Aprendizaje de Sistemas Multiagentes por medio de Aprendizaje por Refuerzo

Ing. Edgar Alirio Aguirre Buenaventura, MSc. Giovanni Rodrigo Bermúdez Bohórquez

***Abstract*—La robótica móvil en espacios discontinuos presenta una fuente inagotable de retos para su desarrollo. Es por eso que en éste artículo se presenta el desarrollo de un método de aprendizaje básico implementado a partir de aprendizaje por refuerzo que permite obtener un sistema de toma de decisiones con un bajo costo computacional. Para el desarrollo del presente trabajo se estudiaron dos métodos, el primero basado en el procesos de decisión de Markov (MDP); el segundo basado en Q learning. Estos métodos fueron implementados en agentes sencillos con los cuales se obtuvo un sistema multiagente para analizar sus respuestas en un espacio no supervisado.**

***Index Terms*—**Aprendizaje por refuerzo, procesos de decisión de Markov, Q Learning, Agente, Sistema Multiagente, Robótica móvil**.**

I. Introduction

El primer enfoque del aprendizaje por refuerzo dentro de la robótica fue desarrollado en 1979 por Klopf [1] en donde son discutidos los sistemas adaptables, se expone el proceso de adaptar un comportamiento para maximizar una señal (en especial en un ambiente). Esta fue la primera idea básica para desarrollar el aprendizaje por refuerzo, pero fue olvidado o poco trabajado el concepto de conseguir algo del ambiente como método de aprendizaje.

Durante el transcurso del tiempo ha recobrado importancia y ahora participa en espacios como la psicología, teoría de control, la inteligencia artificial, y neurociencia en donde son desarrollaron diversos métodos como los MDP o Procesos de Decisión de Markov y Q learning, además de otros métodos. Este trabajo desarrolla estos dos métodos con sus respectivos algoritmos de robótica probabilística para compararlos y utilizarlos en diferentes ejercicios en ambientes no supervisados, como la exploración de aéreas con obstáculos, formaciones robóticas y localización de objetos.

II. Aprendizaje por refuerzo

El método de aprendizaje por refuerzo traza un mapa de las situaciones donde se pueden ejecutar movimientos para así maximizar la señal de recompensa numérica de todas estas posibilidades. El método no refleja directamente que acción va a realizar, él descubre que acción produce una mayor recompensa probando estas acciones, afectando la recompensa inmediata y la recompensa de la próxima situación, traduciéndose este procedimiento en una búsqueda de recompensas por medio de ensayo y error.

El agente debe poder intuir como se encuentra el ambiente y debe poder tomar las acciones que afecten ese estado. Por tanto, el agente debe percibir la sensación, el movimiento y el objetivo en sus formas más simples sin contradecirse entre ellos. Esto lo diferencia de un aprendizaje supervisado en donde el proceso es realizado por un grupo de ejemplos y sus acciones son premeditadas.

Los elementos que caracterizan el aprendizaje por refuerzo son una política, una función de recompensa y una función de valor. Algunas variaciones del aprendizaje por refuerzo incluyen un modelo del ambiente. Dentro de este tipo de aprendizaje la política define la manera de actuar de un agente a través de un grupo de reglas de estimulo las cuales mapean los estados y las acciones.

Agente es un sistema computacional que habita en un entorno complejo y dinámico con la capacidad de percibir y actuar autónomamente sobre dicho entorno siendo capaz de cumplir con un conjunto de objetivos o llevar a cabo ciertas tareas para las cuales fue diseñado [2]. El *entorno* es todo aquello que no es el agente y que es de interés para llevar a cabo la tarea que se le ha asignado a dicho agente, por lo tanto dentro del entorno esta el ambiente, los obstáculos y otros agentes que se encuentren en él mismo, la *recompensa* es un valor escalar que indica lo deseable que es una situación para un agente [2][3] indicando un posible plano de navegación del agente. Define el objetivo en un problema de aprendizaje por refuerzo, define que tan conveniente es realizar una acción en un estado. Los *estados* son los posibles espacios físicos donde se puede mover el agente indicando la probabilidad de ejecución de una tarea dentro de un entorno. Las *acciones* son los posibles movimientos o comportamientos que puede ejecutar el agente durante la navegación en los estados de un entorno.

