

Comparativa de algoritmos en el contexto de la evitación de obstáculos en navegación 2D

Robótica Móvil

Martín Romero y Pablo Riveira

Resumen

En este trabajo se ha hecho una comparación de distintos métodos de evitación de obstáculos. Concretamente los métodos PFM (Potential Field Method), VFH (Vector Field Histogram) y CVM (Curvature Velocity Method). En esta comparativa se ha analizado datos extraídos durante las simulaciones, que nos permitieron ver las ventajas e inconvenientes de cada método. Tras estos análisis vimos una gran desventaja del método VFH con respecto a los otros dos, debido a la ventana de percepción que implementa, deduciendo que para estas implementaciones concretas, los métodos PFM y CVM obtienen mejores resultados que VFH.-

Introducción

- El proceso de navegación para la evitación de obstáculos se asocia con el proceso reactivo que ocurre mientras el robot está en movimiento. Estamos ante un proceso de planificación local, donde nuestro algoritmo de evitación de obstáculos hará lo que su nombre indica. Dicho esto, existen muchos enfoques para solventar el problema de evitación: desde simular métodos de potencial basados en la física, hasta basados en comportamientos bioinspirados. Éste es un problema real e importante ya que es crucial para la navegación de robots en entornos estocásticos donde los obstáculos puedan estar en cualquier sitio y donde normalmente la localización por medio de métodos externos para evitar colisiones no puede realizarse con GPS , debido a la mala señal.

- Hasta día de hoy existen muchos abordajes para el problema de la evitación de obstáculos en el contexto de la navegación en robótica. En general hay realmente muchas soluciones , más o menos óptimas y que además se suelen adaptar a ciertos entornos y parámetros específicos. Teniendo esto en cuenta, nuestros algoritmos no pretenden desbancar los que ya existen, pero sí intentaremos dar una visión de su funcionamiento y de su comparación a la hora de resolver el problema de evitación de obstáculos.

- Así pues, vamos a realizar 3 implementaciones de 3 algoritmos: VFH,CVM,PFM de evitación de obstáculos para posteriormente, compararlos en base a una métrica que definiremos más adelante. Emplearemos la interfaz gráfica dispuesta para la práctica 4 de la asignatura, aunque le hemos realizado varios cambios a los archivos base, para adecuarlos y acomodarlos a nuestras pruebas. Una vez realizadas las implementaciones, llevaremos a cabo grupos de unas 200 simulaciones de las cuales extraeremos la información más relevante.

Métodos utilizados

PFM

Primeramente, mencionamos aquí que en nuestra tarea de realizar una comparación entre algoritmos de evitación de obstáculos, hemos decidido añadir también la implementación ,realizada en las prácticas de la asignatura, correspondiente a PFM como “baseline”, i.e., lo usaremos dentro de la comparación suponiendo que , en virtud de los resultados obtenidos de dicha práctica, tiene un funcionamiento bastante eficiente y óptimo.

- Los métodos de potencial PFM, se basan en la analogía de los campos electrostáticos, en los que partículas con carga de distinto signo se atraen y partículas con carga de igual signo se repelen. En el caso de los métodos de potencial en robótica se considera que el robot y los obstáculos tienen carga del mismo signo (i.e. se repelen mutuamente), mientras que el objetivo tiene carga de signo contrario al robot (i.e. se atraen).

VFH

- En cuanto a la implementación del algortimo de VFH o Vector Field Histogram, se utilizan las mediciones de distancia del robot a los obstáculos en su entorno y las convierte en un histograma de campo vectorial (VFH). Luego, utiliza este histograma para calcular la dirección en la que el robot debe moverse para evitar los obstáculos y navegar hacia su destino.

- Se obtendrá el valor de cada “VFH” mediante el cálculo de la magnitud y el ángulo de cada vector de distancia medida en cada ángulo. Posteriormente, se comprueba si hay algún obstáculo cerca, y si es así, se calcula la dirección más apropiada en la que el robot debe girar para evitar el obstáculo. Esto se hace recorriendo el VFH y buscando el primer ángulo con un valor VFH positivo. Este ángulo se considera como la dirección de evasión. Si no hay obstáculos cerca, el robot se moverá en la dirección deseada hacia su destino.

- Destacar que para realizar la detección de obstaculos en el sentido de avance, se establece también una ventana angular de visión, la cual se empleará a la hora de iterar sobre los ángulos candidatos para la navegación.

