

Análisis de un esquema de control de comportamientos basado en sub-espacios nulos para robots móviles

Analysis of the null-space-based behavioral control for autonomous robotic systems

Andres Felipe Acosta Gil¹ Ing., Roberto Carlos Hincapié Reyes¹ Ph. D. & Gustavo Alonso Acosta Amaya² M. Sc.

1. Universidad Pontificia Bolivariana. Colombia

2. Politécnico Jaime Isaza Cadavid. Medellín, Colombia.

andresfelipe.acosta@alfa.upb.edu.co, coordinador.roberto@upb.edu.co, gacostaa@elpoli.edu.co.

Recibido para revisión 18 de junio de 2011, aceptado 18 de octubre de 2011, versión final 24 de noviembre de 2011

Resumen — En éste artículo se presenta el análisis y simulación del modelo de control reactivo conocido como Null-Space-Based Behavioral Control (NSB) para robots móviles [1]. El método consiste en dividir una tarea compleja en tareas más sencillas denominadas comportamientos, las cuales se ordenan de acuerdo a su prioridad y se combinan de tal manera que las de menor prioridad, se proyectan en el sub-espacio nulo de las de mayor prioridad. La simulación llevada a cabo incorpora: i) un método geométrico para la activación de la evasión de obstáculos, y ii) el análisis de la activación de comportamientos y velocidades del robot móvil.

Palabras clave — Inteligencia artificial, agente robótico, control reactivo, control basado en comportamientos.

Abstract — In this paper, a simulation analysis of the Null-Space-Based Behavioral Control (NSB) for mobile robots is presented [1]. The approach divides a complex problem in several simpler tasks (behaviors) which are sorted by priority and combined in such a way that a lower priority task is projected into the null-space (kernel) of the higher priority one. The simulation performed includes: i) a geometric approach to activate the obstacle avoidance task; and ii) system velocities and obstacle avoidance task activation are analyzed.

Keywords — Artificial Intelligence, mobile robot, reactive control, behavioral control.

I. INTRODUCCIÓN

Las aplicaciones de la robótica son cada vez más numerosas. En particular, la robótica móvil empleada en aplicaciones de campo y de servicios ha experimentado grandes avances en la última década. La robótica de servicios, por ejemplo, constituye

un mercado que maneja actualmente millones de dólares, con predicciones de crecimiento del orden de billones de dólares [2].

En el campo de la robótica móvil aún quedan problemas que no han sido completamente resueltos, como la navegación en entornos no estructurados y dinámicos [3] [4], incremento en el grado de autonomía e inteligencia, un mejor conocimiento del entorno en términos de representaciones e interacción con los seres humanos [5] [6], por mencionar solo algunos.

En este artículo se aborda el problema la navegación en entornos no estructurados y dinámicos desde la perspectiva del control basado en comportamientos propuesto por Brooks [7], que se ha constituido en la principal alternativa del control reactivo. En [8], Arkin establece las principales características del control reactivo:

- La percepción y la acción están relacionadas estrechamente permitiendo la actuación en ambientes dinámicos y no estructurados.
- Los comportamientos se definen a partir de los objetivos para los cuales fue construido el agente robótico, son priorizados según el estado del entorno y modelados mediante funciones vectoriales.
- El comportamiento del robot depende de la “realidad” y no de representaciones de ésta. La inteligencia surge de la interacción entre el robot y su entorno.
- Comportamientos emergentes complejos surgen de la combinación de comportamientos individuales más simples.

Estas características son estudiadas ampliamente en [9] [10], donde además se tratan arquitecturas deliberativas e híbridas.

El paradigma reactivo ha sido objeto de numerosas investigaciones. En [11], se estudia una arquitectura de control de comportamientos evaluada experimentalmente en escenarios de navegación y búsqueda de rutas, control multi-agente y aprendizaje de selección de comportamientos. Jaeger y Christaller [12] proponen un esquema que denominan Dual Dynamics, que sirve como modelo formal para el control de comportamientos en el marco de la teoría de sistemas dinámicos y bifurcaciones. Por su parte, Nicolescu y Mataric [13] proponen una arquitectura de control donde se modelan los comportamientos de forma abstracta, lo que permite incorporar una característica propia del control deliberativo como lo es la ejecución secuencial de tareas, posibilitando también la reutilización de comportamientos.

Varios autores coinciden en que los dos paradigmas de control se pueden combinar en una arquitectura híbrida que permita contrarrestar sus desventajas. En [14], se presenta una arquitectura de control basada en teoría de agentes, en la cual se identifican agentes de control reactivo, planificación, localización y de control de bajo nivel, cuyas salidas se combinan utilizando técnicas de lógica fuzzy. En [15], se utiliza un controlador deliberativo que genera una serie de puntos de chequeo a los que se accede mediante un controlador reactivo basado en una red neuronal, el cual es responsable además de la evasión de obstáculos. Stoytchev y Arkin [16] proponen también una arquitectura híbrida para enfrentar ambientes dinámicos y procesar comandos de alto nivel. Esta arquitectura incorpora un subsistema que permite al robot interactuar adecuadamente con seres humanos.

