CONTINUOUS ACTION REINFORCEMENT LEARNING AUTOMATA
Performance and Convergence

Abdel Rodríguez[1,2], Ricardo Grau[1]
[1]*Bioinformatics Lab, Center of Studies on Informatics, Central University of Las Villas, Santa Clara, Cuba*
[2]*Computational Modeling Lab, Vrije Universiteit Brussel, Brussels, Belgium*

Ann Nowé
Computational Modeling Lab, Vrije Universiteit Brussel, Brussels, Belgium

Keywords: CARLA, Convergence, Performance.

Abstract: Reinforcement Learning is a powerful technique for agents to solve unknown Markovian Decision Processes, from the possibly delayed signals that they receive. Most RL work, in particular for multi-agent settings, assume a discrete action set. Learning automata are reinforcement learners, belonging to the category of policy iterators, that exhibit nice convergence properties in discrete action settings. Unfortunately, most applications assume continuous actions. A formulation for a continuous action reinforcement learning automaton already exists, but there is no convergence guarantee to optimal decisions. An improve of the performance of the method is proposed in this paper as well as the proof for the local convergence.

1 INTRODUCTION

Since Artificial Intelligence emerged, it has been trying to emulate human behavior with the hope that some day computers will learn how to act as perfect humans. Reinforcement Leaning is the way animals learn how to maximize their profits in certain situations. It is based on random but not uniform exploration. The basis of Reinforcement Learning is to explore actions and reinforce positively those that resulted in a good outcome for the learner, or reinforce negatively the ones that produced bad results.

The mathematical abstraction of this learning is already formulated for discrete actions, but in many engineering applications it is necessary to control continuous parameters. Continuous formulations of Reinforcement Learning are not developed as good as discrete action learners. For single agents there is already quite a lot of work on continuous action learning but there is not much work done in multi-agent settings.

This paper performs an analysis of the performance of Continuous Action Reinforcement Learning Automaton (CARLA) (Howell et al., 1997) on its usefulness for future exploration in Multi-agent Systems (MAS). Classical definition of random variable (RV), will be used as well as the probability integral transformation for the generation of random numbers following a given probability distribution (Parzen, 1960). Next section introduces the LA and as a first contribution of this paper, subsection 2.2 shows how the numerical calculations can be reduced by some mathematical derivations. Following subsection 2.3 introduces the local convergence proof as well as the way to manage the λ parameter to improve this convergence as a second contribution. To support the theoretical results, some experiments are presented in the section 3. Finally, conclusions and future work are stated in section 4.

2 LEARNING AUTOMATA

The learning automaton is a simple model for adaptive decision making in unknown random environments. The concept of a Learning Automaton (LA) originated in the domain of mathematical psychology (Bush and Mosteller, 1955) where it was used to analyze the behavior of human beings from the view-

Rodríguez A., Grau R. and Nowé A..
CONTINUOUS ACTION REINFORCEMENT LEARNING AUTOMATA - Performance and Convergence.
DOI: 10.5220/0003287104730478
In *Proceedings of the 3rd International Conference on Agents and Artificial Intelligence* (ICAART-2011), pages 473-478
ISBN: 978-989-8425-41-6

point of psychologist and biologists (Hilgard, 1948; Hilgard and Bower, 1966).

The engineering research on LA started in the early 1960's (Tsetlin, 1961; Tsetlin, 1962). Tsetlin and his colleagues formulated the objective of learning as an optimization of some mathematical performance index (Thathachar and Sastry, 2004; Tsypkin, 1971; Tsypkin, 1973):

$$J(A) = \int_A R(a,A)\,\mathrm{d}P \tag{1}$$

where $R(a,A)$ is a function of an observation vector a with A the space of all possible actions. The performance index J is the expectation of R with respect to the distribution P. This distribution includes randomness in a and randomness in R.

The model is well presented by the following example introduced by Thathachar and Sastry (Thathachar and Sastry, 2004). Consider a student and a teacher. The student is posed a question and is given several alternative answers. The student can pick one of the alternatives, following which the teacher responds yes or no. This response is probabilistic – the teacher may say yes for wrong alternatives and vice versa. The student is expected to learn the correct alternative through such repeated interactions. While the problem is ill-posed with this generality, it becomes solvable with an added condition. It is assumed that the probability of the teacher saying 'yes' is maximum for the correct alternative.

