ISSN 1870-4069

Planificación reactiva de movimientos en tiempo real para robots móviles

Enrique Diaz R., Abraham Sánchez L., Mario Serna H., Rogelio Gonzalez V., Beatriz Bernabe L.

Benemérita Universidad Autónoma de Puebla
Computer Science Department
México

{enriqued_93, mario_sh95}@hotmail.com, abraham.sanchez@correo.buap.mx,
{rogelio.gzzvzz, beatriz.bernabe}@gmail.com

Resumen. La replanificación eficiente de movimientos es un problema importante en el campo de la navegación robótica ya que los entornos son raramente estáticos en aplicaciones del mundo real. Un robot móvil necesita continuamente monitorear y recalcular los planes de movimiento. Ya que los robots móviles a menudo estn limitados por recursos computacionales y tiempo, es importante hacer que el proceso de replanificación sea lo más eficiente posible. Este trabajo tiene como objetivo proporcionar planificadores prácticos que consideren acciones reflejas y planificación con técnicas probabilísticas para considerar los cambios de obstóculos. Presentamos resultados experimentales utilizando un simulador desarrollado por los autores, e igualmente experimentos con un robot real.

Palabras clave: métodos probabilistas, zona virtual deformable, tiempo real, robot móvil.

Reactive Motion Planning in Real Time for Mobile Robots

Abstract. Efficient motion replanning is an important problem in the field of robotic navigation since environments are rarely static in real world applications. A mobile robot needs to continuously monitor and recalculate plans of motion. Since mobile robots are often constrained by limited computational resources and time, it is important to make the process of replanning as efficient as possible. This work aims at providing practical planners that consider reflex actions and planning with probabilistic techniques to account for obstacle changes. We present experimental results using a simulator developed by the authors, and also experiments with a real robot.

Keywords: probabilistic methods, virtual deformable zone, real time, mobile robot.

1. Introducción

Actualmente, vivimos una era donde la tecnología y el humano están cada vez más unidos. Podemos notarlo con los robots móviles, cuya participación en nuestra vida cotidiana y laboral sigue aumentando. Existen robots móviles que realizan tareas de limpieza doméstica, asistencia médica, entrega de paquetería e incluso rescate; tareas que deben ser realizadas eficiente y autónomamente. A lo largo de los últimos años, la planificación de movimiento se ha discutido y estudiado enormemente, sobre todo dentro de ambientes estáticos y supervisados. Sin embargo, es imposible hablar de robots móviles autónomos si no consideramos los ambientes dinámicos o no estructurados, en los cuales el robot debe ser capaz de reaccionar a los eventos imprevistos. Hasta ahora, se han realizado aportes importantes para resolver la navegación en ambientes dinámicos pero muchos no llegan a probarse en entornos reales, únicamente se realizan pruebas en simulación donde no existen factores que afecten la precisión en el movimiento del robot como la imperfección e inclinación del piso.

La replanificación eficiente de movimientos es un problema importante en el campo de la navegación robótica ya que los entornos son raramente estáticos en aplicaciones del mundo real. Un robot móvil necesita continuamente monitorear y recalcular los planes de movimiento. Ya que los robots móviles a menudo están limitados por recursos computacionales y tiempo, es importante hacer que el proceso de replanificación sea lo más eficiente posible. Este trabajo tiene como objetivo proporcionar planificadores prácticos que consideren acciones reflejas y planificación con técnicas probabilísticas para considerar los cambios de obstáculos.

La planificación de movimientos se refiere a la capacidad de un sistema para planificar automáticamente sus movimientos, y se considera fundamental para el desarrollo de robots autónomos.

En la última década, se realizaron muchos esfuerzos de investigación sobre la aplicación de métodos probabilísticos para diferentes tipos de problemas (Probabilistic RoadMaps (PRM) y Rapidly-exploring Random Trees (RRT)) [5,6,8].

Hay dos clases principales de planificadores probabilísticos: consulta múltiple y consulta única. Un planificador de múltiples consultas calcula previamente un roadmap (mapa de caminos) y luego lo usa para procesar muchas consultas. Por otro lado, un planificador de consulta única calcula un nuevo roadmap para cada consulta.