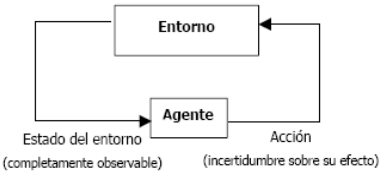

Fig. 1. Modelo de interacción Agente – Entorno [2][3]

En el aprendizaje por refuerzo se tienen dos estados básicos, los supervisados y los no supervisados. Los estados supervisados conocen en todo momento el entorno y por tanto se puede realizar un control óptimo de estados continuos para poder planear todas las posibles acciones que maximice las recompensas. En este caso, el proceso de observación se encarga de generar un modelo del ambiente para poder mapear

todos los estados. Los estados no supervisados desconocen gran parte del ambiente y por tanto, los estados conocidos son pocos y el mapeo es realizado sobre la historia inmediata pasada proyectando la mejor política para la ejecución de acciones.

Al describir el modelo de interacción agente-entorno se toma la visión desde el agente el cual debe poder describir el estado en que se encuentra. En muchos casos se realiza un modelo de observación donde se calcula la probabilidad de ver los estados inmediatos y próximos permitiendo generar las recompensas y generar un mapa de estados y por ende el agente percibe el entorno.

La función de valor determina la utilidad de cada estado [2][3]. Esta función realmente determina para estados supervisados la probabilidad de los estados pasados y para estados no supervisados solo se tiene el estado anterior. Por otro lado, ayuda a determinar el contexto de horizonte finito y horizonte infinito de la solución.

III. PROCESOS DE DECISIÓN DE MARKOV (MDP)

Se toma como modelo básico el mostrado en la figura 1 en donde se asume que el agente percibe el entorno y genera como salida un acción, la cual afecta el nuevo entorno desenvolviéndoos en el ambiente. Aquí se puede determinar que el error generado siempre es observable. Un MDP es un modelo matemático de un problema de decisión secuencial en tiempo discreto definido por los siguientes elementos [4]

S es el conjunto finito de estados (s ε **S**) que puede tomar el entorno.

A es el conjunto finito de acciones (a ε **A**) que puede ejecutar el agente.

T es la función de transición de estados, que para cada estado final s', estado inicial s y acción a, determina la probabilidad p(s'|s, a) (probabilidad de que ejecutando la acción a en el estado s, el entorno pase al estado s'). Por lo tanto, mapea los elementos de **S**x**A** en una distribución de probabilidad discreta sobre **S**.

R es la función de recompensa, que para cada estado s y acción a, determina la recompensa obtenida por el agente al ejecutar la acción a en dicho estado s, r(s, a).

En un MDP que a lo largo de su ejecución va pasando por los estados s0, s1, ..., st-1, st, y realizando las acciones a0, a1, ..., at-1, at, la probabilidad de que el estado en el instante t+1 sea st+1 viene dada por:

$$p(s\_t1 \mid s0 , a0 , s1 , a1 ,..., s\ t , at)\ p(s\_t1 \mid s\ t , at)$$

Esta condición se conoce como propiedad de Markov y significa que el estado y la acción actual proporcionan toda la información necesaria para predecir el próximo estado. En definitiva, toda la historia pasada del sistema queda resumida en su estado actual y el futuro sólo depende de dicho estado y de la acción realizada en él [2][3].

Un MDP es un proceso secuencial de toma de decisiones, y en este sentido es posible trabajar en dos contextos bien distintos. En el primero de ellos, conocido como de horizonte-finito, el agente sólo actúa durante un número finito y conocido de pasos k. En otras ocasiones, el agente actúa durante un número infinito o indefinido de pasos, recurriéndose en estos casos al modelo de horizonte-infinito.

Como el objetivo también es maximizar la recompensa obtenida a largo plazo, utilizándose generalmente para ello un modelo con factor de descuento $0< \gamma <1$, en el que el agente debe maximizar, el factor de descuento permite tener en cuenta las recompensas futuras asegurando que la sumatoria tienda a un valor finito, plazo. Cuanto mayor sea el factor de descuento (más próximo a 1) [2][3], mayor es el peso que tienen las recompensas futuras sobre la decisión actual.