All in all, LA are useful in applications that involve optimization of a function which is not completely known in the sense that only noise corrupted values of the function for any specific values of arguments are observable (Thathachar and Sastry, 2004). Some standard implementations are introduced below

2.1 Learning Automata Implementations

The first implementation we would like to refer to is the Continuous Action Learning Automata (CALA) introduced by Thathachar and Sastry in 2004 (Thathachar and Sastry, 2004). The authors implemented P as the Normal Probability Distribution with mean μ_t and standard deviation σ_t. At every time step t an action is selected according to a normal distribution $N(\mu_t,\sigma_t)$. Then, after exploring action a_t and observing signal $\beta_t(a_t)$, the update rules (2) and (3) are applied resulting in a new value for μ_{t+1} and σ_{t+1}.

$$\mu_{t+1} = \mu_t + \lambda \frac{\beta_t(a_t) - \beta_t(\mu_t)}{\max(\sigma_t,\sigma_L)} \frac{a_t - \mu_t}{\max(\sigma_t,\sigma_L)} \tag{2}$$

$$\sigma_{t+1} = \sigma_t + \lambda \frac{\beta_t(a_t) - \beta_t(\mu_t)}{\max(\sigma_t,\sigma_L)} \left[\left(\frac{a_t - \mu_t}{\max(\sigma_t,\sigma_L)} \right)^2 - 1 \right] - \lambda K(\sigma_t - \sigma_L) \tag{3}$$

where λ is the learning parameter controlling the step size ($0 < \lambda < 1$), K is a large positive constant and σ_L is a lower bound of σ. Authors also introduced the convergence proof for this automaton and tested it in games with multiple learners.

Notice that this first formulation presented in expressions (2) and (3) works with a parametric Probability Density Function (PDF) so it is simple and fast to incorporate the signal into the knowledge of the automaton. Thathachar and Sastry (Thathachar and Sastry, 2004) introduced several examples of how to manage a game of multiple automata meaning that these automata can be used for controlling multiple variables in a MAS. Notice that the update rule needs information about the response of the environment for the selected action a_t but it also needs the feedback for the action which corresponds to the mean of the probability distribution, being μ_t. In most of practical engineering problems it is impossible to explore both actions. Additionally, the convergence proof assumes that the function to optimize should be integrable and the minimal achievable standard deviation σ_L is very sensitive to noise: the stronger the noise, the higher the lower bound σ_L. These constraints are really restrictive for practical applications.

The second implementation we would like to recall is the CARLA (Howell et al., 1997). The authors implemented P as a PDF as well but nonparametric this time. Starting with the uniform distribution over the whole action space A and after exploring action $a_t \in A$ in time step t the PDF is updated as (4) shows.

$$f_{t+1}(a) = \begin{cases} \gamma_t \left(f_t(a) + \beta_t(a_t)\,\alpha e^{-\frac{1}{2}\left(\frac{a-a_t}{\lambda}\right)^2} \right) & a \in A \\ 0 & a \notin A \end{cases} \tag{4}$$

This second formulation (4) saves the unnecessary exploration and the function to optimize is not required to be integrable, just not chaotic. The problem is that it controls the strategy for the action selection of the automaton with a nonparametric PDF so it becomes computational very expensive. The solution is to numerically approximate the function but still, some heavy numerical calculations are necessary for γ_t. No convergence proof is given either.

If the computational cost of this method could be decreased and the convergence proof shown, then the CALA introduced by Thathachar and Sastry could be substituted by the CARLA providing a better way for solving practical problems with a MAS approach.

