MULTI-ACTORS DISTRIBUTED CONTROL SYSTEMS: REINFORCEMENT SIGNAL BY SHANNON'S ENTROPY

Youcef Zennir[(1)], Denis Pomorski[(2)]

[(1)]Laboratoire d'automatique de Skikda,26 route el-hadaik,21000 Skikda, Algérie
[(2)]Laboratoire L.A.G.I.S, Polytech'Lille, Bâtment D, 59655 Villeneuve d'Ascq, France

[(1)]youcefzennir@yahoo.com [(2)]denis.pomorski@univ-lille1.fr

ABSTRACT
This paper presents a multi-actors distributed control systems in an unknown environment. These actors are reactive entities able to react to the stimuli coming from the environment and to choose between several actions. In order to improve their behaviour (i.e. in order to choose the good action) in the course of time, the multi-actors system must be able to use reinforcement learning. This signal of reinforcement is, until now, a signal whose values are previously defined. We propose to raise this technique by using the Shannon's entropy to measure the coherence of the action choice using the transformation of the reinforcement signal table. This stage, of local training will allow the improvement of the control of the global system and coordination between the various actors. The results of the simulation show that the actor can learn to control its trajectory efficiently.

KeyWords: Reinforcement learning, distributed control, Q-learning, multi-actors systems.

1. INTRODUCTION

The multi-actors systems in which actors must learn together how to achieve a common task, this constitutes a very active field of research (Claus 1998), (Littman 2001), (Hu 1998). A difficulty in such systems is of knowing how to coordinate the actors effectively so that they gain ones from the others without harming any of them. The reactive actors considered in this article react to receive stimuli of the environment.

These actors can be observed but, contrary to the cognitive actors, they cannot communicate. As an example of a distributed control system we consider a robot model with several sensors (actors) moving in an unknown environment. The planning of coordination and communication between the actors is not effective. It is then interesting to resort to training, such as the reinforcement learning which is based on a process test/error to acquire the desired behaviour. The object of this work is to study and develop a method of learning for multi-actors systems with the use of the theories of data fusion (Shannon's entropy) to measure the coherence of the choice of the action by the transformation of the reinforcement signal table.

2. REINFORCEMENT LERANING

Reinforcement learning, is one of the most active research areas in artificial intelligence, it is a computational approach for learning whereby an agent tries to maximize the total amount of reward it does receive when interacting with a complex, and uncertain environment.

In the standard reinforcement learning model an agent interacts with its environment (Sutton 1998). This interaction takes the form of the agent sensing the environment based on this sensory input, this enable choosing an action to perform in the environment. The action changes the environment in some manner and this change is communicated to the agent through a scalar reinforcement signal.

There are three fundamental parts of a reinforcement learning problem: the environment, the reinforcement function, and the value function.

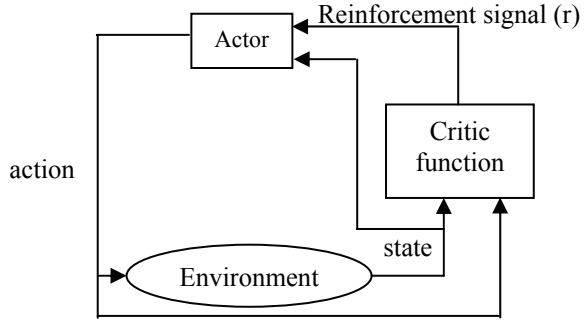

Figure 1: The actor-environment interaction in reinforcement learning.

Q-learning (Zennir 2004) is a recent form of Reinforcement Learning algorithm that does not need a model of its environment and can be used online. Therefore, it is very suited for repeated games against an unknown opponent. Q-learning algorithms works by estimating the values of state-action pairs Q(s,a).

The value Q(s,a) is defined to be the expected discounted sum of the future payoffs obtained by taking action "a" from state "s" and following an optimal policy thereafter. Once these values have been learned, the optimal action from any state is the one with the highest Q-value. Its simplest form, *1-step Q-learning* is defined by

$$Q(s,a) \leftarrow Q(s,a) + \alpha.[r + \gamma.\max_{a'} Q(s',a') - Q(s,a)] \qquad (1)$$

- α : learning rate, γ: discount factor.