En el estado del arte, sin embargo, están poco adaptados a los problemas dinámicos (el costo de reflejar los cambios dinámicos en el roadmap durante las consultas es muy alto). El uso de estos roadmaps no se puede hacer sin la adición de un mecanismo de actualización que tenga en cuenta el contexto actual [10]. Aplicar algoritmos probabilísticos a entornos que requieren una replanificación eficiente no es una idea nueva [4,9]. Bruce y Veloso diseñaron los árboles aleatorios extendidos de exploración rpida (RRT) para aumentar el rendimiento mediante el almacenamiento en memoria caché de planes previos y el sesgo adaptativo de la búsqueda hacia puntos de caminos más antiguos. El objetivo es tener en cuenta los puntos aleatorios a medida que cambia el entorno [1].

Este trabajo fue ampliado y mejorado en los algoritmos MP-RRT [3] y DRRT [12]. En presencia de obstáculos, tomar muestras distribuidas uniformemente en el espacio de configuración no siempre es la mejor idea. A menudo, estos puntos de muestreo aleatorio ocurrirán dentro de un objeto. Para compensar esto, en [2], los autores proponen usar un sesgo adaptativo para generar muestras para el RRT.

Este trabajo tiene como objetivo proporcionar planificadores prácticos que consideren acciones reflejas y planificación con técnicas probabilísticas para dar cuenta de los cambios de obstáculos. Un camino factible libre de colisiones para un robot móvil o cualquier sistema mecánico se calcula utilizando un planificador basado en PRM/RRT sin considerar obstáculos dinámicos La parte dinámica es manejada por un enfoque de zona virtual deformable [13].

2. La zona virtual deformable

La capacidad de un robot para reaccionar a eventos no esperados dentro de ambientes dinámicos es primordial para concluir exitosamente las tareas asignadas. No existen muchos métodos desarrollados e implementados en mecanismos móviles con capacidad reactiva. Uno de los más clásicos es el método de control reactivo mediante la Zona Virtual Deformable(ZVD).

Este método de reacción de comportamiento reflejo propuesto en [13], utiliza el concepto de la ZVD, en el que un robot con cierta cinemática depende de una zona de riesgo que se encuentra alrededor del robot. Esta ZVD es parametrizada por las variables de movimiento del robot y puede deformarse en presencia de información de distancia en el espacio de trabajo del robot. Cuando un obstáculo ingresa al espacio del sensor, induce una deformación de la ZVD que será compensada por el controlador de movimiento del robot. El algoritmo es una especie de juego para dos jugadores: el primero, es decir, el entorno, induce deformaciones indeseadas; el segundo, es decir, el controlador del robot, intenta reconstruir la ZVD.

La Figura 1 ilustra este principio general. La ZVD inicial, Ξ_h está deformada por un vector de deformación Δ que puede escribirse como una función de dos vectores de control. La ZVD deformada se denota Ξ. Esta deformación derivada de la ZVD se puede escribir como:

$$\dot{\Delta} = A\phi + B\psi, \tag{1}$$

Las variaciones en Δ están controladas por un vector de entrada doble $u = [\phi\, \psi]^T$. El primer vector de control ϕ, debido al controlador del robot tiende a minimizar la deformación de la ZVD. Mientras que el segundo vector ψ, es desconocido e inducido por el mismo medio ambiente (y podría, al menos, tratar de maximizar estas deformaciones). Ver [13] y [10] para más detalles.

La Figura 2 se muestra algunos ejemplos de deformaciones controladas. A) velocidad baja, B) velocidad alta, c) giro a la derecha y D) giro a la izquierda. El perfil y el área de la ZVD dependen de dos tipos de parámetros: i) las magnitudes

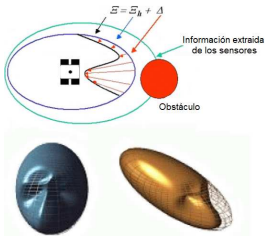

Fig. 1. Ejemplos del principio de la ZVD en 2D y 3D.

que modelan la cinemática del robot y ii) las medidas de proximidad del ambiente proporcionadas por los sensores embarcados del robot.

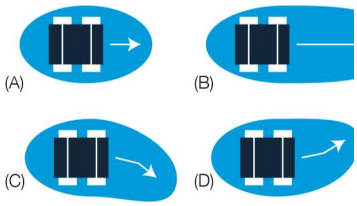

Fig. 2. Ejemplos de deformaciones controladas en un robot móvil.

