Sistema de toma de decisiones basado en emociones y autoaprendizaje para agentes sociales autónomos

María de los Ángeles Malfaz Vázquez

Director:
Dr. Miguel Ángel Salichs

Escuela Politécnica Superior
de la Universidad Carlos III de Madrid

Sistema de toma de decisiones basado en emociones y autoaprendizaje para agentes sociales autónomos

Sistema de toma de decisiones basado en emociones y autoaprendizaje para agentes sociales autónomos

María de los Ángeles Malfaz Vázquez

Director:
Dr. Miguel Ángel Salichs

Tesis Doctoral

2007

Escuela Politécnica Superior
de la Universidad Carlos III de Madrid

Universidad Carlos III

Publication Data:
María de los Ángeles Malfaz Vázquez
Sistema de toma de decisiones basado en emociones y autoaprendizaje para agentes sociales autónomos
Universidad Carlos III

— A los Drs. Malfaz, mis padres y hermanos —

AGRADECIMIENTOS

Supongo que todo esto empezó cuando una noche le pregunté a mi padre qué tenía que estudiar para saber de las estrellas. La respuesta: Físicas. Así que eso hice, estudié física y me di cuenta de que era mucho más divertido eso de saber controlar las cosas de aquí abajo, que estudiar lo de allá arriba. Una cosa llevó a la otra y en unos cuantos años me vi en un despacho, con el que sería mi tutor contándome cosas sobre robots y emociones...

Sin duda, esta tesis y todo el camino previo ha sido posible gracias a mis padres. No puedo decir nada más que gracias por su apoyo constante. A ellos y a mis hermanos les agradezco su cariño y sus ánimos para continuar con la tesis.

Por supuesto, en estos cinco años ha habido mucha gente en el departamento que me ha aportado mucho, personas que ya no están y otras que continúan animándome todos los días, en el café, en la comida, en el pasillo... A todos ellos les agradezco su amistad y confianza.

Me encantaría poder poner aquí los nombres de todos mis amigos para que todos ellos supieran lo importantes que han sido en esta tesis. Como podría ser un poco largo, voy a dejarlo en un agradecimiento profundo a mis amigos de toda la vida de Tenerife, que hacen que siempre quiera volver a casa, y a mis amigos de Madrid que hacen que esta sea mi nueva casa.

En realidad el responsable de que yo presente este trabajo es mi tutor, Miguel Ángel. Gracias a su entusiasmo e imaginación, esta tesis me enganchó desde el primer día y me hizo partícipe de un proyecto único. Este trabajo es el resultado de horas y horas de charla, de discusiones y de risas, gracias. No me olvido de mi grupo, de nuevo gracias por estar siempre apoyándome y animándome. Hago una mención especial a Ramón... ¿qué sería de nosotros sin él? gracias por estar siempre disponible.

Por último, quería dar las gracias a Antonio, por compartir esta tesis y por preocuparse tanto como yo de que todo estuviera perfecto. Gracias por animarme siempre que se me hacía cuesta arriba y enseñarme la parte buena de todo. Gracias a los Lapuente-Iribas por hacerme sentir como en casa.

RESUMEN

"La cuestión no es si las máquinas inteligentes pueden tener alguna emoción, sino si las máquinas pueden ser inteligentes sin ninguna emoción".

Minsky en *La sociedad de la mente* (1986).

El objetivo de esta tesis es desarrollar un sistema de toma de decisiones para un robot personal social y autónomo. Este sistema consiste en varios módulos: el módulo motivacional, el módulo de los *drives* y el módulo de valoración y selección de comportamientos. Estos módulos están basados en motivaciones, *drives* y emociones, conceptos que son ampliamente desarrollados en esta tesis.

Debido a las dificultades de trabajar con un robot real, como primer paso en esta investigación, se ha optado por la implementación previa de esta arquitectura en agentes virtuales. Dichos agentes viven en un mundo virtual que ha sido construido utilizando un juego de rol basado en texto del tipo MUD (Multi User Domain). En este mundo los agentes interaccionan entre sí y con diferentes objetos. Se ha elegido este tipo de juegos basados en texto en lugar de otros más modernos con interfaces gráficas, porque la interpretación de la información es mucho más sencilla.

La selección de los comportamientos es aprendida por el agente mediante la utilización de algoritmos de aprendizaje por refuerzo. Cuando el agente no interactúa con otros agentes utiliza el algoritmo Q-learning. Cuando existe interacción social el refuerzo que recibe el agente es debido a la acción conjunta con el otro agente y, en este caso, hace uso de algoritmos de aprendizaje multi-agente, también basados en Q-learning.

El estado interno del agente se considera como la motivación dominante, lo que provoca que el sistema no sea completamente markoviano y por lo tanto no tan sencillo de tratar. Con el fin de simplificar el proceso de aprendizaje, los estados relacionados con los objetos del mundo se consideran como independientes entre sí. El estado del agente es una combinación de su estado interno y su estado en relación con el resto de agentes y objetos.

El hecho de considerar los estados en relación a los objetos como independientes, hace que no se tenga en cuenta la posible relación entre ellos. En realidad, las acciones que se realizan con un objeto pueden llegar a afectar al estado en relación con los otros objetos, provocando "efectos colaterales". En esta tesis se propone una variación original del algoritmo Q-learning que considera dichos efectos.

En este sistema se utilizan la felicidad y la tristeza como refuerzo positivo y negativo respectivamente. Por lo tanto, los comportamientos no son seleccionados para satisfacer los objetivos determinados por las motivaciones del agente, sino para alcanzar la felicidad y evitar la tristeza.

Las teorías de valoración de las emociones afirman que éstas son el resultado de procesos evaluadores y por lo tanto subjetivos. En base a estas teorías se considera, en este sistema de toma de decisiones, que ciertas emociones son generadas a partir de la valoración del bienestar del agente. Este bienestar mide el grado de satisfacción de las necesidades del agente. De esta forma se asume que la felicidad se produce cuando le sucede algo bueno al agente, aumentando el bienestar. La tristeza, por el contrario, se produce cuando pasa algo malo, provocando una disminución del bienestar.

Finalmente, se introduce la emoción del miedo desde dos puntos de vista: tener miedo a realizar acciones arriesgadas y tener miedo a estar en estados peligrosos. Para ello se considera el miedo como otra motivación del agente, lo cual también coincide con otras teorías de emociones.

ABSTRACT

"The question is not whether the intelligent machines can have any emotions, but whether machines can be intelligent without any emotions".

Minsky in *The society of mind* (1986).

The objective of this thesis is to develop a decision making system for an autonomous and social robot. This system is composed by several subsystems: a motivational system, a drives system and an evaluation and behaviour selection system. All systems are based on motivations, drives and emotions. These concepts are described in detail in this thesis.

Due to the difficulties of working with a real robot, it has been decided to implement this decision making system on virtual agents as a previous step. These agents live in a virtual world which has been built using a text based MUD (Multi-User Domain). In this world the agents can interact with each other, allowing the social interaction, and with the other objects present in the world. The reason why this text based game was selected, instead of a modern one with graphic interfaces, is that the interpretation of the information is much simpler.

The selection of behaviours is learned by the agent using reinforcement learning algorithms. When the agent is not interacting with other agents he uses the Q-learning algorithm. When social interaction exists, the rewards the agent receives depend not only on his own actions, but also on the action of the other agent. In this case, the agent uses multi-agent learning algorithms, also based on Q-learning.

The inner state of the agent is considered as the dominant motivation. This fact causes that the system is not completely markovian and therefore, no so easy to work with. In order to simplify the learning process, the states related to the objects are considered as independent among them. The state of the agent is a combination between his inner state and his state in relation with the rest of agents and objects.

The fact that the states in relation to the objects are considered as independent, causes that the possible relation between objects is ignored. In fact, the actions related to an object, may affect the state in relation to the other objects, causing "collateral effects". In this thesis, a new variation of Q-learning is proposed to consider these effects.

This system uses happiness and sadness as positive and negative reinforcement respectively. Therefore, behaviours are not going to be selected to satisfy the goals determined by the motivations of the agent, but to reach happiness and avoid sadness.

The appraisal emotions theories state that emotions are the result of evaluation processes and therefore they are subjective. Based on those theories, it is going to be considered, in this decision making system, that certain emotions are going to be generated from the evaluation of the wellbeing of the agent. The wellbeing measures how much the needs of the agent are satisfied. Happiness is produced because something good has happened, i.e. an increment of the wellbeing is produced. On the contrary, sadness is produced because something bad has happened, so the wellbeing decreases.

Finally, another emotion is introduced: Fear. Fear is presented from two points of view: to be afraid of executing risky actions, or to be afraid of being in a dangerous state. In this last case, fear is considered as a motivation, in accordance with other emotions theories.

ÍNDICE GENERAL

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

1. INTRODUCTION

1.1. Motivation of this thesis

This thesis is part of a large project which its main objective is to build an autonomous and social robot. The robot must learn to select the right behaviours in order to achieve its goals. The mechanisms involved in the decision making process are inspired on those used by humans and animals. Since it is a social robot, one of the required features would be the life-like appearance. The social aspect of the robot will be reflected in the fact that the human interaction is not going to be considered as a complement of the rest of functionalities of the robot, but as one of its basic features. For this kind of robots autonomy and emotions make them behave as if they were "alive". This feature would help people to think of them not as simple machines, but as real companions. For certain applications, a robot with its own personality is more attractive than another that simply executes the actions that it is programmed to do.

Many decision making architectures based on emotions have been implemented previously on robots. Most of them, as it will be shown in this thesis, place emphasis on the external expression of the emotions [Breazeal, 2002], [Fujita, 2001], [Shibata et al., 1999]. These robots have the possibility of expressing emotions through facial expressions and, sometimes, through their body gestures. In this case, emotions can be considered as a particular type of information, which is exchanged in the human-robot interaction. In nature, emotions have different purposes, and interaction is only one of them.

Emotions have a fundamental role in human behaviour and social interaction. They also influence cognitive processes, particularly problem solving and decision making [Damasio, 1994]. Emotions can also act as control and learning mechanisms [Fong et al., 2002]. In this thesis, emotions are used for trying to imitate their natural function in learning processes and decision making.

1.2. Objectives and methodology

The final goal of this thesis is to design a decision making system, based on emotions, for an autonomous and social robot with no *a priori* knowledge. This means that the robot is the one who decides its own actions, and it interacts with other robots or people. One important feature of the robot is that, using unsupervised learning, it learns the right behaviours to execute through its own experience.

In general, the decision making process can be approached from two points of view: Deliberative process and reactive process. In a deliberative process the robot knows the global model of the world. This model gives the result of every action. Therefore, the decision making process is based on a "What happens if?" mechanism and the robot can "think" about the future and the past. On the other hand, in a reactive process there is no model of the world so the effect of an action is unknown *a priori*. The actions are evaluated automatically and the decisions are based on the present value of the situation. In this thesis the decision making process does not have a model of the world, therefore, the robot acts in a reactive way.

Before the implementation of this system on a real robot, as a previous step, this thesis is going to be developed using virtual agents, instead of robots. The agent lives in a virtual world where objects, needed to survive, and other agents exist. This agent must learn a policy of behaviour to survive, maintaining all his needs inside acceptable ranges. The policies establish a normative about what to do in each situation. This means that the agent must learn the right relation between states and actions. In this system the agent knows the properties of every object, i.e. the agent knows which actions can be executed with each object. What the agent does not know is which action is right in each situation. In order to carry out this learning process, the agent uses reinforcement learning algorithms.

In this thesis it is considered that the role each emotion plays, and how its associated mechanisms work are very specific. This implies that each emotion must be implemented on the robot in a particular way. In this thesis some proposals, of how some emotions, such as happiness, sadness and fear, can be used on autonomous agents, are going to be presented.

1.3. Problems to be solved

In order to carry out the final goal, some problems arise and need to be solved:

- A problem appears when interacting with the objects of the world. From the learning point of view, the management of many objects causes that the number of states increases significantly.
- The selection of the reinforcement function, that is going to be used, must be done. This reinforcement must be inner, this means that it must be produced by the own agent.
- It must be studied how the need of social interaction is going to be handled.
- In relation with the interaction with other agents, another problem arises related to the reinforcement learning. When an agent is interacting with another, the reward received is not due to its own action but due to their joint actions. Therefore, the treatment of this type of interaction must be different.

- It must be investigated to what extent emotions favor the autonomy of the agent and improve his quality of life. The fact that living beings have emotions, make us consider their role in robots, as well as if their implementation can help the consecution of the goal of this thesis.

Other authors, as it is going to be shown in this document, have already presented and implemented other decision making systems similar to the one proposed. The main differences from these systems are related mainly to the generation and use of the emotions, to the treatment of the interaction of the agent with objects and other agents, and to the particular social nature of the agents.

1.4. Related publications

Preliminary results of this thesis have already been published:

- J.F.Gorostiza; R.Barber; A.M.Khamis; M.Malfaz; R.Pacheco; R.Rivas; A.Corrales; E.Delgado; M.A.Salichs. *Multimodal Human-Robot Interaction Framework for a Personal Robot*. RO-MAN 06: The 15th IEEE International Symposium on Robot and Human Interactive Communication. Hatfield. United Kingdom. Sep, 2006.
- M.Malfaz; M.A.Salichs. *Using Emotions for Behaviour-Selection Learning* . The 17th European Conference on Artificial Intelligence. ECAI 2006. Riva del Garda. Italy. Aug, 2006.
- M.A.Salichs; R.Barber; A.M.Khamis; M.Malfaz; J.F.Gorostiza; R.Pacheco; R.Rivas; A.Corrales; E.Delgado. *Maggie: A Robotic Platform for Human-Robot Social Interaction*. IEEE International Conference on Robotics, Automation and Mechatronics (RAM 2006). Bangkok. Thailand. Jun, 2006.
- M.Malfaz; M.A.Salichs. *Emotion-Based Learning of Intrinsically Motivated Autonomous Agents living in a Social World*. International Conference on Development and Learning 2006. ICDL5. Bloomington, In. USA. May, 2006.
- M.Malfaz; M.A.Salichs. *Learning Behaviour-Selection Algorithms for Autonomous Social Agents living in a Role-Playing Game* . Narrative AI and Games, part of AISB'06: Adaptation in Artificial and Biological Systems. University of Bristol. Bristol. England. Apr, 2006.
- M.A.Salichs; M.Malfaz. *Using Emotions on Autonomous Agents. The Role of Happiness, Sadness and Fear*. Integrative Approaches to Machine Consciousness, part of AISB'06: Adaptation in Artificial and Biological Systems. Bristol. England. Apr, 2006.
- M.Malfaz; M.A.Salichs. *A new architecture for autonomous robots based on emotions*. Fifth IFAC Symposium on Intelligent Autonomous Vehicles. Lisbon. Portugal. Jul, 2004.

- M.Malfaz; M.A.Salichs. *Design of an Architecture Based on Emotions for an Autonomous Robot*. 2004 AAAI Spring Symposium. Stanford. California. Mar, 2004.

1.5. Overview of contents

The material in this thesis is arranged by chapters as follows:

- In Chapter 2 a brief review about personal robots is presented. Nowadays, personal robots can be mostly classified in the following: pet robots, housework robots and social robots.

- In Chapter 3 the concept of autonomy, and its meaning from several points of view, is introduced. This autonomy implies the introduction of new concepts: motivations and drives. Both concepts are explained in this chapter. Several theories, that try to explain what motivations are, are presented. Later, learning is going to be presented from the biological point of view, as well as its application to robotics, giving more relevance to the reinforcement learning. Finally, several reinforcement learning algorithms are presented, such as Q-learning and another multi-agent algorithms, also based on Q-learning.

- In Chapter 4 several reasons that justify the use of emotions in robots are presented. Next, different definitions of emotion are given from several research areas such as psychology and neuroscience. Later, the role of emotions in memory, perception, reasoning and learning is analyzed. There are many researchers that prove the great importance of emotions on the decision making process. Next, the different classifications of emotions, from several points of view, are exposed. Lastly, the question "why do robots need emotions?" is proposed. There are many answers and most of them are related with their relevance in the human and animal behaviour.

- In Chapter 5 a wide review of emotion based architectures for robots is presented. Among them, we give more importance to those proposed by Lola Cañamero and Sandra Gadanho respectively, due to their relevance in the development of this thesis. Finally, it is shown how some robots express their emotions through language, facial expressions and body gestures.

- In Chapter 6 the decision making system proposed in this thesis is presented. First, a description of how this system works is given. Next in this chapter, the state of the agent is defined as a combination of the inner state and the external state of the agent. The inner state of the agent is defined based on drives and motivations, and the external state is introduced as the relation of the agent with all the objects present in the world. Later, the reinforcement function used in the learning process, happiness and sadness, as well as the wellbeing of the agent are defined. Due to a simplification done in relation to the state of the agent, the modifications of the Q-learning algorithm and the multi-agent

algorithms are presented. Lastly, the emotion fear is introduced from two points of view: to be afraid of executing risky actions and to be afraid of being in a dangerous state.

- In Chapter 7 the experimental procedure is explained in detail. First, the environment, the virtual world where the agents live, is presented. Next, the experimental settings of the agent are described: the drives and their dynamics, motivations, the states of the agent and the actions that can be executed with each object, including other agents. Finally, the results analysis indicators, which are going to be used in later chapters to analyze the performance of the agent, are introduced.

- In Chapter 8 the results obtained when the agent is living alone are presented. In this chapter, the reinforcement function is selected. Moreover, the values of several parameters related to the learning process, are tuned. Those values are going to be used in the rest of the experiments carried out in this thesis.

- In Chapter 9 the results presented correspond to the situation where the agent lives with other agents. These agents can have one of the three different personalities: good, neutral and bad. Therefore, depending on the personality of the agents presented in the environment, four types of worlds are defined: good, neutral, bad and mixed. The agent lives in each of these worlds using the multi-agent learning algorithms introduced in chapter 6. Finally, the performances of the agent, shown for each world using each algorithm, are compared.

- In Chapter 10 the emotion fear is introduced in the experiments. As it has been shown, fear is going to be considered from two points of view: to be afraid of executing risky actions and to be afraid of being in a dangerous state. Therefore, this chapter is separated in two parts. In the first part, the agent learns to interact with a new object that most of the times is good for him, but occasionally is very noxious. Therefore, executing actions related with this object is risky. The agent uses the emotion fear to deal with this situation. The results obtained when the agent is afraid of selecting those risky actions, as well as when he is not, are presented. In the second part of this last chapter of experiments, the agent lives with other agents, and one of them, once in a while, kicks him. It is considered that when the agent is accompanied by that agent, he is in a dangerous state. In this occasion, the agent uses the emotion fear as a new motivation to try to avoid this dangerous state. The results presented in this part correspond to the performances of the agent when he is using fear, and when he is not.

- In Chapter 11 first, a brief review of the experimental results is presented. Next, the main conclusions are presented as well as the main contributions of this thesis. Finally, some future works are suggested.

2. ROBOTS PERSONALES

2.1. Introducción

Los robots personales son robots diseñados para estar en un entorno común con personas, para entretener o ayudar. Los robots personales podrían tener múltiples funciones como:

- Mascota: Un robot personal podría ser utilizado como una mascota artificial, de manera que el robot es autónomo y además puede aprender y adaptarse a su dueño.
- Vigilancia: El robot actuaría como un guardián para detectar intrusos, fuegos, etc. Esta vigilancia sería realizada de manera autónoma por parte del robot, o podría realizarse de manera teleoperada a través de internet o utilizando un teléfono móvil.
- Robots para trabajos domésticos: Un robot personal podría realizar tareas domésticas como limpiar, aspirar, etc. Por otro lado, también sería capaz de controlar otros dispositivos electrónicos presentes en la casa, como el control de la calefacción, luces, etc.
- Entretenimiento: Por supuesto, un robot personal, podría ser utilizado para entretener al usuario a través de juegos, contar historias, etc. En esta misma linea de interacción con el usuario, también serviría como fuente de información. El robot, al igual que un ordenador personal, accedería a través de internet a la información requerida por el usuario, como datos del tiempo, o las noticias.
- Ayudante personal: Una aplicación bastante interesante de los robots personales es que fuesen capaces de ayudar al usuario a organizar su horario, reuniones, recordarle sus tareas, etc.

Cada una de estas funciones puede dar lugar a distintos tipos de robots personales. La principal diferencia entre un robot mascota o de entretenimiento y un robot para uso doméstico, radica en su comportamiento. Una cualidad esencial de una mascota es su comportamiento autónomo y emocional. Una mascota debe comportarse de forma semejante a un ser vivo y eso implica también responder de forma diferente según cual sea su estado emocional. Es de esperar que una mascota se comporte de forma hiperactiva si está “alegre”, o que se muestre perezosa y no obedezca si está “cansada”. Es decir, su comportamiento debe ser en cierta medida impredecible. Por

ejemplo, en el robot AIBO de Sony, ver la figura 2.1, sólo algunos comportamientos están documentados, los otros deben ser descubiertos por el propio usuario. Incluso en productos futuros los comportamientos podrían variar de un robot a otro para dotarle de cierta "personalidad". Sin embargo, ésa no es una cualidad recomendable en un robot dedicado a la realización de tareas domésticas. A nadie le gustaría tener un robot aspirador que deje repentinamente de funcionar porque está "de mal humor". Sin embargo también en este caso, un limitado comportamiento emocional puede facilitar la interacción con este tipo de robots.

Las aplicaciones potenciales de los robots personales son múltiples, como ya se ha mostrado, y pueden variar con las características de sus propietarios. Para un niño, además de ser un juguete, puede ser un profesor. Para un adulto puede ser un asistente, tanto en el trabajo como en el hogar. Para personas mayores o discapacitadas puede convertirse en un ayudante para hacerles la vida más fácil. Evidentemente, el sistema de comunicación persona-robot debe ser amigable y facilitar el uso del robot por parte de personas sin una formación especial. Aquí tiene un papel importante la implementación de un modelo emocional, ya que por ejemplo en el caso de las personas mayores, los cuales en su mayoría sienten rechazo hacia las máquinas, el hecho de que un robot tenga emociones y se comporte como si estuviese "vivo", haría que estas personas dejaran de pensar en ellos como simples máquinas, tomándolos como animales de compañía. Evidentemente, para el caso de los niños y muchos adultos, un robot que siente y tiene personalidad propia es mucho más atractivo que uno que no hace más que lo está programado en su memoria.

Puede ocurrir también que un mismo robot personal sea completamente autónomo y que además tenga varias de las funciones previamente descritas, como la de entretenimiento y ayudante personal. Así se podría tener robots que fueran capaces de interactuar con humanos de una forma natural y que además participaran en la sociedad. De esta manera, los humanos interaccionarían con el robot como si ellos fueran compañeros, no supervisores del robot. Este tipo de robots personales se denominan robots sociales. Se puede definir un robot social como "un robot autónomo o semi-autónomo que interacciona y se comunica con humanos, siguiendo las normas de comportamiento esperadas por la gente con la que pretende interaccionar" [Bartneck and Forlizzi, 2004].

2.2. Mascotas robots

En los últimos años ha tenido lugar una aparición de robots mascota a gran escala, principalmente en Japón. Quizás el más representativo es, a nivel de ventas, AIBO, el perro-robot desarrollado por Sony. Los primero modelos de AIBO se sacaron al mercado en el año 2001, figura 2.1(a), y durante los siguientes cinco años fueron perfeccionados. Finalmente en marzo del 2006 la compañía Sony dejó de producir AIBOs [Sony, 2006]. El último modelo, figura 2.1(b), puede reconocer su estación para recargarse de forma autónoma, además de reconocer la voz y cara de su dueño. Por otro lado, también se puede encontrar a un gato-robot, NeCoRo de la casa Omron,

figura 2.2, que es muy popular en Japón desde su aparición también en el año 2001 [Omron, 2007]. Este gato-robot está dotado de pelo artificial, lo que le da un aspecto bastante real. Al igual que Aibo, tiene capacidad de reconocer la voz de su dueño y de "sentir". Su personalidad va evolucionando con el tiempo dependiendo de su interacción con el dueño. Más recientemente en el año 2005, Philips desarrolló otro gato-robot llamado iCat (figura 2.3), el cual ha sido diseñado como una plataforma de investigación para el estudio de la interacción humano-robot [Philips, 2007].

(a) AIBO ERS-311, el primer modelo comercial

(b) AIBO ERS-7, el último modelo comercial

Fig. 2.1: *AIBO*

Fig. 2.2: *NeCoRo*

Fig. 2.3: *iCat*

2.3. Robots de uso doméstico

Un sueño, que todavía resulta inalcanzable, es disponer de un robot que realice todo tipo de tareas domésticas: limpiar, planchar, cocinar, etc. Sin embargo, la tecnología actual sólo permite construir robots capaces de llevar a cabo tareas domésticas sencillas, tales como vigilancia, transporte o pasar la aspiradora. Cye de Probotics (figura 2.4) y Roomba de iRobot (figura 2.5) son ejemplos de robots comerciales con estas capacidades. Cye salió al mercado en 1999. Este robot puede transportar cosas usando una bandeja, aspirar el suelo e incluso, usando una cámara, puede ser utilizado como un robot espía [Probotics, 2007]. Roomba, desarrollado en el año 2002, puede ser programado para que limpie cuando el usuario quiera, incluso por la noche o cuando no está en casa. Además, es capaz de ir a recargarse autónomamente cuando ha terminado de limpiar o cuando tiene la batería baja [iRobot, 2007].

Fig. 2.4: *Cye*

Fig. 2.5: *Roomba*

En esta misma línea, Fujitsu comercializa desde el año 2002 el robot Maron-1 (figura 2.6) que está pensado para que realice tareas de vigilancia en una casa. Este robot permite, entre otros modos de comunicación, recibir comandos de su dueño por teléfono móvil y enviar, a través de éste, información sobre lo que ocurre en la casa [Fujitsu, 2007].

Fig. 2.6: *Maron-1*

2.4. Robots sociales

Otros muchos robots personales han sido desarrollados pero no comercializados. Este es el caso de PaPeRo, un pequeño robot social creado para "vivir" entre humanos, fabricado por NEC (figura 2.7) [NEC, 2006]. Este robot es un prototipo de robot personal y está dotado no sólo de la mayoría de las capacidades propias de una mascota (reconocimiento de caras y voz), sino de las de un miembro más de familia, ya que es capaz de mantener conversaciones con varias personas a la vez, contar chistes, dar

recados, etc. Actualmente la aplicación de este tipo de robots es casi únicamente el entretenimiento, aunque tienen un enorme potencial aplicado a la educación de niños pequeños.

Fig. 2.7: *PaPeRo*

Otro robot en esta misma línea es Maggie, un robot desarrollado por el RoboticsLab del Departamento de Ingeniería de Sistemas y Automática de la Universidad Carlos III de Madrid. Este robot ha sido diseñado como un robot social, que es capaz de comunicarse mediante el habla, dando información por ejemplo del tiempo, o de noticias obtenidas a través de internet. Además está provisto de otras capacidades como reconocimiento de caras, funciones de seguimiento del interlocutor, capacidad de diálogo, etc.

Fig. 2.8: *Maggie*

Físicamente Maggie, ver la figura 2.8, mide 1,35m y está construido sobre una base Magellan Pro móvil producida por iRobot. Esta base está equipada con 12 sensores de infrarrojo y 12 de ultrasonido, además de un sensor láser. Maggie ofrece una apariencia antropomórfica, ya que consta de cabeza, torso y brazos. La cabeza tiene dos grados de libertad y está equipada con dos ojos con párpados controlables y una boca que tiene luces sincronizadas con el habla, además de una webcam oculta detrás de ella. Se construyeron dos brazos para dotar de expresión no verbal al robot a través de gestos. En el pecho, Maggie tiene un tablet PC para dar información audiovisual. A este tablet PC van conectados un micrófono sin cables, que funciona por bluetooth, y dos altavoces. El tablet PC es también responsable de mostrar imágenes como respuesta a eventos táctiles en la pantalla. Por otro lado se han incorporado a Maggie varios sensores capacitivos invisibles que funcionan como sensores táctiles en la parte superior del robot [Salichs et al., 2006].

3. AUTONOMÍA: MOTIVACIONES Y APRENDIZAJE

3.1. Introducción a la Autonomía

Tradicionalmente los sistemas de interacción humano-robot están basados en la idea maestro-esclavo. De acuerdo con esta idea, el papel del operador humano es la de supervisar y dar órdenes al robot, mientras que el papel del robot es la de cumplir dichas tareas y, eventualmente, dar la información necesaria al operador. El robot esencialmente actúa como una herramienta usada por un operador. En estos sistemas la interacción con el humano es un factor limitador, que reduce la autonomía del robot. Dependiendo del campo de aplicación existen muchas definiciones de la palabra autonomía. En esta sección se van a exponer algunos puntos de vista sobre qué es la autonomía en robótica y lo que ello implica.

Según Arkin, para que un robot sea realmente autónomo, no sólo debe ser capaz de acciones inteligentes, sino que debe ser también auto-sostenible. Es decir, el robot debe ser capaz de reconocer sus propias necesidades como energía, fallo de algún componente y auto-preservación en general. Si surge un problema, el robot debe adaptar sus acciones y planes, para oponerse a continuar por un camino que le llevaría a la "muerte". El robot debe ser capaz de librarse por sí mismo de situaciones peligrosas, no sólo evitándolas a través de planes de alto nivel, sino que cuando las cosas salen mal debe responder de una manera rápida [Arkin, 1988].

En otras palabras, la autonomía implica tomar decisiones y esto implica algún conocimiento del estado actual del agente y del entorno, incluyendo sus objetivos. Es decir, que el agente debe tener suficiente conocimiento de sí mismo para razonar sobre cómo moverse y actuar en su entorno con todas sus propiedades y capacidades [Bellman, 2003].

Según Cañamero, los agentes autónomos son sistemas naturales o artificiales en permanente interacción con entornos dinámicos, impredecibles, con recursos limitados y en general sociales, que deben satisfacer un conjunto de posibles objetivos conflictivos para sobrevivir. Además, en relación a la toma de decisiones, los agentes pueden ser autónomos a diferentes niveles dependiendo de su morfología, su complejidad funcional y cognitiva, de su entorno y del tipo de interacción con el entorno. En un primer nivel, se dice que un agente es autónomo si puede actuar por sí mismo, sin necesidad de ser dirigido por otro agente. Este tipo de autonomía está relacionada con comportamientos simples, como son los comportamientos reflejos. Un nivel más

alto de autonomía se alcanza cuando el agente puede "elegir" si prestar o no atención y reaccionar ante un estímulo dado del entorno, en función de su importancia para los comportamientos y objetivos del agente. Esto implica que el agente tiene algunos objetivos o motivaciones internos que dirigen su comportamiento y sería, por lo tanto, motivacionalmente autónomo [Cañamero, 2003].

Desde este mismo punto de vista, Gadanho define un agente autónomo como un agente con objetivos y motivaciones, que tiene además alguna manera de evaluar los comportamientos en términos del entorno y de las propias motivaciones. Sus motivaciones son deseos o preferencias que pueden llevar a la generación y adopción de objetivos, siendo los objetivos situaciones que deben ser alcanzadas. Estos objetivos finales de un agente autónomo, o sus motivaciones, deben estar orientados a mantener el equilibrio interno del agente [Gadanho, 1999].

Es muy interesante, por otro lado, lo que Balkenius propone en su tesis doctoral. Este autor opina que se debería formular un principio, según el cual, un animal debería minimizar su pérdida más que maximizar su ganancia. Estas dos estrategias pueden parecer idénticas, pero hay muchos casos en los que son diferentes, lo cual es especialmente cierto cuando el resultado de un comportamiento es incierto. Para sobrevivir, un animal debería evitar tomar decisiones que pudieran llevarle a la muerte, incluso cuando el riesgo fuera pequeño. No importa la cantidad de comida que pueda recibir al realizar una acción potencialmente letal, esta ganancia nunca debería pesar más que el riesgo, si existe otra posibilidad que mantenga al animal vivo con toda seguridad [Balkenius, 1995].

Es interesante también el trabajo de Hallam y Hayes [Hallam and Hayes, 1992], donde exponen una serie de características del comportamiento, extraídas del comportamiento animal, que deberían ser tenidas en cuenta en el diseño de un robot/agente autónomo:

- Percepción: El robot debería ser rico en sensores, tanto en términos de tipos de sensores como en cantidad de información dada por cada tipo de sensor.
- Movimiento: El robot debería ser capaz de moverse de forma competente por su entorno y realizar acciones más elaboradas.
- Objetivos homeostáticos: El robot debería tener unas pocas variables internas que deben ser mantenidas en un rango, como por ejemplo el nivel de energía.
- Reacciones y aprendizaje: El robot debería ser capaz de mostrar reacciones rápidas a algunos de los estímulos de su entorno y además ser flexible para aprender la importancia de los estímulos.
- Navegación: El robot debería tener una base a la que regresar. Aunque esto no es esencial para un agente autónomo, la habilidad de regresar a algún punto de referencia de su entorno permite que el agente utilice comportamientos más complejos.

El aprendizaje, tal y como se expone en [Gadanho, 1999], es una habilidad importante para el agente autónomo ya que le dota de la plasticidad necesaria para ser independiente. El aprendizaje por refuerzo es una de las técnicas más usadas para el aprendizaje autónomo en el campo de la robótica y será tratada posteriormente.

En esta tesis se va a considerar que un agente autónomo es aquél que es capaz de establecer sus propios objetivos y además decidir qué comportamientos elegir para llevarlos a cabo. Dichos comportamientos están dirigidos a mantener el equilibrio interno del agente.

3.2. Homeostasis y *drives*

La homeostasis fue descubierta por Claude Bernard a mediados del siglo XIX, cuando observó que las variaciones corporales como las de la temperatura, la presión arterial y la frecuencia cardíaca tenían como objetivo devolver la estabilidad al cuerpo. Sin embargo, el término homeostasis fue acuñado en 1928 por el biólogo Walter Cannon, que recibió el Premio Nobel por su definición en 1932, en el libro *The Wisdom of the Body* [Cannon, 1932]. Desde un punto de vista menos fisiológico, homeostasis significa mantener el estado interno estable [Berridge, 2004]. Este estado interno es parametrizado por varias variables, las cuales deben estar alrededor de un nivel ideal. Cuando el valor de estas variables difiere del valor ideal, ocurre una señal de error: el *drive* o impulso. Estos *drives* impulsan a actuar para mantener el estado de las variables fisiológicas dentro de un rango determinado [Ávila García and Cañamero, 2004].

En la neurociencia del comportamiento, homeostasis normalmente significa un tipo específico de sistema regulador que usa un valor de referencia o un valor objetivo determinado para mantener estable el estado fisiológico. El valor de referencia se compara constantemente con el valor real del estado fisiológico y con esa comparación se detecta cuándo ocurre un error entre este valor real y su valor objetivo de referencia. Cuando el error es detectado, un mecanismo homeostático lanza las correspondientes respuestas para corregirlo. Por lo tanto, el concepto de homeostasis requiere varios mecanismos en el cerebro: un valor de referencia, un detector de error que mida la situación fisiológica actual y decida si existe un déficit, y un mecanismo de corrección del error para activar las correspondientes respuestas (Ej. Comer). Estas respuestas dan una realimentación negativa que corrige el déficit, el *drive*, y lleva el estado fisiológico real al valor del referencia [Berridge, 2004].

Esta aproximación al mecanismo homeostático es lo que en teoría de control de sistemas físicos se conoce como "control en lazo cerrado", tal y como se muestra en figura 3.1. Este mecanismo de control está formado por un sistema físico, un comparador, que es el que detecta el error entre el estado actual y el valor de referencia, y un controlador que es el que decide la acción a tomar para conseguir finalmente una señal de error nula. Esta señal de error nula implica el mantenimiento de las variables en su valor de referencia.

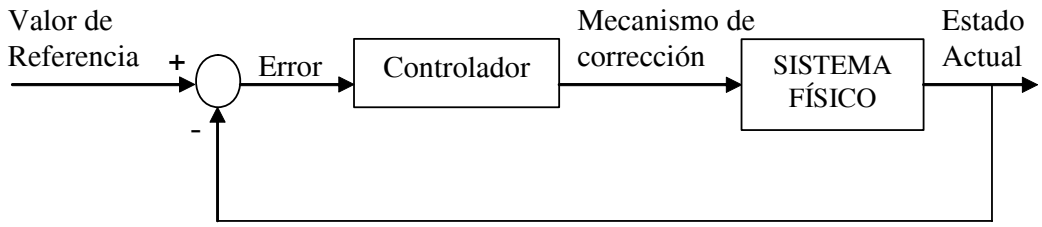


Fig. 3.1: *Bucle de control*

Una de las teorías sobre los *drives* más antiguas fue propuesta por Hull en 1943. Hull propuso que la privación induce a un estado de aversión en el organismo. Este estado fue llamado *drive* o impulso. En su teoría, el *drive* incrementa el nivel general de excitación en un animal. Los *drives* fueron entonces considerados como propiedades de los estados de necesidad, los cuales motivan el comportamiento. Hull consideraba que el *drive* tenía un efecto general de refuerzo sobre el comportamiento animal, pero no determina el siguiente comportamiento a realizar. Otros estudios posteriores [Bolles, 1967] demostraron que el papel de los *drives* es incrementar selectivamente la frecuencia de los comportamientos capaces de reducir el *drive*, lo cual no implica que la reducción del *drive* sea necesariamente lo que controla el proceso de aprendizaje [Hull, 1943].

3.3. Motivaciones en los seres vivos

La palabra motivación deriva del latín *motus* e indica la raíz dinámica del comportamiento, es decir, aquellos factores internos más que externos que incitan a la acción [Santa-Cruz et al., 1989].

Según Fernández Trespalacios [Trespalacios, 1980] "la motivación es la activación del organismo como un todo, por lo que éste pone en ejecución una conducta, ya anteriormente programada, o determina nuevas programaciones de ella".

Craig propuso, basándose en el estudio del comportamiento animal, que todos los comportamientos motivacionales pueden ser divididos en dos fases secuenciales, una fase apetitiva o instrumental seguida de una fase consumatoria. La fase apetitiva del comportamiento motivacional es el comportamiento flexible de aproximación que un animal o persona realiza antes de que el objetivo motivacional es encontrado. El comportamiento apetitivo flexible ayuda a encontrar el objetivo. La fase consumatoria sigue a la fase apetitiva sólo cuando el objetivo es obtenido. El comportamiento consumatorio es provocado por el estímulo del objetivo y por lo tanto consuma la fase apetitiva [Craig, 1918].

Existen varias teorías de la motivación que tratan de explicar la conducta humana y animal. Sin embargo, no existe ninguna clasificación única de todas estas teorías, por lo que en esta sección se van a presentar las que en la literatura parecen ser más relevantes.

3.3.1. Teorías homeostáticas de la motivación

De acuerdo con estas teorías, las conductas están dirigidas a mantener un equilibrio interno. De manera que se explican los comportamientos que se originan por un desequilibrio fisiológico. Existen varias teorías homeostáticas, de entre las cuales se van a distinguir dos: la teoría de reducción del *drive* y la teoría de la motivación y emoción.

Teoría de reducción de los *drives*

Muchas teorías sobre las motivaciones basadas en *drives* entre los años 1930 y 1970 afirmaron que la reducción de los *drives* es el principal mecanismo de recompensa o refuerzo. Si la motivación es debida al *drive*, entonces la reducción de las señales de déficit debería satisfacer estos *drives* y podrían ser esencialmente el objetivo de la motivación entera [Berridge, 2004].

Hull propone la idea de que la motivación está determinada por dos factores. El primero es el estado de necesidad o *drive*. El segundo es el incentivo, que es la presencia de un estímulo externo que predice una futura reducción de la necesidad del animal. Por ejemplo, la presentación de comida constituiría un incentivo para un animal hambriento [Hull, 1943].

Basándose en los trabajos de Hull y otros autores, Balkenius considera la motivación como un estado central que refleja la combinación de las necesidades internas y las posibilidades externas. El estado motivacional tiene dos propiedades importantes: la primera es que el estado motivacional es central a todo el organismo y la segunda es que es dinámico, es decir, todas las necesidades internas y posibilidades externas pueden cambiar o ser re-evaluadas. Además los determinantes de la motivación pueden ser divididos en tres categorías:

- Incentivo externo: La percepción del objeto deseado incrementa la motivación de su correspondiente estado. La percepción de comida incrementa el hambre aunque la necesidad fisiológica no cambie.
- Incentivo interno: El estado motivacional puede cambiar basándose en variaciones de procesos perceptuales o cognitivos. Este es el caso por ejemplo de cuando, después de echar un vistazo al reloj, uno se da cuenta de que el autobús sale en dos minutos y de repente nos vemos motivados a correr a la parada.
- *Drives* internos: El estado homeostático del agente también influye a la motivación. Por ejemplo, la privación de comida aumenta el hambre.

Estos tres determinantes son todos provocadores, en el sentido de que todos intentan incrementar la activación de un estado motivacional. Sin embargo, todos los estados motivacionales provocados no pueden ser permitidos para dirigir al agente al mismo tiempo ya que esto provocaría un comportamiento incoherente. Este problema según Balkenius, no puede ser resuelto mediante una competición de comportamientos, sino que debe resolverse en una etapa previa del proceso. La solución propuesta

es la competición entre motivaciones en proporción a su nivel actual de activación, de forma que sólo una motivación esté activa cada vez. Esto no significa que los otros estados motivacionales sean descartados, ya que éstos pueden ser activados en cualquier momento si los determinantes externos o internos cambian [Balkenius, 1993], [Balkenius, 1995]. Esta competición de motivaciones ciertamente existe. Por ejemplo, se tiene menos predisposición a tener hambre cuando se está cansado y con sueño.

Dentro de los distintos modelos de la motivación vale la pena destacar el modelo de *drives* hidráulico propuesto por Lorenz [Lorenz and Leyhausen, 1973]. El modelo de Lorenz es básicamente una metáfora que sugiere que el *drive* motivacional crece internamente y funciona como la presión de una reserva de líquido que crece hasta que sale precipitadamente a través de una salida. Las causas internas de un *drive* motivacional (ej. Señales de disminución fisiológica u hormonas segregadas relacionadas con el hambre, la sed, etc.) son como las corrientes de entrada que entran en la reserva, llenando el fluido para esa motivación. El estímulo motivacional en el mundo exterior (comida, agua, etc.) actúan para abrir la válvula de salida, liberando el *drive* que será expresado con un comportamiento, ver la figura 3.2.

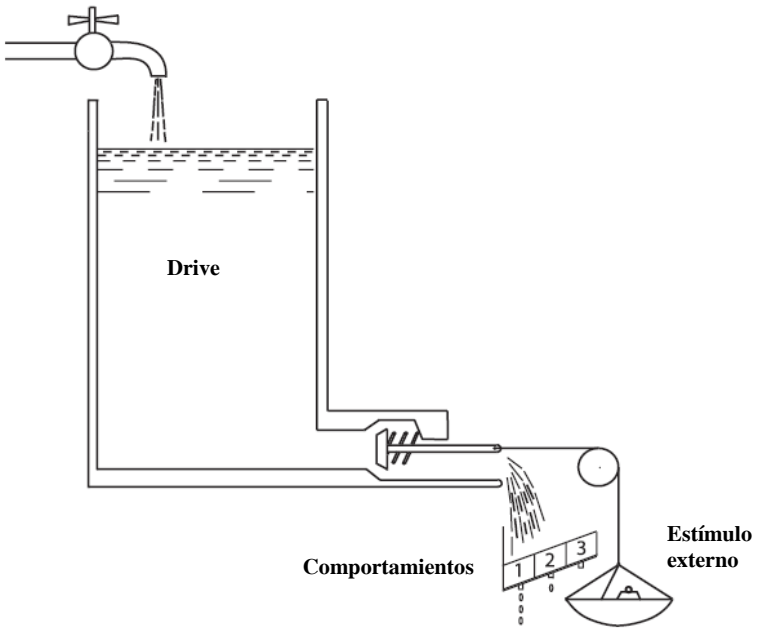


Fig. 3.2: *Modelo Hidráulico del* drive *motivacional de Lorenz*

En este modelo, la fuerza del *drive* interno se suma a la fuerza del estímulo externo. Si el *drive* es bajo, entonces se necesita un fuerte estímulo para activar el comportamiento motivado. Si el *drive* es alto, entonces un estímulo medio es suficiente. Cuanto más se abre la válvula, mas líquido del *drive* caerá, cubriendo más agujeros de desagüe del plato que se encuentra debajo, reuniendo más comportamientos para expresar la motivación.

Si el *drive* es extremadamente alto, puede llegar a desbordarse incluso sin estímulo externo provocando lo que se denomina "reacción de vacío". A pesar de que muchos de los comportamientos motivacionales son provocados según el modelo de Lorenz, algunos efectivamente no lo hacen. Por ejemplo, algunas respuestas motivacionales no reducen la motivación sino que incluso aumentan su intensidad. Este caso se da cuando a veces sin hambre comemos una almendra y de repente nos damos cuenta de que queremos algunas más. Este modelo no fue muy aceptado por la ciencia del comportamiento ya que no ofrecía muchos detalles sobre los mecanismos neurológicos. Sin embargo, los conceptos de los *drives* de Lorenz pueden ser a veces útiles, considerando la interacción entre los factores internos de las motivaciones y los estímulos externos en el control de los comportamientos motivacionales.

Teoría de la motivación y emoción

Existen varias propuestas que afirman que las emociones tienen un papel motivador del comportamiento. En esta sección se van a mostrar algunas de estas propuestas.

Breazeal en [Breazeal, 2002] propone un sistema motivacional para ser implementado en un robot, centrándose en dos clases de sistemas motivacionales: la regulación homeostática y la emoción. La regulación homeostática, como ya se ha explicado, parece ser necesaria para mantener los parámetros críticos de un animal dentro de un cierto rango. Por otro lado, las emociones son un importante sistema de motivación para sistemas complejos. Basándose en los estudios de otros investigadores, Breazeal sostiene que las emociones parecen estar involucradas en la determinación de la reacción a eventos. Estos eventos pueden ser del entorno o eventos internos de gran importancia para las necesidades de los objetivos de la criatura. En el sentido de las emociones entendidas como motivaciones, las emociones positivas son obtenidas por eventos que satisfacen algún motivo, como realzar el poder de supervivencia o demostrar el éxito de las capacidades. Las emociones negativas motivan acciones para corregir o prevenir que ocurran sucesos desagradables.

Para Rolls, sin embargo, existe una distinción entre motivaciones y emociones. La motivación es lo que hace que un individuo trate de obtener una recompensa o de escapar y así evitar un castigo. Un ejemplo de motivación es el hambre o la sed. Una recompensa es un estímulo o evento que uno quiere obtener, como la comida, y un castigo es lo que uno quiere evitar o de lo que trata de escapar, como estímulos dolorosos. La obtención de las recompensas o el evitar los castigos es el objetivo de la acción. Un estado motivacional es aquél en el que un objetivo es deseado. Una emoción es un estado provocado por recompensas o castigos. A pesar de esto, al describir las funciones de las emociones también afirma que las emociones pueden ser motivadores de comportamientos. Pone por ejemplo el miedo, el cual, aprendido por una asociación estímulo-refuerzo, es el que genera la motivación de las acciones para evitar estímulos nocivos [Rolls, 2003], [Rolls, 2005].

Otro punto de vista es que propone Arkin, de manera que las motivaciones tienden a ser más generales que las emociones. Arkin define la diferencia entre emociones y motivaciones desde el punto de vista de la robótica: las emociones constituyen un subconjunto de motivaciones que dan soporte a la supervivencia de un agente en un entorno complejo. Motivaciones y emociones afectan a la realización del comportamiento, pero las motivaciones pueden además llevar a la formulación de comportamientos concretos para alcanzar objetivos, mientras que las emociones tienen que ver con la modulación de los comportamientos existentes para mantener la actividad actual [Arkin, 2004].

Para concluir este punto, Balkenius dice que es posible considerar las motivaciones como estados que dicen al organismo lo que debería hacer, basándose en sus necesidades internas y en sus posibilidades externas. Las emociones, por otro lado, tienen que ver con lo que el animal debería haber hecho. El estado emocional tiene dos funciones. La primera es controlar el aprendizaje que permite al animal hacer frente de mejor forma a una situación similar a una anterior la próxima vez que se presente. La segunda es la de motivar comportamientos emocionales concretos [Balkenius, 1995].

3.3.2. Teorías de incentivo de la motivación

Los conceptos sobre los incentivos de las motivaciones empezaron a surgir en cuanto cayeron los conceptos relacionados con los *drives* en 1960. Se demostró que la teoría homeostática de las motivaciones no era cierta y que la reducción de los *drives* no es realmente un mecanismo de recompensa. Varios casos, entre ellos estudios con animales, demostraron que la idea de satisfacer el apetito no es puramente una cuestión de la reducción de un *drive* fisiológico. Por ejemplo, perros que eran alimentados de forma intravenosa seguían comiendo sus comidas normales por la boca en cuanto se les presentaba la oportunidad, además de recibir sus calorías de forma intravenosa. El *drive* homeostático no era la razón por la que comían y su motivación para comer no era satisfecha por la reducción fisiológica del *drive*. De manera similar, experimentos realizados con ratas sugerían que la motivación es más compatible con los conceptos de incentivos, como el sabor de la comida, que con la reducción del *drive*. Otros experimentos demostraron que las recompensas placenteras debidas al gusto causaban repentinos cambios en el comportamiento de las ratas y que este refuerzo placentero podía cambiar hábitos bien establecidos previamente [Berridge, 2004].

Otros autores revisaron muchos fallos experimentales de las motivaciones basadas en *drives* y de los conceptos de refuerzo de la reducción de *drives*. Se propuso en su lugar que los individuos estaban motivados por las expectativas de los incentivos, no por los *drives* o por la reducción de los *drives*. Las expectativas de los incentivos son esencialmente expectativas aprendidas por una recompensa placentera.

Cabe destacar que también se realizaron otras sugerencias en contra de las antiguas teorías de los *drives*, aunque finalmente es evidente que el *drive* fisiológico es importante para la motivación, incluso si el *drive* no es lo mismo que la motivación. No buscamos comida cuando tenemos sed. El déficit fisiológico como el hambre o la sed

modulan la motivación para conseguir la comida o el agua. Para incorporar el estado del *drive*/déficit fisiológico en los incentivos de la motivación, Toates [Toates, 1986] sugirió que el déficit fisiológico podía aumentar el valor del incentivo de su estímulo objetivo. Esto fue esencialmente una interacción multiplicativa entre el déficit fisiológico y el estímulo externo, el cual determina el valor del incentivo del estímulo. Las señales del déficit fisiológico no tenían porqué dirigir el comportamiento motivado directamente, ellas pueden magnificar el impacto placentero y el valor del incentivo de la recompensa.

3.4. Aprendizaje

El aprendizaje ha sido denominado como la marca distintiva de la inteligencia, por lo que el conseguir capacidades de adaptación y aprendizaje en sistemas artificiales es uno de los mayores retos de la inteligencia artificial [Mataric, 1998].

Lorenz definió el aprendizaje como los cambios adaptativos del comportamiento y ésta es, de hecho, la razón por la que existe en animales y humanos [Lorenz, 1977]. Existen dos mecanismos básicos de adaptación disponibles en los sistemas naturales:

- Filogenia: Adaptación de generación en generación, como resultado de la selección natural a través de la evolución.
- Ontogenia: Adaptación basada en el aprendizaje del individuo durante su vida.

En relación al aprendizaje en robots, Mataric en [Mataric, 1998] dice que el aprendizaje es particularmente difícil en los robots. Esto es debido a que interactuar y sentir en el mundo físico requiere tratar con la incertidumbre debida a la información parcial y cambiante de las condiciones ambientales. Sin embargo, el aprendizaje es una rama activa en robótica y es el aprendizaje por refuerzo uno de los métodos de aprendizaje que ha sido más implementado de forma efectiva en robots. De hecho, según algunos autores, el aprendizaje por refuerzo parece ser la elección natural para el aprendizaje de las políticas de control de robots móviles. En lugar de diseñar una política de control de bajo nivel, se puede diseñar una descripción de las tareas a un alto nivel, a través de la función de refuerzo. Frecuentemente, para las tareas de los robots, las recompensas corresponden a eventos físicos en el entorno. Por ejemplo para la tarea de evitar obstáculos, el robot puede obtener un refuerzo positivo si consigue su objetivo y negativo si choca contra algún obstáculo [Smart and Kaelbling, 2002].

Además de este tipo de aprendizaje existe otro tipo de aprendizaje, el aprendizaje supervisado. Los agentes que utilizan aprendizaje supervisado aprenden típicamente la relación entre entradas y salidas a través del análisis estadístico de multitud de ejemplos de entrenamiento elegidos por un "supervisor". Cada ejemplo contiene las características de la entrada y el valor o etiqueta de salida deseado. Estas técnicas dependen de la disponibilidad de datos etiquetados y no son apropiados en dominios

con un número pequeño de ejemplos, o sin ningún ejemplo. Tampoco son apropiados cuando el entorno cambia tan rápido que los primeros ejemplos dejan de estar relacionados con los últimos.

En esta tesis el sistema de toma de decisiones propuesto utiliza algoritmos de aprendizaje por refuerzo. Por ello se realiza una breve introducción a este tipo de aprendizaje y, posteriormente, se describe el algoritmo utilizado, Q-learning. Debido a que el agente en este sistema va a relacionarse con otros agentes se van a introducir también algoritmos de aprendizaje por refuerzo para el caso multiagente.

3.4.1. Aprendizaje por refuerzo

El agente que usa el aprendizaje por refuerzo trata de aprender mediante la interacción, cómo comportarse para conseguir un objetivo. Se denomina agente al que aprende y toma las decisiones, y se denomina entorno al medio con el que interactúa y que comprende todo lo que está fuera del agente. El agente y el entorno interactúan continuamente: El agente selecciona acciones y el entorno responde a estas acciones y presenta nuevas situaciones al agente. El entorno y el propio agente también generan recompensas, valores numéricos especiales que el agente intenta maximizar a lo largo del tiempo. A cada instante, el agente calcula una función desde los estados a cada una de las acciones. A esta función se le llama política. Los métodos de aprendizaje por refuerzo especifican cómo el agente cambia su política como resultado de su experiencia. El objetivo del agente es maximizar la cantidad total de recompensas que recibe a lo largo del tiempo [Sutton and Barto, 1998].

En otras palabras, este tipo de aprendizaje es una técnica que permite al agente adaptarse a su entorno a través del desarrollo de una política, la cual determina qué acción debería tomar para cada estado del entorno y conseguir maximizar el refuerzo. El refuerzo define la conveniencia de un estado y puede ser expresado en términos de recompensas o castigos.

El aprendizaje por refuerzo ha sido implementado con éxito en múltiples agentes virtuales y en robots [Isbell et al., 2001], [Martinson et al., 2002], [Bakker et al., 2003], [Ribeiro et al., 2002], [Bonarini et al., 2006], [Thomaz and Breazeal, 2006]. Una de las principales aplicaciones es para que el agente o robot aprenda comportamientos complejos, como secuencias de comportamientos básicos. Para ello utilizan técnicas básicas como el algoritmo de Q-learning y modificaciones de éste. Dichos comportamientos complejos permiten optimizar la adaptación del agente o robot a su entorno.

El algoritmo de aprendizaje por refuerzo denominado Q-learning [Watkins, 1989] se ha convertido en uno de los más usados en la robótica autónoma [Touzet, 2003]. Este algoritmo proporciona una manera simple de aprender cómo actuar de forma óptima en dominios Markovianos. Por ello, antes de describir este algoritmo se va a presentar una breve introducción a los procesos de toma de decisión de Markov.

Procesos de decisión de Markov

Considérese como un sistema responde en el tiempo $t+1$ a la acción a tomada en t por un agente. En el caso más general, esta respuesta puede depender de todo lo que ha pasado anteriormente. En este caso, la dinámica puede ser definida sólo especificando la distribución de probabilidad completa:

$$\Pr\{s_{t+1} = s', r_{t+1} = r \,|s_t, a_t, r_t, s_{t-1}, a_{t-1}, ..., r_1, s_0, a_0\} \tag{3.1}$$

para todo s', r, siendo s' el nuevo estado y r la recompensa recibida, y todos los valores pasados: $s_t, a_t, r_t, ..., r_1, s_0, a_0$, donde s_t, a_t y r_t son el estado, la acción tomada y la recompensa recibida en el tiempo t. Si el sistema tiene la propiedad de Markov, entonces la respuesta del sistema en $t+1$ depende sólo de los valores del estado y la acción en t, en cuyo caso la dinámica del entorno puede ser definida especificando solamente:

$$\Pr\{s_{t+1} = s', r_{t+1} = r \,|s_t, a_t\} \tag{3.2}$$

para todo s', r, s_t, y a_t. En otras palabras, un sistema es de Markov, si y sólo si (3.1) es igual a (3.2). La propiedad de Markov es importante en el aprendizaje por refuerzo porque se asume que las decisiones y los valores son funciones sólo del estado actual. Si un sistema tiene la propiedad de Markov, entonces su dinámica (3.2) nos permite predecir las probabilidades del siguiente estado y el próximo refuerzo esperado, dado el estado y la acción actual [Sutton and Barto, 1998].

Un proceso de toma de decisión que satisface la propiedad de Markov, se denomina un proceso de decisión de Markov (MDP). La teoría de los procesos de decisión de Markov [Howard, 1960], asume que el entorno del agente es estacionario, y como tal, no contiene otros agentes adaptativos.

Algoritmo Q-learning

El objetivo del aprendizaje por refuerzo es aprender una función que proporciona el valor a largo plazo que tiene estar en un estado, conocida como función de valor óptima [Smart and Kaelbling, 2002]. La función de valor óptima del Q-learning se define como:

$$Q^*(s,a) = E\left[R(s,a) + \gamma \max_{a'} Q^*(s',a')\right] \tag{3.3}$$

Esto representa el valor esperado del refuerzo recibido por el agente, debido a haber elegido la acción a en el estado s, llevándole al estado s', y actuar de forma óptima en adelante. El factor de descuento γ $(0 < \gamma < 1)$ define cuánto afectan las recompensas futuras. Un valor bajo de γ significa que se presta poca atención al futuro. Un γ alto significa que las recompensas potenciales futuras tienen una mayor influencia en las decisiones actuales [Humphrys, 1997].

Una vez que se obtiene la función óptima Q, $Q^*(s,a)$, es fácil calcular la política óptima, $\pi^*(s)$, viendo todas las acciones posibles para un estado determinado y seleccionando aquella con el mayor valor:

$$\pi^*(s) = \arg\max_a Q(s,a) \tag{3.4}$$

Una política π define el comportamiento de un agente. Una política determinista $\pi : S \longrightarrow \Pi(A)$ es una función que relaciona, con probabilidad 1, las acciones $a \epsilon A$ que se tienen que tomar, con cada estado $s \epsilon S$. La política óptima es aquella que maximiza la recompensa esperada total.

La función Q se almacena típicamente en una tabla o en redes neuronales, indexada por el estado y la acción. Comenzando con valores arbitrarios, se puede iterativamente aproximar la función Q óptima, tomando como base las observaciones del mundo. Cada vez que el agente ejecuta una acción, se genera una secuencia (s, a, r, s'). Cada valor $Q(s,a)$ de la tabla, se actualiza de acuerdo con [Smart and Kaelbling, 2002]:

$$Q(s,a) = (1-\alpha) \cdot Q(s,a) + \alpha \cdot (r + \gamma V(s')) \tag{3.5}$$

Donde:

$$V(s') = \max_{a \in A} (Q(s',a)) \tag{3.6}$$

se denomina valor del estado s' y es lo mejor que el agente puede hacer desde el estado s'. A es el conjunto de acciones, a es cada acción, s' es el nuevo estado, r es el refuerzo, γ es el factor de descuento y α es la tasa de aprendizaje.

En otras palabras, el valor Q es la recompensa esperada por ejecutar la acción a en el estado s y seguir después la política óptima. El objetivo del Q-learning es estimar los valores Q para una política óptima.

El parámetro de descuento γ define las cotas de los valores Q. En [Humphrys, 1997] se prueba un teorema que establece que:

$$Q_{\text{máx}} = \frac{r_{\text{máx}}}{1-\gamma} \tag{3.7}$$

$$Q_{\text{mín}} = \frac{r_{\text{mín}}}{1-\gamma} \tag{3.8}$$

Por lo que los valores Q aprendidos van a depender de los valores máximos y mínimos de las recompensas r y del valor de γ.

El parámetro α $(0 < \alpha < 1)$, la tasa de aprendizaje, controla cuánta importancia se da a la recompensa más reciente. Un valor cercano a 1, de acuerdo con (3.5), implica que se valora más el resultado de $Q(s,a)$ más reciente. A medida que α decrece, el valor Q acumula una media de todas las experiencias, de forma que, una nueva experiencia que resulte inusual, no va a cambiar mucho el valor Q establecido [Humphrys, 1997].

Finalmente, en la figura 3.3 se muestra un esquema que resume cómo funciona el algoritmo Q-learning. Intervienen tres funciones diferentes: evaluación, memorización y actualización. La situación actual es evaluada para seleccionar la mejor acción, es decir, la acción que prometa mayor refuerzo. La nueva situación, como consecuencia de la acción ejecutada, es calificada por la función de refuerzo. Su criterio para calificar (el refuerzo) es usado por el algoritmo de actualización para ajustar los valores Q [Touzet, 2003].

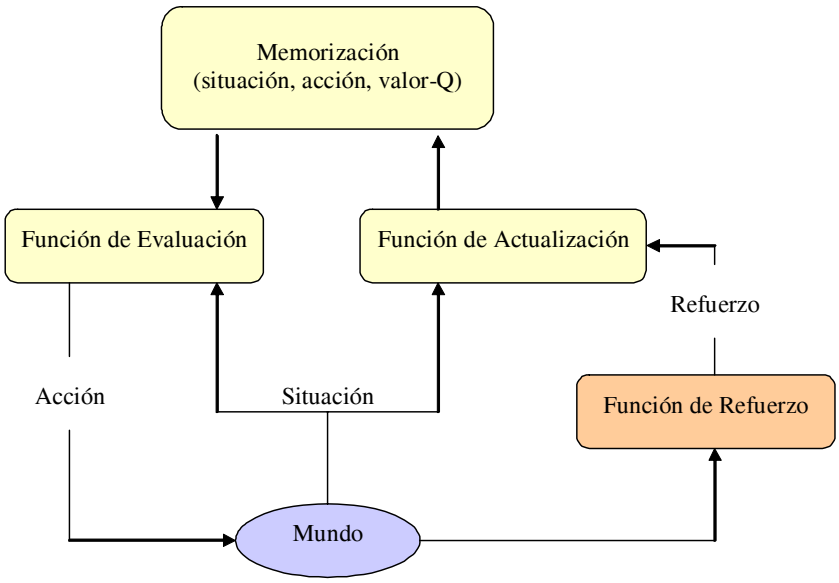


Fig. 3.3: *Descomposición funcional del algoritmo Q-learning [Touzet, 2003]*

Algoritmos de aprendizaje por refuerzo para el caso multi-agente

La teoría de los Procesos de Decisión Markovianos (MDP's), como ya se ha mostrado, asume que el entorno del agente es estacionario y por lo tanto no contiene otros agentes adaptativos [Littman, 1994]. Si el agente está interactuando con otro agente, las recompensas que recibe el agente no dependen sólo de sus propias acciones sino también de la acción del otro agente. Por lo tanto, los métodos de Q-learning individuales no son capaces de modelar las dinámicas de varios agentes aprendiendo simultáneamente en el mismo entorno. Actualmente el aprendizaje multi-agente se ha centrado en el marco teórico de los Juegos Estocásticos (SGs) o Juegos de Markov. Los Juegos Estocásticos parecen ser una extensión natural y poderosa de los dominios de Procesos de Decisión Markovianos a los dominios multi-agente [Yang and Gu, 2004].

La diferencia entre sistemas de un único agente y multi-agente radica en el entorno. En los sistemas multi-agente otros agentes adaptativos hacen que el entorno no sea estacionario, violando la propiedad de Markov. En el marco de aprendizaje de SGs, los agentes intentan maximizar la suma de las recompensas acumuladas. Al contrario que los sistemas de un único agente, en los sistemas multi-agente las acciones conjuntas determinan el siguiente estado y las recompensas de cada agente.

Después de seleccionar las acciones, los agentes pasan al siguiente estado y reciben sus recompensas. El equilibrio de Nash es un concepto importante para la solución del problema de encontrar políticas de aprendizaje óptimas de forma simultánea en presencia de otros agentes. El equilibrio de Nash es una colección de estrategias para cada uno de los jugadores de forma que cada una de las estrategias de los jugadores es la mejor respuesta a la estrategia del otro jugador [Yang and Gu, 2004].

Littman en [Littman, 1994] propuso el algoritmo de aprendizaje denominado Minimax-Q para los juegos de suma-cero, en el que el jugador siempre intenta maximizar su valor esperado, teniendo en cuenta la peor acción que puede elegir su oponente. En los juegos de suma cero el beneficio total para todos los jugadores del juego, en cada combinación de estrategias, siempre suma cero (en otras palabras, un jugador se beneficia solamente a expensas de otros). Los intereses de los jugadores en el juego son opuestos.

El algoritmo de aprendizaje Nash-Q [Hu and Wellman, 1998] es otra generalización del Q-learning a juegos de suma-general, es decir, no necesariamente los jugadores tienen intereses opuestos. Los juegos de suma general incluyen los juegos de suma cero y casos especiales. Este algoritmo de Nash, necesita mantener los valores Q para todos, el propio agente y los otros jugadores. La idea es encontrar un equilibrio de Nash en cada estado, de forma que se obtengan políticas de equilibrio de Nash para la actualización de los valores Q. Un juego puede tener más de un equilibrio de Nash y la recompensa esperada por un jugador puede variar dependiendo del equilibrio considerado. Para garantizar la convergencia, este algoritmo necesita saber si el equilibrio de Nash es único. Existen dos tipos especiales de equilibrio de Nash:

- Un equilibrio adverso es aquel en el que si un agente se desvía del equilibrio, no sólo se daña el agente, sino que ayuda a los otros agentes. No todos los juegos tienen un equilibrio adverso, sin embargo, en un juego de suma cero para dos jugadores, los refuerzos de cada jugador $R_1 = -R_2$ y todos los equilibrios son adversos [Littman, 2001].

- En un equilibrio de coordinación, todos los jugadores alcanzan su recompensa máxima. De nuevo, este tipo de equilibrio no existe siempre, pero en juegos completamente cooperativos con n jugadores $R_1 = R_2 = ... = R_n$, y hay por lo menos un equilibrio de coordinación [Littman, 2001].

Según Littman, aparte de los juegos de suma cero o los juegos completamente cooperativos, para los que los algoritmos de aprendizaje convergente ya se conocen, no se ha demostrado que los juegos de suma general satisfagan las restricciones del teorema de Nash. Por ello, propuso más tarde el algoritmo Q-learning llamado "friend" o "foe" (FFQ) (amigo o enemigo) para juegos de suma-general [Littman, 2001]. Además, en juegos con equilibrio de coordinación o adverso, FFQ converge precisamente a los valores a los que debería converger con Nash-Q. Para ello, se requiere que cada uno de los otros agentes sea identificado como amigo o enemigo, y así sabe qué tipo de oponente debe esperar: amigo (equilibrio de coordinación) o enemigo (equilibrio

adverso). Es mejor ver este algoritmo como dos algoritmos, cada uno aplicado en distintas clases de juegos estocásticos. La clase Amigo consiste en juegos en los cuales, a través de la ejecución del algoritmo, los valores Q de los jugadores definen un juego en el que el refuerzo para cualquier agente bajo esa acción conjunta no es menor que su refuerzo bajo cualquier otra acción conjunta, es decir, que existe un equilibrio de coordinación. La clase Enemigo es aquella en la que los valores Q definen un juego de suma cero, con un equilibrio adverso. Por lo tanto, el Amigo-Q actualiza el valor del estado de manera similar al Q-learning, y el Enemigo-Q como lo hace el Minimax-Q. Comparado con el algoritmo Nash-Q, FFQ no requiere el aprendizaje de las estimaciones de las funciones Q de los oponentes, es fácil de implementar para juegos con varios agentes y garantiza la convergencia siempre que exista un equilibrio adverso o de coordinación.

La mayoría de este tipo de algoritmos de aprendizaje por refuerzo para el caso multi-agente, lo que hacen es extender la función normal del Q-learning de pares de estado-acción $Q(s, a)$ a una función de estados y acciones conjuntas de todos los agentes $Q(s, \vec{a})$. A partir de ahora se va a particularizar para el caso de esta tesis, en el que cada agente puede interactuar sólo con un agente cada vez. Por lo que los valores $Q(s, \vec{a})$ que tienen que ser calculados son $Q(s, a_1, a_2)$, donde a_1 y a_2 pertenecen al conjunto de acciones A_1 y A_2, siendo A_1 el conjunto de acciones del propio jugador y A_2 el conjunto de acciones del oponente.

El valor óptimo de $Q(s, a_1, a_2)$ es la recompensa total acumulada recibida por el agente, cuando ambos agentes ejecutan las acciones (a_1, a_2) en el estado s y después siguen sus estrategias de equilibrio de Nash.