In this case, the learned action-value function, **Q**, directly approximates **Q***, the optimal action-value function, is independent of the policy being followed. This dramatically simplifies the analysis of the algorithm and enabled early convergence proofs. The policy still has an effect in that it determines which state-action pairs are visited and updated. However, all what is required to correct the convergence is that all pairs continue to be updated.

As we have discussed before, this is a minimum requirement in the sense that any guaranteed method must fulfil this requirement to find optimal behaviour in the general case. Under this assumption and based on the usual stochastic approximation conditions on the step-size sequence, **Q** has been shown to converge with probability one to **Q**.

The Q-learning algorithm is shown in procedural form in Figure 2.

Initialize Q(s,a) arbitrarily
Repeat (for each episode):
Initialize **s**

Repeat (for each step of episode):

Choose "**a"** from "**s"** using policy derived from **Q** (e.g., ε-greedy)

Take action "**a"**, observe "**r"**, s'
$Q(s,a) \leftarrow Q(s,a) + \alpha.[r + \gamma.\max_{a'} Q(s',a') - Q(s,a)]$
s←s';

Until "**s"** is terminal

Figure 2: Q-learning algorithm.

Shortly, we have:

- ε: Probability to use a random action instead of the optimal policy.

It was proven, for example in (Jaakkola 1994), that if spaces of states and actions S and A are finished, if α_t is such as:

$$\sum_t \alpha_t = \infty \qquad \text{and} \qquad \sum_t \alpha_t^2 \leq \infty \qquad (2)$$

This algorithm is for one actor, but where more than one actor work in the same space we have other algorithms adapted for this new context.

3. REINFORCEMENT LEARNING IN A MULTI-ACTORS CONTEXT

Let us consider a system made up of N actors A_i (i=1... N) whose interaction with the environment is defined by:
- A space actions $\alpha i=\{a_{i1}... ,a_{ij}... ,a_{iNi}\}$ for each actor A_i (i=1... N), with N_i=card(a_i);
- A space states $S=\{s_1,... ,s_i... ,s_M\}$;

At the times "t", each actor carries out an action which is clean on the basis of state preceding "s_t" of the environment. These joint actions (a_1... a_i... ,a_N) (with $a_i \in \alpha_i$) cause a transition towards the state "s'" (the apostrophe meaning the moment "t+1"). According to architecture used, one critic's function allows to reward the actors. Among architectures of training by reinforcement the most used, we distinguish centralized, distributed and multi-actors architecture.

With a centralized approach of the reinforcement learning, state information are collected only in one decision centres, which updates the utility values and decides actions for each actor. By indicating |A| the number of achievable actions by each actor (the number is identical for all the actors without loss of generalization), and |S| the number of states attainable by the system, the table of Q(s, a) in memory is of size $|S|*|A|^N$.

A same reinforcement signal is allotted to all the actors whatever the contribution of each one of those to the success or the failure of the common task. The expected advantages of this centralized approach are:

- A global vision of the system, allowing examining the all situations accessible by the actions from the actors.
- The possible problems of coordination between actors are solved on a single level of decision.

This approach presents the following inconvenient:

- The total system is sensitive to the failure of the single decision centres.
- The number of pair state/action (s, a) grows exponentially with the number of actors.
- Beneficial actions on the global system can be penalizing for an actor, because it is not held account of the local constraints.

3.1. Distributed architecture

With a distributed architecture (Zennir 2003) the actors do not receive necessarily the same information of state or the same signal reinforcement, and perform their own training. By supposing that N actors share the same information of state, the number of Q(s, a) to update is N*|S|*|A|. The expected advantages of this distributed approach are:

- A greater flexibility (facilitated adaptation to the unforeseen modifications of the environment).
- A greater reliability (the individual error is tolerated).
- A greater robustness (the capacity of resolution results from the collective and not from an individual).
- Each actor can take into account local constraints.
- The number of pair state/action (s, a) grows proportionally with the number of actors.

In the distributed reinforcement learning approach, the strategies which can follow the actors implied in the same task can be individual or collective. According to an individual strategy each actor carries out his learning by ignoring the other actors. That amounts of learning by applying the algorithms acts as if each actor were alone.

The environment of the actor is then non stationary bus during the learning, a "a" share carried out since the same state "s" always does not lead to the same state "s" because the state reached depends on the actions of the other actors.