Un problema relativo a la arquitectura de control reactivo es el de los métodos de combinación de comportamientos individuales para producir comportamientos emergentes, lo que se conoce como control de comportamientos, coordinación de comportamientos o el problema de selección de acción (*Action Selection Problem* [17]). Se destacan dos métodos de control de comportamientos [8]: los competitivos y los cooperativos.

A. Métodos competitivos

Este método consiste en la selección de un solo comportamiento por medio de un mecanismo de arbitraje que usualmente está determinado por el estado actual del entorno y los objetivos del agente, lo cual puede considerarse como un proceso mediante el cual cada comportamiento individual compite para ser activado. Uno de los métodos competitivos más destacado es el propuesto por Brooks [7], denominado *Layered Control System* (LCS) o arquitectura de subsunción. En éste, los comportamientos son organizados en una jerarquía que permite que los comportamientos de mayor prioridad, subsuman a los de menor prioridad. En caso de que el comportamiento de mayor prioridad produzca una salida nula, se selecciona el que se esté un nivel por debajo del mismo en la jerarquía. En esta arquitectura (figura 1), el comportamiento emergente corresponde al comportamiento seleccionado.

Para ilustrar el método, considérese un escenario en el cual un robot móvil es controlado por los comportamientos, en orden de prioridad, evasión de obstáculos e ir a la meta. Si el robot se encuentra a una distancia prudente de un obstáculo, definida por el diseñador, entonces se moverá únicamente bajo la acción del comportamiento ir a la meta, ya que la evasión de obstáculos produce una salida nula. Por otro lado, si el robot se acerca al obstáculo, igualando o sobrepasando el límite de distancia asignado, se ejecuta el comportamiento de evasión de obstáculos en lugar del comportamiento ir a la meta, debido a que el primero produce una salida no nula, subsumiendo al segundo.

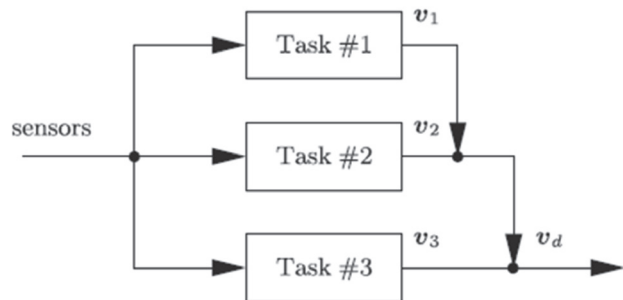


Figura 1. Diagrama del esquema competitivo *Layered control System* (LCS) [1].

B. Métodos Cooperativos.

También conocidos como fusión de comportamientos, se caracterizan por que, a diferencia de los esquemas competitivos, el aporte de varios comportamientos determina la acción final de control, lo cual se logra por medio de mecanismos como los sistemas de votación, las combinaciones lineales y los basados en lógica fuzzy.

En [18] se presenta un sistema de control basado en comportamientos para un vehículo autónomo submarino empleado en la inspección de arrecifes de coral. Se propone una arquitectura distribuida de comportamientos que se comunican con un sistema central de arbitraje. Los comportamientos emplean políticas de votación a favor y en contra de las acciones que se desean implementar. El sistema de arbitraje es responsable de combinar la votación alcanzada por cada comportamiento para generar las acciones que serán enviadas al controlador del vehículo

La figura 2 ilustra el método *Motor Schema Control* (MSC) [19], que corresponde a un caso de control cooperativo basado en una combinación lineal. Se puede notar la presencia de un módulo supervisor el cual se encarga de asignar la ganancia relativa a cada comportamiento.

En éste artículo se presenta un análisis basado en la simulación del esquema de control NSB propuesto por Antonelli et al. [1], en el cual se incorpora: i) un método geométrico para la activación del comportamiento de evasión de obstáculos y

ii) gráficos temporales de activación de comportamientos y sus velocidades asociadas.

Este artículo se organiza de la siguiente manera: en la sección 2 se presenta el modelo matemático del método NSB, en la sección 3 se describe el escenario de simulación y el método de activación del comportamiento evasión de obstáculos, en la sección 4 se presentan los resultados de simulación y su análisis, haciendo una comparación de los métodos competitivo y colaborativo. Finalmente en la sección 5 se presentan las conclusiones y el trabajo futuro.

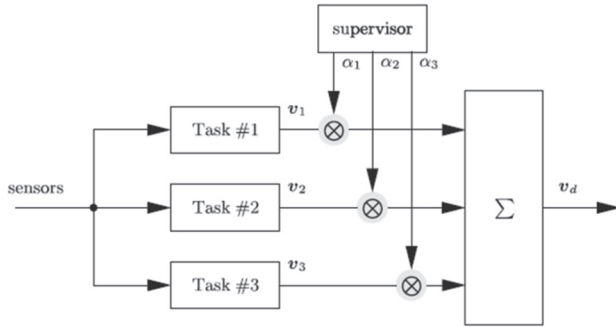


Figura 2. Diagrama del esquema colaborativo *motor schema control* (MSC).

