

# Control local de robots móviles basado en métodos estadísticos y algoritmos genéticos

Domingo Gallardo, Otto Colomina, Francisco Flórez,  
Pilar Arques, Patricia Company, Ramón Rizo  
Grupo i3a: Informática Industrial e Inteligencia Artificial  
Departamento de Tecnología Informática y Computación  
Universidad de Alicante  
San Vicente E-03080, Spain

## Resumen

El problema del control de un robot autónomo viene resolviéndose últimamente utilizando un control local de bajo nivel que proporciona competencias elementales (evitar obstáculos, seguir paredes, entrar en puertas) y una capa superior que utiliza dichas funcionalidades para la navegación efectiva y para la solución de problemas completos. De ahí que la obtención de técnicas eficientes para el control local es un elemento central para obtener una navegación correcta. En este artículo proponemos el diseño automático de esquemas locales de control basado en la generación off-line de conductas óptimas utilizando técnicas de computación evolutiva y en el aprendizaje de las reglas de actuación asociadas a dicho esquema mediante un método estadístico como es el Análisis de Componentes Principales.

**Palabras claves:** robots móviles, navegación, control reactivo, análisis de componentes principales, algoritmos genéticos.

## 1 Introducción

Existen dos tipos básicos de enfoques para controlar la navegación de un robot móvil: técnicas globales y locales. En las técnicas globales, como son los métodos geométricos, la programación dinámica o los métodos de campo de potencial (ver en [Latombe, 1991] resumen y referencias complementarias) se asume totalmente conocida la descripción geométrica del entorno en el que se va a mover el robot. Se trata de métodos potentes y eficaces para generar tanto trayectorias a seguir como secuencias de comandos a ejecutar. Son métodos usados para el control de robots que trabajan en entornos sin ninguna variabilidad y que realizan tareas repetitivas en las que se conocen en todo momento el valor de todas sus variables de estado.

Los métodos locales o reactivos, por el contrario, consideran que el robot va a moverse en un entorno no conocido a priori y proporcionan unas conductas estándar para reaccionar ante lecturas de los sensores del robot (evitar obstáculo, seguir pared, entrar en puerta, alinearse con objeto, etc.). Estas conductas (o *esquemas* en la terminología de Arkin [Arkin, 1990]) son

Método de control local	Esquemas
Histograma de campo de potencial	avanzar-evitando-obstáculos ir-a-objetivo entrar-por-puerta
Velocidad-Curvatura	avanzar-evitando-obstáculos ir-a-objetivo
Ventana Dinámica	avanzar-evitando-obstáculos ir-a-objetivo
Esquemas motores	esquemas configurables

Tabla 1: Esquemas de actuación susceptibles de ser implementados con cada uno de los métodos de control local. Ver en el texto principal las referencias correspondientes a cada uno de los métodos.

aplicables en gran número de entornos distintos y se suelen usar junto con un control de alto nivel que se encarga de secuenciarlas. Este tipo de control es el que vamos a utilizar en el presente trabajo.

## 1.1 Técnicas previas para el control local

Entre las propuestas de control local, cabe destacar el enfoque del *histograma de campo vectorial* ([Borenstein and Korem, 1991]), el método de velocidad-curvatura ([Simmons, 1996]), el método de ventana dinámica [D.Fox *et al.*, 1997] y el método del propio Arkin de esquemas motores [Arkin, 1987].

En la tabla 1 se comparan el tipo de esquemas de conducta susceptibles de ser implementados usando cada uno de estos enfoques. El enfoque de esquemas motores, pese a ser el más genérico, es muy complejo de llevar a la práctica por el gran número de parámetros que deben ser ajustados en las ecuaciones de control del robot. El mismo problema plantea el enfoque del histograma de campo de potencial. Un problema añadido de ambos métodos es que obtienen los comandos del robot en dos fases separadas. En la primera fase se obtiene la dirección objetivo en la que debe moverse el robot. En la segunda fase se generan los comandos de modificación de las velocidades lineales y angulares necesarios para conducir al robot en la dirección deseada. Este enfoque sólo es factible si consideramos que las aceleraciones aplicables al robot son infinitas y el robot puede realizar de forma instantánea los incrementos de velocidades. Sin embargo, la realidad es que las aceleraciones usadas en la navegación de robots móviles debe ser baja para obtener trayectorias suaves y no forzar a la estructura mecánica del robot a tensiones excesivas.