Desarrollada originalmente para asegurar la evasión refleja de un robot móvil, un control por ZVD se ha aplicado en [7]:

- La protección refleja de robots manipuladores.
- La estabilización dinámica de robots con ruedas.
- El pilotaje completo de robots con ruedas.
- El equilibrio dinámico de robots con patas.

3. Estrategias reactivas

Los algoritmos probabilísticos son un esquema de planificación general que construye roadmaps probabilísticos seleccionando aleatoriamente configuraciones del espacio de configuraciín libre e interconectando ciertos pares por cami-

nos factibles simples. Los métodos se han aplicado a una amplia variedad de problemas de planificación de movimientos de robots con notable éxito [5,6].

La adaptación de los planificadores probabilísticos a ambientes con obstáculos estáticos y móviles ha sido limitada hasta ahora. Esto se debe principalmente a que el costo de reflejar los cambios dinámicos en el roadmap durante las consultas es muy alto. Por otro lado, las variantes de consulta única, que calculan una nueva estructura de datos para cada consulta, se ocupan de manera más eficiente con entornos altamente cambiantes. Sin embargo, no mantienen la información que refleja las restricciones impuestas por la parte estática del entorno útil para acelerar las consultas posteriores [4,10].

Dado un problema de planificación de movimientos, la elección del algoritmo de planificación que se utilizará se debe a diferentes factores. Si el problema a resolver involucra solo restricciones cinemáticas, entonces se pueden usar los algoritmos PRM y RRT. En el caso del planificador de una sola consulta, los algoritmos basados en árboles son en general mucho más rápidos (algunas variantes de RRT). Sin embargo, debe señalarse que la velocidad de solución se contrapone con la calidad del camino, ya que estos planificadores se detienen tan pronto como se encuentra un camino. Los planificadores basados en PRM, en cambio, pueden producir un conjunto de caminos, y luego se devuelve el más favorable. En una situación en la que deben resolverse muchas consultas sucesivas, también parece apropiado el uso del algoritmo PRM básico.

Esta sección describe diferentes estrategias reactivas, que integran un método probabilístico como PRM/RRT y el control reactivo por ZVD de la siguiente manera: se calcula un camino factible libre de colisión para un robot móvil mediante algún método probabilístico, el robot comienza a moverse (es decir ejecutar movimientos considerando su cinemática y/o sus restricciones diferenciales), bajo la protección de su ZVD, en ausencia de obstáculos dinámicos, el control se realiza con el seguimiento de caminos propuesto por Lapierre et al. [7] y no requiere comandos reflejos.

Si hay obstáculos dinámicos en su camino, el método reactivo toma el control y genera comandos que forzan al robot a alejarse de los obstáculos intrusos y devuelve su ZVD al estado original. En este punto, el robot ha perdido su camino original, y es necesario buscar un camino de reconexión para alcanzar su objetivo. El nuevo camino encontrado es un camino sencillo libre de colisiones. Si el intento de reconexión es exitoso, el robot ejecuta su nuevo camino hacia la meta. El nuevo camino alternativo se obtiene con el método probabilístico utilizando la información almacenada en la configuración actual del robot, pero si aparece una deformación, los procesos se interrumpen mediante acciones reflejas que obligan al planificador a volver al estado anterior.

El algoritmo puede terminar de tres formas: i) el robot ejecuta su camino con éxito, ii) la acciórefleja no es suficiente y se produce una colisión, o iii) el robot no encuentra un camino alternativo para concluir su tarea. La Figura 3 muestra una descripción de alto nivel del enfoque propuesto.