CVM

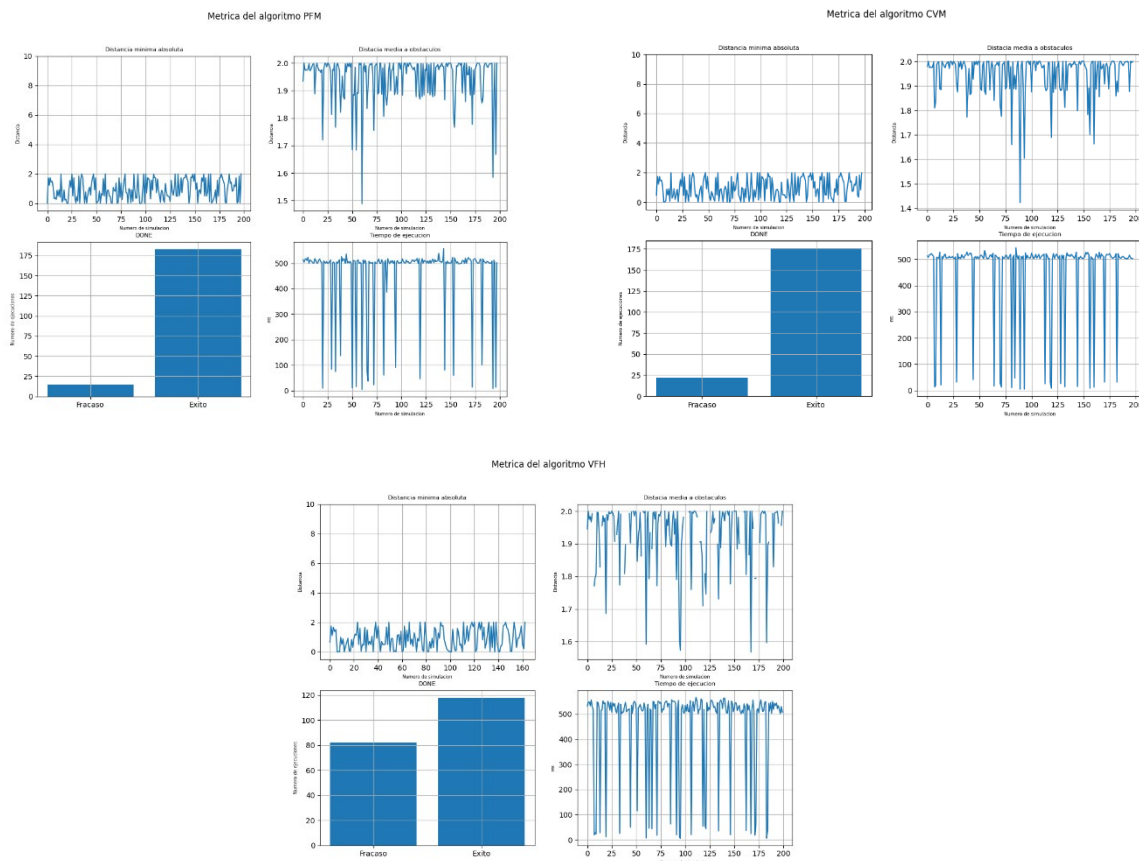
- Con respecto a CVM, es un algoritmo que asume que para cada ciclo, el robot viaja a lo largo de arcos de curvatura constante correspondientes a un punto en el espacio de velocidad. Para incluir restricciones de obstáculos, el robot se considera un punto y los obstáculos se aproximan mediante círculos con el radio del robot agregado a ellos. El algoritmo intentará ejecutar la trayectoria de evitación a la velocidad más alta posible (teniendo también en cuenta las limitaciones físicas del robot). Este método solo controla la evitación de obstáculos, por lo que para que el robot se mueva hacia el objetivo cuando no tiene nada que esquivar, el control del robot se hace mediante una fuerza de atracción (como si fuese un método de potencial). En caso de detectar varios obstáculos, el algoritmo tomará el obstáculo más cercano para calcular las velocidades de evitación.

Resultados

- Una vez implementados los distintos algoritmos de evitación de obstáculos, hemos recogido información relevante para poder analizarlos. Para la obtención de los datos hemos hecho 200 simulaciones con cada algoritmo, con obstáculos en posiciones aleatorias y posiciones iniciales aleatorias. Hay que tener en cuenta que las condiciones en cada experimento deben ser las mismas para cada algoritmo, para que así nuestro análisis pueda tener sentido. Para solucionar esto se han utilizado semillas, que nos permiten obtener los mismos obstáculos aleatorios y posición inicial aleatoria para las tres ejecuciones de los algoritmos.

- Los datos recogidos en cada simulación han sido: distancia mínima a obstáculos, distancia media a obstáculos, tiempo que tarda en llegar al objetivo y confirmación de si ha sido capaz de llegar al objetivo o no.