Por otra parte, los enfoques de Simmons y Fox han demostrado ser capaces de controlar con éxito robots móviles que actúan en entornos de oficina con gran cantidad de obstáculos y personas en movimiento (una universidad en el primer caso, y un museo en el segundo). Sin embargo, se trata de propuestas difícilmente generalizables a otro tipo de esquemas de conducta distintos de la evitación de obstáculos, como puede ser el seguimiento de una pared, la localización de esquinas o la localización y entrada en puertas abiertas.

Hay que hacer notar también que en todos estos enfoques se trabaja con información proporcionada por sensores de rango de baja densidad y

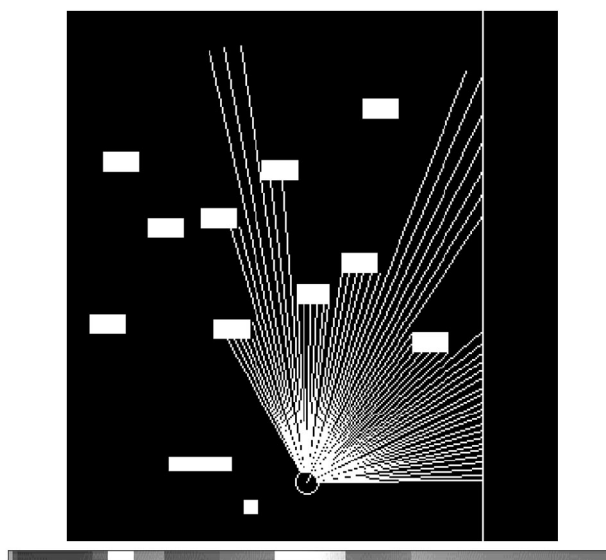


Figura 1: Una muestra del simulador sobre el que se ha realizado la experimentación del artículo. En la parte inferior de la figura se incluye el mapa de profundidad captado por el robot. Tonos de gris oscuros corresponden con lecturas de profundidad cercanas.

limitado alcance, como son los ultrasonidos. Esto provoca, entre otros, el problema denominado *aliasing perceptual*, en el que situaciones del robot distintas, en las que se deberían tomar acciones también distintas, se solapan en una misma percepción.

## 1.2 Control local basado en reconocimiento estadístico de situaciones

En línea con las propuestas de Arkin y Chapman [Chapman, 1991], la acción a ejecutar vendrá indexada por el esquema de conducta activo en ese momento y por la situación del entorno percibida por el robot. En la propuesta que presentamos en este trabajo formulamos el problema del control local de un robot móvil como un problema de reconocimiento de situaciones. Asociado a cada uno de los esquemas de actuación definimos un conjunto de acciones aplicables y aprendemos las situaciones perceptuales en las que esas deben aplicarse.

Para obtener muestras de aprendizaje de un funcionamiento correcto del esquema optimizamos una función que considera la trayectoria seguida por el robot, tanto en el espacio cartesiano como en el espacio de velocidades lineales y angulares, y que premia trayectorias consistentes con el esquema. Por ejemplo, para un esquema seguir-pared consideramos que la distancia cartesiana a la pared debe ser pequeña y uniforme, premiando aquellas trayectorias con mayores velocidades lineales. Dado que el espacio de búsqueda es enorme, y que la función a optimizar no es susceptible

de ser diferenciada, utilizamos para su resolución la técnica de algoritmos genéticos, diseñando una codificación de las trayectorias del robot y un método de cruzamiento que han demostrado tener una alta efectividad.

A partir de las trayectorias obtenidas se generan todas las parejas de percepción y acción que el robot ha ido encontrando en la misma, y se agrupan en situaciones percibidas para un mismo tipo de acción.

La dimensionalidad del espacio perceptual debe ser alta para poder establecer diferencias entre las pautas perceptuales asociadas a distintos esquemas de conducta. Por ello utilizamos como entrada perceptual el campo denso de profundidad existente frente al robot móvil. Aunque hemos desarrollado el trabajo sobre un simulador (ver figura 1), existen técnicas que permiten obtener este campo de profundidad en tiempo real mediante técnicas actuales de visión artificial (ver [Kanade *et al.*, 1995] como ejemplo de utilización de visión estéreo).