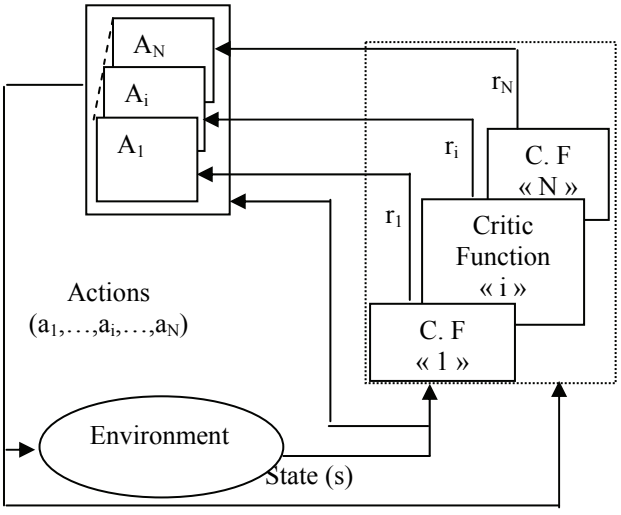

Figure 3: Distributed approach of reinforcement learning: one reinforcement signal « r_i » for each actor.

But this architecture presents the following difficulties: Within the context of a collective task the consequences of the action of an actor depend on the actions of the other actors.

- When they are defined, the local goals pursued by each actor must be compatible with the global objective.
- Mechanisms of co-operation among which, synchronization, collaboration, coordination can be necessary.

3.2. Multi-Actors architecture

Learning behaviours in a multiagent environment is crucial for developing and adapting multiactor systems. Reinforcement learning techniques have addressed this problem for a single actor acting in a stationary environment, which is modelled as a Markov decision process. But, multi-actors environments are inherently non-stationary since the other actors are free to change their behaviour as they also learn and adapt. Hu (Hu 1998), (Zennir 2004) extended the Q-multi-actor algorithm to general-sum games.

The extension requires that each agent maintain values for all the other actors. Also, the linear programming solution used to find the equilibrium of zero-sum games is replaced with the quadratic programming solution for finding an equilibrium in general-sum games.

The game must have a unique equilibrium, which is not always true of general-sum stochastic games. This is necessary since the algorithm strives for the opponent-independence property of Q-multi-actors, which allows the algorithm to converge almost regardless of the other agent's actions. With multiple equilibrium it is important for all the actors to play the same equilibrium in order to have its reinforcing properties. So, learning independently is not possible. It is supposed that at the time to act some actors do not know a priori the actions which are selected by the other actors but within the same given group, each one of them knows a posterior executed action by the other members of the group.

Within the context of a collective vision, we consider an algorithm in with which it is possible to the actors of a group to hold account of the actions selected by the other members. The principle is described in the following example: considering three actors 'i', 'j', 'k' laying out each one of two possible actions noted 0 and 1. For each state s, the actors maintain four Q-value tables corresponding to the four possibilities of action of the two other actors (Figure 5).

Each actor chooses the action leading to a hope of maximum gain i.e. that which, for a given state "s", corresponds to the line comprising the largest value of gains (gain 7 in Figure 5), even if the other actors choose or not the actions corresponding to the column comprising this value. However, at the reception of the reinforcement signal, the actor updates the value also corresponding to the choice of action of the other actors.

Thus in the example of Figure 3, if the actor 'j' executes action 0 and the actor 'k' action 1, it is the Q = - 4 value intersection between the line $a_i = 1$ and the column $a_j = 0$, $a_k = 1$ which is modified.

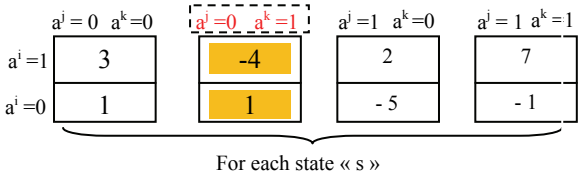

Figure 4: Q-Value for actor 'i' in state 's', with 3 actors 'i' ,'j', 'k' and 2 actions.