Una política es una función que asigna acciones a los estados**, $\pi \rightarrow$ a,** por lo tanto se pueden tener múltiples políticas, pero una política será mejor cuanto mayor sea la recompensa que obtiene a largo plazo.

La función de valor V $\boldsymbol{\pi}$ (s) determina la utilidad de cada estado *s* suponiendo que las acciones se escogen según la política $\boldsymbol{\pi}$ asigna a cada estado un valor numérico que representa la utilidad de dicho estado a largo plazo, diferente a la función de recompensa que asigna para cada una de las acciones que pueden ejecutarse en cada estado, un valor numérico que representa la utilidad inmediata de dicha acción. La función para horizontes finitos es:

$$V\pi, t(s) = r(s\pi_t(s)) + \gamma \sum_{s' \in S} p(s' \mid s, \pi_t(s)) * V\pi.t - 1(s')$$

La función para horizontes infinitos es:

$$V\pi, (s) = r(s\pi_t(s)) + \gamma \sum_{s' \in S} p(s' \mid s, \pi_t(s)) * V\pi(s')$$

En el contexto de horizonte-infinito con factor de descuento, la función de valor de cada estado sólo depende de la política y no del número de pasos futuros (que es siempre infinito) [2][3], En este caso, planteando la ecuación anterior para cada uno de los estados que componen el MDP, se obtiene un sistema lineal de ecuaciones, siendo la función de valor Vp la solución del mismo.

Para resolver un MDP se debe encontrar la política óptima la cual es la que maximiza la función de valor. La función de valor se soluciona con la siguiente ecuación donde se realiza esta función para cada uno de los estados observables, y mapea los estados contra las acciones (SXA), teniendo una función de descuento γ la cual nos permite en momentos futuros cambiar la resolución del nivel de aprendizaje, además se tiene una comparación con la función de valor pasada la cual indica la ganancia de la acción.

$$Qt(s,a) = r(s,a) + \gamma \sum_{s' \in S} p(s' \mid s, a) * Vt - 1(s')$$

La función escogida es la máxima función de valor

$$Vt(s) = \max aQt(s,a)$$

Al tener la máxima función de valor para el estado actual se resta contra la función de valor pasada, el resultado se compara contra un factor ε que nos da mayor resolución y se asegura que la función de valor se aproxima a un valor optimo.

$$|Vt(s) - V_{t-1}(s)| < \in para\_todo - s \in S$$

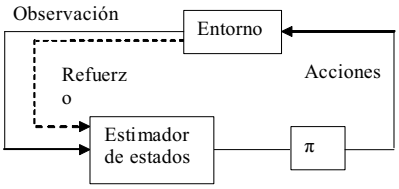

Fig. 2. Modelo de Agente - Entorno MDP

IV. Q-LEARNING

La idea en Q-Learning [5] es escoger una *función-Q*, la cual es una predicción del retorno asociado, con cada acción *a,* Esta predicción puede ser actualizada con respecto a la predicción de los retornos en el próximo estado visitado, Se puede definir la función de evaluación $Q(s,a)$ como el máximo refuerzo descontado acumulado que se obtiene al empezar en un estado inicial s y aplicando la acción a como primera acción, se puede alcanzar una política óptima sin necesidad de tener conocimiento sobre las funciones de transición de estado ni de refuerzo, solamente con el conocimiento de las acciones, en el MDP se juega a encontrar una política optima en Q-Lerninig se obtiene una tabla de estados contra acciones donde con cada acción se actualiza esa tabla generando una recompensa o castigo para ciertos estados ya visitados, por lo tanto a medida que las acciones se ejecutan se conoce el resultado de ciertas acciones en el entorno y se puede escoger cursos de acción diferentes para el agente. En el proceso del algoritmo de $Q(s,a)$ se debe actualizar la evaluación en las entradas de la tabla, $Q(s,a)$ con la ecuación:

$$Q(s,a) \leftarrow r + \gamma \max_a Q(s',a')$$

Una de las principales características de este algoritmo, en lo que a la convergencia con una política óptima se refiere [6], es que no requiere que el agente en su entrenamiento ejecute las secuencias óptimas para converger a dicha política óptima. De hecho, puede aprender la función Q (y por tanto una política óptima) ejecutando en cada paso acciones elegidas de forma aleatoria, siempre que los pares estado-acción sean visitados infinitas veces.