2.2 CARLA Update Rule

Let us restart the analysis on expression (4) in order to look for possible reductions on the computational cost. Let $a_- = \min(A)$ and $a_+ = \max(A)$ be the minimum and maximum possible actions. The normalization factor γ_t can be computed by expression (5).

$$\gamma_t = \frac{1}{\int_{a_-}^{a_+} \left(f_t(a) + \beta_t(a_t)\, \alpha e^{-\frac{1}{2}\left(\frac{a-a_t}{\lambda}\right)^2} \right) \mathrm{d}a} \tag{5}$$

The original formulation is for the bounded continuous action space A which makes the analytical calculation of the normalisation factor γ unlikely. Let us relax this constraint and work over the unbounded continuous action space $\Re$. Then the PDF update rule introduced in (4) should be redefined as (6) where $f_{N(a_t,\lambda)}$ is the normal PDF with mean a_t and standard deviation λ. Analogously, (5) is transformed into (7). Notice that numerical integration is no longer needed for calculating γ_t.

$$\begin{aligned} f_{t+1}(a) &= \gamma_t \left(f_t(a) + \beta_t(a_t)\, \alpha e^{-\frac{1}{2}\left(\frac{a-a_t}{\lambda}\right)^2} \right) \\ &= \gamma_t \left(f_t(a) + \beta_t(a_t)\, \alpha\lambda\sqrt{2\pi} f_{N(a_t,\lambda)}(a) \right) \end{aligned} \tag{6}$$

$$\begin{aligned} \gamma_t &= \frac{1}{\int_{-\infty}^{+\infty} \left(f_t(a) + \beta_t(a_t)\, \alpha\lambda\sqrt{2\pi} f_{N(a_t,\lambda)} \right) \mathrm{d}a} \\ &= \frac{1}{1 + \beta_t(a_t)\, \alpha\lambda\sqrt{2\pi}} \end{aligned} \tag{7}$$

Let δ_t, introduced in (8), be the extra area added to the PDF by stretching the curve beyond the interval $[a_-, a_+]$ then (7), can be written as (9) and (6) as (10).

$$\delta_t = \beta_t(a_t)\, \alpha\lambda\sqrt{2\pi} \tag{8}$$

$$\gamma_t = \frac{1}{1+\delta_t} \tag{9}$$

$$f_{t+1}(a) = \gamma_t \left(f_t(a) + \delta_t f_{N(a_t,\lambda)}(a) \right) \tag{10}$$

In order to generate the actions following the policy f_t the cumulative density function (CDF) is needed (Parzen, 1960) which is introduced in (11).

$$\begin{aligned} F_{t+1}(a) &= \int_{-\infty}^{a} f_{t+1}(z)\, \mathrm{d}z \\ &= \int_{-\infty}^{a} \gamma_t \left(f_t(z) + \delta_t f_{N(a_t,\lambda)}(z) \right) \mathrm{d}z \\ &= \gamma_t \left(F_t(x) + \delta_t F_{N(a_t,\lambda)}(a) \right) \\ &= \gamma_t \left(F_t(x) + \delta_t F_{N(0,1)} \left(\frac{a-a_t}{\lambda} \right) \right) \end{aligned} \tag{11}$$

Although there is no analytical definition for the normal CDF $F_{N(\mu,\sigma)}$ it can be approximated by means of numerical integration. So still numerical integration is needed however one single CDF is required, being $F_{N(0,1)}$ which can be calculated at the beginning of learning – only once – and there is no more need for integration during the learning process.

Finally, the original constraint has to be met for practical solutions, that is $\forall t : a_t \in A$ – see (4). So γ_t and F_{t+1} defined in (9) and (11) should be transformed as shown in (12) and (13) where $F_t^{diff}(x,y) = F_{N(0,1)}\left(\frac{y-a_t}{\lambda}\right) - F_{N(0,1)}\left(\frac{x-a_t}{\lambda}\right)$.

$$\gamma_t = \frac{1}{1 + \delta_t F_t^{diff}(a_-, a_+)} \tag{12}$$

$$F_{t+1}(a) = \begin{cases} 0 & a < a_- \\ \gamma_t \left(F_t(a) + \delta_t F_t^{diff}(a_-, a_t) \right) & a \in A \\ 1 & a > a_+ \end{cases} \tag{13}$$

For a practical implementation of this method, equations (8), (12) and (13) are sufficient to avoid numerical integration saving lot of calculation time during learning process. We would like to stress that without this reformulation the method was really computationally too heavy to be applied in practice, but with this change it turns to be computationally feasible. In the next subsection we will perform an analysis of λ used in expression (8) which will result in better convergence properties.

