0 \\ v_i \cos(\theta(t_i)) \cdot t, \omega = 0 \end{cases} \quad (7)$$

Se sigue un proceso análogo para el caso de  $y$ .

Por lo tanto se observa que si  $\omega_i = 0$  el robot seguirá una línea recta, mientras que si  $\omega_i \neq 0$  el robot describirá un círculo. Definiendo

$$M_x^i = - \frac{v_i}{\omega_i} \cdot \sin \theta(t_i) \quad (8)$$

$$M_y^i = \frac{v_i}{\omega_i} \cdot \cos \theta(t_i) \quad (9)$$

se puede por tanto encontrar que

$$(F_x^i - M_x^i)^2 + (F_y^i - M_y^i)^2 = \left( \frac{v_i}{\omega_i} \right)^2 \quad (10)$$

en donde se observa que la  $i$ -th trayectoria se trata de un círculo  $M_i$  con centro en  $(M_x^i, M_y^i)$  y radio  $M_r^i = \frac{v_i}{\omega_i}$ .

Además de las condiciones iniciales, las ecuaciones obtenidas hasta este momento como se puede observar dependen solamente de las velocidades. A pesar de ello no es posible asignar velocidades arbitrarias al momento de realizar el control, ya que existe restricciones dinámicas que restringen los valores que se puede tomar la velocidad en intervalos subsecuentes.

## 2.2 Límite Superior de la Aproximación del Error

El error debido a las suposición hechas durante nuestro desarrollo esta limitado de manera lineal en el tiempo comprendido entre los puntos  $t_{i+1} - t_i$ .

Si consideramos los errores  $E_x^i, E_y^i$  para las coordenadas  $x, y$  dentro del intervalo  $[t_i, t_{i+1}]$ . Entonces un límite para los errores en el momento  $i$  esta descrito por la relación  $E_x^i, E_y^i \leq |v(t_{i+1}) - v(t_i) \cdot \Delta t_i|$

## 3 Dynamic Window Approach

Este enfoque se encarga de buscar comando de control obtenidos directamente del espacio de velocidades. La dinámica del robot es tomada en cuenta con ello se reduce el espacio de búsqueda a aquellas alcanzables bajo restricciones dinámicas. Solamente se toman en cuenta aquellas velocidades que son seguras con respecto a los obstáculos detectados. Este filtro es hecho durante la primera etapa del algoritmo. Como segunda etapa se busca obtener aquella velocidad que maximice una función objetivo.

A continuación se da una breve explicación de los pasos mencionados

**1. Espacio de búsqueda:** El espacio de búsqueda es reducido a través de tres pasos:

- (a) **Trayectorias circulares:** Se considerar solamente trayectorias circulares determinadas por el par  $v, \omega$ . Lo cual lleva a espacio bi-dimensional
- (b) **Velocidades alcanzables:** Se considerar solamente un conjunto de velocidades  $v, \omega$  que sean seguras. Se considera como seguro un par de velocidades si el robot es capaz de detenerse antes de alcanzar el obstáculo mas cercano dentro de la trayectoria actual.
- (c) **Ventana dinámica:** Esta característica limita las posibles velocidades a aquellas que pueden ser alcanzadas durante un corto periodo de tiempo dadas las capacidades de aceleración del robot

**2. Optimización:** Se busca maximizar la función objetivo:

$$G(v, \omega) = \sigma(\alpha \cdot heading(v, \omega) + \beta \cdot dist(v, \omega) + \gamma \cdot vel(v, \omega)) \quad (11)$$

con respecto a la posición y orientación actual se consideran los siguientes aspectos:

- (a) **Target heading:** *heading* se define como la medida en la cual la trayectoria se dirige mas directamente hacia el objetivo.
- (b) **Clearance:** *dist* La distancia hacia el obstáculo mas cercano dentro de la trayectoria. Entre mas cercano se encuentre mas buscará el robot rodearlo.
- (c) **Velocity:** *vel* se trata de la velocidad de avance del robot y la capacidad de movimientos rápidos

$\sigma$  suaviza la suma, obteniendo como resultado mayor o menor espacio con respecto a los objetos

### 3.1 Trayectorias Circulares

Definimos *curvaturas* como la secuencia segmentos circulares que aproximan la trayectoria. Cada curvatura se define de manera única por un vector de velocidad  $(v, \omega)$  al cual nos referiremos simplemente como *velocity*. Por lo tanto para generar una trayectoria a un punto objetivo, para los  $n$  segmentos de los cuales se compone se necesitarían definir el mismo número de conjuntos  $(v, \omega)$  para cada uno de los  $n$  intervalos entre  $t_0, t_n$ .