Para aprender estos valores Q, un agente necesita mantener m tablas Q para sus propios valores Q, donde m es el número de estados. Para cada agente k, $(k = 1, 2)$ una tabla Q tiene sus filas correspondientes a $a_1 \in A_1$ y las columnas correspondientes a $a_2 \in A_2$. Por lo tanto el número de valores que cada agente tiene que aprender es $m \times |A_1| \times |A_2|$, donde $|A_1|$ y $|A_2|$ son los tamaños de los espacio de acciones A_1 y A_2. Suponiendo que ambos agentes pueden ejecutar el mismo número de acciones $|A_1| = |A_2| = n$, entonces el número final de valores Q a calcular es $m \times n^2$.

Por lo tanto, en el caso de un juego de suma-general, cada agente tiene que mantener dos tablas Q para cada estado: una para sus propios valores Q y otra para las del otro agente. Esto sería posible si se asume que cada agente puede ver las recompensas que el otro agente recibe, sus acciones anteriores, además de su estado durante el aprendizaje. En el caso de la situación propuesta en esta tesis, esto no es posible debido a las propias limitaciones del juego, es decir, cada agente es capaz de saber la acción que el otro realiza, pero sólo es capaz de conocer su propio estado. En relación a las recompensas recibidas, esto también puede ser sólo conocido por cada uno de los agentes. Por estos motivos, en los algoritmos que van a ser considerados, se va a suponer que la matriz del otro agente con que el interactúa está relacionada con la del agente. Además, cada vez que dos agentes interactúan, cada uno asume que el otro está en su mismo estado, lo cual no siempre va a ser cierto.

Como en el caso del Q-learning, el agente en los sistemas multi-agente actualiza sus tablas Q para un cierto estado después de observar el estado, las acciones tomadas por ambos agentes y la recompensa recibida por el agente. Por lo que el valor $Q(s, a_1, a_2)$ se calcula de la forma siguiente:

$$Q(s, a_1, a_2) = (1 - \alpha) \cdot Q(s, a_1, a_2) + \alpha \cdot (r + \gamma \cdot V(s')) \tag{3.9}$$

Donde $V(s')$ es, de nuevo, el valor del nuevo estado, el cual toma valores distintos dependiendo del algoritmo de aprendizaje por refuerzo, para el caso multi-agente, considerado:

1. Algoritmo Amigo-Q [Littman, 2001]: La idea es que todos los agentes trabajan juntos para obtener la máxima recompensa:

$$V(s') = \max_{a_1 \in A_1} \max_{a_2 \in A_2} (Q(s', a_1, a_2)) \tag{3.10}$$

   En este caso, se busca que los dos obtengan el máximo refuerzo posible debido a una acción conjunta:

$$a_1^* = \arg_{a_1} \max_{a_1 \in A_1} \max_{a_2 \in A_2} (Q(s', a_1, a_2)) \tag{3.11}$$

   Tal y como ya se ha dicho previamente, con este algoritmo ambos agentes obtienen la recompensa máxima. Además, se asume que la matriz Q del oponente es la misma que la del agente, de manera, que lo que es bueno para uno, es bueno para el otro.

2. Algoritmo Enemigo-Q [Littman, 2001]: En este caso, la idea es que los oponentes del agente trabajan juntos para minimizar la recompensa del agente y la política a seguir es buscar el mal menor:

$$V(s') = \max_{a_1 \in A_1} \min_{a_2 \in A_2} (Q(s', a_1, a_2)) \tag{3.12}$$

   El algoritmo Enemigo-Q actualiza de forma similar al algoritmo minimax-Q, es decir, primero se minimiza sobre las acciones del otro agente y entonces se elige el máximo propio [Shoham et al., 2003]:

$$a_1^* = \arg_{a_1} \max_{a_1 \in A_1} \min_{a_2 \in A_2} (Q(s', a_1, a_2)) \tag{3.13}$$

   En este caso, la matriz Q del oponente es la negada $-Q$, de manera que es como si fuera un juego de suma cero.

Estos dos algoritmos son dos posibles soluciones a la hora de que el agente tenga que aprender a cómo comportarse mientras está interactuando con otro agente. Además de estos dos algoritmos, se va a considerar el siguiente:

3. Algoritmo Media-Q: Con este algoritmo en lugar de considerar una acción concreta del otro agente, se calcula el valor medio de cada fila, reduciendo la tabla a un vector de valores medios:

$$V(s') = \max_{a_1 \in A_1} \left(\sum_{a_2 \in A_2} Q(s', a_1, a_2) \Big/ n \right) \tag{3.14}$$

De manera que el agente escogerá la acción a_1 que maximice dicho vector:

$$a_1^* = \arg_{a_1} \max_{a_1 \in A_1} \left(\sum_{a_2 \in A_2} Q(s', a_1, a_2) \Big/ n \right) \tag{3.15}$$

Este algoritmo va a considerar el valor de cada una de sus acciones como el valor medio del resultado de interactuar con otros agentes.

Exploración vs explotación

En [Watkins and Dayan, 1992] se demostró que si cada par estado-acción (s, a) es visitado un número infinito de veces, entonces las tablas de Q-learning convergen a un único conjunto de valores $Q(s, a) = Q^*(s, a)$ que definen una política óptima. Después de la convergencia, el agente maximizará su refuerzo esperado total si siempre escoge la acción con el valor Q^* más alto. Esto es la política óptima, tal y como se describió en (3.4).

En la práctica, como no es posible visitar cada par (s, a) un número infinito de veces, y entonces explotar el conocimiento adquirido, sólo se pueden aproximar los valores Q. Se podría hacer una gran exploración aleatoria y luego explotar las mejores opciones, pero este método tarda mucho tiempo en centrarse en las mejores acciones [Humphrys, 1997]. El principal problema es el decidir cuando se deja de explorar y se comienza a explotar. Si el agente comienza a explotar acciones antes de tiempo puede ser que se exploten acciones cuyo valor Q era alto, pero no el máximo. Esto puede ser debido a que no se exploraron todas las acciones existentes un número suficiente de veces. Como consecuencia, el agente no seguirá la política óptima. En lugar de esto, es mejor utilizar un método que combine la exploración con la explotación. La idea es empezar con una exploración alta e ir reduciéndola hasta que finalmente sólo se exploren aquellas acciones que han funcionado bien anteriormente.

Uno de los métodos que se utilizan es conocido como ϵ-*greedy*, es decir, ϵ-codicioso. Este método consiste en comportarse de forma codiciosa, es decir, explotando las acciones que tienen un valor Q alto, la mayoría del tiempo, aunque una vez cada cierto tiempo, con probabilidad ϵ, selecciona una acción al azar.

Aunque la selección de acciones según el método ϵ-codicioso es una forma efectiva y conocida de equilibrar la exploración con la explotación, un inconveniente es que cuando explora, elige las acciones de manera totalmente aleatoria. Esto significa que tiene la misma probabilidad de ser elegida la acción que obtuvo peores resultados, que la mejor acción. La solución sería variar la probabilidad de las acciones como una función de los valores Q estimados. Para ello se utiliza el método *Softmax* que usa la distribución de Boltzmann [Sutton and Barto, 1998]. Dado un estado s, el agente ejecuta la acción a con probabilidad:

$$P_s(a) = \frac{e^{\frac{Q(s,a)}{T}}}{\sum_{b \in A} e^{\frac{Q(s,b)}{T}}} \tag{3.16}$$

Nótese que $e^{\frac{Q(s,a)}{T}} > 0$ tanto si el valor Q es positivo o negativo y que $\sum_a P_s(a) = 1$. La temperatura T controla la cantidad de exploración, es decir, la probabilidad de ejecutar acciones distintas a las que tienen el mayor valor Q. Si T es alto, o si todos los valores Q son iguales, se elegirá una acción aleatoria. Si T es bajo y los valores Q son diferentes, esto hará que se tienda a elegir la acción con el valor Q más alto.

4. EMOCIONES

4.1. ¿Qué son las emociones?

La palabra "emoción" se deriva de la palabra latina *emovere* que significa remover, agitar o excitar.

La respuesta a la pregunta de qué son las emociones no se encuentra en un sólo libro. A pesar de ser un concepto muy cotidiano, no existe una definición clara y única de qué son y para qué sirven las emociones. A continuación se van a exponer algunas de las definiciones encontradas.

Por ejemplo, Frijda afirma que las emociones son parte de una provisión para asegurar la seguridad y satisfacción de los objetivos del sistema. Un sistema independiente no debería esperar a que alguien lo mantenga, lo ayude, etc. En su trabajo se expone que los programas de acción emocional tienden a dominar a los no emocionales. Las emociones provocan la interrupción de la actividad actual por miedo, deseo, etc [Frijda and Swagerman, 1987].

Por otro lado, Ortony y sus colaboradores, propusieron que las emociones se producen a través de procesos cognitivos y que por lo tanto van a depender de la interpretación de cada uno. Esta propuesta fue denominada *Teoría de la emoción basada en evaluación*. Este modelo asume que las emociones ocurren debido a una reacción valorada (positiva o negativamente) de situaciones consistentes en eventos, agentes u objetos. Por ejemplo, la misma situación puede provocar en distintos agentes emociones diferentes, como por ejemplo en un partido de fútbol. Además, algunas emociones (ej.disgusto) son menos cognitivas que otras (ej.vergüenza) [Ortony et al., 1988].

Oatley, define la emoción como un estado normalmente causado por un evento de importancia para el sujeto. Típicamente esto incluye (a) un estado mental consciente con una calidad reconocible de sentimiento y dirigido hacia algún objeto, (b) una perturbación corporal de alguna clase, (c) expresiones reconocibles de la cara, tono de voz y gestos, (d) una disposición para ciertos tipos de acciones [Oatley, 1996].

Para Scherer, las emociones son una secuencia de cambios sincronizados e interrelacionados en los estados de todos los subsistemas del organismo, en respuesta a la evaluación de un estímulo externo o interno que es relevante para las prioridades del organismo [Scherer, 1998].

Edmund Rolls sostiene que las emociones son estados provocados por refuerzos (recompensas o castigos) [Rolls, 2003]. Las emociones pueden ser provocadas por la entrega, la omisión o la terminación de un estímulo de recompensa o castigo [Rolls, 2005].

Todas las definiciones dadas hasta este momento provienen del área de la psicología. Otras definiciones, como las que se muestran a continuación, pueden ser encontradas desde el punto de vista neurocientífico.

Por ejemplo, Antonio Damasio afirma que la emoción es la combinación de un proceso evaluador mental, simple o complejo, con respuestas a dicho proceso. La mayoría de estas respuestas están dirigidas hacia el cuerpo, que producen un estado corporal emocional, pero también hacia el mismo cerebro, que producen cambios mentales adicionales [Damasio, 1994].

Más recientemente, Fellows define las emociones como patrones de neuromodulaciones (se refiere a las acciones en células nerviosas de una gran familia de sustancias llamadas neuromoduladores, que incluye la dopamina, norepinefrina y serotonina), que afectan a áreas del cerebro involucradas en todos los niveles de funciones desde control motor de bajo-nivel a planificación y cognición de alto-nivel. Por lo tanto se espera que los reflejos estén menos afectados por las emociones que la planificación [Fellows, 2004].

Todas estas definiciones del concepto emoción son sólo un ejemplo para mostrar la evidente falta de acuerdo entre la comunidad científica con respecto al tema emocional. No sólo las definiciones de emociones son distintas dependiendo de si se tratan desde un punto de vista psicológico o neurofisiológico sino que se carece de un acuerdo a la hora de referirse a términos como "humor", "sentimiento", "afecto", etc. Aún así parece existir un cierto acuerdo en decir que las emociones provienen de un proceso evaluador y que están relacionadas con el conocimiento.

Esta tesis doctoral va a estar centrada principalmente en las funciones de las emociones. Se dará una visión general del papel de las emociones en el comportamiento humano desde el punto de vista científico, para justificar su introducción en la robótica.

4.2. El papel de las emociones en los seres vivos

Existen múltiples estudios que ponen en evidencia que las emociones influyen en muchos mecanismos cognitivos como la memoria, la atención, la percepción y el razonamiento [Lewis, 2004], [Gadanho, 1999], [Picard, 1998] y [Rolls, 2005]. Además, las emociones juegan un papel muy importante en la supervivencia, la interacción social y el aprendizaje de nuevos comportamientos. Las emociones y su expresión están más desarrolladas en las especies sociales, siendo los humanos los más emocionales y expresivos de todos. Desde el punto de vista funcional, las respuestas emocionales acercan al animal a aquellas cosas que favorecen su supervivencia y les motiva a evitar aquellas circunstancias que van en decremento de su bienestar [Breazeal, 2003].

Más aún, Lisetti en [Lisetti, 1999] enumera algunas de las funciones de las emociones para resaltar su importancia en el comportamiento inteligente: organización de memoria y aprendizaje; percepción; categorización y preferencia; generación de

objetivos; evaluación y toma de decisiones; planificación de estrategias y determinación de prioridades; atención; motivación y actuación; intención; comunicación; y aprendizaje.

Según Lewin, hasta mitad de los 90, la influencia del filósofo francés Renée Descartes en la cultura occidental, hizo pensar que la relación entre emociones y conocimiento era antagónica. Esto llevó a un énfasis del uso de la lógica pura en el desarrollo de la Inteligencia Artificial como los programas de razonamiento y los sistemas expertos basados en reglas [Lewin, 2001]. Los argumentos del papel positivo de las emociones humanas en el conocimiento ganaron importancia cuando Antonio Damasio publicó *El Error de Descartes* [Damasio, 1994]. Él encontró evidencias de que el daño producido sobre el sistema emocional del cerebro causaba que la persona tomase decisiones incorrectas a pesar de tener intactas las habilidades del razonamiento lógico. Damasio demostró que personas cuyos centros emocionales estaban dañados podrían tener las facultades del razonamiento tradicional intactos, pero que no podían tomar decisiones apropiadas. Esta prueba convenció a un gran número de investigadores de robótica e inteligencia artificial para explorar el posible papel de las emociones en sus sistemas.

Distintos estudios neurocientíficos, entre ellos los de Damasio, que serán descritos posteriormente, demostraron en los 90 la existencia del llamado sistema límbico o cerebro emocional. El sistema límbico es la porción del cerebro situada inmediatamente debajo de la corteza cerebral y que comprende centros importantes como el tálamo, el hipotálamo, el hipocampo y la amígdala cerebral. Según Damasio y otros científicos, en el ser humano, éstos son los centros de afectividad, es allí donde se procesan las distintas emociones. El sistema límbico está en constante interacción con la corteza cerebral y esto es lo que explica que podamos tener control sobre nuestras emociones. Los lóbulos prefrontales y frontales juegan el papel de gestor de nuestras emociones, asumiendo dos importantes tareas:

- En primer lugar, moderan nuestras reacciones emocionales, frenando las señales del sistema límbico.
- En segundo lugar, desarrollan planes de actuación concretos para situaciones emocionales. Mientras que la amígdala del sistema límbico proporciona los primeros auxilios en situaciones emocionales extremas, el lóbulo prefrontal se ocupa de la delicada coordinación de nuestras emociones.

Este papel de gestor de emociones se puso de manifiesto en los estudios de Damasio con pacientes que habían sufrido la destrucción total o parcial del lóbulo frontal. Esto ocasionaba un aplanamiento generalizado de la vida afectiva del paciente. Ante una decisión simple, el paciente procede a evaluar una gran cantidad de alternativas racionales, no reconociendo además, la relación entre opciones peligrosas y malos sentimientos, por lo que repite decisiones equivocadas en lugar de aprender de los errores. Como consecuencia, la vida personal y social se destruyen.

Damasio sugiere que el cerebro de estos pacientes carece de las "marcas somáticas", que son cambios fisiológicos producidos por una emoción, que asocian ciertas decisio-

nes con sentimientos positivos o negativos. Estos sentimientos recortarían la búsqueda mental dirigiendo a la persona lejos de las alternativas asociadas a los malos sentimientos. Estas marcas son lo que llamamos pensamientos subjetivos, corazonadas o intuición.

Cabe destacar que a pesar de la calidad del trabajo de Damasio en el campo de las emociones, recientemente la idea de la existencia de un centro emocional en el cerebro ha sido científicamente desechada [Fellows, 2004]. En cambio, se afirma que sí existen áreas específicas del cerebro involucradas con emociones determinadas [LeDoux, 2002], aunque ninguna de estas áreas puede ser denominada "centro emocional". Desafortunadamente, según Fellows, la idea de "centros emocionales" todavía prevalece en el pensamiento de muchos en inteligencia artificial, psicología y filosofía.

4.3. Clasificación de emociones

Al igual que no existe una definición única del concepto emoción, tampoco existe un acuerdo en relación a la clasificación del tipo de emociones. Aún así, algunos autores realizan esta clasificación basándose en distintos tipos de procesos mentales.

Por ejemplo, en la clasificación de las emociones dada por Damasio [Damasio, 1994], éste pone de manifiesto la existencia de dos procesos: un proceso reactivo, por el cual se generan las emociones primarias y un "proceso mental", que da lugar a las emociones secundarias:

- Emociones primarias (como huir de un gran obstáculo, huir de un gruñido, etc.): el estímulo apropiado activa la amígdala y genera respuestas internas, musculares, viscerales y respuestas a los núcleos neurotransmisores y al hipotálamo.
- Emociones secundarias (dolor por la pérdida de un ser querido, etc.): el estímulo puede ser procesado directamente a través de la amígdala, pero ahora también es analizado en el proceso del pensamiento y puede activar la corteza frontal.

La idea de la existencia de estos dos procesos, reactivo y deliberativo, es compartida también por otros autores. Como ya se ha mencionado, Edmund Rolls sostiene que las emociones son estados provocados por refuerzos (recompensa o castigo), de manera que nuestras acciones estarán dirigidas a obtener recompensas y evitar castigos. Por lo que seleccionar entre las recompensas disponibles con sus costos asociados y evitar los castigos con sus costos asociados, es un proceso que puede tener lugar tanto implícitamente (inconscientemente) como explícitamente, a través de la creación de planes a largo plazo. Según esto, distingue entre tres tipos de respuesta ante una entrada: reflejas (a nivel de la médula espinal), respuestas de comportamiento implícito y respuestas de comportamiento explícito.

En la figura 4.1 se resumen algunas de las emociones asociadas con diferentes planes de refuerzo. La intensidad aumenta continuamente desde el centro del diagrama. La clasificación creada por los distintos planes de refuerzo consiste en: la presentación de un refuerzo positivo (S+), la presentación de un refuerzo negativo (S-), la omisión

de un refuerzo positivo (S+) o la terminación de un refuerzo positivo (S+ !), y la omisión de un refuerzo negativo (S-) o la terminación de un refuerzo negativo (S- !)[Rolls, 2003].

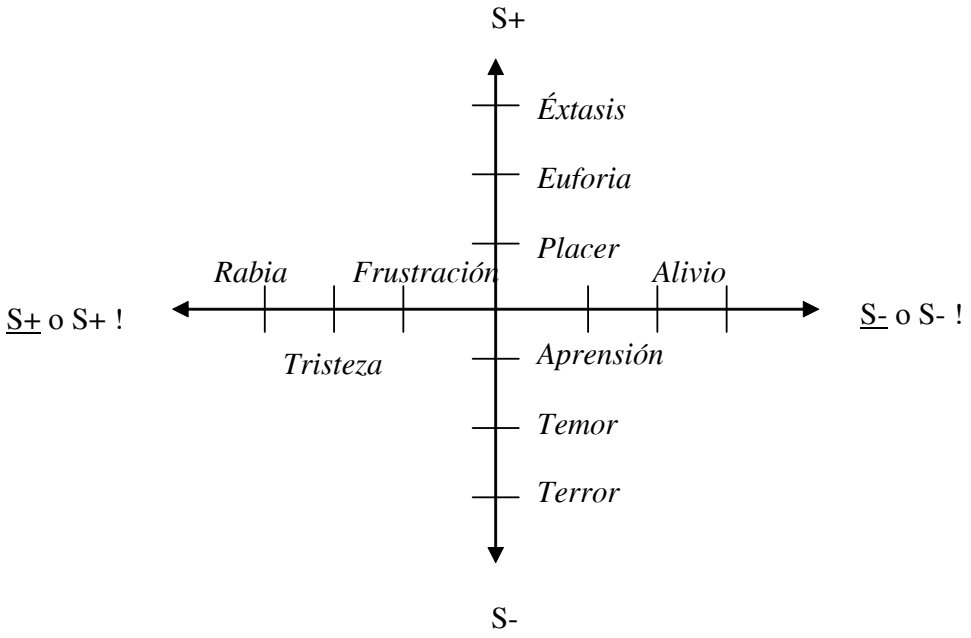


Fig. 4.1: *Algunas de las emociones asociadas con diferentes planes de refuerzo según Rolls.*

Aaron Sloman también propone una clasificación de emociones basándose en una arquitectura que consta de tres procesos: procesos reactivos, procesos deliberativos y procesos reflexivos [Sloman, 2003].

En los procesos reactivos, cada organismo tiene un almacén de planes preestablecidos para hacer frente a la mayoría de situaciones.

El nivel deliberativo se requiere cuando la historia evolutiva y las oportunidades de entrenamiento no dan una variedad suficientemente extensa de planes, que pueden ser aprendidos de forma segura. Este proceso requiere un mecanismo de razonamiento denominado: "Qué pasa-si". Aquí, en lugar de que el objetivo active inmediatamente comportamientos en el nivel reactivo, puede activar un planificador de comportamientos en el nivel deliberativo.

Por último, el nivel reflexivo o de meta-administración, provee de capacidades de auto-monitorización, auto-evaluación y auto-control, incluyendo el control de la atención y procesos del pensamiento. Según el autor, este último proceso es necesario y su objetivo es hacer al sistema interno lo que los otros procesos hacen al entorno. Es muy importante que este proceso funcione en paralelo con los otros porque, por ejemplo, mientras estamos planificando, podemos darnos cuenta repentinamente de que estamos yendo en círculos. Funcionaría como el encargado de toda la arquitectura.

Por lo tanto, Sloman hace la siguiente clasificación de las emociones basada en esta arquitectura: *Emociones primarias*, que dependen sólo del proceso reactivo, como disgustarse por algún olor, aterrorizarse por la visión de un objeto amenazante, etc. *Emociones secundarias*, que dependen del mecanismo deliberativo, las cuales pueden ocurrir durante la planificación, durante reflexiones de acciones pasadas, etc., y los resultados pueden ser varias clases de ansiedad, alivio, temor, placer. Por último, las *emociones terciarias* que dependen del proceso reflexivo. Éste puede incluir estados tales como sentirse avergonzado, humillado, orgulloso, etc. Sloman además dice que estas son las emociones típicamente humanas y la mayoría de ellas implican interacciones sociales.

Por otro lado, en los primeros trabajos de Ortony se propuso un modelo referido normalmente como OCC [Ortony et al., 1988]. En estos trabajos se propuso un esquema que acomodaba un amplio rango de emociones basándose en la *Teoría de la emoción basada en evaluación* (figura 4.2).

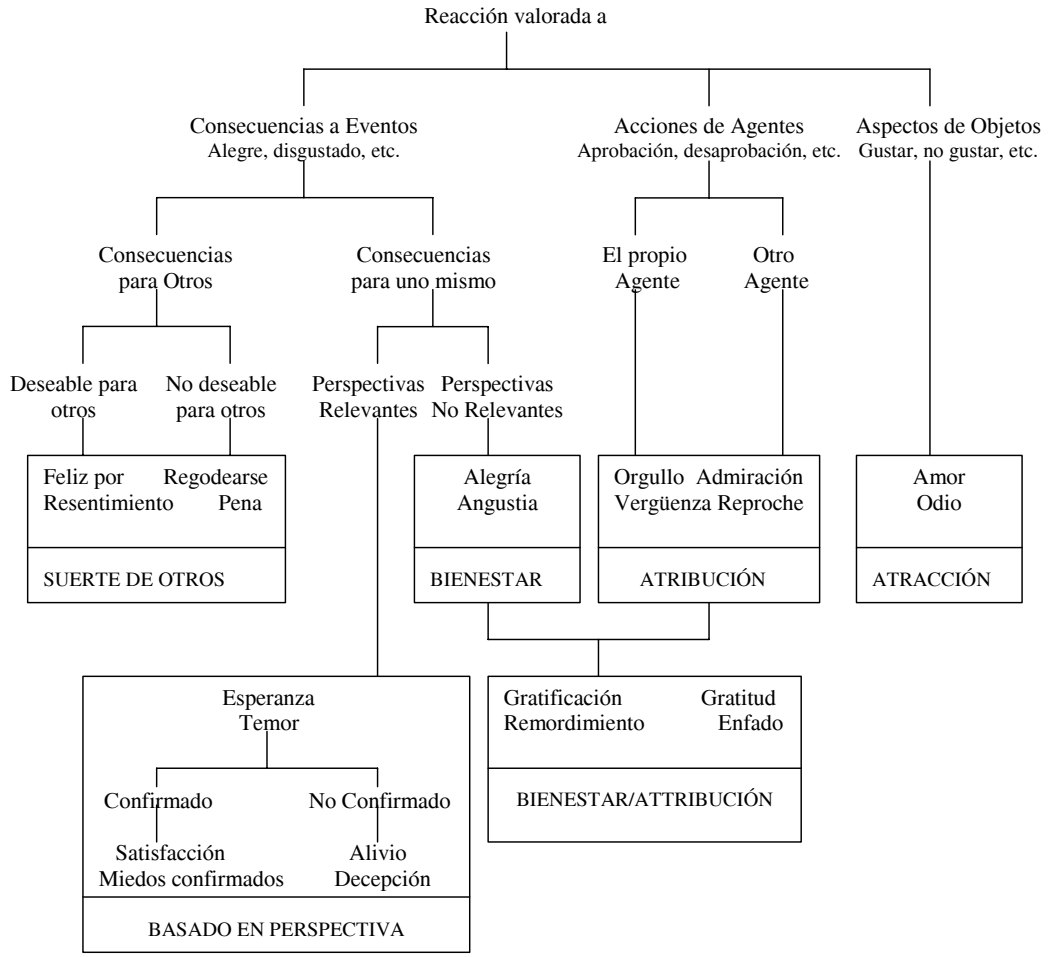


Fig. 4.2: *Estructura global de los tipos de emociones según Ortony et al (1988)*

Posteriormente, en un trabajo en el cual Ortony analizaba la construcción de agentes creíbles, pensó que quizás en lugar de considerar todo el modelo completo, valdría la pena reducir algunas de las categorías originales a varias especializaciones distintas positivas y otras tantas negativas de dos tipos básicos de reacciones afectivas, las positivas y las negativas:

- Reacciones positivas:

  1. Debido a que algo bueno ha pasado (alegría)
  2. Por la posibilidad de que pase algo bueno (esperanza)
  3. Debido a que algo que temíamos no ha pasado (alivio)
  4. Por un acto bueno que hemos hecho (orgullo)
  5. Por un acto bueno que ha hecho otro (gratitud, admiración)
  6. Debido a que uno encuentra a alguien/algo atractivo (amor, gustar)

- Reacciones negativas:

  1. Debido a que algo malo ha pasado (tristeza)
  2. Por la posibilidad de que pase algo malo (miedo)
  3. Debido a que algo bueno que esperábamos no ha pasado (decepción)
  4. Por un acto malo que hemos hecho (vergüenza)
  5. Por un acto malo que ha hecho otro (enfado)
  6. Debido a que uno encuentra a alguien/algo no atractivo (odio, antipatía)

La primera entrada de cada grupo de seis son las reacciones no diferenciadas (positivas o negativas). Las cinco entradas restantes son las especializaciones: las dos primeras basadas en el objetivo (2)(3), la tercera y cuarta basadas en normas de conducta (4)(5) y la última entrada, basada en el gusto (6). De acuerdo con sus conclusiones, estas categorías tienen suficiente capacidad productiva, para dotar a cualquier agente afectivo con el potencial para una rica y variada vida emocional [Ortony, 2003].

4.4. Emociones en Robótica

4.4.1. Introducción

Hasta ahora hemos estado hablando de las emociones y de su influencia en el comportamiento humano, pero ¿Por qué se plantea la posibilidad de crear robots con emociones? Quizás nos preguntemos si esto es realmente necesario, si el hecho de que estén dotados de emociones les hará más inteligentes o más productivos, que al fin y al cabo es lo que nos interesa de un robot.

En Robótica e Inteligencia Artificial una de las principales áreas de investigación es la de tratar de imitar el comportamiento humano y el funcionamiento de la mente, basándose para ello en los estudios realizados por psicólogos sobre cómo funciona la mente y cuáles son los factores que intervienen en la toma de decisiones de cada individuo. De hecho, existen evidencias de que dos acciones altamente cognitivas dependen no sólo de leyes y reglas, sino de emociones: la toma de decisiones y la percepción [Picard, 1998].

A principios de los años sesenta, el precursor de la inteligencia artificial, Herbert Simon, sostenía que era necesario que los modelos cognitivos dieran cuenta de las emociones para aproximarse a la mente humana [LeDoux, 1996]. Pero sólo hasta hace pocos años la llamada inteligencia emocional fue tenida en cuenta como un factor esencial para comprender y valorar el comportamiento humano. Según Mayer y Salovey: "La inteligencia emocional es la capacidad de comprender los sentimientos y las emociones propias y de los demás, y de utilizar esta información como guía para el pensamiento y las acciones de uno mismo" [Mayer and Salovey, 1993].

Varios psicólogos y otros autores interesados en la Inteligencia Artificial han reconocido que las emociones son esenciales en el pensamiento y comportamiento inteligente, alterando las prioridades y generando interrupciones en el comportamiento [Frijda and Swagerman, 1987]. De hecho, Minsky en *La sociedad de la mente* en 1986, concluyó que, "La cuestión no es si las máquinas inteligentes pueden tener alguna emoción, sino si las máquinas pueden ser inteligentes sin ninguna emoción" [Minsky, 1986].

4.4.2. ¿Por qué necesitan emociones los robots?

En esta sección se va ha realizar una revisión de las distintas opiniones de varios expertos sobre el porqué las máquinas necesitan emociones.

Rosalind Picard en su libro *Los ordenadores emocionales* [Picard, 1998], realiza un completo estudio sobre este tema basándose en las investigaciones de varios psicólogos incluyendo a Damasio. Picard propone criterios de diseño para un ordenador que pueda expresar emociones, e incluso establece que un ordenador tiene emociones si posee los cinco componentes siguientes que están presentes en los sistemas emocionales de las personas sanas:

1. El comportamiento del sistema parece generado por las emociones. Actualmente este componente es el más implementado en las máquinas con el fin de parecer más naturales o creíbles.

2. El sistema tiene reacciones rápidas "primarias" (en el sentido de Damasio) en ciertas situaciones.

3. El sistema puede generar emociones de forma cognitiva, razonando sobre situaciones (serían las emociones secundarias de Damasio).

4. El sistema puede tener experiencia emocional. Quiere decir que puede percibir el propio estado emocional, lo que nos lleva al problema de la conciencia.

5. Las emociones del sistema se relacionan con otros procesos que imitan las funciones cognitivas y físicas humanas, por ejemplo: la memoria, la percepción, la toma de decisiones, etc.

Estos puntos, según Picard, sirven para distinguir si una máquina tiene emociones, pero no dicen nada sobre por qué debe tenerlas. Más adelante, en [Picard, 2003], da cuatro razones para dotar a las máquinas de ciertas habilidades emocionales. Un objetivo es construir robots y personajes sintéticos que puedan emular a humanos y animales. El segundo objetivo es hacer máquinas inteligentes, aunque es imposible encontrar una definición mundialmente aceptada de la inteligencia de una máquina. Un tercer objetivo es tratar de entender las emociones humanas mediante su modelización. Aunque estos tres objetivos son importantes, el principal es hacer a las máquinas menos frustrantes para interactuar con ellas, es decir, facilitar la interfaz máquina-usuario.

Es decir, parece que para Picard la implementación de emociones en máquinas es esencial para los humanos, no para las propias máquinas. Es razonable pensar que las emociones facilitan la interacción entre máquinas y robots, pero no es su única misión. Arkin, de hecho, considera además la importancia de las emociones para la propia máquina, en este caso robots, además de su importancia en la interacción con el hombre. En [Arkin, 2004] resume en dos los papeles cruciales que las emociones y motivaciones tienen en robótica:

- Supervivencia: las emociones sirven como mecanismo para completar la autonomía y ayudan a los sistemas naturales a hacer frente al mundo.

- Interacción: muchos robots, que son creados para funcionar en proximidad con gente, necesitan ser capaces de relacionarse con ellos de una manera predecible y natural. Por lo que para interaccionar con la gente de una manera efectiva y eficiente, es útil para ellos reaccionar de una forma que sea familiar y cómoda para los humanos.

Como se puede apreciar, este es un punto de vista totalmente diferente, las emociones son básicas para el robot, tanto para su supervivencia, como para poder relacionarse con humanos.

Cañamero sostiene que las emociones favorecen la adaptación y autonomía de los agentes biológicos. Por este motivo considera, al igual que Arkin, que podría ser muy útil explotar ese papel de las emociones para diseñar mecanismos para un robot autónomo. Esto implica que el robot tiene algún objetivo interno o motivación que dirige su comportamiento [Cañamero, 2003]. Además afirma que las emociones juegan un papel muy importante en la producción de acciones en robots autónomos:

- El hecho de que las emociones estén relacionadas con objetivos generales, más que con patrones de comportamiento, provoca que las emociones permitan generar comportamientos más flexibles, ricos y variados.

- Pueden ser conceptualizadas como mecanismos de control de segundo orden. Éstos constantemente monitorizan el entono interno y externo para detectar y responder a "amenazas" potenciales de diferentes clases y por lo tanto, interrumpir o continuar el comportamiento que esté siendo llevado acabo.

- Modifican/amplifican la motivación, produciendo cambios en las prioridades motivacionales o de objetivos para tratar ciertos tipos de eventos importantes más eficientemente.

- Pueden constituir también factores motivacionales o "sistemas de valor", que afectan a la selección de objetivos y de comportamientos dirigidos a objetivos.

Tal y como se muestra en estos puntos, el papel de las emociones va mucho más allá de facilitar la interacción humano-robot. Las emociones parecen ser esenciales para la autonomía de un robot. Sin embargo, Cañamero opina que la inclusión de elementos emocionales en las arquitecturas de los robots no los va a hacer más valiosos *per se*. Por el contrario, somos los investigadores los que deberíamos ser capaces de demostrar precisamente, que las emociones mejoran el comportamiento y las capacidades de interacción de nuestros robots [Cañamero, 2005].

Bellman coincide en cierto sentido con Cañamero en los motivos por los que debemos considerar las emociones en robots. Ante la pregunta de por qué no dejar las emociones sólo como un fenómeno humano, o animal, el autor responde que las emociones permiten que los animales que las poseen sobrevivan mejor que los que no las tienen. Ya que las emociones son críticas en los organismos biológicos, se podría creer que algún tipo de analogía a las habilidades emocionales es necesaria para los robots, siempre que se quiera conseguir un comportamiento inteligente e independiente en un entorno real [Bellman, 2003].

Dando un paso más, Ortony explica que los robots necesitan emociones por la misma razón que los humanos las necesitan. Una de las funciones fundamentales de las emociones es que son un requisito para establecer recuerdos a largo plazo. Una segunda función importante es que las emociones proveen de oportunidades para aprender, desde simples formas de aprendizaje por refuerzo, a planificaciones conscientes y complejas. Las emociones también tienen importantes consecuencias en la localización de la atención. El miedo tiende a centrar la atención en detalles locales, mientras que bajo condiciones de afecto positivo, la gente tiende a centrarse en un campo más amplio [Ortony et al., 2005].

Por otro lado, para Fellows una de las principales funciones de las emociones es la de conseguir una comunicación multi-nivel de información simplificada, pero de alto impacto. Por ejemplo, un grito es extremadamente pobre en información (no dice nada acerca de la causa de la alarma) pero su impacto en las personas que nos rodean es alto. En su trabajo defiende, de la misma manera que Bellman y Cañamero, que ya que los animales tienen emociones en su sentido funcional, los robots podrían ser dotados de características que funcionalmente estén relacionadas con emociones. Por lo tanto, los robots podrían tener "emociones-robot", de la misma

forma que los animales tienen "emociones-animales". Por supuesto, deja claro que las emociones que se podrían implementar en los robots no deberían ser llamadas emociones sino "emociones-robot", ya que sólo es una imitación de la funcionalidad de las emociones humanas y animales. Desde la robótica, este punto de vista funcional puede ser trasladado, por lo menos, en tres dominios "implementables" importantes [Fellows, 2004] [Arbib and Fellows, 2004]:

1. Comunicación: no prestar atención a las expresiones emocionales puede causar desastres.

2. Fuente de movilización, conservación y priorización de comportamientos.

3. Separar estímulo y respuesta. Sin el aburrimiento podríamos estar estancados procesando el mismo estímulo de la misma forma. Sin curiosidad, nunca probaríamos algo nuevo. Curiosidad y aburrimiento (si se aceptan como emociones) cambian la percepción del mundo y la forma en la que la procesamos. Las emociones pueden llegar a ser ellas mismas estímulos y empujar al organismo a actuar.

De nuevo, las utilidades que tienen las emociones en los robots, dadas por Fellows, parecen fundamentales para la supervivencia del robot.

También Gadanho, en su tesis doctoral [Gadanho, 1999], desde el punto de vista de la ingeniería, utiliza las emociones para aumentar la autonomía del robot, no para mejorar su conocimiento acerca de la naturaleza de las emociones. En su trabajo usa las propiedades de las emociones que, según la autora, pueden ser útiles para una criatura artificial y autónoma. Dichas propiedades son, de forma resumida, las siguientes:

- Fuente de motivación, donde motivación significa cualquier cosa que controla el centro de atención y orienta el razonamiento actual del agente.

- Control de atención. Las emociones influyen en la percepción, centrando la atención del agente en las características más relevantes, para resolver el problema inmediato.

- Fuente de refuerzo. Las emociones son frecuentemente asociadas con sentimientos agradables o desagradables que pueden ser usados como refuerzo.

- Memorias asociadas a emociones.

- Ayuda en el razonamiento. Esta función está basada en las ideas de Damasio, previamente presentadas.

- Las tendencias de comportamiento o incluso las respuestas estereotipadas están normalmente asociadas con ciertas situaciones emocionales.

- Activación fisiológica del cuerpo. Una emoción fuerte está normalmente asociada con una liberación general de energía como previsión a una posible respuesta.

- Soporte en la interacción social. La expresión de las emociones permite a los individuos transmitir a otros, mensajes que son frecuentemente esenciales para su supervivencia.

Es muy interesante observar que estas propiedades de las emociones, parecen ser un resumen de todas las expuestas por los distintos autores previamente.

Una vez que se han expuesto las razones, todas ellas válidas, por las que un robot sí debería tener emociones, se debería tener en cuenta la opinión de Picard al respecto. Picard advierte sobre la implementación en máquinas de las funciones que el sistema emocional humano posee. Según la autora, las computadoras no tienen emociones como los humanos. Ellas podrían sentir y clasificar ciertos eventos físicos como categorías de "sensaciones", pero no experimentarían los sentimientos como los humanos. La metodología de la ciencia es tratar de reducir fenómenos complejos, como las emociones, a una lista de requerimientos funcionales. El reto de muchos investigadores, en la ciencia de computación, es tratar de duplicar éstos en computadores a distintos niveles, dependiendo de las motivaciones de la investigación. Pero se debe tener cuidado a la hora de presentar este reto al público, que piensa que la emoción es la frontera que separa el hombre de la máquina [Picard, 1998].

Sobre este punto, Sloman opina, al igual que Fellows, que debemos distinguir entre las emociones de un adulto, de un niño, de un animal, de un robot, etc. y por lo tanto, se tienen que reemplazar preguntas como: "¿Puede un robot tener emociones?" por "¿Qué clase de emociones puede tener un robot?", por lo que quizás se pueda definir un gran número de clases de emociones, relativas a los distintos tipos de "sujetos". De esta forma no existiría confusión, propia de la ciencia ficción, a la hora de hablar de emociones en robots como la superación de la frontera hombre-máquina [Sloman, 2003].

5. IMPLEMENTACIÓN DE EMOCIONES EN ROBOTS

5.1. Introducción

Han sido muchos los que han implementado modelos emocionales en robots. Pero de la misma forma que existen diversas interpretaciones de las emociones en los seres humanos, cada uno de los investigadores tiene su propia interpretación sobre qué tipo de emociones debe tener un robot y cómo deben ser implementadas.

En esta sección se hará un repaso de algunos de los modelos emocionales implementados en distintos tipos de robots. Se va a comenzar por una revisión de dos arquitecturas que han sido básicas en el desarrollo de esta tesis doctoral. Estas arquitecturas fueron desarrolladas por Lola Cañamero y por Sandra Gadanho. Posteriormente se hará una revisión de varios robots, con arquitecturas que usan emociones, dando más importancia al trabajo desarrollado por Velásquez y Cynthia Breazeal.

5.2. Arquitectura propuesta por Lola Cañamero

Lola Cañamero en uno de sus primeros artículos publicados en este tema, presenta los resultados de la implementación de una criatura simulada, en un mundo virtual de dos dimensiones [Cañamero, 1997]. Este agente es autónomo y sus comportamientos están dirigidos por estados motivacionales y emociones básicas. En aquel momento, el hecho de usar motivaciones internas y estados emocionales para dirigir el comportamiento del agente y su aprendizaje, hizo que su trabajo fuera muy novedoso en relación a otras arquitecturas de toma de decisión.

Los comportamientos motivacionales, tal y como se ha visto en el capítulo 3, se distinguen entre consumatorios (que satisfacen un objetivo y que sólo se pueden ejecutar ante la presencia del estímulo incentivo) y apetitivos o instrumentales (que dirigen hacia el objetivo y lo que hacen es buscar el estímulo incentivo determinado). Un comportamiento sólo puede ser ejecutado si: a) ha sido seleccionado por el estado motivacional/emocional del agente, b) su estímulo incentivo está presente. Estos estímulos incentivos que hacen que se pueda ejecutar un comportamiento, son por ejemplo, la comida para poder comer, o el agua para poder beber.

Cañamero adopta la aproximación homeostática para modelar las motivaciones. Por lo tanto, las motivaciones pueden ser vistas como procesos homeostáticos, los cuales mantienen una variable fisiológica controlada dentro de un cierto rango. Un

detector de errores genera una señal de error, el *drive*, cuando el valor de esa variable no cuadra con su valor ideal. Este *drive* lanza elementos de control para ajustar la variable en la dirección correcta, como comer, beber, o escapar.

Cada motivación tiene un nivel de activación proporcional a la magnitud del error y una intensidad calculada en base a este nivel de activación. La motivación con el nivel más alto se activa y organiza el comportamiento del agente para satisfacer su *drive*. Primero mirará los comportamientos que más pueden contribuir a su satisfacción. Si no se encuentra ninguno, entonces seleccionará una lista de comportamientos que pueden contribuir en mayor grado a la satisfacción del *drive*.

Las emociones están caracterizadas por: un estímulo incentivo; una intensidad proporcional a su nivel de activación; una lista de hormonas que la emoción libera cuando se activa; una lista de síntomas fisiológicos; y una lista de variables fisiológicas que pueden ser afectadas. Las emociones influyen en la toma de decisiones de dos formas. Primero, pueden modificar la intensidad de la motivación actual, dependiendo del efecto de la hormona liberada y como consecuencia la intensidad del comportamiento. En casos extremos pueden evitar la ejecución del comportamiento. Segundo, modifican la lectura de los sensores que monitorizan las variables que las emociones pueden afectar, por lo que alteran la percepción del estado corporal.

Posteriormente, Cañamero en [Cañamero, 2000] y [Cañamero, 2003] explica de forma bastante más detallada la arquitectura propuesta y da más detalles sobre las emociones. Elige un subconjunto de categorías discretas correspondientes a emociones "primarias" o "básicas" que funcionan como mecanismos de monitorización para enfrentarse con situaciones importantes relacionadas con la supervivencia. Estas emociones son:

- Enfado: Es un mecanismo para detener las influencias del entorno mediante la parada de la situación actual. El evento que la activa es el hecho de que el objetivo no está acabado.
- Aburrimiento: Es un mecanismo para detener el comportamiento repetitivo que no contribuye a satisfacer las necesidades del agente. El evento que la activa es la actividad repetitiva e ineficiente.
- Miedo: Es un mecanismo de defensa contra amenazas externas. Se activa ante la presencia de enemigos.
- Felicidad: Es un doble mecanismo. Por un lado, un mecanismo de re-equilibrio lanzado por la consecución de un objetivo. Por otro, un mecanismo de apego, el cual no fue explotado en esta implementación.
- Interés: Es un mecanismo para que el agente establezca una interacción con el mundo. Se activa cuando encuentra algo nuevo.
- Tristeza: Es un mecanismo para detener una relación activa con el entorno cuando el agente no está en condición de conseguir satisfacer una necesidad. El evento que la activa es la incapacidad de llevar a cabo el objetivo.

Estas categorías discretas también tienen, sin embargo, propiedades de valencia (a través de las hormonas que actúan como mecanismos de dolor o placer) y despertar (actividad fisiológica).

Es muy interesante el hecho de que considera a las emociones como un controlador de comportamiento de "segundo orden". Este controlador funciona en paralelo y por encima del control de las motivaciones, permitiendo continuamente a las emociones monitorizar el entorno (tanto interno, como externo) en situaciones en las que la relación del agente con el entorno tiene alguna importancia para los objetivos del agente. Las emociones afectan a la selección de comportamientos de forma indirecta mediante la modificación de los efectos del sistema motivacional.

En [Ávila García and Cañamero, 2002], esta arquitectura de toma de decisiones, aunque sin emociones, fue implementada en robots reales por Ávila-García y Cañamero. Se realiza un estudio comparativo de distintas estrategias de selección de comportamientos, utilizando para ello una plataforma de simulación de robots. La conclusión fue que la estrategia "El ganador se lo lo lleva todo" (WTA) obtuvo en general mejores resultados.

Esta estrategia realiza el siguiente ciclo:

1. Se calcula la motivación ganadora, la de mayor intensidad.

2. Se calcula la intensidad de cada comportamiento ligado, a través de la fisiología, con la motivación ganadora.

3. Se ejecuta el comportamiento con la intensidad más alta.

La ejecución de un comportamiento tiene un impacto en el nivel de variables fisiológicas específicas. Los comportamientos pueden ser activados y ejecutados con distintas intensidades, que dependen de las motivaciones relacionadas con ellos. Es decir, que son comportamientos motivados, por lo que la arquitectura siempre ejecutará el comportamiento que mejor satisfaga la motivación activa más alta.

Este trabajo fue continuado en años posteriores [Ávila García and Cañamero, 2004] y [Ávila García and Cañamero, 2005]. Para modelar las motivaciones, m_i, se adoptó el modelo descrito por la ecuación (5.1) donde d_i son los déficit fisiológicos (la señal de error) y c_i son factores externos (entradas), como estímulos externos o señales incentivas que permiten ejecutar comportamientos consumatorios y por lo tanto satisfacer necesidades corporales.

$$m_i = d_i + (d_i \times \alpha c_i) \tag{5.1}$$

El parámetro α se introdujo para tener en cuenta otros factores como la calidad de los estímulos (sabor de la comida), estados corporales anormales (enfermedad), niveles hormonales, etc. En la figura 5.1 se muestra el sistema de control motivacional para esta arquitectura.

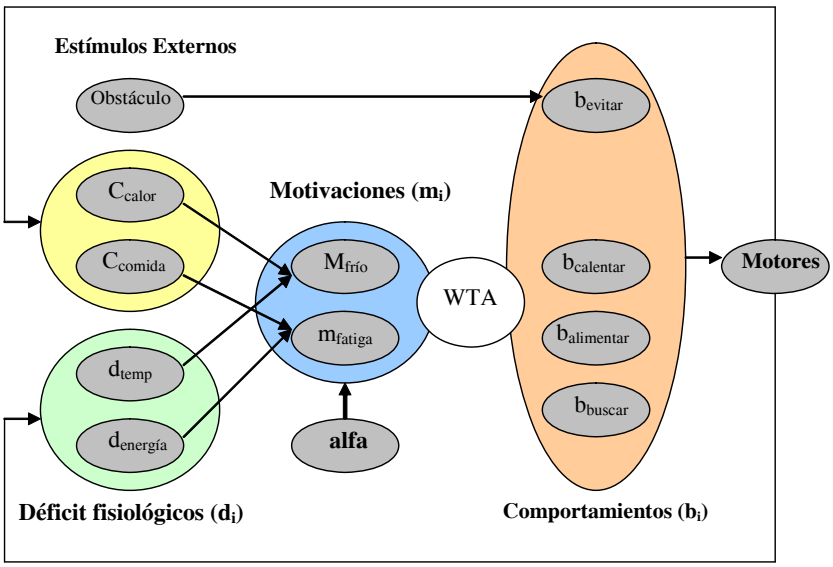


Fig. 5.1: *Sistema de control motivacional para la arquitectura propuesta por Avila-García y Cañamero*

En estos nuevos trabajos, lo interesante es el uso de una realimentación hormonal para modular la selección de comportamientos. Basándose en el artículo original de Cañamero [Cañamero, 1997], se exploran los efectos moduladores de las hormonas en la percepción de los estímulos externos. Para ello se modifica el valor del parámetro α, que determina la importancia del estímulo externo.

En los experimentos, dos robots compiten para sobrevivir en un mismo entorno, uno de ellos está dotado de este sistema hormonal y el otro no. Los resultados demuestran cómo la modulación hormonal de la percepción del estímulo externo, puede adaptar la misma arquitectura a nuevas circunstancias ambientales, donde el robot en lugar de estar sólo en el entorno, debe competir con otro por los mismos recursos. El robot con la modulación hormonal actúa mejor que el que no la tiene. Además exhibe algunos fenómenos de comportamientos emergentes que podrían ser interpretados por un observador externo como comportamientos agresivos/defensivos.

En otra línea paralela de investigación y, de nuevo, como extensión de la arquitectura propuesta en [Cañamero, 1997], Cos-Aguilera junto a Cañamero proponen una arquitectura de aprendizaje y selección de comportamientos. En esta arquitectura existen dos tipos de procesos de aprendizaje: el aprendizaje de las propiedades de los objetos y el aprendizaje de las políticas de comportamiento.

Las propiedades de un objeto se definen como las funcionalidades que el objeto ofrece al agente. El aprendizaje de las propiedades de los objetos fue llevado a cabo en [CosAguilera et al., 2003] y [CosAguilera et al., 2005b]. En esta arquitectura, el robot sabe de antemano qué comportamiento tiene que realizar para satisfacer su necesidad dominante, es decir, para cada motivación hay un comportamiento ligado a él. El problema está en que no todos los objetos sirven para todo, es decir, que no

todos los objetos, por ejemplo, son comestibles, y si se intenta comer un objeto sin esta propiedad, a pesar de haber elegido el comportamiento correcto, el agente no va a satisfacer su hambre. Por este motivo el agente debería conocer de antemano las funcionalidades de todos los objetos.

Estas funcionalidades en [Cañamero, 1997] eran conocidas de antemano, ya que estaban pre-programadas. En esta nueva línea de investigación el agente tiene que aprender qué objetos ofrecen las funcionalidades adecuadas, dependiendo del comportamiento que está tratando de realizar. Para ello utiliza un extractor de propiedades, que se basa en el hecho de que cuando se ejecuta una acción específica con un objeto, el nivel de una o varias variables homeostáticas varía, aumentando o disminuyendo su error.

Los resultados demuestran que, los agentes que están dotados de la capacidad de conocer de antemano que un objeto puede servir para satisfacer una necesidad particular, pueden usar ese conocimiento para decidir si vale la pena interactuar con un objeto en concreto, antes de cualquier interacción. Los experimentos también demostraron que los agentes que ya conocían las funcionalidades de los objetos sobrevivían mucho más tiempo, debido a una mejor política de interacción, de manera que se aumentaban las interacciones satisfactorias.

Por otro lado, el aprendizaje de las políticas de comportamiento fue descrito en [CosAguilera et al., 2005a], el cual estaba basado en un algoritmo de aprendizaje por refuerzo "actor-crítico". Este algoritmo presenta una estructura separada para la selección de acciones (actor) y el aprendizaje (crítico). Sin embargo, ambos son modelados como función del estado motivacional que es función de los *drives* y las funcionalidades del objeto más cercano al agente. El "actor" realiza la selección de los comportamientos calculando la probabilidad, para cada comportamiento, de obtener el máximo refuerzo acumulativo dado el estado motivacional actual. Para ello elige durante un 80 % del tiempo, el comportamiento que haga máxima dicha probabilidad y durante el 20 % del tiempo, un comportamiento de forma aleatoria por motivos de exploración. El "crítico" es el núcleo del sistema de aprendizaje y estima el refuerzo acumulativo resultante de la ejecución del patrón de comportamiento elegido por el actor.

La idea es que, si la ejecución de un comportamiento produjo un refuerzo positivo, esa política debería ser incentivada y a la inversa si el refuerzo fue negativo. El refuerzo es modelado como una valoración del efecto fisiológico provocado por la ejecución de un comportamiento.

En estos experimentos los agentes estaban dotados del conocimiento de las funcionalidades de los objetos. Los resultados mostraron que el efecto de estas funcionalidades en la selección de comportamientos es muy notable. De hecho, si no conocen las funcionalidades de los objetos, aumenta el tiempo de aprendizaje de políticas eficientes, convirtiendo la elección de comportamientos en una selección ciega.

5.3. Arquitectura propuesta por Sandra Clara Gadanho

En la tesis de Sandra Clara Gadanho en 1999 [Gadanho, 1999] y en los posteriores artículos en 2001 y 2002 en colaboración con John Hallam, [Gadanho and Hallam, 2001] y [Gadanho and Hallam, 2002], se investiga cómo las emociones artificiales pueden mejorar el comportamiento de un robot autónomo solitario. Este robot se adapta a su entorno usando un controlador adaptativo que se ajusta utilizando aprendizaje por refuerzo. Las emociones son usadas para influir la percepción, como Cañamero, y para proveer una función de refuerzo en un marco de aprendizaje por refuerzo.

Las características más importantes de las emociones que el modelo propuesto intenta capturar, son las siguientes:

- Las emociones tienen valencias: positivas y negativas
- Las emociones persisten en el tiempo, cambios repentinos entre distintas emociones no deben ser permitidos.
- El que ocurra una emoción depende de la entrada sensitiva y de la historia emocional reciente del sujeto.
- La percepción está coloreada por el estado emocional.
- El estado emocional puede ser neutro o dominado por una emoción. Existe un mecanismo que decide qué emoción es la dominante en cada momento.

El modelo emocional desarrollado, ver figura 5.2, está basado en cuatro emociones básicas: felicidad, tristeza, temor y enfado. Estas emociones fueron seleccionadas basándose en los estudios de Ekman [Ekman, 1992], en los que se afirmaba que éstas, junto al disgusto, son las emociones más universalmente expresadas. Además consideraron que estas emociones eran las más adecuadas y útiles para sus experimentos.

Cuando una emoción está activa, esto es, que su intensidad es significativamente grande, entonces influye en el cuerpo a través del sistema hormonal. El sistema hormonal en este modelo está simplificado. Consiste en tener una hormona asociada con cada sentimiento. La intensidad de un sentimiento no es un valor obtenido directamente de la sensación corporal que la ha originado, sino de la suma de la sensación y del valor hormonal. Los valores hormonales pueden ser, positiva o negativamente, suficientemente grandes para esconder totalmente las sensaciones reales de la percepción que tiene el robot de su cuerpo. Las cantidades de hormonas producidas por cada emoción están directamente relacionadas con su intensidad y su dependencia con los sentimientos asociados. Cuanto más fuerte sea la dependencia con un cierto sentimiento, mayor será la cantidad de hormona producida por la emoción.

Por otra parte, el mecanismo hormonal aporta la competitividad entre las emociones para ganar el control sobre el cuerpo. Además, lo que el robot siente no depende sólo de sus sensaciones sino también de su estado emocional. La emoción dominante

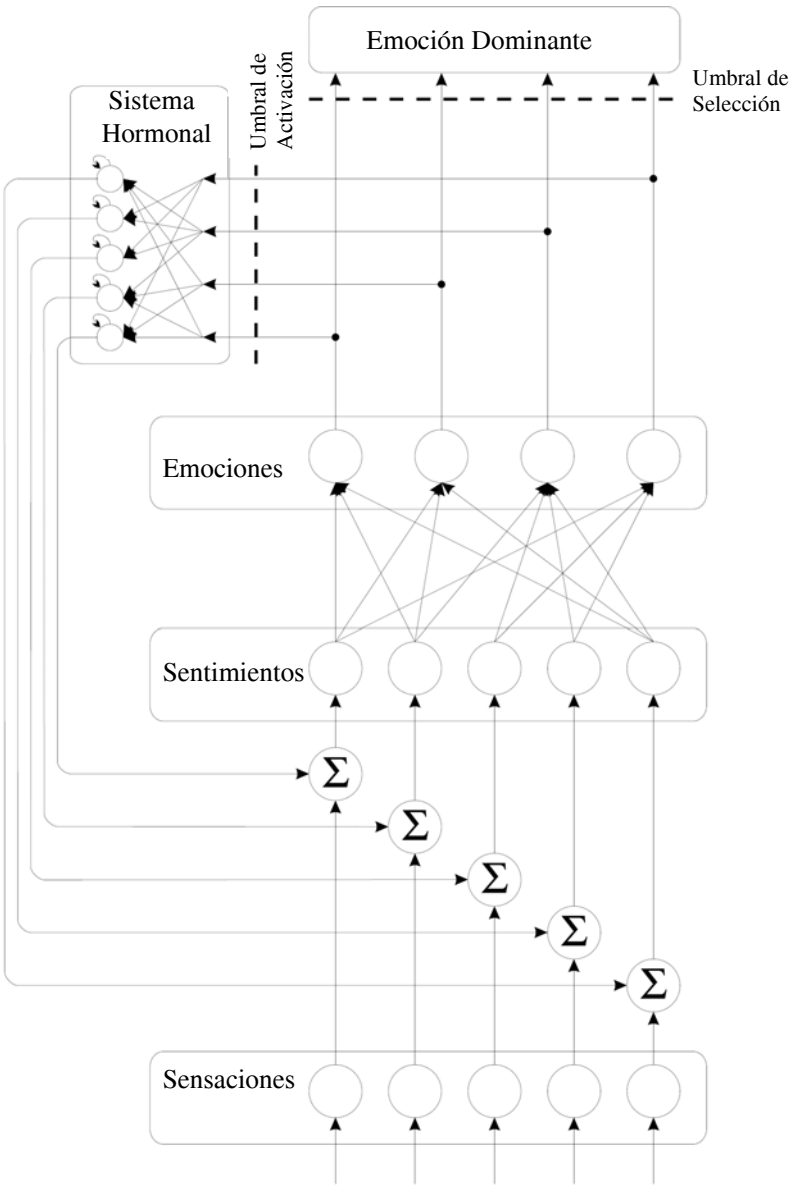


Fig. 5.2: *Modelo emocional propuesto por Gadanho y Hallam*

es la emoción con mayor intensidad, a menos que ninguna intensidad emocional exceda el límite de selección. En ese caso, no existiría emoción dominante y el estado emocional sería neutro. Las emociones están divididas en dos categorías: positivas y negativas. Las que son consideradas como "buenas" son positivas (sólo la felicidad, para el grupo de emociones usado) y las otras son consideradas negativas. Con el objetivo de evaluar el papel funcional de las emociones en el razonamiento, el estado emocional, según Gadanho, debería ser usado para el control de un agente, determinando su comportamiento.

Los experimentos fueron llevados a cabo en un robot simulado, el cual tenía la tarea de obtener energía de fuentes de comida repartidas en el entorno. Esta tarea puede ser traducida a múltiples objetivos, como moverse en el entorno para encontrar diferentes fuentes de energía y, si la encuentra, extraer la energía de la fuente. Además, el robot no debería permanecer quieto en el mismo sitio por mucho tiempo ni chocar con obstáculos.

Para tener un estado emocional del robot compatible con su tarea, la dependencia de las emociones en los sentimientos es tal que:

- El robot está *feliz* si no hay ningún problema en la situación actual. Estará particularmente contento si ha estado usando los motores mucho o si está en el proceso de obtener energía.
- Si el robot tiene muy baja la energía y no está adquiriendo energía, entonces su estado será *triste*.
- Si el robot choca contra las paredes entonces el dolor le hará sentir *miedo*. Tendrá menos miedo si tiene hambre o está inquieto.
- Si el robot permanece en el mismo sitio mucho tiempo empezará a inquietarse. Esto le *enfadará*. El enfado persiste hasta que el robot no se mueva o no cambie su acción actual. Un robot hambriento tiende a enfadarse más.

La selección de comportamientos la realiza el controlador adaptativo. El controlador adaptativo trata de maximizar la evaluación recibida debida a la selección de un comportamiento, teniendo en cuenta los sentimientos actuales del robot y las evaluaciones recibidas previamente. Este controlador se ajusta utilizando el algoritmo de aprendizaje por refuerzo Q-learning. Para la implementación de este algoritmo, Gadanho usa redes neuronales para aprender los valores Q, que como se ha explicado en el capítulo 3, son los refuerzos esperados acumulados que recibirá el agente después de ejecutar una acción en respuesta a un estado.

Tal y como se muestra en figura 5.3, el controlador adaptativo está formado por dos módulos: el módulo de memoria asociativa y el módulo de selección de comportamientos. El módulo de memoria asociativa usa redes de realimentación para asociar los sentimientos del robot con el valor esperado actual de cada uno de los comportamientos. Las salidas de este módulo son las evaluaciones esperadas para cada comportamiento, los valores Q. El módulo de selección de comportamientos, basándose en la información dada por el módulo anterior, hace una selección estocástica del comportamiento próximo a elegir.

Según Gadanho y Hallam, ya que se asume frecuentemente que la toma de decisiones humanas consiste en la maximización de emociones positivas y la minimización de las negativas, la función de refuerzo fue ideada de manera que extrae el valor de juicio del sistema emocional considerando la intensidad de la emoción dominante y si es positiva o negativa.

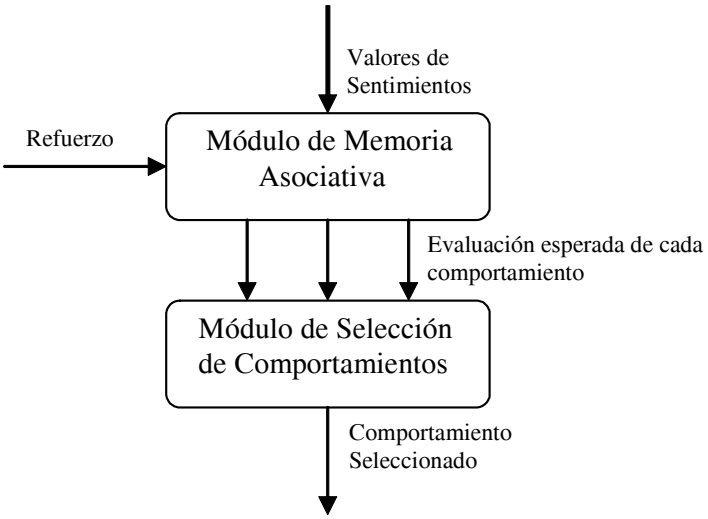


Fig. 5.3: *Controlador adaptativo propuesto por Gadanho y Hallam*

Uno de los problemas que aparecen cuando se emplean técnicas de aprendizaje por refuerzo en robótica es determinar cuándo ha ocurrido una transición de estado, es decir, cuándo el controlador necesita re-evaluar su decisión previa y tomar una nueva. Las emociones son frecuentemente interpretadas como mecanismos esenciales para los agentes autónomos, precisamente por el papel que tienen las emociones asociado a procesos de interrupción de comportamientos para tratar nuevas e inesperadas situaciones que necesitan ser atendidas. Gadanho toma como inspiración este papel que tienen las emociones de interrumpir comportamientos en sistemas naturales, para determinar las transiciones de estado en su sistema de aprendizaje por refuerzo, véase la figura 5.4.

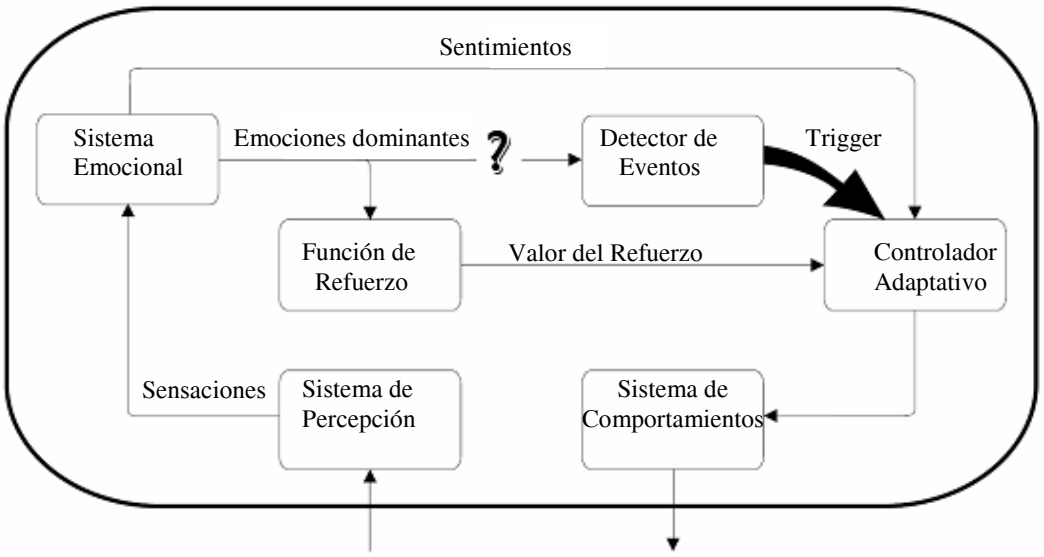


Fig. 5.4: *Emociones y Control según Gadanho y Hallam*

Cuando ocurre una transición de estado, el controlador hace una evaluación del comportamiento previo basado en el estado emocional actual y se selecciona un nuevo comportamiento acorde con la nueva situación. Si no hay una transición de estado, el comportamiento actual se seguirá ejecutando. Las simulaciones demostraron que las emociones, tal y como se propone, pueden ser usadas como refuerzos o como detectores de eventos en esta arquitectura de aprendizaje por refuerzo.

En el año 2002, Gadanho continuó este trabajo en colaboración con Luis Custodio [Gadanho and Custodio, 2002a]. El sistema emocional se sustituyó por un sistema de objetivos que representa una abstracción de ese sistema emocional con un comportamiento similar. El motivo de este cambio fue establecer una clara diferencia entre motivaciones (u objetivos) y emociones. En este sistema de objetivos, las emociones toman la forma de evaluaciones o predicciones del estado interno del agente.

La arquitectura propuesta basada en emociones, ver figura 5.5, está compuesta por el sistema de objetivos y el sistema adaptativo, que no varía en relación al controlador anteriormente descrito.

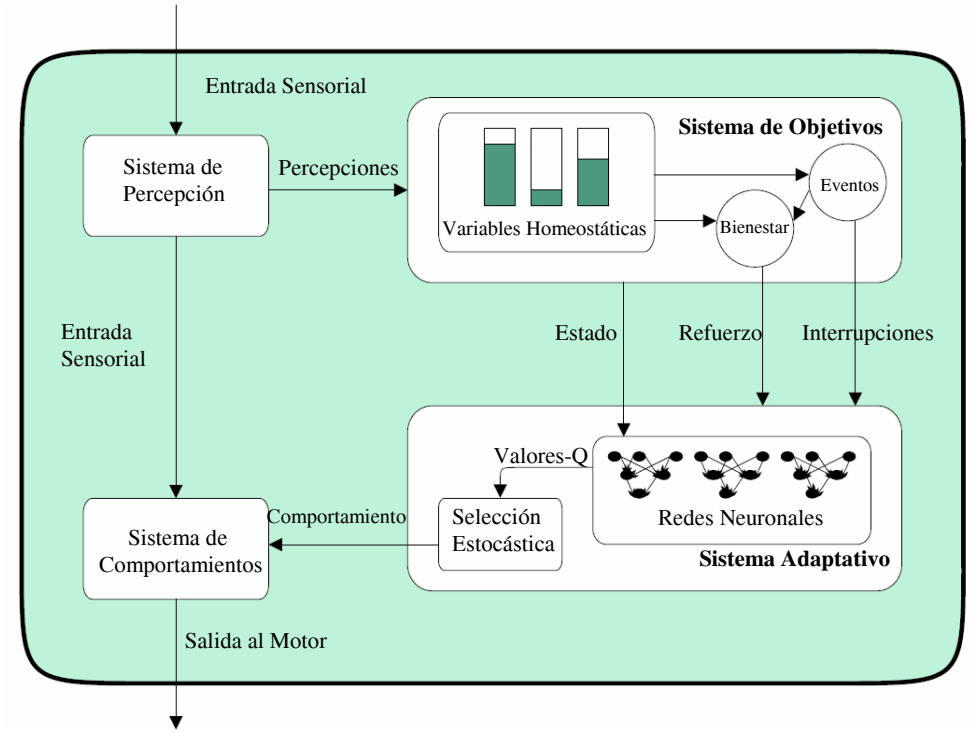


Fig. 5.5: *El controlador del robot según Gadanho y Custodio*

El sistema de objetivos está basado en un conjunto de variables homeostáticas las cuales hay que mantener dentro de un cierto rango. Las variables homeostáticas implementadas fueron: energía, bienestar y actividad. Los objetivos están explícitamente asociados e identificados con las variables homeostáticas. Para reflejar el estado hedónico actual del agente, se creó un valor de bienestar. Este valor depende principalmente del valor de los estados de las variables homeostáticas.