II. MODELO TEÓRICO

Sea $\mathbf{p} \in \mathbb{R}^n$ la configuración de un robot móvil. Se define una función de tarea genérica o comportamiento $\boldsymbol{\sigma} \in \mathbb{R}^m$ como:

$$\boldsymbol{\sigma} = \mathbf{f}(\mathbf{p}) \quad (1)$$

Con su relación diferencial dada por:

$$\dot{\boldsymbol{\sigma}} = \frac{\partial \mathbf{f}(\mathbf{p})}{\partial \mathbf{p}} \mathbf{v} = \mathbf{J}(\mathbf{p}) \mathbf{v} \quad (2)$$

Siendo $\mathbf{J} \in \mathbb{R}^{m \times n}$ la matriz Jacobiana de la tarea $\boldsymbol{\sigma}$ y $\mathbf{v} \in \mathbb{R}^n$ la velocidad del robot móvil que se obtiene al derivar la ecuación (1). En éste punto se pueden generar velocidades de referencia a partir de los valores deseados de la función de tarea, una técnica común en el control de manipuladores robóticos. Dado que en general, la función de tarea es de una dimensión menor a la de la configuración del robot ($m < nm < n$), entonces la matriz Jacobiana es rectangular, y el sistema posee infinitas soluciones. Generalmente se escoge la solución de menor magnitud que implica una optimización de la velocidad por mínimos cuadrados. De ésta manera, la solución puede ser expresada como:

$$\mathbf{v}_d = \mathbf{J}^{\dagger} \dot{\boldsymbol{\sigma}}_d = \mathbf{J}^{\dagger} (\mathbf{J} \mathbf{J}^{\dagger})^{-1} \dot{\boldsymbol{\sigma}}_d \quad (3)$$

Siendo $\mathbf{J}^{\dagger} \in \mathbb{R}^{n \times m}$, la matriz jacobiana pseudo-inversa y

$\dot{\boldsymbol{\sigma}}_d \in \mathbb{R}^m$, el valor deseado de la función de tarea.

Es posible que el controlador de movimiento del robot móvil necesite posiciones de referencia en lugar de velocidades. Éstas pueden ser obtenidas mediante la integración numérica de la ecuación (3). Sin embargo, pueden resultar desviaciones respecto del valor teórico, lo cual se puede contrarrestar por medio de la cinemática inversa de lazo cerrado (Closed-Loop Inverse Kinematics CLIK), así:

$$\mathbf{v}_d = \mathbf{J}^{\dagger} (\dot{\boldsymbol{\sigma}}_d + \boldsymbol{\Lambda} \tilde{\boldsymbol{\sigma}}) \quad (4)$$

Con:

$$\tilde{\boldsymbol{\sigma}} = \boldsymbol{\sigma}_d - \boldsymbol{\sigma} \quad (5)$$

Donde $\boldsymbol{\Lambda}$ es una matriz de ganancias definidas por el diseñador.

Suponiendo que se tienen diferentes funciones de tarea, las cuales se denotan por medio del subíndice i se obtiene de (4), para cada tarea:

$$\mathbf{v}_i = \mathbf{J}_i^{\dagger} (\dot{\boldsymbol{\sigma}}_{i,d} + \boldsymbol{\Lambda}_i \tilde{\boldsymbol{\sigma}}_i) \quad (6)$$

Finalmente, si el subíndice i denota también el grado de prioridad de la tarea, entonces el NSB establece que el comportamiento emergente se obtiene, considerando solo tres tareas:

$$\mathbf{v}_d = \mathbf{v}_1 + \mathbf{N}_1 [\mathbf{v}_2 + \mathbf{N}_2 \mathbf{v}_3] \quad (8)$$

Definiendo el operador \mathbf{N}_i de proyección sobre el sub-espacio nulo de la i -ésima tarea como:

$$\mathbf{N}_i = \mathbf{I} - \mathbf{J}_i^{\dagger} \mathbf{J}_i \quad (9)$$

Se llega a:

$$\mathbf{v}_d = \mathbf{v}_1 + (\mathbf{I} - \mathbf{J}_1^{\dagger} \mathbf{J}_1) [\mathbf{v}_2 + (\mathbf{I} - \mathbf{J}_2^{\dagger} \mathbf{J}_2) \mathbf{v}_3] \quad (10)$$

En la ecuación (10) se observa la forma como se compone un comportamiento emergente. Se calcula cada comportamiento individual y se organiza según su prioridad dependiendo del contexto del entorno. Antes de sumar las contribuciones, los comportamientos de menor prioridad son proyectados en el sub-espacio nulo de los de mayor prioridad. Puesto que los sub-espacios nulos son ortogonales a los espacios de su función tarea asociada, se garantiza que no hay componentes de las tareas de menor prioridad que interfieran con aquellas de mayor prioridad. Lo anterior significa que la tarea de mayor prioridad se ejecuta completamente, mientras que las de menor prioridad se ejecutan en un espacio que no entra en conflicto con la de mayor prioridad. El esquema del NSB se ilustra en la figura 3.