Después de una acción refleja exitosa, el robot móvil recupera el estado intacto de su ZVD, pero el camino inicial planificado se pierde (caso 2)), y el

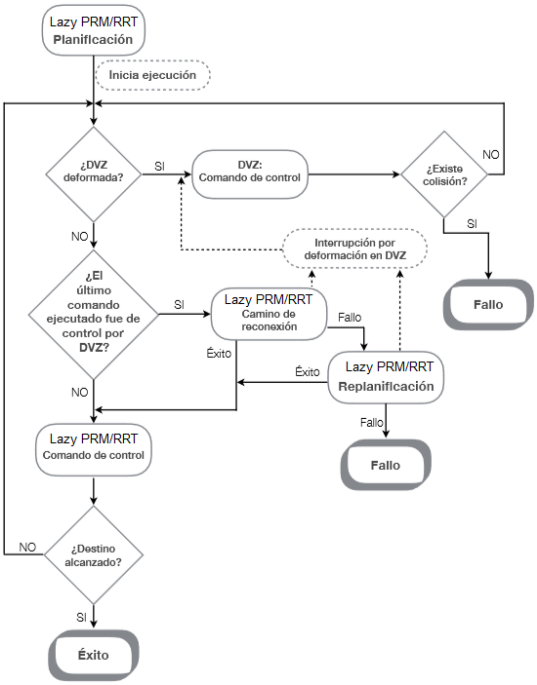

Fig. 3. Descripción de alto nivel de nuestros enfoques propuestos.

método de seguimiento de caminos necesita tener un camino para 'empujar' al robot móvil hacia la meta. Por esta razón, es necesario proporcionar un camino para tal fin.

Debido a que el costo computacional de una replanificación completa es alto, se evita en la medida de lo posible al ejecutar un proceso que consiste en una reconexión con el camino planificado utilizando un único camino libre de colisiones (caso 3)). Inicialmente, el algoritmo prueba un camino local que se interrumpió por un objeto dinámico. El algoritmo ejecutará una acción refleja para volver a conectarse con el punto más cercano que no tenga colisiones en el camino original. Si no se puede reconectar después de un cierto número de intentos, se debe a que tal vez los posibles caminos de reconexión están bloqueados por obstáculos dinámicos, el robot permanecerá inmóvil durante un cierto instante de tiempo antes de ejecutar un nuevo intento (caso 4)).

El proceso se repetirá varias veces, pero si la ZVD se deformó por una intrusión, el proceso de reconexión se modificará y ejecutará los comandos reflejos (vase la Figura 4). Si los intentos de reconexión fallan, puede suceder que los caminos estén bloqueados por muchos objetos dinámicos, o que un objeto en movimiento se estacione obstruyendo el camino planificado. En este caso, el planificador ejecuta uno de los métodos probabilísticos (la configuración inicial es la configuración actual en el robot). El planificador probabilístico se llamará

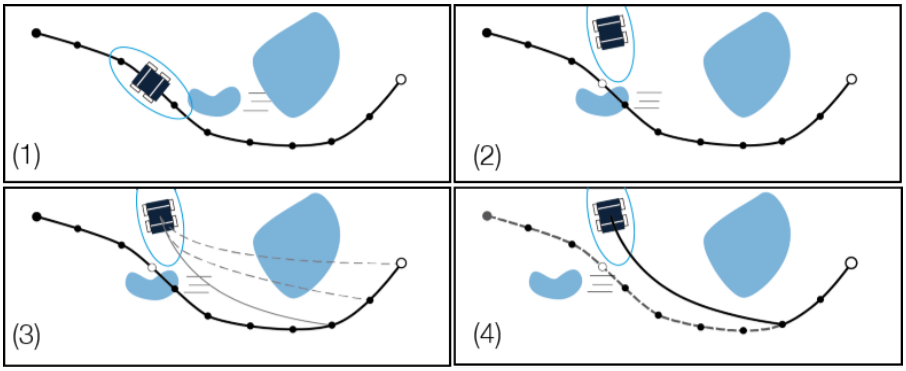

Fig. 4. Los casos del proceso de reconexión. 1) para evitar un obstáculo dinámico, 2) después de una acción refleja, 3) después de muchos intentos, 4) reconexión exitosa.

varias veces hasta que regrese un camino libre de colisiones. Si después de algunos intentos no se puede encontrar un camino libre de colisiones, el planificador informa de un error.

En el caso de que el robot móvil navegue en un entorno estático (o parcialmente estático), el camino planificado es suficiente como para evitar una colisión. Bajo esta suposición, no es necesario generar ninguna acción refleja cuando un obstáculo fijo ingresa a la ZVD.