The value of Q, which is updated at each iteration, is that which corresponds to the actions really executed. According to this approach, for each member 'i' of a given group of K actors, the Q-learning algorithm becomes:

For each i in a group
Initialise $Qi(s, a_1,..,a_{i-1}, a_i, a_{i+1}, ... a_K)$ to 0i
For each episode
For any stage of the episode
For each actor i
Choose action "a_i" from "s" using policy Q_i (eg ε-greedy) whatever the actions chosen by other actors,
Observe s', r, a_k for all k !=i
Actualise Q_i
$Q_i(s, a_1,..,a_{i-1}, a_i, a_{i+1}, ... a_K) \leftarrow Q_i(s, a_1,..,a_{i-1}, a_i, a_{i+1}, ... a_K)$
$+ \alpha_i.[r_i+\gamma_i.\max_{ai'}Q_i(s',\_, ai', \_) - Q_i(s, a_1,..,a_{i-1}, a_i, a_{i+1}, ... a_K)]$

$s \leftarrow s'$

Figure 5: Q-multi-actors Algorithm's.

4. REINFORCEMENT'S SIGNAL USED IN LITERATURE

The signal of reinforcement is determined by the critical function, based on rules. The choice of the rules of delivery of this signal largely conditions the success of the learning and the final behaviour of the actor (Touze 1993). To be convinced some, let us evoke some examples:

- To make move a robot towards a goal, one can at every moment give a reward which is inversely proportional to his distance with the goal or which is a function of the final result.
- To build an actor which the goal is to leave a labyrinth, one can give a null reward most of the time, and "+1" as soon as the actor reaches the exit (Kaelbling 1996).

To prevent such an actor does not knock himself against the walls; one can sanction it each time that it touches a wall by allotting to him a penalty (Buffet 2003). In general the choice of these rules is based on the intuition. The signal of reinforcement is a discrete signal, limited, composed of two or three values to the maximum.

5. VALIDATION OF ACTOR CHOICES

Knowing the state "s" in which the environment is, the actor must make a decision as for the action which it must take. With an aim of identifying this action, and especially of knowing if it is single or not, we can use the resulting tools from the information's theory. The variable which one seeks to explain is thus the action "a_i", knowing the state "s". Thus, knowing that the environment is in the state " s_i", several cases of figures can occur:

- The actor A_i will be able to make a decision have single if there is only one signal of positive reinforcement for $S=s_i$.
- In the contrary case, the signal of reinforcement will not make it possible to the actor to make a no ambiguous decision.

We can thus propose to use the reinforcement transformation table in the following way

$$p_{j/k} = \frac{r_{kj}}{\sum_{j=1}^{Ni} sign(r_{kj}) \text{ avec } r_{kj} > 0} \text{ if } P_{kj} > 0 \quad (3)$$

$p_{j/k} = 0$ else

We can thus draw up the following table:

	a_i						
S		α_1	α_2	..	α_j	… …	α_{Ni}
s_1					.		
s_2					.		
…					.		
s_k		…..	…	..	$P_{j/k}$	…	….
….							
s_m							

Figure 6: Coherence table $[P_{J/K}]$

where : $P_{J/K}:=\{p_{j/k}; k \in K, j \in J\}$,
$K=\{1,2,...,m\}$;
$m = \text{card } S$;
$J=\{1,2,...,N_i\}$;
$N_i = \text{card } a_i$

$P_{j/k}$ can be seen as a measurement of choice coherence's of the action J knowing that the environment is in the state s_k.

The coherence of the decision-makings of the actor "a_i" is thus maximum if there is only one "1" per line of the preceding table. In the opposite case, the data are incoherent. We can quantify this inconsistency by used concepts resulting from the information's theory: indeed, the intensity of connection between two variables "S" and "a_i" can be moderate by means of the conditional entropy (Pomorski 1991):

$$H(a_i/S) = H(S,a_i) - H(S) = -\sum_{k,j} p_{kj}.\log p_{j/k} \quad \text{with} \quad p_{kj} = \sum_{k} p_{j/k} \qquad (4)$$

Based on H(Y/X) and H(Y/S), two indices of "modelisability" are used:

$$. \; m(a_i/S) = 1 - \frac{H(a_i/S)}{H(a_i)} \qquad (5)$$

Moderate of a_i par S. It is a measure of the inconsistency of the action's sensors A_i.

$$. \; q(Y/S) = \frac{H(Y) - H(Y/S)}{H(Y) - H(Y/X)} \qquad (6)$$

Moderate the quality of the model, more easy $Y=\tilde{f}(S)$ compared to the quality of the Y=f(X) model.