Una forma de caracterizar estas situaciones es utilizar una técnica estadística estándar como es el Análisis de Componentes Principales ([Fukunaga, 1990]) para reducir la dimensionalidad de las muestras de aprendizaje correspondientes a cada situación. Veremos que el análisis de componentes principales permite reducir los mapas densos de profundidad a unos pocos parámetros en los que se mantienen la identidad propia de cada situación y que pueden utilizarse de forma efectiva para el reconocimiento.

## 2 Generación de comportamientos

Los algoritmos genéticos son métodos de búsqueda y optimización inspirados en mecanismos de tipo biológico ([Goldberg, 1989, Back *et al.*, 1997]). Se han utilizado con éxito en contextos de aplicación muy diferentes (para ver un resumen de aplicaciones, consultar [Back *et al.*, 1997]). En concreto, se han realizado diferentes trabajos en el contexto de la generación de trayectorias para robots móviles ([Xiao *et al.*, 1997, Doyle, 1995, Rendas and W.Tetenoire, 1997, Ahuactzin *et al.*, 1992]). No obstante, a diferencia del presente caso, la mayor parte de los enfoques que se han adoptado hasta la fecha realizan una búsqueda de trayectorias considerándolas simplemente como secuencias de coordenadas espaciales sin tener en cuenta la dinámica del control del robot.

Definiremos primero el modelo dinámico del robot y después entraremos en la descripción de las características de los algoritmos genéticos aplicados a esta problemática (representación de las soluciones, función de evaluación y operadores genéticos).

### 2.1 Modelo dinámico del robot

Esta sección describe las ecuaciones de movimiento fundamentales de un robot *synchro-drive*, definiendo la componente dinámica del control del mismo. El aprendizaje de comportamientos óptimos y las funciones de recompensa asociadas a los mismos se basan en estas ecuaciones.

Las variables de estado del robot móvil son su posición ( $x, y$ ), su orientación ( $\theta$ ) y sus velocidades angular y lineal ( $\omega$  y  $v$ , respectivamente).

Consideraremos constantes las aceleraciones angulares y lineales ( $a_\omega$  y  $a_v$ ). Estas aceleraciones tienden a no ser demasiado altas en implementaciones reales de robots móviles, para no someter a la estructura mecánica del robot a tensiones excesivas y para conseguir movimientos suaves. Debido a ello estas aceleraciones no deben ser despreciadas si se quieren conseguir resultados aplicables a la realidad.

El control del robot lo modelamos con las variables  $c_v$  y  $c_\omega$  que toman valores discretos  $\{-1, 0, 1\}$  y definen si, para un instante de tiempo determinado, las velocidades deben ser decrementadas, no modificadas o incrementadas. El incremento de estas velocidades vendrá dado por la constante de aceleración.

$$\omega(t_n) = \omega(t_0) + \int_{t_0}^{t_n} a_\omega c_\omega(t) dt \quad (1)$$

$$v(t_n) = v(t_0) + \int_{t_0}^{t_n} a_v c_v(t) dt \quad (2)$$

La dinámica de las variables de posición y dirección angular del robot se modela con las siguientes ecuaciones.

$$\theta(t_n) = \theta(t_0) + \int_{t_0}^{t_n} \omega(t) dt \quad (3)$$

$$x(t_n) = x(t_0) + \int_{t_0}^{t_n} v(t) \cos \theta(t) dt \quad (4)$$

$$y(t_n) = y(t_0) + \int_{t_0}^{t_n} v(t) \sin \theta(t) dt \quad (5)$$

Las ecuaciones anteriores pueden simplificarse si se asume que el robot debe controlarse de forma discreta, con intervalos de control de tiempo  $\Delta t$ . Suponemos, pues, constantes las variables de control  $c_v$  y  $c_\omega$  para estos intervalos de tiempo. De esta forma se simplifican las anteriores ecuaciones.

$$x(t_n) = x(t_0) + \sum_{i=0}^{n-1} \int_{t_i}^{t_i+\Delta t} (v(t_i) + \Delta_{t_i}^t v) \cos(\theta(t_i) + \Delta_{t_i}^t \theta) dt \quad (6)$$

donde

$$\Delta_{t_i}^t v = a_v c_v(t) (t - t_i) \quad (7)$$

$$\Delta_{t_i}^t \omega = \omega(t_i) c_\omega(t_i) (t - t_i) + \frac{1}{2} a_\omega (t_i) c_\omega(t_i) (t - t_i)^2 \quad (8)$$

## 2.2 Generación de trayectorias mediante algoritmos genéticos

En el problema de la generación de trayectorias óptimas, la elección del uso de algoritmos genéticos se fundamenta en dos razones principales: en primer lugar, es una técnica adecuada para realizar búsquedas en espacios

de dimensión elevada, como en este caso. Por otro lado, el método impone pocas restricciones de tipo matemático en la forma de la función a optimizar, de tal manera que es aplicable a la generación de trayectorias para cualquier tipo de comportamiento (evitar obstáculos, seguir paredes, etc.).