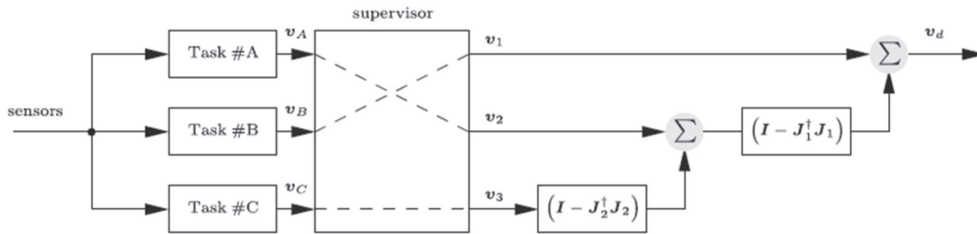


Figura 3. Diagrama del esquema de control de comportamientos basado en sub-espacios nulos.

III. ESCENARIO DE DEMOSTRACIÓN

El modelo teórico presentado en la sección anterior se probó en un escenario en el cual se definieron dos tareas: “ir a la meta” y “evasión de obstáculos”.

Se consideraron los siguientes supuestos: i) un robot holonómico con dos grados de libertad (x, y); ii) tanto el robot como el obstáculo se modelaron como puntos; iii) no se consideraron perturbaciones ni ruido y iv) las posiciones del robot y el obstáculo eran conocidas en todo momento.

Las tres primeras asunciones simplifican la simulación resaltando el esquema de control de comportamientos. La última asunción se asocia a modelos basados en funciones vectoriales, como por ejemplo, los campos potenciales, y se sustenta en los avances recientes en fusión sensorial y estimación de la posición [20] [21] [22] [23].

El modelamiento de las funciones de tarea es como sigue.

3.1 Evasión de obstáculos

La evasión de obstáculos es una tarea mono dimensional, corresponde a la distancia entre el robot móvil y el obstáculo:

$$\sigma_1 = \|p - p_0\| \in \mathbb{R} \tag{10}$$

$$\sigma_{1,d} = d \tag{11}$$

$$J_1 = \hat{r}^T \in \mathbb{R}^{1 \times 2} \tag{12}$$

Siendo p_0 la posición del obstáculo, $\sigma_{1,d}$ el valor deseado para esta tarea y \hat{r} el vector unitario en la dirección robot-obstáculo dado por:

$$\hat{r} = \frac{p - p_0}{\|p - p_0\|} \tag{13}$$

Reemplazando las ecuaciones (10) y (11) en (6) y la (12) en (9) en se llega a:

$$v_1 = J_1^T \lambda_1 (d - \|p - p_0\|) \tag{14}$$

$$N_1 = I - \hat{r} \hat{r}^T \tag{15}$$

El valor deseado $\sigma_{1,d}$ es la distancia mínima que debe mantenerse entre el robot y el obstáculo. Se puede notar en la ecuación (14) a medida que la distancia robot-obstáculo se aleja del valor d , la magnitud de la velocidad crece.

Modelo geométrico para la activación de la evasión de obstáculos

En la simulación, el comportamiento evasión de obstáculos se activa solo cuando el robot detecta el obstáculo y su velocidad se dirige en dirección al mismo. Para verificar estas condiciones, considérese la figura 4.

En donde:

- thr (threshold) es la distancia a partir de la cual se detecta el obstáculo
- d es la distancia máxima permitida entre el robot y el obstáculo
- r vector en dirección robot-obstáculo
- v_d velocidad del robot, puede ser producto o no de la combinación de varios comportamientos
- θ ángulo entre r y v_d
- rng ángulo entre r y la línea tangente a la circunferencia de radio d que pasa por el robot
- Se define la zona de detección como aquella donde $\sigma_1 < thr$
- Se define la zona prohibida como aquella donde $\sigma_1 < d$

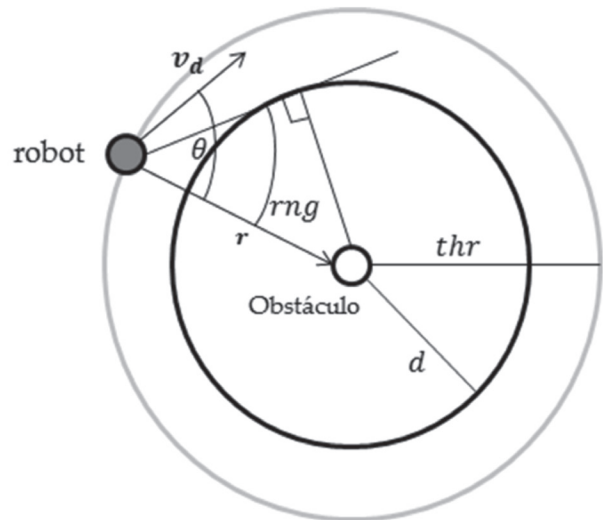


Figura 4. Modelo geométrico para la activación de evasión de obstáculos

La evasión del obstáculo se activa si se cumple que:

$$\|r\| < thr \wedge \theta < rng \quad (16)$$

Se espera además que ésta tarea entregue una velocidad no nula cuando $\sigma_1 < d$. En tal caso, la velocidad entregada por la función de tarea de menor prioridad “ir a la meta”, se multiplica por la ecuación (15) antes de ser sumada a (14), logrando así la proyección sobre el sub-espacio nulo de la tarea “evadir obstáculo”.