2.3 CARLA Convergence

The analysis will be performed for normalized reward signals $\beta : \Re \to [0,1]$ – no generality is lost because any closed interval can mapped to this interval by a linear transformation. The final goal of this analysis is to find the necessary restrictions to guarantee convergence to local optima.

The sequence of PDF updates is a Markovian process, where for each time-step t an action $a_t \in A$ is selected and a new f_t is returned. At each time-step t, f_t will be updated as shown in expression (10). The expected value $\bar{f}_{t+1}$ of f_{t+1} can be computed following equation (14).

$$\bar{f}_{t+1}(a) = \int_{-\infty}^{+\infty} f_t(z)\, f_{t+1}(a \mid a_t = z)\, dz \tag{14}$$

Let $\gamma_{t_z} = \gamma_t | a_t = z$ be the value for γ_t if $a_t = z$ and $\bar{\gamma}_t$ the expected value of γ_t then (14) could be rewritten

as (15).

$$\begin{aligned}\bar{f}_{t+1}(a) &= \int_{-\infty}^{+\infty} f_t(z)\gamma_{t_z}\left(f_t(a)+\alpha\beta_t(z)e^{-\frac{1}{2}\left(\frac{a-z}{\lambda}\right)^2}\right)dz \\ &= f_t(a)\bar{\gamma}_t+\alpha\int_{-\infty}^{+\infty} f_t(z)\gamma_{t_z}(z)\beta_t(z)e^{-\frac{1}{2}\left(\frac{a-z}{\lambda}\right)^2}dz\end{aligned} \tag{15}$$

Let us have a look at the right member of the integral. $f_t(z)$ is multiplied by the factor composed by the normalization factor given that $a_t = z$, the feedback signal $\beta_t(z)$ and the distance measure $e^{-\frac{1}{2}\left(\frac{a-z}{\lambda}\right)^2}$ which can be interpreted as the strength of the relation of actions a and z, the higher the value of this product, the bigger the relation of these actions. Let us call this composed factor $G_t(a,z)$ and $\bar{G}_t(a)$ its expected value at time-step t with respect to z. Then equation (15) could be finally formulated as (16).

$$\begin{aligned}\bar{f}_{t+1}(a) &= f_t(a)\bar{\gamma}_t+\alpha\bar{G}_t(a) \\ &= f_t(a)\left(\bar{\gamma}_t+\frac{\alpha\bar{G}_t(a)}{f_t(a)}\right)\end{aligned} \tag{16}$$

The sign of the first derivative of f_t depends on the factor $\bar{\gamma}_t+\frac{\alpha\bar{G}_t(a)}{f_t(a)}$ of expression (16) so it behaves as shown in (17).

$$\frac{\partial f_t}{\partial t}\begin{cases}<0 & \left(\bar{\gamma}_t+\frac{\alpha\bar{G}_t(a)}{f_t(a)}\right)<1\\ =0 & \left(\bar{\gamma}_t+\frac{\alpha G_t(a)}{f_t(a)}\right)=1\\ >0 & \left(\bar{\gamma}_t+\frac{\alpha G_t(a)}{f_t(a)}\right)>1\end{cases} \tag{17}$$

Notice $\bar{\gamma}_t$ is a constant for all $a \in A$ and $\int_{-\infty}^{+\infty} f_t(z)\,dz = 1$ so:

$$\begin{gathered}\exists_{b_1,b_2\in A}:\frac{\bar{G}_t(b_1)}{f_t(b_1)}\neq\frac{\bar{G}_t(b_2)}{f_t(b_2)}\Longrightarrow \\ \exists_{A^+,A^-\subset A,A^+\cap A^-=\emptyset},\forall_{a^+\in A^+,a^-\in A^-}: \\ \left(\frac{\partial f_t(a^+)}{\partial t}>0\right)\wedge\left(\frac{\partial f_t(a^-)}{\partial t}<0\right)\end{gathered} \tag{18}$$

From logical implication (18) it can be assured that the sign of $\frac{\partial f_t(a)}{\partial t}$ will be determined by the ratio $\frac{\bar{G}_t(a)}{f_t(a)}$. Notice subsets A^+ and A^- are composed by the elements of A that have not reached their value for the probability density function in equilibrium with $\bar{G}_t(a)$. That is, the A^+ subset is composed by all $a \in A$ having a value of probability density function which is too small with respect to $\bar{G}_t(a)$ and vice versa for A^-.