El modelo no puede distinguir si una intrusión es causada por un obstáculo en movimiento o estático porque el método de la ZVD no usa ningún modelo del entorno. Para resolver este problema, es necesario usar una imagen auxiliar (obtenida con un sistema de visión) que represente el entorno y se actualice cada vez que se llaman los procedimientos de replanificación o reconexión. Cuando los sensores en el robot detectan un obstáculo que deforma la ZVD, las coordenadas del objeto intruso se revisan para ver si ya había un obstáculo, registrado en la imagen auxiliar; si este es el caso, el sistema asume la presencia de un obstáculo fijo y no hay necesidad de una acción refleja, de lo contrario, sin duda se supone que el objeto está en movimiento.

4. Resultados en simulación y en tiempo real

A continuación se describe el trabajo realizado para la implementación simulada de nuestra propuesta para resolver el problema de navegación de robots móviles en ambientes dinámicos. Se determinó que toda la programación se realizaría utilizando el lenguaje de programación C++ debido a su integración con todos los componentes que participarán en la implementación, es decir el Pioneer 3-DX, las librerías ARIA de Adept y el sensor láser Hokuyo URG. También se eligió por ser un lenguaje de programación muy robusto.

El procedimiento deberá tener como entrada, el mapa del entorno y las configuraciones inicial y final del robot móvil. Como entrada adicional, se podrán

modificar algunos parámetros como velocidad máxima, ángulo de dirección y constantes de la ZVD que ya poseen un valor por omisión. El procedimiento comenzará con el cálculo del camino formado por curvas Reeds & Shepp (R&S) libres de colisión que será recorrido por el robot tipo carro [14]. Después, iniciará la ejecución del movimiento teniendo en cuenta que existirán dos controles para el robot: el control por Lazy PRM/RRT (se puede utilizar el método de seguimiento de caminos propuesto en [7]) y el control por ZVD.

El primer paso fue crear un mecanismo para poder monitorear nuestros resultados, para esto se diseñó un entorno gráfico con OpenGL en donde se muestra el ambiente simulado donde se desenvolverá el robot y los controles que auxilian en la entrada de información y los parámetros para el procedimiento a realizar. Este ambiente en 3D cargará el archivo, creado con Mapper3 de Adept, correspondiente al mapa seleccionado por el usuario y desplegará los cuerpos geométricos adecuados para representar los obstóculos indicados por el mismo archivo. La Figura 5 muestra algunos escenarios simulados.

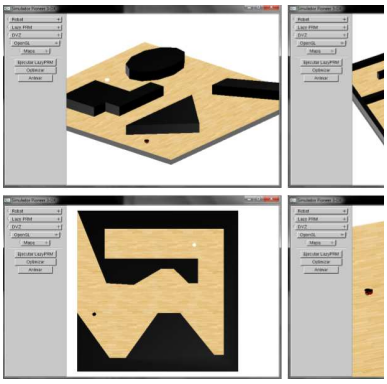

Fig. 5. Entornos de simulación para las pruebas simuladas.

Para la realización de estas pruebas se utilizó una computadora personal con las siguientes características relevantes: Procesador Intel Core i5@2.4 GHz, Memoria RAM de 6 GB DDR3 @665 MHz, Sistema Operativo Windows 7 de 64 bits.

Las pruebas de la ejecución de caminos, se realizaron con el objetivo de visualizar la cantidad de fallos y éxitos que suceden al ejecutar un camino previamente planificado (Tabla 1). La cantidad de fallos o éxitos dependerán del dinamismo del ambiente, el cual está dado por los obstáculos móviles. Cada instancia de prueba se ejecutó 20 veces.

La Figura 6 muestra algunas ejecuciones en una escenario de prueba. En 1) se muestra el camino planificado antes iniciar la ejecución, en 2) se inicia el seguimiento del camino con el método propuesto en [7]. En 3) se detecta una intrusión debida a un obstáculo móvil, pero esta no genera una deformación. En 4) el obstáculo móvil deforma a la ZVD, por lo tanto el robot cambia de

dirección y se reconecta con la configuración final. Se puede apreciar en 5) que el robot móvil se desplaza a través del camino de reconexión y finalmente en 6) se tiene el final de la ejecución.