*Dynamic Window Approach* considera de manera exclusiva el primer intervalo de tiempo y asume que las velocidades restantes son constantes (con lo cual estamos asumiendo nula aceleración para estos intervalos). Una de las principales causas de esto es ya que la búsqueda de nuevas soluciones es realizada cada intervalo de tiempo.

### 3.2 Velocidades Aceptables

Los obstáculos en el ambiente cercano al robot imponen restricciones a los valores de las velocidades de traslación y rotación. Si asumimos que para el conjunto  $(v, \omega)$ , el término  $dist(v, \omega)$  representa la distancia hacia el obstáculo mas cercano sobre la curvatura correspondiente. El conjunto  $(v, \omega)$  es considerado como aceptable si el robot es capaz de detenerse dentro de la trayectoria antes de colisionar con el obstáculo. Considerando  $\dot{v}_b, \dot{\omega}_b$  como las velocidades de ruptura, podemos considerar al conjunto  $V_a$  como admisible si

$$V_a = \{v, \omega \mid v \leq \sqrt{2 \cdot dist(v, \omega) \cdot \dot{v}_b} \wedge \omega \leq \sqrt{2 \cdot dist(v, \omega) \cdot \dot{\omega}_b}\} \quad (12)$$

Por lo tanto  $V_a$  permitirá al robot detenerse antes de chocar con un obstáculo.

### 3.3 Dynamic Window

Con objeto de tomar en cuenta la limitante en la aceleración, el espacio de búsqueda se reduce a la ventana dinámica la cual contiene solo las velocidades que pueden ser alcanzadas dentro del siguiente intervalo de tiempo. Asumiendo a  $t$  como el intervalo

de tiempo en el cual las aceleraciones  $\dot{v}, \dot{\omega}$  serán aplicadas, y  $(v_a, \omega_a)$  las velocidades actuales. Entonces la ventana dinámica  $V_d$  se define como

$$V_d = \{(v, \omega) \mid v \in [v_a - \dot{v} \cdot t, v_a + \dot{v} \cdot t] \wedge \omega \in [\omega - \dot{\omega} \cdot t, \omega + \dot{\omega} \cdot t]\} \quad (13)$$

La ventana dinámica se encuentra centrada alrededor de las velocidades actuales y su extensión depende de las aceleraciones que puedan ser ejercidas. Todas las curvaturas que se encuentren fuera no pueden ser alcanzadas.

### 3.4 Espacio de Búsqueda Resultante

Siendo  $V_s$  el espacio de todas las posibles velocidades, entonces definimos a  $V_r$  como la intersección de los espacios antes mencionados

$$V_r = V_s \cap V_a \cap V_d \quad (14)$$

### 3.5 Maximizando la función objetivo

Después de haber determinado el espacio de búsqueda  $V_r$ , se selecciona un conjunto *velocity*. A través de realizar una discretización sobre el espacio resultante  $V_r$  se busca el máximo de la función de optimización

$$G(v, \omega) = \sigma(\alpha \cdot heading(v, \omega) + \beta \cdot dist(v, \omega) + \gamma \cdot vel(v, \omega)) \quad (15)$$

#### 3.5.1 Target Heading

El término  $heading(v, \omega)$  es una medida del alineamiento del robot con respecto a la dirección objetivo. Esta dado por  $180 - \theta$ , donde  $\theta$  es el ángulo del punto objetivo con respecto a la dirección actual del robot.  $\theta$  se calcula con respecto a la posición predecida del robot, esta posición se calcula asumiendo que el robot se mueve con el conjunto de velocidades seleccionados durante el próximo intervalo de tiempo. Para una aproximación mas realista se calculará  $\theta$  en la posición en la cual el robot haya alcanzado la máxima desaceleración después de dicho intervalo.

#### 3.5.2 Clearance

El término  $dist(v, \omega)$  representa la distancia hacia el obstáculo mas cercano que intersecta con la curvatura. Si no existe un obstáculo dentro de la curvatura el valor crece a una constante alta.