Let $a^* \in A$ be the action that yields the highest value for $\int_{-\infty}^{+\infty}\beta_t(z)e^{-\frac{1}{2}\left(\frac{a-z}{\lambda}\right)^2}dz$ for all time-steps as shown in (19). It is important to stress that a^* is not the optimum of β_t but the point yielding the optimal vicinity around it and defined by $e^{-\frac{1}{2}\left(\frac{a^*-z}{\lambda}\right)^2}$ which depends on λ.

$$\forall_{t\in\Re,a\in A}:\int_{-\infty}^{+\infty}\beta_t(z)e^{-\frac{1}{2}\left(\frac{a^*-z}{\lambda}\right)^2}dz\geq\int_{-\infty}^{+\infty}\beta_t(z)e^{-\frac{1}{2}\left(\frac{a-z}{\lambda}\right)^2}dz \tag{19}$$

It is a fact that $\forall_{a\in A}:\bar{G}_t(a)\leq\bar{G}_t(a^*)$ and since the first derivative depends on $\frac{\bar{G}_t(a)}{f_t(a)}$, the value of $f_t(a^*)$ necessary for keeping $\frac{\partial f_t(a^*)}{\partial t}=0$ is also higher than any other $f_t(a)$:

$$\forall_{a\in A}: f_t(a)\geq f_t(a^*)\Rightarrow\frac{\partial f_t(a)}{\partial t}<\frac{\partial f_t(a^*)}{\partial t} \tag{20}$$

Notice that the maximum update that $f_t(a^*)$ may receive is obtained when $a_t = a^*$ – centering the bell at a^* –, then if $f_t(a^*)$ reaches the value $\frac{1}{\lambda\sqrt{2\pi}}$, its first derivative will not be higher than 0 as shows (21) since $\beta:\Re\rightarrow[0,1]$.

$$\begin{aligned}f_{t+1}(a^*) &= \gamma_t\left(f_t(a)+\beta_t(a_t)\alpha e^{-\frac{1}{2}\left(\frac{a-a_t}{\lambda}\right)^2}\right)\\ &\leq\frac{1}{1+\alpha\lambda\sqrt{2\pi}}\left(\frac{1}{\lambda\sqrt{2\pi}}+\alpha\right)\\ &\leq\frac{1}{1+\alpha\lambda\sqrt{2\pi}}\left(\frac{1+\alpha\lambda\sqrt{2\pi}}{\lambda\sqrt{2\pi}}\right)\\ &\leq\frac{1}{\lambda\sqrt{2\pi}}\end{aligned} \tag{21}$$

Then the equilibrium point of $f_t(a^*)$ has the higher bound $\frac{1}{\lambda\sqrt{2\pi}}$. Notice that the closer the $\beta_t(a^*)$ to 1, the closer the equilibrium point of $f_t(a^*)$ to its higher bound.

$$\frac{\partial f_t(a^*)}{\partial t}\begin{cases}<0 & f_t(a^*)>\frac{1}{\lambda\sqrt{2\pi}}\\ =0 & f_t(a^*)=\frac{1}{\lambda\sqrt{2\pi}}\\ >0 & f_t(a^*)<\frac{1}{\lambda\sqrt{2\pi}}\end{cases} \tag{22}$$

We can conclude from (20) and (22) that the highest value for f will be achieved at a^* as shown in (23) which has the higher bound $\frac{1}{\lambda\sqrt{2\pi}}$.

$$\begin{aligned}\forall_{a\in A\setminus\{a^*\}}: lim_{t\rightarrow\infty}f_t(a) &< f_t(a^*)\\ lim_{t\rightarrow\infty}f_t(a^*) &\leq\frac{1}{\lambda\sqrt{2\pi}}\end{aligned} \tag{23}$$