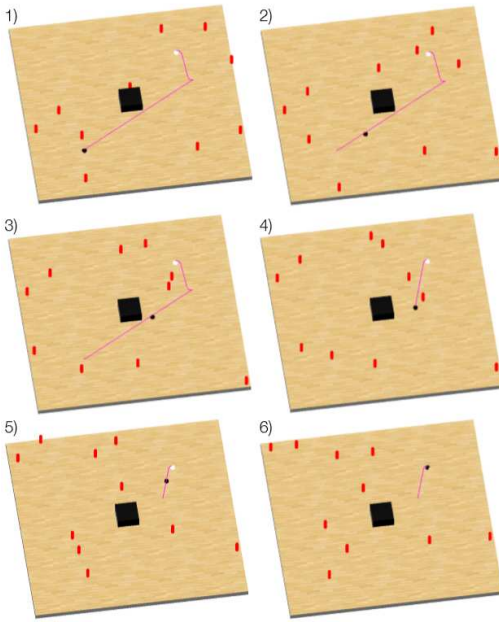

Fig. 6. Escenario de prueba que muestra las estrategias reactivas propuestas.

Tabla 1. Prueba de ejecución de caminos para el escenario de prueba.

Número de obstáculos móviles	Número de reconexiones	Tiempo de reconexión (seg)	Número de replanificaciones	Camino no encontrado	Colisión	Exito
10	36	0.0019	0		No	Sí
	41	0.0005	0		No	Sí
	58	0.0002	0		No	Sí
	149	0.011	1	Sí	Sí	No
	9	0.005	0		No	Sí
15	4	0.0012	0		No	Sí
	154	0.0022	1	Sí	Sí	No
	8	0.0006	0		No	Sí
	24	0.0016	0		No	Sí
	65	0.0010	0		No	Sí

4.1. Implementación en tiempo real

Una vez que se realizaron las pruebas a nivel simulación, en este trabajo también nos planteamos la posibilidad de realizar una implementación de las estrategias reactivas en entornos reales haciendo uso del robot Pioneer 3-DX. Esta implementación se basó en el simulador en gran medida. La diferencia radica principalmente en la conexión con un agente externo, que es el Pioneer 3-DX.

Para esto se optó por hacer uso de las librerias de ARIA con el lenguaje de programación C++, ya que nos brinda las herramientas robustas para comunicar y controlar al Pioneer 3-DX. Además, la versión real encontrará caminos para robots móviles con restricciones de movimiento no holonómicas o sin ellas.

En esta implementación, es necesario establecer un medio entre el programa y el Pioneer 3-DX para enviar y recibir comandos de control e información. El Pioneer posee un microcontrolador integrado que gestiona a bajo nivel los motores, sensores y demás componentes. Además, el robot es capaz de monitorear su estado intrínseco (incluyendo la configuracin actual global) mediante estimaciones con un grado de error considerablemente bajo. Para dicha conexión se utilizó un cable Serial a USB entre el Pioneer 3-DX y la PC montada sobre la cubierta del mismo. La Figura 7 muestra al robot y al equipamiento.

Fig. 7. Montaje y conexión del robot Pioneer 3-DX.

La conexión establecida entre el programa y el robot puede gestionarse de manera síncrona o asíncrona. En nuestro programa se decidió trabajar de manera

asíncrona para poder monitorear en todo momento el estado del robot Pioneer y presentarlo en pantalla, mientras en segundo plano se ejecuta todo el procedimiento, ver la estructura de la comunicación en tiempo real de nuestra prueba en tiempo real. Por otro lado, el sensor láser Hokuyo URG (modelo 04LX-UG01) será conectado directamente a la PC mediante un cable miniUSB a USB sin pasar información al robot. Esto con el fin de no elevar el procesamiento que deba hacer el microcontrolador del Pioneer, siendo el microprocesador del PC el que gestione dicho trabajo (vase la Figura 8).