El sistema de objetivos evalúa la actuación del sistema adaptativo mediante aprendizaje por refuerzo en términos de sus variables homeostáticas. Determina el refuerzo a cada paso y determina cuándo se debe interrumpir un comportamiento. Como se puede apreciar, este sistema tiene la misma funcionalidad que el antiguo sistema de emociones.

Los resultados demostraron una actuación muy similar a la obtenida con el antiguo controlador emocional. De hecho, el nuevo controlador demostró ser mucho más competente y mostró una mejor actuación que usando controladores más tradicionales.

En los años posteriores, Gadanho propone y desarrolla la arquitectura ALEC [Gadanho, 2002], [Gadanho and Custodio, 2002b] y [Gadanho, 2003]. Esta arquitectura pretende una mejora del aprendizaje añadiendo, a la arquitectura previa basada en emociones, un sistema cognitivo. Este sistema complementa sus capacidades de adaptación basadas en emociones actuales, con un conocimiento explícito de reglas extraído de la interacción del agente con el entorno, ver la figura 5.6.

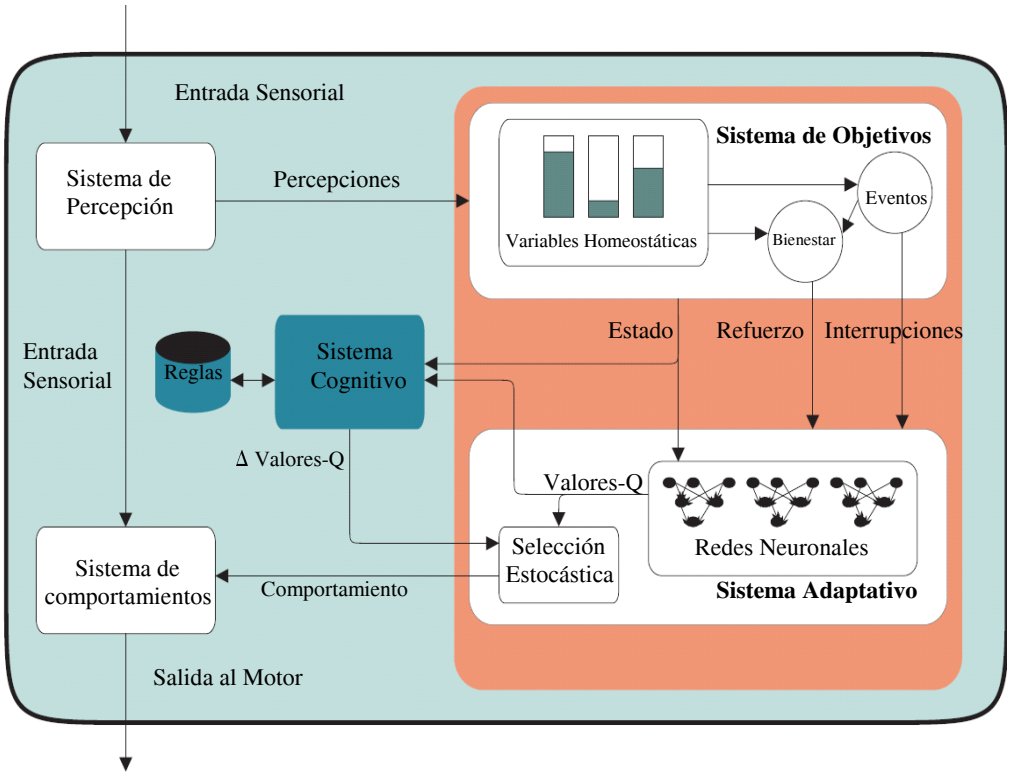


Fig. 5.6: *La arquitectura ALEC según Gadanho*

En esta arquitectura, el sistema de objetivos y el sistema adaptativo son referidos como el sistema emocional. El sistema cognitivo se añade para dar un proceso de toma de decisiones alternativo al sistema emocional. El sistema cognitivo propuesto es un sistema basado en reglas del modelo CLARION, que es un modelo cognitivo híbrido

que aborda el problema del aprendizaje de habilidades de bajo nivel y conocimiento de alto nivel. Este modelo está descrito detalladamente en los artículos anteriormente referidos.

El sistema cognitivo recoge información de manera independiente y puede tomar medidas para corregir las decisiones del sistema emocional. La colección de reglas del sistema cognitivo le permite tomar decisiones basadas en experiencias pasadas positivas. El aprendizaje de reglas está limitado a aquellos casos para los cuales hay una selección de comportamiento particularmente satisfactoria dejando los otros casos al sistema emocional. Si la regla es frecuentemente satisfactoria el agente intenta generalizarla haciendo que cubra un estado del entorno cercano. Si por el contrario, el éxito de la regla es muy pobre, intentará hacerla más específica. En la arquitectura ALEC, un comportamiento se considera un éxito si da lugar a una transición positiva del estado interno del agente, o más específicamente, de sus variables homeostáticas.

Los resultados de la implementación de la arquitectura ALEC en un robot demostraron no sólo que aprende más rápido que con la arquitectura original basada en emociones, sino que además alcanza un nivel final de actuación mejor. La manera en la que el sistema emocional influye en el sistema cognitivo es semejante a la hipótesis de Damasio de las marcas somáticas [Damasio, 1994].

5.4. AIBO: El perro robot

Tal y como se mostró en el capítulo 2, uno de los robots que más éxito comercial ha tenido en el mundo ha sido AIBO, el perro robot diseñado y fabricado por la compañía SONY. Fue el primer robot de entretenimiento autónomo, diseñado para el hogar, en utilizar inteligencia artificial [Pransky, 2001].

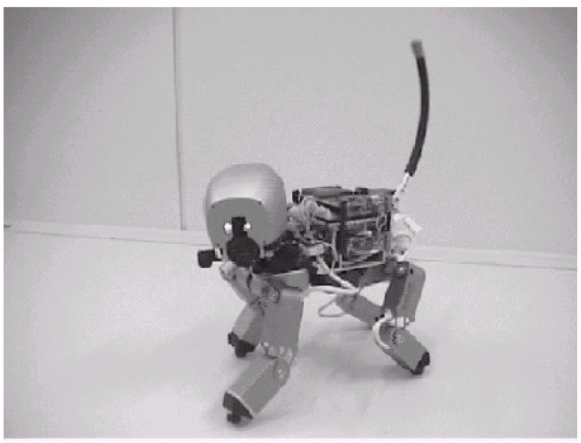


Fig. 5.7: *MUTANT*

A finales de los noventa, Fujita y Kitano [Fujita and Kitano, 1998], presentaron el prototipo inicial de AIBO: MUTANT (figura 5.7). Este robot fue desarrollado

para investigar la posibilidad de usar robots como una herramienta de entretenimiento. MUTANT fue dotado con un sistema de generación de comportamientos que consiste en un módulo de Instinto/Emoción, un módulo de conocimiento de alto nivel y subsistemas de comportamiento reactivo. En ese momento, sólo tres estados instintivo/emocional fueron asignados: Novedad/Aburrimiento, Fatiga/Activación y Felicidad/Enfado. Algunos de los sensores de MUTANT proveen la entrada de este módulo de forma que se varían los estados instintivo/emocional, los cuales generan algún comportamiento emocional final como buscar, dormir o enfadarse. Posteriormente se implementó un módulo instinto/emoción a escala completa con gran variedad de parámetros de estado emocionales y del cuerpo.

A partir de la arquitectura de MUTANT, se desarrolló la arquitectura de AIBO (figura 5.8). Para incrementar la complejidad de los comportamientos, se mejoró la anterior arquitectura en relación a la aleatoriedad, los módulos instinto/emoción, la capacidad de aprendizaje, etc.

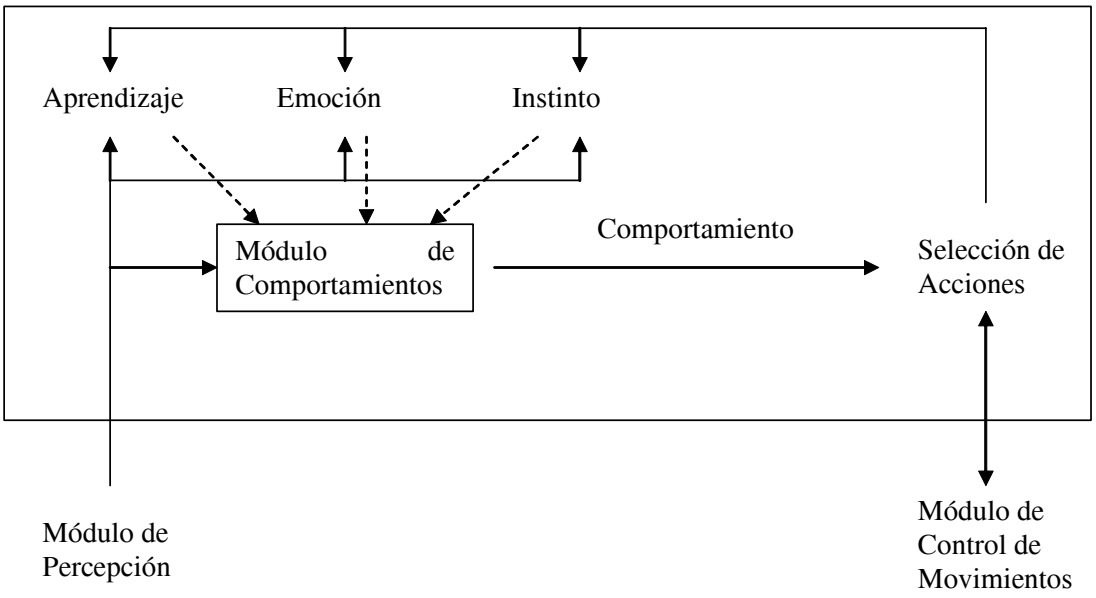


Fig. 5.8: *Arquitectura de AIBO*

Cuando finalmente se introdujo AIBO (figura 5.9) al mercado en 1999 [Fujita, 2001], este robot estaba dotado de un modelo emocional que permitía al robot tener seis emociones: alegría, tristeza, enfado, disgusto, sorpresa y miedo. Se evalúa el sensor de entrada con respecto a las emociones básicas de alegría y enfado y se asigna la dinámica apropiada a estas emociones básicas para configurar este modelo. Tanto las emociones como los instintos cambian sus valores de acuerdo con ecuaciones, las cuales son funciones de estímulos externos e instintos. Por ejemplo, cuando la alegría tiene un valor muy alto, el robot "da la pata" si ve una mano delante de él, pero se negará a hacerlo si el enfado tiene un valor muy alto.

Fig. 5.9: *AIBO 220 (izquierda) y AIBO 210 (derecha)*

En el caso de las mascotas robots como AIBO, el principal objetivo es mantener la apariencia de que está vivo. El modelo de emociones e instintos de AIBO procesa la información de una forma similar al cerebro de los mamíferos y tiene en cuenta sus comportamientos. Pero el motivo de introducir este modelo no es para ver lo bien que se pueden imitar estas emociones e instintos mamíferos, lo que se quiere es usar este modelo emocional para mejorar el comportamiento de un robot autónomo. Maximizar la apariencia viva es considerado el problema más importante para las mascotas robots. Por lo que se busca maximizar la complejidad de las respuestas y los movimientos. Los comportamientos de AIBO son generados por lo siguiente:

a. Una fusión de un comportamiento reflexivo y deliberativo a lo largo del tiempo.

b. Una fusión de motivaciones independientes dadas por las partes del robot (cabeza, cola, patas).

c. Una fusión de comportamientos que obedecen a estímulos externos y deseos internos (instintos, deseos).

El estado interno (instintos y emociones) cambia el comportamiento del robot hacia los estímulos externos. Además, el estado interno puede cambiar acorde con los estímulos externos. Por lo tanto, la complejidad total de los comportamientos exhibidos se incrementa. Al ser introducida la adaptación a través del aprendizaje, el grado de complejidad aumenta cuando el robot es observado por un largo período del tiempo.

5.5. NeCoRo: El gato robot

También en 1999, Tashima y Shibata [Tashima et al., 1999], desarrollaron un gato robot fabricado por Omron, NeCoRo, introducido en el capítulo 2, con un sistema emocional que podía generar también seis emociones: satisfacción, enfado, disgusto, temor, sorpresa y ansiedad. Previamente en 1996, Shibata y sus colaboradores [Shibata et al., 1996] ya estaban investigando el papel de las emociones en los robots, inspirándose en los sistemas biológicos donde las emociones, como ya hemos visto previamente, juegan un papel muy importante en la selección de comportamientos. En esta primera aproximación, las emociones se dividían en básicas y en sofisticadas. Las básicas son las emociones innatas y las segundas son las adquiridas a través del aprendizaje con interacción social.

En relación a la toma de decisiones proponen un algoritmo para la generación de comportamientos espontáneos, ver la figura 5.10. Los comportamientos son vistos como una combinación de acciones simples y pueden ser considerados como estrategias. Se asume que existen varios robots en el mismo espacio, que existe interacción y que cooperan entre ellos.

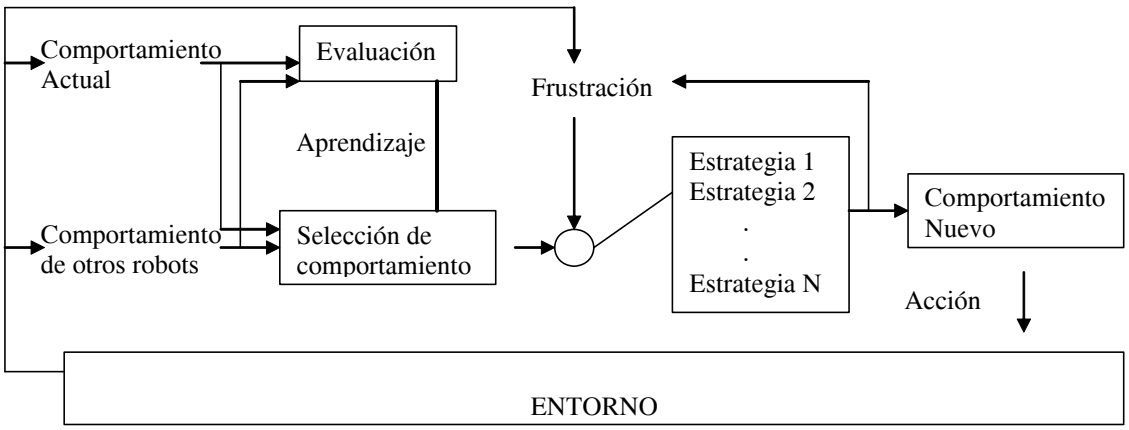


Fig. 5.10: *Un algoritmo para la generación de comportamientos espontáneos*

Este algoritmo selecciona la estrategia más apropiada para cada robot en cada situación. Si la estrategia elegida no es la adecuada, aparece la frustración en el robot, lo que produce un cambio de estrategia y una nueva situación y relación con el resto de los robots. Esta frustración es usada por el robot como una función de ajuste al entorno. Si el robot está frustrado, como consecuencia siente miedo y enfado por lo que cambiará su comportamiento para variar su relación con el resto de robots y su situación.

Posteriormente, este mismo grupo junto a otros colaboradores pertenecientes a OMRON [Shibata et al., 1999] continuaron esta misma línea de investigación. Estudiaron la interacción entre los humanos y criaturas emocionales artificiales, utilizando dos mascotas robot (un gato y una foca). Las emociones emergentes están basadas en la interacción física con los humanos. El estado interno del robot depende de la in-

formación sensorial e información recurrente. Estas mascotas robot están dotadas de sensores táctiles, auditivos y de posición para percibir la acción humana y su entorno. Como continuación a este trabajo de Tashima [Tashima et al., 1999], sale al mercado el gato robot de la casa OMROM, NeCoRo (figura 5.11). Esta mascota robot consiste en tres elementos fundamentales:

1. Sensores que perciben las emociones del usuario e intenciones, al igual que del entorno.

2. Un modelo emocional que genera emociones y deseos a través de la interacción con personas y su entorno.

3. Un generador de comportamientos que expresa sus emociones y le da apariencia de ser vivo.

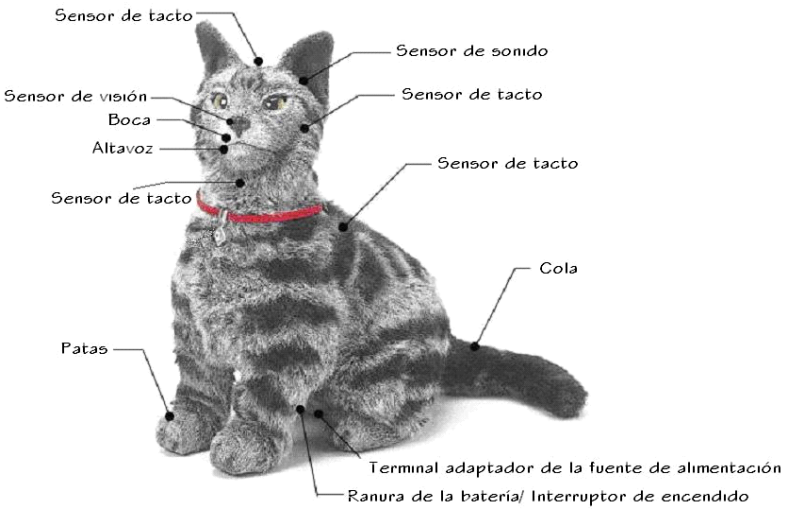


Fig. 5.11: *NeCoRo*

El modelo emocional consiste en mecanismos reactivos y deliberativos y es usado para generar las emociones del robot, deseos y personalidad. El mecanismo reactivo facilita el mapeo directo desde los sensores hasta el generador de comportamientos, donde un comportamiento no motivacional es generado. El mecanismo deliberativo tiene dos procesos: cognitivo y emocional. Estos procesos interaccionan para generar las seis emociones. Los comportamientos motivados están basados en deseos. El robot es capaz de aprender la relación causal entre estímulos y deseos.

Para NeCoRo, el planificador de acciones define la acción básica que se lleva a cabo como respuesta a una emoción que se ha generado. El generador de comportamientos determina el comportamiento acorde a la intensidad de las emociones y deseos. La acción seleccionada está combinada con factores de movimiento para determinar el comportamiento final del robot.

Las mascotas robot tienen en común, en cuanto a la forma de generar emociones mediante los modelos emocionales, la existencia de sensores externos, de manera que el estado emocional interno dependerá de la interacción física del robot con el dueño. Además en AIBO y NeCoRo aparecen los conceptos de instintos y deseos, que en ambos esquemas vienen a representar lo mismo, ya que son los que generan un comportamiento motivacional. El estado emocional también se verá afectado por la consecución o no, del objetivo de la motivación. El hecho de que este tipo de robots tengan un comportamiento complejo e incluso a veces carente de sentido, hace que el usuario piense que su robot tiene personalidad propia y por eso "hace lo que le da la gana".

5.6. El robot estudiante

Por otra parte, Michaud, Audet y colaboradores [Michaud et al., 2001] desarrollaron un robot autónomo cuya misión era la de registrarse él mismo en una conferencia como estudiante, realizar varias tareas y dar una presentación de sus ideas en la conferencia.

Los módulos que producen el comportamiento son los componentes básicos que controlan los actuadores del robot (figura 5.12). Las intenciones del robot provienen de: las condiciones determinadas por las entradas sensoriales (del *módulo implícito*), de la organización jerárquica de los objetivos del robot (organizado por el *módulo egoísta*) y del razonamiento usando el conocimiento innato o adquirido sobre la tarea a realizar (dado por el *módulo racional*). El *módulo de selección de comportamientos* combina todas las recomendaciones y activa el comportamiento que es más deseable. Los *motivos* son los que manejan los objetivos del robot, mientras que las *emociones* monitorizan el cumplimiento de los objetivos a lo largo del tiempo.

La idea es usar el nivel de energía de un motivo como una representación abstracta de cuánto progreso ha conseguido el robot en cumplir el objetivo asociado con el motivo. El modelo emocional usado es un modelo 2D bipolar con cuatro emociones emparejadas: Alegría/Tristeza y Enfado/Temor.

- Alegría/Tristeza: Monitoriza un descenso o un incremento en el nivel energético del motivo, indicando la presencia o ausencia de progreso en cumplir el objetivo asociado con los motivos activados.
- Enfado/Temor: Examina las oscilaciones o constancia en el nivel energético, indicando dificultad o no progreso en la consecución de los objetivos.

El nivel energético de los motivos y la prioridad también influyen en las emociones. Cuando el robot tarda mucho en cumplir una tarea, el nivel energético del motivo asociado se incrementa durante tanto tiempo, que la emoción tristeza se activa plenamente. Esto indica que el robot no puede realizar el objetivo y debería pedir ayuda. Si esto no funciona, el motivo será desactivado y el robot dejará de intentar participar en la conferencia. En este caso las emociones no están asociadas a entradas externas, sino a la realización de una tarea.

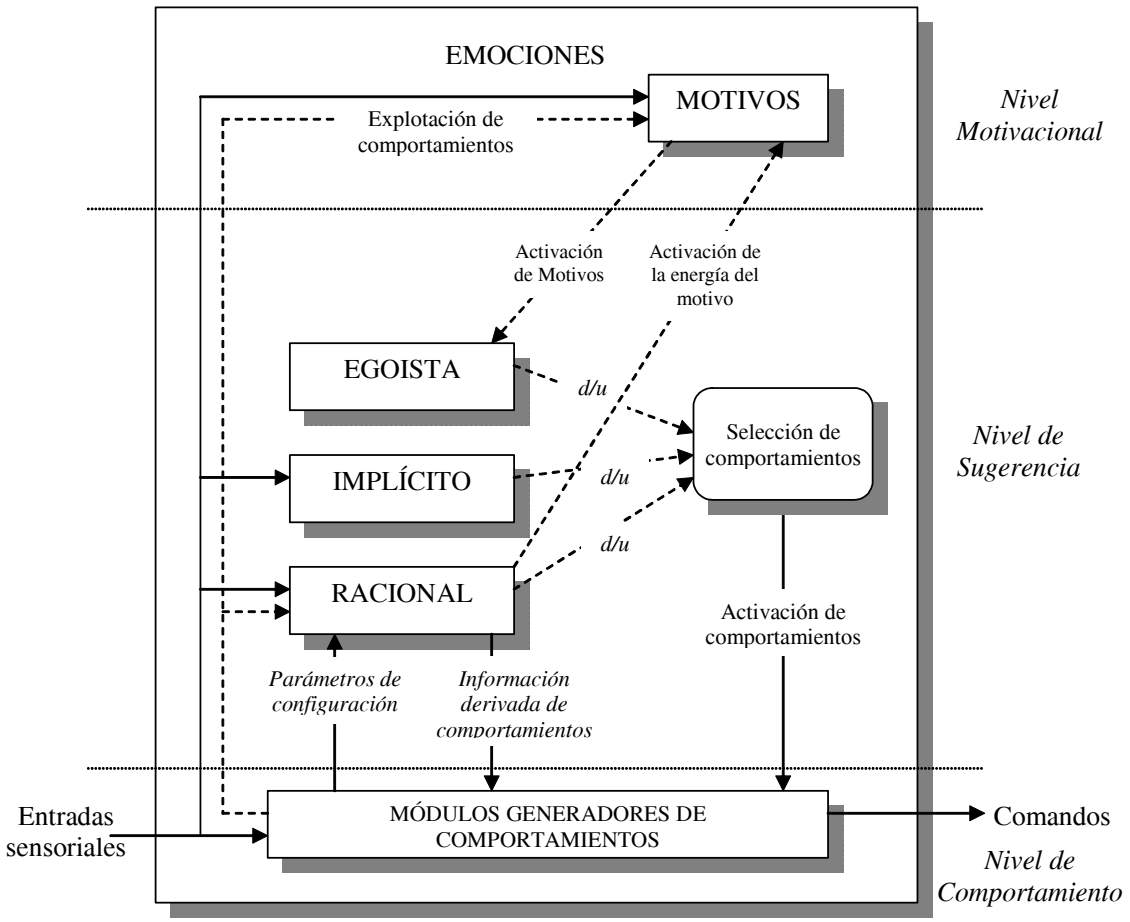


Fig. 5.12: *Arquitectura software usada por Michaud et al*

5.7. Arquitectura propuesta por Juan Velásquez

Uno de los primeros investigadores en desarrollar un modelo de generación de emociones para agentes autónomos es Juan Velásquez. En 1997 presentó la arquitectura llamada Cathexis [Velásquez, 1997]. Aunque el principal objetivo de su investigación era modelar varios aspectos de la generación de emociones, la arquitectura también muestra simples modelos para otras motivaciones y un algoritmo de selección de acciones. Inicialmente dicha arquitectura era bastante simple y posteriormente fue completada con un sistema de *drives* para formar un modelo de toma de decisiones basado en emociones [Velásquez, 1998a]. Los sistemas más importantes del modelo están representados por módulos (figura 5.13). Los sistemas de percepción obtienen información del mundo y proveen de las características de los estímulos y objetos, a los sistemas emocional y de comportamiento. Estos sistemas también reciben señales de error de los sistemas de los *drives*. Los sistemas emocionales valoran la importancia emocional de los estímulos y de acuerdo con eso, hacen tender las respuestas de comportamiento y las percepciones futuras.

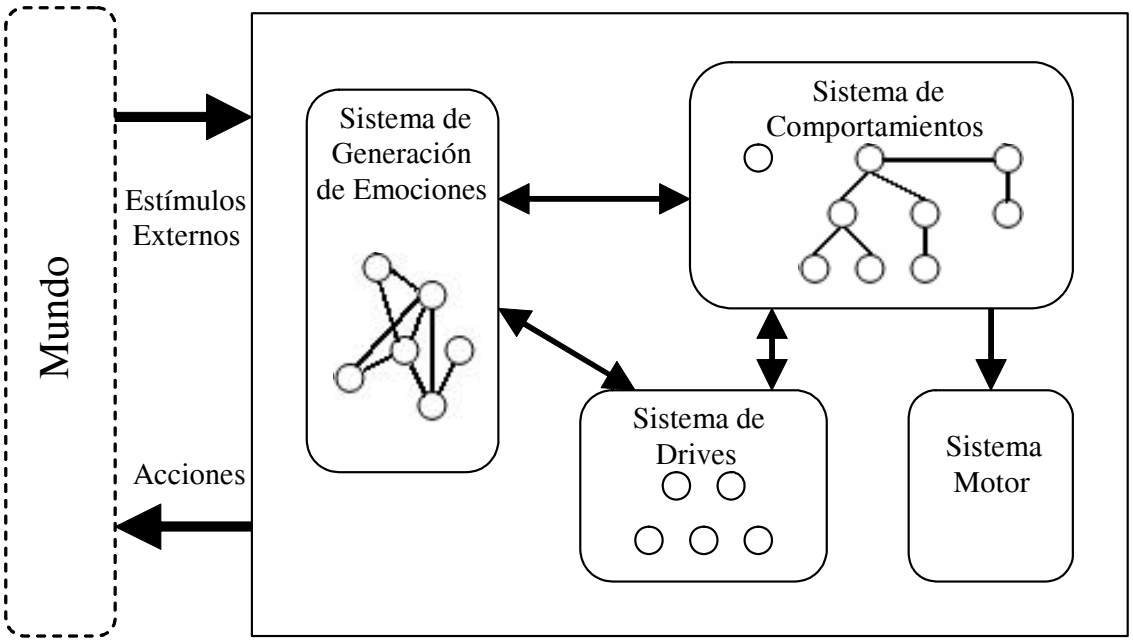


Fig. 5.13: *Arquitectura Cathexis, Velásquez*

Los *drives*, según Velásquez, son sistemas motivacionales que representan urgencias, que llevan al agente a la acción. Sin embargo, en este modelo, es el sistema emocional el que constituye la principal motivación del agente. Incluso el sistema de *drives* explota su influencia para seleccionar comportamientos específicos. Por ejemplo, la señal de error del *drive* Hambre y la angustia causado por ello, lleva al agente a obtener comida.

Con respecto al sistema de generación de emociones, de nuevo como en otros autores, las emociones implementadas están basadas en los estudios de Ekman: enfado, miedo, tristeza, alegría, disgusto y sorpresa [Ekman, 1992]. Define también distintos tipos de activadores de las emociones: cognitivos y no-cognitivos. Los cognitivos no están pre-establecidos sino que son aprendidos por el agente a medida que interacciona con su entorno. Los no-cognitivos están relacionados con el estado de los *drives*, expresiones faciales y otros procesos físicos [Velásquez, 1998b].

En este modelo son los comportamientos los que compiten entre sí para obtener el control, de forma que sólo un comportamiento está activo cada vez. Este modelo incluye las memorias emocionales basadas en la teoría de Damasio [Damasio, 1994] de las marcas somáticas. Estas memorias forman parte del proceso de toma de decisión. Por lo tanto, si el agente se encuentra con un estímulo "marcado", como por ejemplo una comida que no le gustó, la memoria asociada será revivida reproduciendo el estado emocional que se experimentó previamente, influyendo así en la toma de decisión.

Con este modelo se muestra cómo los *drives*, las emociones y los comportamientos pueden ser integrados en una arquitectura robusta. Esta arquitectura usa alguno de los mecanismos de las emociones para adquirir recuerdos de experiencias emocionales

pasadas. Dichas experiencias sirven como mecanismos de predisposición mientras se toman las decisiones durante el proceso de selección de acción.

Este modelo, y sus versiones previas, fueron probados satisfactoriamente en agentes virtuales como Simon [Velásquez, 1997], que simulaba a un bebé humano y Yuppy [Velásquez, 1999], que era una mascota robot emocional. Posteriormente este modelo fue ampliado para el estudio del aprendizaje social de robots, en concreto con el robot Kismet, el cual será descrito a continuación [Breazeal and Velásquez, 1998].

5.8. Kismet

El equipo dirigido por Cynthia Breazeal lleva bastante tiempo investigando la interacción humano-robot. Los dos robots más relevantes dentro de un proyecto dedicado a robots sociables han sido ***Kismet*** y ***Leonardo***. El principal interés del proyecto es el estudio de cómo mecanismos inspirados en las emociones, pueden mejorar la forma en la que los robots funcionan en un entorno humano. Además estudia cómo dichos mecanismos pueden mejorar la habilidad de los robots para trabajar de forma efectiva, colaborando con gente [Breazeal and Brooks, 2004].

Fig. 5.14: *Kismet*

Originalmente, en el proyecto de los robots sociales se exploró la interacción social humana más simple [Breazeal and Aryananda, 2002] (guiada e inspirada por lo que ocurre entre un niño y su cuidador) y para ello se construyó un robot social llamado Kismet (figura 5.14). Kismet puede comunicar su estado afectivo y otras señales sociales a un humano a través de expresiones faciales, posturas del cuerpo (es un torso), dirección de la mirada y modulación de la voz.

Usando como inspiración los modelos de inteligencia en los sistemas naturales, el diseño de la arquitectura de Kismet contiene un sistema cognitivo y un sistema emocional. El sistema cognitivo está formado por los sistemas de percepción, atención, *drives* y comportamientos. El sistema emocional a su vez, está formado por los liberadores afectivos, las evaluaciones, los extractores y los procesos de huída que organizan las respuestas emocionales.

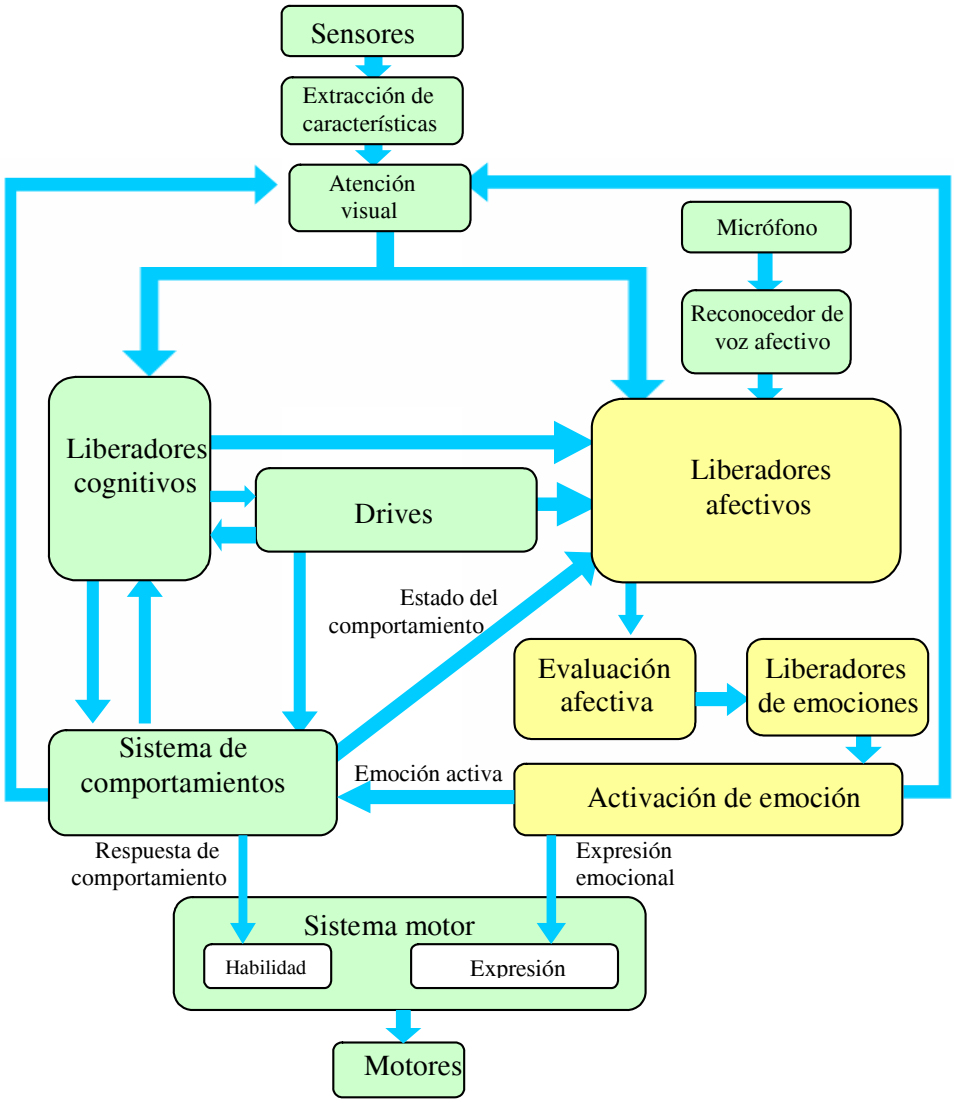


Fig. 5.15: *Arquitectura de Kismet*

El sistema emocional está diseñado para ser un complemento del sistema cognitivo que media entre los eventos del entorno, sociales e internos, para generar una respuesta adaptativa. Esta respuesta realiza las funciones tanto social como de automantenimiento.

Las emociones que han sido implementadas en Kismet son las seis emociones básicas propuestas por Ekman [Ekman, 1992]: enfado, disgusto, miedo, alegría, pena y sorpresa. Además se añaden otras tres: calma, interés y aburrimiento. Una descripción más detallada sobre la implementación de cada una de las respuestas emotivas se puede encontrar en [Breazeal, 2002].

5.9. Leonardo

Leonardo (figura 5.16), es el sucesor de Kismet y es la continuación del trabajo de Breazeal, en la construcción de robots sociables que son capaces de comunicarse y aprender de la gente.

Fig. 5.16: *Leonardo*

El sistema afectivo del robot está basado en los modelos computacionales de las emociones básicas inspiradas por Kismet. Cuando el robot expresa su estado interno, el humano es capaz de ayudar al robot de forma intuitiva en su proceso de regulación (como fue demostrado con Kismet). Leonardo comunica su estado emocional principalmente a través de su expresión facial, mezclando continuamente siete poses faciales que caracterizan su espacio de expresión emocional. Además, el estado emocional influye en el comportamiento [Thomaz et al., 2005].

Pero lo más interesante de Leonardo, además del estudio de la interacción humano-robot, son los métodos de aprendizaje implementados. Actualmente este grupo está explorando múltiples formas de aprendizaje guiado socialmente. El primero es el *Aprendizaje por Tutela* para enseñar a Leonardo cómo realizar los nuevos objetivos y tareas. El segundo es el *Aprendizaje por Imitación* para enseñar a Leonardo cómo llevar a cabo nuevas habilidades, así como por ejemplo, inferir la intención de una acción observable. Otro más es el *Aprendizaje por Referencia Social* mediante el cual Leonardo aprende cómo valorar nuevos objetos, observando la respuesta emocional que tiene otra persona al interactuar con dicho objeto [Breazeal et al., 2005]. La referencia social representa un nuevo canal de comunicación emocional entre humanos y robots, el cual permite al humano formar el entendimiento del robot y la exploración de su entorno [Thomaz et al., 2005].

5.10. Expresión de emociones en los robots

La expresión de las emociones es necesaria para facilitar la interacción humano-robot. Aunque es cierto que en los humanos la expresión no siempre es el reflejo real del estado emocional, en el caso de los robots no tendría sentido, ya que si por ejemplo no entendiese alguna orden, y nos sonríe, nosotros creeríamos que sí nos ha entendido. Asimov, en *Robots al amanecer* [Asimov, 1983], describe a los robots del futuro como humanoides, capaces de entender situaciones de lo más complicadas, a la vez que interpretan las acciones y situaciones humanas, pero incapaces de expresar emociones. Como consecuencia, los humanos están continuamente preguntando a los robots si han entendido las órdenes. Ni un extremo ni otro, ni socialmente educados ni tampoco unos robots totalmente inexpresivos.

5.10.1. Lenguaje

El lenguaje es un método altamente efectivo para comunicar emociones. Los principales parámetros que gobiernan el contenido emocional del lenguaje son: la intensidad, el tono (nivel, variación, rango) y la métrica. La calidad del lenguaje sintetizado es significantemente más pobre que las expresiones faciales y el lenguaje del cuerpo sintetizados. A pesar de estos problemas, ha sido probado que es posible generar un lenguaje emocional. Cahn [Cahn, 1990] describe un sistema para mapear la carga emocional (como la pena) en los parámetros de los sintetizadores del lenguaje, incluyendo la articulación, el tono y la calidad de voz.

5.10.2. Expresión facial

En términos de expresión, algunos robots son sólo capaces de mostrar emociones de manera limitada, como con luces que parpadean o labios manejados individualmente. Otros robots tienen muchos grados de libertad activos y pueden por lo tanto proporcionar movimientos y gestos más ricos. La cara humana sirve para muchos propósitos, como mostrar la motivación de un individuo, lo que ayuda a que su comportamiento sea más predecible y entendible para los otros. Los gestos de la cara y la expresión, también comunican información y ayudan a regular el diálogo.

El comportamiento expresivo de las caras robóticas generalmente no tiene una apariencia real. Esto refleja las limitaciones del diseño y del control mecatrónico. Por ejemplo, las transiciones entre las expresiones tienden a ser bruscas, ocurriendo de forma rápida y sin previsión. Dos de las caras más simples aparecen en Sparky y Feelix (figura 5.17) [Fong et al., 2002]. La cara de Sparky tiene cuatro grados de libertad (cejas, párpados y labios) los cuales muestran un set de emociones básicas. Feelix es un robot desarrollado usando el kit de construcción de robots de LEGO Mindstorm. La cara de Feelix también tiene cuatro grados de libertad (dos cejas y dos labios) y está diseñado para mostrar seis expresiones faciales (enfado, tristeza, temor, felicidad, sorpresa y neutro), más un número de mezclas de emociones [Cañamero and Fredslund, 2000].

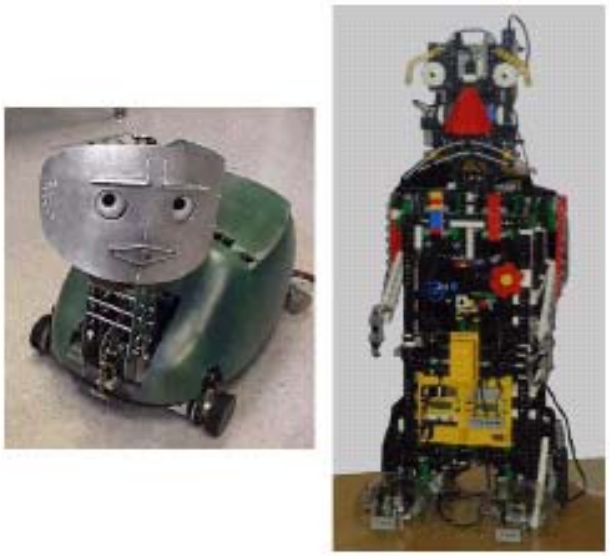


Fig. 5.17: *Sparky (izquierda) y Feelix (derecha)*

En contraste con Sparky y Feelix, el robot quizás más conocido por su capacidad de expresar emociones es Kismet (figura 5.18), el cual tiene cejas, orejas, ojos, párpados controlables, una boca con dos labios y un cuello movible. Breazeal y Aryananda [Breazeal and Aryananda, 2002] publicaron un artículo centrado en el reconocimiento de la comunicación afectiva usando una comunicación directa sin tener en cuenta el contenido lingüístico. Los comportamientos son básicamente reacciones emocionales, produciendo sonidos vocales y expresiones faciales del robot. La ventaja de Kismet está en detectar señales emocionales de los humanos, lo que le permite devolver la respuesta emocional apropiada. Como resultado, Kismet, es capaz de mostrar una gran variedad de expresiones faciales que son el espejo de su estado afectivo.

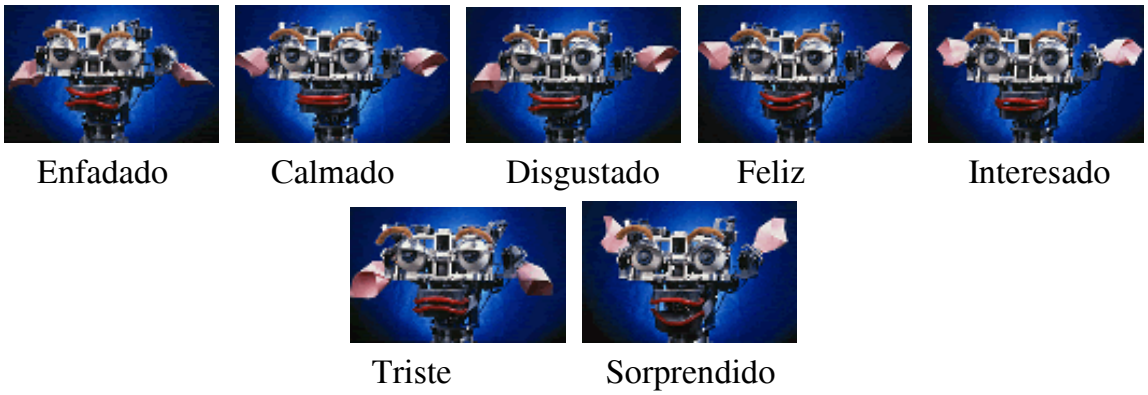


Fig. 5.18: *Expresiones de Kismet*

Leonardo (figura 5.19) tiene una expresiva cara de silicona (24 grados de libertad, GDL, sin contar con las orejas) capaz de un nivel de expresión similar al humano y un sistema de visión binocular activo (4 GDL). Esto lo convierte en una plataforma ideal para la implementación de imitación facial. Además, este robot está equipado con dos brazos con 6 GDL, dos manos con 3 GDL, dos orejas dirigibles con 3 GDL y un cuello con 5 GDL. El resto de los GDL están en los hombros, la cintura y caderas. Todos estos grados de libertad proporcionan a Leonardo una apariencia muy "viva".

Fig. 5.19: *Leonardo*

El robot iCat, ver la figura 5.20, como se mostró en el capítulo 2, es una plataforma para estudiar la interacción humano-robot. Este robot tiene 13 motores que controlan diferentes partes de la cara, como las cejas, ojos, párpados, boca y posición de la cabeza. iCat puede expresar diferentes expresiones emocionales (feliz, sorpresa, enfado, triste), necesarias para crear diálogos de interacción social con los humanos [Philips, 2007].

Fig. 5.20: *Expresiones de iCat*

En el caso de tratarse de robots-guía, en [Fong et al., 2002] se sugieren tres características críticas, para el éxito de los robots que deben exhibir interacciones espontáneas en entornos públicos. La primera, es que el robot debe tener un punto focal, el cual sirve como punto de atención para el humano. Una cara robótica o animada es usada para esta función en muchos robots-guía. Segunda, el robot debe comunicar su estado emocional a los visitantes como una manera eficiente de comunicar su intención, por ejemplo, su frustración al ser bloqueado por turistas. Tercera, el robot debe tener la capacidad de adaptar sus parámetros de interacción humana, basándose en los resultados de interacciones pasadas, de forma que pueda continuar mostrando un comportamiento final abierto, como los visitantes en su entorno.

Por ejemplo, Minerva (figura 5.21) desarrollado en la Universidad de Carnegie Mellon, es un robot-guía que fue probado en un Museo de Historia Americana. Minerva usa una caricatura facial monitorizada que consiste en ojos, labios y cejas que proveen de punto focal y el medio para comunicar su estado emocional. Un simple diagrama de transición de estado modula la expresión facial y el lenguaje, de feliz a neutral y a enfadado, basado en la cantidad de tiempo durante el cual el camino de Minerva ha estado bloqueado.

Fig. 5.21: *Minerva (CMU)*

Quizás las caras de robots más realistas sean las diseñadas por el Laboratorio de Robótica Inteligente de la Universidad de Osaka. Estos robots están diseñados con la idea de ser "copias" de humanos, con el fin de que éstos sean sustituidos a la hora de realizar ciertas tareas. Estas caras están explícitamente diseñadas para ser similares a las humanas e incorporan pelo, dientes, y una capa de piel hecha de silicona. Numerosos puntos de control actuados por debajo de la piel producen una amplia variedad de movimientos faciales y expresiones humanas, ver la figura 5.22 [Ishiguro, 2007].

Fig. 5.22: *Repliee Q2, Universidad de Osaka*

5.10.3. Expresión corporal

Además de las expresiones faciales, la comunicación no-verbal es realizada frecuentemente a través de gestos y movimientos del cuerpo. En el caso de las mascotas robots, está claro que la expresión de las emociones, mediante gestos, es esencial para dotarles de una apariencia de ser vivo.

En el caso de AIBO y NeCoRo, expresan esas emociones considerando que uno es un perro y el otro un gato. Por ejemplo, cuando AIBO está contento dará la pata, si está triste bajará la cabeza y emitirá un sonido triste y, si por ejemplo, entra en un estado de euforia, se comportará como un perro exaltado moviéndose nerviosamente. NeCoRo en cambio "ronronea" si está contento (figura 5.23) y si le tiran de la cola se queja.

Fig. 5.23: *AIBO jugando (izquierda) y NeCoRo ronroneando (derecha)*

Por otro lado, PaPeRo, que ya fue introducido en el capítulo 2 (figura 5.24), al no tratarse de un "animal", tiene otras funciones más similares quizás a las de un humano. Dispone también de un software de reconocimiento de caras, de manera que si se encuentra con alguien que le gusta, entonces hace parpadear sus LEDs, se acerca a la persona, habla con ella y toca música. Si en cambio se encuentra con alguien que no le gusta, se aleja de la persona, se enfada y no hace caso de sus órdenes ignorándolo. Evidentemente, el hecho de que estos robots expresen sus emociones facilita enormemente la interacción y el entendimiento entre el robot y los usuarios.

Fig. 5.24: *PaPeRo interaccionando con niños*

6. SISTEMA DE TOMA DE DECISIONES

6.1. Presentación del sistema

En este capítulo se va a presentar el sistema de toma de decisiones propuesto en esta tesis. El sistema ha sido desarrollado en base a los conceptos de motivación, *drive*, emoción y aprendizaje por refuerzo. Tal y como se ha mostrado en los capítulos previos, todos estos conceptos son esenciales a la hora de estudiar el comportamiento tanto animal como humano.

El objetivo de esta tesis es conseguir que un agente/robot sea completamente autónomo y por lo tanto tome sus propias decisiones. Esta toma de decisiones tiene que ser aprendida a través de su propia experiencia: sus éxitos y fracasos. A través de ellos, mediante aprendizaje por refuerzo, el agente aprende políticas de comportamiento adecuadas para sobrevivir.

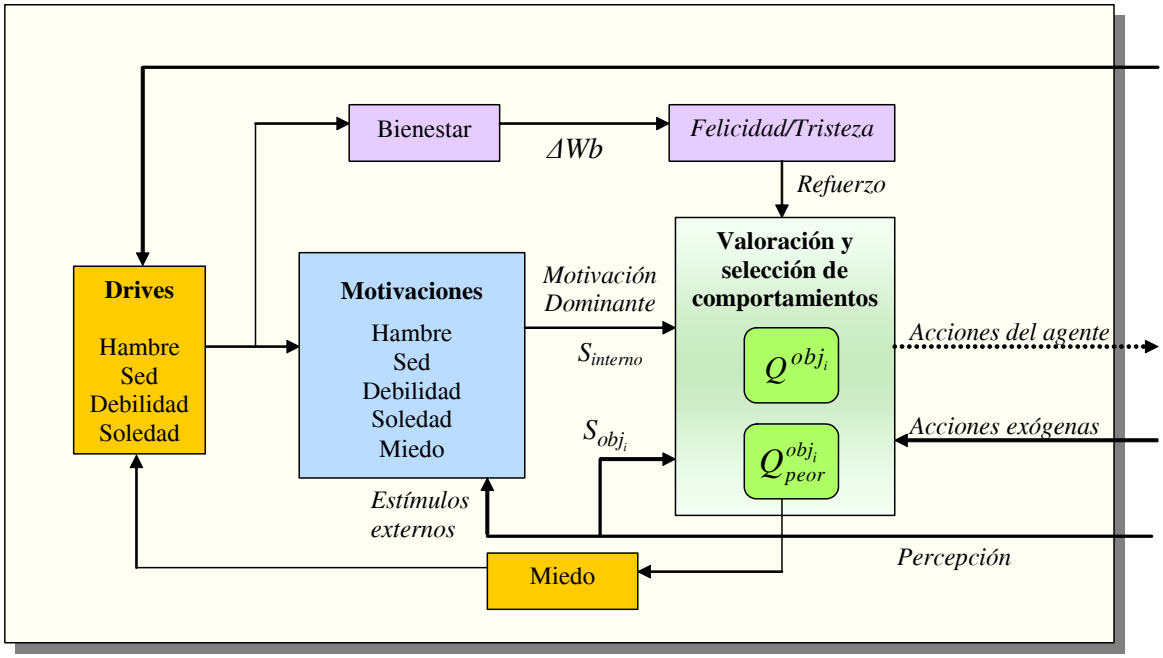


Fig. 6.1: *Sistema de toma de decisiones propuesto*

El sistema de toma de decisiones propuesto está formado por varios módulos: el módulo motivacional, el módulo de los *drives* y un módulo de valoración y selección de comportamientos. La función de refuerzo utilizada es la felicidad o la tristeza que, como serán definidas posteriormente, están relacionadas con la variación del bienestar del agente. Como aportación original se introduce el miedo como elemento esencial en la toma de decisiones, ver la figura 6.1. El proceso que sigue este sistema se describe a continuación.

Los *drives* o necesidades del agente varían su valor a cada paso de simulación siguiendo cada uno una dinámica determinada. Estos valores se introducen, junto con los valores de los estímulos externos, en el módulo motivacional, donde se calculan las intensidades de las motivaciones relacionadas con cada *drive*. La motivación de mayor intensidad es la motivación dominante y va a ser la que determine el estado interno del agente. Este estado interno del agente junto con el estado externo del agente van a determinar el estado del agente.

Por otro lado, incluido dentro del grupo de los *drives* del agente, está el miedo. En la figura 6.1 se ha representado aparte para especificar que su dinámica es distinta a la de los otros *drives*. El miedo tiene que ver con las peores valoraciones, tanto de las acciones propias, como de las acciones realizadas por otros agentes, acciones exógenas.

El módulo de valoración y selección de comportamientos elige los comportamientos de acuerdo con una política de selección determinada. La valoración de los comportamientos, cuando el agente está en un cierto estado, se realiza utilizando algoritmos de aprendizaje por refuerzo, en concreto Q-learning. Por lo que el agente aprende qué acción escoger cuando está en un estado determinado. El refuerzo utilizado para valorar el resultado de la ejecución de un comportamiento es la felicidad y la tristeza, definidas en función de la variación que experimenta el agente en su bienestar: ΔWb. El bienestar es una función de las necesidades del agente, por lo que este refuerzo mide el efecto de la acción escogida en las necesidades del agente. Tal y como se justificará posteriormente en este capítulo, las variaciones positivas y negativas del bienestar están directamente relacionadas con las emociones de felicidad y tristeza respectivamente. El agente, por lo tanto, utiliza estas emociones para valorar sus propias acciones y aprender cuáles son las más apropiadas en cada estado.

6.2. Estado del agente

Tal y como se ha mostrado en el capítulo 3, es necesario conocer el estado del agente para la toma de decisiones. En este sistema, el estado del agente es la combinación de su estado interno $S_{interno}$ y su estado externo $S_{externo}$.

$$S = S_{interno} \times S_{externo} \tag{6.1}$$

A continuación se van a definir los estados interno y externo.

6.2.1. Estado interno

El estado interno del agente depende de las motivaciones, que están ligadas a las necesidades, es decir, los *drives* del agente. En este sistema no se van a considerar otros factores que posiblemente afecten al estado interno de un ser humano, tales como los ciclos naturales, factores psicológicos, etc.

Drives

En este sistema se va a considerar que los *drives* del agente están relacionados con necesidades fisiológicas, y además, con la necesidad de interacción social y la necesidad de seguridad. Se considera que a medida que la necesidad aumenta, la intensidad del *drive* aumenta. Los *drives* que van a ser considerados son los siguientes:

- Hambre
- Sed
- Debilidad
- Soledad
- Miedo

El valor ideal, e inicial, de todos los *drives* está seleccionado como cero. Se considera que un *drive* está satisfecho cuando su valor es cero, es decir, no hay necesidad.

Los valores de los *drives* Hambre y Sed se incrementan una cierta cantidad a cada paso de la simulación. Estos *drives* no aumentan al mismo ritmo. Estudios fisiológicos determinan que en la mayoría de los seres humanos la necesidad de agua, la sed, se produce antes que la necesidad de comer, el hambre. En [Gautier and Boeree, 2005] se presenta cómo Marlow descubrió que ciertas necesidades prevalecen sobre otras. Por ejemplo, si se está hambriento y sediento, se tenderá a calmar la sed antes que el hambre. Después de todo, se puede sobrevivir sin comer varios días, pero sólo se podrá estar un par de días sin agua. La sed es una necesidad "más fuerte" que el hambre.

El *drive* Debilidad va a aumentar cada vez que el agente se mueve, es decir, será el coste que tiene que pagar el agente por moverse.

En cuanto al *drive* Soledad, o *drive* social, se va a considerar que también aumenta a cada paso de la simulación una cierta cantidad, pero a un ritmo mucho más lento que los *drives* Hambre y Sed.

Por último, el *drive* Miedo, tal y como se justificará al final de este capítulo, no aumenta de forma progresiva. Si el agente se encuentra en un estado "seguro" el *drive* valdrá cero y si no, un valor alto.

Modelado de las motivaciones

En esta tesis se va a considerar que los estados motivacionales representan tendencias para comportarse de cierta manera, como consecuencia de factores internos (los *drives*) y externos (estímulos incentivos) [Ávila García and Cañamero, 2004]. En otras palabras, el estado motivacional es una tendencia para corregir el error, el *drive*, a través de la ejecución de comportamientos. Además también se considera que el miedo es una motivación, por lo que se van a combinar las dos teorías motivacionales homeostáticas presentadas en la sección 3.3.1.

Para modelar las motivaciones del agente, excepto el miedo, se va a utilizar como inspiración el modelo hidráulico de Lorenz y Leyhausen [Lorenz and Leyhausen, 1973]. Como ya ha sido explicado en el capítulo 3, este modelo es esencialmente una metáfora que sugiere que el *drive* crece internamente como la presión de un fluido en un tanque, hasta que sale de golpe a través de una salida, tal y como se mostró en la figura 3.2. El estímulo motivacional del mundo exterior actúa para abrir una válvula de escape, liberando el *drive* que se expresa en un comportamiento. En este modelo, la fuerza del *drive* interior interacciona con la fuerza del estímulo externo. Si el *drive* está bajo, se necesita entonces un estímulo fuerte para activar el comportamiento motivado. Por otro lado, si el *drive* está alto, un estímulo medio será suficiente [Berridge, 2004].

Siguiendo esta idea, la intensidad de las motivaciones, excepto el miedo, en la arquitectura (M_i) es la suma de la intensidad del *drive* relacionado (D_i) y el valor del estímulo externo relacionado (w_i), tal y como se expresa en la siguiente ecuación:

$$M_i = D_i + w_i \tag{6.2}$$

Los estímulos externos o motivacionales, son determinados objetos que el agente puede encontrarse en el mundo. Estos estímulos son los mismos estímulos incentivos utilizados por Cañamero [Cañamero, 1997]. De esta manera, determinados comportamientos, los comportamientos consumatorios, sólo pueden ejecutarse en presencia de estos estímulos. De acuerdo con la ecuación (6.2), una motivación puede tener una intensidad alta debido a dos razones:

1. El valor del *drive* correspondiente es alto.
2. El estímulo incentivo asociado está presente.

Este modelo puede explicar el hecho de que, ante la disponibilidad de comida delante de nosotros, algunas veces comamos aunque no tengamos mucha hambre.

En este sistema de toma de decisiones, tal y como se propone en [Balkenius, 1993] y [Balkenius, 1995], una vez que se calculan las intensidades de todas las motivaciones, éstas entran en competición. La motivación con la intensidad más alta es la motivación dominante y es la que determina el estado interno del agente.

También se han introducido niveles de activación (L_d) para las motivaciones, de manera que las motivaciones cuyo *drive* relacionado supere este nivel, entran en com-

petición para convertirse en la motivación dominante. Si no es el caso, la intensidad de la motivación asociada será cero, tal y como refleja la siguiente ecuación:

$$\begin{array}{l} \text{Si } D_i < L_d \text{ entonces } M_i = 0 \\ \text{Si } D_i \geq L_d \text{ entonces se aplica (6.2)} \end{array} \tag{6.3}$$

Puede ocurrir que ninguno de los *drives* tenga un valor superior a este límite. En este caso, no hay ninguna motivación dominante y se puede considerar que el agente no tiene ninguna necesidad.

El estado interno del agente, tal y como se dicho, va a estar determinado por la motivación dominante. Las motivaciones van a competir entre sí de manera que sólo una de ellas va a ser la que determina el estado interno del agente. El hecho de considerar que el estado interno del agente es la motivación dominante, no deja de ser una simplificación. El agente, en un determinado momento, puede tener varias motivaciones con intensidades altas compitiendo entre sí, por lo que, considerar que el agente, por ejemplo, sólo está hambriento cuando su *drive* Sed es también alto, es una simplificación de la situación real. Esta simplificación ayuda a que el tratamiento del estado del agente sea más sencillo. Si no se hiciera así, los estados internos del agente tendrían que ser combinaciones de las cinco motivaciones existentes, lo que hace que aumente mucho el número de estado internos a considerar.

En otras arquitecturas, como la propuesta en [Ávila García and Cañamero, 2002], también existe esta competición entre motivaciones. Finalmente, es la motivación de mayor intensidad la que determina qué comportamientos, ligados a través de la fisiología a esa motivación dominante, pueden ser ejecutados. En este sistema propuesto, sin embargo, el agente debe aprender cuáles son los comportamientos que debería ejecutar para conseguir la reducción del *drive* asociado a la motivación dominante.

Por otro lado, al asumir que el estado interno es la motivación dominante el entorno deja de ser Markoviano. El motivo es que, debido a la dinámica de crecimiento de los *drives*, puede existir una transición de estado interno no debida a una acción ejecutada por el agente. Por lo tanto, el entorno considerado en esta tesis no cumple las condiciones de un proceso de Markov. Aún así, para tratar este problema de aprendizaje por refuerzo, se va a utilizar el algoritmo Q-learning. Este algortimo será modificado para tener en cuenta algunos efectos producidos por la simplificación que se va a considerar en la sección siguiente.

6.2.2. Estado externo

El estado externo es el estado del agente en relación a todos los objetos, pasivos y activos, que puedan interactuar con él:

$$S_{externo} = S_{obj_1} \times S_{obj_2} ... \tag{6.4}$$

Debido a que esto implica un número elevado de estados, en este sistema se va a considerar que los estados relacionados con los objetos son independientes entre sí. Es decir, el agente en cada momento va a considerar que su estado en relación a la comida es independiente de su estado en relación al agua, a la medicina, etc. Esta simplificación va a reducir en gran manera el número de estados que se tienen que considerar en el proceso de aprendizaje del agente.

Sin esta simplificación, el número de estados del agente en relación a todos los objetos sería muy grande. Por ejemplo, si existieran 10 objetos en el mundo y se asume que para cada objeto se tienen 3 variables: tener el objeto, estar junto al objeto y saber dónde está el objeto, se tendrían $2^3 = 8$ estados relacionados con cada objeto. Si el estado externo del agente es su estado en relación a todos los objetos, en total existirían $8^{10} = 1073741824$ estados, un número enorme. Sin embargo, con la simplificación se considera que el estado externo es el estado del agente en relación a cada uno de los objetos por separado, por lo que para los 10 objetos se obtendrían $10 \times 8 = 80$ estados, lo que implica una reducción muy importante del número de estados.

6.3. Función de refuerzo: Felicidad y tristeza

6.3.1. Bienestar

Como ya se ha mostrado, los *drives* del agente están relacionados con necesidades fisiológicas (Hambre, Sed y Debilidad), y en este sistema además, con la necesidad de interacción social (Soledad) y la necesidad de seguridad (Miedo). Por lo tanto se va a definir el bienestar del agente como el grado de satisfacción de sus necesidades. De manera que, cuando todos los *drives* del agente están satisfechos, es decir que todos los *drives* valen cero, el bienestar es máximo.

Tal y como se muestra en la ecuación (6.5), el bienestar de un agente es una función de los valores de sus *drives* D_i y factores de "personalidad" α_i, los cuales valoran la importancia de cada uno de los *drives* para el bienestar del agente.

$$Wb = Wb_{ideal} - \sum_i \alpha_i \cdot D_i \tag{6.5}$$

Wb_{ideal} es el valor ideal del bienestar del agente. A medida que los valores de los *drives* del agente van aumentando con el tiempo, o como efecto de alguna otra acción, el bienestar del agente disminuye. Dependiendo de los valores de los factores de personalidad, el aumento de los *drives* puede afectar en mayor o menor medida en el bienestar del agente. Siempre que exista una reducción de los valores de los *drives*, existe un aumento del bienestar.

El valor del bienestar del agente se calcula en cada paso de la simulación, además de su variación ($\triangle Wb$). Esta variación del bienestar se calcula como el valor actual del bienestar menos el valor en el paso anterior, tal y como se muestra en (6.6).

$$\Delta Wb^{k+1} = Wb^{k+1} - Wb^{k} \tag{6.6}$$

Las mayores variaciones positivas del bienestar se van a producir cuando se reduce el *drive* relacionado con la motivación dominante.

6.3.2. Felicidad y tristeza

Tomando como base las definiciones de las emociones establecidas por Ortony [Ortony et al., 1988] descritas previamente en el capítulo 4, se considera que las emociones ocurren debido a una reacción valorada (positiva o negativamente) de las situaciones. De acuerdo con este punto de vista, se vio que en [Ortony, 2003], Ortony propone que la felicidad se daba cuando le pasa algo bueno al agente y la tristeza, al contrario, cuando le pasa algo malo. Esto mismo, en el modelo propuesto, se puede traducir en que la felicidad y la tristeza, ambas emociones, están relacionadas con variaciones positivas o negativas del bienestar del agente:

$$\begin{array}{l} \text{if } \Delta Wb > L_h \Rightarrow \text{Felicidad} \\ \text{if } \Delta Wb < L_s \Rightarrow \text{Tristeza} \end{array} \tag{6.7}$$

Donde ΔWb es la variación del bienestar y $L_h \geq 0$ y $L_s \leq 0$ son las variaciones mínimas del bienestar del agente que producen felicidad o tristeza.

Tal y como se expuso en el capítulo 4, Rolls [Rolls, 2003] propone que las emociones son estados provocados por refuerzos (recompensa o castigo), de manera que nuestras acciones estarán dirigidas a obtener recompensas y evitar castigos. Siguiendo este punto de vista, en el sistema de toma de decisiones propuesto en esta tesis, se van a utilizar las emociones de felicidad y tristeza como refuerzo positivo y negativo respectivamente en el proceso de aprendizaje de selección de comportamientos.

El uso de las emociones felicidad y tristeza, como refuerzo en el sistema de aprendizaje, está además relacionado con el concepto de la reducción de los *drives* como refuerzo, tal y como se muestra en la sección 3.3.1. Esta relación viene dada por la definición de las emociones de felicidad y tristeza como variaciones positivas y negativas del bienestar, respectivamente. Las variaciones positivas, según (6.5), están relacionadas con la reducción de los *drives*, mientras que las negativas están relacionadas con su aumento.

6.4. Q-learning modificado para aprender a interactuar con objetos estáticos

Tal y como se ha expuesto, debido a la consideración de tomar el estado interno del agente como la motivación dominante, el entorno no cumple las condiciones de un proceso de Markov. A pesar de ello, para que el agente aprenda una política de comportamiento óptima, se va a utilizar el algoritmo Q-learning, introducido en la sección 3.4.1.

La simplificación realizada sobre los estados en relación a los objetos hace que, por ejemplo, el agente cuando tiene hambre aprende qué hacer con la comida ($s \epsilon S_{hambre} \times S_{comida}$) sin tener en cuenta su relación con el resto de objetos del entorno. Por lo tanto el estado total del agente en relación a cada uno de los objetos es:

$$s \epsilon S_{interno} \times S_{obj_i} \tag{6.8}$$

Considerando esta simplificación, la ecuación (3.5), que rige la actualización del valor $Q^{obj_i}(s,a)$ correspondiente a un par estado-acción para un estado interno, que define la motivación dominante, y un objeto i queda adaptada como:

$$Q^{obj_i}(s,a) = (1-\alpha) \cdot Q^{obj_i}(s,a) + \alpha \cdot \left(r + \gamma V^{obj_i}(s')\right) \tag{6.9}$$

Donde:

$$V^{obj_i}(s') = \underset{a \in A_{obj_i}}{\text{máx}} \left(Q^{obj_i}(s',a)\right) \tag{6.10}$$

es el valor del objeto i en el estado s', A_{obj_i} es el conjunto de acciones relacionadas con el objeto i y s' es el nuevo estado del agente en relación al objeto i. De nuevo, r es el refuerzo, γ es el factor de descuento y α es la tasa de aprendizaje.

Tal y como se expresa en (6.9), la simplificación hecha sobre los estados del agente hace que los valores Q aprendidos, en lugar de ser almacenados en una tabla de dimensión *número total de estados* $\times$ *numero total de acciones* para un estado interno determinado, queden almacenados en una tabla por cada objeto, cada una de dimensión: *número de estados en relación con el objeto* $\times$ *acciones posibles relacionadas con ese objeto.*

Utilizando el algoritmo Q-learning tradicional, esta actualización del valor Q, tiene en cuenta el efecto de la acción realizada, a, sobre el estado interno del agente, la motivación dominante y el estado del agente en relación al objeto i. Esta simplificación en el aprendizaje, implica que el valor de las acciones realizadas en relación a un objeto determinado son independientes de su relación con el resto de objetos que le rodean. Esto no es cierto, ya que por ejemplo, si el agente tiene hambre y sed, teniendo hambre con más urgencia, el hecho de comer estando ya en posesión de agua, no debería tener el mismo valor que comer y no tener agua. En otras palabras, el valor Q de estar en el estado "hambriento y tengo comida", y realizar la acción de comer, se calcula de la siguiente forma:

$$\begin{aligned} &Q^{comida}(\textit{hambre y tengo comida},\ \textit{comer}) = \\ &(1-\alpha) \cdot Q^{comida}(\textit{hambre y tengo comida},\ \textit{comer}) + \\ &\alpha \cdot \left(r + \gamma V^{comida}(\textit{sed y no tengo comida})\right) \end{aligned} \tag{6.11}$$

Donde:

$$V^{comida}(sed\ y\ no\ tengo\ comida) = \max_{a \in A_{comida}} \left(Q^{comida}(sed\ y\ no\ tengo\ comida, a)\right) \tag{6.12}$$

De acuerdo con esto, el valor que tiene el nuevo estado, que supongamos es "tener sed y no estar en posesión de comida", es independiente de la relación del agente con el resto de objetos. Sin embargo, el valor de este nuevo estado debería tener en cuenta si, por ejemplo, el agente ya está en posesión de agua o no.

