Discovering Sensing Capability in Multi-Agent Systems

Cristina Parpaglione
Departamento de Ingeniera Informática
Instituto Tecnológico de Buenos Aires (ITBA)
Buenos Aires, Argentina
cparpag@itba.edu.ar

Juan Miguel Santos
Departamento de Ingeniera Informática
Instituto Tecnológico de Buenos Aires (ITBA)
Buenos Aires, Argentina
jsantos@itba.edu.ar

***Abstract*—What should be the sensing capabilities of agents in a Multi-Agent System be to solve a problem efficiently, quickly and economicly? This question often appears when trying to solve a problem using Multi-Agent Systems. This paper introduces a method to find these sensing capabilities in order to solve a given problem. To achieve this, the sensing capability of an agent is modeled by a parametrized function and then Genetic Algorithms are used to find the parameters' values. The individual behavior of the agents are found with Reinforcement Learning.**

***Keywords*-Multi-Agent Systems; Genetic Algorithms; Sensing Capability; Reinforcement Learning; Sensing Parametrization;**

I. Introduction

In previous works [1], a method was proposed to find the individual behaviors of a set of N agents to solve the problem of grouping them anywhere in a toroid. In that study, knowing what the sensing capability of each agent should be was a difficulty in itself. This was accomplished by trial and error, using the situations that the agents could not solve as feedback. For example, the sensing capability had to be modified to avoid symmetric situations that led to cyclic behaviors, and also to avoid obstruction between agents whose movements interfered with each other. After each change, a new learning experience was made and new conclusions were reached. However, when the grouping criterion was changed (e.g. agents should group in a well defined geometric shape) the sensing capability previously found was not enough. The perceptual aliasing (due to limited sensing capability [4]) made it impossible to distinguish certain states that were predecessors to the goal from other states that did not belong to the problem's solution.

Finding the best sensing by trial and error is extremely tedious and time consuming. Therefore, it was decided to tackle the sensing problem by proposing a method based on parametrization.

In Section II the proposed method is presented. Section III describes two experiments carried out for testing the method. Section IV presents the conclusions.

II. Method

The method introduced here proposes to model the sensing capability of each agent by using a parametrized sensing function. The parameter values are determined using an optimization technique such as Genetic Algorithms. Characteristics of several sensors such as infrared proximity detectors, ultrasonic sensors, scanner and video camera, were taken into consideration when the parameters were defined. For example, an ultrasonic sensor has limited scope but can measure the distance to an object, while a CCD camera has unlimited scope but gives no information about the distance of the objects in front of it.

Five parameters are used to model the main characteristics of several sensors:

- **Scope (ρ)**: denoting the radius of the sensing area. Integer value in $[1, \max(height, width)]$,
- **Aperture (σ)**: denoting the sensing aperture angle. Integer value in $[0, \rho]$,
- **Closeness (β)**: denoting the number of layers in which the sensing is divided. The layers are numbered, 1 being the layer nearest to the agent and so on. Integer value in $[1, \rho]$,
- **Density(δ)**: denoting how to measure the quantity of agents in each layer. There is a value for the first layer (δ_1) and another for the rest of the layers (δ_r). Integer value in $[1, N]$,
- **Expansion (ν)**: denoting the shape of the sensing area. Integer value in $[-\rho + 1, \rho]$,

where $height$ and $width$ are the toroid arena sizes and N is the number of agents involved in the problem. The shape of the sensing area is determined by the values of σ, ρ and ν. Figure 1 shows, for one sensor, several examples for different values of these three parameters.

The range of the value of each parameter can depend on the values of some of the other parameters. For example, the maximum value of β depends on ρ. The number of possible configurations of the agent is determined as a function of β, δ_1 and δ_r for each sensor (front, right, back, left), where a configuration is the state representation given by the sensors of the agent.