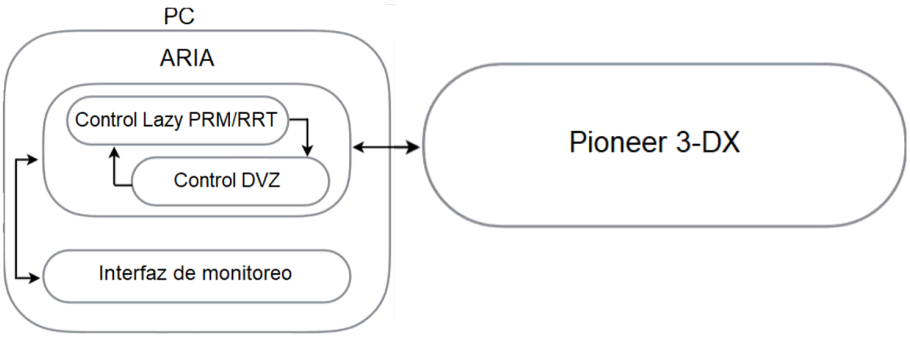

Fig. 8. Diagrama que muestra la estructura de alto nivel de la comunicación en el escenario de prueba que muestra las estrategias reactivas propuestas en tiempo real de las estrategias reactivas.

El Pioneer 3-DX es un robot con tracción diferencial, esto quiere decir que posee dos ruedas propulsadas y controladas independientemente montadas en un único eje. Esto permite al Pioneer una habilidad de movimiento superior a un robot de tipo carro, pues es capaz de cambiar su orientación sin movimientos de traslación (rotación sobre su propio eje). A pesar de ser un robot diferencial, el Pioneer puede realizar el recorrido de caminos compuestos por curvas Reeds & Shepp.

La parte experimental se realizó teniendo en cuenta que el robot es capaz seguir una trayectoria geométrica previamente calculada por un método probabilístico (Lazy PRM o RRT), consideramos un modelo del medio ambiente en escala. En ausencia de obstáculos, el robot sigue la trayectoria hasta llegar a la región meta, si hay obstádesconocidos, el robot ejecuta controles reactivos para evitarlos y volver a su trayectoria.

La Figura 9 ilustra este experimento en tiempo real en donde el robot evita obstáculos desconocidos. Uno puede ver que el robot claramente evita el obstáculo y vuelve al camino nominal.

Una problema importante que resolvimos para la parte experimental, fue el control reactivo por ZVD. Que a continuación bosquejamos brevemente.

En la descripción del método de control por ZVD se maneja el hecho de tener un núomero $n_{sensores}$ de sensores utilizados para realizar mediciones hacia

Fig. 9. Ejemplo de una ejecución en tiempo real.

el ambiente en una dirección dada. Sin embargo, el sensor láser Hokuyo URG es capaz de realizar más de 600 mediciones cada 100 milisegundos abarcando un arco de medición de 240^0. Esto implica dos problemas. El primer problema es la falta de sensores o lecturas hacia la parte trasera del robot.

A pesar de que la teoría del control por ZVD indica la necesidad de mediciones en todas direcciones a partir del robot, se optó por omitir dichas mediciones debido a la limitación del sensor láser Hokuyo URG. Esto puede provocar que un movimiento en reversa generado por un comando reflejo haga colisionar al Pioneer con un obstáculo situado justo detrás de él.

Para disminuir la posibilidad de esta colisión, se utiliza la información del ambiente y antes de la ejecución del comando reflejo se verifica que se dirija a

una configuracin libre de colisión. En caso de detectar que estáo muy próximo a un obstáculo en el mapa del ambiente y que el comando reflejo planea dirigirlo hacia él, el procedimiento terminará la ejecución y reportará fallo por la imposibilidad de generar un comando reflejo capaz de devolver la ZVD a su estado no deformado.

El otro problema es el número de posibles mediciones por parte del sensor láser: 683 mediciones. Si tomáramos este valor como $n_{sensores}$, la cantidad de cálculos por parte del control por ZVD se elevaría extremadamente haciéndolo imposible. La forma de evitarlo fue mantener a $n_{sensores}$ como un parámetro gestionado por el usuario y dividir el arco completo de medición del sensor láser en $2n_{sensores}1$ regiones del mismo ángulo de apertura para tomar a cada región impar r_i como el sensor c_i. La menor distancia obtenida dentro de la regin r_i será la medición para el sensor c_i. El resto del procedimiento del control por ZVD se mantiene de la misma manera.

5. Conclusiones y trabajo futuro

La parte reactiva estudiada en este trabajo muestra que los planificadores basados en muestreo son buenas opciones para resolver el problema de los robots móviles en presencia de obstáculos móviles, pero estas estrategias reactivas no ofrecen una solución completa para todos los casos, debido a la complejidad del problema de planificación de movimientos en ambientes dinámicos.