Para entornos no determinados, la ecuación 6 no es apropiada, ya que no tiene en cuenta las probabilidades en las transiciones de estado. La siguiente ecuación resuelve este problema para entornos con transiciones de estado no deterministas en los que la ejecución de una misma acción desde un mismo estado puede llevar al agente a estados distintos, y por tanto recibir recompensas distintas.

$$Q_n(s,a) \leftarrow (1-\alpha_n)Q_{n-1}(s,a) + \alpha_n\{r + \gamma \max_a Q_{n-1}(s,a)\}$$

En este caso, el parámetro ∞ hace referencia a las probabilidades involucradas, y es calculada en cada momento mediante la siguiente ecuación.

$$\alpha_n \leftarrow \frac{1}{1 + visitas_n(s,a)}$$

V. DESARROLLO DEL PROYECTO

El trabajo se baso en desarrollar algoritmos de MDP y Q-Learning y proponer ejercicios con agentes. En primera instancia se desarrollo los procesos de decisión de Markov MDP, tomando como base un conjunto de 25 estados y 9 estados observables.

S_1=0	S_2=0	S_3=0	S_4=0	S_5=0
S_6=0	S_7=1	S_8=0	S_9=1	S_{10}=0
S_{11}=0	S_{12}=0	S_{13}=0	S_{14}=0	S_{15}=0
S_{16}=1	S_{17}=0	S_{18}=0	S_{19}=0	S_{20}=1
S_{21}=0	S_{22}=0	S_{23}=0	S_{24}=0	S_{25}=0

Fig. 3. Conjunto de Estados

Los estados S_7 y S_9 se toman como obstáculos y el estado donde se encuentran los agentes son S_{16} y S_{20}. El ejercicio es realizar la exploración del conjunto de estados (entorno) con dos agentes. Lo primero que se hace es definir el conjunto de estados que en este caso son (S_0, S_1, S_2, S_3, S_4, S_5, S_6, S_7, S_8, S_9, …, S_{25}), generando las recompensas de movilidad para cada estado al ser obstáculos S_7 y S_9 se les dan un valor de 1 como estados ocupados, y para el estado de los agente S_7 y S_9 se le da un valor de 1, los demás estados tiene una probabilidad de movilidad y están vacíos se les da un valor de 0, por lo tanto en el conjunto de estados la probabilidad de movilidad va asociada a los espacios libres.

Las políticas indican un mapa de acciones ya que toda política conlleva una acción, se pueden tener muchas políticas y por lo tanto iguales o mayor numero de acciones. Por tanto al tener una política activa, la probabilidad de movimiento cambia dependiendo del conjunto de acciones que se puedan realizar en este caso de exploración la primera política dice π_1 = 0 si no hay obstáculo muévase hacia adelante, y se le da una recompensa de 0,7 y la probabilidad de movimiento cambia 0,7/25 = 0,028.

π	Descripción política	Recompensa
1	si no hay obstáculo muévase hacia adelante	0,7
2	muévase a donde haya mayor probabilidad de no encontrar obstáculo	1
3	si encuentra 2 o más probabilidades de movilidad, seleccione la primera	1
4	si llega a una frontera gire 90 grados	1
5	si encuentra un agente ubíquelo como un obstáculo y ejecute la política 2	1
6	después de 3 movimientos debidos a la política 1 gire 90 grados	1

Tabla 1. Políticas – Recompensas

La recompensa dividida entre el número de estados observables, al ser la primera política es la primer a acción que se ejecuta y el agente se mueve hacia adelante en la Figura 4 se puede observar el conjunto de políticas con su recompensa. Se escogieron seis (6) políticas para realizar el ejercicio de exploración del entorno, existiendo políticas y subpoliticas, cuando se ejecuta una política que tiene una subpolitica el comportamiento del agente podrá variar si se cumple alguna condición.