Finalmente, a partir de un análisis trigonométrico, se obtiene:

$$\theta = \cos^{-1}\left(\frac{r \cdot v_d}{\|r\| \|v_d\|}\right) \quad (17)$$

$$rng = \Re\left[\sin^{-1}\left(\frac{d}{\|r\|}\right)\right] \quad (18)$$

3.2 Ir a la meta

La función de tarea “ir a la meta” se define como:

$$\sigma_2 = p \in \mathbb{R}^{2 \times 1} \quad (19)$$

$$\sigma_{2,d} = p_g \in \mathbb{R}^2 \quad (20)$$

$$J_2 = I \in \mathbb{R}^{2 \times 2} \quad (21)$$

Donde p_g es la posición de la meta.

Llevando las ecuaciones (19) y (20) a la ecuación (6) se llega a:

$$v_2 = \Lambda_2(p_g - p) \quad (22)$$

Tal como se mencionó anteriormente Λ_2 es una matriz de ganancias de lazo cerrado cuyos valores son elegidos por el diseñador.

Como lo indica la ecuación (22), la velocidad es proporcional a la diferencia entre la posición de la meta y la posición del robot. Por lo tanto se debe incorporar en la simulación un nivel de saturación.

IV. SIMULACIÓN Y ANÁLISIS

Teniendo en cuenta los comportamientos y asunciones establecidas en secciones anteriores, se implementó un algoritmo de simulación en Matlab con los siguientes parámetros:

Descripción	Valor
Posición inicial	[0,0] cm
Posición de la meta (p_g)	[40,-3] cm
Posición del obstáculo (p_0)	[20,-5] cm
thr	10 cm (detección de obstáculo)
d	8 cm (evasión de obstáculo)
λ_1	10 (ganancia CLIK evasión de obstáculos)
Λ_2	I (ganancia CLIK ir a la meta)
Δt	50 ms
Saturación de la velocidad	5 cm/s

Resultados con el Layered Control System (LCS)

En la figura 5 se puede ver la trayectoria obtenida con el método competitivo LCS. Las tareas están organizadas en una jerarquía tal que la evasión de obstáculo, tiene mayor prioridad que ir a la meta. Inicialmente el robot se mueve bajo la acción de la tarea “ir a la meta”, ya que la tarea “evasión de obstáculo” produce una velocidad nula. Una vez la distancia entre robot y el obstáculo es menor que d , la tarea “evasión de obstáculo” produce una velocidad no nula la cual subsume la producida por “ir a la meta”, es decir, el robot se mueve únicamente bajo la acción de la tarea “evadir obstáculo”. De ésta manera el robot ingresa y se aleja de la zona prohibida en repetidas ocasiones (figura 5b), hasta que finalmente se aleja del obstáculo (figura 5a). El tiempo de simulación es de 17.95 segundos, el mayor obtenido con respecto a los demás métodos simulados.

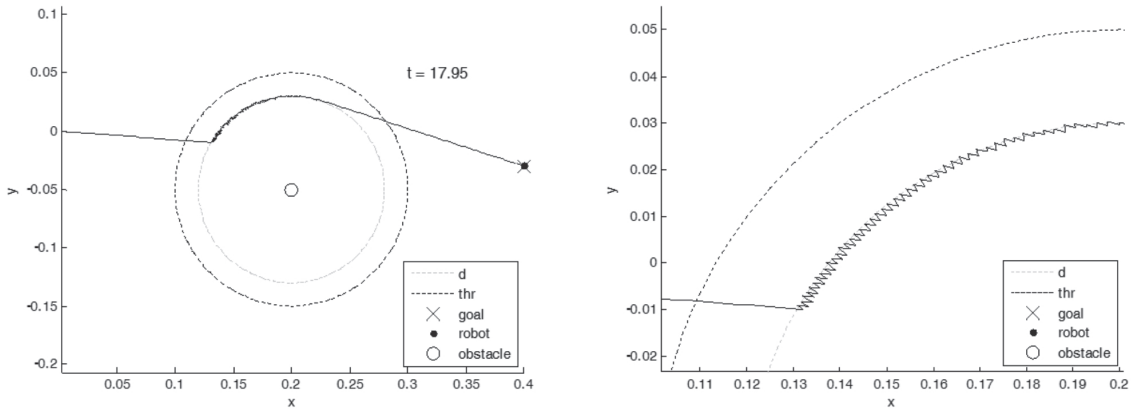


Figura 5. a) Trayectoria obtenida con el LCS en un escenario de movimiento con evasión de obstáculo. b) Zoom de la figura a) donde se evidencia que el robot ingresa y sale de la zona prohibida en repetidas ocasiones.