Otro método reactivo implementado en este trabajo que se propuso en [4], utilizó un planificador de tipo PRM para llevar a cabo su planificación, y resolver la parte reactiva sin usar las acciones reflejas proporcionadas por el método de la ZVD. Esta modalidad es muy rápida, pero en una fuerte presencia de obstáculos en movimiento, el tiempo de reacción aumenta en consecuencia.

Las estrategias de replanificación de caminos requieren que el sistema diseñe un camino seguro, entre obstáculos, que asegure que el vehículo evite los obstáculos y finalmente alcance el objetivo deseado. La ventaja de estos métodos es que, bajo la suposición que si hay un mapa preciso del entorno disponible, diseñar una solución global y garantizar que el sistema alcanzará su objetivo, si es que la solución existe.

Además, dado que el sistema está necesariamente equipado con sensores de proximidad, el requisito para construir un mapa de ambiente, se cumple. Esta consideración permite el uso de SLAM para sistemas de navegación precisos, en los que los métodos de replanificación de caminos ocurren naturalmente. Sin embargo, el inconveniente aquí, es el tiempo computacional necesario, que puede hacer que el sistema se detenga frente a un obstáculo desconocido, incluso si este fenómeno ocurre con robots reales. Finalmente, el objetivo del control es seguir el camino calculado en seguridad.

Referencias

1. Bruce, J., Veloso, M.: Real-time randomized path planning for robot navigation. In: IEEE Int. Conf. on Intelligent Robots and Systems, pp. 2383–2388 (2002)

2. Esposito, J.M., Kim, J., Kumar, V.: Adaptive RRTs for validating hybrid robotic control systems. In: Proc. Workshop on Algorithmic Foundation of Robotics (2004)
3. Fergunson, D., Kalra, N., Stentz, A.: Replanning wth RRTs. In: IEEE Int. Conf. on Robotics and Automation, pp. 1243–1248 (2006)
4. Jaillet, L., Siméon, T.: A PRM-based motion planner for dynamically changing environments. In: IEEE Int. Conf. on Intelligent Robots and Systems, pp. 1606–1611 (2004)
5. Kavraki, L.E., Svestka, P., Latombe, J.C., Overmars, M.H.: Probabilistic roadmaps for path planning in high-dimensional configuration spaces. IEEE Transactions on Robotics and Automation 12(49), pp. 566–579 (1996)
6. LaValle, S.M., Kuffner, J.J.: Rapidly-exploring random trees: Progress and prospects. Proc. Workshop on the Algorithmic Foundations on Robotics (2000)
7. Lapierre, L., Zapata, R., Lepinay, P.: Simultaneous path following and obstacle avoidance control of unicycle-type robot. In: IEEE Int. Conf. on Robotics and Automation, pp. 2617–2622 (2007)
8. LaValle, S.M.: Planning algorithms. Cambridge University Press (2006)
9. Leven, P., Hutchinson, S.: Toward real-time path planning in changing environments. In: Proc. Workshop on Algorithmic Foundation of Robotics (2000)
10. Sánchez, A., Zapata, R., Cuautle, R., Osorio, M.A.: Reactive motion planning for mobile robots. Mobile Robots Motion Planning, New Challenges, I-Tech Education and Publishing, pp. 469–486 (2008)
11. Sarabia-Cortés, G.: Estrategias reactivas para la planificación dinámica de movimientos en robótica móvil. BS Thesis, (FCC-BUAP) (2009)
12. Zucker, M., Kuffner, J., Branicky, M.: Multipartite RRTs for rapid replanning in dynamic environments. In: IEEE Int. Conf. on Intelligent Robots and Systems, pp. 1603–1609 (2007)
13. Zapata, R., Lépinay, P., Thompson, P.: Reactive behaviors of fast mobile robots. Journal of Robotic Systems 11(1), pp. 13–20 (1994)
14. Sussmann, H.J., Tang, G.: Shortest paths for the Reeds-Shepp car: A worked out example of the use of geometric techniques in nonlinear optimal control. Report SYCON, pp. 91–10 Rutgers University (1991)