Table I shows the number of rows in each layer depending

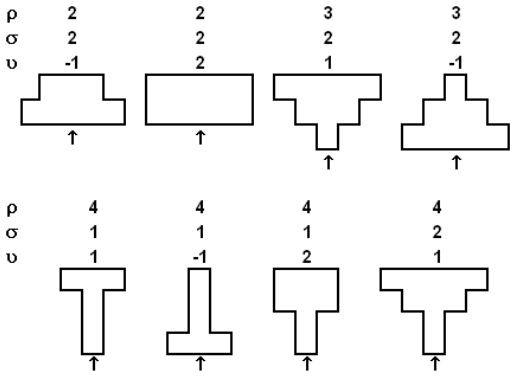

Figure 1. Scope, Aperture and Expansion Examples. The arrow (↑) represents the agent. The arrowhead points to the front sensor.

on the value of β and ρ. For example, if $\beta = 3$ and $\rho = 4$, the first layer has 1 row, the second layer has 1 row and the third layer has the remaining 2 rows. Figure 2 shows four possible values for β.

Table I
NUMBER OF LAYERS AND LAYER ROWS.

	ρ				
β	**1**	**2**	**3**	**4**	...
1	1	2	3	4	...
2	-	1,1	1,2	1,3	...
3	-	-	1,1,1	1,1,2	...
4	-	-	-	1,1,1,1	...

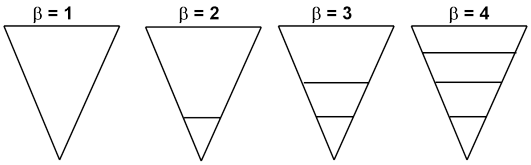

Figure 2. Closeness Examples.

Regarding the density of the first layer, the nearest layer to the agent, it has more varieties in values than the remaining layers. Table II shows the set of possible values for the first layer. Remaining layers only have two values as shown in Table III. In both tables column *Meaning* shows the different possibilities for the parameter δ.

Figure 3 shows a parametrization example for a sensor and Figure 4 shows a parametrization example of an agent with four sensors.

The state of each agent is represented by a 5-tuple (F, R, B, L, G), where F, R, B and L are the output values of the Front, Right, Back and Left sensors. Position G represents the grouping of all agents in any place in the toroid. When an agent arrives at some position where there are other agents, they start to interchange information about

Table II
DENSITY FOR FIRST LAYER.

$\delta_1 = i$	
Value	**Meaning**
0	No agents.
1	At least 1 agent on either side.
2	At least 2 agents on either side.
3	...
$i-1$	At least $i-1$ agents on either side.
i	1 agent in front.
$i+1$	1 agent in front and at least 1 agent on either side.
$2i-2$	1 agent in front and at least $i-2$ agents on either side.
$2i-1$	1 agent in front and at least $i-1$ agents on either side.

Table III
DENSITY FOR THE REMAINING LAYERS.

$\delta_r = 1$	
Value	**Meaning**
0	No agents.
1	At least 1 agent.

$\delta_r = 2$	
Value	**Meaning**
0	No agents.
1	At least half of the positions with agents.
2	More than half of the positions with agents.
3	All positions with agents.

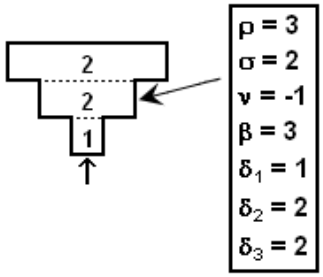

Figure 3. Sensor Parametrization. The arrow (↑) represents the agent. The arrowhead points to the front sensor.

whether they are seeing other agents or not. If all agents are together, then the component G is set to 1, otherwise it is 0. The remaining elements of the 5-tuple are calculated as follows. Being $s_F(layer)$, $s_R(layer)$, $s_B(layer)$ and $s_L(layer)$ the given values of front, right, back and left sensors, respectively, for each layer of an agent in particular. Let $\boldsymbol{s_F}$, $\boldsymbol{s_R}$, $\boldsymbol{s_B}$ and $\boldsymbol{s_L}$ be the corresponding arrays for each sensor in each layer, and $\boldsymbol{\delta_F}$, $\boldsymbol{\delta_R}$, $\boldsymbol{\delta_B}$ and $\boldsymbol{\delta_L}$ the density arrays for each layer for each sensor. Then, the sensor output value for this agent is calculated as follows:

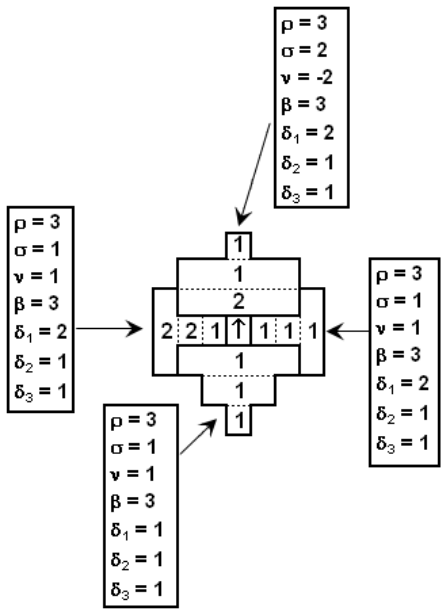

Figure 4. Four Sensor Parametrization. The arrow (↑) represents the agent. The arrowhead points to the front sensor.

$$F(\beta, \boldsymbol{\delta_F}, \boldsymbol{s_F}) = s_F(\beta) + \sum_{i=1}^{\beta-1} 2^{\beta-i} * s_F(i) * \prod_{j=i+1}^{\beta} \delta_F(j), \quad (1)$$

$$R(\beta, \boldsymbol{\delta_R}, \boldsymbol{s_R}) = s_R(\beta) + \sum_{i=1}^{\beta-1} 2^{\beta-i} * s_R(i) * \prod_{j=i+1}^{\beta} \delta_R(j), \quad (2)$$

$$B(\beta, \boldsymbol{\delta_B}, \boldsymbol{s_B}) = s_B(\beta) + \sum_{i=1}^{\beta-1} 2^{\beta-i} * s_B(i) * \prod_{j=i+1}^{\beta} \delta_B(j), \quad (3)$$

$$L(\beta, \boldsymbol{\delta_L}, \boldsymbol{s_L}) = s_L(\beta) + \sum_{i=1}^{\beta-1} 2^{\beta-i} * s_L(i) * \prod_{j=i+1}^{\beta} \delta_L(j). \quad (4)$$

According to the number of possible values of each sensor, the agent can represent actual states of the environment in a number of $Conf(.)$ different configurations, given by:

$$Conf(\beta, \boldsymbol{\delta_F}, \boldsymbol{\delta_R}, \boldsymbol{\delta_B}, \boldsymbol{\delta_L}) = \\ = 2^{4\beta} \prod_{i=1}^{\beta} (\delta_F(i) * \delta_R(i) * \delta_B(i) * \delta_L(i)). \quad (5)$$

To obtain the parameter values, Genetic Algorithms [3] are used as a searching method in the parameters' space of the sensing function. This algorithm has the following requirements:

- The chromosome is composed of gene groups representing each sensor parameter values.
- Elite/Roulette is used for both, selection and replacement.
- Crossover uses one parent elected from the set of chromosomes obtained with Elite and the other parent chosen from the set obtained with Roulette.
- Q-learning, a Reinforcement Learning technique, is added as a genetic operator. It is used to train agents to learn (using the sensing parametrized in the chromosome) policies that allow them to solve a given problem. The number of trainings that take place with the information in the chromosome is established at the beginning of the method. The values obtained for Q during training i are not initialized in training $i+1$ for each agent in each chromosome.
- The fitness function used is the result of testing individual policies obtained by Q-learning, starting at different initial states each time. The testing process is carried out using the sensing capability represented in a chromosome That is, the agent uses the sensing capability expressed in the chromosome and the policy obtained during training (described above). The testing process counts as successful each time the grouping problem is solved. This process is performed a predetermined number of times, all with the same number of iterations. The fitness value is the percentage of times that agents remain grouped.

It is necessary to know the ceiling values used during mutation. Max Density for layer is defined with the values of ρ, σ, ν and ρ and it is calculated by:

$$\delta(l) = 1 + 2 \sum_{i=l}^{f(l)+l-1} ap(i) \qquad \forall 1 \leq l \leq \beta, \quad (6)$$

where l is the layer number, $\delta(l)$ is the ceiling density in the layer l, (consider that if $l = 1 \Rightarrow \delta(l) = \delta_1$ and if $l > 1 \Rightarrow \delta(l) = \delta_r$). Function $f(l)$ used in (6) determines the number of rows in each layer and can be calculated by:

$$f(l) = \begin{cases} 1 & l \neq \beta \\ \rho - \beta + 1 & l = \beta, \end{cases} \quad (7)$$