6. SIMULATION

We considered two actors removing themes selves in an unknown environment and obstacles are placed in different positions. The objective of each actor is to find as quickly as possible the goal supposed to be in a fixed position. The research of the goal is based on a reinforcement learning process of the Q-learning type. In this application, we study the application of the two approach of reinforcement learning (distributed and multi-actors architecture) with the use of the Shannon's entropy for measurement of reinforcement signal and to study the choice of the actions and coordination between the actors.

Simulation with distributed architecture, each actor has a process of training based on Q-learning.

Each actor has a function Q-value (s, a) represented by a table of dimension 400 lines (a number of the states) and 4 columns (a number of possible actions for each actor).

We applied ε-greedy strategy for the exploration of the actions. Figure 7 represents the training of the actor "i" with a distributed architecture.

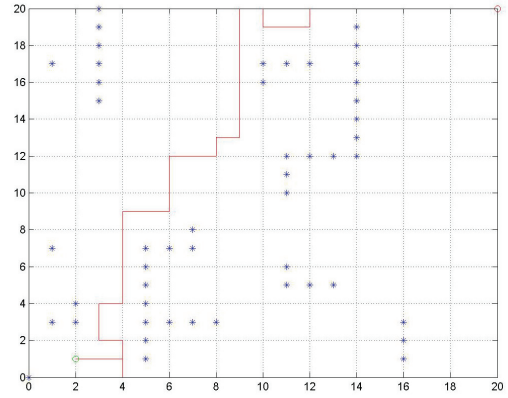

Figure 7: Trajectory of the actor « i ». *Certitude « i »= 0.1094, γ=0.9, α=0.1, ε=0..1.*

In figure.6, we noticed that more than 1000 iterations, the actor "i" arrives to find the goal more than 960 times in the phase of training and 1000 times in the test. We noticed that the time of training (the iteration count to find the goal) is smaller when "ε" is closer to 1.

For the simulation with multi-actors architecture, each actor has a process of training based on Q-learning. The choice of the action for an actor independent of the action is chosen by the other actor, but the update function Q(s, a_j) of an actor to take into accounts the action chosen by the other actor.

Each actor (i and j) has a critical function. We applied ε-greedy strategy for the exploration of the actions. We have test with two actors. The following figures show the results obtained.

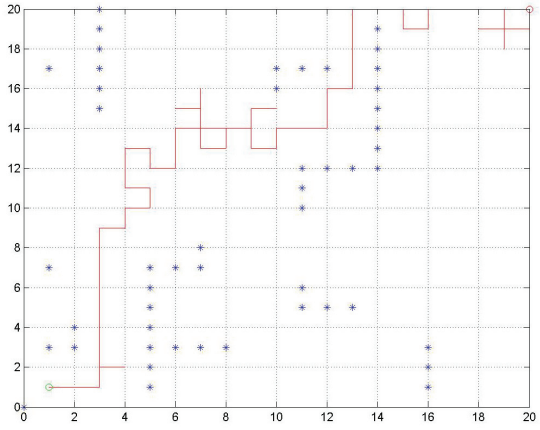

Figure 8: Trajectory of the actor « j ». *Certitude « j »= 0.2004, γ=0.5, α=0.1, ε=0.5.*

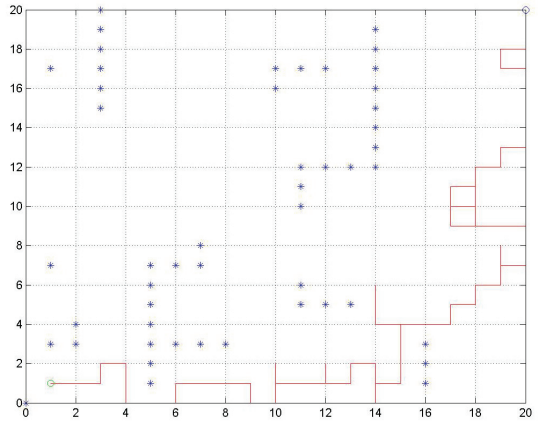

Figure 9: Trajectory of the actor « i ». *Certitude* = 0.1565, $\gamma = 0.9$, $\alpha = 0.1$, $\varepsilon = 0.5$.