Los algoritmos genéticos parten de una *población* inicial que se suele generar de manera aleatoria y que representa un conjunto de posibles soluciones al problema planteado. En nuestro caso cada *individuo* de la población representará una posible trayectoria del robot en el espacio de velocidades.

En las sucesivas generaciones, los individuos de la población son modificados mediante la aplicación de una serie de *operadores genéticos*. El operador de *cruce* combina dos individuos con el fin de obtener una solución mejor, mientras que la *mutación* modifica aleatoriamente algunas de las características de un individuo determinado. Los individuos resultantes de los cruces y las mutaciones, junto con los ya existentes, pasan a competir por formar parte de la nueva generación, de manera que los individuos que representan soluciones mejores tienen más probabilidades de sobrevivir.

### 2.2.1 Representación de las soluciones

Para aplicar algoritmos genéticos es necesario representar cada trayectoria que puede seguir el robot mediante una cadena de símbolos (en la terminología habitual en algoritmos genéticos, un  *cromosoma*). Como se refiere en la sección 2.1, el robot se controla mediante comandos  $c_v$  y  $c_w$  que controlan si se debe incrementar, decrementar o mantener estable la velocidad lineal y angular, respectivamente. De este modo una trayectoria se representará mediante una secuencia de comandos de velocidad aplicados en instantes de tiempo discretos  $t_i, i = 1, \dots, n$ . Así, una trayectoria se representa como un vector de pares de comandos:

$$\langle (c_{v_{t_1}}, c_{w_{t_1}}), (c_{v_{t_2}}, c_{w_{t_2}}), \dots, (c_{v_{t_n}}, c_{w_{t_n}}) \rangle$$

donde los  $c_v$  y  $c_w$  pueden tomar los valores  $\{-1, 0, 1\}$  (indicando decremento, no variación e incremento de velocidad, respectivamente).

### 2.2.2 Función de evaluación

Es necesario disponer de una *función de evaluación*  $f_i$  que permita medir la idoneidad de cada individuo (de cada trayectoria). La forma de la función de evaluación variará según cuál sea el esquema de comportamiento deseado. Para la tarea de evitación de obstáculos la función de evaluación se ha definido como:

$$f_i = \begin{cases} d_i & \text{si el robot no choca con ningun obstaculo} \\ +\infty & \text{en caso contrario} \end{cases}$$

Donde  $d_i$  mide la distancia del punto final de la trayectoria seguida por el robot al límite superior del entorno. En la práctica, a las trayectorias que chocan con algún obstáculo se les asigna un valor de  $f_i$  muy grande en comparación con las demás. De este modo, el problema de conseguir una trayectoria óptima para la tarea de evitación de obstáculos se convierte en el de encontrar aquella trayectoria cuyo valor de  $f_i$  sea mínimo.