$ap(i)$ is the Aperture in each row and is calculated by:

$$ap(i) = \begin{cases} \sigma & Cond_1 = True \\ 0 & Cond_2 = True \\ ap(i + sgn(\nu)) - 1 & otherwise, \end{cases} \quad (8)$$

where

$$Cond_1 = \begin{cases} True & i = g(sgn(\nu)) \vee \\ & \vee\ sgn(|\nu| - \rho + i * sgn(\nu)) = sgn(\nu) \\ False & otherwise, \end{cases} \tag{9}$$

$$Cond_2 = \begin{cases} True & i \neq g(sgn(\nu)) \wedge \\ & \wedge\ ap(i + sgn(\nu)) - 1 < 0 \\ False & otherwise, \end{cases} \tag{10}$$

Table III shows that if $l \geq 2 \wedge \delta(l) > 2 \Rightarrow \delta(l) = 2$.

Function $g(x)$ used in (9) and (10) determines how to calculate the aperture and can be obtained by:

$$g(x) = \frac{\rho - 1}{2} * x + \frac{\rho + 1}{2}. \tag{11}$$

III. Experiments and Results

Two different experiments were conducted to validate the proposed method: 1) find the sensing capability to solve the grouping problem and compare it with a known one (explained in section III-A); 2) find the appropriate sensing capability when the agents must group in a particular geometric shape. The aim of this experiment was to force policy differentiation (explained in section III-B).

In both cases the fitness value was reached by testing the obtained policy 300 times after 7 trainings with 1400 iterations each [1]. The population was composed of 25 individuals. In both experiments, each agent had four sensors with three genes each. The three genes represented: β, δ_1 and δ_r, respectively. Once the optimal set of parameters has been obtained, the policy is improved with an extra training. For this to happen, the agents were trained 15 times with 11200 iterations.

A. Experiment 1

In this experiment the proposed method was tested by finding a sensing capability to solve the grouping problem and comparing it with that obtained by trial and error. The agents were indistinguishable, so it is not necessary to learn different policies.

Parameters ρ, σ and ν were set in an analogous way for each sensor with the values shown in Table IV.

Table IV
Fixed parameter values used in experiment 1.

	ρ	σ	ν
Front	2	1	2
Right	2	1	1
Back	2	1	2
Left	2	1	1

Table V
Fixed parameter values used in experiment 2.

	ρ	σ	ν
Front	4	1	-1
Right	2	1	1
Back	2	1	2
Left	2	1	1

Table VI shows the values for β, δ_1 y δ_r for sensing capability found by trial and error and Table VII shows the values of the same parameters but for the sensing capability found with the method. Effectiveness with both sets of parameters was 100%.

Table VI
Parameter values for β, δ_1 and δ_r obtained by trial and error.

	β	δ_1	δ_r
Front	2	2	1
Right	2	1	1
Back	1	1	1
Left	2	1	1

Table VII
Parameter values for β, δ_1 and δ_r obtained with the proposed method.

	β	δ_1	δ_r
Front	2	2	1
Right	1	3	1
Back	1	1	1
Left	2	1	1

Notice the difference between the number of configurations for the sensing capability obtained by trial and error, which was 1024, versus the number of possible configurations for the sensing capability obtained using the proposed method, which was only 768. A considerable decrease can be observed in the quantity of configurations without affecting the problem resolution.

B. Experiment 2

In this experiment the agents must group in a square shape, which is shown in Figure 5. Position numbers and agent numbers are equivalent. Notice that the way in which each agent contributes to building a specific geometric shape of grouping (e.g. triangle, square, etc) depends on the region shape to be completed. It is necessary to distinguish two groups of different positions in the shape:

- the four corners (positions 0, 2, 6 and 8), named set of even agents, and
- the four remaining places (positions 1, 3, 5 and 7), named set of odd agents.

Parameters ρ, σ and ν are set in an analogous way for each sensor with the values shown in Table V.

It is important to notice that in this case agents should learn different policies (differentiation) and they could

Figure 5. Shape to be reached in experiment 2. Each position in the objective shape has been enumerated from 0 to 8 in consecutive order. Position 4 determines where the shape has to be formed in the toroid.

probably acquire different sensing capabilities. For this reason chromosomes have a group of 12 genes for each agent.

This experiment had two phases:

1) Obtaining the sensing capability and behavior for four agents (Agents 0, 2, 5 and 7 where used).
2) Obtaining the sensing capability and behavior for two agents (Agents 0 and 7 where used).