Algoritmo MDP

El algoritmo del proceso de decisión de Markov nos indica

el proceso que se realizo, se debe tener en cuenta que al iniciar el algoritmo todos los valores deben estar en cero menos la función de valor pasada que ya tiene un valor preestablecido, el algoritmo es como se describe a continuación:

Inicio
Inicializar el instante de tiempo t = 0
Inicializar la función de valor $V_O(s) = 0$, para todo s∈**S**
Inicializar el factor de descuento en γ = 0,2
En t = 0 $V_\pi(s`) = 0{,}3$ "$V_\pi(s`)$ = valor función de valor pasada"
En t > 0 = $V_\pi(s`)$
Seleccionar ε < 0

Repetir:
(a) Incrementar t
(b) **Repetir** para todo s ε **S** y para toda a ε **A** aplicando

$$Qt(s,a) = r(s,a) + \gamma \sum_{s' \in S} p(s' \mid s,a) * Vt - 1(s')$$

$$Vt(s) = \max aQt(s.a) \text{ hasta que}$$

$$|Vt(s) - V_{t-1}(s)| < \in para\_todo - s \in S$$

Si ε < 0, entonces disminuya γ = 0,1

Este algoritmo contiene la solución de la función de valor y una posterior verificación de comparación para mejorar la resolución de la solución del ejercicio. El mismo ejercicio se plantea para Q learnig, La exploración del entorno se realiza en 8 pasos.

Algoritmo Q learning

Para cada par $(s \in S, a \in A)$ inicializar la tabla $Q(s,a)$ a 0.
Observar el estado actual *s*.
Seleccionar una acción *a* y ejecutarla
Recibir el refuerzo inmediato *r*
Observar el nuevo estado *s*'
Actualizar la entrada de la tabla, $Q(s,a)$ con la ecuación:

$$Q(s,a) \leftarrow r + \gamma \max_a Q(s',a')\}$$

Asignar a *s* el estado *s*'
Ejecutar la acción premiada
Repetir

VI. RESULTADOS

Al realizar los ejercicios de aprendizaje por refuerzo planteados se trabajo inicialmente con un área de trabajo distribuida en 25 estados que podrían ser explorados o podrían estar ocupados por un obstáculo, durante el ejercicio se ubicaron dos obstáculos los cuales redujeron la probabilidad de movilidad en un 92% de todos los estados del entorno, resultando una exploración de 22 estados, con una probabilidad de exploración de 95,6%. El agente tiene un porcentaje de visión de los estados de 39,1% este es el conocimiento total del entorno por parte del agente. Cuando el agente llega a una frontera la cantidad de estados de movilidad se reduce, debido a que el método promedia estos estados se asigna con valor de cero a los estados que ocupan la frontera.

Se escogieron 6 políticas básicas, las cuales tienen prioridad dependiendo de su recompensa, como resultado se observo que unas políticas deben inhibir a otras para que no se vuelvan repetitivas y anulen el objetivo del agente, por este motivo la selección de las políticas y subpolíticas es importante, dentro de las políticas seleccionadas se escogió una política de ir hacia adelante y como subpolítica dice que después de 3 movimientos debidos a la política 1 (ir hacia adelante) gire 90 grados, esto porque en un mapa donde todos los estados frontales estén vacíos el comportamiento del agente será repetitivo y nunca podrá explorar los demás espacios internos del entorno, también se ubica una política de moverse a donde haya mayor probabilidad de no encontrar obstáculo, con una subpolítica que indica, si encuentra 2 o más probabilidades de movilidad, seleccione la primera, esta subpolítica se debe a que cuando se promedia la probabilidad de movilidad por estados, hay varios que se pueden ejecutar, por lo tanto se debe seleccionar con algún método le acción que se debe realizar, en este caso se selecciona la primera opción encontrada.

Dentro del método de aprendizaje por refuerzo encontramos una variable la cual es la recompensa inmediata que se genera por una política especifica, esta recompensa se genera cuando al observar los estados inmediatos identificamos si están vacíos u ocupados entones la política dicta la forma de premiar los estados, por ejemplo si todos los estados inmediatos están desocupados aplica la política numero 1 que dice si se puede mover al frente, como es afirmativo solo va a generar una recompensa a ese estado y los demás estados por así decirlo serán castigados, entonces la mejor recompensa del conjunto de estados es la que corresponde a moverse hacia adelante, al observar el ejercicio realizado, la tendencia hacia una política en la mayoría de los casos fue ir hacia adelante