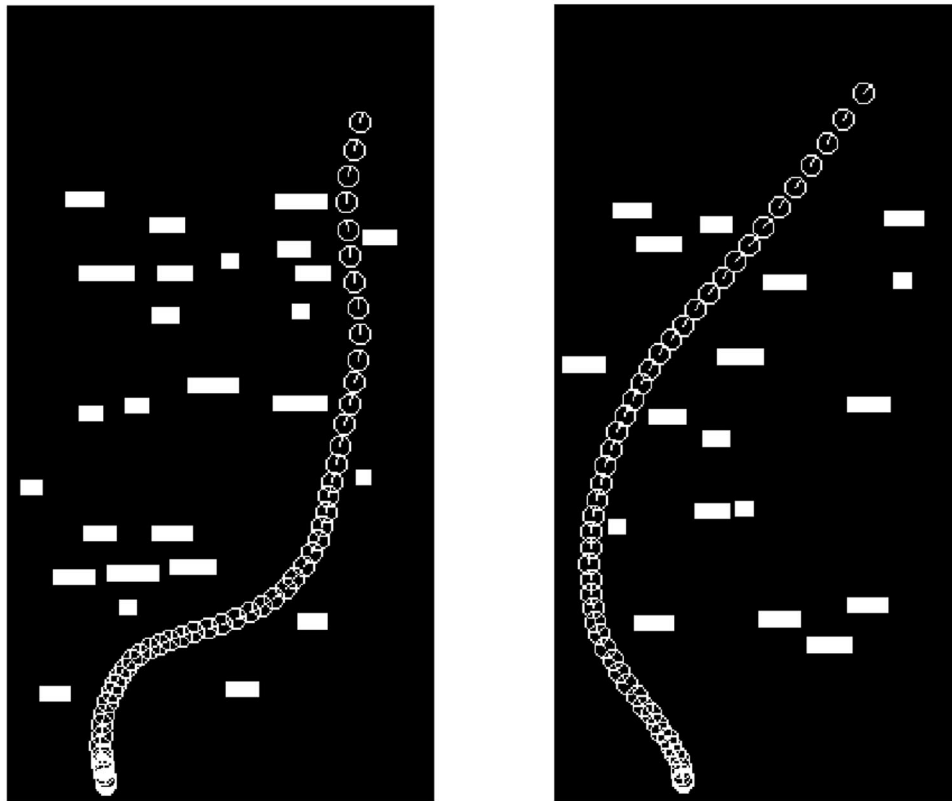


Figura 2: Resultado para distintos entornos aleatorios

### 2.2.3 Operadores genéticos

- **Selección:** Cada individuo tiene una probabilidad de ser seleccionado para reproducirse y pasar a la generación siguiente que será proporcional a su adecuación.
- **Cruce:** este operador combina los comandos de dos trayectorias para obtener una trayectoria mejor. Para ello se calcula de manera aleatoria un punto de cruce para cada una de los cromosomas que representan a las dos trayectorias y se intercambian las partes que quedan a la derecha de los respectivos puntos de cruce. Como el punto de cruce será en general distinto para cada uno de los "padres" este método da lugar a individuos de longitud variable. Esto permite que la longitud de la trayectoria vaya creciendo en las sucesivas generaciones y el robot avance evitando los obstáculos.
- **Mutación:** la mutación consiste en variar aleatoriamente el valor de un comando  $c_v$  o  $c_w$  con el objeto de modificar una trayectoria de la población actual. La modificación de un comando de velocidad lineal tiene en general poco efecto sobre la forma global de la trayectoria, mientras que la modificación de un comando de velocidad angular produce perturbaciones mayores. Ambos cambios van destinados a permitir que el robot cambie su trayectoria para esquivar los obstáculos

## 3 Aprendizaje y clasificación de situaciones

Una vez se han generado un conjunto de trayectorias correctas para el esquema que se está aprendiendo, se trata de caracterizar las situaciones en las que el robot se va a encontrar cuando evolucione siguiendo ese esquema.

Modelamos el estado en el que puede encontrarse un robot móvil evolucionando por un entorno mediante: 1) un mapa de denso de profundidad observado por el robot  $d(\theta)$  y 2) las velocidades lineales y angulares del mismo ( $v$  y  $\omega$ ). Definimos un mapa denso de profundidad como el vector  $d(\theta)$ , que nos indica la distancia a la que se encuentra el obstáculo más cercano en la orientación  $\theta$  (considerando como 0 la orientación frontal). La obtención de estos mapas de profundidad es inmediata en el simulador. Los límites del ángulo  $\theta$  vienen dados por las características de la cámara. Para el presente artículo hemos variado  $\theta$  entre -45 y 45 grados.

A partir de las trayectorias generadas por algoritmo genéticos se generan los estados que ha encontrado el robot en su evolución siguiendo dichas trayectorias. Estos estados se agrupan en conjuntos de situaciones prototipo que son aprendidos y reconocidos de la forma que se explica a continuación. Un ejemplo de campos de profundidad asociado a una situación determinada (en concreto, velocidad lineal del robot entre 60 y 70 cm/s) se presenta en la figura 3. Cada fila de la figura corresponde al mapa de profundidad en un instante de tiempo.

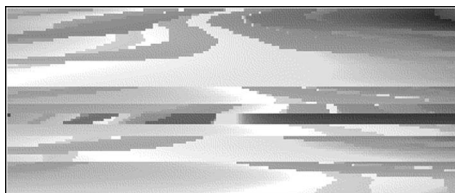


Figura 3: Ejemplo del conjunto de muestras de entrenamiento de la situación correspondiente a  $v_{60}$  ( $v \in [60cm/s, 70cm/s]$ ). Cada fila de la figura corresponde al mapa de profundidad en un instante de tiempo.