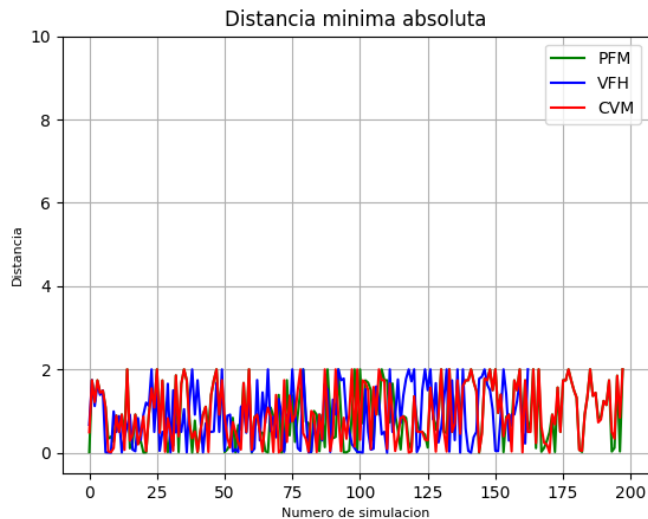
- Para ello, mostramos los resultados para cada algoritmo en primer lugar:



- En cuanto a la tasa de éxito o fracaso vemos claramente que el algoritmo VFH tiene una tasa enorme de fracaso, y tanto CVM como PFM tienen una tasa de fracaso bastante baja y parecida. Tras ver esto pensamos que la alta tasa de fracaso de VFH se debe a que la implementación incluye una ventana de percepción en el sentido de avance de forma que para ciertas combinaciones de entornos y obstáculos, dicho parámetro deberá adecuarse a mano o implementando tal vez algún método de aprendizaje automático como RL, por ejemplo.

- Vamos ahora a ver en comparación las métricas anteriores, para percibir más visualmente como se comportan los algoritmos.

- Con respecto a la métrica de distancia mínima a los obstáculos:



- Podemos ver cómo en general los 3 algoritmos están muy parejos. Recordando que cada unidad en el eje horizontal equivale a una ejecución donde se han tenido las mismas condiciones iniciales, se observa que en ciertas simulaciones se han obtenido casi las mismas mediciones en cuanto a este parámetro. Esto tiene sentido por 2 motivos: por una parte es muy probable que cuando se genera un mapa y obstáculos lejos del objetivo y del origen, las mediciones sean las mismas e iguales a 2 (de ahí los frecuente y comunes picos) y por otra parte resulta que el camino, en general, hacia el objetivo resulta en trayectorias muy similares, ergo, detección de distancias similares en los 3 algoritmos.

- Realizamos también la media y std de todas las ejecuciones, para intentar extraer algo más de significado:

-----MEDIA y DESVIACION TIPICA de DISTANCIAS MINIMAS-----

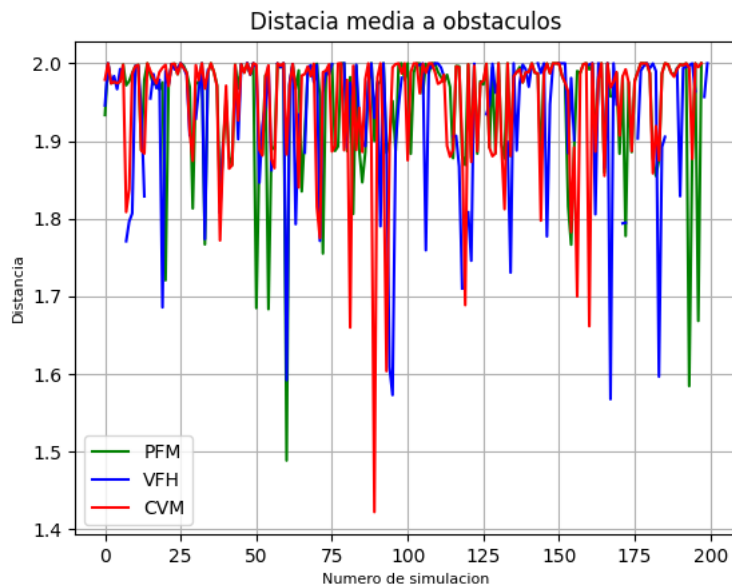
-->PFM: mean: 0.9453466993434343 std: 0.676767689535239

-->CVM mean: 0.946933589868686 std: 0.6723656094027708

-->VFH mean: 0.8826376009570552 std: 0.6809615511683753

- Viendo estos datos, también apreciamos un gran parecido, si bien el algoritmo VFH hace una detección mínima menor con algo más de desviación típica.