Debido a esta tendencia se presenta la figura 4-a donde se observa este comportamiento en uno de los agentes, se grafico todas las acciones realizadas contra su respectiva recompensa, y las recompensas mas altas fueron las de la política 1, por lo tanto se puede obligar al agente que tome cierto tipo de desiciones por medio de las políticas para que actué o tienda a comporte de una forma especifica. Dentro de estas graficas de tendencias observamos en la figura 4-b la tendencia del segundo agente, el cual tuvo 2 tendencias claras, la primer es ir a hacia adelante la cual se repitió en la mayoría de oportunidades, y la segunda es ir hacia la derecha, este comportamiento es predecible porque el agente tiende a ir hacia adelante, pero recordemos que la política que me indica esta acción tiene una subpolítica que dicta que cada cierto tiempo que se realice la acción de ir hacia adelante gire 90 grados esto quiere decir que gire hacia la derecha.

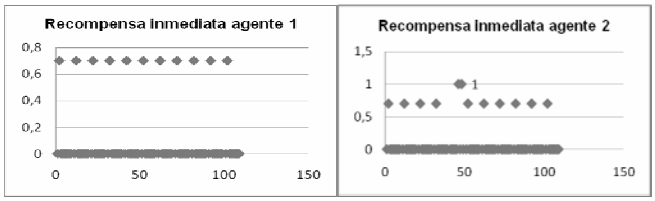

Fig. 4. a) Recompensa inmediata agente 2 que indica mejor opción adelante. b) Recompensa inmediata agente 2 que indica mejor opción adelante y la marcada con número 1 que es a la derecha.

El objetivo del agente es explorar el entorno, por lo tanto cada paso corresponde a la disminución de posibilidades de

conocer el mapa, en la figura 6 se describe la probabilidad de ocupar un estado de todo el conjunto de estados posibles de ser conocidos o visitados, esta probabilidad va decreciendo como lo muestra la figura 6.

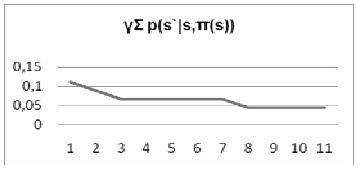

Fig. 6. probabilidades de movimiento.

El método de aprendizaje por refuerzo premia algún elemento en particular de los factores que intervienen en el proceso, bien sean la políticas o las acciones, pero consiste en premiar o castigar ciertos elementos que permiten generar decisión, en el MDP se premia la política y el resultado de operar el método se conoce como función de valor, realmente esta función dicta la acción que se desea realizar para observar el cambio en el tiempo de esta función se resta la función de valor actual con la función de valor pasada, en un inicio del ejercicio empieza con un valor alto pero se va reduciendo a medida que ejecuta mas pasos y tiende a estabilizarse en un rango como en la figura7-a, en la figura 7-b se observa el cambio de la función de valor del segundo agente el cual genera una función Q que premia las acciones o las castiga, en esta grafica la tendencia tiende a ser estable pero como la reacción es mas rápida en el método de Q learning esta genera valores altos para compensar la tendencia.

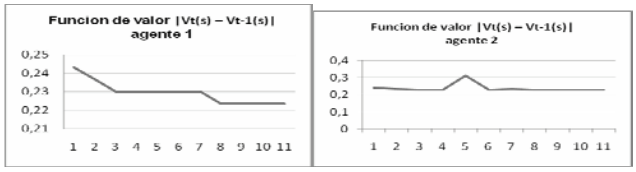

Fig. 7. a) |Vt(s) – Vt-1(s)| MDP. b) |Vt(s) – Vt-1(s)| Q learning

El factor de descuento corresponde a una variable que se selecciona para darle mayor resolución a la función de valor o a la función Q que se modifica en el ultimo paso del método, al comparar la diferencia de la función de valor o función Q pasada y la actual con un valor llamado *E* que es fijo, si la función supera este valor *E* se debe cambiar la variable *Y* de la función de valor con un dato mas bajo, así la resolución de aprendizaje tendera a ser mas detallado como se muestra en la figura 8 la tendencia es estabilizarse.

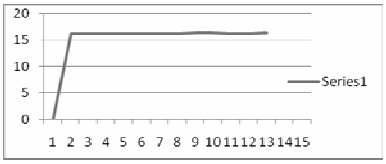

Fig. 8. |Vt(s) – Vt-1(s)| ,< E ,Q learning

a. Simulación multiagente.