Por lo tanto, para tener en cuenta estos "efectos colaterales" se propone una modificación del algoritmo de aprendizaje por refuerzo Q-learning, tal y como se muestra en la siguiente ecuación:

$$Q^{obj_i}(s,a) = (1-\alpha) \cdot Q^{obj_i}(s,a) + \alpha \cdot \left(r + \gamma \cdot V^{obj_i}(s')\right) \tag{6.13}$$

Donde:

$$V^{obj_i}(s') = \max_{a \in A_{obj_i}} \left(Q^{obj_i}(s',a)\right) + \sum_m \Delta Q^{obj_m}_{\text{máx}} \tag{6.14}$$

es el valor del objeto i en el estado nuevo, considerando los posibles efectos de la acción ejecutada con el objeto i, sobre el resto de objetos. Para ello, al valor del objeto i en el nuevo estado, definido previamente en la ecuación (6.10), se añade la suma de las variaciones de los valores-Q máximos de cada uno de los objetos restantes.

Estos incrementos se calculan de la siguiente manera:

$$\Delta Q^{obj_m}_{\text{máx}} = \max_{a \in A_{obj_m}} \left(Q^{obj_m}(s',a)\right) - \max_{a \in A_{obj_m}} \left(Q^{obj_m}(s,a)\right) \tag{6.15}$$

Cada uno de estos incrementos mide la diferencia entre lo mejor que el agente puede hacer en el estado nuevo y lo mejor que podía hacer en el estado anterior, para cada uno de los objetos. Siguiendo con el ejemplo anterior, el valor de estar en el nuevo estado queda modificado como:

$$\begin{aligned} &V^{comida}(sed\ y\ no\ tengo\ comida) = \\ &\text{máx}_{a \in A_{comida}} \left(Q^{comida}(sed\ y\ no\ tengo\ comida, a)\right) + \sum_m \Delta Q^{obj_m}_{\text{máx}} \end{aligned} \tag{6.16}$$

Suponiendo, en este ejemplo, que el agente tiene agua y que las variaciones sufridas en relación al resto de objetos son cero, salvo para el agua, su incremento se calcula como:

$$\Delta Q^{agua}_{\text{máx}} =$$

$$\text{máx}_{a \in A_{agua}} \left(Q^{agua}(sed\, y\, tengo\, agua, a)\right) - \text{máx}_{a \in A_{agua}} \left(Q^{agua}(hambre\, y\, tengo\, agua, a)\right) \tag{6.17}$$

En este caso, parece lógico pensar que el valor que tiene "tener agua" cuando se tiene sed, es mayor que el valor que tendría cuando se tiene hambre. Por ello, el incremento para este ejemplo, es positivo, lo que hace que el valor de estar en el nuevo estado aumente. Si el agente no estuviera en posesión de agua, posiblemente este incremento sería menor y por lo tanto también lo sería el valor de estar en el nuevo estado.

Una vez realizada la actualización del valor $Q^{obj_i}(s, a)$, el agente tiene que seleccionar una nueva acción a realizar. Para ello, si sigue una política óptima, se considera la acción a que maximice el valor Q^{obj_i} para la motivación dominante actual, considerando todos los objetos y por lo tanto todas las acciones disponibles.

$$a^* = \arg \max_{a \in A} Q^{obj_i}(s, a) \quad \forall i \tag{6.18}$$

6.5. Q-learning modificado para aprender a interactuar con objetos activos

Previamente se ha mostrado cómo el algoritmo Q-learning se modifica para tener en cuenta los denominados "efectos colaterales". De esta misma manera, los algoritmos de aprendizaje multiagente descritos en la sección 3.4.1, también son modificados tal y como se muestra a continuación.

$$Q^{obj_i}(s, a_1, a_2) = (1 - \alpha) \cdot Q^{obj_i}(s, a_1, a_2) + \alpha \cdot \left(r + \gamma \cdot V^{obj_i}(s')\right) \tag{6.19}$$

1. Algoritmo Amigo-Q:

$$V^{obj_i}(s') = \max_{a_1 \in A_1} \max_{a_2 \in A_2} \left(Q^{obj_i}(s', a_1, a_2)\right) + \sum_m \Delta Q^{obj_m}_{\text{máx}} \tag{6.20}$$

Se elige la acción tal que:

$$a_1^* = \arg_{a_1} \max_{a_1 \in A_1} \max_{a_2 \in A_2} \left(Q^{obj_i}(s', a_1, a_2)\right) \tag{6.21}$$

2. Algoritmo Enemigo-Q:

$$V^{obj_i}(s') = \max_{a_1 \in A_1} \min_{a_2 \in A_2} \left(Q^{obj_i}(s', a_1, a_2)\right) + \sum_m \Delta Q^{obj_m}_{\max} \tag{6.22}$$

Se elige la acción tal que:

$$a_1^* = \arg_{a_1} \max_{a_1 \in A_1} \min_{a_2 \in A_2} \left(Q^{obj_i}(s', a_1, a_2)\right) \tag{6.23}$$

3. Algoritmo Media-Q:

$$V^{obj_i}(s') = \max_{a_1 \in A_1} \left(\sum_{a_2 \in A_2} Q^{obj_i}(s', a_1, a_2) \Big/ n \right) + \sum_m \Delta Q^{obj_m}_{\max} \tag{6.24}$$

De manera que el agente escogerá la acción a_1 que maximize dicho vector:

$$a_1^* = \arg_{a_1} \max_{a_1 \in A_1} \left(\sum_{a_2 \in A_2} Q^{obj_i}(s', a_1, a_2) \Big/ n \right) \tag{6.25}$$

Estos incrementos se calculan utilizando la ecuación (6.15) de la misma manera que con Q-learning.

6.6. Miedo

Con los mecanismos normales de aprendizaje se aprenden los valores medios de las acciones propias o exógenas. De esta manera, si una acción es nociva frecuentemente, el agente aprende a evitarla. Sin embargo, cuando los efectos negativos de una acción son ocasionales, se necesitan otros mecanismos para poder aprender a evitar estos riesgos. Este es el papel del miedo en este sistema de toma de decisiones: es un mecanismo adicional para poder aprender a evitar las acciones, tanto propias como exógenas, que causan efectos negativos pero con muy baja probabilidad.

El miedo se produce cuando existe la posibilidad de que algo malo puede pasar [Ortony, 2003]. Esto significa que el bienestar del agente podría disminuir. En este sistema, para hacer frente al miedo, se va a tener en cuenta la acción que produce el efecto negativo. Se va a considerar que existen dos tipos de miedo, uno relacionado con las acciones ejecutadas por el agente y el otro relacionado con las acciones exógenas llevadas a cabo por otros elementos del entorno, como otros agentes.

Cuando una acción propia del agente tiene malos resultados frecuentemente, el agente, utilizando los mecanismos normales de aprendizaje, aprende que no debe elegirla, ya que el valor de la acción es muy bajo debido a los refuerzos negativos.

Por ejemplo, hay acciones, como caminar por un borde muy fino, que tienen grandes probabilidades de provocar un efecto negativo, caernos. Si hemos ido diez veces por el borde y nos hemos caído siete, al final, tendremos miedo de ir por el borde y procuraremos no volver a hacerlo. Un caso muy diferente es cuando una acción no siempre es nociva, sólo lo es ocasionalmente. Esto se puede ver con el siguiente ejemplo: puede ocurrir que a una persona le gusten las setas, normalmente sientan bien y quitan el hambre, pero si en alguna ocasión sufre una intoxicación grave, debido a una seta en mal estado, probablemente no vuelva a probarlas a pesar de que sólo le hicieron daño una vez.

Por otro lado, si la acción que produce el efecto negativo no es ejecutada por el agente, es decir, es una acción exógena, entonces el agente no tiene control de la situación y siente miedo. Estas acciones exógenas pueden ser realizadas por otros agentes, e incluso, podrían ser cualquier objeto activo capaz de realizar acciones. Este miedo es, por ejemplo, el miedo que se siente a estar en una habitación oscura. Cuando estamos en una habitación oscura tenemos miedo a que nos pase algo ya que no vemos nada y no se tiene control sobre lo que puede ocurrir. Esto no significa que todas las veces anteriores que se estuvo en una habitación oscura pasara algo malo, basta con que ocurriese alguna vez. El miedo a estar en ese estado, que se puede considerar "peligroso", ayuda a evitar dichos estados.

En las siguientes secciones se va a mostrar la manera en que el miedo, relacionado con las acciones propias y el relacionado a las acciones exógenas, se implementa en este sistema de toma de decisiones.

6.6.1. Tener miedo a realizar acciones arriesgadas

Cuando se tiene miedo a realizar una acción es porque alguna vez, o muchas veces, esta acción produjo un efecto nocivo. Este efecto se traduce en una disminución del bienestar y, por lo tanto, en un refuerzo negativo.

Para considerar el miedo a realizar esta acciones, en los experimentos de esta tesis se almacenan los peores resultados experimentados por el agente para cada par estado-acción, en una variable llamada $Q^{obj_i}_{peor}(s,a)$, la cual será actualizada después de la ejecución de la acción.

$$Q^{obj_i}_{peor}(s,a) = \text{mín}(Q^{obj_i}_{peor}(s,a), r + \gamma \cdot V^{obj_i}_{peor}(s')) \tag{6.26}$$

Donde:

$$V^{obj_i}_{peor}(s') = \underset{a \in A_{obj_i}}{\text{máx}} (Q^{obj_i}_{peor}(s',a)) \tag{6.27}$$

es el valor peor del objeto i en el nuevo estado y se considera lo mejor que se puede hacer con los peores valores Q. Los peores valores se calculan como el refuerzo recibido y el valor peor del objeto en el nuevo estado.

Estos valores se calculan para cada objeto del mundo, con independencia del estado interno y de los otros objetos. La idea es que, por ejemplo, si el agente tiene hambre y se come una seta que le sienta muy mal, el miedo está relacionado sólo con la seta, independientemente del resto de los objetos. Por lo tanto, en el cálculo de los peores valores, no se van a tener en cuenta los "efectos colaterales" introducidos en (6.14).

Esta manera de calcular los peores valores, considerando el nuevo peor estado de cada par estado-acción permite que, no sólo la acción que produjo un refuerzo negativo importante tenga un valor bajo de $Q^{obj_i}_{peor}(s,a)$, sino también las acciones que le llevaron a ese estado. Es decir, siguiendo el ejemplo anterior, si se tiene miedo a comer una seta, no sólo no se come, sino que tampoco se coge, ni se cocina.

El efecto de tener miedo puede ser considerado como elegir la acción que maximiza $Q^{obj_i}_{miedo}$, en lugar de elegir aquella que maximiza Q^{obj_i},

$$\begin{aligned} &Q^{obj_i}_{miedo}(s,a) = \beta \cdot Q^{obj_i}(s,a) + (1-\beta) \cdot Q^{obj_i}_{peor}(s,a) \longrightarrow Si\ Q^{obj_i}_{peor}(s,a) \leq L_m \\ &Q^{obj_i}_{miedo}(s,a) = Q^{obj_i}(s,a) \longrightarrow En\ otro\ caso \end{aligned} \tag{6.28}$$

Donde L_m es un límite de manera que, sólo cuando lo peor que le ha pasado al agente sea realmente malo, es cuando va a temer realizar esa acción. No tendría sentido que se considerasen todos los peores valores, sino sólo aquellos que han supuesto realmente un descenso importante en el bienestar del agente.

Usando esta aproximación, el resultado esperado de cada acción es considerado al mismo tiempo que el menos favorable. El parámetro β, siendo $0 \leq \beta \leq 1$, mide el grado de atrevimiento del agente y su valor dependerá de la personalidad del agente. Si el agente no tiene miedo de nada, es muy atrevido, β será casi 1; mientras que para un agente miedoso, el cual siempre trata de minimizar el riesgo, β será próximo a 0. Si $\beta = 1$ el agente está usando la política asignada por el Q-learning.

Cuando las acciones provocan refuerzos negativos frecuentemente, el valor Q obtenido con el Q-learning será similar al valor Q peor y por eso no serán elegidas de nuevo. Sin embargo, cuando las acciones son nocivas ocasionalmente, los valores Q y los peores serán muy distintos. En este caso si el agente no considerase lo peor que le ha pasado al realizar esa acción entonces la volvería a elegir, pero si lo considera lo más probable es que no lo haga de nuevo. Por lo que el hecho de considerar el peor resultado obtenido para cada objeto, es un mecanismo que completa el Q-learning para tratar con este tipo de acciones. En el capítulo 10 se mostrará la utilidad del miedo a realizar estas acciones arriesgadas.

6.6.2. Tener miedo a un estado "peligroso"

Un estado peligroso es aquel en el que el agente sabe que le puede pasar algo malo y no lo controla, ya que no depende de sus propias acciones. Es decir, puede que al agente le pase algo malo sin que él haya hecho nada. Cuando el agente sabe que

puede sufrir efectos negativos en un estado, como consecuencia de eventos exógenos, éste siente miedo. El miedo se expresa ahora como un *drive*: D_{miedo}.

El *drive* Miedo es tratado de la misma forma que el resto de los *drives* y, por lo tanto, su motivación asociada puede ser la dominante. En este caso, el agente aprenderá por sí mismo qué hacer cuando tiene miedo.

Por ejemplo, si vamos caminando por la calle y nos encontramos con alguien que nos pega, sin motivo, nosotros no tenemos control sobre esa acción. El castigo recibido no es debido a ninguna acción nuestra, sino que depende de nuestro acompañante. El resultado es que al final tendremos miedo de estar junto a esa persona, ya que nos puede hacer daño. El miedo, como motivación, va a dirigir nuestro comportamiento. Esta es la idea que se quiere implementar en esta parte de la tesis.

Como se ha mostrado previamente, las acciones exógenas pueden ser realizadas por cualquier objeto activo que pueda ejecutar acciones. En esta tesis se va a considerar que las acciones exógenas van a ser ejecutadas por otro agente. Estas acciones exógenas pueden ocurrir al mismo tiempo que cualquiera de las acciones del agente. Por lo que los efectos negativos de esos eventos exógenos serán reflejados en todas las acciones del agente. Para separar los efectos de las acciones del agente y los efectos de las acciones exógenas, el estudio se va a centrar en el agente cuando está interaccionando con otro agente y él "no esté haciendo nada", o lo que es lo mismo cuando ejecute la acción de "esperar". En ese caso, se supondrá que todos los cambios sufridos por el agente son consecuencia de elementos externos.

Para tratar con este tipo de miedo, también se van a tener en cuenta los peores valores Q, de manera similar a (6.26) y (6.27), pero para el caso multiagente:

$$Q^{obj_i}_{peor}(s, a_1, a_2) = \text{mín}(Q^{obj_i}_{peor}(s, a_1, a_2), r + \gamma \cdot V^{obj_i}_{peor}(s')) \tag{6.29}$$

Donde:

$$V^{obj_i}_{peor}(s') = \max_{a \in A_{obj_i}} (Q^{obj_i}_{peor}(s', a_1, a_2)) \tag{6.30}$$

es el valor peor del objeto i en el nuevo estado. De nuevo estos valores se calculan con independencia del estado interno del agente, es decir, lo que importa es lo peor que ha pasado con ese objeto, que en este caso es el oponente.

Se considerará que un estado es un estado "peligroso" cuando:

$$\min_{a_2} Q^{obj_i}_{peor}(s, Nada, a_2) < L_{miedo} \tag{6.31}$$

siendo L_{miedo} el valor mínimo aceptable del peor valor esperado cuando el agente no está haciendo nada. En este caso, el valor del *drive* Miedo se incrementa.

En el caso contrario, cuando:

$$\min_{a_2} Q^{obj_i}_{peor}(s, Nada, a_2) > L_{miedo} \tag{6.32}$$

se considera que el agente está en un estado "seguro" y el valor del *drive* Miedo se reduce a cero.

La idea es que si cuando el agente está acompañado por otro agente, y lo peor que le ha pasado mientras él no hacia nada, es menor que un cierto límite, entonces el agente tiene miedo. En ese momento el *drive* Miedo toma un valor alto y entonces podría pasar a ser la motivación dominante.

De nuevo, el miedo va a ser muy útil cuando el oponente, que se porta mal con el agente, no lo hace frecuentemente. Es decir, si el oponente siempre se porta mal, el agente no necesita del miedo para saber que no debe interactuar con él. El problema aparece cuando de media, el oponente es bueno y sólo a veces es malo. En este caso, el hecho de considerar lo peor que le ha pasado estando con él, hace que el agente le tenga miedo y entonces aprenda qué acciones tomar.

El miedo en este caso está relacionado con estar en un estado peligroso. Si el agente tiene miedo al estar acompañado, su *drive* aumenta, provocando un descenso del bienestar, pero si se va y deja de estar acompañado, el *drive* Miedo se reduce a cero, por lo que hay un refuerzo positivo. Es de esperar, por lo tanto, que el agente aprenda que cuando tiene miedo a estar en ese estado peligroso, lo mejor es moverse. Tal y como se mostrará en el capítulo 10, el agente aprende comportamientos de escape cuando tiene miedo a estar en un estado peligroso.

7. PROCEDIMIENTO EXPERIMENTAL

7.1. Introducción

El objetivo final de esta tesis es el diseño de un sistema de toma de decisiones para un robot autónomo y social. Como paso previo a la implementación de ese sistema en un robot real, esta tesis va a ser desarrollada utilizando agentes virtuales, en lugar de robots. El agente vive en un mundo virtual donde existen objetos, los cuales necesita para sobrevivir, y otros agentes.

7.2. Descripción del entorno virtual

Bellman en [Bellman, 2003] propone el uso de mundos virtuales como plataformas de experimentación para sus trabajos con agentes artificiales. Una de las cosas más importantes que se necesitan es un entorno en el que se puedan estudiar las relaciones entre objetivos, las capacidades del agente, los comportamientos, las interacciones con el entorno y las consecuencias o resultados en ese entorno. Con los mundos virtuales esto se puede conseguir, aunque por supuesto, estos mundos no son tan ricos como los mundos reales.

Los mundos virtuales surgen a partir de tres importantes líneas de desarrollo y experiencia: (1) Juegos de rol en red y multi-usuario llamados MUVEs "Multi-user virtual environments" (Entornos virtuales multi-usuario); (2) Entornos de realidad virtual y simulación distribuida avanzada, especialmente aquellas usadas en ejercicios de entrenamiento militar; (3) Entornos de computación distribuidos, incluyendo internet.

El uso de estos mundos virtuales como plataformas de experimentación se está extendiendo mucho por la comunidad robótica y de inteligencia artificial. Por ejemplo en [Isbell et al., 2001] se presentan los estudios de aprendizaje por refuerzo de un agente artificial, el cual "vive" en un entorno multi-usuario llamado LambdaMOO. Este entorno es uno de los más antiguos juegos de rol multi-usuario basados en texto, que está formado por habitaciones interconectadas, pobladas de usuarios y objetos que se pueden mover de habitación en habitación. Los mecanismos de interacción social son diseñados para reforzar la ilusión de que el usuario está presente en el espacio virtual. Otro ejemplo del uso de juegos de ordenador como plataforma de experimentación es el trabajo presentado en [Thomaz and Breazeal, 2006], en el que los jugadores interactivamente entrenan a un robot virtual para realizar una tarea. De la misma forma que en LabdaMOO, es un jugador externo el que da el refuerzo al agente para que aprenda a realizar una tarea en concreto o a sobrevivir.

7.2.1. El mundo virtual: Coffeemud

Los MUDs, "Multi-User Domains" (Dominios multi-usuario), fueron desarrollados originalmente en 1979 y se refieren a juegos multi-usuario basados en texto centrados en el género de la aventura fantástica como "Dragones y Mazmorras". Para desarrollar el mundo virtual, se decidió utilizar este tipo de juegos basados en texto en lugar de utilizar uno de los más modernos, con vistosas interfaces visuales. Esto se debió a que se necesitaba un juego que permitiera una manera sencilla de enviar y adquirir información del mundo. Usando un juego basado en texto, para este robot virtual o agente adquirir información es equivalente a leer texto, y actuar (moverse, coger cosas, etc.) es mandar texto, lo que permite un tratamiento más sencillo de los datos. Además, un MUD ofrece la forma ideal de crear escenarios virtuales (áreas) con todos los objetos que se necesiten (comida, agua, medicina) para interactuar.

Entre muchos y diferentes códigos de MUD's disponibles en la red, finalmente se decidió elegir el llamado CoffeeMud [Zimmerman, 2007], basado en Java, ya que el software está muy bien documentado y proporciona explicaciones y manuales de uso claros.

En un MUD típico, una persona se conecta a un servidor MUD utilizando un cliente Telnet. Como se quería que los agentes viviesen en el mundo, se crearon varios programas en C, los cuales se conectan a un servidor a través de la interfaz Telnet simulando varios jugadores. Estos agentes se comportan de acuerdo al sistema de decisiones propuesto en esta tesis.

Para hacer los experimentos se decidió crear el área llamado *Passage*. *Passage* fue diseñada de una manera similar a la planta del Departamento de Ingeniería de sistemas y automática de la Universidad Carlos III de Madrid, es decir un pasillo largo con despachos a ambos lados, ya que la aplicación principal futura es la implementación de esta arquitectura en un robot real, el cual se moverá por dicho entorno.

7.2.2. Agentes en el área *Passage*

Este área está formada por 20 habitaciones, 8 de ellas constituyen un pasillo y el resto son oficinas distribuidas a ambos lados del pasillo. En este área, el agente puede encontrarse como se ha dicho, con distintos objetos. Estos objetos pueden ser pasivos, que no pueden ejecutar acciones, o activos, que ejecutan acciones.

Los objetos que existen en este mundo son:

- Comida (pasivo)
- Agua (pasivo)
- Medicina (pasivo)
- Mundo (pasivo)
- Otro Agente (activo)

A excepción de los agentes, los cuales se van moviendo de manera autónoma, los objetos están distribuidos por algunas habitaciones del área, de manera que existen habitaciones con comida, otras con agua y otras con medicinas. La cantidad de objetos es muy alta de manera que se considera que son recursos ilimitados. El agente, cuando comienza el juego, desconoce dónde encontrar los objetos. Es a medida que juega cuando se va encontrando con los objetos y recordando su posición, de manera que si en un futuro los necesita, el agente recordará dónde encontrarlos.

No existen puertas entre las habitaciones de este área, de manera que la forma en la que el agente se mueve es dando un comando de "dirección": norte, sur, este y oeste. Con una orden de movimiento, el agente pasa de una habitación a otra. Los comandos utilizados para interactuar con los objetos activos y estáticos serán descritos posteriormente, ya que forman el conjunto de acciones disponibles para cada agente.

Vale la pena decir que existen dos comportamientos que son "explorar" e "ir a" los cuales utilizan dos tipos de algoritmos matemáticos. En el caso de la exploración es un algoritmo DFS (Deep First Search) el cual da una ruta de exploración desde la habitación donde se encuentra el agente, recorriendo todas las habitaciones del área. Para ir a una habitación determinada por el camino más corto, entre la habitación origen y la final, el agente utiliza el algoritmo Dijkstra.

7.2.3. Interfaz gráfica

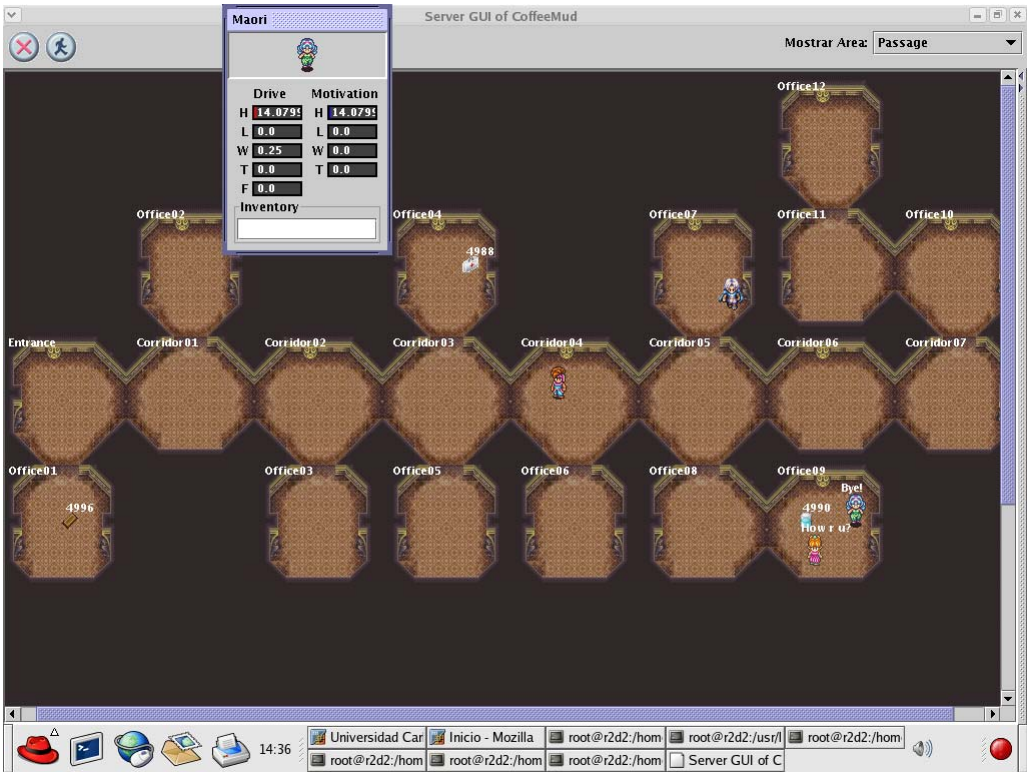


Fig. 7.1: *Interfaz gráfica del entorno*

Debido a la naturaleza del MUD, la interacción entre el agente y el juego está basada en texto. Aunque es bastante fácil detectar objetos en el área, resulta difícil tener una visión global de todo el juego ya que cada agente sólo ve la habitación en la que se encuentra. En nuestro caso, se necesitaba tener esa visión global de todos los agentes en el área para saber cómo está yendo el juego y cómo lo está haciendo cada uno de ellos. Por este motivo se desarrolló una interfaz gráfica, la cual se muestra en la figura 7.1.

Con esta interfaz gráfica se pueden seguir todas las acciones de los agentes, además de ver los valores de sus *drives* y motivaciones y los objetos que tiene en su poder gracias a una sub-ventana, ver la figura 7.2, desarrollada con ese objetivo, para cada agente.

Fig. 7.2: *Ventana del agente*

7.3. Descripción del agente

7.3.1. *Drives*

Como ya se ha dicho en el capítulo anterior, los *drives* y motivaciones considerados son los siguientes:

- Hambre
- Sed
- Debilidad
- Soledad
- Miedo

Estos *drives* han sido seleccionados teniendo en cuenta que este sistema de toma de decisiones está implementado en un agente artificial. Un jugador típico de este tipo de juegos tiene que comer y beber para sobrevivir. Los *drives* Debilidad y Soledad han sido implementados para incrementar la complejidad de la arquitectura de toma de decisiones. Ya que el objetivo final de esta tesis doctoral es diseñar un sistema de toma de decisiones para agentes autónomos y sociales, la necesidad de relacionarse está incluida como una de las necesidades internas del agente.

Algunos *drives*, necesidades del agente, después de ser satisfechos, no vuelven a crecer inmediatamente sino que sólo a partir de un cierto tiempo éstos vuelven a aumentar. Este tiempo se denomina tiempo de satisfacción. Esto sucede de la misma forma que cuando se come, no se tiene hambre de nuevo hasta unas cuantas horas después.

En el agente se introducen unos tiempos de satisfacción para algunos de los *drives*, los cuales dependerán de la urgencia de cada uno. Estos *drives* van a ser Hambre, Sed y Soledad. Como ya se ha visto antes, los *drives* Debilidad y Miedo van a seguir una dinámica diferente.

$$\begin{aligned} T_{sed} &= 50 \\ T_{hambre} &= 100 \\ T_{soledad} &= 150 \end{aligned} \tag{7.1}$$

De acuerdo con esto, el *drive* Sed va a ser la más urgente, ya que tarda menos tiempo en volver a crecer. En general, como ya se mostró en la sección 6.2.1, se tiene sed con más frecuencia que se tiene hambre. La necesidad social, el *drive* Soledad, tarda mucho más tiempo en volver a aumentar su valor, no es una necesidad muy urgente. Una vez que estos tiempos de satisfacción se cumplen, los *drives* aumentan de la siguiente forma:

$$\begin{aligned} D^{k+1}_{hambre} &= D^{k}_{hambre} + 0{,}08 \\ D^{k+1}_{sed} &= D^{k}_{sed} + 0{,}1 \\ D^{k+1}_{soledad} &= D^{k}_{soledad} + 0{,}06 \end{aligned} \tag{7.2}$$

Como se puede ver, el *drive* Sed crece a un ritmo más alto que el *drive* Hambre y éste a su vez, más alto que Soledad. La variación del *drive* Debilidad a cada paso que da el agente es la siguiente:

$$D^{k+1}_{debilidad} = D^{k}_{debilidad} + 0{,}05 \tag{7.3}$$

Además, algunos *drives* se ven afectados por las acciones ejecutadas por otro agente. De hecho, cuando al agente le roban algo:

$$D^{k+1}_{soledad} = D^{k}_{soledad} + 1 \tag{7.4}$$

y, cuando al agente le pegan:

$$\begin{aligned} D^{k+1}_{soledad} &= D^{k}_{soledad} + 1 \\ D^{k+1}_{debilidad} &= D^{k}_{debilidad} + 3 \end{aligned} \tag{7.5}$$

Para el *drive* Miedo, el cambio no es progresivo sino que:

$$\begin{aligned} &\text{Si el agente está en un estado “seguro” entonces } D^{k+1}_{miedo} = 0 \\ &\text{Si el agente está en un estado “peligroso” entonces } D^{k+1}_{miedo} = 5 \end{aligned} \tag{7.6}$$

7.3.2. Motivaciones del agente

Según la ecuación (6.2), las motivaciones están definidas como una suma de los *drives* y de los estímulos externos. Estos estímulos w_i son los distintos objetos que el agente se puede encontrar durante el juego. De manera que:

$$\begin{aligned} &\text{Si el estímulo está presente entonces } w_i \neq 0 \\ &\text{Si el estímulo no está presente entonces } w_i = 0 \end{aligned} \tag{7.7}$$

En la tabla 7.1 se muestran las motivaciones, los *drives*, excepto Miedo, y sus estímulos motivacionales, o incentivos.

Tab. 7.1: *Motivaciones,* Drives *y estímulos motivacionales*

Motivación/*Drive*	Estímulo motivacional
Hambre	Comida
Sed	Agua
Debilidad	Medicina
Soledad	Otro agente

En la ecuación (6.3) se describe la aplicación de los niveles de activación L_d para calcular el valor de las motivaciones. En los experimentos:

$$L_d = 2 \tag{7.8}$$

7.3.3. Bienestar

En relación al bienestar del agente, que como ya se visto, es una función de sus *drives*, adaptando la ecuación (6.5) al diseño del agente, queda que:

$$Wb = Wb_{ideal} - (\alpha_1 D_{hambre} + \alpha_2 D_{sed} + \alpha_3 D_{debilidad} + \alpha_4 D_{soledad} + \alpha_5 D_{miedo}) \tag{7.9}$$

Los factores de personalidad α_i, son los que valoran la importancia de cada uno de los *drives* en el bienestar del agente. Para los experimentos, todos los *drives* van a tener la misma importancia, por lo que todos los factores son iguales entre sí:

$$\alpha_i = 1 \tag{7.10}$$

7.3.4. Estado del agente

De acuerdo con la sección 6.2.1 en este escenario el estado interno del agente está definido como:

$$S_{interno} = \{Hambriento,\ Sediento,\ Débil,\ Solo,\ Asustado,\ OK\} \tag{7.11}$$

En relación con los objetos estáticos, el agente puede estar en los siguientes estados:

$$S_{obj} = Estar\_en\_posesión\_de \times Estar\_cerca\_de \times Saber\_dónde\_hay \tag{7.12}$$

Y en relación con otro agente:

$$S_{obj} = Estar\_cerca\_de \tag{7.13}$$

Todas las variables son evaluadas como $= \{si, no\}$.

En nuestro entorno, vamos a considerar que existe un objeto distinto a los anteriores, el "mundo", de manera que el estado del agente en relación al mundo, por ahora, es único:

$$S_{obj} = Estar\_en \tag{7.14}$$

Por lo tanto, según la definición de estado dada por la ecuación (6.4), el agente podría estar, por ejemplo, en el estado: "hambriento, no tengo comida, no estoy junto a comida y sé dónde hay comida".

7.3.5. Acciones del agente

Los conjuntos de las acciones que el agente puede realizar, dependiendo de su estado en relación a los objetos, son los siguientes:

$$A_{comida} = \{Comer,\ Coger,\ Ir\ a\} \tag{7.15}$$

$$A_{agua} = \{Beber,\ Coger,\ Ir\ a\} \tag{7.16}$$

$$A_{medicina} = \{Tomar\ medicina,\ Coger,\ Ir\ a\} \tag{7.17}$$

$$A_{otro\ agente} = \begin{cases} Robar\ comida/agua/medicina \\ Dar\ comida/agua/medicina \\ Saludar \\ No\ hacer\ nada \\ Pegar \end{cases} \tag{7.18}$$

$$A_{mundo} = \{Quedarse\ quieto,\ Explorar\} \tag{7.19}$$

Entre estos comportamientos hay algunos que reducen o aumentan algún *drive*, tal y como se refleja en la tabla 7.2, produciendo una variación en el bienestar del agente.

Tab. 7.2: *Efectos de las acciones sobre los* drives

Acción	*Drive*	Efecto
Comer	Hambre	Reducir a cero (satisfacción del *drive*)
Beber Agua	Sed	Reducir a cero (satisfacción del *drive*)
Tomar medicina	Debilidad	Reducir a cero (satisfacción del *drive*)
Que te saluden	Soledad	Reducir a cero (satisfacción del *drive*)
Que te roben algo	Soledad	Incrementar cierta cantidad
Que te den algo	Soledad	Reducir a cero (satisfacción del *drive*)
Que te peguen	Soledad	Incrementar cierta cantidad
Que te peguen	Debilidad	Incrementar cierta cantidad
Explorar/Ir a	Debilidad	Incrementar cierta cantidad

La acción de "no hacer nada" no tiene ningún efecto en los *drives* de ninguno de los agentes que interactúan.

Por supuesto, como estas acciones afectan al nivel de los *drives*, dependiendo de si la variación del bienestar es lo suficientemente importante, provocará una emoción de acuerdo con la ecuación (6.7).

7.4. Selección de comportamientos

En los siguientes capítulos se van a presentar distintos experimentos en los que el agente, variando alguna de sus características, tendrá que aprender a sobrevivir en distintos tipos de entorno.

Al comienzo de cada experimento los valores iniciales de las matrices-Q son cero. El agente, a medida que va viviendo en el mundo, va explorando todas las acciones e irá actualizando los valores de dichas matrices. Como ya se ha comentado en el capítulo anterior, la exploración aleatoria tarda mucho en centrarse en las mejores acciones, así que en lugar de eso se va a utilizar un método que combina exploración y explotación.

El método de exploración específica que se va usar es un estándar que fue usado, obteniendo buenos resultados, en [Watkins, 1989]. El agente prueba las acciones de manera probabilística basándose en los valores Q, usando la distribución de Boltzmann, definida en el capítulo anterior, pero que será recordada a continuación:

$$P_s(a) = \frac{e^{\frac{Q(s,a)}{T}}}{\sum_{b \in A} e^{\frac{Q(s,b)}{T}}} \tag{7.20}$$

Esta es la probabilidad con la que el agente ejecuta una acción a, en un estado s. La temperatura T controla la cantidad de exploración, es decir, la probabilidad de ejecutar acciones distintas a las que tienen el mayor valor Q. Si T es alto, o si todos los valores Q son iguales, se elegirá una acción aleatoria. Si T es bajo y los valores Q son diferentes, esto hará que se tienda a elegir la acción con el valor Q más alto.

A la hora de elegir el valor del parámetro T que, como se acaba de ver, va a determinar la aleatoriedad en la elección de acciones, en varios experimentos se puso en evidencia que este valor es dependiente de los valores de Q. Por lo que en esta tesis se propone que si se quiere mantener una aleatoriedad fija, se tiene que definir este parámetro T como una función del valor medio de los valores Q. De manera que:

$$T = \delta * \text{valor medio de Q} \tag{7.21}$$

En los experimentos el parámetro que se va a ajustar para determinar la aleatoriedad es el parámetro δ. Tal y como se muestra en (7.21), un valor alto de δ favorecerá la exploración. Por el contrario, un valor bajo favorecerá la explotación de las acciones que hayan resultado más favorables.

7.5. Protocolo de interacción social

Cuando dos agentes se encuentran, tiene lugar la siguiente secuencia o protocolo de interacción:

1. Si alguno de los dos quiere interactuar, le preguntará al otro: ¿Quieres interaccionar conmigo?

2. Si el otro agente quiere interactuar dirá "Sí", si no quiere, puede irse simplemente o realizar otra acción.

3. Si quiere interactuar: El que preguntó realiza su acción y después él realiza la suya.

4. Después se despiden.

Si alguno de los dos quiere volver a interactuar con el otro, se empieza de nuevo el protocolo. En el caso de que los dos pregunten simultáneamente, existe una espera aleatoria y se repite hasta que uno de los dos lo haga primero. Tres o más agentes no pueden interactuar entre sí.

7.6. Indicadores de análisis de resultados

En [Ávila García and Cañamero, 2002], se definen unos indicadores de viabilidad para poder comparar las actuaciones de los robots cuando utilizaban diferentes arquitecturas de toma de decisiones. Estos indicadores eran: el tiempo de vida, el confort global, el balance fisiológico y la cantidad de tiempo en la que existió riesgo de muerte. Tomando como referencia estos indicadores, se van a definir otros para analizar los resultados de nuestros experimentos. Para ello hay que tener en cuenta que en los experimentos que se van a llevar a cabo, los agentes no mueren, sino que tienen un tiempo de vida fijo. Por lo tanto, la actuación del agente se va a estudiar mediante el análisis de su bienestar, lo que da una idea acertada de lo bien que le ha ido durante cada experimento.

Antes se va a definir un concepto importante en este trabajo: Zona Segura. Se define la Zona Segura como un intervalo de valores del bienestar, de manera que se puede considerar que si el bienestar del agente se encuentra dentro de dicho intervalo, el agente lo está "haciendo bien".

De acuerdo con la ecuación (6.5), el valor ideal del bienestar sera Wb_{ideal}, es decir, cuando todos los *drives* sean iguales a cero. En los experimentos este valor se ha fijado a $Wb_{ideal} = 100$ por conveniencia. También se ha indicado en (6.3) que para todos los *drives*, si su valor es inferior a un cierto límite L_d, su motivación no entra en competencia con las otras. De nuevo, por conveniencia, este límite como se ha mostrado está fijado $L_d = 2$. Por lo tanto, mientras todos los *drives* tengan un valor inferior a este límite, podemos considerar que el agente no tiene ninguna "necesidad" urgente, es decir, no existe una motivación dominante. Si excluimos al *drive* Miedo, definimos el intervalo de la Zona Segura como $ZS = [100, 92]$.

Para analizar la actuación del agente en cada experimento, se han definido dos indicadores para medir la actuación del agente:

1. Valor medio del Bienestar: Este indicador muestra el valor medio alcanzado durante el experimento. Este indicador da una idea general de lo bien o mal que lo ha hecho el agente, pero no proporciona una idea clara acerca de su calidad de vida, ya que un valor medio bueno puede ser debido tanto a una buena actuación general, como a una combinación de muy buenos y muy malos momentos.

2. Porcentaje de permanencia en la Zona Segura: Con este indicador se tiene una idea más clara sobre la calidad de vida del agente durante el experimento. Lo que interesa no es que el agente tenga sólo un buen valor medio del bienestar sino que además haya vivido bien.

Será la combinación de estos dos indicadores lo que indique si una estrategia de comportamiento es mejor que otra.

8. RESULTADOS EXPERIMENTALES: AGENTE SOLITARIO

8.1. Descripción del experimento

En este primer capítulo de resultados experimentales se describe el comportamiento de un agente viviendo en solitario en el mundo previamente descrito.

En los capítulos anteriores se han descrito todos los parámetros que definen el comportamiento del agente. Los valores finalmente elegidos, lo han sido después de un proceso de selección. La mejor manera de ver qué valores son los que van a hacer que el agente se comporte de manera óptima, es variar un parámetro mientras los otros permanecen constantes. Se ha decidido empezar por realizar este ajuste mientras el agente está solo, ya que cuando está con otros agentes, como se mostrará en el siguiente capítulo, existen demasiadas variables en el entorno.

Primero se va a justificar el uso de la variación del bienestar, es decir, la felicidad y tristeza, como función de refuerzo, en lugar de utilizar el propio bienestar, tal y como usan otros autores [Gadanho and Custodio, 2002a]. A continuación se ajustarán los valores de los parámetros relacionados con el algoritmo de aprendizaje: α la tasa de aprendizaje, δ que va a definir la política de exploración/explotación de los comportamientos y γ, el factor de descuento. Al final de este capítulo también se expondrá cómo afecta el valor de los estímulos motivacionales.

Para llevar a cabo estos experimentos, se ha anulado el crecimiento, y por lo tanto el efecto, de los *drives* Soledad y Miedo. Esto es debido a que cuando el agente está sólo no tiene sentido considerar el *drive* Soledad, mientras que el *drive* Miedo se añadirá en experimentos posteriores.

8.2. Refuerzo: Bienestar vs felicidad/tristeza

Tal y como se mostró en el capítulo 5, en los estudios realizados por Gadanho [Gadanho and Custodio, 2002a], una señal de bienestar es generada de manera similar que en nuestro esquema y es utilizada como función de refuerzo dentro de un esquema de aprendizaje por refuerzo.

La función de refuerzo que se propone en esta tesis es la felicidad y tristeza, que se definen como las variaciones de la función de bienestar definidas en (6.7). Tomando

los límites L_h y L_s, descritos en esta ecuación, como nulos, la felicidad y la tristeza quedan definidas finalmente como:

$$\begin{array}{l} \text{if } \Delta Wb > 0 \Rightarrow \text{Felicidad} \\ \text{if } \Delta Wb < 0 \Rightarrow \text{Tristeza} \end{array} \tag{8.1}$$

Como ya se ha mostrado en la sección 6.3.1, esta variación es justamente la diferencia del valor del bienestar del agente entre dos pasos de simulación:

$$\Delta Wb^{k+1} = Wb^{k+1} - Wb^{k} \tag{8.2}$$

Al comienzo de las investigaciones realizadas en esta tesis, parecía lógico pensar en el bienestar como función de refuerzo, ya que da una idea de cómo se encuentra el agente después de realizar una acción. Sin embargo, tomando como base los estudios sobre las emociones mostrados en el capítulo 4, y tal como ha sido explicado en la sección 6.3.2, parece más apropiado el uso de la felicidad y tristeza, la variación del bienestar, como función de refuerzo. De hecho, esta variación da una idea bastante más exacta de cómo ha afectado una acción al bienestar del agente.

Con el fin de comparar el comportamiento de un agente en solitario utilizando ambas funciones de refuerzo, se han realizado dos experimentos fijando los siguientes parámetros del agente:

- El valor de la tasa de aprendizaje α, que define la velocidad de aprendizaje, $\alpha = 0,3$.
- El factor de descuento, $\gamma = 0,8$.
- El parámetro δ, que es el que va a determinar la aleatoriedad a la hora de la elección de los comportamientos, también va a permanecer constante, $\delta = 1$.
- El valor del estímulo motivacional, $w_i = 1$.
- El valor del límite de activación de la motivación, $L_d = 2$.

En la figura 8.1, se muestra el bienestar del agente cuando se utiliza como función de refuerzo la propia señal de bienestar. Como se puede apreciar, durante toda la vida del agente, el bienestar es variable continuamente, a pesar de que finalmente parece que se "estabiliza" dentro de un rango de $Wb \, \epsilon \, [40, 100]$.

Según parece, el agente no aprende una buena política de comportamiento ya que, aunque el valor medio del bienestar es 73,9, el porcentaje de permanencia en la zona de seguridad es tan sólo 13,5 %. De hecho, en la figura 8.2 se muestran los valores Q de las acciones de "comer", "beber medicina" y "beber agua", cuando el agente tiene sed. Como se observa, la acción que tiene más valor no corresponde a "beber agua", sino a "beber medicina".

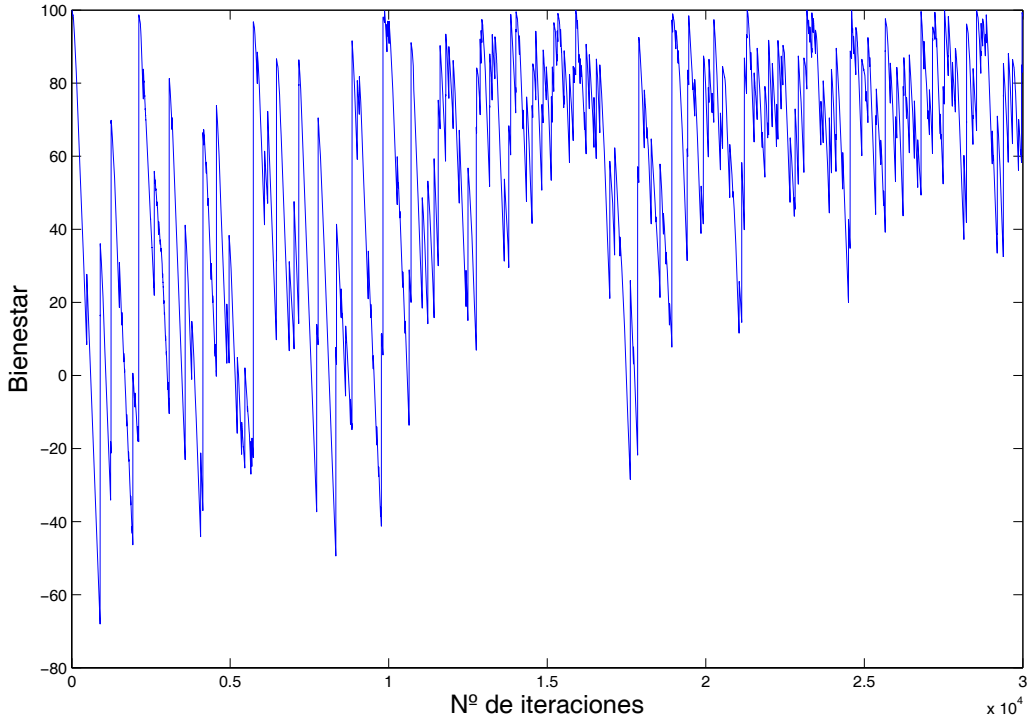


Fig. 8.1: *Bienestar del agente utilizando la señal del bienestar como función de refuerzo*

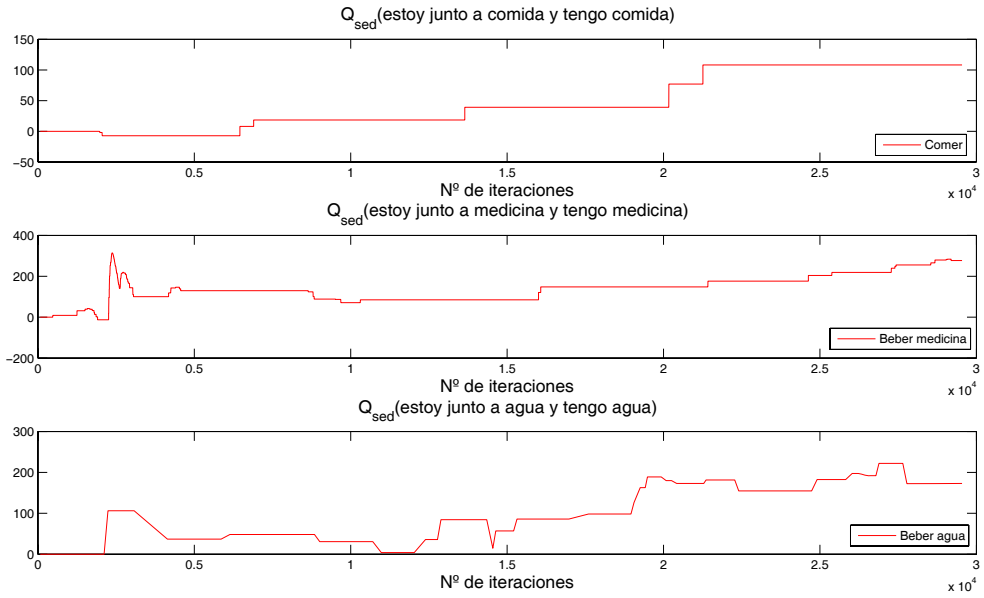


Fig. 8.2: *Valores Q cuando el agente tiene sed y utiliza el bienestar como refuerzo*

Sin embargo, cuando se utilizan las emociones de felicidad y tristeza como función de refuerzo, la señal de bienestar permanece siempre dentro de un rango de valores bastante más aceptable $Wb \epsilon [90, 100]$, ver la figura 8.3. De hecho, el valor medio del bienestar es 98,7 y el porcentaje de permanencia en la zona de seguridad es del 100 %.

Esto indica que el agente sí que aprende una política de comportamiento correcta, es decir, la secuencia de acciones que le llevan a satisfacer el *drive* correspondiente con la motivación dominante. En la figura 8.4 se muestran, de nuevo, los valores Q de las acciones de "comer", "beber medicina" y "beber agua" cuando el agente tiene sed. En esta ocasión el valor de "beber agua" es significativamente más alto que los de "comer" y "beber medicina".

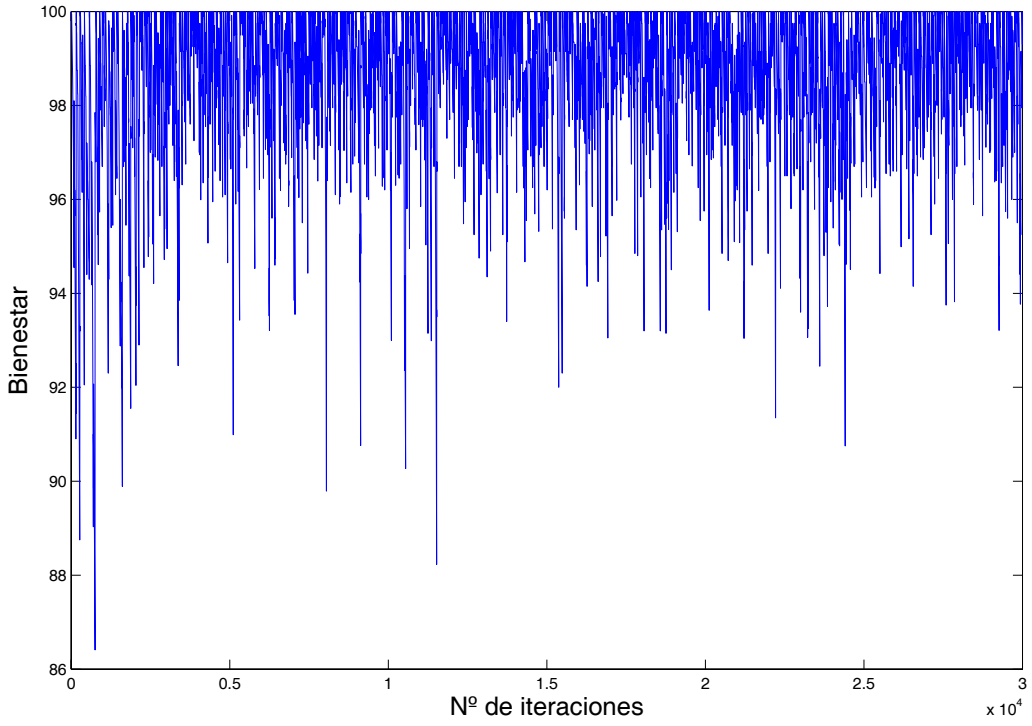


Fig. 8.3: *Bienestar del agente utilizando la felicidad y tristeza como función de refuerzo*

A la vista de los resultados, se llegó a la conclusión de que la mejor función de refuerzo es la variación del bienestar del agente. Conviene recordar que en ambos experimentos, los valores de los parámetros que definen la velocidad de aprendizaje α, y el nivel de exploración/explotación δ, están fijos a lo largo de toda la vida del agente. Por lo tanto es razonable que no se observe una tendencia más pronunciada de crecimiento de la señal del bienestar, propia de una explotación de las políticas de comportamiento aprendidas, sino que en ambos gráficos dicha tendencia sea bastante suave.

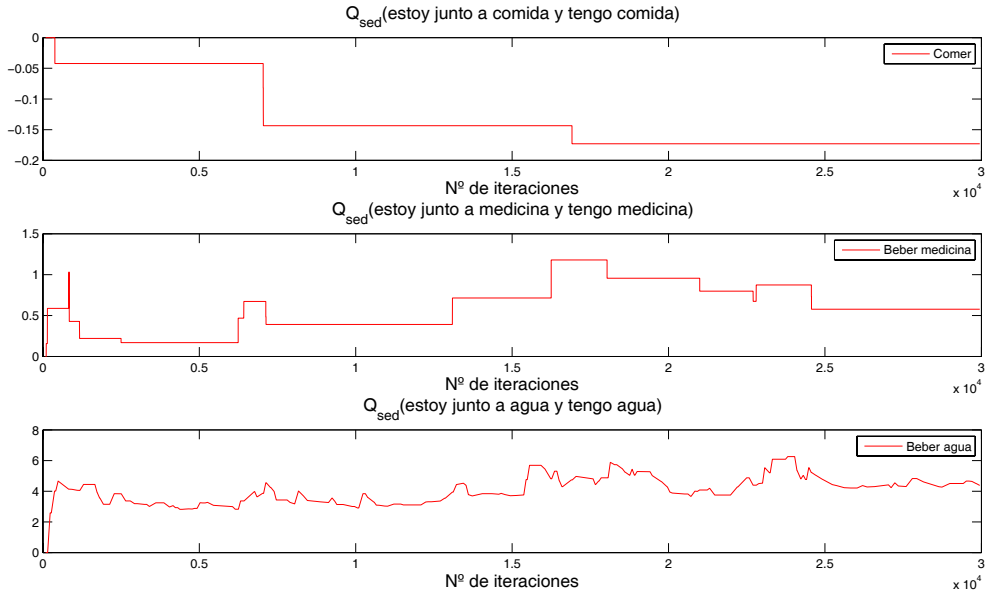


Fig. 8.4: *Valores Q cuando el agente tiene sed y utiliza la felicidad y la tristeza como refuerzo*

8.3. Hay que pasarlo mal en la juventud para poder aprender

La justificación de esta afirmación se va a enfocar desde el punto de vista de dos parámetros básicos en el aprendizaje por refuerzo:

- El parámetro δ, que es el que va a determinar el nivel de exploración/explotación de las acciones.
- La tasa de aprendizaje α .

Está claro que para aprender una buena política de comportamiento, se tienen que probar todas las acciones posibles en cada estado y decidir cuál es la mejor. En cada experiencia, el agente va actualizando los valores Q de cada par estado-acción. Finalmente, aquellas acciones que sean las más adecuadas en cada estado tendrán un valor Q alto. Según (3.4), la política óptima elige la acción que maximiza $Q(s, a)$, por lo que, si no se explorasen todas las acciones, no se podría asegurar la idoneidad de cada una de ellas en cada estado.

Para asegurar que se exploran todas las acciones, se aumenta el parámetro δ. Este parámetro fue introducido en la sección 7.4 y determina la aleatoriedad a la hora de seleccionar una acción. Cuando su valor es alto, todas las acciones tienen la misma probabilidad de ser elegidas, sin preferencia entre ellas. Esto provoca que, como se prueban todas las acciones posibles, algunas de ellas no son las adecuadas y el bienestar del agente, como consecuencia, disminuye. Por lo que es cierto que, para aprender la mejor política de comportamiento, hay que pasarlo mal durante un

tiempo. Esto se puede ver reflejado en la figura 8.5 donde se muestra la señal de bienestar del agente cuando funciona con un $\delta = 1{,}8$ durante toda su vida. Este valor escogido de δ, se ha tomado como suficientemente alto para explorar todas las acciones disponibles del agente en este mundo en concreto. Posiblemente, en un mundo más complejo, con más acciones entre las que elegir, este parámetro deberá aumentar su valor.

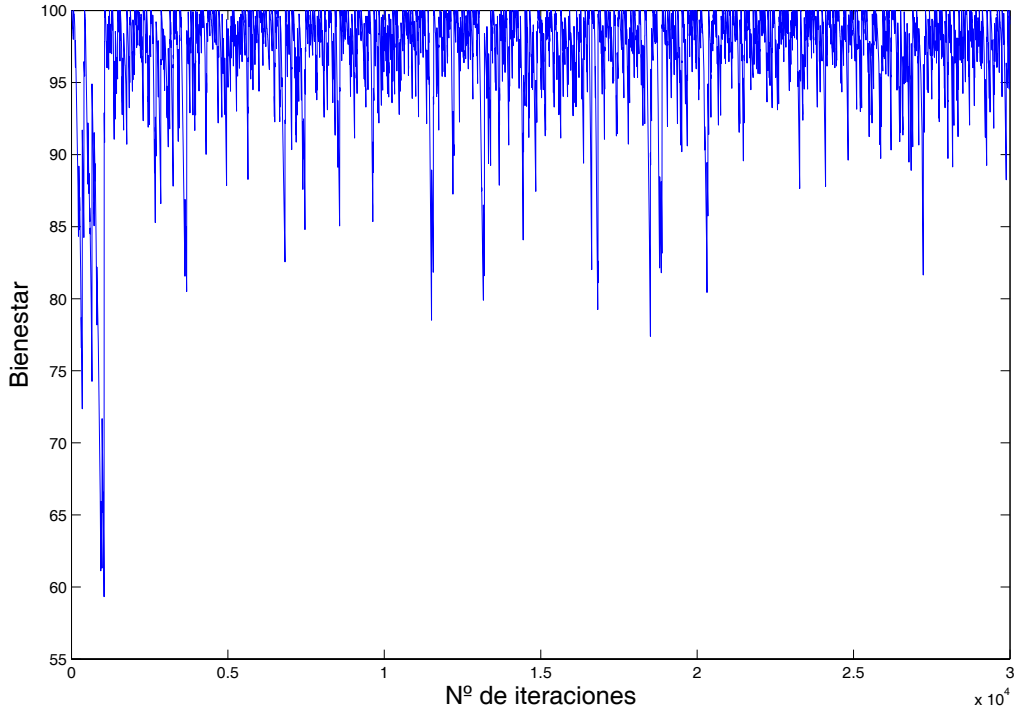


Fig. 8.5: *Bienestar del agente utilizando un valor alto de δ, lo que favorece la exploración de todas las acciones*

Si en lugar se escoger un valor alto de δ, se elige uno pequeño, $\delta = 0{,}05$, el agente no llega a explorar todas las acciones y por lo tanto no podrá aprender bien. Tal y como se muestra en la figura 8.6, con un valor bajo de δ, la exploración de las acciones es pequeña y las acciones exploradas comienzan a ser explotadas desde el principio sin haberse asegurado que dichas acciones son las mejores. Esto provoca que en este caso, como se aprecia en el gráfico, el bienestar del agente no deja de disminuir. Este continuo decrecimiento del bienestar es debido, tal y como se muestra en la figura 8.7, que muestra la evolución de los *drives* del agente, a que el *drive* Hambre no se ha satisfecho nunca. Sin embargo, se puede apreciar que el *drive* Debilidad a partir de un cierto momento permanece constante, mientras que el *drive* Sed está continuamente satisfecho. Lo que indica que el agente se queda junto a agua, la coge y la bebe, sin moverse y así hasta el final, mientras, el *drive* Hambre sigue creciendo.

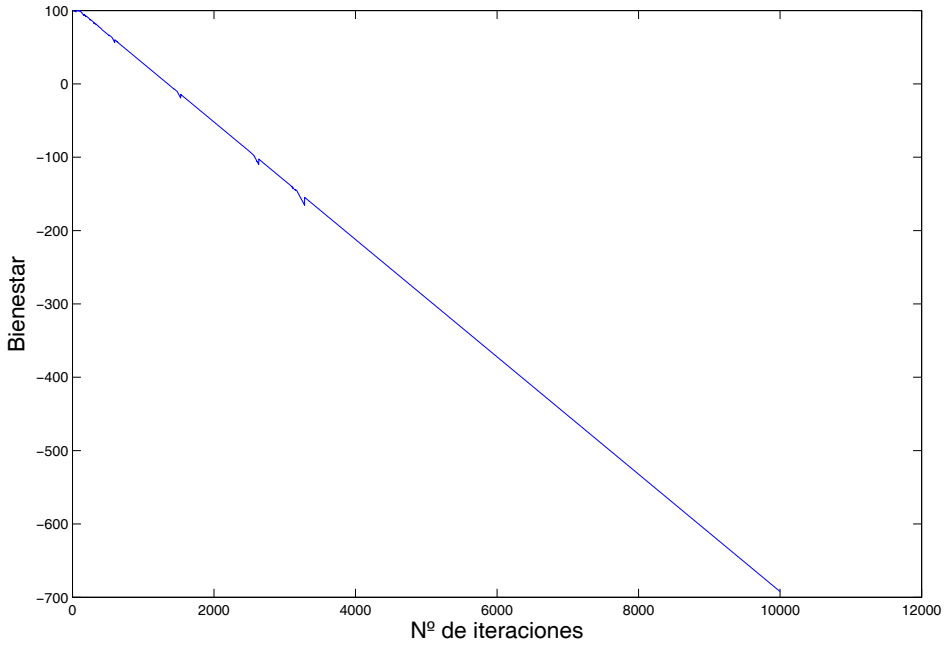


Fig. 8.6: *Bienestar del agente utilizando un valor bajo de δ, favoreciendo la explotación de las acciones en lugar de su exploración*

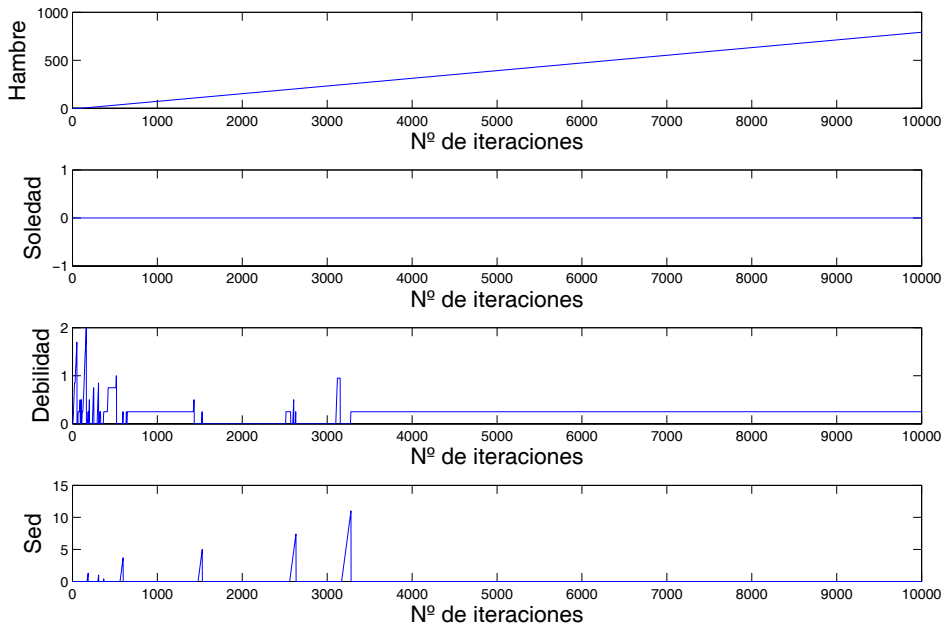


Fig. 8.7: Drives *del agente utilizando un valor bajo de δ*

Esto, probablemente, es debido a que al comienzo de su vida el agente ejecutó las acciones de "explorar", "quedarse quieto", "ir a por agua", "beber agua", al igual que las de "ir a por medicina" y "beber medicina". Debido a que el factor δ es tan bajo, comenzó a explotarlas obteniendo buenos refuerzos por las reducciones de los *drives* Debilidad y Sed, siendo mayores los de Sed, lo que hizo que los valores Q correspondientes creciesen. Cuanto mayores fueron esos valores, menores fueron las probabilidades de que las acciones no exploradas, relacionadas con la comida, fueran elegidas.

Como conclusión, lo ideal sería una combinación de ambos efectos, es decir, que cuando todas las acciones han sido exploradas, el agente comience a explorar aquellas que efectivamente han sido las más convenientes en cada estado. Para ello, el parámetro δ tiene que disminuir su valor después de que, debido a un valor alto de δ, el agente haya explorado todas las acciones. De esta manera, las acciones que hayan resultado más favorables tienen una mayor probabilidad de ser elegidas.

La tasa de aprendizaje α se presentó en la ecuación (3.5), donde se define la actualización de los valores Q del aprendizaje por refuerzo Q-learning. El efecto de este parámetro es dar más o menos importancia a los valores aprendidos ante las nuevas experiencias vividas. Un valor bajo de α, implica que el agente es más conservador y por lo tanto da más importancia a las experiencias pasadas. Si el valor de α es alto, cercano a 1, el agente valorará en mayor medida la experiencia más reciente.

Como ya se ha visto, es necesario que para que el agente aprenda a vivir de manera óptima, al comienzo de su vida el parámetro δ debe tener un valor alto de forma que se exploren todas las posibles acciones. Por otro lado, el valor del parámetro α tiene que ver con la variabilidad de los resultados de las experiencias vividas. Es decir, puede ocurrir que una misma acción ejecutada en el mismo estado, tenga refuerzos diferentes.

Un valor alto de este parámetro, $\alpha = 0{,}8$, provoca cambios muy bruscos en los valores aprendidos de Q, tal y como se muestra en la figura 8.8. Por otro lado, un valor muy bajo de este parámetro, $\alpha = 0{,}1$, hará que el aprendizaje sea muy lento, ya que el agente será muy conservador y dará muy poca importancia a las nuevas experiencias, ver la figura 8.9. Lo mejor, por lo tanto, sería un valor intermedio de este parámetro. Este valor intermedio, debería asegurar una buena relación entre la variabilidad de los valores Q y la importancia de las nuevas experiencias.

8.4. Cuando se vive bien conviene renunciar a seguir aprendiendo

Como conclusión del apartado anterior, lo mejor para aprender una buena política de comportamiento es que durante la primera etapa de "juventud" y aprendizaje, el agente pruebe todas las acciones y aprenda de ellas, utilizando un valor alto de δ, $\delta = 1{,}8$, y un valor intermedio de la tasa de aprendizaje, $\alpha = 0{,}3$.

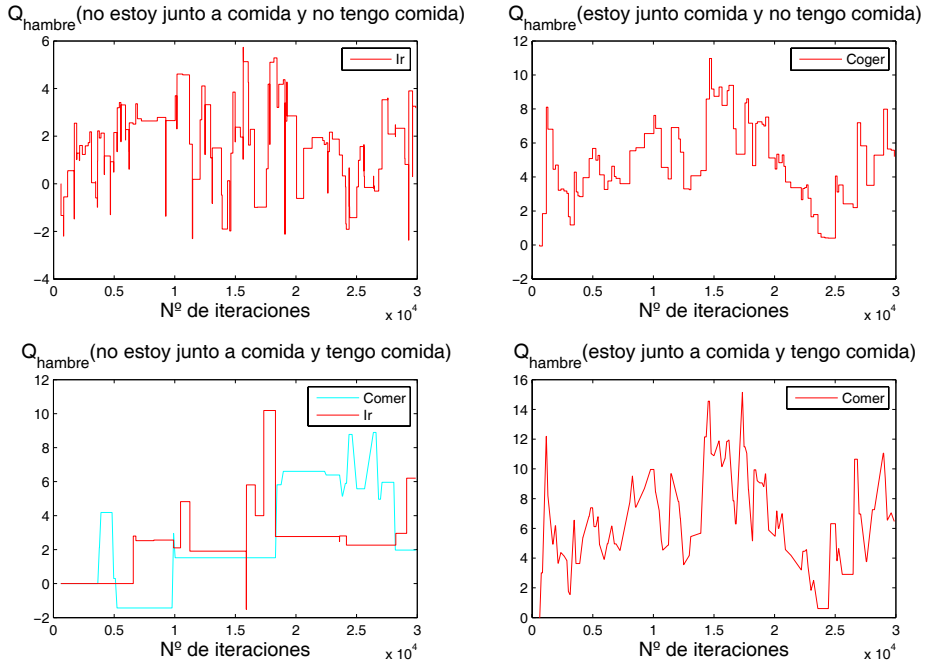


Fig. 8.8: *Valores Q de las acciones relacionadas con comida cuando el agente tiene hambre, con un valor alto de* α

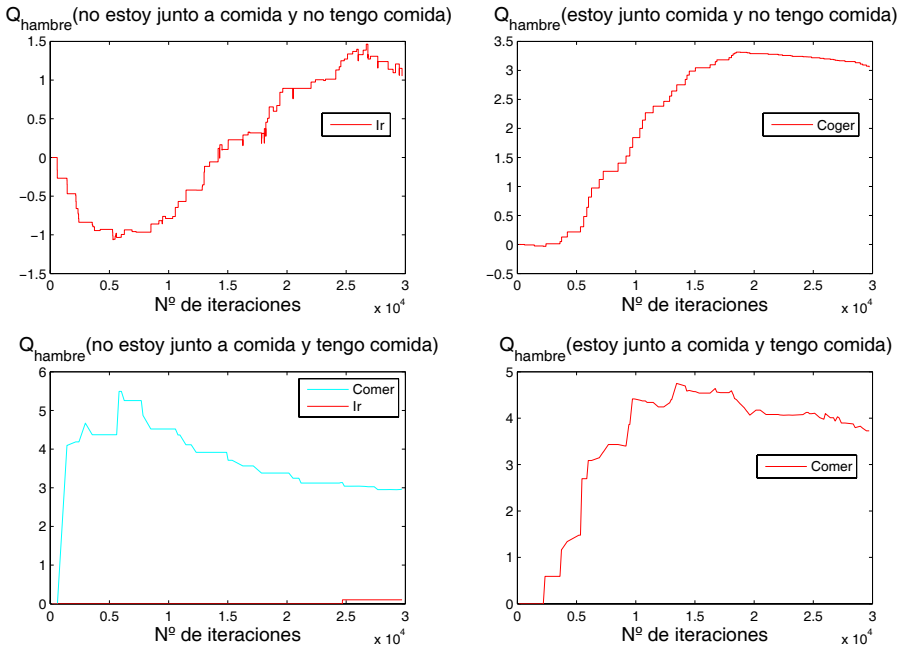


Fig. 8.9: *Valores Q de las acciones relacionadas con comida cuando el agente tiene hambre, con un valor bajo de* α

Una vez que se considera esta etapa de aprendizaje terminada, comienza la etapa de "madurez". Se considera, para los experimentos que esta etapa de "madurez" comienza en la iteración número 15000. Durante esta etapa se disminuye el valor del parámetro δ, $\delta = 0{,}1$, para favorecer ahora la explotación de las mejores acciones.