### 3.1 Análisis de Componentes Principales

El análisis de componentes principales ha sido utilizado con éxito recientemente en la comunidad de visión artificial para representar imágenes de caras humanas [Sirovich and Kirby, 1997] y para reconocer imágenes de caras [Turk and Pentland, 1991]. Con esta técnica se calculan los autovectores del conjunto de muestras de alta dimensionalidad y son usados como base ortogonal para representar cada una de las muestras individuales. Estos autovectores constituyen la dimensión de un subespacio de muestras, denominado el *autoespacio*, en el que las muestras se pueden representar de forma compacta. Utilizaremos este enfoque aplicándolo al conjunto de mapas de profundidad de cada una de las situaciones.

Representamos, pues, esta distribución de mapas de profundidad mediante una función de perturbación alrededor de un mapa de profundidad medio correspondiente a la situación:

$$d(\theta) = d_0(\theta) + \rho(\theta) \quad (9)$$

donde  $\rho(\theta)$  son pequeñas fluctuaciones  $\rho/d_0 < 1$  que capturan la identidad de cada una de las muestras. El mapa de profundidad medio se calcula a partir de las muestras de entrenamiento correspondientes a la situación

$$d_0(\theta) = \frac{1}{N} \sum_{i=1}^n d_i(\theta) \quad (10)$$

Para representar  $\rho(\theta)$  de cada situación realizamos un análisis de las componentes principales y se expanden las fluctuaciones en términos de un conjunto de autovectores  $\Psi_l$  extraídos a partir de cada distribución siguiendo el procedimiento estándar [Fukunaga, 1990].

$$d(\theta) = d_0(\theta) + \sum_l a_l \Psi_l(\theta) \quad (11)$$

De esta forma, un mapa denso de profundidad correspondiente a una situación determinada pasa a representarse como una combinación lineal de los modos de variación más importantes ( $\Psi_l$ ) sumado al mapa de profundidad medio. Por ello, considerando el nuevo espacio paramétrico definido por los modos de variación principales, un mapa de profundidad pasa a representarse por el vector  $\mathbf{b} = (b_1, \dots, b_l)$  correspondiente a las coordenadas en el nuevo espacio paramétrico, reduciéndose considerablemente

la dimensionalidad de la distribución. Los autovalores  $\lambda_l$  asociados a cada uno de los autovectores representan la varianza de cada uno de los modos de variación principales.

### 3.2 Clasificación

La proyección  $b$  de un mapa de profundidad  $d$  en un autoespacio  $P_l$  se obtiene mediante la ecuación

$$b = P_l^T (d - d_0), \quad (12)$$

siendo  $P_l = [\Psi_1, \dots, \Psi_l]$  la matriz con los  $l$  primeros autovectores.

Una métrica muy usada para cuantificar la pertenencia de una muestra a una distribución es la distancia de Mahalanobis, que mide la distancia de la muestra al centro de una distribución, ponderada por la varianza en cada uno de las dimensiones del espacio:  $D_{\text{Mah}}$ :

$$D_{\text{Mah}}(b, \lambda) = \sum_{k=1}^l \left( \frac{b_k^2}{\lambda_k} \right) \quad (13)$$

El criterio usado para medir a que situación pertenece un mapa de profundidad percibido es escoger aquella situación cuyo autoespacio minimiza la distancia de Mahalanobis con el mapa de profundidad percibido:

$$\text{situación actual} = \min_{i=0}^t D_{\text{Mah}}(b_i, \lambda_i), \quad (14)$$

siendo  $t$  el numero de situaciones aprendidas,  $b_i$  la proyección del mapa de profundidad actual en el autoespacio correspondiente a la situación  $i$  y  $\lambda_i$  los autovalores de la situación  $i$ .

## 4 Resultados

Se ha realizado una implementación del esquema avanzar-evitando-obstáculos siguiendo la propuesta del trabajo. Para ello se han generado comportamientos correspondientes a ese esquema en distintos entornos aleatorios.