- Con respecto a la métrica de distancia media medida los obstáculos:



-----MEDIA y DESVIACION TIPICA de DISTANCIAS MEDIAS-----

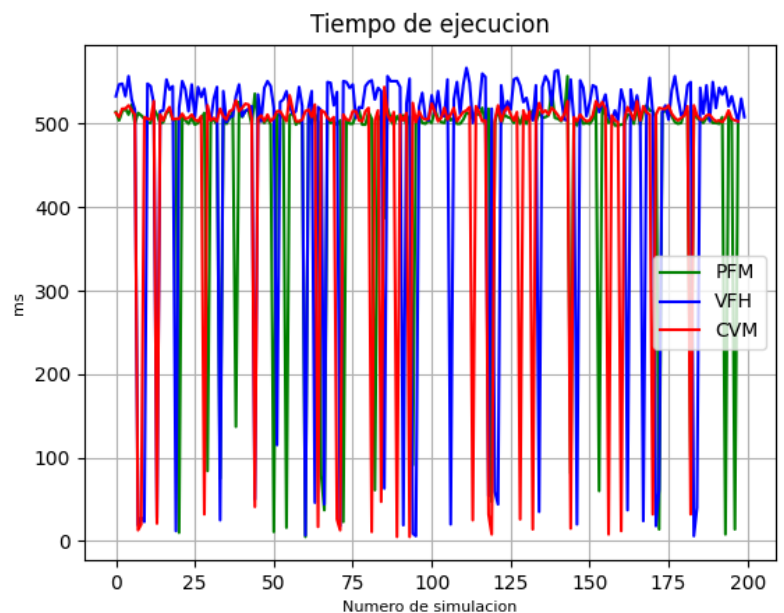
-->PFM: mean: 1.9474919696969695 std: 0.07964483827149134

-->CVM mean: 1.94833398989899 std: 0.0814268711618715

-->VFH mean: 1.9320577300613497 std: 0.09571930996921335

En cuanto a los datos de distancia media a obstáculos, vemos que generalmente el robot se mantiene alejado de obstáculos, nos confirma que por lo general los algoritmos detectan los obstáculos y se alejan de ellos, aunque también el hecho de que sean valores muy cercanos al 2 (rango de detección máximo del robot) puede deberse a que al generar los obstáculos de forma aleatoria, muchas veces estos obstáculos no llegan a aparecer en el rango de visión del robot, o a estar lo suficientemente cerca como para influir en el comportamiento del mismo. Por otro lado, vemos que hay picos en los que las distancias son más pequeñas, cosa que podría deberse a veces en que el robot se inicia muy cerca de un obstáculo choca, no siendo capaz de esquivarlo.

- Por último con respecto al tiempo requerido para alcanzar el objetivo:



- En general podemos apreciar una tendencia en alcanzar al objetivo en torno a un intervalo entre los 500 ms y los 550 ms, existiendo bastantes “outliers” que , teniendo en cuenta la correspondencia del num de simulación y lo visto en las otras 2 métricas, asociamos a 2 situaciones: por una parte a mapas y orígenes que debido a la semilla aleatoria resultan en orígenes próximos al objetivo y alejados de obstáculos, y por otra parte a posibles simulaciones que fracasen, ya sea por imposibilidad de llegar al objetivo o por “*crashear*” (este en mucha menor proporción)

- Podemos realizar las medias y std de cada algortimo en total, de forma que podemos fijarnos en que si tenemos en cuenta los “outliers”, las medias son menores pero las desviaciones típicas , bastante grandes:

-----MEDIA y DESVIACION TIPICA del TIEMPO-----

-->PFM: mean: 463.06060606060606 std: 135.50264212876115

-->CVM mean: 457.04545454545456 std: 154.48527423054378

-->VFH mean: 461.24 std: 173.63577511561377

- Si por otro lado, descartamos esos valores, obtenemos una correspondencia mejor con la tendencia observada en las medias y unpos valores más bajos en las desviaciones:

-----MEDIA y DESVIACION TIPICA del TIEMPO-----

-->PFM: mean: 506.8826815642458 std: 12.062809625018405

-->CVM mean: 511.59090909090907 std: 7.460709757298204

-->VFH mean: 530.9127906976744 std: 17.227899874113255

Conclusiones

- Llegados a este punto está claro que las implementaciones de los distintos algortimos de evitación de obtáculos presentados, son muy similares en comportamiento. Tal vez, podríamos afirmar que dependiendo de la situación y del entorno, se pueda elegir entre ellos. En general VFH es el que resultados más pobres ha dado, tal vez debido a la ventana angular de percepción del entorno y donde la implementación podría ser mejorada. Para trabajos futuros/mejoras podría ser un objetivo a lograr, codificar los algoritmo de forma más eficiente llegando a implementar su versión óptima , i.e. VFH*,CVM*, PFM* , ardua tarea. Aún con todo, hemos podido aprender/ver la gran cantidad de formas que hay para resolver el problema de evitación de obstáculos para distintas situaciones y entornos.