Resultados con el Motor Schema Control (MSC)

En la figura 6 se muestra la trayectoria obtenida en la simulación del método MSC. Cuando el robot ingresa en la zona prohibida, y dado que la velocidad resulta de una combinación lineal de las tareas de evasión de obstáculo e ir a la meta, hay componentes de la segunda tarea sobre el espacio de la primera que hacen que el robot ingrese y permanezca durante algún tiempo en la zona prohibida. A pesar de que el robot ingresa a la zona prohibida, alcanza a evadir el obstáculo. Esta simulación se completa en 9.95 segundos.

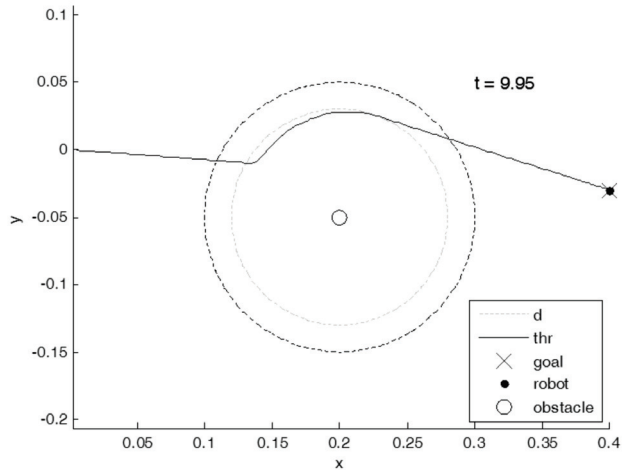


Figura 6. Trayectoria obtenida con el método MSC en un escenario de movimiento con evasión de obstáculo.

Resultados con el Null-Space-Based Behavioral Control (NSB)

En la figura 7a, se muestra la trayectoria obtenida mediante el NSB. Se puede observar que cuando hay evasión de obstáculo, de la tarea de segunda prioridad se tiene en cuenta solamente la componente ortogonal a la de primera prioridad, es decir, en dirección tangente a la circunferencia, resultando en un movimiento de deslizamiento a lo largo de la misma¹. Lo anterior se puede evidenciar en la figura 7b, donde se observa que la velocidad producida por la tarea “ir a la meta” es de gran magnitud, ya que la distancia entre el robot y la meta es relativamente grande. La velocidad producida por la tarea de evasión de obstáculos se encuentra en dirección radial respecto a la circunferencia de radio d , su magnitud es mucho menor debido a que la diferencia entre la distancia robot-obstáculo y el parámetro d es muy pequeña, lo que significa que el robot ingresó levemente en la zona prohibida e inmediatamente salió de ella bajo la acción de la tarea “evadir obstáculo”. Finalmente, la velocidad calculada con el NSB se puede observar en dirección tangente a la circunferencia.

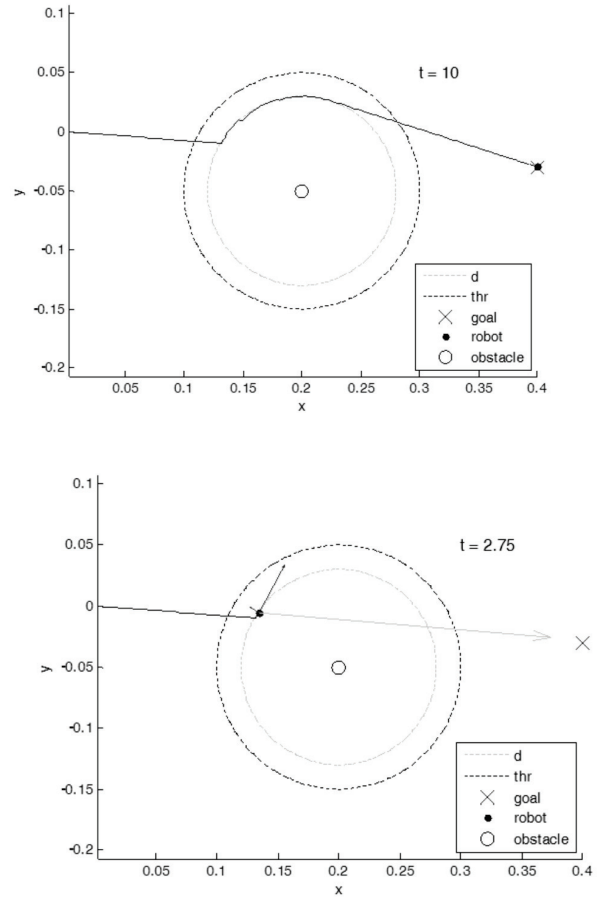


Figura 7. a) Trayectoria obtenida con el NSB en un escenario de movimiento con evasión de obstáculo. b) Comportamiento de las velocidades obtenidas con el método NSB.