Estas simulaciones se realizaron con tres agentes donde se observo el comportamiento debido a las políticas, el mapa mostrado en la figura 10-a corresponde al entorno con un grupo de obstáculos y 3 agentes los cuales tienen como misión explorar el mapa, cada agente realizo un total de 69 pasos teniendo como política de multiagente que si encuentra un agente tómelo como un obstáculo, utilizando una política reactiva, las posiciones iniciales de los agentes en el plano coordenado son, (180, 90), (190, 90), (200, 90). Los agentes deben explorar el mapa a través de MDP y cada agente tomo sus propias desiciones de movimiento dependiendo del entorno, en la figura 10-b se observa como se desarrollo esta simulación, así podemos seguir su recorrido, observamos que aunque los agentes inician desde lugares cercanos, las rutas que toman son muy distintas.

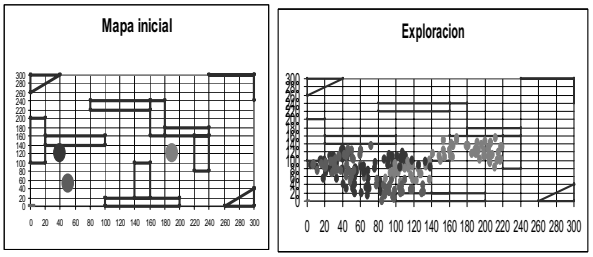

Fig. 10. a) Mapa de inicio multiagente. b) Simulación exploración multiagente

Encada paso el agente selección un ángulo, esto dependiendo de que política pudiera operar, la tendencia fue hacia 90 grados esto indica que el movimiento que mas se realizo fue hacia delante. Tomando la exploración de la figura 10-b, se muestra la ruta general de un agente en la figura 11-a, con esta ruta se genero una amplia exploración en todos los ejes del mapa pudiendo seleccionar diferentes tipos de rutas, es muy importante la exploración inicial porque nos va a permitir tomar nuevas desiciones con los resultados obtenidos.

La ruta que se muestra en la figura 12-a corresponde a un grupo de puntos seleccionados de los datos de exploración inicial que son los puntos menores al punto de inicio de la exploración, en la figura 12-b se muestran los valores mayores al valor de inicio. El desplazamiento de la ruta se muestra en la figura 12-a, la cual esta en centímetros y segundos, con estos datos también podemos sacar la velocidad promedio de desplazamiento la cual nos da un valor de 10cm/s

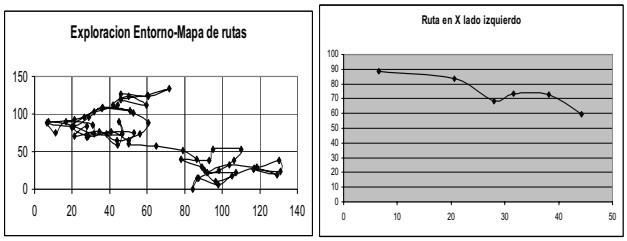

Fig. 11. a) Mapa de ruta general. b) Ruta lado izquierdo

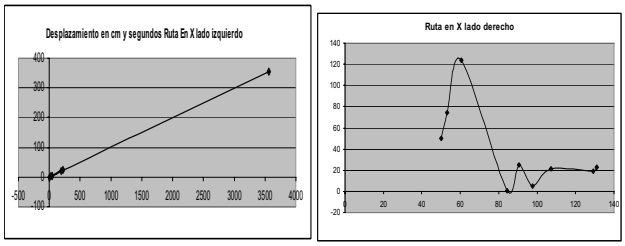

Fig. 12. a) desplazamiento. b) Ruta lado derecho.

VII. Conclusiones

Dentro de los métodos seleccionados de aprendizaje por refuerzo se concluye que estos métodos a través de las simulaciones son óptimos para la ejecución de tareas de agentes robóticos individuales y cooperativos aclarando que depende del tipo de objetivo que se desea lograr depende el método que se debe implementar, el método de Q learning es más rápido ya que en menos iteraciones realiza la tarea en cambio el método MDP realiza mas iteraciones, si se desea que la respuesta sea muy rápida se utilizara el método de Q learning, pero si lo que se busca es que el objetivo se cumpla con la mayor exactitud entonces se debe usar el MDP que permite una mayor resolución al resolver el ejercicio.