Si el parámetro α se dejara fijo durante esta última etapa, es decir, si el agente sigue aprendiendo se puede observar, ver la figura 8.10, que el bienestar disminuye en lugar de aumentar. Esto es debido a que el agente "olvida" lo aprendido al vivir mejor. Es decir, cuando el agente comienza a explotar las mejores acciones, los refuerzos ahora recibidos son mucho menores que antes, ya que no pasa tanta necesidad.

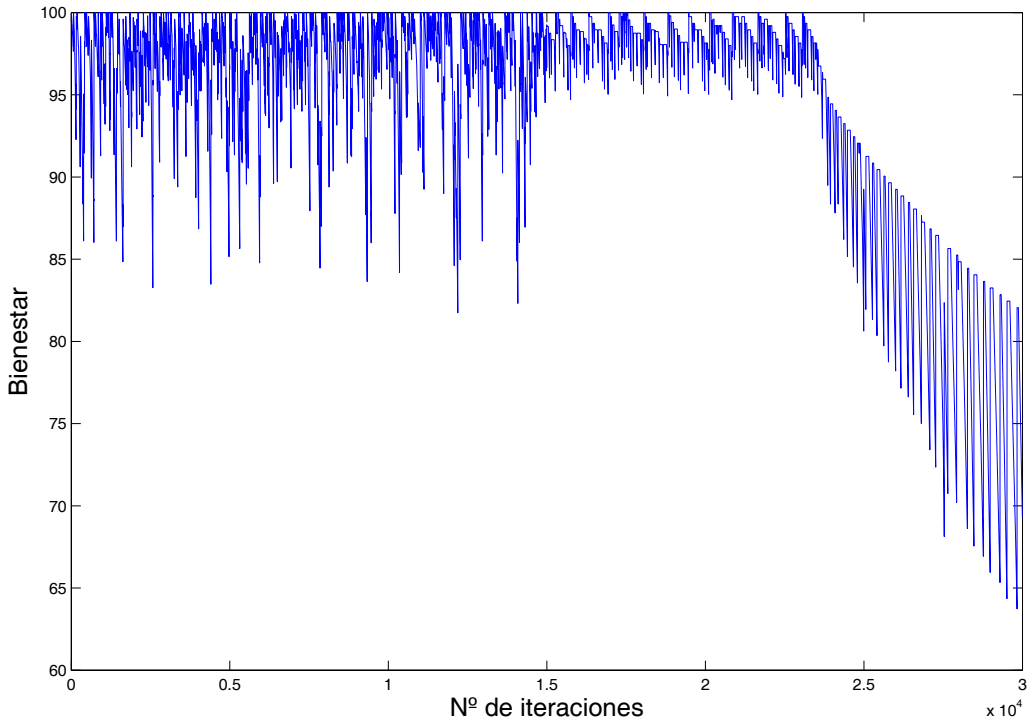


Fig. 8.10: *Bienestar del agente cuando se aprende en dos fases, una de exploración, $\delta = 1{,}8$, y otra de explotación, $\delta = 0{,}1$, pero manteniendo la tasa de aprendizaje constante $\alpha = 0{,}3$*

Si el agente sigue aprendiendo, los valores Q se verán afectados y por lo tanto aparece el efecto mostrado en la figura figura 8.11. En este gráfico se representa la variación del valor Q correspondiente a la motivación dominante Debilidad cuando el agente está junto a comida y sabe dónde hay medicina. Como se observa en el sub-gráfico superior, el valor que tiene la acción "ir a por medicina" va disminuyendo, hasta que se llega a un punto en el que este valor es inferior al valor de "coger comida" (sub-gráfico inferior). Debido a que el agente sigue aprendiendo, estos dos valores Q se van recalculando y tomando distintos valores, tal y como se muestra en dicha figura.

Este efecto no deja de ser curioso, ya que efectivamente, en la vida real una vida acomodada no favorece el aprendizaje de supervivencia, al tener todas las necesidades básicas cubiertas.

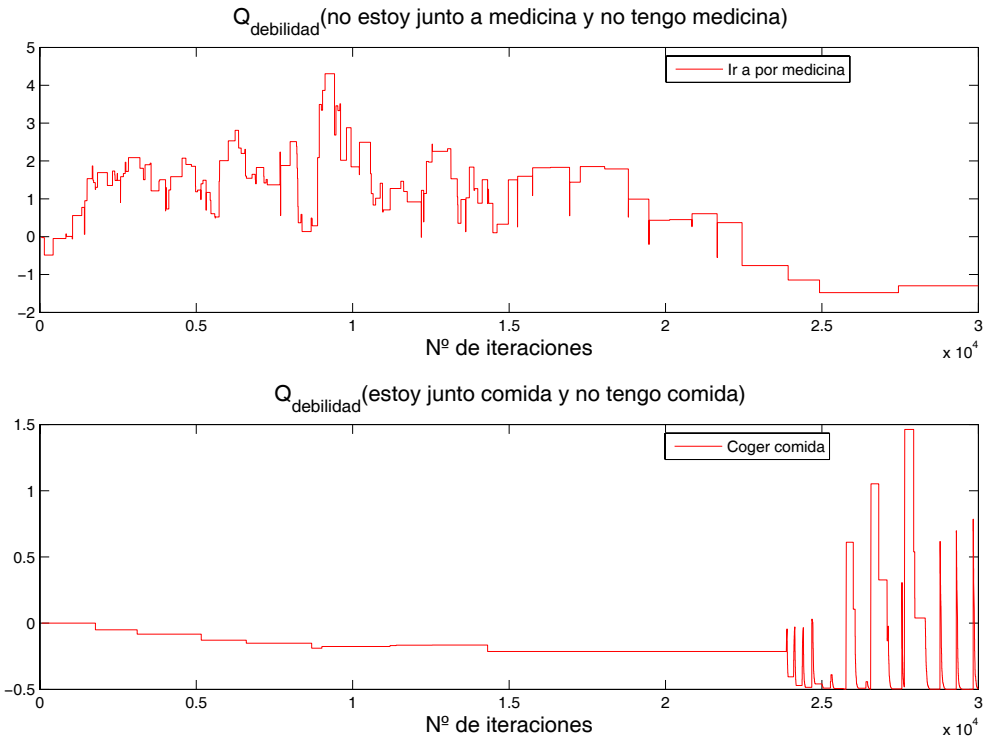


Fig. 8.11: *Valores Q de las acciones relacionadas con comida cuando la motivación dominante es Debilidad, cuando se aprende en dos fases, una de exploración, $\delta = 1{,}8$, y otra de explotación, $\delta = 0{,}1$, manteniendo la tasa de aprendizaje constante $\alpha = 0{,}3$*

Como consecuencia de este hecho, se llega a la conclusión de que una vez que el agente decide explotar las mejores acciones ($\delta = 0{,}1$) y por lo tanto vivir mejor, debe renunciar a seguir aprendiendo ($\alpha = 0$). En la figura 8.12, se puede observar cómo, siguiendo esta estrategia, durante la segunda etapa de la vida del agente, el bienestar del agente aumenta y permanece estable.

En los resultados que se han mostrado, la vida del agente se ha separado en dos fases, una de aprendizaje, "juventud", y otra de explotación de lo aprendido, "madurez". En la realidad, ambas fases no están claramente diferenciadas sino que la vida es un proceso de aprendizaje progresivo. A partir de ahora, se va a considerar la vida como una etapa larga de aprendizaje progresivo, en la que los valores de α y δ van a variar de forma lineal desde valores altos a valores bajos. Al final de la vida del agente, se va a realizar una fase en la que el agente vive utilizando lo aprendido. En la figura 8.13 se muestra el bienestar del agente durante estas dos fases.

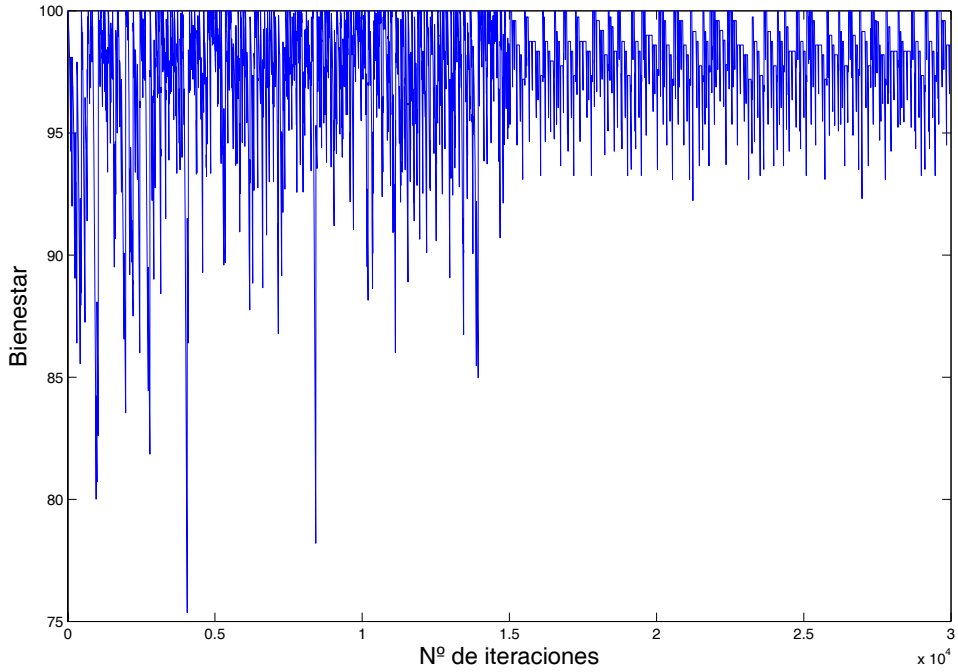


Fig. 8.12: *Bienestar del agente cuando se aprende en dos fases, una de exploración y aprendizaje, $\delta = 1{,}8$ y $\alpha = 0{,}3$, y otra de explotación de lo aprendido, $\delta = 0{,}1$ y $\alpha = 0$*

Finalmente cada experimento consta de dos fases: *fase de aprendizaje* y *fase permanente*. Durante la fase de aprendizaje, el agente comienza con todos los valores-Q iniciales iguales a cero. A medida que el agente va "viviendo" en el mundo, irá aprendiendo de su propia experiencia, e irá actualizando sus valores de las matrices-Q. Una vez que la fase de aprendizaje termina, comienza la fase de permanencia. En esta fase el agente "vive" de acuerdo a los valores-Q aprendidos. En la figura 8.14, se muestra cómo varían los valores de los parámetros δ y α durante las dos fases.

8.5. No es recomendable buscar prioritariamente la felicidad inmediata

Existe otro parámetro utilizado en el aprendizaje Q-learning (6.13), el cual define cuánto afectan actualmente las recompensas futuras. Este es el parámetro γ, denominado factor de descuento. Tal y como fue explicado en la sección 3.4.1, un valor alto de este parámetro va a dar más importancia a las recompensas que se puedan recibir en el futuro. Un valor bajo, por el contrario, no se interesa por lo que pueda pasar en el futuro. A continuación se muestra de nuevo la ecuación (6.13) de actualización del valor Q de un par estado-acción $Q^{obj_i}(s, a)$:

$$Q^{obj_i}(s,a) = (1-\alpha) \cdot Q^{obj_i}(s,a) + \alpha \cdot \left(r + \gamma \cdot V^{obj_i}(s')\right) \tag{8.3}$$

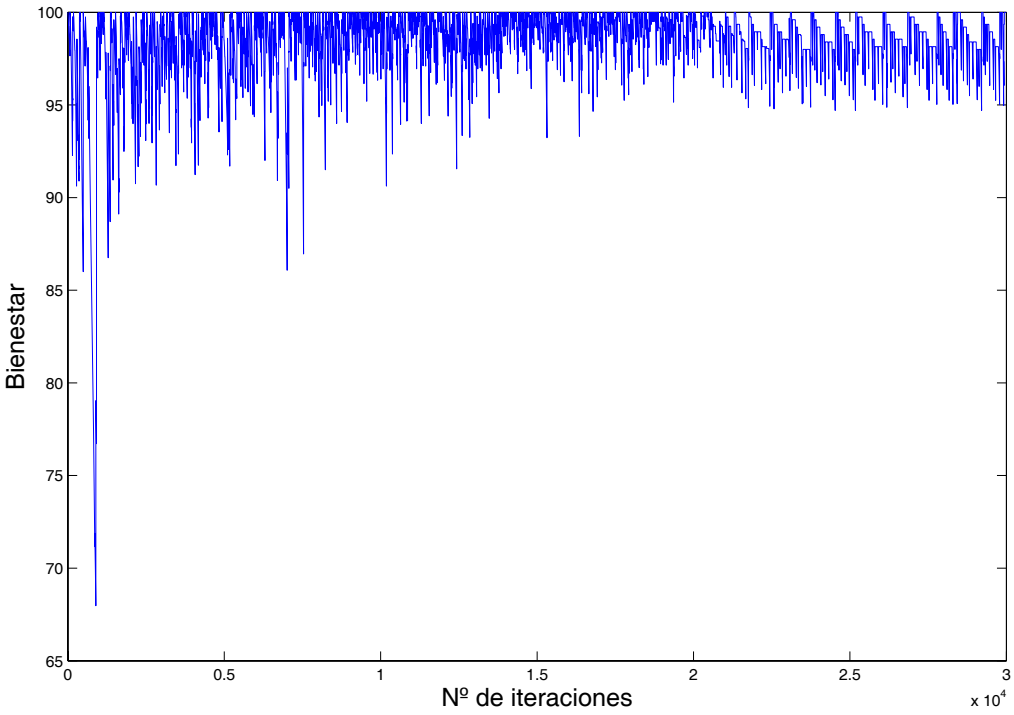


Fig. 8.13: *Bienestar del agente cuando δ y α varían de forma progresiva*

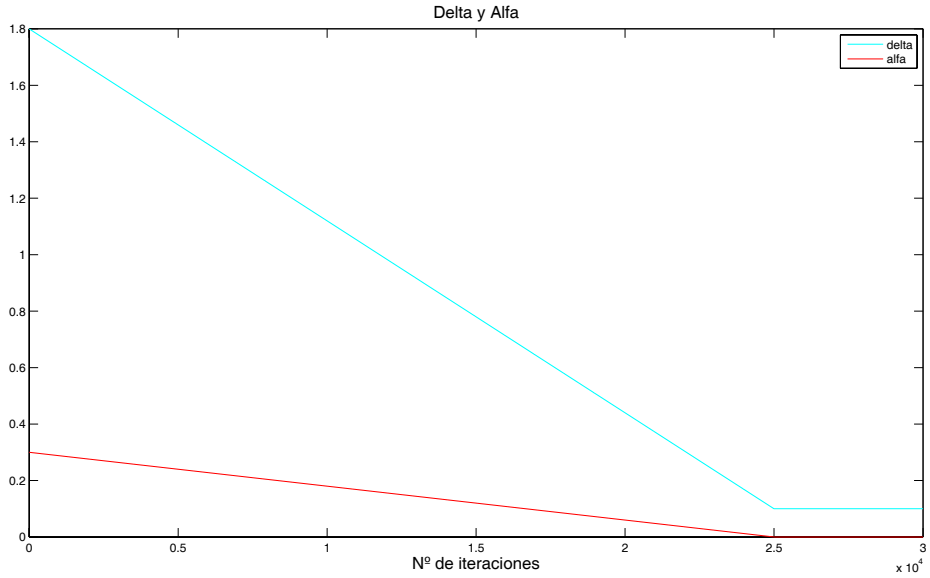


Fig. 8.14: *Valores de los parámetros δ y α durante las fases de aprendizaje y permanencia*

Donde:

$$V^{obj_i}(s') = \left(\max_{a \in A_{obj_i}} \left(Q^{obj_i}(s', a) \right) + \sum_m \Delta Q^{obj_m}_{\text{máx}} \right) \tag{8.4}$$

es el valor del objeto i en el estado nuevo. El factor de descuento γ, multiplica a $V^{obj_i}(s')$ que, como ya se ha dicho, mide lo mejor que el agente puede hacer a partir de ese estado. Por lo tanto, parece lógico pensar que, para que el agente aprenda una secuencia de acciones adecuada, es decir, que aprenda que para poder comer tiene que ir primero a por comida y cogerla, el γ ha de tener un valor alto. Tal y como se mostró en la sección 3.4.1, este parámetro también influye en el valor máximo que pueden tomar los valores Q, (3.7). De acuerdo con esta ecuación, γ nunca podrá ser igual a 1.

Para ver cómo se aprende una secuencia, se muestra el siguiente ejemplo: cuando el agente tiene hambre y come, recibe una recompensa inmediata, y por lo tanto, tendrá un valor alto de Q. También debe aprender que para comer tiene que coger la comida y para cogerla, primero tiene que haber llegado a ella. En (8.3) se muestra que el factor γ va a multiplicar al valor del objeto en el estado siguiente. Por lo que la actualización del valor Q, para un par estado-acción, consta de dos aportaciones: el refuerzo recibido en ese momento (r) y la importancia que se le de a lo mejor que puede pasar a partir del nuevo estado ($\gamma \cdot V^{obj_i}(s')$).

A continuación se van a mostrar los resultados obtenidos en el aprendizaje de esta secuencia, para distintos valores de γ. En la figura 8.15, el valor utilizado es $\gamma = 0{,}8$. Cuando el agente tiene hambre, está junto a comida, tiene comida y come, el valor Q es alto por la recompensa recibida (ver gráfico inferior derecho de la figura 8.15). La siguiente vez que el agente este junto a comida y coja comida, el valor Q de esta acción se actualiza siguiendo la ecuación (8.3). El nuevo estado es "tener comida", y lo mejor que puede hacer es comerla, por lo que el valor de este estado es alto. Como consecuencia, a pesar de no obtener una recompensa inmediata al cogerla, el hecho de que el valor del estado nuevo, sea multiplicado por un valor alto de γ, hará que el valor Q de "coger comida", sea alto (ver gráfico superior derecho de la figura 8.15). Esto mismo ocurrirá cuando ejecute la acción de "ir a por comida" y el estado nuevo sea "estar junto a comida", ya que su valor, como se ha visto, es alto. Finalmente, el agente utilizando un valor alto de γ aprende correctamente la secuencia de los comportamientos que le llevan a satisfacer el *drive* Hambre.

Para un valor un poco más bajo de γ, $\gamma = 0{,}5$, esta misma secuencia no se aprende tan bien como con un valor alto. Tal y como se aprecia en la figura 8.16, el valor Q de la acción de coger comida, no es tan alta como antes. Esto es debido a que su valor, como se ha dicho, depende del valor del siguiente estado ya que no se obtiene ninguna recompensa inmediata. Al ser γ no muy alta, no se valora el nuevo estado lo suficiente. Como consecuencia, la acción de "ir a por comida", tampoco es lo suficientemente alta. Por lo tanto, es de esperar que el agente no aprenda bien una política de comportamiento correcta.

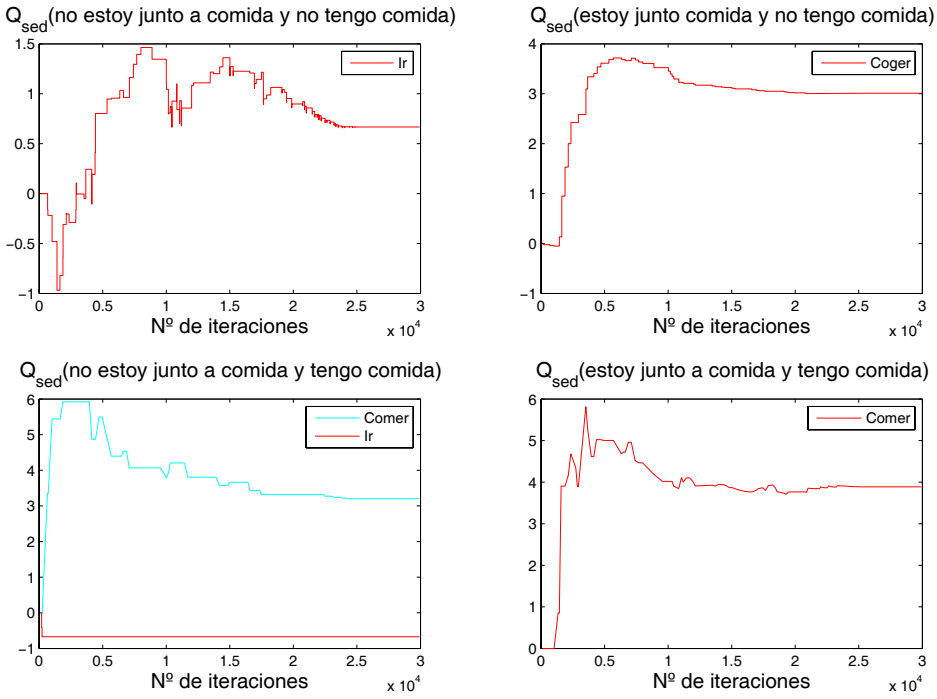


Fig. 8.15: *Valores Q de las acciones relacionadas con comida, cuando el agente tiene hambre, con* $\gamma = 0{,}8$

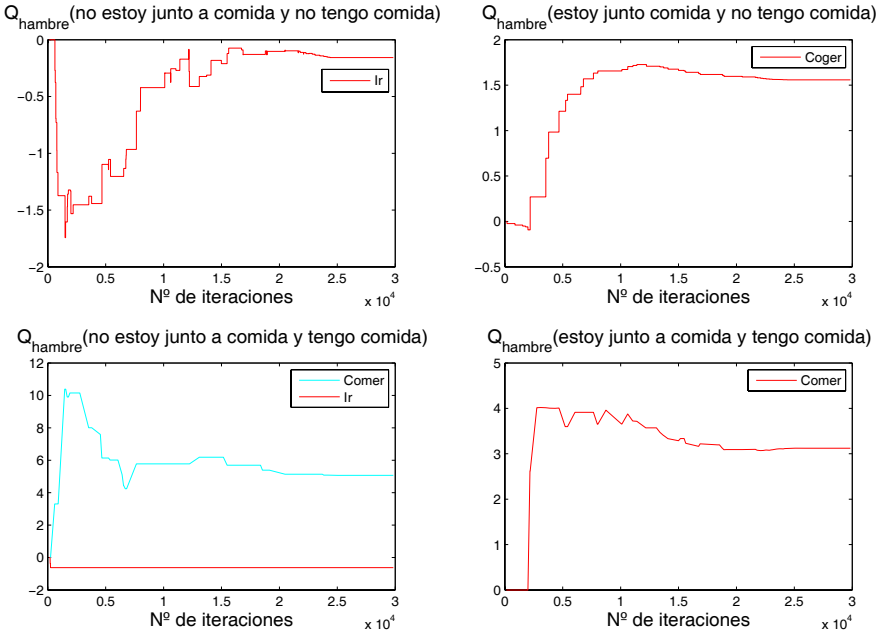


Fig. 8.16: *Valores Q de las acciones relacionadas con comida, cuando el agente tiene hambre, con* $\gamma = 0{,}5$

Por último, con un valor bajo de γ, $\gamma = 0{,}2$, esta secuencia, definitivamente, no se aprende bien. Cuando el agente realiza la acción de "ir a por comida", no va a valorar que finalmente podrá comer. En la figura 8.17, se puede apreciar los bajos valores Q de las acciones de " ir a por comida" y "coger comida".

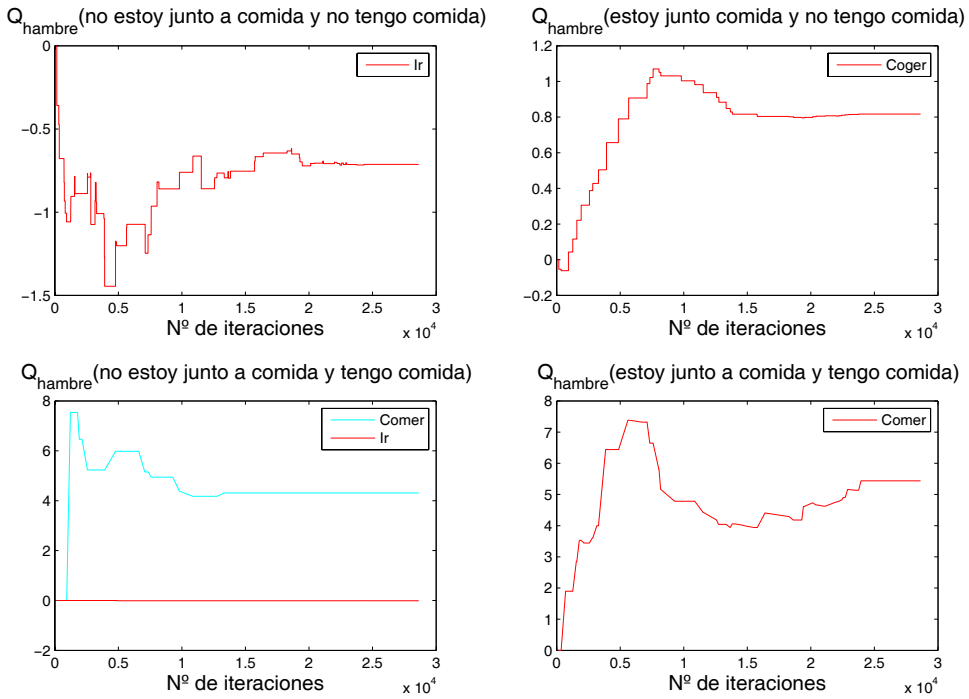


Fig. 8.17: *Valores Q de las acciones relacionadas con comida, cuando el agente tiene hambre, con* $\gamma = 0{,}2$

8.6. Valor del estímulo motivacional

Tal y como fue expuesto en 6.2.1, la intensidad de las motivaciones es una suma de la intensidad del *drive* relacionado y de un estímulo determinado (6.2). Este estímulo externo es lo que se denomina estímulo motivacional o incentivo.

Se han realizado experimentos con distintos valores para estos estímulos. El mundo utilizado es un mundo en el que la comida y la medicina están situados en habitaciones extremas, y el agua está en una zona de paso. De esta manera, el agente pasa al lado del agua casi siempre. Estos experimentos han mostrado que cuando el valor de este estímulo es alto, el agente coge y bebe agua cada vez que pasa al lado. De esta manera, casi nunca llega a tener sed, sino que pasa a ser la motivación dominante cuando está junto a agua. Por lo que la acción de "ir a por agua" no es ejecutada casi nunca. El agente, por lo tanto, no aprende que cuando tiene sed tiene que ir a por el agua, tal y como se muestra en la figura 8.18.

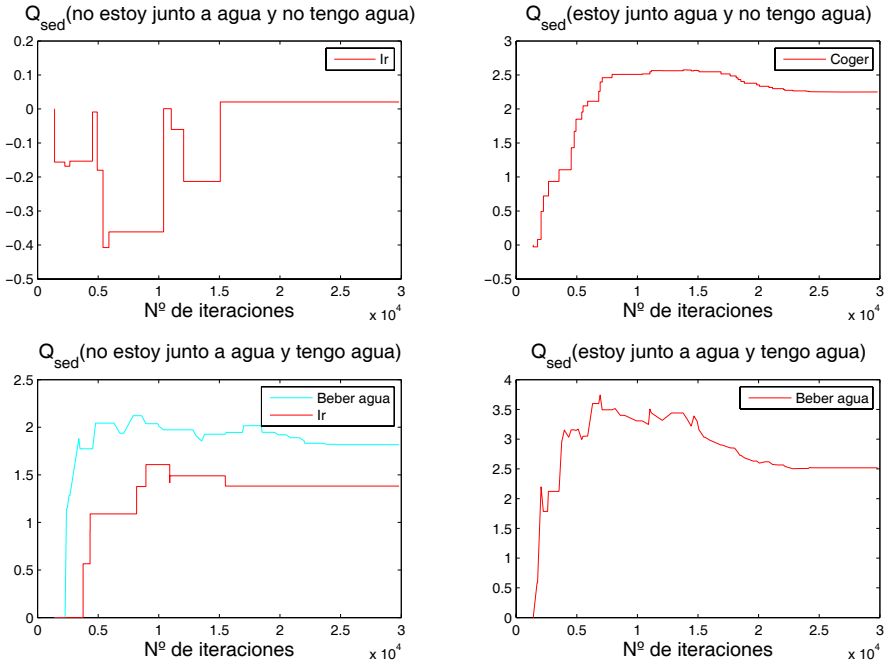


Fig. 8.18: *Valores Q de las acciones relacionadas con agua, cuando el agente tiene sed, con un valor del estímulo alto*

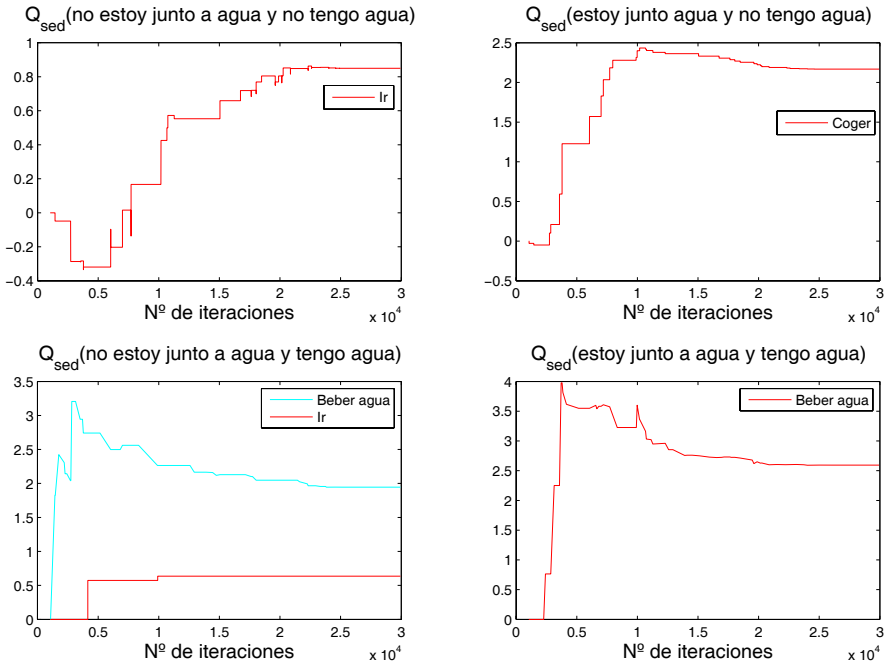


Fig. 8.19: *Valores Q de las acciones relacionadas con agua, cuando el agente tiene sed, con un valor del estímulo bajo*

Sin embargo, con un valor bajo del estímulo, cuando el agente esté junto a agua, no siempre va a ser Sed la motivación dominante. Esto sólo ocurrirá cuando el *drive* Sed y otros tengan valores similares. Como consecuencia, los resultados conseguidos son muy similares a los obtenidos cuando no se considera el estímulo en el cálculo de la intensidad de las motivaciones. En este caso, el agente llega a tener sed, estando lejos del agua y sí aprende a ir a por el agua con el fin de coger el agua y beberla, ver la figura 8.19.

8.7. Resumen y Conclusiones

En este primer capítulo de resultados experimentales se ha querido mostrar el ajuste de varios parámetros relacionados con el aprendizaje por refuerzo. Para ello se han realizado varios experimentos en los que se han variado dichos parámetros y se han comparado los resultados obtenidos.

El agente en estos experimentos se encuentra sólo, es decir no existen otros agentes en el mismo entorno. Por ello, se anuló el efecto del *drive* Soledad al igual que el *drive* Miedo. Este último será el objeto de estudio del último capítulo.

Como una de las principales conclusiones de este capítulo se determina que para estos experimentos, lo mejor es el uso de la variación del bienestar como función de refuerzo. Esto permite que el agente sea capaz de aprender políticas de comportamiento basándose en la teoría presentado por Rolls [Rolls, 2003]. Esta teoría propone que nuestras acciones están dirigidas a obtener recompensas y evitar castigos. Más aún, asumiendo que las emociones de felicidad y tristeza son definidas como evaluaciones positivas o negativas de una situación [Ortony, 2003], éstas se van a utilizar como refuerzo positivo y negativo respectivamente, en el proceso de aprendizaje.

Otra de las conclusiones importantes es que, para poder aprender una política correcta de comportamiento, es necesario primero explorar todas las acciones posibles, lo que implicará equivocarse e incluso vivir mal. Esto se traduce en un aumento del parámetro δ, que define la relación exploración/explotación de las acciones. Además es importante el valor de la tasa de aprendizaje, el parámetro α. Con un valor intermedio-bajo, se determinó que el agente aprendía correctamente.

Por otro lado se concluye que para "vivir bien" es necesario dejar de aprender. Esto se debe a que al ser el refuerzo la variación del bienestar, cuando se vive bien estos refuerzos disminuyen, llegando a "confundir" al agente.

Finalmente, después de varios experimentos, se llega a la conclusión de que es necesario que, a medida que la vida del agente transcurre, ambos parámetros disminuyan sus valores de manera progresiva. Los experimentos, por lo tanto, van a realizarse en dos fases, una fase de aprendizaje y una fase permanente. En la primera fase, el agente al comienzo explora y aprende rápido y, de forma progresiva, se comienza a explotar y a aprender más despacio. Al final, en la fase permanente, el agente no aprende, sólo vive utilizando los valores Q aprendidos.

Por otro lado, se determina que, con el fin de aprender una política de comportamiento correcta, el parámetro γ debe tener un valor alto. Es necesario, al valorar una acción en un estado, considerar las recompensas futuras. Finalmente, se ha realizado varios experimentos para analizar el efecto de los estímulos motivacionales. Los resultados finales son muy similares, pero analizando bien las secuencias de comportamientos, se observan diferencias. Si el estímulo incentivo es demasiado grande, el agente no aprende a ir a por él cuando lo necesita, simplemente pasa por su lado, lo coge y lo toma. Cuando el valor del estímulo es bajo, solamente en las ocasiones en las que el agente tiene varios *drives* al mismo nivel, existirá un cambio de motivación dominante.

Para concluir, en la tabla 8.1, se resumen los valores más adecuados de todos los parámetros involucrados en el diseño del agente.

Tab. 8.1: *Parámetros del agente*

	Fase aprendizaje	*Fase permanente*
Refuerzo	ΔWb	ΔWb
δ	1,8 $\rightarrow$ 0,1	0.1
α	0,3 $\rightarrow$ 0	0
γ	0.8	0.8
ω_i	1	1

9. RESULTADOS EXPERIMENTALES: AGENTE ACOMPAÑADO

9.1. Descripción del experimento

En este capítulo se va a presentar y analizar la actuación del agente cuando convive con otros agentes en el mismo entorno. Se van a presentar los resultados de varios experimentos en los que el agente vive, con distintas estrategias de interacción social y en distintos entornos. Para ello, en cada experimento se implementa en el agente un tipo de algoritmo de aprendizaje multiagente distinto, presentados en la sección 3.4.1, que va a definir la clase de interacción social. Cada uno de estos algoritmos de aprendizaje multiagente son probados en distintos entornos. El agente por lo tanto, aprende una política de comportamiento distinta, dependiendo del algoritmo de interacción implementado y del entorno en el que tiene que sobrevivir.

Los algoritmos que serán implementados en el agente son los siguientes:

- Amigo-Q: Utilizando este algoritmo se asume que el oponente es un amigo y, por lo tanto, el agente intenta escoger su acción de manera que se maximice la recompensa de ambos jugadores.

- Enemigo-Q: Utilizando este algoritmo el agente asume que su oponente es su enemigo y supone que el otro va a elegir una acción perjudicial para él. Considerando esta acción, el jugador elige aquella acción que haga máxima su propia recompensa.

- Media-Q: Cuando el agente usa este algoritmo elige la acción que simplemente hace máxima la recompensa media recibida.

- Q-learning: Utilizando este algoritmo, el agente calcula el valor de sus acciones sin considerar que es debido a su interacción con un oponente. Por lo tanto, cuando elige una acción sólo va a considerar aquella que haga máxima su valor Q.

Cada uno de estos algoritmos de aprendizaje multiagente será probado en distintos entornos. Los indicadores de análisis de resultados se calcularán a partir de ahora durante la fase permanente, ya que es en esta etapa donde el agente vive según la política aprendida.

9.1.1. Los mundos

Los entornos o "mundos", en los que el agente va a tener que vivir, están definidos por el tipo de oponentes que va a encontrar, es decir, por la "personalidad" de los otros agentes.

Para ello, se han creado tres tipos de oponentes con una política de comportamiento fija, es decir, que para cada estado en el que se encuentren, la acción que tienen que ejecutar está determinada de antemano. En relación a su interacción con los objetos estáticos, las políticas de comportamiento serán iguales para todos los objetos, por ejemplo: "Si tiene hambre y está junto a comida, entonces coge la comida y la come". Sin embargo, en relación a los objetos activos, como los otros agentes, es decir, en relación a la interacción social, cada uno de los agentes se va a comportar de distinta manera, lo que va a definir tres tipos de oponentes:

- Agente bueno: Este agente, cada vez que interactúa con otro agente, va a ejecutar acciones que son favorables para su oponente, es decir, que elegirá aleatoriamente entre las siguientes acciones:

$$A_{bueno} = \begin{cases} Saludar \\ Dar\ comida/agua/medicina \\ No\ hacer\ nada\ = Esperar \end{cases} \tag{9.1}$$

- Agente malo: Este agente por el contrario, cuando interactúa con otro agente, elegirá con mayor probabilidad alguna de las siguientes acciones:

$$A_{malo} = \begin{cases} Robar\ comida/agua/medicina \\ No\ hacer\ nada\ = Esperar \\ Pegar \end{cases} \tag{9.2}$$

  Las otras acciones, las "buenas", pueden llegar a ser elegidas pero de manera poco frecuente.

- Agente neutro: Este agente va a elegir de manera aleatoria entre cualquiera de las acciones disponibles a la hora de interaccionar, sin ninguna predilección.

$$A_{neutro} = \begin{cases} Saludar \\ Robar\ comida/agua/medicina \\ Dar\ comida/agua/medicina \\ No\ hacer\ nada\ = esperar \\ Pegar \end{cases} \tag{9.3}$$

Cada uno de estos agentes va a querer interactuar cuando su motivación dominante sea Soledad.

Una vez definidos los tres tipos de agentes que pueden estar presentes en cada experimento, se van a definir cuatro tipos de mundos o entornos en los que se van a probar cada uno de los algoritmos de aprendizaje multiagente:

1. **Mundo bueno**: en este mundo conviven tres oponentes "buenos".
2. **Mundo malo**: en este mundo conviven tres oponentes "malos".
3. **Mundo neutro**: en este mundo conviven tres oponentes "neutros".
4. **Mundo mixto**: en este mundo conviven los tres tipos de oponentes, uno "bueno", uno "malo" y otro "neutro".

Cada uno de los experimentos está caracterizado por el tipo de algoritmo de aprendizaje multiagente y por el mundo en el que se encuentra el agente. El agente aprende políticas de comportamiento distintas dependiendo de su forma de interacción social (definido por el algoritmo de aprendizaje multiagente) y de su entorno (definido por los oponentes). Para la realización de cada experimento se va a utilizar un agente "sin miedo", es decir, que este gente no va a tener el *drive* Miedo. Por lo tanto el agente sólo tiene cuatro *drives*: Hambre, Sed, Debilidad y Soledad.

9.2. Mundo bueno

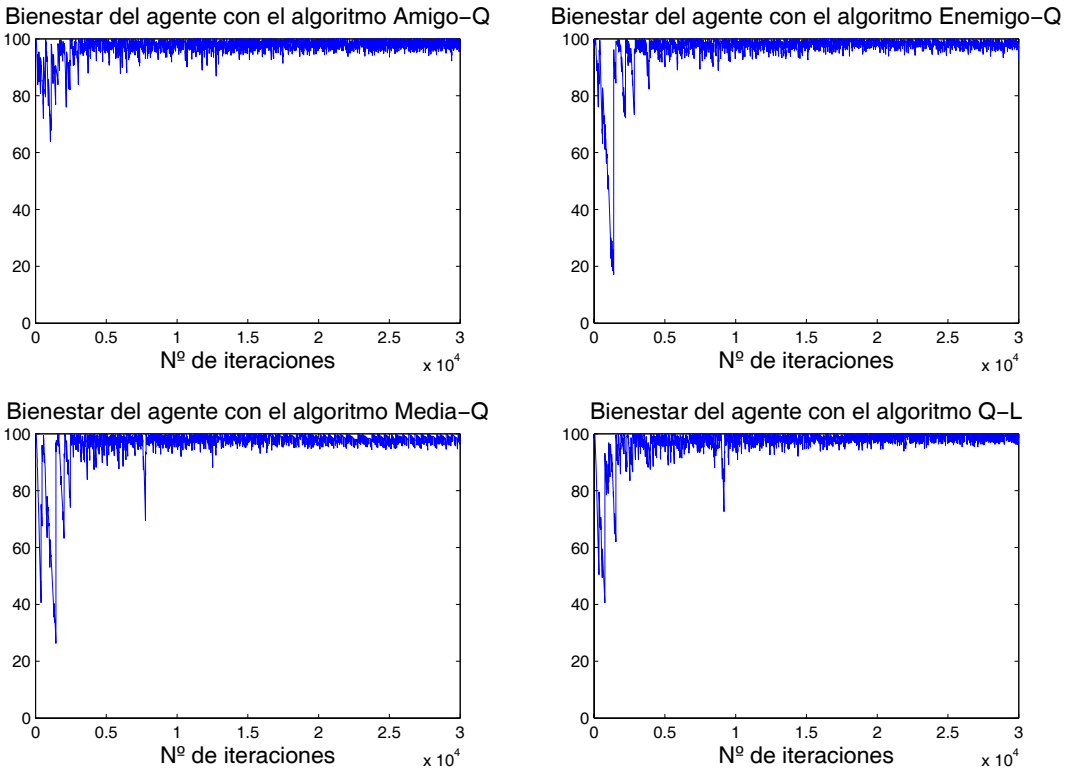


Fig. 9.1: *Bienestar del agente cuando vive en un mundo bueno, para cada algoritmo de interacción social utilizado*

En esta sección se van a presentar los resultados obtenidos para el caso de un agente que vive en un mundo bueno y utiliza los distintos algoritmos de aprendizaje para la interacción social. La actuación del agente se va a analizar utilizando la señal de bienestar y posteriormente, los valores de los indicadores de análisis de resultados, definidos en 7.6.

En la figura 9.1 se presentan las evoluciones del bienestar del agente cuando utiliza los distintos algoritmos de aprendizaje viviendo en el mundo bueno. En esta gráfica se puede apreciar que en este mundo todos los algoritmos dan buenos resultados, es decir, independientemente del algoritmo utilizado, el agente aprende una política de comportamiento correcta. Este resultado parece lógico considerando que este mundo no presenta ningún problema para el agente. De hecho, es curioso que en este mundo, el agente nunca llega a "necesitar" la interacción social. Independientemente del algoritmo utilizado, el *drive* Soledad nunca aumenta, ver la figura 9.2.

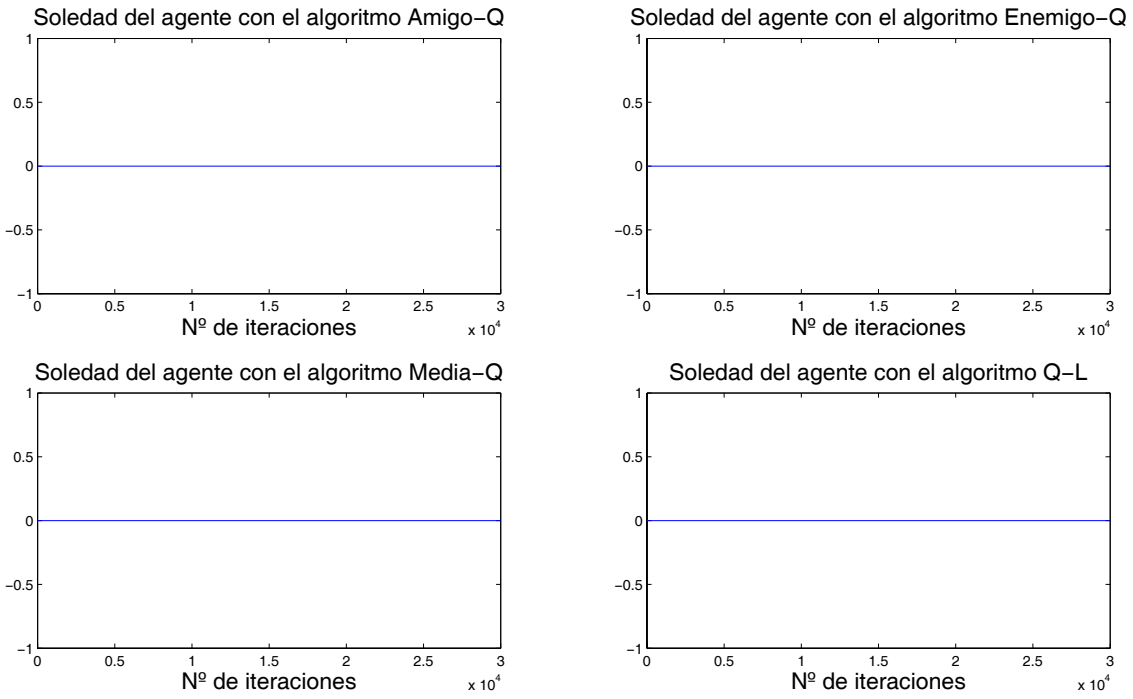


Fig. 9.2: *Soledad del agente cuando vive en un mundo bueno, para cada algoritmo de interacción social utilizado*

Esto es debido a que el resultado de todas las interacciones del agente con los otros agentes son siempre positivas, lo que hace que los valores Q aprendidos correspondientes a la interacción multiagente sean altos. Estos valores altos provocan que cada vez que el agente se encuentra a un oponente bueno, casi siempre le sale "rentable" interaccionar con él y como consecuencia de esta interacción, su *drive* Soledad queda satisfecho.

En la tabla 9.1 se muestran los valores de los indicadores de análisis de resultados, para cada uno de los algoritmos utilizados. Como se puede apreciar estos valores son muy buenos, de hecho, el porcentaje de permanencia en la zona de seguridad es del 100 % para todos los casos. Por otro lado, el valor medio del bienestar es superior a 98 para todos los algoritmos. A la vista de los resultados obtenidos, se puede concluir que en este mundo, la actuación del agente es siempre positiva, independientemente del algoritmo de interacción social utilizado.

Tab. 9.1: *Indicadores para el mundo bueno*

Algoritmo	*Valor medio del bienestar*	*% de permanencia en ZS*
Amigo-Q	98.5	100 %
Enemigo-Q	98.6	100 %
Media-Q	98.5	100 %
Q-learning	99.2	100 %

9.3. Mundo neutro

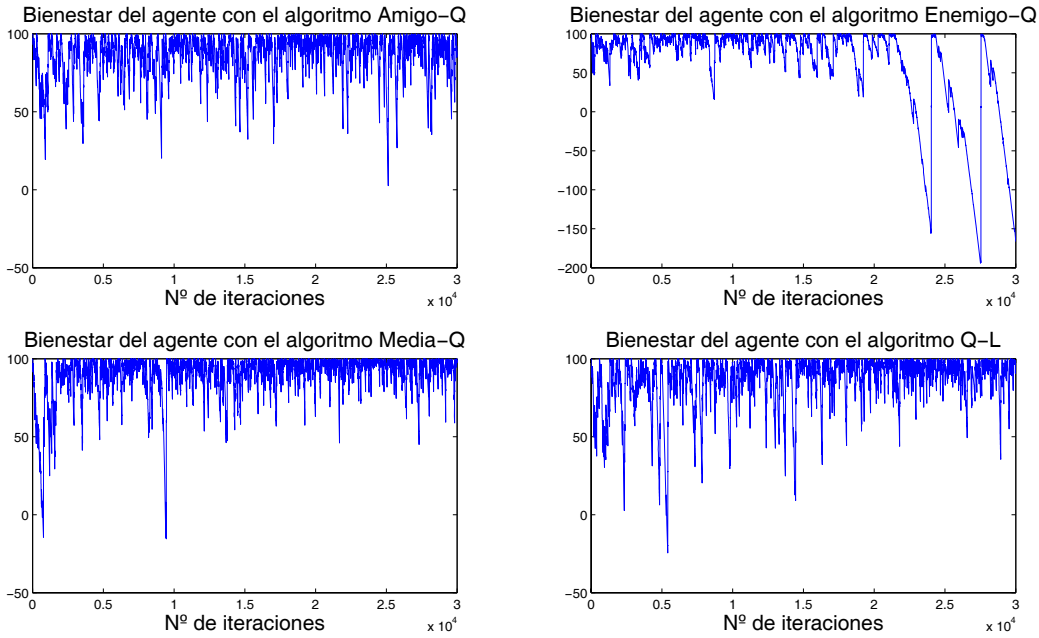


Fig. 9.3: *Bienestar del agente cuando vive en un mundo neutro, para cada algoritmo de interacción social utilizado*

En esta sección se presentan los resultados obtenidos para el caso de un agente viviendo en un mundo en el que todos sus oponentes son neutros. Al existir las mismas posibilidades de que un oponente neutro se porte bien o mal, ahora existirán diferencias en los resultados dependiendo del algoritmo de interacción usado.

Tal y como se muestra en la figura 9.3, en este caso, no con todos los algoritmos de aprendizaje el agente llega a aprender buenas políticas de comportamiento. En el mundo neutro, todos los algoritmos, menos el Enemigo-Q, permiten al agente sobrevivir con valores relativamente altos de bienestar. A continuación se va a analizar el porqué de estas diferencias de actuación del agente con cada algoritmo.

9.3.1. Amigo-Q en el mundo neutro

Cuando el agente utiliza el algoritmo de aprendizaje multiagente Amigo-Q, se puede ver que el bienestar, a pesar de tener un buen valor medio, 85,0 tiene un porcentaje de permanencia en la zona de seguridad no muy alto, 45,6 %. A lo largo de toda la vida, la señal de bienestar ha oscilado mucho, llegando frecuentemente a valores mínimos de 20 ó 30.

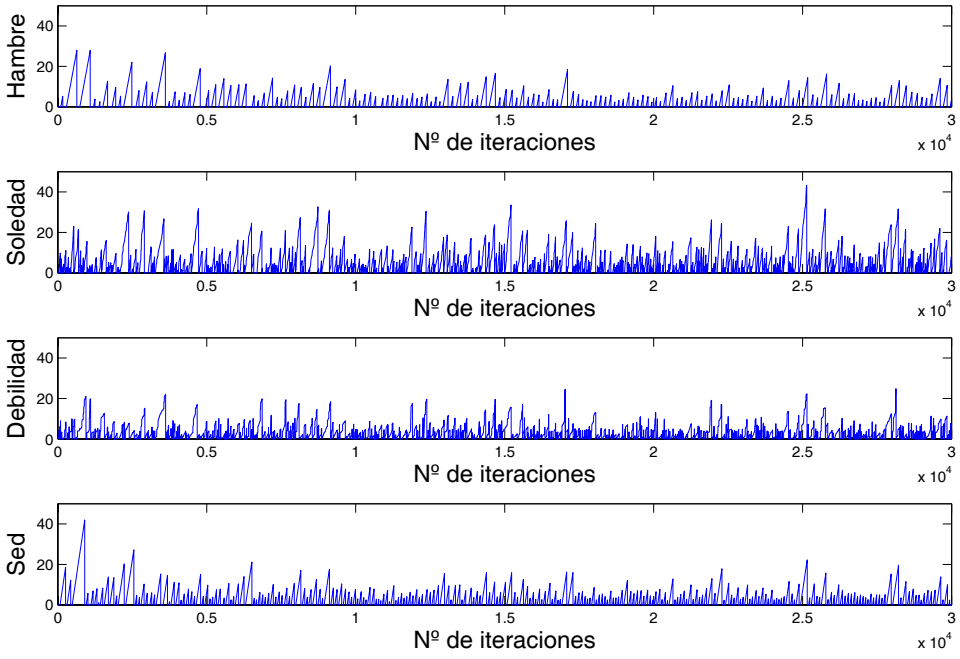


Fig. 9.4: Drives *del agente cuando vive en un mundo neutro utilizando el Amigo-Q*

Si se observa la evolución de los *drives* en la figura 9.4, se puede ver que la mayoría de los picos se producen para el *drive* Soledad. Esto se debe a que con este algoritmo, el agente piensa que su oponente es su amigo, por lo que va a querer interaccionar con él. El problema es que ahora, a diferencia de lo que pasa en el mundo bueno, el

otro agente no siempre se va a portar bien con él, lo que produce que, además de no satisfacer el *drive* Soledad, hace que aumenten, éste y posiblemente otros *drives*. De hecho, se puede ver cómo el *drive* Debilidad presenta picos muy pronunciados aproximadamente al mismo tiempo que los picos de Soledad. Estos picos son debidos a que un agente neutro le pega, lo que produce un aumento de ambos *drives*, Debilidad y Soledad, siendo mayor el aumento para el *drive* Debilidad.

También se observan picos de mayor tamaño que la media para los *drives* Hambre y Sed, los cuales ocurren, la mayoría de las veces, de nuevo al mismo tiempo que los picos que presentan los otros *drives*. Estos se producen porque para el caso de que el agente, por ejemplo, tenga hambre, el valor que tiene "ir a por comida", ver la figura 9.5, es menor que el valor que tiene interaccionar, ver la figura 9.6.

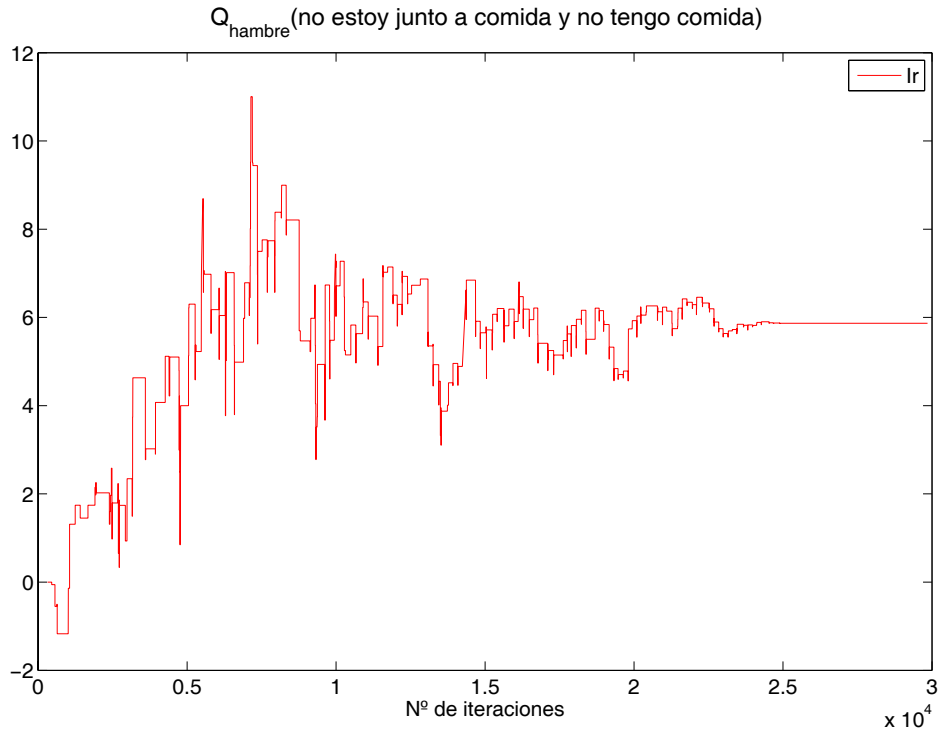


Fig. 9.5: *Valores Q de las acciones relacionadas con comida cuando la motivación dominante es Hambre, en un mundo neutro utilizando el Amigo-Q*

En la figura 9.5 se puede ver que el valor final de Q correspondiente a la acción "ir a por comida" cuando tiene hambre es 5,85. Por otro lado, el valor de interaccionar, usando este algoritmo Amigo-Q, será el máximo valor entre todas las filas y columnas de la matriz Q de interacción $Q(a_1, a_2)$ representada en la figura 9.6. Para este caso, es el valor Q correspondiente a $Q(pegar, saludar) = 6{,}25$. Por lo tanto, mientras el agente está interactuando con otro agente, creyendo que le va a favorecer, el *drive* Hambre sigue creciendo. El agente, en este caso "pierde el tiempo" interactuando.

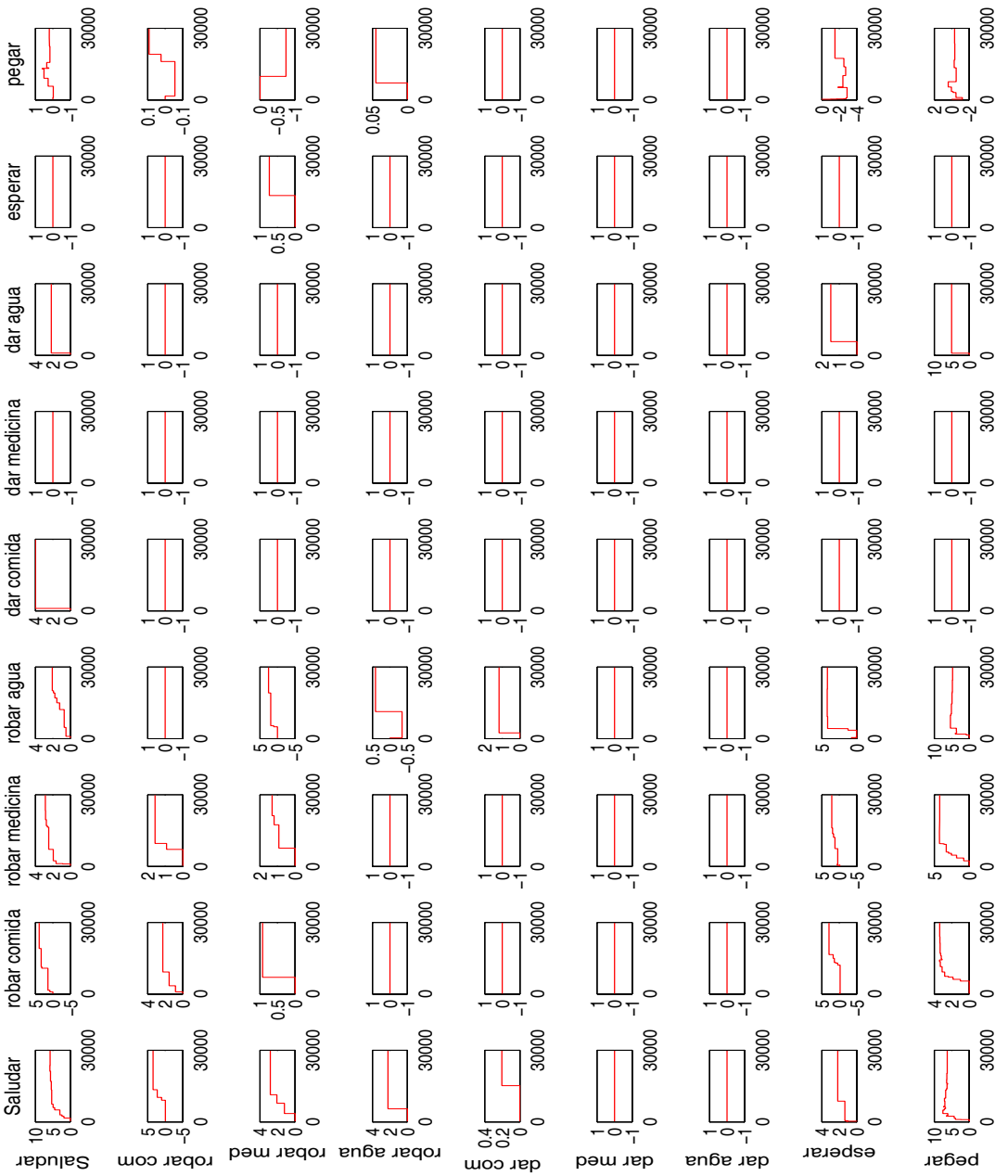


Fig. 9.6: *Valor de la matriz $Q(a_1, a_2)$ cuando la motivación dominante es Hambre, en un mundo neutro utilizando el Amigo-Q*

9.3.2. Enemigo-Q en mundo neutro

En este caso el agente supone que los otros agentes son sus enemigos, es decir, utiliza el algoritmo Enemigo-Q. En la figura 9.7 se pueden observar unos picos de gran magnitud del *drive* Soledad, lo que provoca que su bienestar llegue a valores muy negativos en la fase permanente.

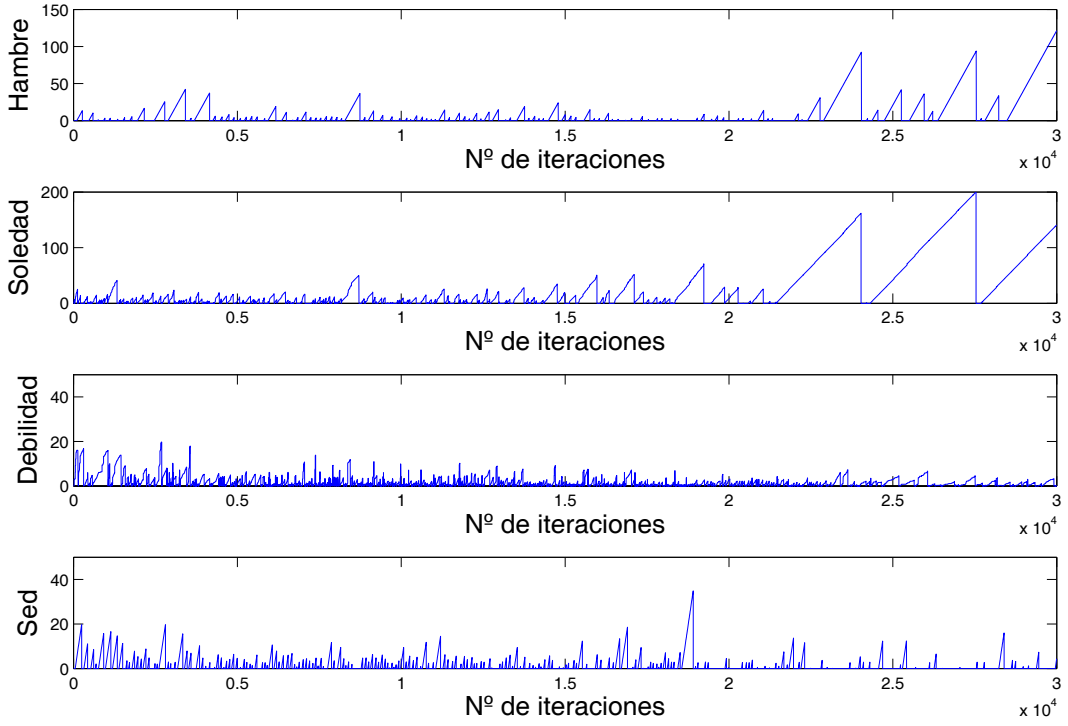


Fig. 9.7: Drives *del agente cuando vive en un mundo neutro utilizando el Enemigo-Q*

Utilizando este algoritmo, el agente supone que el otro va a escoger la acción más perjudicial para él, y por lo tanto, lo que hace es ver qué acción del otro, a_2, es la que tiene un valor $Q(a_1, a_2)$ menor para cada una de sus propias acciones a_1. Finalmente escoge aquella acción que hace máximo esos valores mínimos. En la figura 9.8, se muestran estos valores para la motivación dominante Soledad. Como se observa, la acción que finalmente se escoge es la correspondiente a $Q(dar\ comida, robar\ comida) = -0{,}9$. Por lo tanto ocurre que, cuando el agente necesita interacción social, el valor de interaccionar, $-0{,}9$, es inferior al valor de otras acciones, como por ejemplo, a los valores de las acciones relacionadas con el agua, como se muestra en la figura 9.9. Esto lleva al agente a preferir realizar otras acciones antes que interaccionar, por lo que el *drive* Soledad crece mucho.