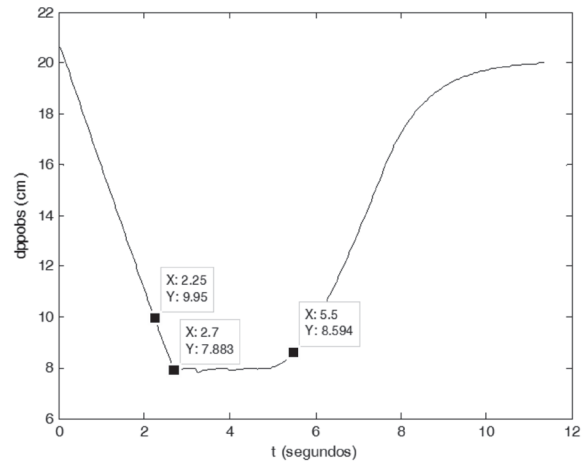


Figura 8. Distancia entre el robot y el obstáculo en función del tiempo.

La figura 8 muestra la distancia entre el robot y el obstáculo en función del tiempo. El robot ingresa en la zona de detección en 2.25 segundos, tiempo a partir del cual se cumple la primera parte de la condición (16). Luego se acerca al obstáculo e ingresa en la zona prohibida a los 2.7 segundos. A partir de allí ingresa

1. Con el fin de ofrecer una mejor ilustración del comportamiento obtenido con los diferentes métodos, los videos de la simulación se han publicado en: <http://www.youtube.com/user/acostarobots?blend=12&ob=5>

esporádicamente a esta zona, saliendo rápidamente de ella y manteniéndose, la mayor parte del tiempo, a la distancia mínima de 8 cm. Después de los 5 segundos, el robot comienza a alejarse del obstáculo, tiempo en el cual se sigue cumpliendo la primera parte de la condición (16) hasta que el robot sale de la zona de detección.

En la figura 9 se muestra el comportamiento de las variables θ y rng en el tiempo. Se observa que en 2.7 segundos, se cumple la segunda parte de la condición (16), por lo tanto, se activa la tarea de evasión de obstáculos la cual produce una velocidad de 1.17 cm/s, como puede verse en la figura 10.

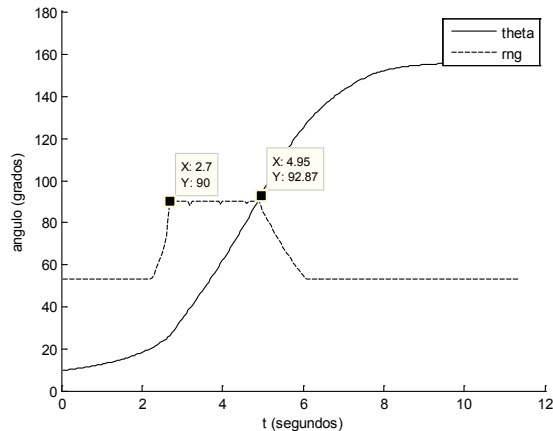


Figura 9. Valor de las variables θ y rng en función del tiempo.

A partir de los 4.95 segundos, se puede concluir de la figura 9 que no se cumple la segunda parte de la condición (16), ya que $\theta > rng$, lo que significa que el robot no se dirige hacia el obstáculo. Lo anterior hace que aunque en 5.5 segundos se cumpla la primera parte de la condición (16) (figura 8), la evasión de obstáculos se encuentra desactivada (figura 10).

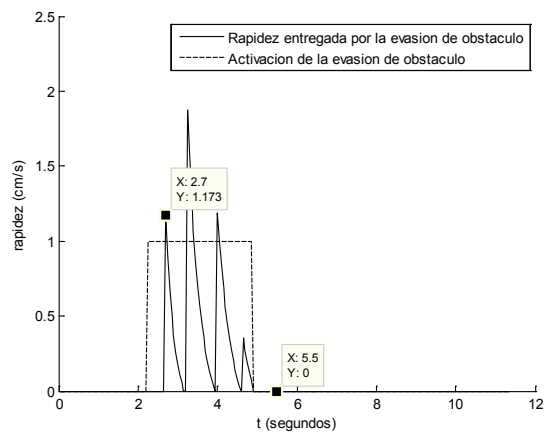


Figura 10. Activación de la tarea “evasión de obstáculos” en el tiempo.

V. CONCLUSIONES Y TRABAJO FUTURO

En este artículo se presentó un análisis del esquema de control de comportamientos basado en sub-espacios nulos (NSB) a partir de su simulación. Los resultados fueron comparados con los esquemas LCS y MSC, como casos de los métodos competitivos y colaborativos respectivamente. El NSB combina las tareas ordenándolas por prioridad y luego proyecta las de menor prioridad en los sub-espacios nulos de las de mayor prioridad. Los resultados de simulación muestran que la tarea de mayor prioridad siempre es llevada a cabo a plenitud, éstos concuerdan con los del artículo en el cual se propone el método.

Adicionalmente se presentó un método geométrico para la activación de la tarea “evasión de obstáculos”. Los resultados muestran que ésta se ejecuta solamente cuando el robot está a una distancia determinada del obstáculo y se dirige hacia el mismo.

Debido a que el método se basa en funciones vectoriales, se presenta el problema de los mínimos locales. Dicho problema se puede encontrar en un escenario donde robot, obstáculo y meta se encuentren sobre la misma línea.