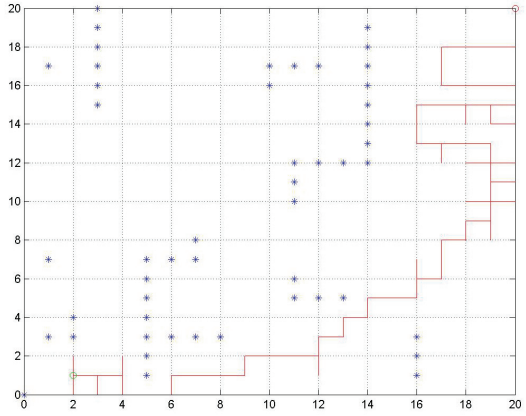

Figure 10: Trajectory of the actor « j ». *Certitude « j »*= 0.1402, $\gamma = 0.9$, $\alpha = 0.1$, $\varepsilon = 0.1$.

We have noticed that with the Q-multi-actors approach, the number of times that the actors "i" and "j" have to find the goal is larger than in the distributed architecture. We also noticed that the certainty for the actor "I" or "J" are larger compared to the two preceding architecture. The multi-actors approach with a strategy of communities gives the best results.

7. CONCLUSION

We have presented in this paper some problems of distributed control system in a multi-actors system. Then we gave a short definition of the reinforcement learning with its principle and various architectures for the improvement of actors' behaviours. In the third part we have been dealing with Shannon's entropy which we have used to treat the coordination and the training of the actors and the measurement of the coherence choices of the action for the transformation of the reinforcement signal table.

The results show that the main advantage of distributed is the reduction in communication costs. The Q-learning distributed and Q-multi-actors algorithms which we have presented in this paper with Shannon's entropy technique for reinforcement signal calculated show that with a given actor, the training is faster and that the Shannon's entropy shows well that the actor did not manage to learn and to coordinate with the other actors and it shows that there is coherence in the choice of the actions.

Finally, generally the number of equation to solve simultaneously increases very quickly with the number of actors. Our future work is as follows:

We study the application of these learning's architecture on a great number of actors to treat the complexity of coordination and to deal with the problems of training time. We will study too with multi-actors architecture the problems of interaction, communication and the co-operation between the various actors.

REFERENCES

Claus C., and Boutillier C., 1998. *The Dynamics of Reinforcement Learning in Cooperative Multiagent Systems.* AAAI, pp.746-752.

Littman M. L*, 2001. Value-function reinforceùment learning in Markov games.* Journal of cognitive Systems Research 2, Elsevier, pp. 55-66.

Hu.J, Wellman M.P.,1998. *Multiagent Reinforcement Learning: Theoretical Framework and an Algorithm.* 15th International Conference on Machine Learning, in Madison, Wisconsin, USA, pp. 242-250.

Sutton R.S., and Barto A.G., 1998. *Reinforcement Learning*, Mit press, Cambridge, Bradford book, 322p., ISBN 0-262-19398-1.

Jaakkola J., Jordan M.I., and Singh S.P., 1994. *Renforcement learning algorithm for partially observable markov decision problems*", In G.Tesauro, Eds, advances in neural information processing systems, vol. 7, pp.345-352.

Touzet C., 19993. *Apprentissage par renforcement neuronal d'un comportement d'obstacles pour le mini-robot Khepera*. Second European congress on systems sciences, Prague, pp.5-8.

Buffet O., Dutech A., and Charpillet F., 2003. *Apprentissage par renforcement pour la conception de systèmes multi-Agents réactifs"*. In Journées Francophones sur les Systèmes Multi-Agents, Hammamet, Tunis, pp.219-231.

Pomorski D., Staroswiecki M., Barboucha M, 1990. *Modelling complex systems via the analysis of qualitative data.* MIM-s2, Bruxelles, p6.

Zennir Y., and Couturier P., 2004. *Control of the trajectory of a hexapod robot based on distributed Q-learning.* IEEE International Symposium on Industrial Electronics, Ajaccio, France, pp.277-282.

Zennir Y., Couturier P., and Bétemps M., 2003. *Distributed reinforcement learning of a six-legged robot to walk.* IEEE 4th International conference on control and automation, Control chapter, Singapore, Montreal section, Montréal, Canada, pp.896-900.