Finally

$$\begin{aligned}lim_{\lambda\downarrow 0}lim_{t\rightarrow\infty}f_t(a^*)&=\infty\\ \forall_{a\neq a^*}: lim_{\lambda\downarrow 0}lim_{t\rightarrow\infty}f_t(a)&=0\end{aligned} \tag{24}$$

This analysis has been developed under really restrictive assumptions, such as $t\rightarrow\infty$, $\lambda\downarrow 0$, α is

small enough – the bigger the α the bigger the first derivative of probability law through time allowing a fast convergence but also the bigger the difference between the actual probability law and its expected value – and the reward function is noiseless enough to assure (19).

The best solution for the problem stated above about the constrains of λ is to start with a wide enough bell – allowing enough exploration – and make it thinner as it approaches the optimum – to meet (24). A good measure of convergence could be the standard deviation of the actions selected lately. When the standard deviation of actions is close to the λ that is been used to update the probability density function then the maximum value for $f_t(a^*)$ has been reached as stated in (23).

Since f_0 is the Uniform Density Function, the standard deviation of actions should start at $\sigma_0 = \frac{max(A)-min(A)}{\sqrt{12}}$. We are proposing to use expression (25) for the convergence ($conv_t$) value of the method given the standard deviation of actions σ_t. Then, (26) could be used as λ necessary in equation (8) for each time-step t improving the learning process of the automaton.

$$conv_t = 1 - \frac{\sqrt{12}\sigma_t}{\max(A) - \min(A)} \tag{25}$$

$$\lambda_t = \lambda(1 - conv_t) \tag{26}$$

3 EXPERIMENTAL RESULTS

In order to validate these ideas the standard method will be tested against the new proposal in 2 scenarios: noiseless and noisy. All examples will be introduced by the characteristic function form (von Neumann and Morgenstern, 1944). Formally, a characteristic function form game is given as a pair (N,v), where N denotes a set of players and $v : S^N \longrightarrow \Re$ is a characteristic function with $S \subset \Re$ being the action space.

3.1 Noiseless Scenarios

Three examples will be introduced in this subsection $(\{la\},\beta^i)$,$(\{la\},\beta^{ii})$ and $(\{la\},\beta^{iii})$ where la is a learning automaton. Their analytical expressions are presented in (27), (28) and (29) respectively. Figure 1 shows them graphically. The operator union is defined as $a \bigcup b = a + b - ab$ and the bell function as $Bell(a, a_0, \sigma) = e^{-\frac{1}{2}\left(\frac{a-a_0}{\sigma}\right)^2}$

$$\beta^i(a_t) = Bell(a_t, 0.5, 0.2) \tag{27}$$

$$\beta^{ii}(a_t) = 0.8 Bell(a_t, 0.2, 1) \tag{28}$$

$$\beta^{iii}(a_t) = (0.9 Bell(a_t, 0.2, 0.4)) \bigcup Bell(a_t, 0.9, 0.3) \tag{29}$$

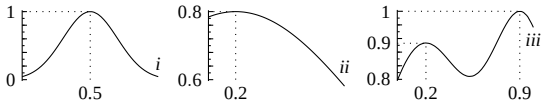

Figure 1: Characteristic function for scenarios *i*, *ii* and *iii*.

Figure 2 shows the average reward obtained over time. The selected learning rate was 0.1 and $\lambda_0 = 0.2$. The gray curve, shows the rewards collected with the standard method and the black one shows the same but using the convergence to tune the bell through time. It is clear that the results obtained with the improvement show better convergence properties. These differences are more remarked for the first scenario which has a very easy to learn function. The differences for the other two scenarios are not so big.

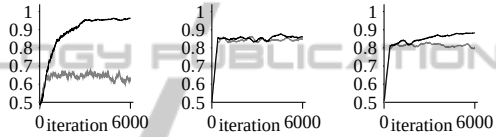

Figure 2: Average rewards.