En la realización de simulación se observo la importancia del modelo de observación de estados este nos indica cuales son las mayores probabilidades de movilidad exitosa, así que se debe ampliar la zona de visión del agente para que perciba mas estados de movilidad y así el modelo de observación entregara resultados más significativos a la maximización de la tarea.

Se deben determinar las políticas necesarias, no muchas no pocas, cuando se tienen varias políticas se deben mapear todas estas, pero al ser pocas pueden presentarse incongruencias, se deben escoger las políticas teniendo encuentra la estrategia de sumisión para inhibir ciertas políticas, se deben escoger políticas que preferiblemente tengan una sola acción para evitarse utilizar métodos de selección de valores de una política.

Al observar la solución de los ejercicios se concluyo que el método de Q learnig es muy bueno en estados conocidos, pero en horizontes infinitos la situación cambia, ya que la recompensa se dirige a las acciones y sabe en qué estados se realizaron ciertas acciones, pero cuando el agente no reconoce que ya visitado un estado, quiere decir que perdió la memoria para ese estado por qué no lo reconoce, entonces solo depende de su acción pasada para su acción futura, y el MDP premia directamente el conjunto de políticas pudiéndose guardar estas premiaciones en una memoria y así realizar una mejor elección de la política que se va a escoger.

Generalmente cuando se habla de aprendizaje se asocia con una base de conocimiento la cual es aprendida previamente, en el caso de aprendizaje por refuerzo esa base de conocimiento se construye paso a paso, pero no memorísticamente si no funcionalmente dependiendo del estado actual, por lo tanto el gasto computacional es bajo.

La generación de una exploración previa sobre un entorno desconocido nos permite realizar un mapa de posibles rutas para planear diferentes tipos de tareas, la generación de un mapa de rutas único nos proporciona datos de donde se pueden derivar distancias, obstáculos, giros, y tiempos de desplazamiento, datos necesarios para conocer un entorno desconocido y generar una estrategia de decisión sobre la meta que se desea lograr. El procedimiento a seguir después de obtener estos resultados es implementar el método en plataformas reales para confrontar los resultados de las simulaciones y obtener unos nuevos resultados basados en situaciones reales.

References

[1] Richard S. Sutton and Andrew G. Barto, Reinforcement Learning: An Introduction, A bradford book the mit press cambridge, massachusetts london, England.

[2] Maes, Pattie (1995), "Artificial Life Meets Entertainment: Life like Autonomous Agents," *Communications of the ACM,* Belmont, CA: Wadsworth, 1993, pp. 123–135, 38, 11, 108-114

[3] María Elena López Guillén, Sistema de navegación global basado en procesos de decisión de markov parcialmente observables. Aplicación a un robot de asistencia personal. Tesis Doctoral, UNIVERSIDAD DE ALCALÁ, España, 2004, pp. 57

[4] M.L. Puterman. "Markov Decision Processes – Discrete Stochastic Dynamic Programming", John Wiley & Sons, Inc., New York, NY. 1994.

[5] Watkins, C.J. "Learning from Delayed Rewards, PhD Thesis", King's College, Cambrige University, UK. 1989.

[6] Mitchell1997, Tom M. Mitchell, *Machine Learning*. , McGraw-Hill, 1997.

BIOGRAFÍA

Edgar Alirio Aguirre Buenaventura. Ingeniero en Control Electrónico e Instrumentación de la Universidad Distrital Francisco José de Caldas. Tecnólogo en Electrónica de la misma universidad en el 2005. Integrante del Grupo de investigación en Robótica Móvil Autónoma - ROMA, con intereses en el área del control electrónico y robótica móvil. Email: roma@udistrital.edu.co

Giovanni Rodrigo Bermudez Bohórquez. Ingeniero Electricista de la Universidad Nacional de Colombia. Magíster en Ingeniería Electrónica de la Universidad de los Andes. Director del Grupo de investigación en Robótica Móvil Autónoma – ROMA. Director del Centro de Investigaciones y Desarrollo Científico de la Universidad Distrital. Profesor Asociado de la Universidad Distrital. Email: gbermudez@udistrital.edu.co