A partir de estas trayectorias hemos considerado 8 situaciones correspondientes a velocidades lineales ( $v_{10}, v_{20}, \dots, v_{80}$ ), con  $v_i$  agrupando las velocidades lineales en el rango ( $i$  cm/s,  $i + 10$  cm/s), y 3 situaciones correspondientes a velocidades angulares ( $\omega_{-10}, \omega_0, \omega_{10}$ ), con  $\omega_i$  agrupando las velocidades angulares en el rango ( $i - 5$  grados/s,  $i + 5$  grados/s). Se han obtenido muestras de los mapas de profundidad percibidos por el robot en cada una de estas situaciones (en la figura 3 se puede observar un ejemplo de los mapas de profundidad asociados las situaciones  $v_{50}, v_{60}$  y  $v_{70}$ ).

Se ha realizado un análisis de componentes principales de las muestras correspondientes a cada una de las situaciones, obteniéndose el autoespacio asociado a cada una de ellas. En la figura 5 se muestran los mapas de profundidad medios correspondientes a las situaciones  $v_{10}, \dots, v_{80}$ . En ellos se representa la distancia media (en centímetros) a la que se encuentran obstáculos (desde -45 grados hasta 45 grados en dirección frontal) cuando



Figura 4: Ejemplo del conjunto de muestras de entrenamiento de las situaciones correspondiente a  $v_{50}$ ,  $v_{60}$  y  $v_{70}$ .

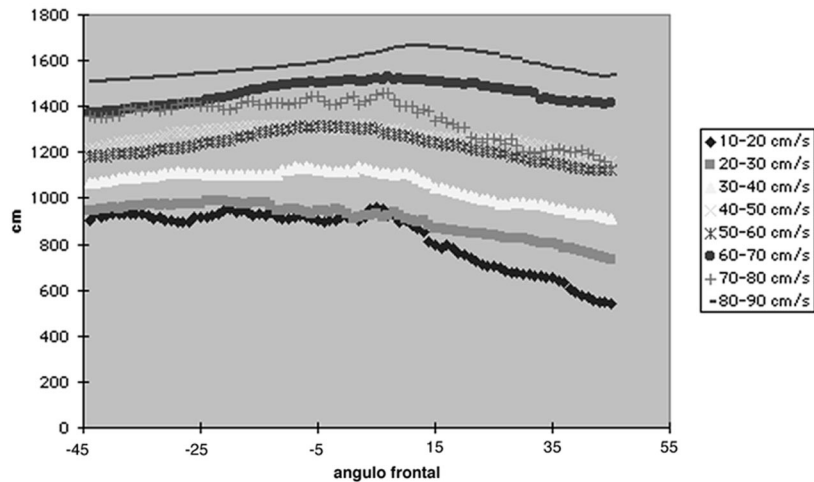


Figura 5: Mapas de profundidad medios correspondientes a cada una de las distintas situaciones de velocidad lineal.

el robot se movía a la velocidad correspondiente a cada una de las situaciones. Se puede ver que es coherente con lo esperado: a velocidades más altas el robot se encuentra los obstáculos a mayor distancia.

Por último, la figura 6 muestra un ejemplo de evolución del robot utilizando el reconocimiento de situaciones propuesto anteriormente. Cada 0,2 segundos se realiza una lectura del mapa de profundidad del entorno, se proyecta esa lectura sobre los autoespacios correspondientes a cada una de las situaciones aprendidas y se obtiene la velocidad lineal y angular a la que debería estar moviéndose el móvil (aquellas con las que se minimiza su distancia de Mahalanobis), modificándose las velocidades actuales consecuentemente.

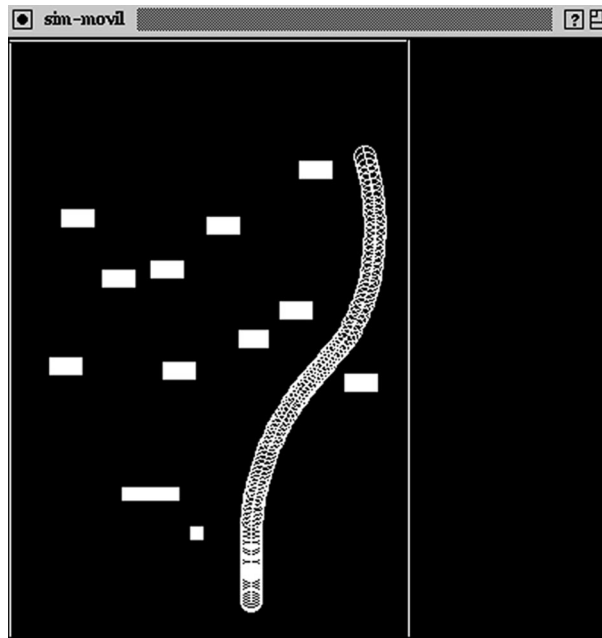


Figura 6: Ejemplo de trayectoria seguida por el robot aplicando el algoritmo de control basado en reconocimiento estadístico de situaciones.