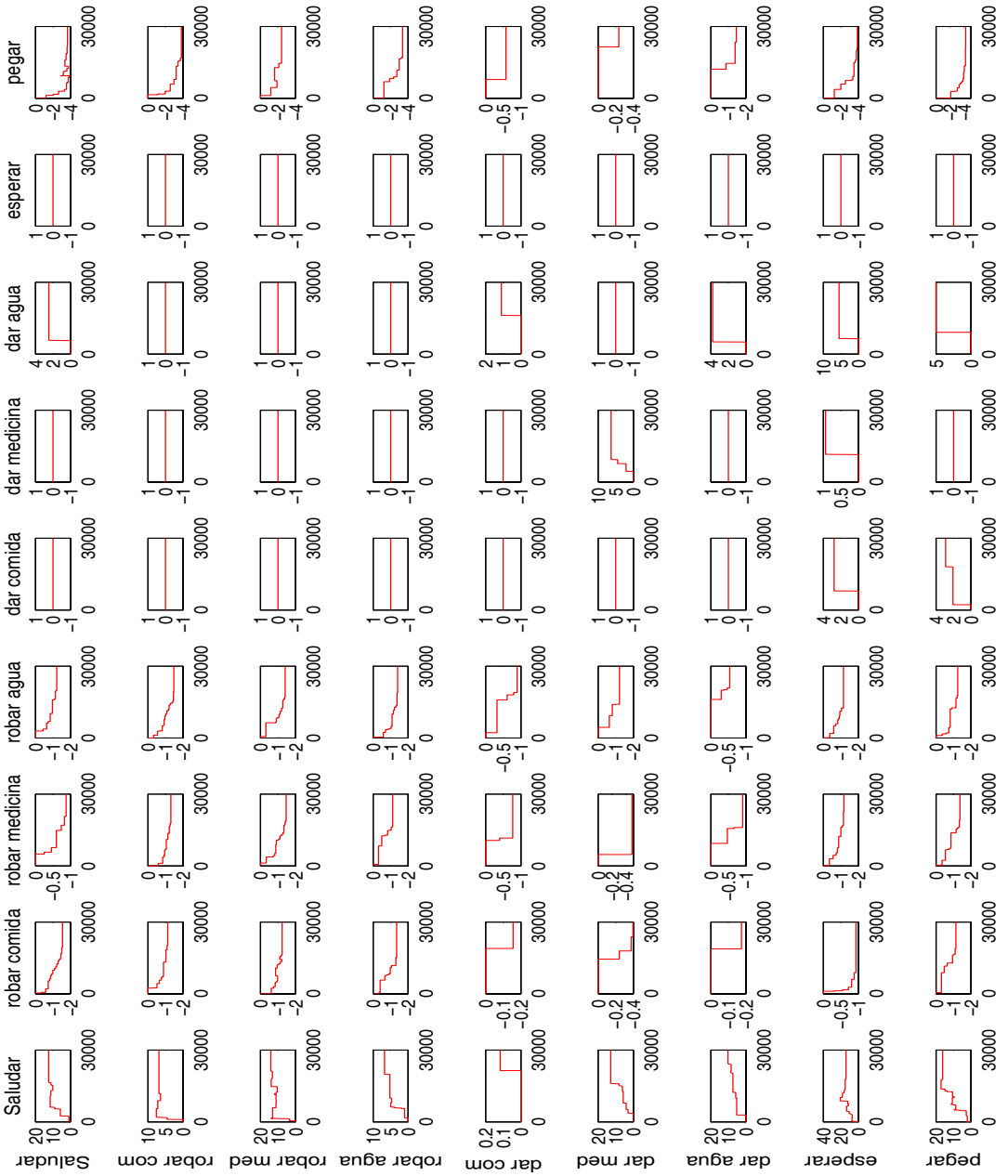


Fig. 9.8: *Valor de la matriz $Q(a_1, a_2)$ cuando la motivación dominante es Soledad, en un mundo neutro utilizando el Enemigo-Q*

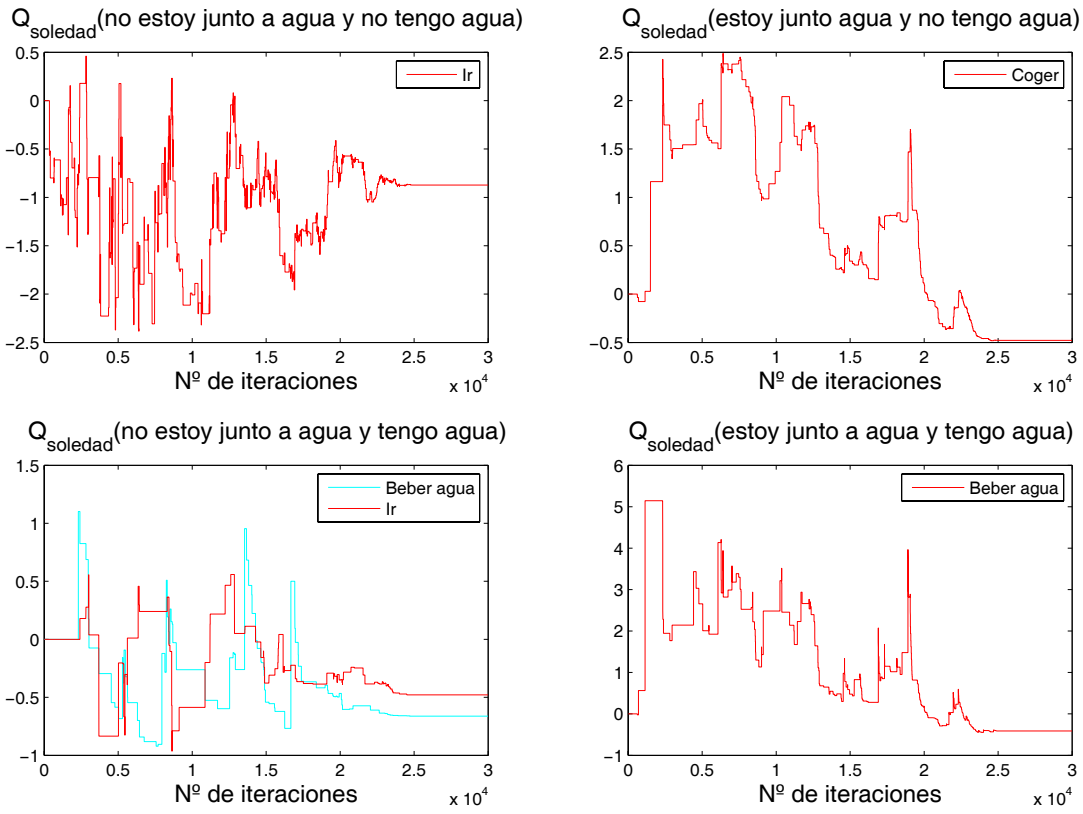


Fig. 9.9: *Valores Q de las acciones relacionadas con agua cuando la motivación dominante es Soledad, en un mundo neutro utilizando el Enemigo-Q*

9.3.3. Media-Q en mundo neutro

Si el agente, en lugar de considerar lo mejor o lo peor que le ha pasado durante la interacción social, considera el valor medio de cada una de sus acciones, los resultados mejoran considerablemente. Cuando el agente utiliza el algoritmo Media-Q, los valores de sus indicadores: valor medio del bienestar y porcentaje de permanencia en la zona de seguridad, son altos. Esto indica que cuando el agente tiene una visión no tan positiva como con el Amigo-Q, ni tan negativa, como con el Enemigo-Q, es capaz de vivir mejor en el mundo neutro.

Tal y como se muestra en la figura 9.10, los valores de los *drives* permanecen acotados, sin crecer indefinidamente. Los picos que aparecen en el *drive* Soledad, al igual que los de Debilidad, son debidos de nuevo, a que durante esas veces el oponente le pega.

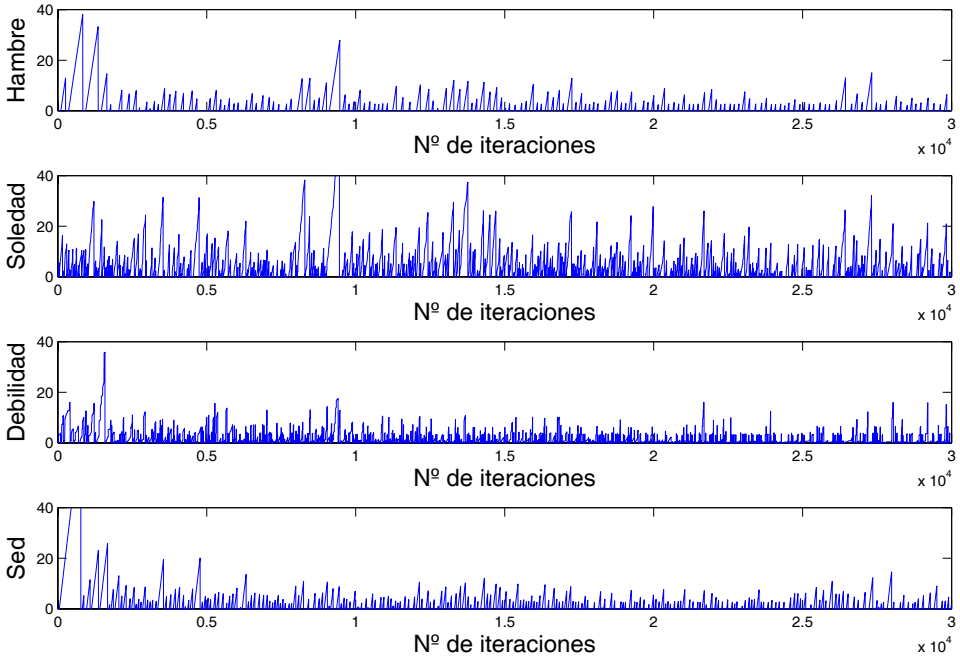


Fig. 9.10: Drives *del agente cuando vive en un mundo neutro utilizando el Media-Q*

9.3.4. Q-learning en el mundo neutro

Los resultados obtenidos utilizando el algoritmo Q-learning son similares al algoritmo Media-Q. Tanto el valor medio del bienestar del agente como el porcentaje de permanencia son también altos. El valor de cada acción en la interacción, va a depender de la cantidad de veces que se portaron bien o mal con el agente.

A pesar de que parece que el agente ha aprendido una política de comportamiento adecuada, no es así. En la figura 9.11 se pueden observar dos picos en Hambre de valor 20 aproximadamente, durante la fase permanente. Es decir, que el agente tarda en satisfacer su hambre. Si la política aprendida hubiese sido la correcta, el valor de "ir a por comida" tendría más valor que cualquier otra, y esto es lo que no es así.

De hecho, en esta gráfica se observa que al mismo tiempo que el *drive* Hambre crece, tanto el *drive* Sed como Debilidad no crecen, permaneciendo en valores muy bajos. Esto se debe a que, tal y como se muestra en la figura 9.12, el valor de "ir a por comida" es un poco inferior a "coger agua" y "beber agua". El hecho de que sea un poco inferior favorecerá, debido a la pequeña aleatoriedad que se conserva durante esta fase final, que el agente finalmente elija "ir a por comida", aunque sea un poco tarde.

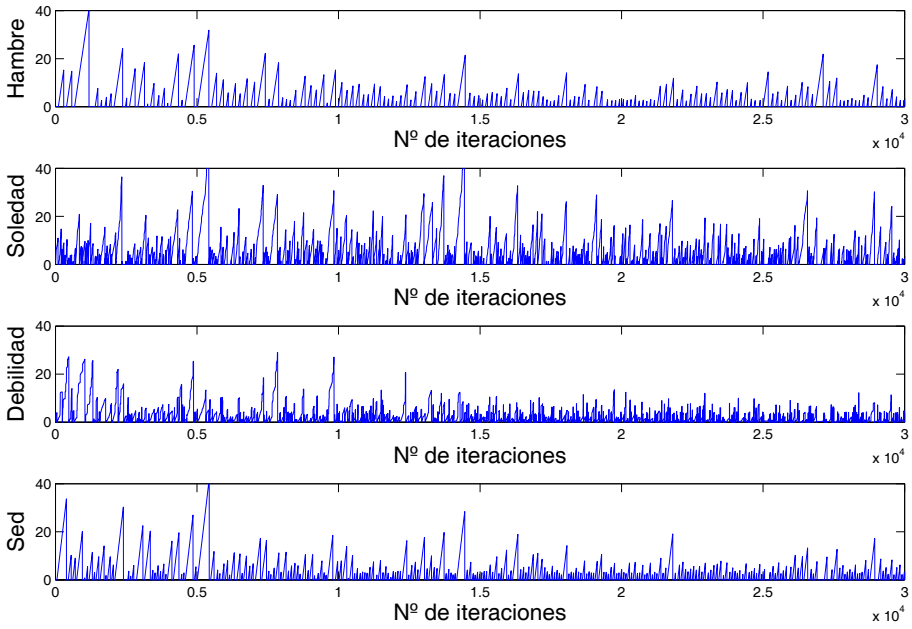


Fig. 9.11: Drives *del agente cuando vive en un mundo neutro utilizando el Q-learning*

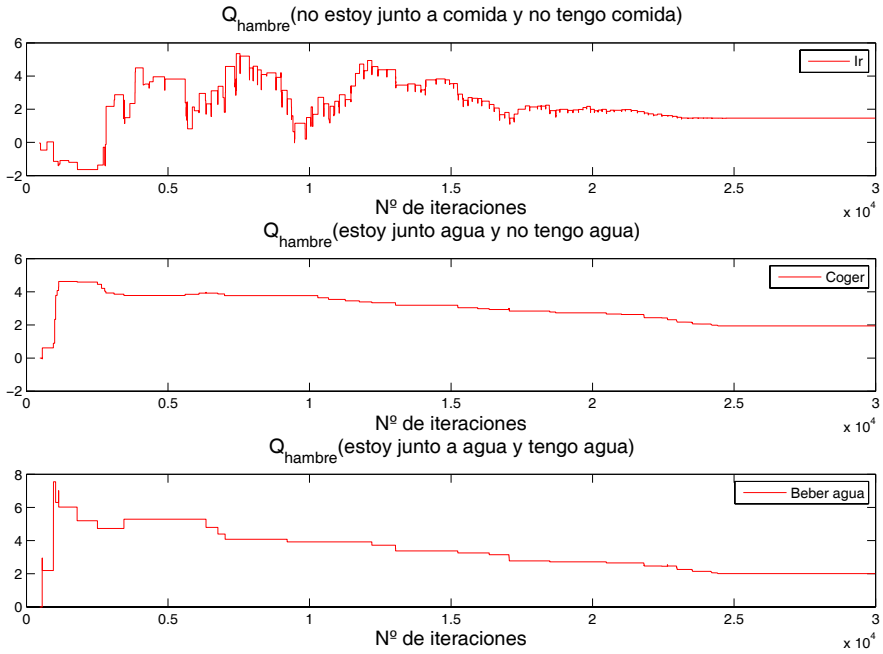


Fig. 9.12: *Valores Q cuando la motivación dominante es Hambre, en un mundo neutro utilizando el Q-learning*

9.3.5. Indicadores del mundo neutro

En la tabla 9.2 se muestran los valores de los indicadores de análisis del bienestar para el agente en el mundo neutro. Se puede observar que para el caso del algoritmo Enemigo-Q, ambos parámetros no han sido calculados ya que, como se vio en la figura 9.3, durante la fase de permanencia (5000 últimas iteraciones), el valor medio del bienestar es muy negativo.

Tab. 9.2: *Indicadores para el mundo neutro*

Algoritmo	*Valor medio del bienestar*	*% de permanencia en ZS*
Amigo-Q	85.0	45.6 %
Enemigo-Q	NO	NO
Media-Q	92.84	70.86 %
Q-learning	90.68	58.7 %

Como conclusión, debido a la naturaleza del algoritmo Amigo-Q y del Enemigo-Q, ninguno de los dos son apropiados para sobrevivir en este mundo neutro. La visión negativa del Enemigo-Q va a causar que el agente no quiera interaccionar con los oponentes, mientras que el Amigo-Q, hace que el agente ignore que no siempre las interacciones son positivas. Tanto el Media-Q, como el Q-learning tienen en cuenta tanto los efectos positivos, como los negativos, por lo que cualquiera de ellos son apropiados para la supervivencia en este mundo neutro.

9.4. Mundo mixto

Cuando el agente vive en este mundo, se va a tener que enfrentar con los tres tipos de oponentes: el bueno, el malo y el neutro. Debido a que el agente no identifica a su oponente, es decir, trata a todos por igual, su actuación dependerá en gran medida de la cantidad de veces que interactúe con cada uno. Por lo tanto, los resultados obtenidos utilizando los distintos tipos de algoritmos de aprendizaje multiagente y el Q-learning, van a ser distintos entre sí, tal y como se muestra en la figura 9.13. Por ello, se va a analizar la actuación del agente utilizando cada algoritmo, por separado.

9.4.1. Amigo-Q en mundo mixto

En la figura 9.13, se aprecia que el bienestar del agente cuando vive en un mundo mixto, suponiendo que los oponentes son sus amigos, sufre un descenso brusco casi al final de la fase permanente. En la figura 9.14 se puede ver que esta caída es debida a que todos los *drives* comienzan a crecer se manera descontrolada.

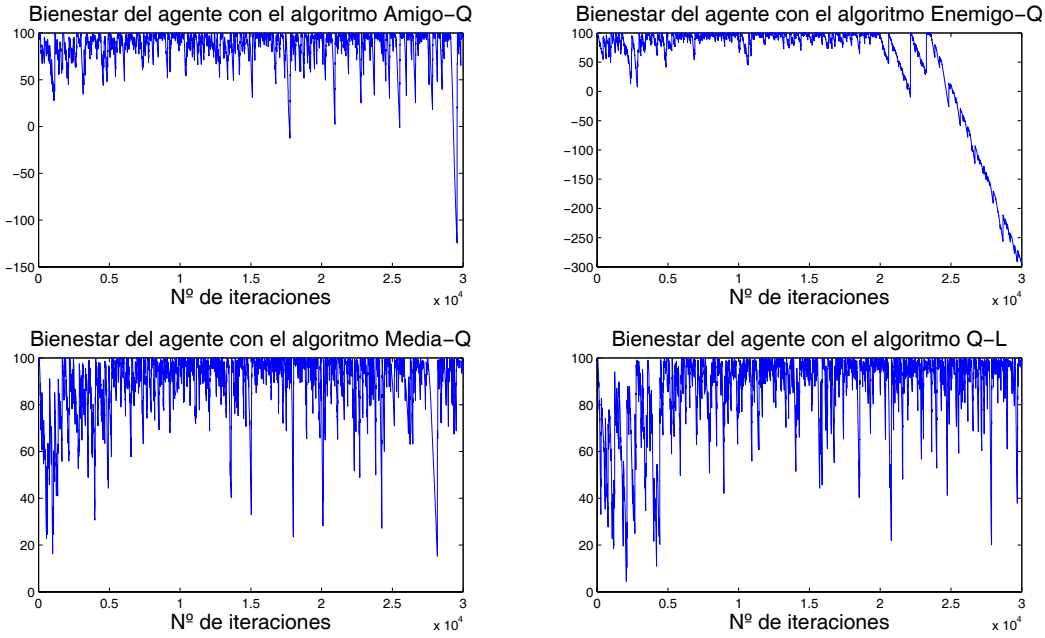


Fig. 9.13: *Bienestar del agente cuando vive en un mundo mixto, para cada algoritmo de interacción social utilizado*

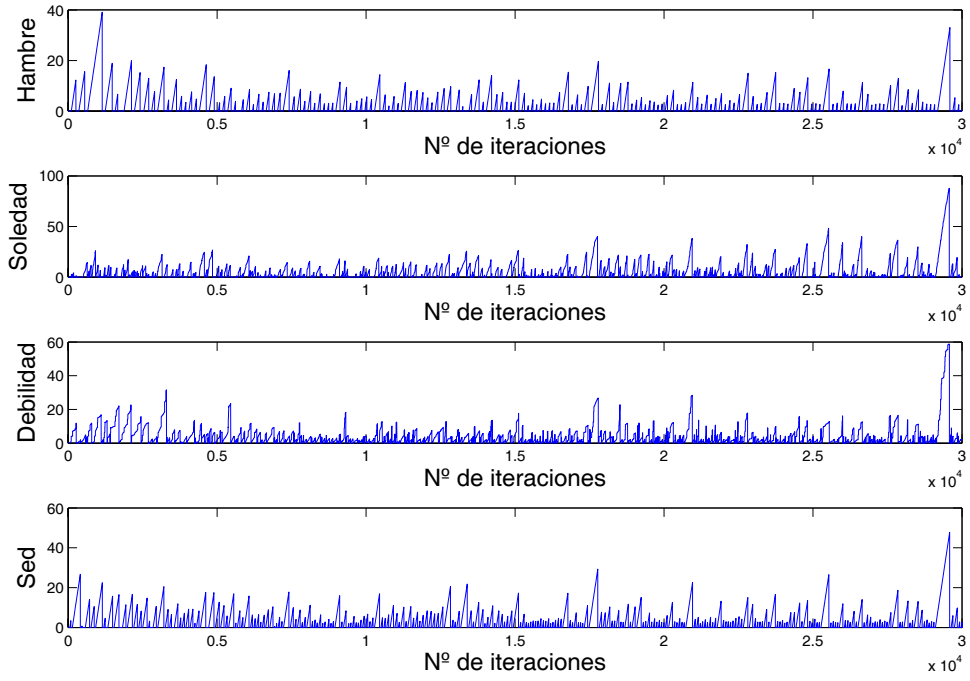


Fig. 9.14: Drives *del agente cuando vive en un mundo mixto utilizando el Amigo-Q*

Este algoritmo, como se puso en evidencia en el mundo neutro, tiene un problema: ser demasiado optimista. Cuando el agente supone que todos los oponentes son sus amigos, sólo considera los valores Q máximos como resultado de su interacción con ellos. Por este motivo, cuando se encuentra a otro agente, la mayoría de las veces va a querer interaccionar con él ya que los valores de la matriz $Q(a_1, a_2)$ de interacción son altos. Por ejemplo, cuando no existe motivación dominante el valor Q más alto en relación con los objetos pasivos es el de "explorar", 4,4, ver la figura 9.15. Cuando el agente "explora" tiene grandes posibilidades de encontrarse con otro agente y cuando está acompañado, el valor de interaccionar, ver la figura 9.16, que es el valor máximo entre filas y columnas, $Q(pegar, robar\ medicina) = 5{,}66$, es mayor que el valor de "explorar", por lo que comienza la interacción.

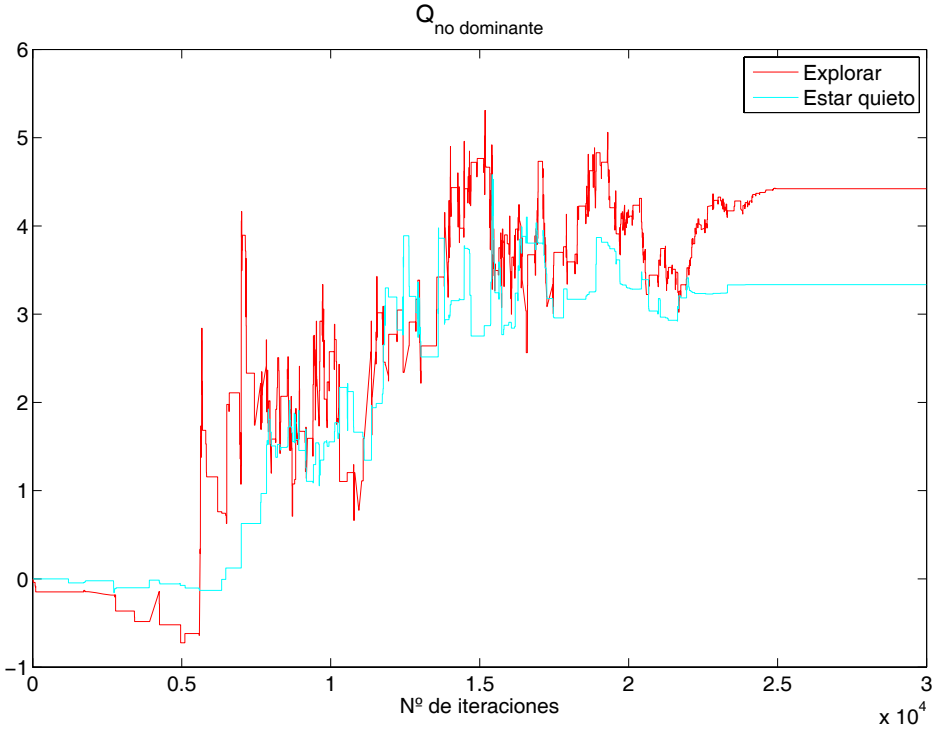


Fig. 9.15: *Valores Q de "explorar" y "estar quieto" cuando no hay motivación dominante, en un mundo mixto utilizando el Amigo-Q*

El problema aparece cuando el oponente es "malo", ya que puede que le pegue, tal y como sucede, lo cual se ve en los picos del *drive* Debilidad y Soledad en la fase permanente. Como el agente utiliza este algoritmo de Amigo-Q, va a querer seguir interactuando con este agente "malo", por lo que mientras este otro agente no se vaya, la interacción entre lo dos continúa y por eso el resto de los *drives* siguen creciendo causando esa gran caída del bienestar.

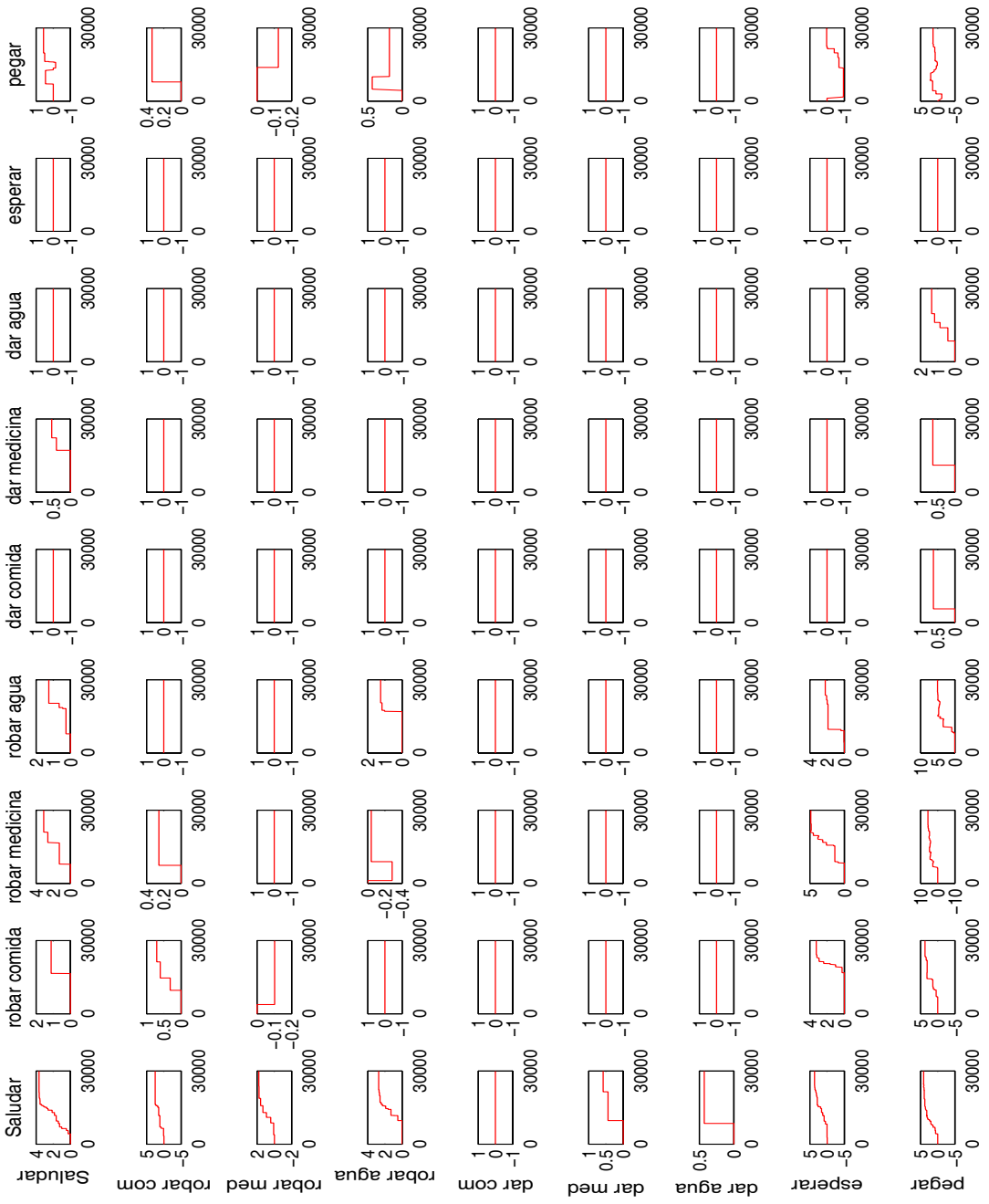


Fig. 9.16: *Valor de la matriz $Q(a_1, a_2)$ cuando no hay motivación dominante, en un mundo mixto utilizando el Amigo-Q*

9.4.2. Enemigo-Q en mundo mixto

La actuación del agente, cuando piensa que los otros agentes son sus enemigos, es la esperada. De la misma manera que en el caso del mundo neutro, viendo cómo evolucionan los *drives* se ve que el descenso del bienestar del agente se debe al aumento indefinido del *drive* Soledad, ver la figura 9.17.

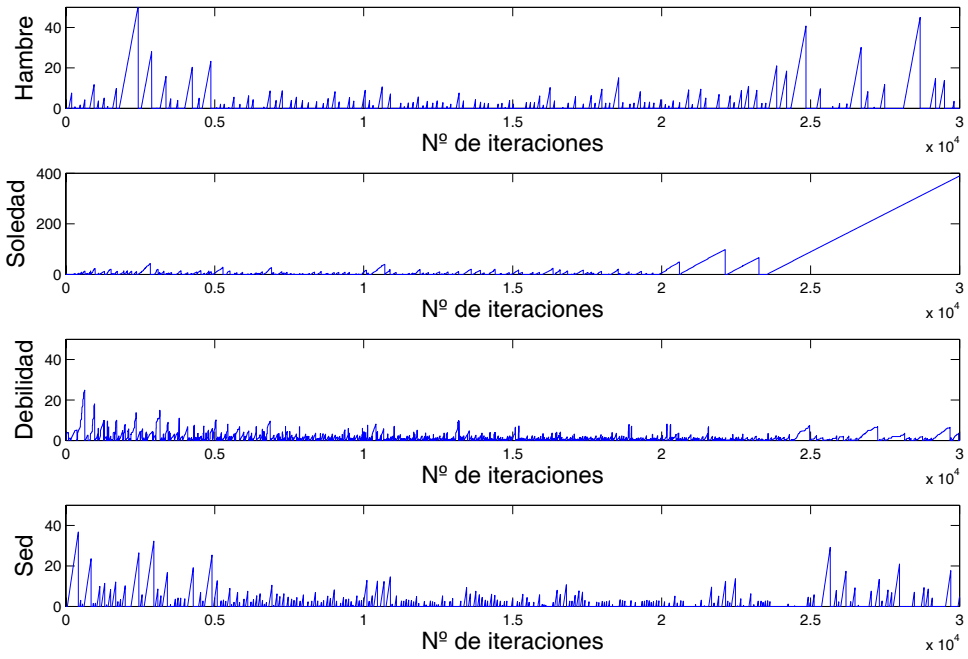


Fig. 9.17: Drives *del agente cuando vive en un mundo mixto utilizando el Enemigo-Q*

Se puede observar que a medida que el agente va viviendo, parece que le cuesta más interaccionar, es decir, tarda en satisfacer el *drive* Soledad, hasta que al final decide no interaccionar. Esto se debe a que, como ya ha sido explicado previamente, el valor de interaccionar es menor que los valores Q de otras acciones. De hecho, se ve que durante la fase permanente, mientras el *drive* dominante es Soledad, el agente sí que satisface los otros *drives*.

En la figura 9.18 se muestra la matriz de interacción $Q(a_1, a_2)$. El valor que tiene interaccionar con el otro agente es el valor máximo de los mínimos de cada fila, por lo que queda $Q(dar\ comida, robar\ agua) = -1{,}1$.

Viendo los valores que tienen otras acciones para esta motivación dominante, se puede explicar por qué los otros *drives* se satisfacen. Por ejemplo, para el caso de los valores Q de las acciones relacionadas con el agua, ver la figura 9.19, se puede observar que todos los valores son superiores a $-1{,}1$. Por lo que el agente, cuando la

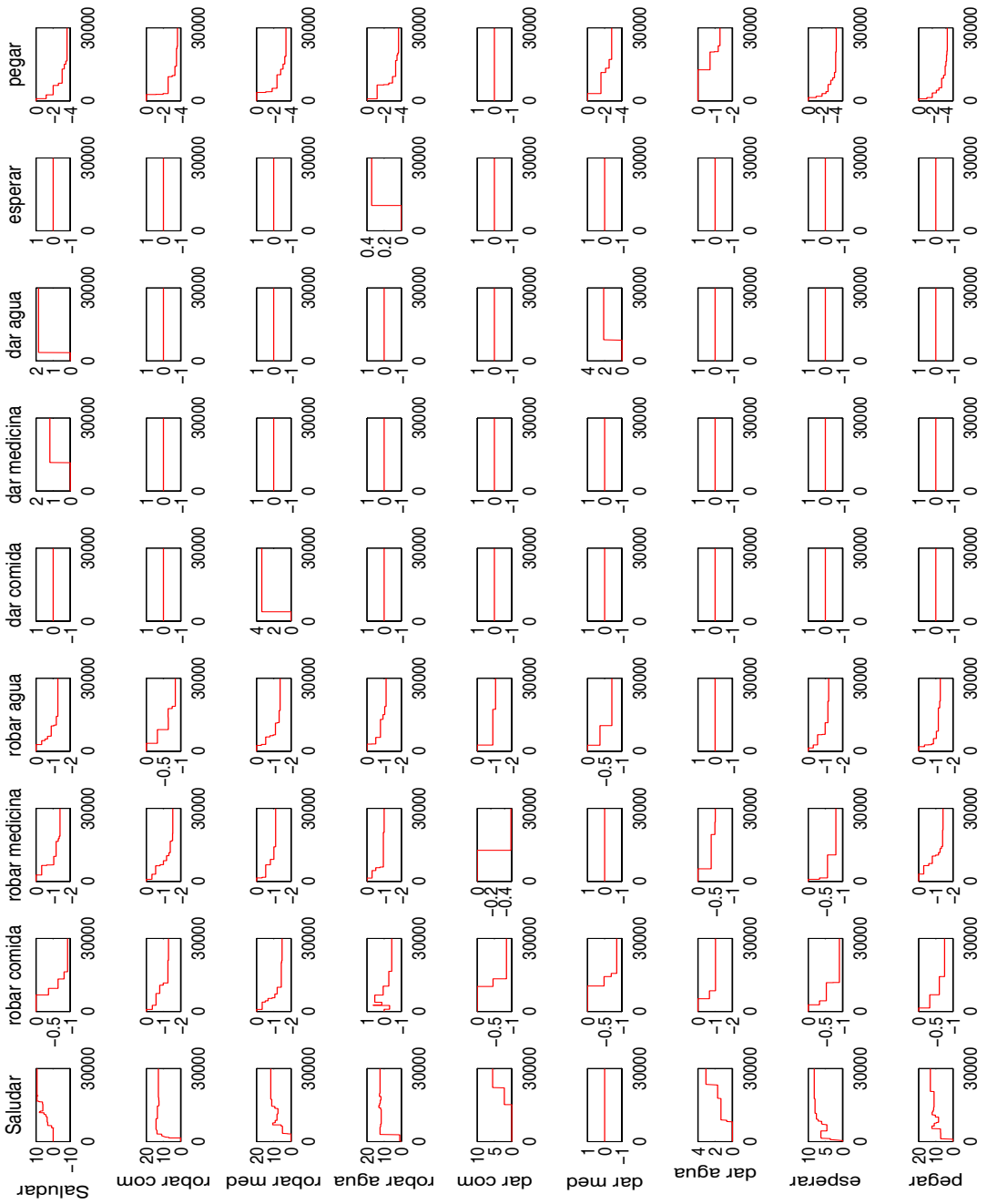


Fig. 9.18: *Valor de la matriz $Q(a_1, a_2)$ cuando la motivación dominante es Soledad, en un mundo mixto utilizando el Enemigo-Q*

motivación dominante es Soledad, va a por agua, la coge y se la bebe. Lo mismo pasa en relación a lo otros objetos, los gráficos son muy parecidos, por lo que no van a ser presentados. Cabe decir que los valores de "ir a" son todos muy similares entre sí, por lo que se explica que a veces, gane el " ir a por agua", luego "ir a por comida" o "ir a por medicina". La pequeña aleatoriedad, presente en esta última fase, hace que éstas sean elegidas por el agente de manera casual.

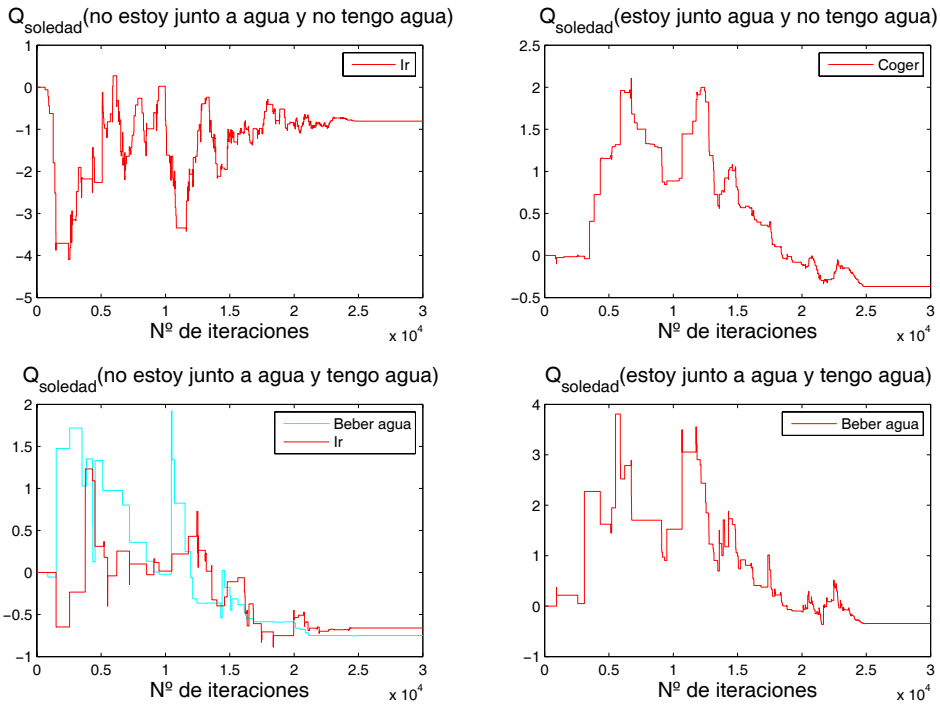


Fig. 9.19: *Valores Q de las acciones relacionadas con agua cuando la motivación dominante es Soledad, en un mundo mixto utilizando el Enemigo-Q*

9.4.3. Media-Q en mundo mixto

El bienestar del agente cuando utiliza el algoritmo Media-Q, ver la figura 9.13, presenta varios descensos bruscos a la mitad de su vida. Éstos se deben principalmente de nuevo a los picos que existen en el *drive* Soledad, ver la figura 9.20. Lo llamativo de este caso es que la principal caída que sufre el bienestar durante la fase permanente no tiene que ver con la interacción con otros agentes. Este descenso es debido, por el contrario, a que no ha aprendido bien lo que tiene que hacer cuando tiene hambre. Si se observa la gráfica de los *drives*, la figura 9.20, se ve que el *drive* Hambre está creciendo, mientras que Sed está a cero y Debilidad tiene un valor muy pequeño y constante. Es decir, que ha aprendido que cuando tiene hambre, lo mejor es quedarse junto al agua y beber. Tal y como se ve reflejado en la figura 9.21, el valor Q de "ir a por comida" es 1,06, el valor Q de "coger agua" es 1,46 y el de "beber agua" es 1,7.

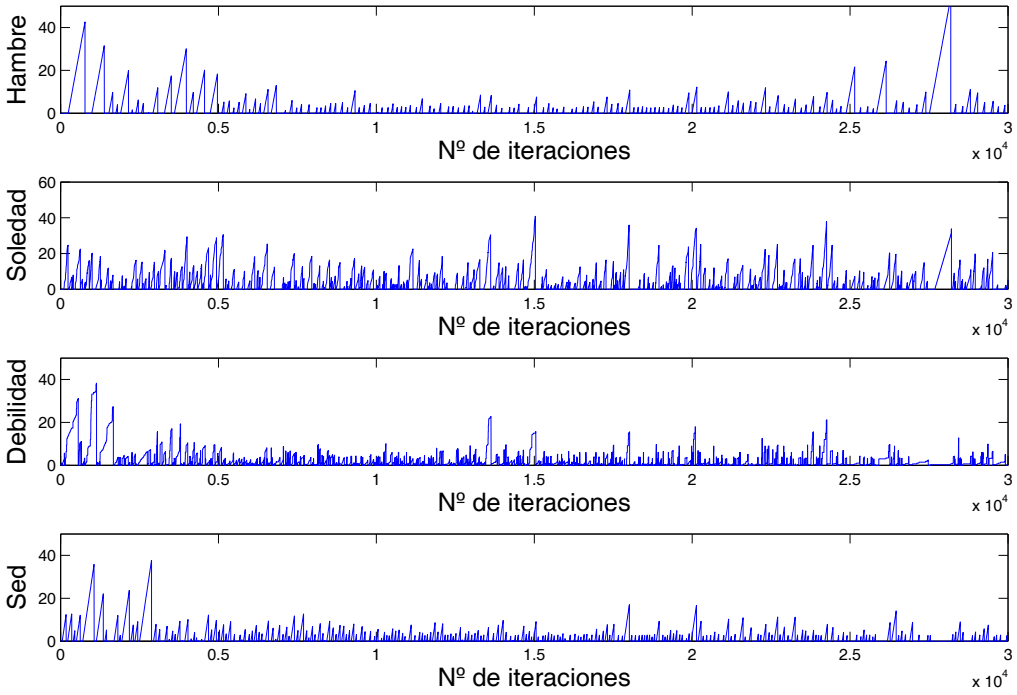


Fig. 9.20: Drives *del agente cuando vive en un mundo mixto utilizando el Media-Q*

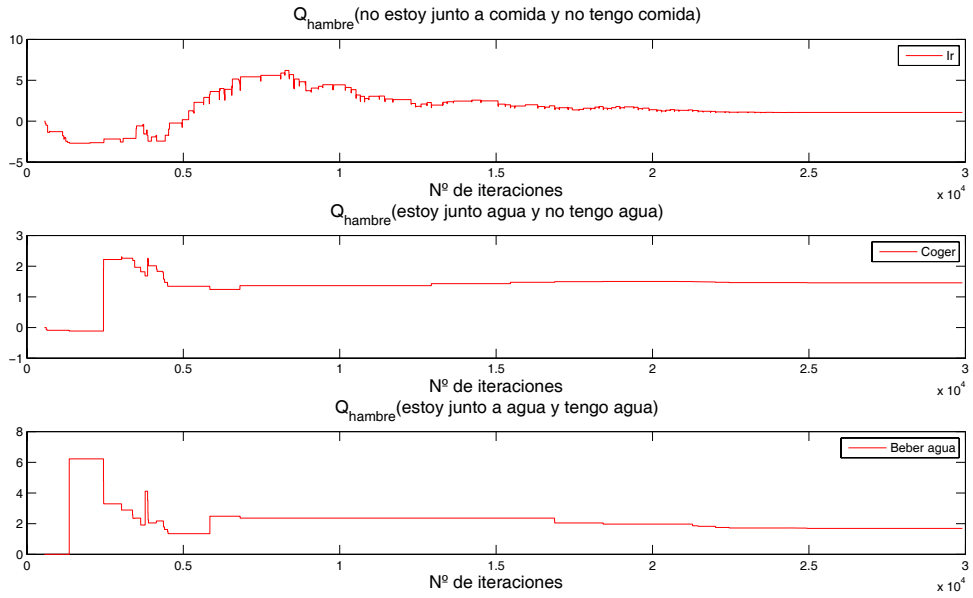


Fig. 9.21: *Valores Q cuando la motivación dominante es Hambre, en un mundo mixto utilizando el Media-Q*

En este caso, el valor de la interacción con otros agentes es sólo mayor para el caso de la motivación dominante Soledad. El agente sólo querrá interaccionar con otros, cuando lo "necesite". Esta es la principal diferencia en relación a considerar a los otros como amigos (casi siempre se quiere interaccionar) o como enemigos (se acaba por no querer interaccionar nunca).

9.4.4. Q-learning en el mundo mixto

Con este algoritmo el agente obtuvo los mejores valores de los indicadores de análisis del bienestar. Tanto el valor medio como el porcentaje de permanencia en la zona de seguridad son los mayores obtenidos por el agente viviendo en el mundo mixto. Esto puede ser debido a que con este algoritmo el agente valora sus acciones de interacción social, de forma que se tienen en cuenta las veces que le ha ido bien o mal. De esta manera, el agente quizás es más realista en relación a su oponente.

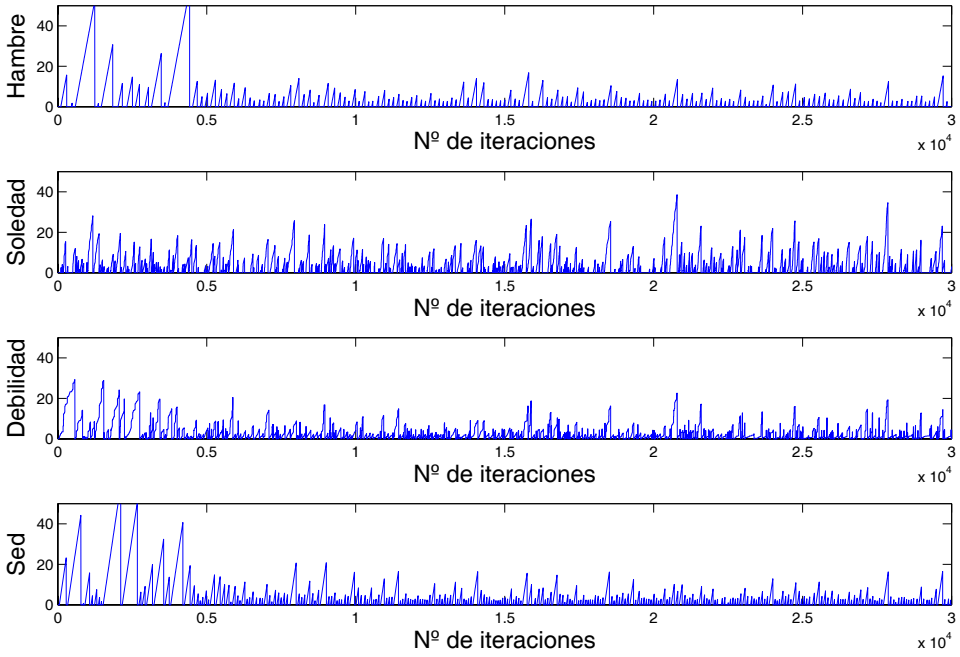


Fig. 9.22: Drives *del agente cuando vive en un mundo mixto utilizando el Q-learning*

Con respecto a la evolución de los *drives*, ver la figura 9.22, se puede observar como, de nuevo, los principales descensos del bienestar mostrados en la figura 9.13, corresponden a los picos mostrados por el *drive* Soledad. De nuevo, estos picos también coinciden con los picos que presenta el *drive* Debilidad, lo que hace suponer que se ha encontrado varias veces con el oponente "malo".

Cuando el agente utiliza el Q-leaning para interaccionar con todos los objetos del mundo, incluidos los otros agentes, la política aprendida es correcta. Esto quiere decir que los *drives* se satisfacen siguiendo una secuencia de acciones lógica, es decir, si el agente tiene sed, va a donde haya agua, la coge y se la bebe, porque los valores Q de dichas acciones son los máximos entre todas los posibles. Esto sucede con todos los *drives*. Es interesante que, cuando el agente necesita interacción social, el agente va a preferir estar quieto antes que moverse, ver la figura 9.23. Cuando otro agente se encuentre con él entonces interactúa, ya que el valor máximo de sus acciones de interacción, que se muestran en la figura 9.24, $Q(saludar) = 4{,}85$, es mayor que el de estar quieto, que vale 3,47. Esta misma secuencia se va llevar a cabo para el caso de que el agente no tenga ninguna motivación dominante.

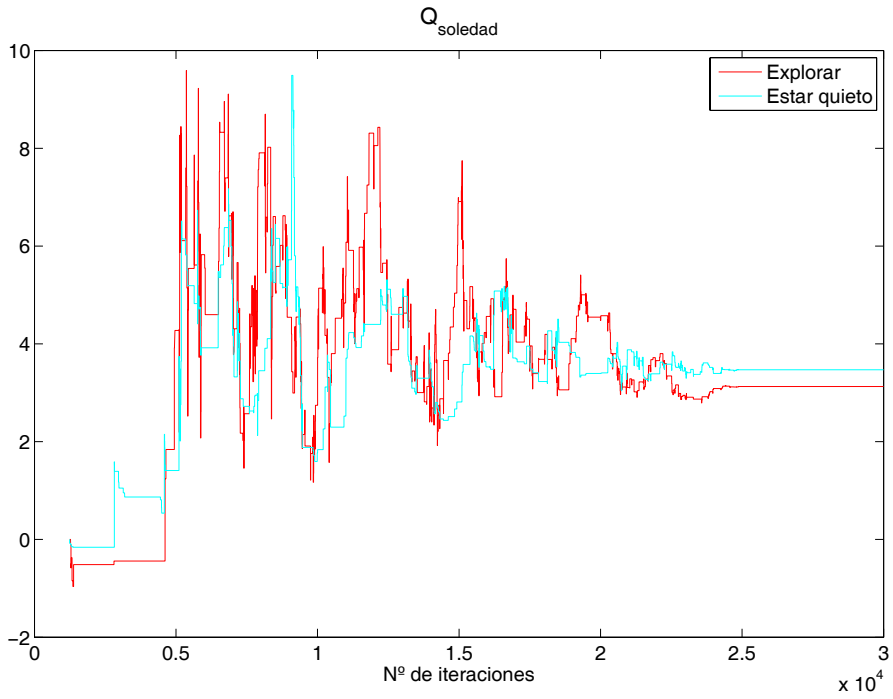


Fig. 9.23: *Valores Q cuando la motivación dominante es Soledad, en un mundo mixto utilizando el Q-learning*

9.4.5. Indicadores del mundo mixto

En la tabla 9.3, se presentan los valores de los indicadores para este mundo. Se aprecia de nuevo que, para el caso del agente que utiliza el algoritmo Enemigo-Q, estos indicadores no se calculan.

En el mundo mixto, de nuevo, el algoritmo del Enemigo-Q hace que el agente no quiera interaccionar con otro agente, debido a las malas experiencias previas. Cuando el agente piensa que los oponentes son sus amigos, con el Amigo-Q, el resultado tampoco es óptimo, ya que la mayoría de las veces va a querer interaccionar sin tener

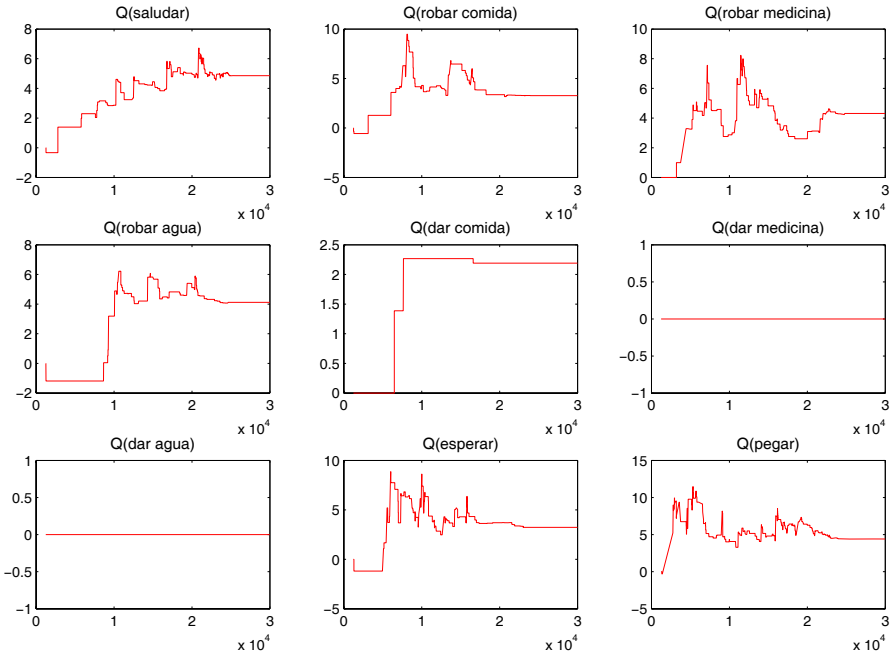


Fig. 9.24: *Valor del vector Q cuando la motivación dominante es Soledad, en un mundo mixto utilizando el Q-learning*

Tab. 9.3: *Indicadores para el mundo mixto*

Algoritmo	*Valor medio del bienestar*	*% de permanencia en ZS*
Amigo-Q	76.6	57.4 %
Enemigo-Q	NO	NO
Media-Q	88.5	58.2 %
Q-learning	91.81	71.0 %

en cuenta que el otro puede que le haga daño. Los mejores resultados se obtienen de nuevo con los algoritmos de Media-Q y Q-learning, aunque dependan de las veces que el agente interacciona con el oponente bueno o malo.

9.5. Mundo malo

El agente en este caso, intenta sobrevivir en un mundo en el que todos sus oponentes son malos. Por lo tanto, las probabilidades de que algo bueno le pase cuando interactúe con los otros agentes es muy baja. Por este motivo, tal y como se muestra en la figura 9.25, la actuación del agente es similar para todos los algoritmos de inter-

acción social. El agente no es capaz de sobrevivir con un valor del bienestar aceptable, ya que con todos ellos, dichos valores medios son negativos. Por lo tanto para este mundo no tiene sentido el cálculo de los indicadores de análisis de resultados.

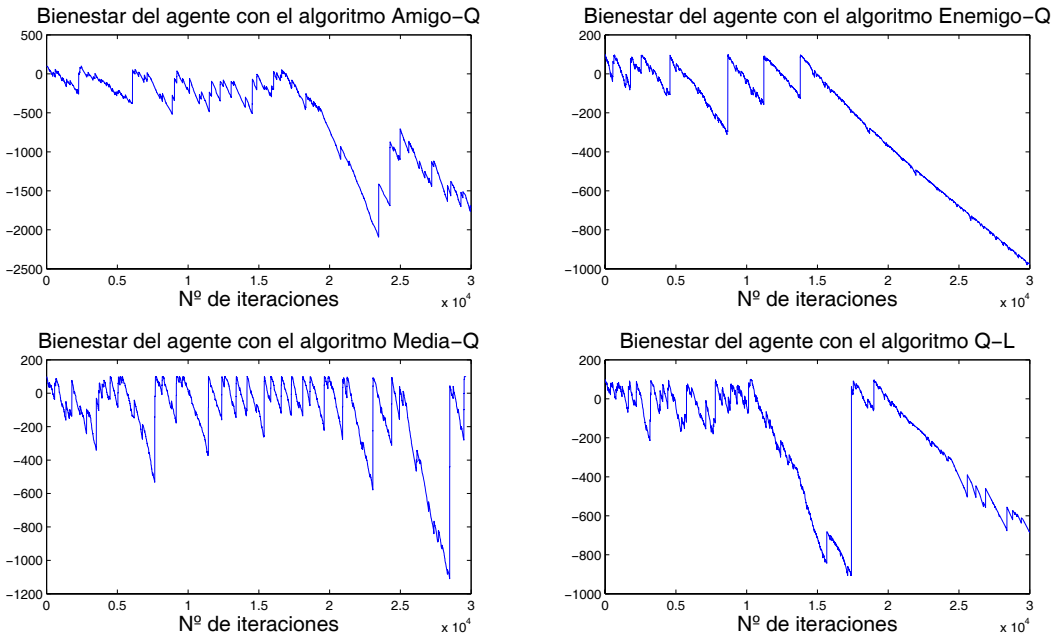


Fig. 9.25: *Bienestar del agente cuando vive en un mundo malo, para cada algoritmo de interacción social utilizado*

Lo interesante de este caso es que, a pesar de que es lógico pensar que todos estos resultados se deben a las mismas causas, esto no es así. Lo razonable sería asumir que el motivo por el que el bienestar no logra "recuperarse" permaneciendo en valores negativos sería la falta de interacción social "positiva". Esto causaría el crecimiento continuo de Soledad, lo que explica la tendencia negativa del bienestar. Esta explicación es cierta para todos los algoritmos salvo para el Amigo-Q. A continuación se van a estudiar las diferencias entre las actuaciones del agente utilizando cada uno de los algoritmos de interacción social.

9.5.1. Amigo-Q en mundo malo

Tal y como se ve reflejado en la figura 9.26, es el *drive* Hambre el que crece de forma indefinida desde casi la mitad de su vida. Por el contrario, el *drive* Soledad sí que llega a ser satisfecho en varias ocasiones. Esto es debido a la propia naturaleza del algoritmo Amigo-Q. Como ya se ha expuesto a lo largo de este capítulo, este algoritmo hace que el agente considere los valores Q máximos como resultado de la interacción con otro agente. Puede ocurrir que una sola interacción positiva cause un valor Q muy alto y que sea la máxima, sin importar que sea el resultado de una única interacción.

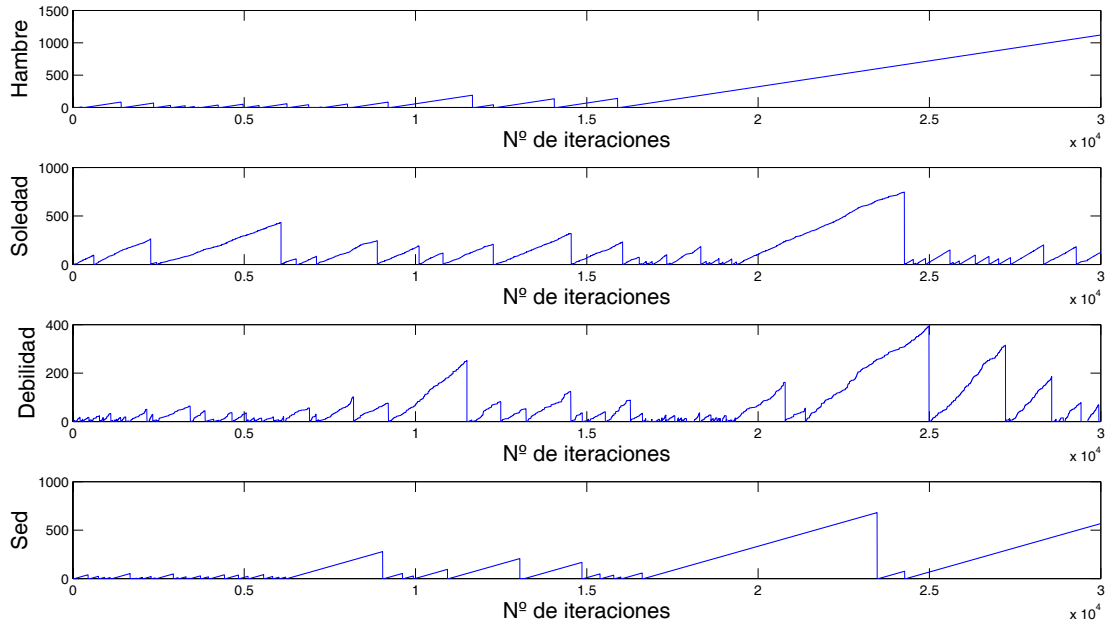


Fig. 9.26: Drives *del agente cuando vive en un mundo malo utilizando el Amigo-Q*

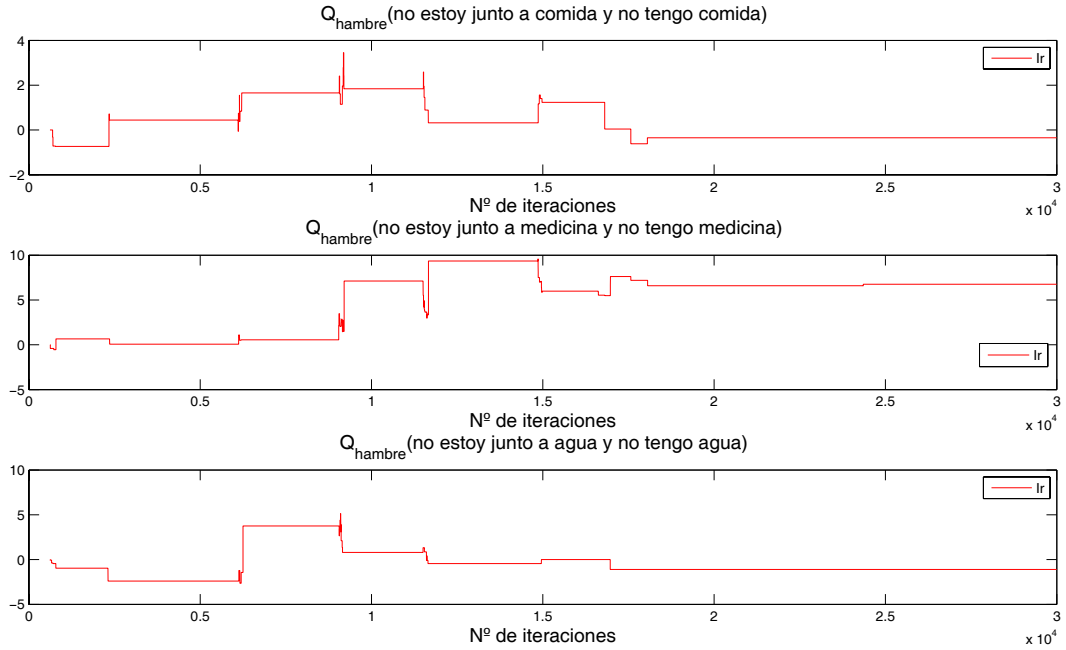


Fig. 9.27: *Valores Q cuando la motivación dominante es Hambre, en un mundo malo utilizando el Amigo-Q*

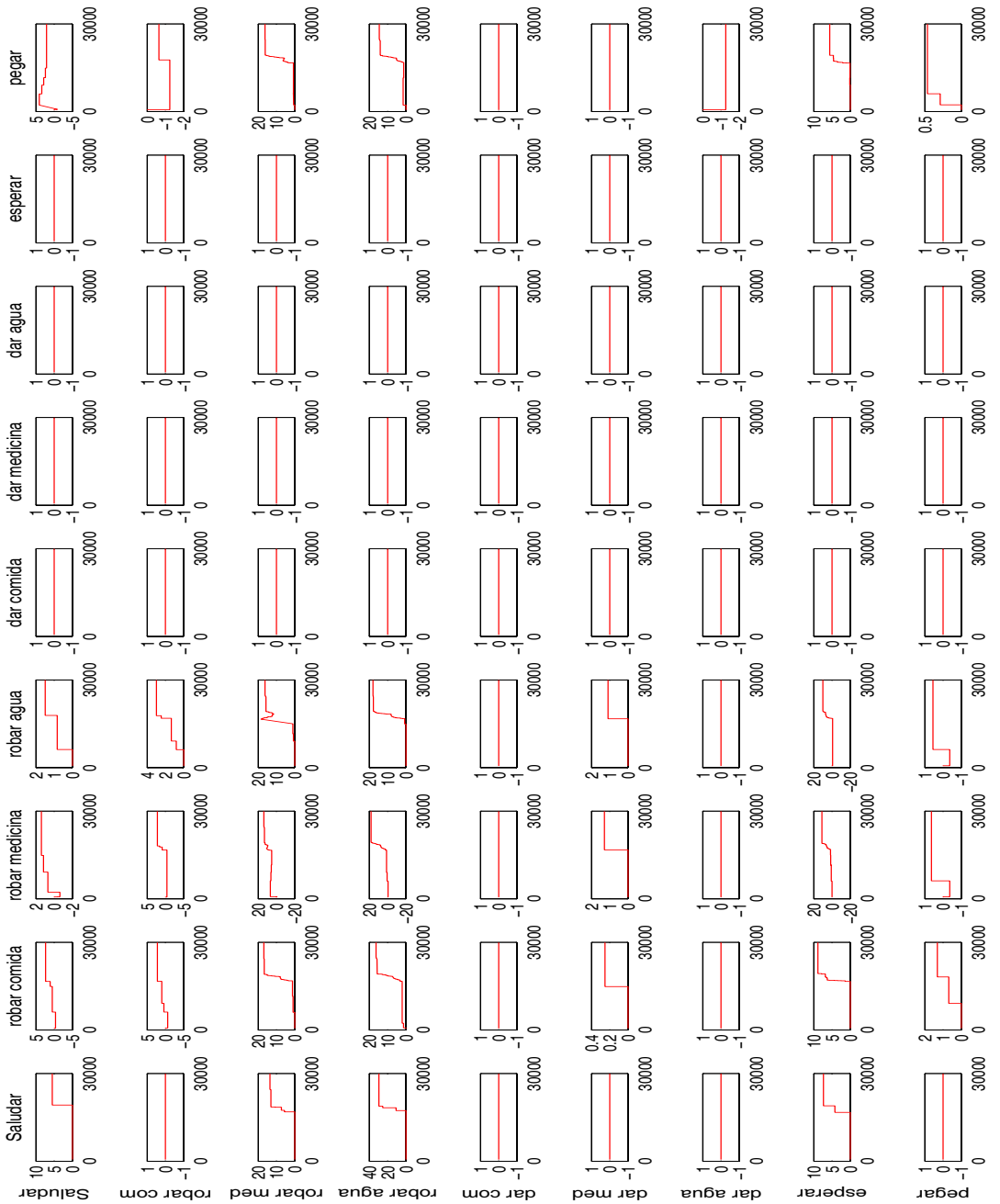


Fig. 9.28: *Valor de la matriz $Q(a_1, a_2)$ cuando la motivación dominante es Hambre, en un mundo malo utilizando el Amigo-Q*

También se observa cómo el *drive* Sed tampoco se satisface durante toda la fase permanente. Sin embargo, el *drive* Debilidad sí que es satisfecho varias veces mientras Hambre es la motivación dominante. Esto se debe a que el valor Q de "ir a por medicina", 6,7, es mayor que los valores Q de "ir a por comida", −0,35, e "ir a por agua", −1,1, tal y como se muestra en la figura 9.27. Por lo tanto, mientras el agente esté sólo y tenga hambre, éste irá donde haya medicina. Si mientras camina se encuentra a otro agente, debido a los valores altos de la matriz de interacción $Q(a_1, a_2)$, la figura 9.28, interaccionará con él. Esta interacción será muy negativa para el agente tal y como se puede deducir de la evolución de los *drive* Soledad y Debilidad, la figura 9.26.

9.5.2. Enemigo-Q en mundo malo

Un caso muy distinto es el del agente que utiliza el algoritmo Enemigo-Q. En la figura 9.25 se ve cómo el bienestar del agente comienza a disminuir de forma continua a partir de la mitad de su vida. Hay que recordar que con este algoritmo, el agente asume que los otros agentes son sus enemigos, por lo que, al igual que ocurría en el mundo mixto, el agente finalmente decide no interaccionar nunca más, tal y como se muestra en la figura 9.29.

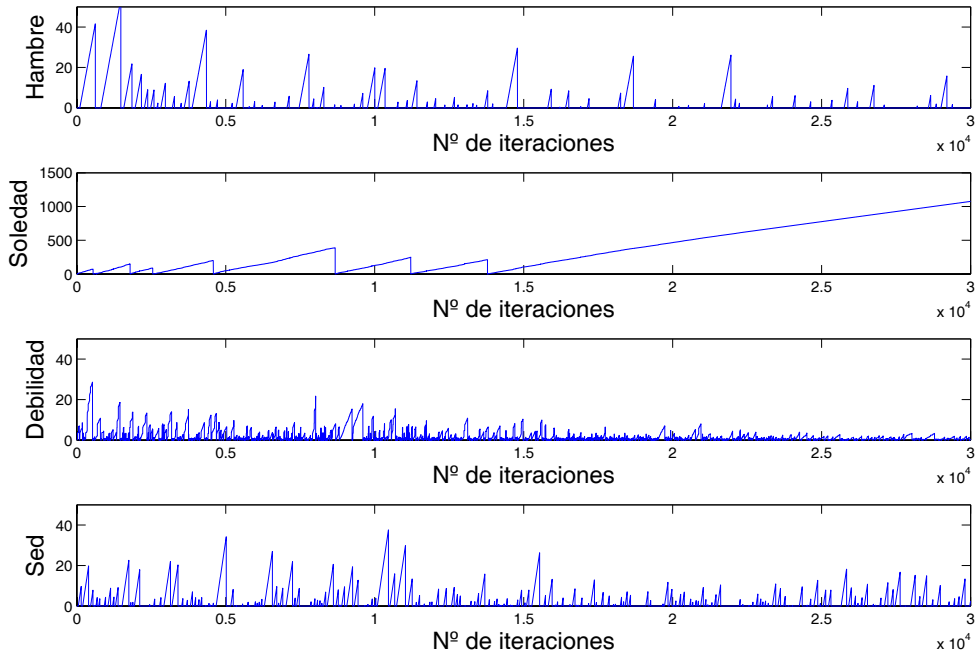


Fig. 9.29: Drives *del agente en un mundo malo utilizando el Enemigo-Q*

Lo interesante de este caso es que a pesar de que el *drive* Soledad nunca se satisface, los otros *drives* sí que llegan a satisfacerse. De hecho, parece que el agente sigue una política de comportamiento con el fin de satisfacerlos. Esto tiene bastante sentido, ya que el agente recibe su refuerzo no por el valor del bienestar, sino por su variación.

9.5.3. Media-Q en mundo malo

Para el caso del agente utilizando el algoritmo Media-Q como algoritmo de interacción social, parece que sucede algo similar. La evolución de los *drives* presenta un gran crecimiento del *drive* Soledad mientras que los otros *drives* son satisfechos en varias ocasiones, ver la figura 9.30. La diferencia es que ahora el *drive* Soledad sí que llega a ser satisfecho durante la fase permanente, lo que indica que sí que llega a interaccionar con otros agentes.

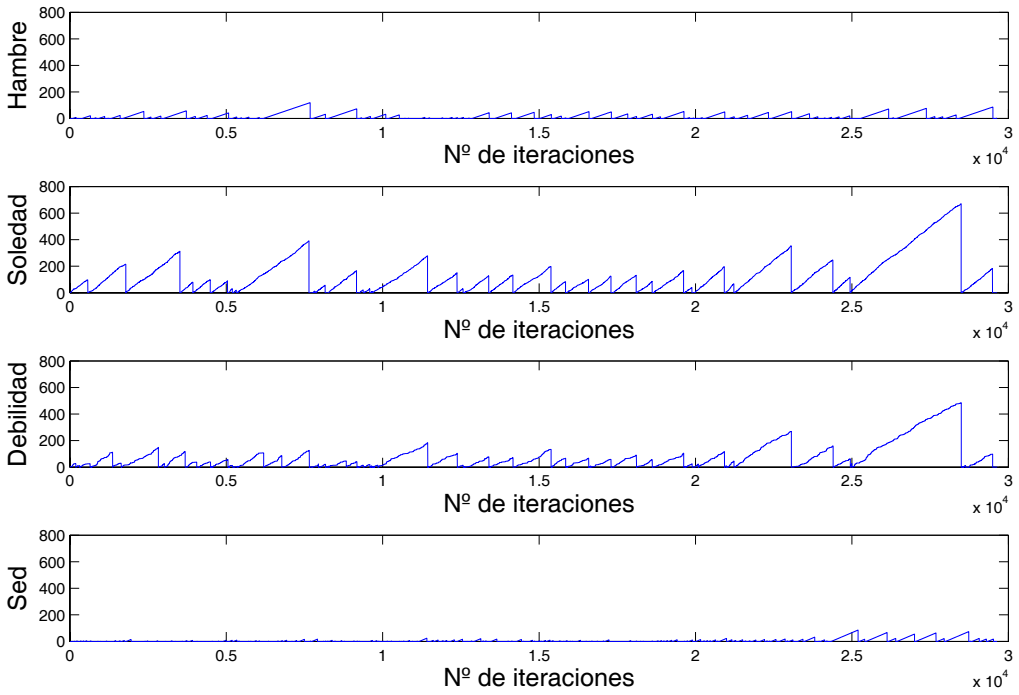


Fig. 9.30: Drives *del agente cuando vive en un mundo malo utilizando el Media-Q*

En la figura 9.31 se representan los valores Q de la matriz de interacción social cuando la motivación dominante es Soledad. Cuando el agente utiliza el algoritmo Media-Q, el valor de cada una de sus acciones es el valor medio de cada fila. Como se puede apreciar, el valor medio máximo corresponde con la acción "robar agua", y es aproximadamente 22. Este valor es mayor que el valor Q máximo de las acciones que puede realizar cuando el agente está solo, y que es aproximadamente 12. El motivo de que sea tan alto el valor de interaccionar con otro agente, a pesar de vivir en un mundo malo, se debe a que el refuerzo recibido cuando un oponente "malo" hace algo bueno, es muy grande. Esto hace que el valor medio total de las acciones propias del agente aumente. Por lo tanto, cuando el agente necesita interacción social, interaccionará con otro agente, aunque el tiempo que transcurra antes de que su *drive* Soledad se satisfaga, sea muy largo.