3.2 Noisy Scenarios

We can add some random noise to the previous formulations as (30), (31) and (32) shows. Figure 3 plots these functions.

$$\beta^{i'}(a_t) = 0.8\beta^i(a_t) + rand(0.2) \tag{30}$$

$$\beta^{ii'}(a_t) = 0.875\beta^{ii}(a_t) + rand(0.2) \tag{31}$$

$$\beta^{iii'}(a_t) = 0.8\beta^{iii}(a_t) + rand(0.2) \tag{32}$$

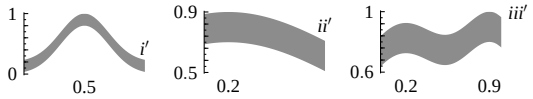

Figure 3: Reward functions for scenarios i', ii' and iii'.

Figure 4 shows the average rewards collected over time. The same parameter setting was used here. The results obtained here are similar to the ones of the previous subsection.

Table 1 sums up results for 100 runs of the algorithms for the above mentioned examples. Better results are observed for the new learner since the improved method reduces λ through the learning process. In case of environments with a high noise level,

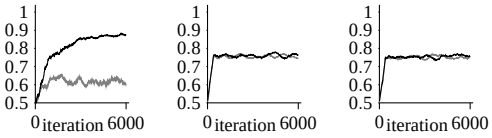

Figure 4: Average rewards.

λ cannot be reduced that much, and both methods give similar results.

Table 1: Average long-run reward

		standard	improved
noiseless	*i*	0.62	0.96
	ii	0.85	0.86
	iii	0.80	0.88
noisy	i'	0.60	0.87
	ii'	0.74	0.76
	iii'	0.74	0.75

A final remark. Notice that the standard automaton did not show good convergence properties for none of the scenarios introduced in this paper. Despite the new derivation of the method to reduce the λ as the learner reaches better convergence levels for more difficult functions such as $\left(\{la\}, \beta^{ii'}\right)$ and $\left(\{la\}, \beta^{iii'}\right)$ – where the difference in the signal received for actions around the optimum are quite similar or there are multiple optima – the learner does not converge to the optimum as fast as necessary. Future work should be focussed on the amplification of the perception of the learner of the signals to allow a more accurate convergence to the optimum.

4 CONCLUSIONS

Learning automata are reinforcement learners, belonging to the category of policy iterators, that exhibit nice convergence properties in discrete action settings. In this paper an improve of the performance of the method was proposed in order to avoid unnecessary numerical integration – speeding up the calculations – as well as the proof for the local convergence and a way to adjust the λ parameter during learning to speed up the learning itself.

In future work we want to investigate the convergence of these LA in multi-agent settings. It has been shown that a set of agents applying independently from each other an LA update scheme can converge to a Nash equilibrium in a discrete action games. We will study if this convergence result can be extended to continuous action games.

ACKNOWLEDGEMENTS

This paper was developed under the Cuba-Flanders collaboration within the VLIR project.

REFERENCES

Bush, R. and Mosteller, F. (1955). *Stochastic Models for Learning*. Wiley.

Hilgard, E. (1948). *Theories of Learning*. New York: Appleton-Century-Crofts.

Hilgard, E. and Bower, G. (1966). *Theories of Learning*. New Jersey: Prentice Hall.

Howell, M. N., Frost, G. P., Gordon, T. J., and Wu, Q. H. (1997). Continuous action reinforcement learning applied to vehicle suspension control. *Mechatronics*, 7(3):263 – 276.

Parzen, E. (1960). *Modern Probability Theory And Its Applications*. Wiley-interscience, wiley classics edition edition.

Thathachar, M. A. L. and Sastry, P. S. (2004). *Networks of Learning Automata: Techniques for Online Stochastic Optimization*. Kluwer Academic Publishers.

Tsetlin, M. (1961). The behavior of finite automata in random media. *Avtomatika i Telemekhanika*, pages 1345–1354.

Tsetlin, M. (1962). The behavior of finite automata in random media. *Avtomatika i Telemekhanika*, pages 1210–1219.

Tsypkin, Y. Z. (1971). *Adaptation and learning in automatic systems*. New York: Academic Press.

Tsypkin, Y. Z. (1973). *Foundations of the theory of learning systems*. New York: Academic Press.

von Neumann, J. and Morgenstern, O. (1944). Theory of games and economic behavior.