## 5 Conclusiones

Se ha presentado un enfoque con el que es posible aprender automáticamente esquemas locales de conducta que guían la navegación de un robot móvil, aplicándose al ejemplo concreto de avanzar evitando obstáculos. El método se puede resumir en: 1) generación off-line de trayectorias consistentes con el esquema de navegación, 2) aprendizaje de las situaciones perceptuales (mapas densos de profundidad) que el robot se ha encontrado cuando estaba evolucionando en el entorno, agrupados por situaciones discretas (robot moviéndose entre 10 y 20 cm/s, robot moviéndose entre 20 y 30 cm/s, etc.) y 3) control del robot basado en el reconocimiento de situaciones.

El primer aspecto se ha llevado a cabo utilizando técnicas de algoritmos genéticos, el segundo con un análisis de componentes principales y el tercero utilizando las distancias a las distribuciones aprendidas.

Como trabajo futuro, estamos comenzando a caracterizar otros esquemas utilizando estas técnicas (como *seguir-pared* o *entrar-en-puerta*) al tiempo que pretendemos comprobar la validez del planteamiento en un robot real.

## Referencias

[Ahuactzin *et al.*, 1992] J.M. Ahuactzin, E.G. Talbi, P. Bessiere, and E. Mazer. Using genetic algorithms for robot motion planning. In *European Confer-*

*ence on Artificial Intelligence (ECAI92, 1992.*

- [Arkin, 1987] R.C. Arkin. Motor schema - based mobile robot navigation. *The International Journal of Robotics Research*, 8(4):92-112, 1987.
- [Arkin, 1990] R.C. Arkin. Integrating behavioral, perceptual, and world knowledge in reactive navigation. *Robotics and Autonomous Systems*, 6:105-122, 1990.
- [Back *et al.*, 1997] T. Back, D. Fogel, and Z. Michalewicz (Eds.). *Handbook of Evolutionary Computation*. Oxford University Press, 1997.
- [Borenstein and Korem, 1991] J. Borenstein and Y. Korem. The vector field histogram - fast obstacle avoidance for mobile robots. *IEEE Transactions on Robotics and Automation*, 7(3):278-288, 1991.
- [Chapman, 1991] D. Chapman. *Vision, instruction and action*. MIT Press, 1991.
- [D.Fox *et al.*, 1997] D.Fox, W. Burgard, and S. Thrun. The dynamic window approach to collision avoidance. *IEEE Robotics and Automation Magazine*, march:23-33, 1997.
- [Doyle, 1995] A. Doyle. *Algorithms and computational techniques for robot path planning*. PhD thesis, School of Electronic Engineering and Computer Systems, University of Wales, 1995.
- [Fukunaga, 1990] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, 1990.
- [Goldberg, 1989] David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, 1989.
- [Kanade *et al.*, 1995] T. Kanade, H. Kano, S. Kimura, A. Yoshida, and K. Oda. Development of a video-rate stereo machine. In *Proc. of International Robotics and Systems Conference (IROS-95)*, 1995.
- [Latombe, 1991] J. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, 1991.
- [Rendas and W.Tetenoire, 1997] M.J. Rendas and W.Tetenoire. Definition of exploratory trajectories in robotics using genetic algorithms. In *Tenth International Conference in Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*, 1997.
- [Simmons, 1996] R. Simmons. The curvature-velocity method for local obstacle avoidance. In *Proc. IEEE Int. Conf. Robotics and Automation*, 1996.
- [Sirovich and Kirby, 1997] L. Sirovich and M. Kirby. Low dimensional procedure for the characterization of human faces. *Journal of Optical Society of America*, 4:519-524, 1997.
- [Turk and Pentland, 1991] M. A. Turk and A. P. Pentland. Face recognition using eigenfaces. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, 1991.

[Xiao *et al.*, 1997] J. Xiao, Z. Michalewicz, L. Zhang, and K. Trojanowski. Adaptive evolutionary planner/navigator for robots. *IEEE Trans. on Evolutionary Computation*, 1, 1997.