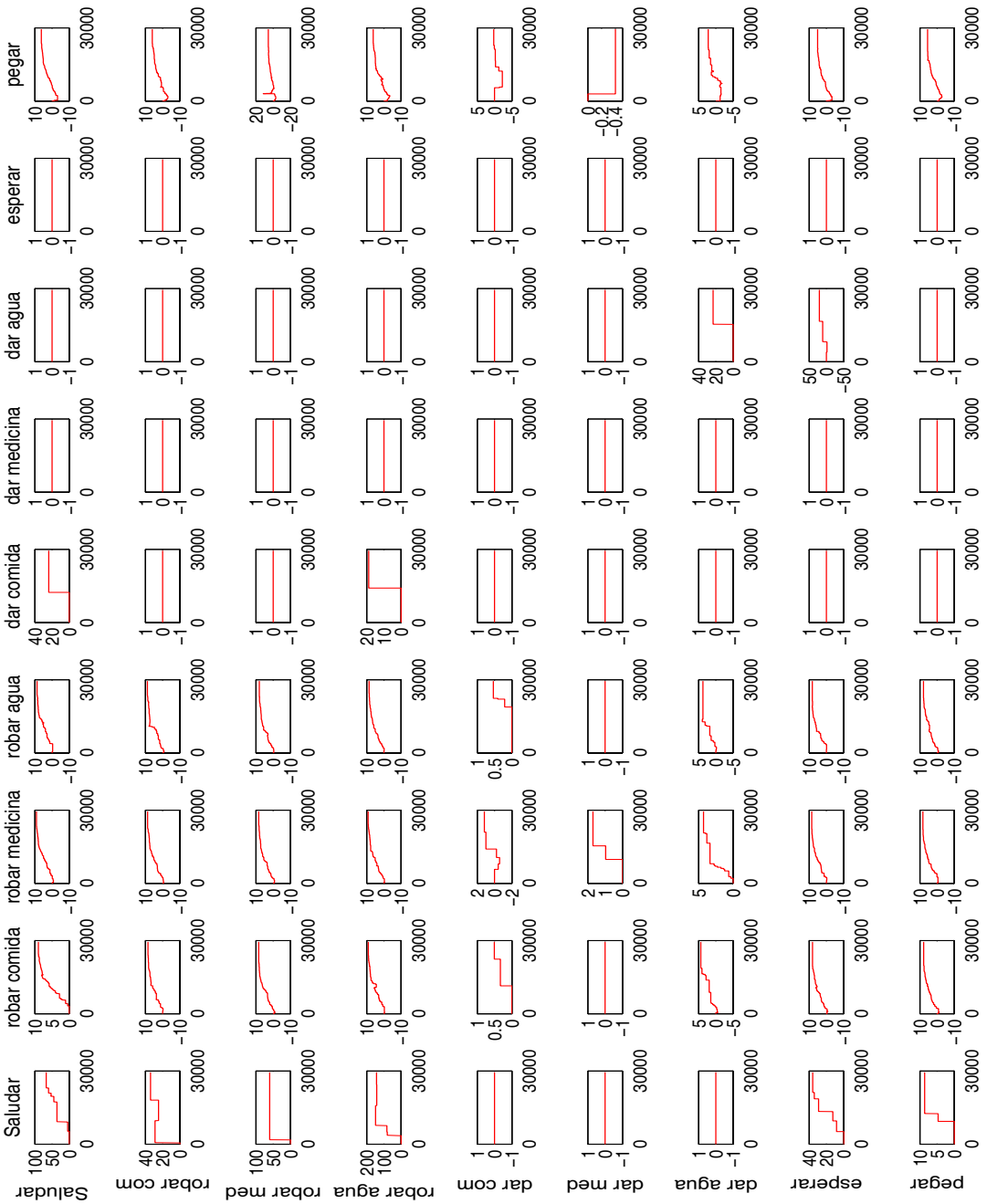


Fig. 9.31: *Valor de la matriz $Q(a_1, a_2)$ cuando la motivación dominante es Soledad, en un mundo malo utilizando el Media-Q*

9.5.4. Q-learning en mundo malo

Por último, cuando el agente utiliza el Q-learning como algoritmo de interacción social, la evolución de los *drives* es similar al caso del Enemigo-Q, ver la figura 9.32. De nuevo parece que el *drive* Soledad crece de forma indefinida, por lo que el agente decide no volver a interaccionar con otros agentes.

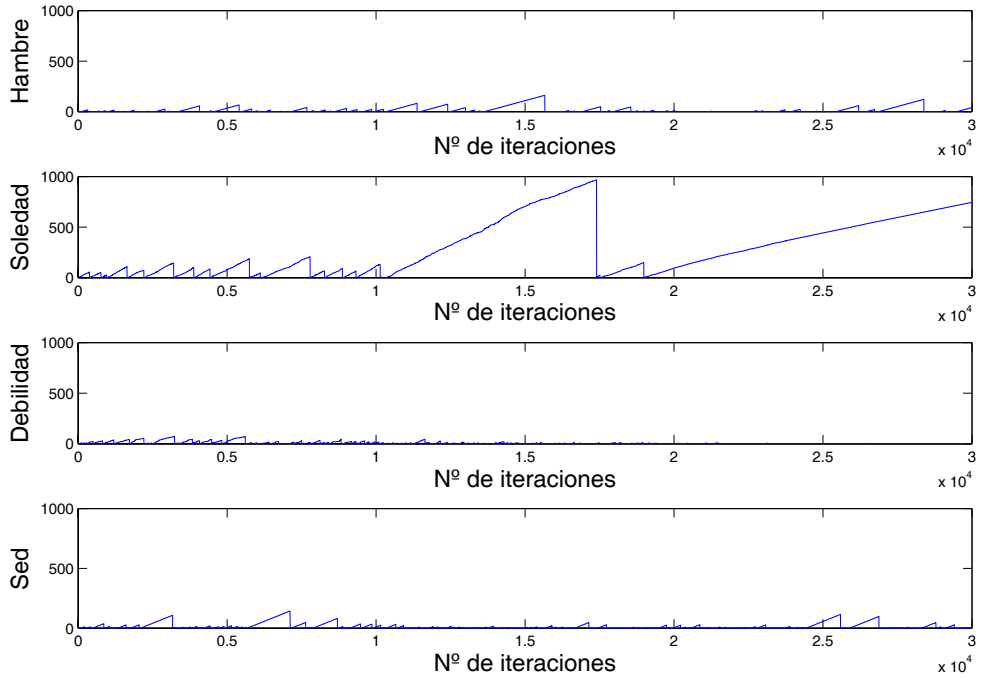


Fig. 9.32: Drives *del agente cuando vive en un mundo malo utilizando el Q-learning*

Esto, de nuevo, es debido a que este algoritmo toma en cuenta los efectos de todas las acciones de sus oponentes y la cantidad de veces que han sido ejecutadas, no sólo lo mejor o lo peor. En este mundo, la mayoría de las interacciones tienen un efecto negativo, por lo que es lógico que el valor de sus acciones sea bajo. Como consecuencia el agente decide no volver a interaccionar con otros agentes.

9.6. Resumen y conclusiones

En este capítulo se han presentado y analizado las actuaciones de un agente que tiene el *drive* Soledad y por lo tanto necesita de la interacción social. Al ser un agente social, va a necesitar interaccionar con otros agentes presentes en el mismo entorno. Dependiendo de la personalidad de estos otros agentes, se definen cuatro tipos de mundos: bueno, neutro, mixto y malo.

Tal y como fue expuesto en la sección 3.4.1, cuando el agente interacciona con otro agente, las recompensas recibidas son debidas a las acciones de ambos. Existen distintos tipos de algoritmos de aprendizaje multiagente que definen la forma de

interaccionar con un oponente. En este capítulo se muestran los resultados obtenidos cuando el agente utiliza los siguientes algoritmos de aprendizaje multiagente: Amigo-Q, Enemigo-Q y Media-Q. Además también se utiliza el algoritmo de aprendizaje Q-learning, en el que no se tiene en cuenta que el refuerzo recibido, durante la interacción social, es debido a la acción del oponente. Cada uno de estos algoritmos fue probado en cada uno de los mundos llegándose a las conclusiones que se exponen a continuación. En la tabla 9.4, se muestran los valores de los indicadores de análisis de los resultados obtenidos para cada uno de los casos, durante la fase permanente.

Tab. 9.4: *Resumen de los Indicadores de Análisis de Resultados*

Algoritmo	***M.bueno***		***M.neutro***		***M.mixto***		***M.malo***	
	VM	*%ZS*	*VM*	*%ZS*	*VM*	*%ZS*	*VM*	*%ZS*
Amigo-Q	98.5	100	85.0	45.6	76.6	57.4	NO	NO
Enemigo-Q	98.6	100	NO	NO	NO	NO	NO	NO
Media-Q	98.5	100	92.84	70.9	88.5	58.2	NO	NO
Q-learning	99.2	100	90.68	58.7	91.81	71.0	NO	NO

Tanto el algoritmo de Amigo-Q, como el de Enemigo-Q son demasiado "radicales". No hay que olvidar que el Amigo-Q está diseñado para un juego de colaboración entre agentes con un objetivo común. Por otro lado, el Enemigo-Q, es el ideal en un juego de competición, llamados juegos de suma cero. El entorno en el que se encuentran los agentes no cumple ninguna de las dos características.

La principal característica del Amigo-Q es la de suponer que cuando interactúa con otro agente, éste va a realizar la acción que es mejor para los dos. Es decir, siempre será muy optimista y por lo tanto tiene una gran tendencia a interaccionar. Esto da buenos resultados en el caso de que el agente se encuentre en un mundo bueno, ya que todas las interacciones van a tener consecuencias positivas. Para el resto de los mundos esto no va a dar tan buenos resultados, ya que tanto en el mundo neutro como en el mixto, va a depender un poco de la "suerte" del agente de encontrarse con oponentes malos. En el caso del mundo malo, el hecho de interaccionar muchas veces provoca que se "descuiden" otras necesidades.

El agente, cuando utiliza como algoritmo de aprendizaje multiagente el Enemigo-Q, va a asumir que los otros agentes son sus enemigos. Es decir, va a asumir que el otro agente va a elegir la acción que más le va a perjudicar. Esto provoca que el agente va a intentar interaccionar el menor número de veces. En el caso de que viva en un mundo bueno no hay problema ya que lo peor que le puede pasar es "bueno".

Sin embargo, en el resto de los mundos donde existe la posibilidad de que le hagan algo "malo", el agente al intentar protegerse, acaba por no interaccionar con nadie y por lo tanto se la Soledad crece indefinidamente.

Cuando el agente utiliza como algoritmo de aprendizaje multiagente el Media-Q, no va a considerar sólo lo mejor o lo peor que le ha pasado mientras interactuaba con otro agente. La bondad de este algoritmo radica en considerar como valor Q de sus acciones, el valor medio de todos los valores Q obtenidos para cada una de sus acciones, es más realista. Esto hará que cuando se encuentre en un mundo neutro o mixto, "modere" la cantidad de interacciones sociales. Es decir, tal y como se ha mostrado, el agente querrá interaccionar cuando lo necesite, cuando su motivación dominante sea Soledad. En un mundo malo, como se ha visto, a pesar de que la probabilidad de que al interaccionar le ocurra algo malo es muy alta, el hecho de que algunas veces ocurra algo bueno hace que la media de los valores Q aumente. En este caso, de nuevo, querrá interaccionar sólo cuando lo necesite.

Por último, si el agente utiliza el Q-learning como algoritmo de interacción social, ignora el hecho de que la recompensa o castigo que recibe mientras interactúa con otro agente es el resultado de su acción conjunta. Por lo tanto, cuando está interactuando, el valor Q final de cada una de sus acciones será un valor que pondera las experiencias positivas y las negativas. En un mundo neutro o mixto, donde el número de experiencias negativas y positivas son casi iguales, el resultado final es similar al del agente utilizando el algoritmo Media-Q. Sin embargo, en el mundo malo, el número de experiencias negativas resultantes de la interacción con otro agente es mucho mayor que el de las positivas, por lo que parece razonable que decida finalmente no interaccionar durante la fase permanente.

10. RESULTADOS EXPERIMENTALES: AGENTE CON MIEDO

10.1. Introducción

En este capítulo, se van a presentar los resultados obtenidos cuando se añade en el agente la emoción miedo. Tal y como se introdujo en la sección 6.6, se van a considerar dos tipos de miedo: miedo a realizar acciones arriesgadas y miedo a estar en un estado peligroso. El miedo, en este segundo caso, es considerado como un *drive* más del agente y por lo tanto puede llegar a ser la motivación dominante.

10.2. Agente con miedo a realizar acciones arriesgadas

10.2.1. Descripción del experimento

El miedo a realizar acciones arriesgadas es el miedo a hacer algo que puede tener efectos muy negativos. Como ya se ha visto en la sección 6.6.1, cuando la probabilidad de que una acción tenga efectos negativos es alta, el agente aprende a tener miedo a ejecutarla utilizando el Q-learning. Sin embargo, cuando la probabilidad de que una acción sea nociva es baja, es decir, sólo es mala ocasionalmente, el Q-leaning no hace que el agente tenga miedo a estas acciones. Por lo tanto, se introduce un mecanismo que complementa al Q-learning, de manera que, no sólo se consideran los valores medios de las acciones, sino también los peores resultados obtenidos.

En esta sección se va a implementar este mecanismo que hace que el agente tenga miedo a realizar acciones que son nocivas, aunque sólo sea ocasionalmente. Ninguno de los objetos ya implementados hasta ahora tiene ningún efecto negativo, por lo que no existe ningún riesgo. Por ello, con el fin de demostrar la utilidad del miedo a realizar acciones arriesgadas, se implementa un nuevo objeto en el entorno: el elixir. De manera que si el agente bebe el elixir:

- El 95 % de la veces:

$$D_{hambre}^{k+1} = 0$$
$$D_{debilidad}^{k+1} = 0 \tag{10.1}$$
$$D_{sed}^{k+1} = 0$$

- El 5 % de la veces:

$$D_{hambre}^{k+1} = D_{hambre}^{k} + 4$$
$$D_{debilidad}^{k+1} = D_{debilidad}^{k} + 4 \tag{10.2}$$
$$D_{sed}^{k+1} = D_{sed}^{k} + 4$$

Es decir, que este objeto normalmente tiene un gran efecto positivo en el bienestar y ocasionalmente es un "veneno", que provoca que aumente Hambre, Sed y Debilidad. Lo que implica que el bienestar sufre un descenso de magnitud 12.

Para implementar este tipo de miedo, el procedimiento es el siguiente:

- Durante toda la fase de aprendizaje, el agente almacena lo peor que le ha pasado para cada par estado-acción. Tal y como se explicó en la sección 6.6.1, estos valores son independientes del estado interno del agente, es decir, de la motivación dominante.

  La fase de aprendizaje, para este experimento, va a tener una duración más larga que en los experimentos previos. Debido a que se introduce un nuevo objeto, es necesario aumentar el número de iteraciones de esta fase, para permitir que el agente explore todas las acciones, incluidas las del nuevo objeto. Las acciones de este nuevo objetos son las siguientes:

$$A_{elixir} = \{Beber\ elixir,\ Coger,\ Ir\ a\} \tag{10.3}$$

  La forma de almacenar los peores valores está definida por las ecuaciones (6.26) y (6.27), que se vuelven a mostrar a continuación:

$$Q_{peor}^{obj_i}(s,a) = \text{mín}(Q_{peor}^{obj_i}(s,a), r + \gamma \cdot V_{peor}^{obj_i}(s')) \tag{10.4}$$

  Donde:

$$V_{peor}^{obj_i}(s') = \underset{a \in A_{obj_i}}{\text{máx}} (Q_{peor}^{obj_i}(s',a)) \tag{10.5}$$

  es el valor peor del objeto i en el nuevo estado. Estos valores se calculan para cada objeto del mundo, con independencia del estado interno. De manera que al final de la fase de aprendizaje, se tiene una matriz con los peores valores Q registrados para cada acción con cada objeto. La idea es que el miedo está relacionado con el objeto que ha causado daño, no con el estado interno del agente.

- En esta ocasión, la fase permanente, en la que el agente deja de aprender y vive con los valores Q aprendidos, va a ser mucho más extensa que en los experimentos previos. De esta manera se podrá observar la actuación del agente mientras

se varía su factor de atrevimiento β. Este factor de atrevimiento definido en la sección 6.6.1, es el que va a ponderar el efecto de lo peor que le ha pasado al agente frente a lo que vale cada acción de media.

Durante la fase permanente, el agente va a vivir de manera que elige las acciones que hagan máxima la matriz $Q^{obj_i}_{miedo}(s,a)$, definida como:

$$Q^{obj_i}_{miedo}(s,a) = \beta Q^{obj_i}(s,a) + (1-\beta)Q^{obj_i}_{peor}(s,a) \longrightarrow Si\ Q^{obj_i}_{peor}(s,a) \leq L_m$$

$$Q^{obj_i}_{miedo}(s,a) = Q^{obj_i}(s,a) \longrightarrow En\ otro\ caso \tag{10.6}$$

Para los experimentos, este límite L_m se ha fijado, después de varias observaciones experimentales, a un valor igual al refuerzo negativo que recibe el agente cuando el elixir es veneno:

$$L_m = -12 \tag{10.7}$$

De acuerdo con la ecuación (10.6), a medida que el factor de atrevimiento disminuye, el agente tiene más en cuenta lo peor que le ha pasado. Como consecuencia, se comprobará cómo el agente deja de realizar las acciones arriesgadas.

10.2.2. Presentación de resultados

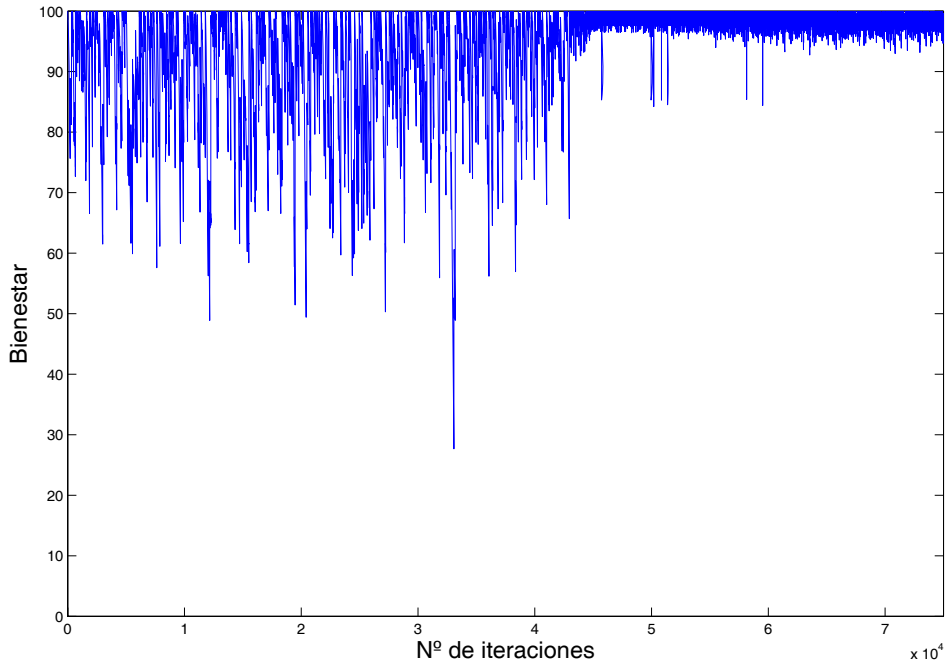


Fig. 10.1: *Bienestar del agente con miedo a realizar acciones arriesgadas*

En la figura 10.1 se muestra la evolución del bienestar del agente a lo largo de las dos fases, la primera de aprendizaje, hasta 45000 iteraciones y la fase permanente, desde 45000 hasta 75000. Tal y como se puede observar, durante la fase de aprendizaje, el bienestar del agente presenta importantes descensos. Estos descensos son debidos en su mayoría, a la cantidad de veces que tomó el elixir y éste le sentó mal, provocando los picos en los *drives* Hambre, Debilidad y Sed, ver figura 10.2.

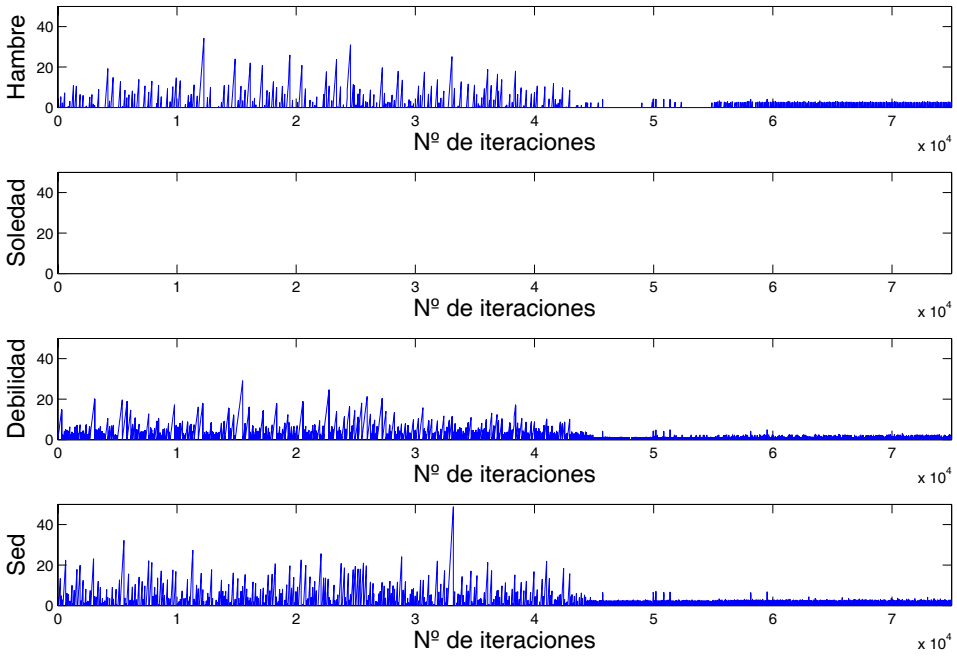


Fig. 10.2: Drives *del agente con miedo a realizar acciones arriesgadas*

En este experimento, la fase permanente ha sido dividida en distintas zonas con valores del factor de atrevimiento β distintos. Tal y como se observa en la figura 10.3, durante la fase permanente, el bienestar varía a medida que disminuye el factor de atrevimiento β. Esta variación está directamente relacionada con la cantidad de veces que el agente bebió el elixir y le sentó mal. De hecho, se puede observar como a medida que el factor de atrevimiento disminuye, los descensos debidos al "envenenamiento" van desapareciendo. Al mismo tiempo se detecta una tendencia de disminución del valor medio del bienestar.

En la tabla 10.1 se muestran los valores de β y de los indicadores de análisis del bienestar para cada etapa, al igual que el número de veces que el agente bebió el elixir y el número de veces que le sentó mal (número de picos). Los valores mostrados, confirman que a medida que el agente deja de tomar el elixir, debido a que le tiene miedo, el valor medio del bienestar disminuye. Al mismo tiempo, el porcentaje de permanencia en la zona del bienestar aumenta, ya que al no tomar el elixir, no existe la posibilidad de envenenamiento. De acuerdo con la definición de $Q^{obj_i}_{miedo}(s,a)$, los

resultados que se obtendrían para valores inferiores del factor de atrevimiento serían similares a los obtenidos en las últimas etapas con $\beta = 0{,}5$. Esto es debido a que una vez que se ya no se consideran las acciones arriesgadas, la política de comportamiento será siempre la misma.

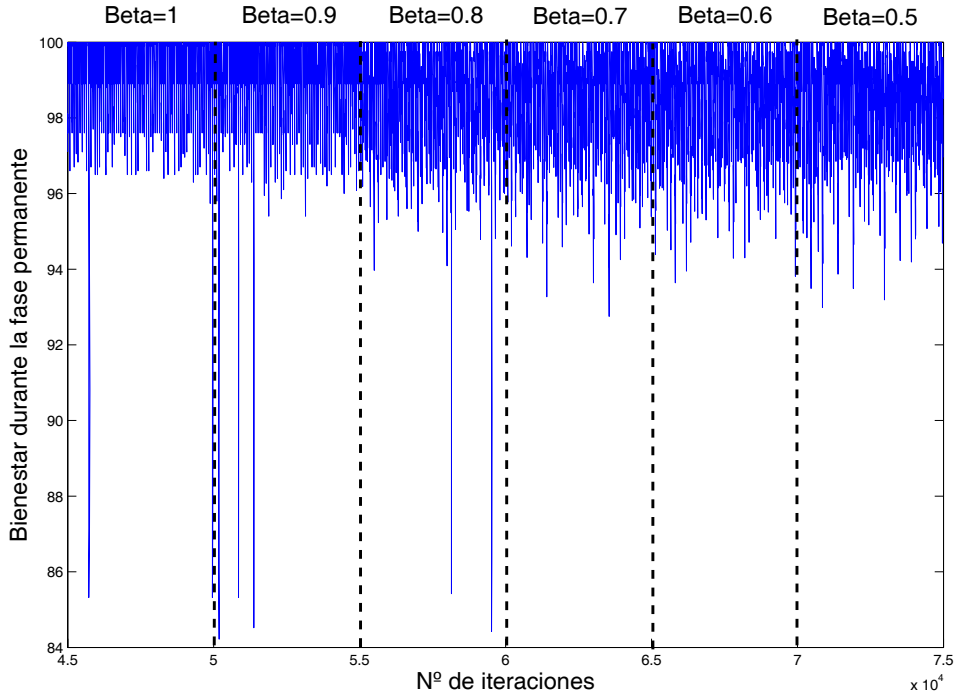


Fig. 10.3: *Bienestar del agente durante la fase permanente a medida que se varía el factor de atrevimiento β*

Tab. 10.1: *Resultados de la fase permanente*

Fase permanente	*Valor de β*	*Nº bebe el elixir*	*Nº picos*	*Wb Medio**	*%ZS**
de 45000 a 50000	1	68	2	99.3	99.9
de 50000 a 55000	0.9	68	3	99.1	99.88
de 55000 a 60000	0.8	21	2	98.69	99.9
de 60000 a 65000	0.7	0	0	98.56	100
de 65000 a 70000	0.6	0	0	98.51	100
de 70000 a 75000	0.5	0	0	98.42	100

** Wb Medio: Valor medio del bienestar *%ZS: Porcentage de permanencia en la zona de seguridad*

A continuación se va a presentar el análisis de los valores Q de las acciones relacionadas con los objetos, para cada motivación dominante. Posteriormente se mostrarán los peores valores Q de las acciones relacionadas con el elixir.

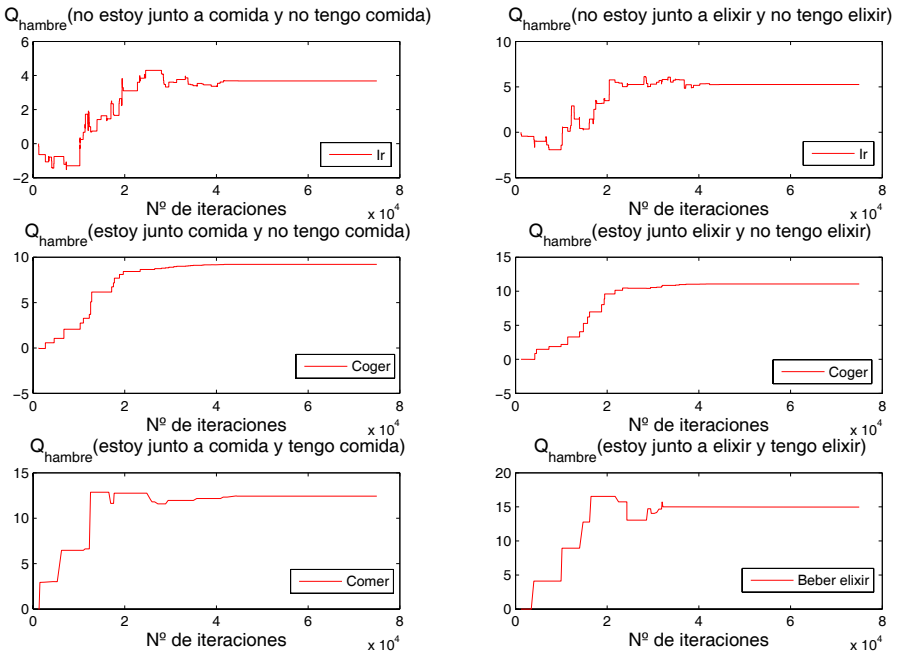


Fig. 10.4: *Valores Q de las acciones relacionadas con comida y elixir, cuando la motivación dominante es Hambre*

Cuando la motivación dominante del agente es **Hambre**, en la figura 10.4 se muestran los valores de las acciones relacionadas con la comida y con el elixir. En dicha gráfica se puede apreciar que los valores Q correspondientes a las acciones relacionadas con la comida, cuando tiene hambre, son inferiores a los correspondientes con el elixir. Por lo que cuando el agente sin miedo tiene hambre, va a preferir ir a donde está el elixir, lo cogerá y lo beberá.

Cuando el agente tiene **sed**, se puede observar en la figura 10.5, que los valores relacionados con el elixir son superiores a los del agua. Por lo que el agente, de nuevo, va a preferir beber el elixir cuando tiene sed, que beber agua.

Por el contrario, cuando el agente tiene como motivación dominante **Debilidad**, los valores relacionados con el elixir son ahora ligeramente inferiores a los de la medicina. Esto va a causar que cuando el agente esté débil, prefiera tomar medicina antes que el elixir, ver la figura 10.6.

Por último, cuando el agente **no tiene ninguna motivación dominante**, los valores Q de mayor valor son los de las acciones relacionadas con la medicina, ver la figura 10.7. Los valores relacionados con el resto de objetos son todos inferiores y no se muestran.

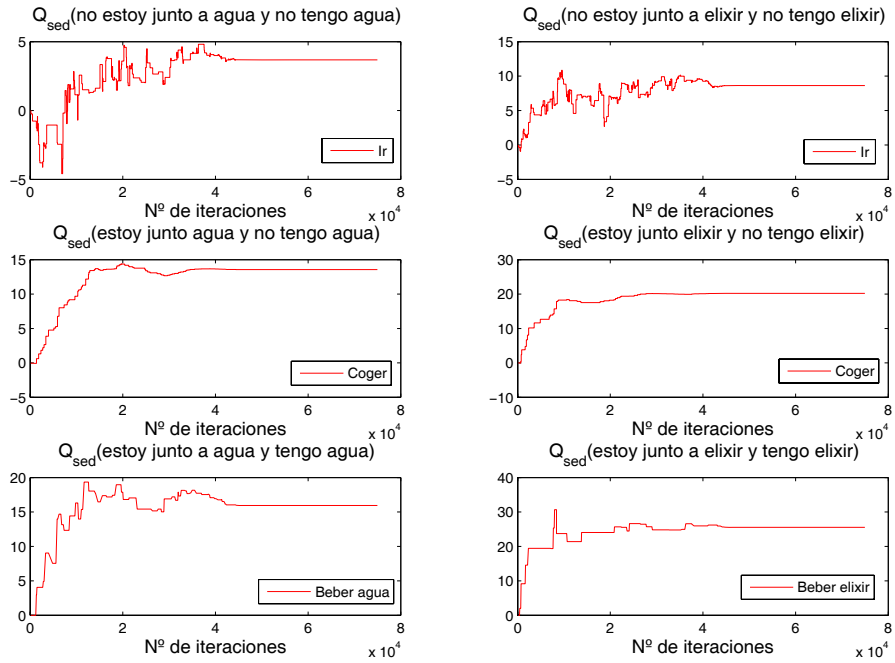


Fig. 10.5: *Valores Q de las acciones relacionadas con el agua y el elixir, cuando la motivación dominante es Sed*

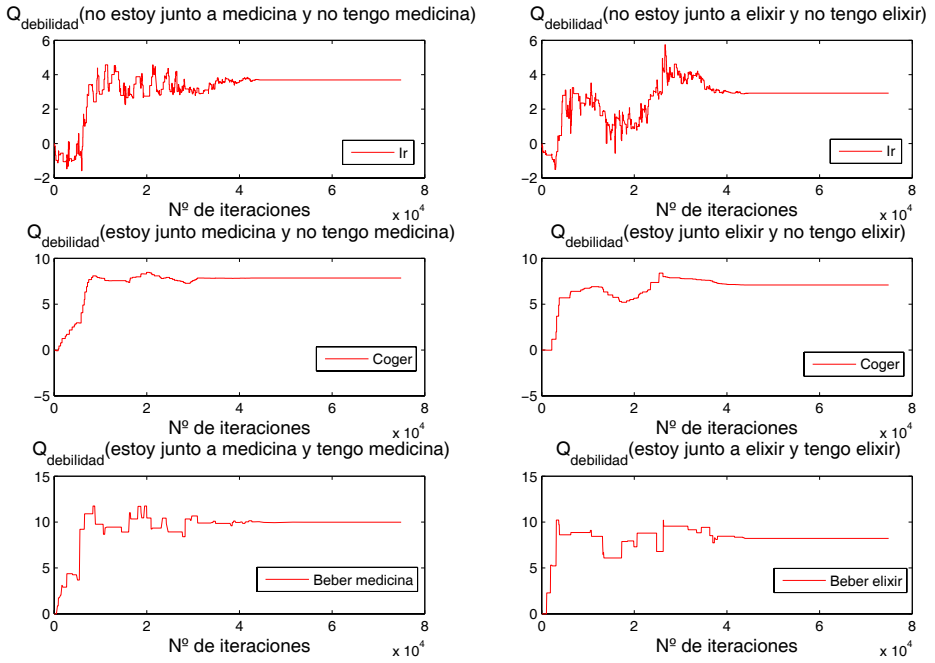


Fig. 10.6: *Valores Q de las acciones relacionadas con la medicina y el elixir, cuando la motivación dominante es Debilidad*

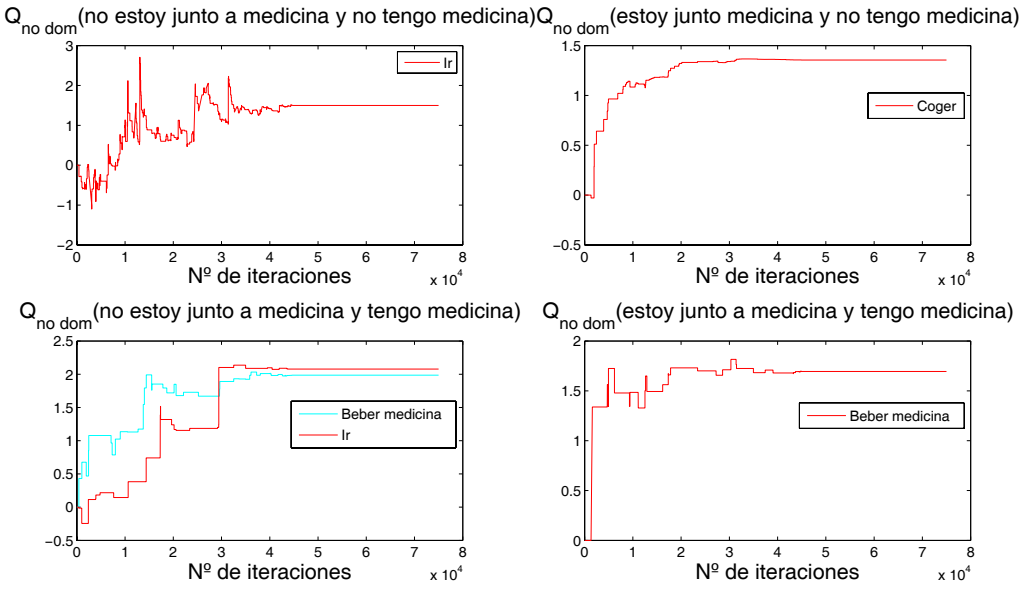


Fig. 10.7: *Valores Q de las acciones relacionadas con la medicina, cuando no hay motivación dominante*

Por lo tanto, como **resumen**, cuando el agente no considera lo peor que le ha pasado, es decir, el factor de atrevimiento es $\beta = 1$, el agente vive según los valores previamente expuestos. Es decir, el agente va a preferir tomar el elixir cuando tiene hambre y cuando tiene sed, ya que la mayoría de las veces es muy positivo. Sin embargo, cuando el agente está débil o no tiene motivación dominante va a preferir tomar la medicina.

Cuando el agente considera lo peor que le ha pasado, es cuando la política de comportamiento cambia. Los peores valores registrados para las acciones relacionadas con la comida, el agua y la medicina son valores negativos, pero cercanos al 0. Esto es debido a que nunca pasa nada realmente malo cuando se trata con estos objetos. Por el contrario, el elixir, ocasionalmente tiene efectos muy negativos, por lo que es de esperar que los peores valores registrados sean bajos.

En la figura 10.8 se muestran los peores valores registrados relacionados con el elixir. Tal y como se esperaba, estos valores son todos inferiores al límite propuesto en (10.7), por lo que de acuerdo con (10.6), los valores de $Q^{obj_i}_{miedo}(s, a)$ en relación al elixir van a ir disminuyendo a medida que β aumente.

Por lo tanto, dependiendo del valor de β, llegará un momento en el que el agente no vuelva a elegir tomar el elixir, ya que su valor es muy pequeño. Es interesante que no sólo el agente decide no "bebe elixir" sino que tampoco realiza ninguna acción que le lleve a ello. Esto es debido a que, como se muestra en la figura 10.8, los valores de "ir a por elixir" y "coger elixir" son también inferiores al límite L_m.

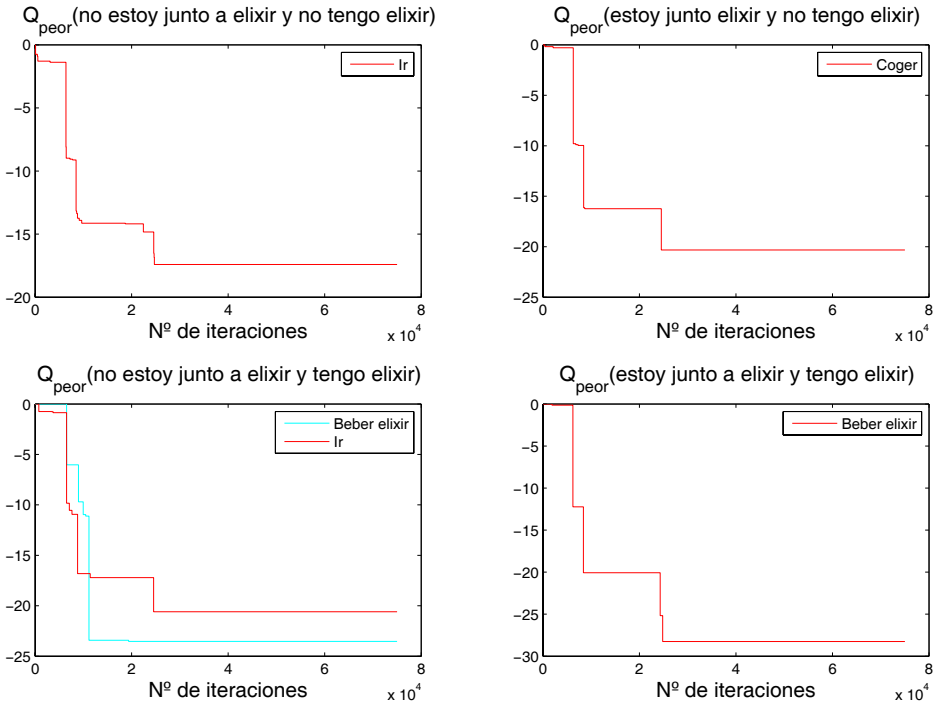


Fig. 10.8: *Los peores valores Q de las acciones relacionadas con el elixir*

Resumiendo, al principio el agente toma el elixir cuando es más "atrevido" y logra que se satisfagan varios *drives* a la vez. Esto hace que, como se ha mostrado al comienzo, el valor medio del bienestar sea muy alto aunque existan las caídas ocasionales debidas al envenenamiento. Al final, cuando el agente tiene miedo al elixir, prescinde de él y sigue la política de comportamiento "sin riesgo", es decir, que cuando el agente tiene hambre toma la comida y cuando tiene sed, el agua. Esta política es igual de buena que la que se sigue cuando el agente no tiene miedo del elixir, aunque ahora se satisfacen los *drives* de uno en uno. Esto provoca la leve disminución del valor medio del bienestar pero sin descensos bruscos, lo que mejora la calidad de vida del agente.

10.3. Agente con miedo a estar en estados peligrosos

10.3.1. Descripción del experimento

En esta segunda parte del capítulo, se va a implementar el miedo como una motivación más del agente. El hecho de añadir el miedo como una motivación más va a hacer que se introduzca un nuevo estado interno del agente: "atemorizado". De esta manera, el agente, tendrá que aprender lo que tiene que hacer cuando tiene miedo. Tal y como se presentó en la sección 6.6.2, cuando el agente puede sufrir algunos efectos negativos en un estado como consecuencia de eventos exógenos, éste siente miedo. Se considerará que un estado es un estado peligroso cuando:

$$\min_{a_2} Q^{obj_i}_{peor}(s, Nada, a_2) < L_{miedo} \tag{10.8}$$

en el caso contrario, es un estado seguro. El límite L_{miedo} se estableció como:

$$L_{miedo} = -4 \tag{10.9}$$

De manera que:

$$\begin{aligned} &\text{Si el agente está en un estado seguro entonces } D^{k+1}_{miedo} = 0 \\ &\text{Si el agente está en un estado peligroso entonces } D^{k+1}_{miedo} = 5 \end{aligned} \tag{10.10}$$

Tal y como se expresa en la ecuación (10.8), el agente está en un estado peligroso y por lo tanto tiene miedo, cuando lo peor que le ha pasado interaccionando con alguien sin él hacer nada, es menor que un cierto límite. Por este motivo, como se expuso en la sección 6.6.2, en este experimento se van a almacenar los peores valores Q registrados durante la interacción con otros agentes:

$$Q^{obj_i}_{peor}(s, a_1, a_2) = \min(Q^{obj_i}_{peor}(s, a_1, a_2), r + \gamma \cdot V^{obj_i}_{peor}(s')) \tag{10.11}$$

Donde:

$$V^{obj_i}_{peor}(s') = \max_{a \in A_{obj_i}} (Q^{obj_i}_{peor}(s', a_1, a_2)) \tag{10.12}$$

es el valor peor del objeto i en el nuevo estado. De nuevo, estos valores se calculan con independencia del estado interno del agente, lo que importa es lo peor que ha pasado con el objeto, que en este caso es un oponente.

Para realizar este experimento se hace convivir al agente con dos nuevos tipos de agentes:

- Un agente neutro, pero que a diferencia del descrito en el capítulo 9, va a elegir aleatoriamente sus acciones entre:

$$A_{neutro} = \begin{cases} Saludar \\ Robar\ comida/agua/medicina \\ Dar\ comida/agua/medicina \end{cases} \tag{10.13}$$

Por lo que no va a pegar, ni a esperar.

- Un agente casi-bueno, que se va a comportar de manera que:

  Un 95 % de las veces, elige sus acciones entre:

$$A_{casi-bueno_{95\,\%}} = \begin{cases} Saludar \\ Dar\ comida/agua/medicina \end{cases} \tag{10.14}$$

  Mientras que el otro 5 % de las veces:

$$A_{casi-bueno_{5\,\%}} = Pegar \tag{10.15}$$

De esta manera, sólo el agente "casi-bueno" es el que muy ocasionalmente pega al oponente. El castigo que recibe el agente cuando le pegan, se ha modificado en relación a lo definido en el capítulo 7. Ahora, cuando al agente le pegan:

$$D^{k+1}_{debilidad} = D^{k}_{debilidad} + 20 \tag{10.16}$$

Por lo tanto, se hace evidente que interaccionar con este agente "casi-bueno" puede ser peligroso. Tal y como se vio en el capítulo del agente acompañado, capítulo 9, uno de los principales problemas que se presentaban cuando el agente vivía con agentes con distinta "personalidad" es la falta de identificación de los oponentes. Es decir, al no identificar el agente a su oponente, trataba a todos por igual, por lo que su actuación dependía del número de veces que interaccionaba con el agente malo, con el bueno o con el neutro. En este experimento esto se ha cambiado. El agente va a identificar a su oponente y tiene matrices de interacción social distintas para cada agente.

En este experimento, los otros agentes tienen nombre: el agente neutro se llama "Aran" y el casi-bueno se llama "Pepe". Por lo tanto, en lugar de tener una única matriz de valores Q correspondientes a las acciones de interacción con otros agentes, el agente tiene dos matrices, una para Aran y otra para Pepe.

Para decidir el tipo de algoritmo de aprendizaje para la interacción multiagente, se recuerdan las conclusiones del capítulo anterior. En base a los resultados presentados en el capítulo del agente acompañado, el algoritmo más efectivo es el Q-learning. Por lo tanto, en lugar de tener matrices Q de interacción para cada agente, se tienen vectores. Esto hace que se modifique (10.8), de manera que un estado es peligroso si:

$$Q^{obj_i}_{peor}(s, Nada) < L_{miedo} \tag{10.17}$$

En este experimento va a ser importante el hecho de que el agente casi-bueno, Pepe, sólo pegue muy ocasionalmente. Este agente normalmente es muy bueno, pero de vez en cuando es muy malo. De esta forma, si el agente no tiene implementado el miedo, va a querer interaccionar con el casi-bueno, Pepe. Cuando el agente tiene Miedo como motivación, finalmente "huirá" de Pepe porque tiene miedo a estar junto a él. Lo interesante de este experimento es que el agente genera comportamientos de escape, sin haberlos programado previamente.

A continuación se van a presentar los resultados obtenidos con un agente conviviendo con Pepe y Aran sin el *drive* Miedo, y posteriormente con miedo. Cada experimento consta de nuevo de dos fases, una de aprendizaje que dura 25000 iteraciones y otra permanente que dura 5000.

10.3.2. Resultados con el agente sin miedo

Cuando el agente convive sin miedo con los agentes neutro, Aran y casi-bueno, Pepe, el bienestar del agente varía tal y como se muestra en la figura 10.9.

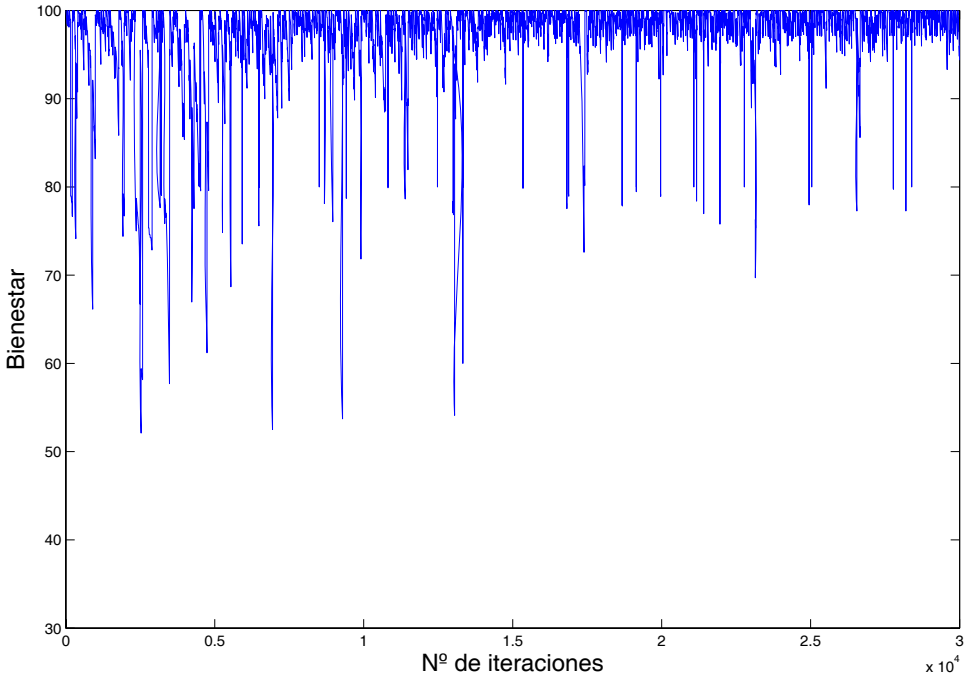


Fig. 10.9: *Bienestar del agente cuando no tiene miedo a estar en un estado peligroso*

Tal y como era de prever, el bienestar del agente durante la fase permanente, en la que vive con los valores Q ya aprendidos, tiene un valor medio muy alto. De hecho el valor medio vale 98,64 y el porcentaje de permanencia en la zona de seguridad es de 98,66 %. Es evidente que, a excepción de los descensos bruscos del bienestar en varias ocasiones, el bienestar permanece la mayoría del tiempo en la zona de seguridad.

Estos descensos, tal y como se muestra en la figura 10.10, son debidos a las veces que el agente fue pegado, y por lo tanto se produjeron los picos de magnitud 20 en el *drive* Debilidad. El agente interacciona con Pepe a pesar de que este puede, en algún momento, agredirle. De hecho en este experimento el número de veces que el agente interacciona con Aran, el agente neutro, es de 326 veces y con Pepe, el casi-bueno, 214 veces.

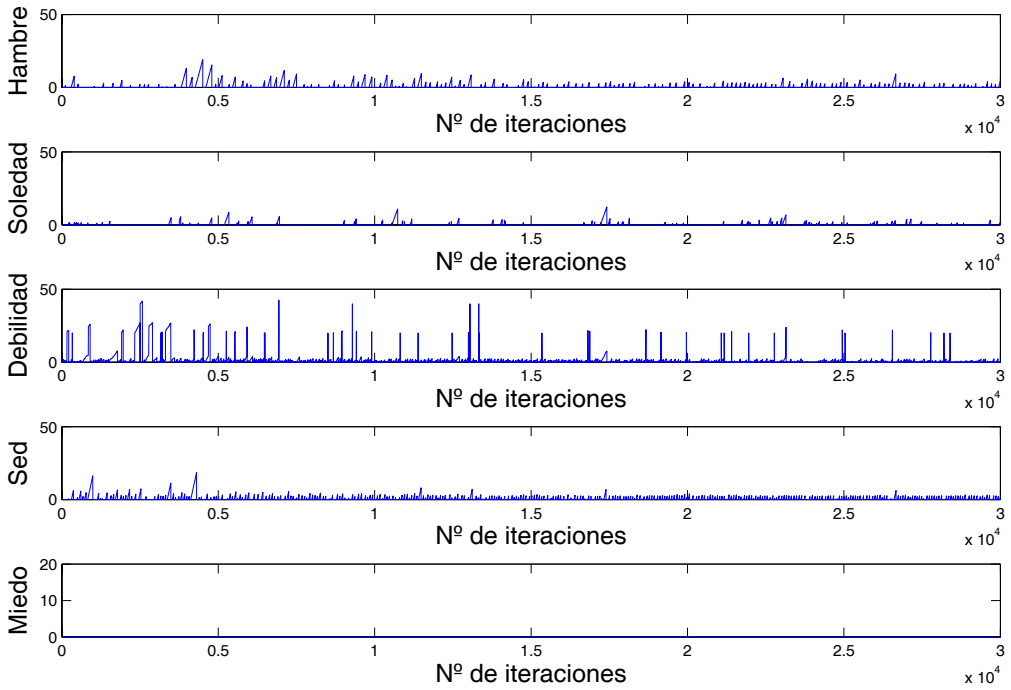
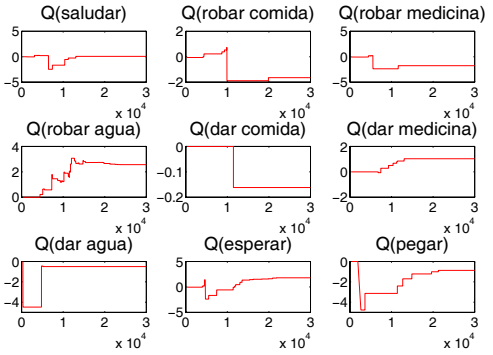


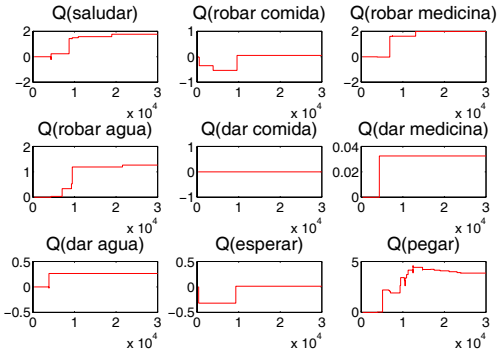
Fig. 10.10: Drives *del agente cuando no tiene miedo a estar en un estado peligroso*

A continuación se van a mostrar los valores Q obtenidos para cada una de las motivaciones dominantes, con el fin de justificar los resultados de este experimento. El hecho de que el número de veces que el agente interacciona tanto con Aran como con Pepe, se debe que para la mayoría de las motivaciones, los valores de interacción social son superiores a los de las acciones relacionadas con los objetos.

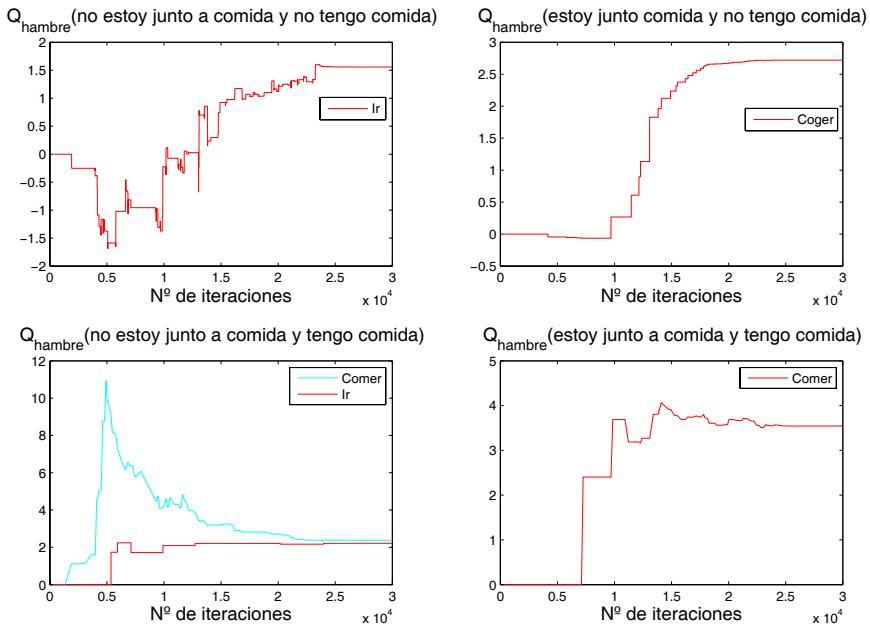
Para el caso de que el agente tiene **hambre**, en la figura 10.11(c) se muestran los valores Q de las acciones relacionadas con la comida. También se muestran los valores del vector Q de las acciones del agente cuando interacciona con Pepe, la figura 10.11(a), y con Aran, la figura 10.11(b). Como se muestra en las gráficas, los valores máximos de los vectores Q de interacción con Pepe, $Q(robar\ agua) = 2{,}56$, y Aran, $Q(pegar) = 3{,}84$, son superiores al valor que tiene "ir a por comida" cuando el agente tiene hambre, 1,55. Por lo tanto cuando el agente tiene hambre y está sólo, irá a por la comida, pero si se encuentra con Pepe o con Aran interaccionará con ellos.



(a) Valores Q de interacción con Pepe, agente casi-bueno



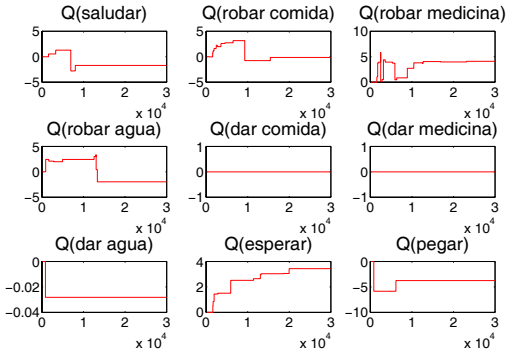
(b) Valores Q de interacción con Aran, agente neutro



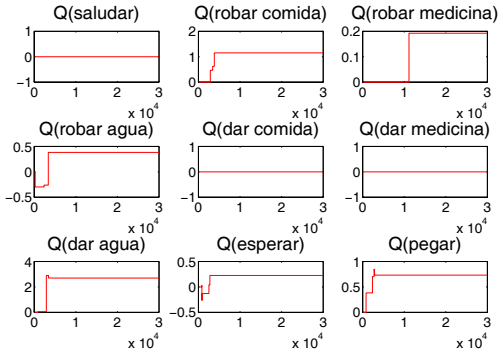
(c) Valores Q en relación a comida

Fig. 10.11: *Valores Q del agente cuando la motivación dominante es Hambre, sin miedo a estar en un estado peligroso*

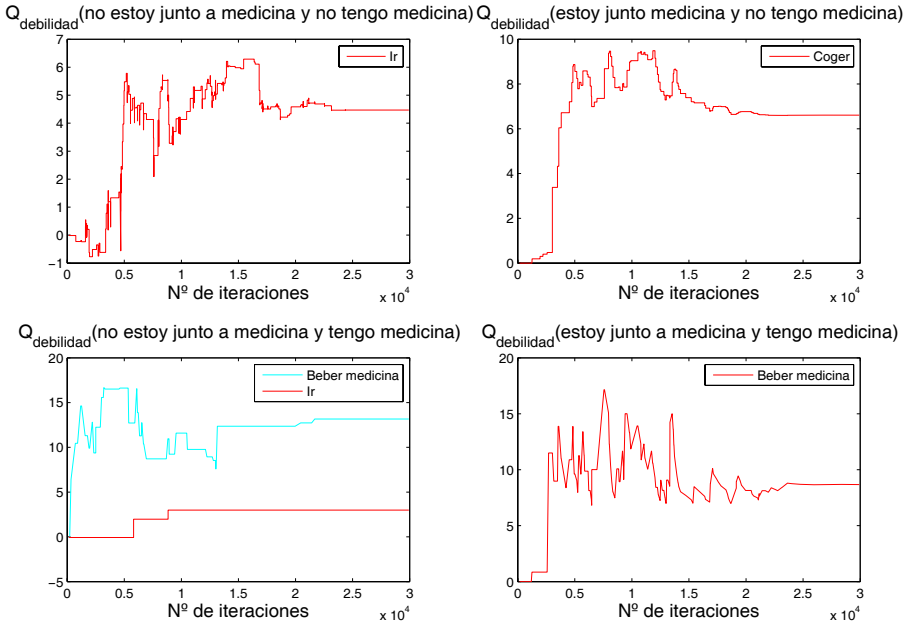
Sin embargo, otra cosa ocurre cuando el agente está **débil**. En la figura 10.12(c) se representan los valores Q de las acciones relacionadas con la medicina. Si se comparan estos valores con los valores máximos de los vectores Q de interacción con Pepe, en la figura 10.12(a), $Q(robar\ medicina) = 4{,}1$, y con Aran, en la figura 10.12(b), $Q(dar\ agua) = 2{,}7$, se puede apreciar que los valores relacionados con medicina son superiores. Por lo tanto, cuando el agente está débil, no va a interaccionar con ninguno de los dos agentes, sino que irá a por la medicina, la cogerá y se la beberá.



(a) Valores Q de interacción con Pepe, agente casi-bueno



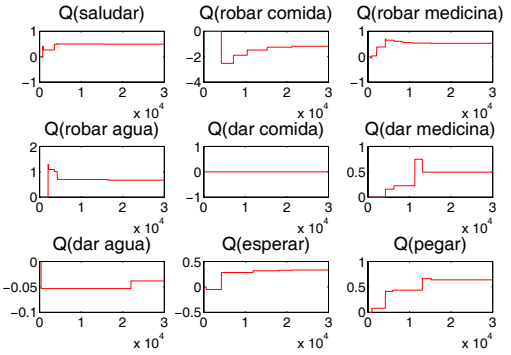
(b) Valores Q de interacción con Aran, agente neutro



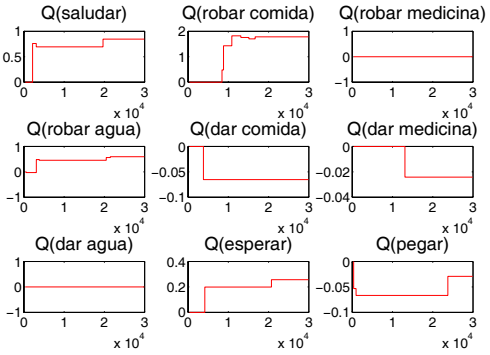
(c) Valores Q en relación a medicina

Fig. 10.12: *Valores Q del agente cuando la motivación dominante es Debilidad, sin miedo a estar en un estado peligroso*

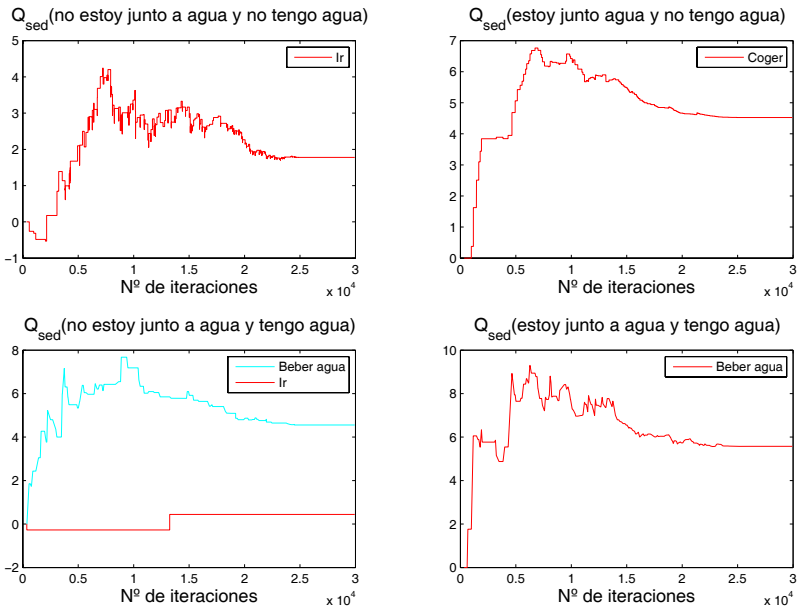
En el caso de que el agente tiene **sed**, en la figura 10.13(c) se muestran los valores de las acciones relacionadas con el agua. Los vectores Q de interacción con Pepe y con Aran se muestran en las figuras 10.13(a) y 10.13(b) respectivamente. En el caso de la interacción con Pepe, el valor máximo es $Q(robar\ agua) = 0{,}67$ y para Aran, $Q(robar\ comida) = 1{,}78$. De nuevo, los valores máximos Q de interacción social son menores que los valores Q de las acciones relacionadas con el agua. Como consecuencia, cuando el agente tiene sed no quiere interaccionar con nadie.



(a) Valores Q de interacción con Pepe, agente casi-bueno



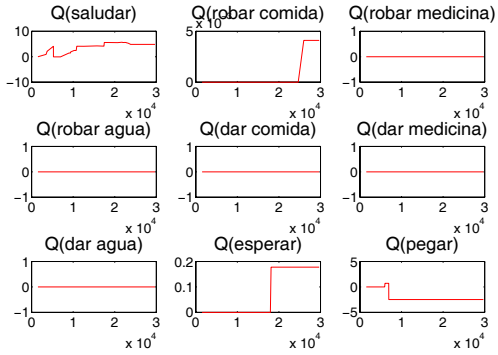
(b) Valores Q de interacción con Aran, agente neutro



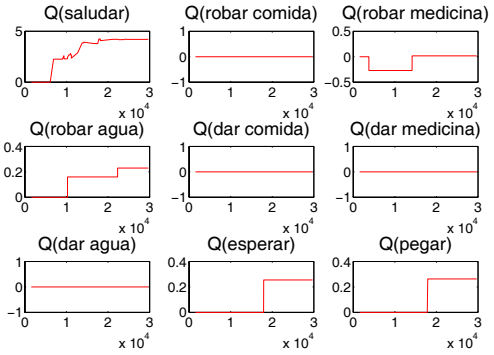
(c) Valores Q en relación a agua

Fig. 10.13: *Valores Q del agente cuando la motivación dominante es Sed, sin miedo a estar en un estado peligroso*

Cuando la motivación dominante es **Soledad**, los valores máximos de los vectores Q de interacción social para Pepe, el agente casi-bueno, $Q(saludar) = 4{,}85$, y para Aran, agente neutro, $Q(saludar) = 4{,}2$, son en general altos. Estos valores son superiores a los valores Q de las acciones relacionadas con el resto de los objetos, con valores máximos en torno a 1,2. Por lo tanto, en esta ocasión el agente querrá interaccionar con cualquiera de los dos agentes. En las figuras 10.14(a) y 10.14(b), se muestran los valores de interacción con Pepe y Aran respectivamente.



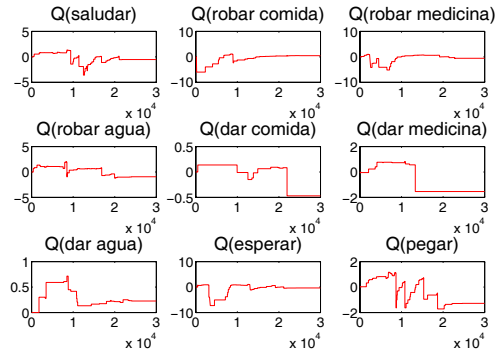
(a) Valores Q de interacción con Pepe, agente casi-bueno



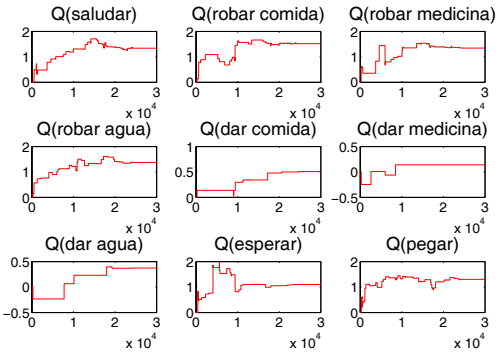
(b) Valores Q de interacción con Aran, agente neutro

Fig. 10.14: *Valores Q cuando la motivación dominante es Soledad, sin miedo a estar en un estado peligroso*

Por último, cuando **no hay una motivación dominante**, los valores de las acciones relacionadas con los objetos son de nuevo muy bajos, con valores máximos alrededor de 0,2. Sin embargo, los valores máximos de los vectores Q de interacción social, tanto de Pepe como de Aran son superiores. En el caso del agente casi-bueno, Pepe, ver la figura 10.15(a), el valor máximo, $Q(robar\ comida) = 0{,}48$, no es mucho mayor. Sin embargo con el agente neutro, Aran, ver la figura 10.15(b), el máximo, $Q(robar\ comida) = 1{,}52$, es bastante más alto.



(a) Valores Q de interacción con Pepe, agente casi-bueno



(b) Valores Q de interacción con Aran, agente neutro

Fig. 10.15: *Valores Q cuando no hay motivación dominante, sin miedo a estar en un estado peligroso*

Como **resumen**, el agente va a querer interaccionar con ambos agentes cuando tiene hambre, cuando necesite de interacción social y cuando no tenga ninguna motivación dominante. En el caso de que el agente esté débil o sediento, el agente no querrá interaccionar con ninguno de los dos. Es lógico, por lo tanto, que el número de interacciones con los agentes durante la fase permanente sea tan alto para ambos oponentes.