Se propone para un trabajo futuro, extender la simulación para robots no holonómicos y obstáculos no puntuales, y manejar el problema de los mínimos locales por medio de comportamientos como adición de ruido y seguimiento de paredes.

BIBLIOGRAFÍA

- [1] Antonelli G.; Arrichiello F. y Chiaverini S., 2008. The null-space-based behavioral control for autonomous robotic systems. En: Intelligent Service Robotics, Vol. 1, pp. 27-39.
- [2] Gelin R. y Christensen H., 2005. Sectoral Report on Service Robotics. En: <ftp://ftp.cordis.europa.eu/pub/ist/docs/europ/rob-plat-4.pdf>
- [3] García E. y González de Santos P., 2004. Mobile-robot navigation with complete coverage of unstructured environments. En: Robotics and Autonomous Systems, Vol. 46, pp. 195-204.
- [4] Ferrein A. y Lakemeyer G., Logic-based Robot Rontrol in Highly Dynamic Domains. En: Robotics and Autonomous Systems, Vol. 56, pp. 980-991.
- [5] Feil-Seifer D. y Matari’c M., 2005. Defining Socially Asistive Robotics., 2005 En: Proceedings of the 2005 IEEE 9th International Conference on Rehabilitation Robotics, pp. 465-468.
- [6] Severinson-Eklundh K.; Green A. y Hüttenrauch H., 2003. Social and collaborative aspects of interaction with a service robot. En: Robotics and Autonomous Systems, Vol. 42, pp. 223-234.
- [7] Brooks R., 1986. A robust layered control system for a mobile robot. En: Robotics and Automation, Journal of, Vol. 2, pp. 14-23.
- [8] Arkin R., 1998. Behavior-Based Robotics. The MIT press. 505 P
- [9] Molina JM. y Matellán V., 1996. Robots Autónomos: Arquitecturas y Control. En: <http://www.giaa.inf.uc3m.es/investigacion/publicaciones/publicacionesperiodicas/nacionales/Molina-Buran-1996/Molina-Buran-1996.pdf>.

- [10] Poza J L. y Posadas J L., 2009. Revisión de las arquitecturas de control distribuido. En: <http://riunet.upv.es/bitstream/handle/10251/6407/Arquitecturas%20de%20control%20distribuido.pdf?sequence=1>.
- [11] Mataric M., 1997. Behavior-Based Control: Examples from Navigation, Learning, and Group Behavior. En: *Journal of Experimental and Theoretical Artificial Intelligence*, Vol. 9, pp. 323-336.
- [12] Jaeger H. y Christaller T., 1997. Dual Dynamics: Designing Behavior Systems for Autonomous Robots. En: *Artificial Life and Robotics*, Vol. 2, pp. 76-79.
- [13] Nicolescu M. y Mataric M., 2002. A Hierarchical Architecture for Behavior-Based Robots. En: *AAMAS '02 Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 1*, pp. 227-233.
- [14] Gómez A.; Martínez H. y García P., 1998. Una arquitectura de agentes difusos para robots autónomos móviles. En: <http://ants.dif.um.es/~humberto/papers/1998-estylf.pdf>
- [15] Low K H.; Leow W K. y Ang Jr. M., 2002. A Hybrid Mobile Robot Architecture with Integrated Planning and Control. En: *In proc. AAMAS*, pp. 219-226.
- [16] Stoychev A. y Arkin R., 2001. Combining Deliberation, Reactivity, and Motivation in the Context of a Behavior-Based Robot Architecture. En: *Computational Intelligence in Robotics and Automation 2000, Proceedings 2001 IEEE International Symposium on*, pp. 290-295.
- [17] Pirjanian P., 1999. Behavior Coordination Mechanisms – State-of-the-art. En: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.109.2487&rep=rep1&type=pdf>.
- [18] Rosenblatt J.; Williams S. y Durrant-Whyte H., 2006. Behavior-Based Control for Autonomous Underwater Exploration. En: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=844166.
- [19] Arkin R., 1987. Motor Schema-Based Mobile Robot Navigation. En: *IEEE International Conference on Robotics and Automation*, pp. 264-271.
- [20] Crowley J., 1989. World Modeling and Position Estimation for a Mobile Robot Using Ultrasonic Ranging. En: *1989 IEEE Conference on Robotics and Automation ICRA*, Vol. 3, pp. 1574-1579.
- [21] Chenavier F. y Crowley J., 1992. Position Estimation for a Mobile Robot Using Vision and Odometry. En: *Proceedings of the 1992 IEEE International Conference on Robotics and Automation*, Vol. 3, pp. 2588-2593.
- [22] Garulli A. et al., 2005. Mobile robot SLAM for line-based environment representation. En: *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC '05. 44th IEEE Conference on*, pp. 2041-2046.
- [23] Kam M.; Zhu X. y Karata P., 1997. Sensor Fusion for Mobile Robot Navigation. En: *Proceedings of the IEEE*, Vol. 85, pp. 108-119