#### 3.5.3 Velocity

Mientras que el término  $velocity(v, \omega)$  evalua el progreso del robot sobre la trayectoria. Se trata simplemente de una proyección sobre la velocidad de traslación.

### 3.5.4 Smoothing

Por último se normaliza la suma de los términos hacia un intervalo  $[0, 1]$ . El proceso aumenta el espacio libre lateral del robot.

Se debe hacer notar que todos los términos que componen a  $G$  son necesarios. Esto porque al combinar los tres elementos, el robot esquivo las posibles colisiones tan rápido como puede tomando en cuenta las restricciones impuestas, además de seguir progresando en su camino hacia la meta.

El intervalo de tiempo en el cual se calcula la ventana dinámica está fijado a 0.25 seg. Además de que debe hacerse notar que el comportamiento cambia con respecto a las velocidades actuales.

### 3.5.5 Dependencia de la aceleración

El espacio de búsqueda restringido, la evaluación de la función y la ventana dinámica también dependen de la aceleración disponible. Ya que si se tiene un menor rango de aceleraciones esto provocará que el espacio de velocidades disponibles sea más pequeño también. Por lo tanto el tamaño de la ventana disminuirá al depender de las aceleraciones.

## 4 Implementación y Resultados Experimentales

### 4.1 RHINO

El enfoque descrito de manera anterior ha sido implementado y probado usando el robot RHINO, una plataforma con sistema de locomoción síncrono. Equipado con un anillo de 24 sensores ultrasónicos Polariod, 56 detectores infrarrojos, y un sistema de cámara estéreo.

### 4.2 Campo de Obstáculos Línea

Como modelo local del mundo se hace uso de un campo de obstáculos en línea. Este se define como una descripción en dos dimensiones de la información obtenida a través de los sensores del robot con respecto a la posición del robot. El sistema se calibró de manera que las mediciones más erróneas fueran las que representaran la distancia más grande. Cada lectura es convertida a una línea de obstáculos.

Esta línea es perpendicular al eje principal del haz del sensor. Mientras que la distancia de dicha línea es determinada por la amplitud calculada del haz en la distancia obtenida. A través de estos datos se calcula la distancia hacia estos obstáculos en las posibles curvaturas calculadas. Si definimos a  $r$  como el radio de la trayectoria circular y  $\gamma$  el ángulo entre la intersección de la línea de obstáculo con la curvatura y la posición del robot a partir del centro de rotación. Por lo tanto la distancia al siguiente obstáculo está dada por  $r \cdot \gamma$

Con el objetivo de permitir al robot reaccionar de manera más rápida a los cambios que se pueden presentar. Limitamos el número de líneas a 72 y aplicamos la estrategia

*first-in-first-out* para remover las líneas menos actuales del campo de obstáculos de líneas.

### 4.3 Mayores detalles acerca de la implementación

Las siguientes estrategias ayudaron a mejorar la *maneobrabilidad* y la elegancia en el movimiento del robot.

- **Modo *Rotate Away*.** En caso de que el robot pudiera encontrarse estancado en un mínimo local se aplica esta estrategia. Consiste en rotar lejos del obstáculo hasta que sea capaz de trasladarse de nuevo.
- **Speed dependent side clearance.** Se trata de adaptar la velocidad del robot con respecto a la distancia que tiene con respecto a las paredes. Consiste en que el robot viajará a velocidades altas si es que se encuentra en corredores despejados y amplios y a velocidades bajas si tiene trasladar a través de puertas, espacios reducidos o congestionados.

#### Ajustes de los parámetros

Sin recurrir a un ajuste de parámetros muy exhaustivo se encontró que los valores de 0.8, 0.1 y 0.1 para  $\alpha, \beta, \gamma$  respectivamente han dado buenos resultados.

#### Role de la Ventana Dinámica

Se puede identificar como punto crucial de la trayectoria la zona en la cual se tiene que hacer un cambio de dirección súbito con tal de poder llegar al punto objetivo, esta zona es conocida como **área de decisión**. Se concluyó que solamente si la velocidad actual del robot y las posibles aceleraciones que este podría sufrir permiten un giro brusco en la dirección requerida, el robot se moverá directamente hacia el objetivo.

#### Movimiento en Corredores Rectos

Se observó que después de evadir un obstáculo la plataforma intentó seguir líneas rectas tanto como fuera posible, evitando asimismo movimientos oscilatorios.