Es destacable que, a pesar de que el agente casi-bueno, Pepe, a veces es muy malo para el agente, los valores en general del vector de interacción Q son buenos. Esto es debido, como ya se ha explicado, a que el agente utiliza el algoritmo Q-learning para calcular el valor de sus acciones. El resultado es que el valor de cada acción en la interacción social, pondera las veces que salieron bien y las veces que salieron mal. Cuando el agente interacciona con Pepe, las veces que este agente se porta bien son muchas más que las veces que le pegó, por lo que finalmente los buenos resultados pesan más que los malos. Es decir, tiene sentido que los valores correspondientes a la interacción con Pepe no sean malos.

Para el caso de la interacción con Aran, el agente neutro, los valores obtenidos son similares a los obtenidos en el capítulo anterior cuando el agente vivía en un mundo neutro utilizando el algoritmo Q-learning.

10.3.3. Resultados con el agente con miedo

Cuando el agente tiene miedo a estar en un estado peligroso mientras convive con los agentes casi-bueno y neutro, Pepe y Aran, su bienestar se ve afectado por el *drive* Miedo, ver la figura 10.16.

Como se puede apreciar en la gráfica, el bienestar del agente durante la fase permanente permanece acotado en su mayor parte dentro de la zona de seguridad durante un 94,4 %. De hecho el otro indicador de análisis de resultados, el valor medio, vale 97,69. Si se comparan estos resultados con los obtenidos cuando el agente no tiene el *drive* Miedo, se puede apreciar que el valor medio es un poco inferior, valía 98,64, y el porcentaje de permanencia en la zona de seguridad es prácticamente 4 puntos más bajo, valía 98,66 %.

Este descenso es debido a varios factores: el primero es que, tal y como se verá posteriormente, si el agente tiene miedo de interaccionar con algún agente pero necesita de esa interacción, pasará un tiempo hasta que se satisfaga el *drive* correspondiente. Segundo, cuando el agente se encuentra en presencia de otro agente, éste comprueba, utilizando la ecuación (10.17), lo peor que le ha pasado con éste agente mientras que él no hacía nada. Si resulta que esto es menor que un límite, el agente tendrá miedo, es decir, el *drive* Miedo aumenta. Esto quiere decir que cada vez que el agente se encuentra con el agente al que tiene miedo, se va a producir un descenso de su bienestar. Partiendo de la ecuación (10.10), se puede determinar que esta variación del bienestar es negativa, de magnitud 5.

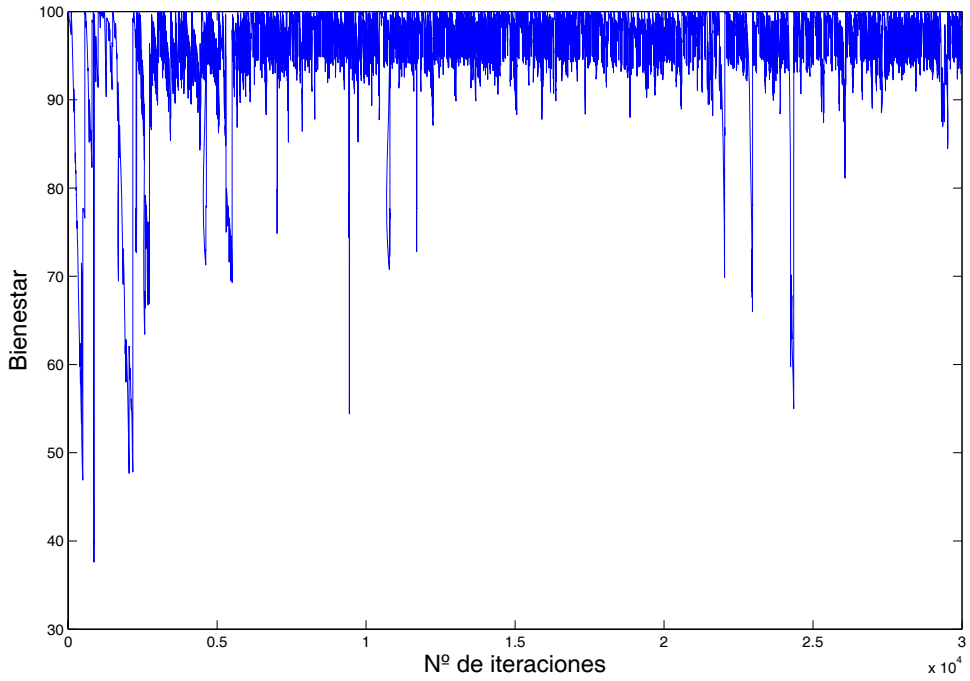


Fig. 10.16: *Bienestar del agente cuando tiene miedo a estar en un estado peligroso*

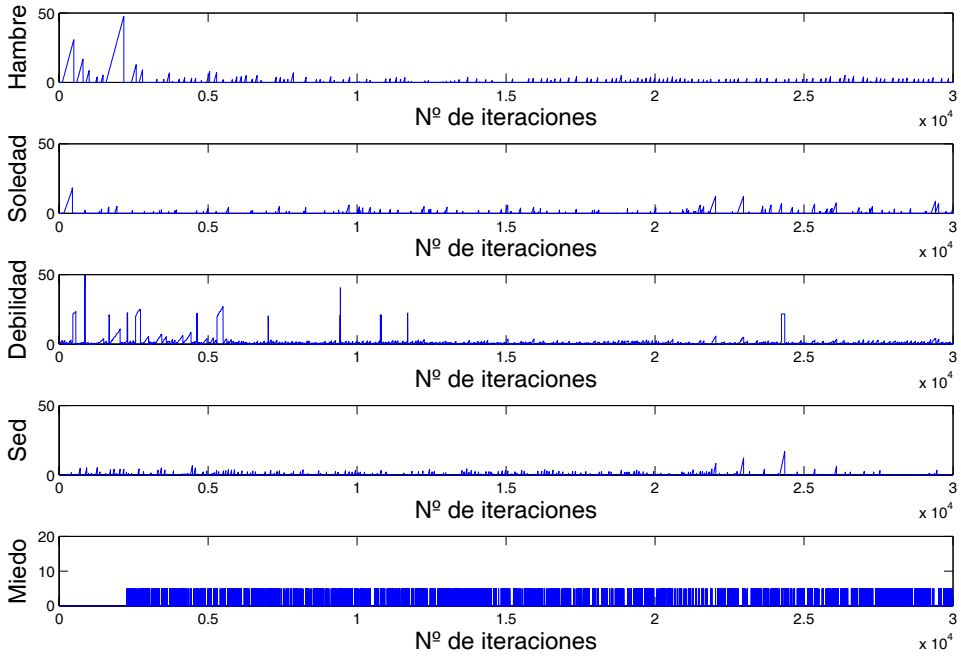


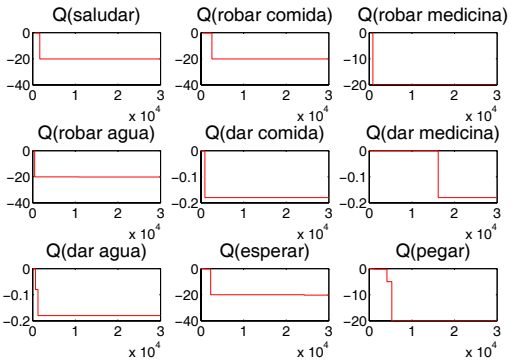
Fig. 10.17: Drives *del agente cuando tiene miedo a estar en un estado peligroso*

Por otro lado, a partir de la evolución de los *drives*, ver la figura 10.17, también se pueden sacar otras conclusiones. Lo primero que se puede observar es que el miedo comienza a aparecer a partir de la iteración 2200 aproximadamente. Esto es después de que el agente haya interaccionado con Pepe, el casi-bueno, y éste le haya pegado varias veces.

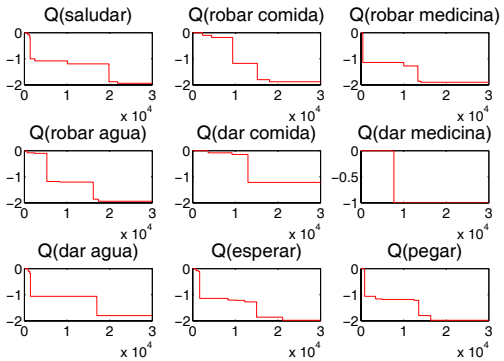
Si se observa la evolución del *drive* Debilidad, se puede ver cómo el número de picos, debidos a que le peguen, disminuye radicalmente a partir de la mitad de la fase de aprendizaje. Por lo tanto parece que el agente aprende a no interaccionar con el agente que le puede hacer daño, tiene miedo de estar junto a Pepe. De hecho, durante la fase permanente, el número de veces que el agente interaccionó con Pepe, el casi-bueno, fue sólo 4, mientras que con Aran, el neutro, fueron 376 veces. Por otro lado, el *drive* Soledad presenta picos en la última parte de la fase de aprendizaje y durante la fase permanente. Previsiblemente esto es debido a que como no quiere interaccionar con Pepe, tarda más tiempo en satisfacer este *drive*.

Todas estas conclusiones van a ser detalladas a continuación utilizando los valores Q aprendidos en cada motivación. La principal diferencia ahora será que existe una nueva motivación: Miedo. Ya se ha dicho que el agente tiene miedo cuando lo peor que le ha pasado al estar junto a otro agente, mientras él no hacía nada, es muy malo.

Para comenzar con este análisis se van a mostrar los vectores Q_{peor}, que almacenan los peores valores correspondientes a la interacción con Pepe y con Aran. En la figura 10.18(a) se puede ver que el peor valor Q registrado cuando el agente interacciona con Pepe, el casi-bueno, y él no hace nada, esto es, "espera", es -20, por lo que es menor que el límite -4. En cambio para Aran, el neutro, este valor es -2, mayor que -4, ver la figura 10.18(b). Por lo tanto, el agente tiene miedo cuando está junto a Pepe, tal y como era de esperar.



(a) Valores Q_{peor} de interacción con Pepe, agente casi-bueno



(b) Valores Q_{peor} de interacción con Aran, agente neutro

Fig. 10.18: *Valores Q_{peor}*

Cuando el agente tiene **miedo**, es decir, cuando la motivación dominante del agente es Miedo, los valores Q de las acciones en relación a los objetos son máximos para las acciones de movimiento. En concreto el valor Q máximo corresponde a la acción de "ir a por comida" cuando el agente no está junto a comida y tiene comida, que vale 2, tal y como se muestra en la figura 10.19. Por lo tanto, cuando el agente está junto a Pepe, tiene miedo de él y tendrá que decidir la acción a realizar. Puede escoger entre interaccionar, ver la figura 10.20, o irse. Como se puede observar en la figura 10.20, el valor máximo del vector de interacción social con Pepe, que es casi-bueno, es bastante más pequeño que el valor que tiene irse, $Q(dar\ comida) = 0{,}28$. Por lo que cuando el agente cuando está junto a Pepe, prefiere irse antes que quedarse a su lado: el agente "huye" de esa situación.

Este hecho es muy importante, el agente aprende que cuando tiene miedo, lo mejor es escapar. Esta acción de escape no es ninguna acción programada *a priori*, simplemente el agente valora de forma muy positiva las acciones de movimiento. Esto es razonable, ya que si tiene miedo y se mueve a otra habitación donde está solo, ya no tiene miedo. Esta acción de movimiento hace que el *drive* Miedo pase de valer 5 a valer 0, lo que implica que esa acción tiene un refuerzo de 5. Por lo tanto, cada vez que el agente se encuentra con Pepe, lo que hace es moverse en la dirección hacia la comida. Una vez que ya no tiene miedo, vuelve a elegir la acción a realizar.

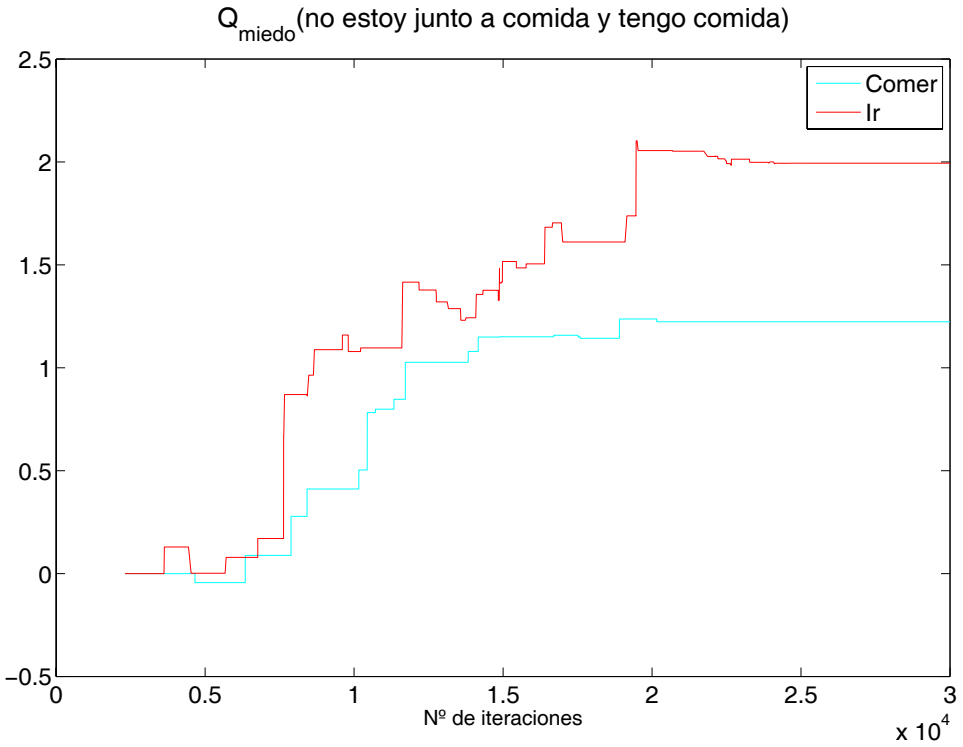


Fig. 10.19: *Valor Q de "ir a por comida" cuando cuando la motivación dominante es Miedo*

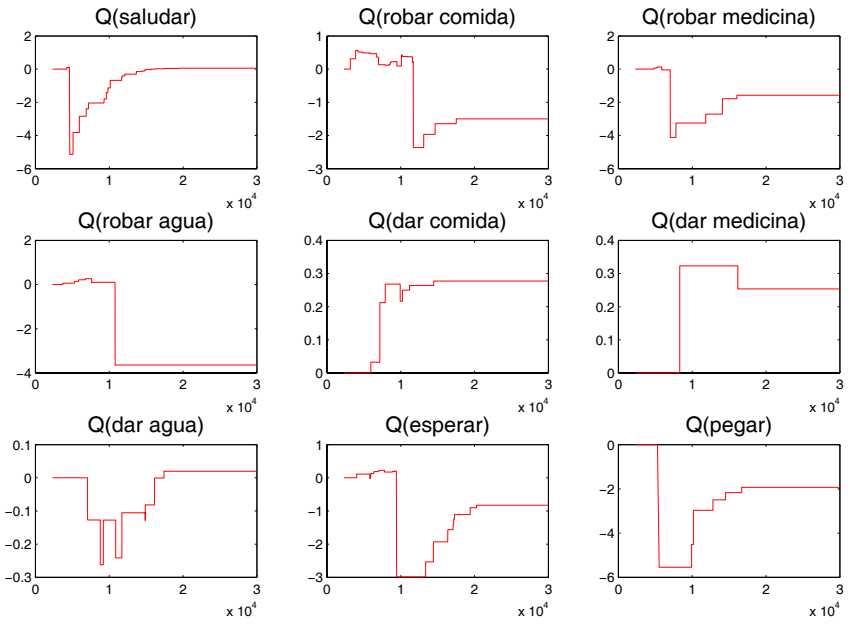
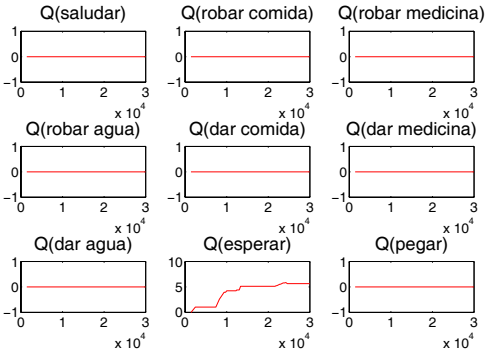


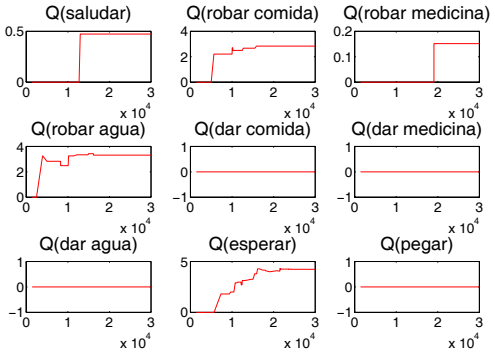
Fig. 10.20: *Valor del vector Q de interacción con Pepe, agente casi-bueno, cuando la motivación dominante es Miedo*

En el caso de que el agente necesite **interacción social**, el valor Q máximo en relación a los objetos vuelve a corresponder a la acción "ir a por comida", que vale 0,06, ver la figura 10.21(c). Los valores de los vectores de interacción social con Pepe y Aran se muestran en la figura 10.21(a) y la figura 10.21(b) respectivamente. Tal y como se aprecia en las gráficas, el valor que tiene interaccionar con Aran cuando necesita de interacción social, es alto, $Q(esperar) = 4{,}22$. Por lo que cuando el agente se lo encuentre, querrá interaccionar con él. Sin embargo, lo que puede parecer curioso es que el valor máximo del vector de interacción con Pepe, $Q(esperar) = 5{,}64$, sea incluso mayor que el de Aran, que es neutro. Lo que sucede es que cuando el agente necesita interacción social, pero sin mucha "urgencia", es decir, el *drive* Soledad es el máximo pero no es muy alto, cuando se encuentra con Aran interacciona. Por el contrario, si se encuentra con Pepe, su *drive* Miedo valdría 5 y puede ocurrir que pase a ser Miedo la motivación dominante, por lo tanto "huirá" de él y el *drive* Soledad seguirá creciendo. Sólo cuando la necesidad de interacción social supere al miedo, el agente interaccionará con Pepe. Esta es la explicación de los picos que aparecen en el *drive* Soledad. El agente no puede controlar a quién se va a encontrar en el camino, por lo que si no se encuentra a Aran, el *drive* Soledad seguirá aumentando hasta que se venza el miedo a interaccionar con Pepe.

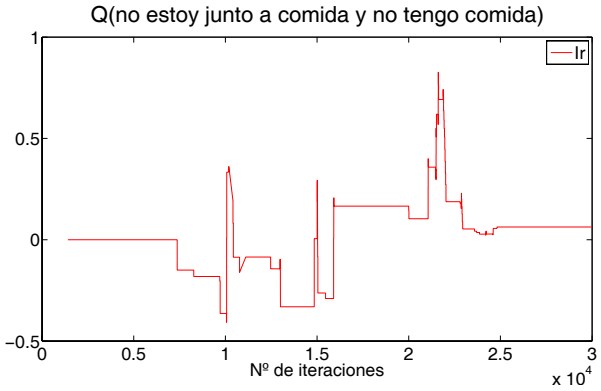
Por otro lado, cuando el agente tiene **hambre**, el valor de "ir a por la comida", ver la figura 10.22(c), es 1,74. Este valor es el máximo entre los valores Q del resto de las acciones de "ir a" relacionadas con todos los objetos. Por lo tanto, cuando el agente tiene hambre, va a por comida. Si en el camino se encuentra a Aran, agente neutro, a partir de la figura 10.22(b), se deduce que sí que va a querer interaccionar,



(a) Valores Q de interacción con Pepe, agente casi-bueno



(b) Valores Q de interacción con Aran, agente neutro

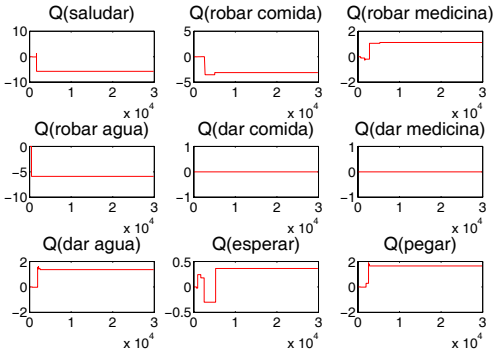


(c) Valor Q de "ir a por comida"

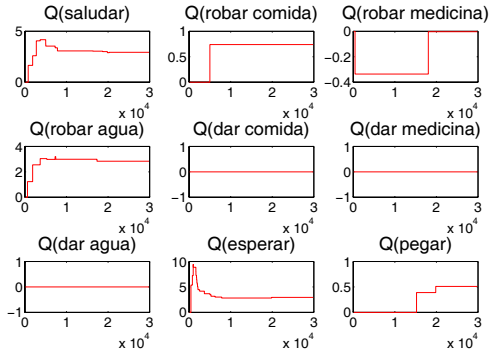
Fig. 10.21: *Valores Q del agente cuando la motivación dominante es Soledad*

ya que el valor máximo del vector Q es $Q(esperar) = 2{,}95$. Sin embargo, cuando se encuentra con Pepe, agente casi-bueno, el valor Q máximo de su vector de interacción es $Q(pegar) = 1{,}65$ menor que el valor de "ir a por comida", ver 10.22(a). Por lo que en esta ocasión, aunque el hambre sea mayor que el miedo, no va a interaccionar con Pepe.

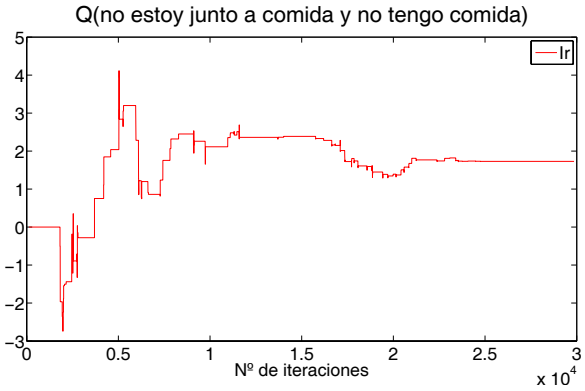
Cuando el agente está **débil**, sucede que, tal y como se muestra en la figura 10.23(c), el valor de "ir a por medicina" es 2,22, el valor Q máximo de interaccionar con Aran es $Q(pegar) = 2{,}35$, ver la figura 10.23(b) y con Pepe es $Q(pegar) = 6{,}1$, ver la figura 10.23(a). Como consecuencia, cuando el agente está débil va a por medicina y si se encuentra con Aran, el neutro, interaccionará con él. Si por el contrario, se encuentra con el agente casi-bueno, Pepe, ocurrirá lo mismo que cuando necesita interacción social. Si su debilidad no es muy alta, la *motivación* Miedo pasa a ser la motivación dominante y huirá de Pepe. Sólo si su debilidad supera al miedo, interaccionará con Pepe arriesgándose a que le pegue.



(a) Valores Q de interacción con Pepe, agente casi-bueno



(b) Valores Q de interacción con Aran, agente neutro

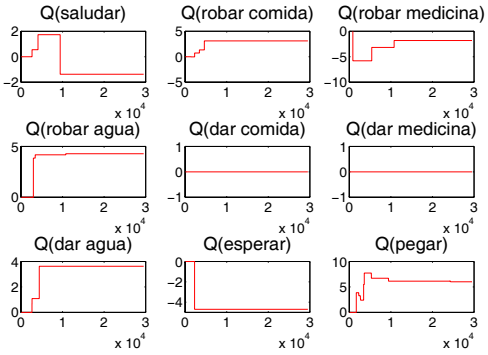


(c) Valor Q de "ir a por comida"

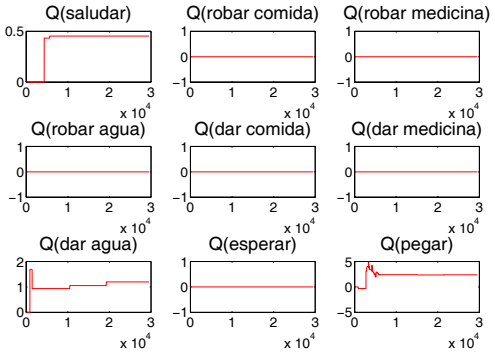
Fig. 10.22: *Valores Q del agente cuando la motivación dominante es Hambre*

En el caso de que **Sed** sea la motivación dominante, la situación cambia. El agente valora de forma más positiva, "explorar" que "ir a por agua" cuando tiene sed, ver la figura 10.24(a) y la figura 10.24(b). Si mientras está explorando se encuentra con Pepe, el agente casi-bueno, o con Aran, el agente neutro, tanto la figura 10.25(a) como la figura 10.25(b), muestran que el agente no interacciona con ninguno. Esto se debe a que los valores máximos de los vectores Q de interacción con Pepe y con Aran son mucho más pequeños que el valor de "explorar". Es interesante observar que los valores de interacción con Pepe, sólo tienen cambios en las primeras iteraciones, después su valor es constante. Esto indica que a partir de que en el agente aparece el miedo, en la iteración 2200 aproximadamente, la sed no es nunca superior al miedo de estar junto al agente casi-bueno, Pepe.

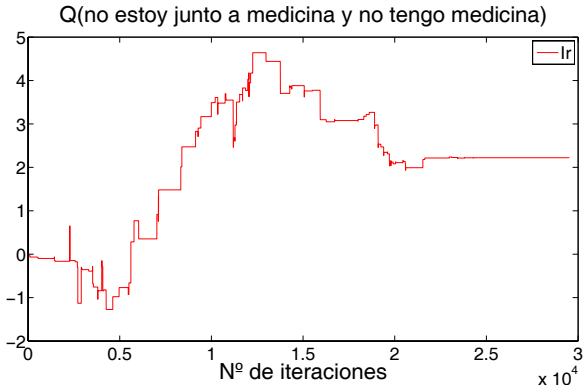
Por último, si el agente **no tiene ninguna motivación dominante**, lo que ocurre es que si está solo, los valores Q máximos corresponden a las acciones relacionadas con el agua, siendo todos inferiores a 0,5, ver la figura 10.26. Por lo que irá a por agua, la cogerá y se la beberá. Por ese motivo, durante la fase permanente el *drive*



(a) Valores Q de interacción con Pepe, agente casi-bueno

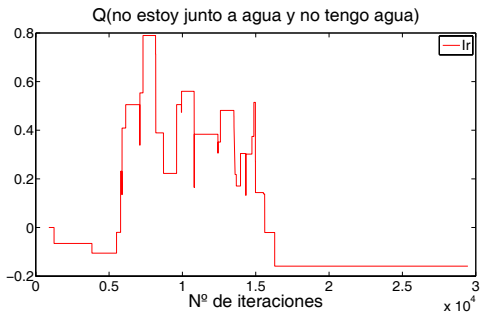


(b) Valores Q de interacción con Aran, agente neutro

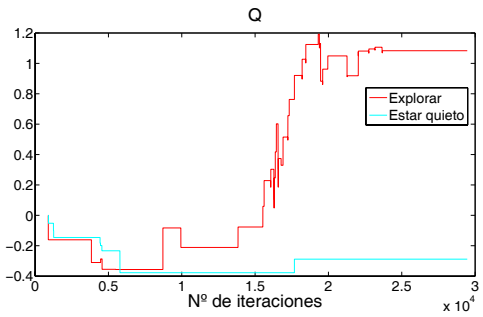


(c) Valor Q de "ir a por medicina"

Fig. 10.23: *Valores Q del agente cuando la motivación dominante es Debilidad*



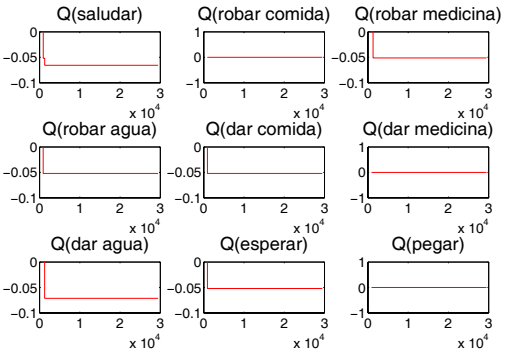
(a) Valor Q de "ir a por agua"



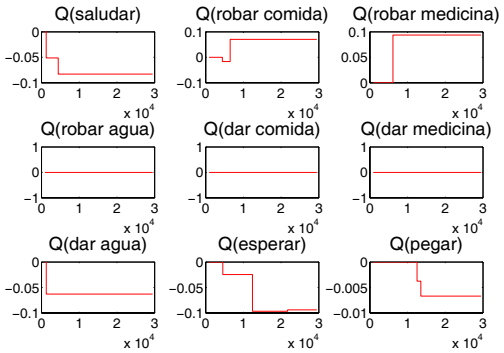
(b) Valor Q de "explorar"

Fig. 10.24: *Valores Q del agente cuando la motivación dominante es Sed*

Sed permanece satisfecho la mayor parte del tiempo, ver figura 10.17. Si el agente se encontrara acompañado por el agente casi-bueno, Pepe, Miedo pasaría a ser la motivación dominante, ya que el resto de los *drives* son inferiores a 2 y se iría de su lado. Si en cambio, es el agente neutro, Aran, el que le acompaña, se puede ver en la figura 10.27 que querrá interaccionar con él, ya que el valor máximo del vector de interacción es $Q(robar\ medicina) = 1{,}18$.



(a) Valores Q de interacción con Pepe, agente casi-bueno



(b) Valores Q de interacción con Aran, agente neutro

Fig. 10.25: *Valores Q de interacción del agente cuando la motivación dominante es Sed*

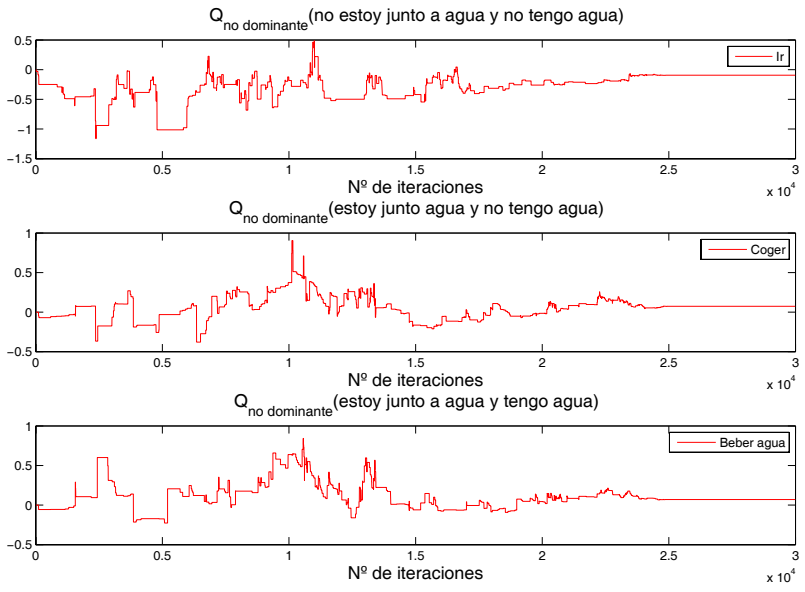


Fig. 10.26: *Valores Q de las acciones relacionadas con el agua cuando no hay motivación dominante*

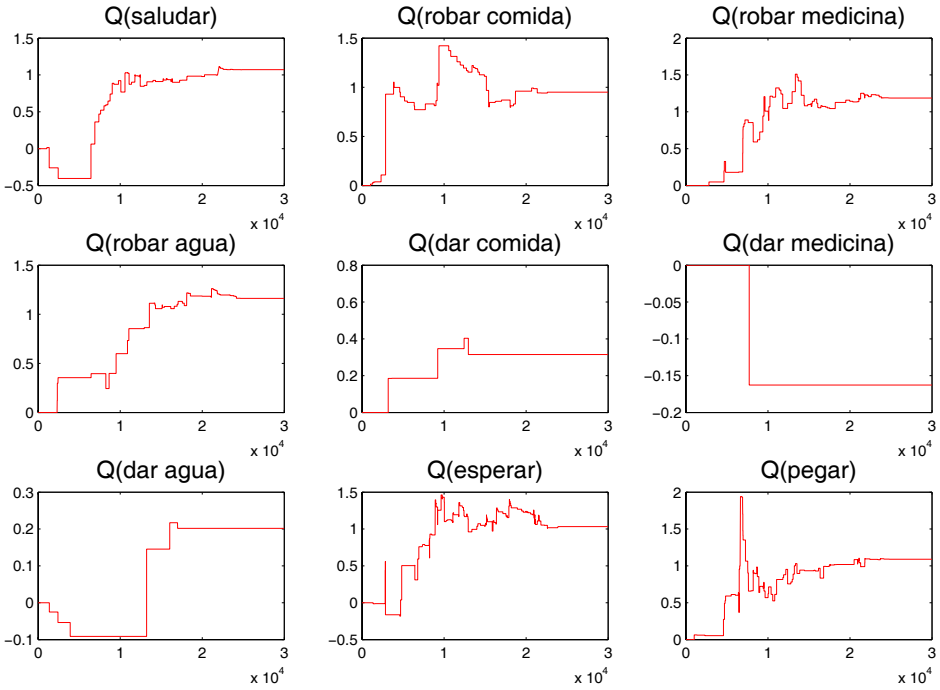


Fig. 10.27: *Valor del vector Q de interacción con Aran, agente neutro, cuando no hay motivación dominante*

10.4. Resumen y conclusiones

En la primera parte de este capítulo, se muestra la utilidad del miedo para evitar realizar acciones que ocasionalmente son nocivas. Para ello se creó un nuevo objeto, el elixir, de forma que el 95 % de las veces hace que se satisfagan los *drives* Hambre, Debilidad y Sed. El otro 5 % de las veces provoca que estos mismos *drives* aumenten 4 unidades cada uno, de manera que existe un refuerzo negativo de -12.

Para poder implementar este miedo, se almacenan los peores valores experimentados por el agente, para cada par estado-acción y para cada objeto, con independencia del estado interno. Finalmente el agente, en lugar de escoger la acción que maximiza los valores Q, dados por el Q-learning, va a escoger la que maximiza una combinación de estos valores Q y los peores valores Q, dada por la ecuación (10.6).

El factor de atrevimiento β es el que va a determinar la importancia de lo peor que le ha pasado al agente frente a los valores Q, obtenidos con el Q-learning. En el experimento presentado, durante la fase de aprendizaje el agente va almacenando lo peor que le ha pasado. En la fase permanente se va disminuyendo el valor del factor de atrevimiento, de manera que el agente da más importancia a lo peor que le ha pasado. Los resultados muestran que cuando el factor de atrevimiento es máximo, es decir, el agente no considera lo peor que le ha pasado, el agente bebe varias veces el elixir, a pesar de sus posibles efectos negativos. A medida que el factor de atrevimiento

disminuye, el agente va dando más importancia a lo peor que le ha pasado. Por lo tanto, el agente tiene miedo del elixir y va dejando de beberlo hasta que finalmente no lo vuelve a coger.

En la segunda sección de este capítulo se han presentado los resultados obtenidos para un agente que convive con dos tipos de oponentes: uno casi-bueno, llamado Pepe, y otro neutro, distinto al descrito en el capítulo 9, llamado Aran.

El agente casi-bueno, un 95 % de las veces que interacciona con otro agente es bueno, es decir, que le saluda o le da objetos. El otro 5 % de las veces, pega. El agente neutro cada vez que interacciona puede hacer tanto cosas buenas (saludar y dar cosas) como malas (robar cosas) con la misma probabilidad, excepto pegar. El castigo recibido cuando pegan al agente es de -20, por lo que es muy nocivo para el bienestar.

El algoritmo utilizado para el aprendizaje durante la interacción con otros agentes es el Q-learning, ya que permite ponderar los castigos y recompensas según el número de veces que han sido recibidos.

Cuando el agente vive en este mundo sin miedo, se ha observado que de media vive mejor, es decir, el valor medio de su bienestar es alto. Sin embargo, al no tener miedo al agente casi-bueno, interacciona varias veces con él y como consecuencia éste le llega a pegar varias veces, recibiendo el castigo correspondiente. En este experimento se observa que el agente cuando tiene hambre, cuando necesita de interacción social o cuando no hay ninguna motivación dominante, prefiere interaccionar con los otros agentes.

Cuando el agente considera lo peor que le ha pasado con otro agente cuando él no hacía nada, aparece el miedo como motivación. En este experimento, después de haber interaccionado con Pepe varias veces y que éste le haya pegado, el valor de lo peor que le ha pasado con él mientras "esperaba", es menor que el límite que hace que el miedo aparezca. Por lo tanto, el agente tiene miedo de estar junto a Pepe, de manera que llega a aprender que cuando tiene miedo, lo mejor es escapar. Es muy importante este hecho: el agente aprende comportamientos de escape sin haber sido programados previamente. Sólo cuando en algunas ocasiones la necesidad social o la debilidad superan al miedo, el agente querrá interaccionar con Pepe, el agente casi-bueno.

En resumen, cuando el agente tiene Miedo como un *drive* más, el valor medio del bienestar y el porcentaje de permanencia en la zona de seguridad son un poco inferiores que cuando no tenía miedo. Esto es debido al propio miedo, sin embargo consigue que el agente casi-bueno no le vuelva a pegar, por lo que desaparecen las descensos bruscos en el bienestar y se considera que el agente vive mejor.

11. CONCLUSIONS AND FUTURE WORKS

11.1. Summary of results

The results of the experiments shown in chapter 8, carried out when the agent lives alone in the virtual world, showed that the agent is able to learn a correct policy of behaviour by his own. The agent uses a modification of the Q-learning algorithm to learn the correct function between states and actions. Using the variation of the wellbeing of the agent, happiness and sadness, as the reinforcement function, the agent learns to survive maintaining all his drives in acceptable ranges.

It must be said that in order to achieve these results, some parameters involved in the learning algorithm and the behaviour selection strategy, had to be tuned. These parameters are the following:

- The parameter δ that defines the relation between exploration and exploitation of the actions.
- The learning rate α which controls how much weigh is given to the reward just experienced.
- The discounted factor γ that defines how much expected future rewards affect decisions now.

It has been proved that, in order to learn a good policy, the agent has to explore all the possible actions. This implies that the parameter δ must be high. In relation to the learning rate, it was proved that the agent learns correctly with an intermediate-low value of α. One of the main conclusions is that in order to have a good life, it is necessary to stop learning, $\alpha = 0$. For this reason, to take advantage of the knowledge acquired, it was decided to separate the life of the agent in two phases: the learning phase and the steady phase.

During the learning phase the values of both parameters, δ and α, are decreased gradually. This implies that the agent, at the beginning of this phase, explores all the actions and learns from his experience. As the agents lives, he starts to exploit the actions that led to good results, as well as to give less importance to the new experiences, i.e. the agent begins to be more conservative. During the steady phase the agent stops learning and lives according to the policy learned. Moreover, it has

been also proved that, for learning a correct policy, the discounted factor γ must be high. It is necessary, when evaluating an action taken in a certain state, to consider the future rewards.

In chapter 9 the agent lives with other agents using the values of the parameters previously tuned. In relation to the need of social interaction, a new drive was implemented: Loneliness. This drive was implemented such that the agent satisfied it by interacting with another agent, therefore, the agent needs to interact with others in order to survive. In order to solve the problem about the reward due to the joint action when interacting with other agent, the agent uses several multi-agent reinforcement learning algorithms for learning the right policy of behaviour. These multi-agent algorithms are the Friend-Q, the Foe-Q and the Average-Q. Besides, the agent also tries the new algorithm based on Q-learning proposed in this thesis.

There are three other agents living in this world, and they live according to a fixed policy, so they are not learning. According to their policy of interaction with other agents, each agent has a certain personality: good, neutral and bad. Depending on the personality of each of these agents present in the world, four worlds are defined: good world, neutral world, mixed world and bad world. The agent uses every algorithm in each world trying to survive. Several conclusions will be extracted from these experiments as follows.

The main conclusion in relation to the use of the Friend-Q learning is that this algorithm makes the agent to be too optimistic. This means that the agent will always think that his opponent is going to be good to him, and therefore every time that he meets others, he will want to interact with them. In the case that the agent lives in a good world, where all the opponents are good, the performance of the agent is quite good since all the interactions have positive results. However, when the agent lives in the other worlds thinking that everyone is his friend, this will cause that the results of some interactions may damage him, since sometimes he will interact with a bad or a neutral agent. Besides, the fact that the agent wants to interact all the time causes that, many times, other drives are ignored.

On the other hand, when the agent uses the Foe-Q algorithm, he assumes that the other agents are his enemies. He will think that his opponent will select the worst action for him. When the agent lives in a good world, since the worst action for him is still a "good" one, the agent will interact with the opponents several times. On the contrary, in the other worlds where the other agents can hurt him, the agent, at the end, decides not to interact with anyone. This will cause that the Loneliness drive is never satisfied. The main conclusion in relation to this algorithm is that it is not very useful in a "real" life where, for sure, not everyone is a good opponent.

In the case that the agent uses the Average-Q as the learning algorithm, he will not only consider the best or the worst thing that happened to him while interacting with other agent. The goodness of this algorithm is that the Q value of each action is considered as the average value of all the Q values obtained for that action, while he was interacting with other agents. This will cause that when the agent lives in a

neutral or mixed world, he will adjust the number of social interactions. As it was proved, the agent will only want to interact when he needs it, i.e. when the dominant motivation is Loneliness. When the agent lives in a bad world, despite of the high probability of something bad happens, the fact that sometimes the other agents treat him right makes the average value increase. In this case, again, the agent will want to interact when he needs social interaction.

Finally, when the agent uses the new algorithm based on Q-learning, he ignores the fact that the reward received, while he is interacting, is due to their joint action. Therefore, the value of his actions, when interacting, will weigh the positive and the negative results experienced. In a neutral and mixed world, since the number of positive and negative experiences are almost equal, the final result is similar to those obtained when using the Average-Q. However, when the agent lives in a bad world, the number of negative experiences is higher than the positive ones. As a result, the agent decides not to interact with other agents during the steady phase.

Finally, in chapter 10, it was shown the role of the emotion fear. Happiness and sadness have been used as positive and negative reinforcement respectively, allowing the agent to learn a correct policy to survive alone in the world, as well as in company. The emotion fear is used by the agent as a tool for avoiding risky actions and dangerous states.

When the agent executes actions that cause damages to himself frequently, he will learn using Q-learning that these actions must not be taken again. It can be said that the agent is afraid of selecting that action. However, if an action does not cause damage frequently, but only once in a while, the value of this action, calculated using Q-learning, will be high, since it weighs the positive and the negatives experiences. Therefore, a mechanism is needed to learn to avoid this kind of actions: the risky actions. In order to be afraid of these actions, the agent also stored the worst results experienced. Finally, the agent instead of selecting the action that maximizes the expected value, he will select the one that maximizes a combination between the expected and the worst value. Depending on the value of the daring factor, the agent will give more importance to the worst result over the expected one. In the experiments the agent, during the steady phase, starts with the maximum value of the daring factor so he does not consider the worst thing that could happen to him. At the end of this phase the daring factor is decreased and the worst value is more considered. Therefore, the actions whose worst values were very low are not taken anymore. It has been proved that when the agent avoid these risky actions, the quality of life, given by the percentage of permanence in the security zone, increases.

In the second part of this chapter, the case of the agent being in a state where bad things may happen to him, is presented. Therefore in this situation, the agent will be afraid of being in that state. In the experiments, the agent lives with other two agents and one of them is bad with him, but only once in a while. If that agent was bad always, then the learning agent, using Q-learning, would learn not to interact with him. But when the opponent is bad occasionally, then something more is needed to avoid that state, because otherwise, the agent will interact with the opponent despite

the possibility of being hurt. In order to deal with these dangerous states, the emotion fear is introduced as a new drive. The agent feels fear when the worst thing that could happen to him, when he is doing nothing, is lower than a certain limit. Therefore, the agent needs also to store the results obtained from the worst experiences. It has been proved that the agent, when Fear is the dominant motivation, learns to flee. This is the main conclusion from this last part since the flee behaviour was not programmed in advanced. When the agent is with another agent, and he knows that he can be hurt without doing nothing, then he feels fear and learns that the best thing to do is to move.

11.2. Fulfillment of objectives

As it was shown in the introduction chapter, the main objective of this thesis is to design a decision making system, based on emotions and using unsupervised learning, for an autonomous and social robot. In order to carry out this main objective, as a previous step, it was decided to develop this decision making system in a virtual agent. Related to this objective, some problems needed to be solved. Let us show how these problems have been solved along this thesis.

First, there was a problem related to the management of the objects of the world, since the existence of many objects in the world implied a great number of states. As it has been shown in section 6.2, this problem was solved by considering the states related to each object as independent. As a consequence of this simplification, a new learning algorithm based on Q-learning was proposed.

The emotions of happiness and sadness have been defined as the positive and negative variation of the wellbeing of the agent. The wellbeing measures the degree of satisfaction of the drives. In chapter 8, it was proved that, in order to learn a right policy of behaviour, the reinforcement function must be happiness and sadness. Therefore, emotions are used as positive and negative rewards.

In chapter 9, it has been proved that the agent is also able to learn correct policies of behaviour when he shares his environment with other agents. The fact of having a Loneliness drive causes that the agent shows social behaviours. In order to solve the problem about the reward, due to the joint action when interacting with another agent, the agent uses several multi-agent reinforcement learning algorithms for learning the right policy of behaviour. Based on the values of the indicators of the average value of the wellbeing and the percentage of permanence in the security zone showed in table 9.4, it can be said that the best learning algorithm to deal with the social interaction is the new algorithm based on Q-learning. Using this algorithm, the agent obtained the highest values of the indicators for two out of the four worlds. This means that the agent was able to survive in a complex world maintaining his drives with low values.

Emotions are useful in the decision making system since happiness and sadness are used as positive and negative rewards. Therefore, these emotions are essential for

the learning of policies. In relation to fear, when the agent uses fear to avoid risky actions, he improves his quality of life. Moreover, the use of Fear as a motivation makes the agent to learn to escape from dangerous states. Therefore, it has been proved the usefulness of the emotion fear.

11.3. Main contributions

The use of drives, motivations and emotions to model a decision making architecture is not new. In fact, other architectures have also implemented reinforcement learning algorithms to learn behaviour policies, as well as the use of emotions as reinforcement functions. However, in this thesis some novelties have been introduced to improve the performance of the agent.

- Reinforcement function based on emotions: The definitions of the emotions of happiness and sadness, as the positive and negative variations of the wellbeing of the agent, are new. Happiness is produced when something good happens to the agent, and sadness when something bad occurs. Although the use of emotions as reinforcement functions in learning process is not new, the idea of using the variation of the wellbeing of the agent as positive and negative rewards is new.

- Proposal of a new learning algorithm based on Q-learning: In relation to the state of the agent, as it was explained in section 6.2, the number of states was significantly reduced by considering that the inner state of the agent is the dominant motivation and the state of the agent in relation with one object is independent from his states in relation with the rest of objects. Due to this assumption that the states related with the object were independent, the learning process was simplified, but it did not consider the "collateral effects". These effects take into account the possible influence of an action related to an object, over the rest of the objects. Therefore, the learning algorithm used, the Q-learning, was modified to consider these effects.

- Treatment of the multi-agent interaction: The use of multi-agent reinforcement algorithms to deal with this social interaction is novel.

- Implementation of the fear emotion: In relation with the use of fear, this is also new since it has been implemented from two different points of view: to be afraid of executing risky action, and to be afraid of being in a dangerous state. Both types of fear require the additional storing of the worst results experienced. The fact that the agent is afraid of executing actions that occasionally have bad results, improve the quality of life of the agent. On the other hand, when the agent feels fear because he is in a dangerous state, then the agent learns to escape.

11.4. Future works

Even though this work may improve on what is known about decision making architectures, there is still much work to do in this area:

- Currently, the personality factors α_i have the same value, therefore all the drives have the same importance for the wellbeing of the agent. This could be changed by giving different values to these factors, allowing the agent to have different "personalities". Moreover, these parameters could also be learned by the agent, depending on his experience, adapting himself to the environment.
- The wellbeing function could be modified. Instead of being a linear function of the values of the drives, other functions would be considered, even non-linear ones. Emotions could also be involved in the calculation of the wellbeing.
- The identification of the opponent was included in the experiments carried out with Fear as a motivation but not in the multi-agent case. The agent could identify each of his opponents and treat them differently, maybe using different multi-agent algorithms. This could improve the results obtained in the mixed world.
- Other emotions could be implemented such as anger, surprise, etc.
- Emotions, in real life, have a permanence in time, but in this thesis they have been considered as punctual events, which is an approximation. A theoretical research about how emotions are affected by time must be done.
- Happiness and sadness have been considered just as positive and negative rewards for the learning process. If emotions have a certain duration in time, maybe happiness and sadness could also be defined as states, such as fear, and therefore, the agent would learn what to do when he is happy or sad. Moreover, we could consider their effects on other factors. For example, happiness and sadness could affect two known parameters: the daring factor and the parameter δ. When the agent is happy, maybe the daring factor may increase as well as the exploration, this means that δ is also increased. On the contrary, when the agent is sad he would become more coward, decreasing the daring factor, and more conservative, in the sense of exploiting instead of exploring new actions by decreasing δ.
- The incentive or motivational stimuli could be related to the hedonic value of the object, and they could be learned by the agent instead of being fixed a priori. The value of the stimulus would be higher if the agent likes it, for example, a chocolate is much more desirable than an apple. Therefore, the value of the chocolate is higher than the value of apple.
- The state related to the object "world" is unique. Maybe this state could depend on the physical situation of the agent. This means that it could be dependent on the room where the agent is.

- As it has been said, the inner state of the agent is the dominant motivation. Therefore if the dominant one is Hunger, then the agent is hungry, without considering the intensity of the motivation. In real life, people are more or less hungry, so maybe these degrees of motivations could be implemented.
- In relation to the consuming of food, water or medicine, there is no limit. This means, that the agent can eat or drink many times consecutively with no effect. A limit could be set in such a way that when the agent eat too much, then this action is not positive but negative (upset stomach).
- Another important issue is that when the agent lives with other agents, these agents are not learning since they have fixed policies. It would be interesting to test what would happen if every agent in the world is learning.
- As it was said, the final goal of this work is that this decision making system must be implemented on a real robot. More specifically in Maggie, the robot developed by the RoboticsLab at the Carlos III University of Madrid and that it was introduced in section 2.4.

BIBLIOGRAFíA

[Arbib and Fellows, 2004] Arbib, M. A. and Fellows, J. M. (2004). Emotions:from brain to robot. *Trends in Congnitive Sciences*, 8 (12):554–561.

[Arkin, 1988] Arkin, R. C. (1988). Homeostatic control for a mobile robot: Dynamic replanning in hazardous environments. In *SPIE Conference on Mobile Robots, Cambrige, MAA*.

[Arkin, 2004] Arkin, R. C. (2004). *Who needs emotions? The brain meets the robots*, chapter Moving up the food chain: Motivation and Emotion in behavior-based robots. Oxford University Press.

[Asimov, 1983] Asimov, I. (1983). *Los robots del amanecer*. Plaza & Janés.

[Bakker et al., 2003] Bakker, B., Zhumatiy, V., Gruener, G., and Schmidhuber, J. (2003). A robot that reinforcement-learns to identify and memorize important previous observations. In *the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS2003*.

[Balkenius, 1993] Balkenius, C. (1993). Motivation and attention in an autonomous agent. In *Workshop on Architectures Underlying Motvation and Emotion Ð WAUME 93, University of Birmingham*.

[Balkenius, 1995] Balkenius, C. (1995). *Natural Intelligence in Artificial Creatures*. PhD thesis, Lund University Cognitive Studies 37.

[Bartneck and Forlizzi, 2004] Bartneck, C. and Forlizzi, J. (2004). A design-centered framework for social human-robot interaction. In *2004 IEEE International Workshop on Robot and Human Interactive Communication. Kurashiki, Okayama, Japan*, pages 31–33.

[Bellman, 2003] Bellman, K. L. (2003). *Emotions in Humans and Artifacts*, chapter Emotions: Meaningful mappings between the individual and its world. MIT Press.

[Berridge, 2004] Berridge, K. C. (2004). Motivation concepts in behav-ioural neuroscience. *Physiology and Behaviour*, (81):179–209.

[Bolles, 1967] Bolles, R. C. (1967). *Theory of Motivation*. New York: Harper and Row.

[Bonarini et al., 2006] Bonarini, A., Lazaric, A., Restelli, M., and Vitali, P. (2006). Self-development framework for reinforcement learning agents. In *the 5th International Conference on Developmental Learning (ICDL)*.

[Breazeal, 2002] Breazeal, C. (2002). *Designing Sociable Robots*. The MIT Press.

[Breazeal, 2003] Breazeal, C. (2003). *Biological inspired Intelligent Robots*, chapter Cognitive Modeling for Biomimetic Robots. SPIE Press.

[Breazeal and Aryananda, 2002] Breazeal, C. and Aryananda, L. (2002). Recognition of affective communicative intent in robot- directed speech. *Autonomous Robots*, 12:83–104.

[Breazeal and Brooks, 2004] Breazeal, C. and Brooks, R. (2004). *Who Needs Emotions: The Brain Meets the Robot*, chapter Robot Emotion: A Functional Perspective. MIT Press.

[Breazeal et al., 2005] Breazeal, C., Buchsbaum, D., Gray, J., Gatenby, D., and Blumberg, B. (2005). Learning from and about others: Towards using imitation to bootstrap the social understanding of others by robots. *Artificial Life*, 11:1–32.

[Breazeal and Velásquez, 1998] Breazeal, C. and Velásquez, J. (1998). Toward teaching a robot 'infant' using emotive communication act. In *1998 Simulation of Adaptive Behavior, Workshop on Socially Situated Intelligence, Zurich Switzerland*.

[Cañamero, 1997] Cañamero, L. (1997). Modeling motivations and emotions as a basis for intelligent behavior. In *First International Symposium on Autonomous Agents (Agents'97), 148-155. New York, NY: The ACM Press.*

[Cañamero, 2000] Cañamero, L. (2000). Designing emotions for activity selection. Technical report, Dept. of Computer Science Technical Report DAIMI PB 545, University of Aarhus, Denmark.

[Cañamero, 2003] Cañamero, L. (2003). *Emotions in Humans and Artifacts*, chapter Designing emotions for activity selection in autonomous agents. MIT Press.

[Cañamero, 2005] Cañamero, L. (2005). Emotion understanding from the perspective of autonomous robots research. *Neural Networks*, 18:445–455.

[Cañamero and Fredslund, 2000] Cañamero, L. and Fredslund, J. (2000). I show you how i like you: Human-robot interaction through emotional expression and tactile stimulation. Technical report, University of Aarhus, Denmark.

[Cahn, 1990] Cahn, J. (1990). The generation of affect in synthesized speech. *Journal of the American Voice I/O Society*, 8.

[Cannon, 1932] Cannon, W. (1932). *The Wisdom of the Body*. Norton, New York.

[CosAguilera et al., 2003] CosAguilera, I., Cañamero, L., and Hayes, G. (2003). Motivation-driven learning of object affordances: First experiments using a simulated khepera robot. In *the 9th International Conference in Cognitive Modelling (ICCM'03), Bamberg, Germany*.

[CosAguilera et al., 2005a] CosAguilera, I., Cañamero, L., and Hayes, G. (2005a). Ecological integration of affordances and drives for behaviour selection. In *MNAS2005, Edinburgh*.

[CosAguilera et al., 2005b] CosAguilera, I., Cañamero, L., and Hayes, G. (2005b). Motivation-driven learning of action affordances. In *Agents that Want and Like: Motivational and Emotional Roots of Cognition and Action, Workshop of the AISB05 Convention. Hertfordshire, UK*.

[Craig, 1918] Craig, W. (1918). Appetites and aversions as constituents of instincts. *Biol Bull Woods Hole*, 34:91–107.

[Damasio, 1994] Damasio, A. (1994). *Descartes' Error - Emotion, reason and human brain*. Picador, London.

[Ekman, 1992] Ekman, P. (1992). An argument for basic emotions. *Cognition and Emotion*, 6(3/4):169–200.

[Fellows, 2004] Fellows, J. (2004). From human emotions to robot emotions. Technical report, AAAI 2004 Spring Symposium on Architectures for Modelling Emotion: Cross- Disciplinary Foundations.SS-04-02. AAAI Press.

[Fong et al., 2002] Fong, T., Nourbakhsh, I., and Dautenhahn, K. (2002). A survey of socially interactive robots: Concepts, design, and applications. Technical report, CMU-RI-TR-02-29.

[Frijda and Swagerman, 1987] Frijda, N. and Swagerman, J. (1987). Can computers feel? theory and design of an emotional model. *Cognition and Emotion*, (1 (3)):235–357.

[Fujita, 2001] Fujita, M. (2001). Aibo: Toward the era of digital creatures. *The International Journal of Robotics Research*, (vol 20, nº 10):781–794.

[Fujita and Kitano, 1998] Fujita, M. and Kitano, H. (1998). Development of an autonomous quadruped robot for robot entertainment. *Autonomous Robots*, 5:7–18.

[Fujitsu, 2007] Fujitsu (2007). http://www.pfu.fujitsu.com/maron/.

[Gadanho, 1999] Gadanho, S. (1999). *Reinforcement Learning in Autonomous Robots: An Empirical Investigation of the Role of Emotions*. PhD thesis, UNiversity of Edinburgh.

[Gadanho, 2002] Gadanho, S. (2002). Emotional and cognitive adaptation in real environments. In *Sympo-sium ACE'2002 of the 16th European Meeting on Cybernetics and Systems Research, Vienna, Austria*.

[Gadanho, 2003] Gadanho, S. (2003). Learning behavior-selection by emotions and cognition in a multi-goal robot task. *The Journal of Machine Learning Research. MIT Press Cambridge, MA, USA*, (4):385–412.

[Gadanho and Custodio, 2002a] Gadanho, S. and Custodio, L. (2002a). Asynchronous learning by emotions and cognition. In *From Animals to Animats VII, Proceedings of the Seventh International Conference on Simulation of Adaptive Behavior (SAB'02), Edinburgh, UK*.

[Gadanho and Custodio, 2002b] Gadanho, S. and Custodio, L. (2002b). Learning behavior-selection in a multi-goal robot task. Technical report, RT-701-02, Instituto de Sistemas e Robotica, IST, Lisboa, Portugal.

[Gadanho and Hallam, 2001] Gadanho, S. and Hallam, J. (2001). Emotion- triggered learning in autonomous robot control. *Cybernetics and Systems*, 32(5):531–59.

[Gadanho and Hallam, 2002] Gadanho, S. and Hallam, J. (2002). Robot learning driven by emotions. *Adaptive Behavior*, 9(1):42–64.

[Gautier and Boeree, 2005] Gautier, R. and Boeree, G. (2005). *Teorías de la Personalidad: una selección de los mejores autores del S. XX*.

[Hallam and Hayes, 1992] Hallam, B. and Hayes, G. (1992). Comparing robot and animal behaviour. Technical report, DAI Research Paper 598, University of Edinburgh.

[Howard, 1960] Howard, R. (1960). *Dynamic Programming and Markov Decision Processes*. MIT Press.

[Hu and Wellman, 1998] Hu, J. and Wellman, M. P. (1998). Multiagent reinforcement learning: theoretical framework and an algorithm. In *of the Fifteenth International Conference on Machine Learning, San Francisco, California*.

[Hull, 1943] Hull, C. L. (1943). *Principles of Behavior*. New York: Appleton Century Crofts.

[Humphrys, 1997] Humphrys, M. (1997). *Action Selection methods using Reinforcement Learning*. PhD thesis, Trinity Hall, Cambridge.

[iRobot, 2007] iRobot (2007). http://www.irobot.com/.

[Isbell et al., 2001] Isbell, C., Shelton, C. R., Kearns, M., Singh, S., and Stone, P. (2001). A social reinforcement learning agent. In *the fifth international conference on Autonomous agents, Montreal, Quebec, Canada*.

[Ishiguro, 2007] Ishiguro, H. (2007). http://www.ed.ams.eng.osaka-u.ac.jp/.

[Kubota et al., 2001] Kubota, N., Kojima, F., and Fukuda, T. (2001). Self- consciousness and emotion for a pet robot with structured intelligence. In *Joint 9th IFSA World Congress and 20th NAFIPS International Conference. NJ, USA*.

[LeDoux, 1996] LeDoux, J. (1996). *El cerebro emocional*. Ariel/Planeta.

[LeDoux, 2002] LeDoux, J. (2002). *Synaptic Self: How brains become who we are*. Penguin Books.

[Lewin, 2001] Lewin, D. (2001). Why is that computer laughing? *IEEE Intelligent Systems*, 16(5):79–81.

[Lewis, 2004] Lewis, S. C. (2004). *Computational Models of Emotion and Affect*. PhD thesis, University of Hull.

[Lisetti, 1999] Lisetti, C. L. (1999). Emotion generation via a hybrid architecture. In *Autonomous Agents Workshop on Emotion-Based Agent Architecture (EBAA'99)*.

[Littman, 1994] Littman, M. (1994). Markov games as a framework for multiagent learning. In *Proceedings of the Eleventh International Conference on Machine Learning, San Francisco, California*, pages 157–163.

[Littman, 2001] Littman, M. (2001). Friend-or-foe q-learning in general-sum games. In *Proceedings of the Eighteenth International Conference on Machine Learn-ing*, pages 322–328.

[Lorenz, 1977] Lorenz, K. (1977). *Behind the Mirror*.

[Lorenz and Leyhausen, 1973] Lorenz, K. and Leyhausen, P. (1973). *Motivation of human and animal behaviour; an ethological view*, volume xix. New York: Van Nostrand-Reinhold.

[Martinson et al., 2002] Martinson, E., Stoytchev, A., and Arkin, R. (2002). Robot behavioral selection using q-learning. In *of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), EPFL, Switzerland*.

[Mataric, 1998] Mataric, M. (1998). Behavior-based robotics as a tool for synthesis of artificial behavior and analysis of natural behavior. *Trends in Cognitive Science*, 2(3):82–87.

[Mayer and Salovey, 1993] Mayer, J. D. and Salovey, P. (1993). The intelligence of emotional intelligence. *Intelligence*, 17(4):433–442.

[Michaud et al., 2001] Michaud, F., Audet, J., Létourneau, D., Lussier, L., ThébergeTurmel, C., and Caron, S. (2001). Experiences with an autonomous attending aaai. *IEEE Intelligent Systems*, 16(5):23–29.

[Minsky, 1986] Minsky, M. (1986). *The Society of Mind*. Simon and Schuster.

[NEC, 2006] NEC (2006). http://www.incx.nec.co.jp/robot/.

[Oatley, 1996] Oatley, K. (1996). *Understanding Emotions*. Blackwell.

[Omron, 2007] Omron (2007). http://www.necoro.com/.

[Ortony, 2003] Ortony, A. (2003). *Emotions in Humans and Artifacts*, chapter On making Believable Emotional Agents Believable, pages 188–211. MIT Press.

[Ortony et al., 1988] Ortony, A., Clore, G. L., and Collins, A. (1988). *The Cognitive Structure of Emotions*. Cambridge University Press. Cambridge, UK.

[Ortony et al., 2005] Ortony, A., Norman, D. A., and Revelle, W. (2005). *J.M. Fellous and M.A. Arbib, Who needs emotions: The brain meets the machine*, chapter Affect and proto-affect in effective functioning.

[Philips, 2007] Philips (2007). http://www.research.philips.com/robotics.

[Picard, 1998] Picard, R. W. (1998). *Los ordenadores emocionales*. Ed. Ariel S.A.

[Picard, 2003] Picard, R. W. (2003). *Emotions in Humans and Artifacts*, chapter What does it mean for a computer to have emotions? MIT Press.

[Pransky, 2001] Pransky, J. (2001). Aibo- the nº.1 selling service robot. *Industrial Robot: An International Journal*, 28(1):24–26.

[Probotics, 2007] Probotics (2007). http://www.personalrobots.com.

[Ribeiro et al., 2002] Ribeiro, C. H. C., Pegoraro, R., and RealiCosta, A. H. (2002). Experience generalization for concurrent reinforcement learners: the minimax-qs algorithm. In *AAMAS 2002*.

[Rolls, 2003] Rolls, E. (2003). *Emotions in Humans and Artifacts*, chapter Theory of emotion, its functions, and its adaptive value. MIT Press.

[Rolls, 2005] Rolls, E. (2005). *Emotion Explained*. Oxford University Press.

[Salichs et al., 2006] Salichs, M., R.Barber, A.M.Khamis, M.Malfaz, J.F.Gorostiza, R.Pacheco, R.Rivas, A.Corrales, and E.Delgado (2006). Maggie: A robotic platform for human-robot social interaction. In *IEEE International Conference on Robotics, Automation and Mechatronics (RAM 2006). Bangkok. Thailand.*

[Santa-Cruz et al., 1989] Santa-Cruz, J., Tobal, J. M., Vindel, A. C., and Fernández, E. G. (1989). Introducción a la psicología. Facultad de Psicología. Universidad Complutense de Madrid.

[Scherer, 1998] Scherer, K. (1998). Analysing emotion blends. In *ISRE 98 Symposium.*

[Shibata et al., 1996] Shibata, T., Ohkawa, K., and Tanie, K. (1996). Spontaneous behaviour of robots for cooperation. emotionally intelligent robot system. In *the 1996 IEEE. International Conference on Robotics and Automation*.

[Shibata et al., 1999] Shibata, T., Tashima, T., Arao, M., and Tanie, K. (1999). Interpretation in phisical interaction between human and artificial emotional creature. In *the 1999 IEEE. International Workshop on Robot and Human Interaction*.

[Shoham et al., 2003] Shoham, Y., Powers, R., and Grenager, T. (2003). Multi-agent reinforcement learning: a critical survey. Technical report, Computer Science Department, Stanford University, Stanford.

[Sloman, 2003] Sloman, A. (2003). *Emotions in Humans and Artifacts*, chapter How many separately evolved emotional beasties live within us. MIT Press.

[Smart and Kaelbling, 2002] Smart, W. D. and Kaelbling, L. P. (2002). Effective reinforcement learning for mobile robots. In *International Conference on Robotics and Automation (ICRA2002)*.

[Sony, 2006] Sony (2006). http://www.sonydigital-link.com/aibo/index.asp.

[Sutton and Barto, 1998] Sutton, R. S. and Barto, A. G. (1998). *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, A Bradford Book.

[Tashima et al., 1999] Tashima, T., Saito, S., Kudo, T., and Osumi, M. (1999). Interactive pet robot with an emotion model. *Advanced Robotics*, 13(3):225–226.

[Thomaz et al., 2005] Thomaz, A. L., Berlin, M., and Breazeal, C. (2005). An embodied computational model of social referencing. In *Fourteenth IEEE Workshop on Robot and Human Interactive Communication (Ro-Man05)*.

[Thomaz and Breazeal, 2006] Thomaz, A. L. and Breazeal, C. (2006). Transparency and socially guided machine learning. In *the 5th International Conference on Developmental Learning (ICDL)*.

[Toates, 1986] Toates, F. (1986). *Motivational systems*. Cambridge (MA): Cambridge Univ. Press.

[Touzet, 2003] Touzet, C. (2003). *The Handbook of Brain Theory and Neural Networks*, chapter Q-learning for robots, pages 934–937. MIT Press.

[Trespalacios, 1980] Trespalacios, J. F. (1980). *Psicología General II*. Madrid: U.N.E.D.

[Velásquez, 1997] Velásquez, J. (1997). Modeling emotions and other motivations in synthetic agents. In *Fourteenth National Conf. Artificial Intelligence*.

[Velásquez, 1998a] Velásquez, J. (1998a). Modelling emotion-based decision-making. In *1998 AAAI Fall Symposium Emotional and Intelligent: The Tangled Knot of Cognition*.

[Velásquez, 1998b] Velásquez, J. (1998b). When robots weep: Emotional memories and decision making. In *Proceedings of AAAI-98*.

[Velásquez, 1999] Velásquez, J. (1999). An emotion-based approach to robotics. In *I999 IEEE/RSJ International Conference on Intelligent Robots and Systems*.

[Ávila García and Cañamero, 2002] Ávila García, O. and Cañamero, L. (2002). A comparison of behavior selection architectures using viability indicators. In *Proc. International Workshop Biologically-Inspired Robotics: The Legacy of W. Grey Walter(WGW'02).*

[Ávila García and Cañamero, 2004] Ávila García, O. and Cañamero, L. (2004). Using hormonal feedback to modulate action selection in a competitive scenario. In *Proc. 8th Intl.Conference on Simulation of Adaptive Behavior (SAB'04).*

[Ávila García and Cañamero, 2005] Ávila García, O. and Cañamero, L. (2005). Hormonal modulation of perception in motivation-based action selection architectures. In *Agents that Want and Like: Motivational and Emotional Roots of Cognition and Action, Workshop of the AISB05 Convention. University of Hertfordshire, UK.*

[Watkins, 1989] Watkins, C. J. (1989). *Models of Delayed Reinforcement Learning.* PhD thesis, Cambridge University, Cambridge, UK.

[Watkins and Dayan, 1992] Watkins, C. J. and Dayan, P. D. (1992). Technical note: Qlearning. *Machine Learning*, 8(3):279–292.

[Yang and Gu, 2004] Yang, E. and Gu, D. (2004). Multiagent reinforcement learning for multi-robot systems: A survey. Technical report, CSM-404.University of Essex.

[Zimmerman, 2007] Zimmerman, B. (2007). http://www.coffeemud.org/.