The method found the parameters shown in Tables VIII, IX, X, y XI, with 6% as a fitness value.

Phase 1 showed that the policies and sensing capability for agent 2 could be interchanged with the policy for agent 0. And the same happened with agents 5 and 7.

Table VIII
PARAMETER VALUES OBTAINED BY THE PROPOSED METHOD AND USED FOR TESTING DURING EXPERIMENTS FOR AGENT 0.

	β	δ_1	δ_r
Front	2	2	1
Right	2	1	1
Back	2	3	1
Left	1	2	1

Table IX
PARAMETER VALUES OBTAINED BY THE PROPOSED METHOD AND USED FOR TESTING DURING EXPERIMENTS FOR AGENT 2.

	β	δ_1	δ_r
Front	4	2	1
Right	2	1	1
Back	1	4	1
Left	2	1	1

Table X
PARAMETER VALUES OBTAINED BY THE PROPOSED METHOD AND USED FOR TESTING DURING EXPERIMENTS FOR AGENT 5.

	β	δ_1	δ_r
Front	2	2	1
Right	1	2	1
Back	2	1	2
Left	2	1	1

The next question to answer was: which were the agents that contributed to such low fitness?. To answer it, individual agents were tested. That is, eight agents were located in the toroid in such a way that they formed an incomplete square (for example, to test Agent 0, they formed a square with position 0 empty). The fitness value for each test is shown in Table XII. It is easy to see that the fitness for Agents 2 and 5 are lower than those for the other two agents.

Table XI
PARAMETER VALUES OBTAINED BY THE PROPOSED METHOD AND USED FOR TESTING DURING EXPERIMENTS FOR AGENT 7.

	β	δ_1	δ_r
Front	4	3	1
Right	2	1	2
Back	1	4	1
Left	1	2	1

Table XII
FITNESS VALUE FOR POLICIES AND SENSING CAPABILITIES FOUND WITH THE PROPOSED METHOD.

Agent	0	2	5	7
Percentage	97	63	64	92

In the goal shape, Agents 0 and 2 have equivalent positions and the same happens with Agents 5 and 7. The next step was to use policy and sensing capability from Agent 0 in Agent 2 and do the same test with Agents 5 and 7. The results showed that the fitness for Agents 2 and 5 was increased by using the policies and sensing capabilities from the agents with better performance. It can be seen in Table XIII.

Table XIII
FITNESS FOR AGENTS 2 AND 5 USING INFORMATION FROM AGENTS 0 AND 7, RESPECTIVELY.

Agent	2	5
Percentage	95	86

The remaining agents were tested using the policy and sensing capability obtained for Agents 0 and 7. Namely, the policy and sensing capability obtained for Agent 0 were used for Agents 6 and 8, whereas the information obtained for Agent 7 was used for Agents 1 and 3. It is worth mentioning that Agent 4 is used as a seed determining the place in the toroid where the shape will be taking place. For that reason it is not trained or tested. The results for each agent (1, 3, 6 and 8) are shown in Table XIV.

Table XIV
FITNESS OBTAINED FOR NOT TRAINED AGENTS.

Agent	1	3	6	8
Percentage	90	89	95	94

The underlying idea in this experiment was to obtain different behaviors for distinct agents. To verify specialization a particular test had to be done. Agent 0 was tested using the policy and sensing capability learned by

Agent 7 and vice versa. The results are shown in Table XV. The fitness value obtained by Agent 0 being place in position 7 (both positions and agents are interchangeable) has a lower effectiveness rate than if it used its own information (see Table XV). The sensing capability in Agent 0 is not enough to avoid the perceptual aliasing, decreasing the performance of the agent (see Tables VIII and XI with the sensing capabilities for each agent obtained during the learning process). On the other hand, when Agent 0 was tested using information obtained for Agent 7, the fitness value was 82%.

Table XV
FITNESS OBTAINED TESTING DIFFERENTIATION. COLUMN 0 SHOWS FITNESS FOR AGENT 0 USING INFORMATION OBTAINED FOR AGENT 7. COLUMN 7 SHOW FITNESS FOR AGENT 7 USING INFORMATION OBTAINED FOR AGENT 0.

Agent	0	7
Percentage	4	82

Table XVI
FITNESS VALUES FOR AGENTS WITHOUT LEARNING. INFORMATION FOR AGENT 0 WAS USED FOR EVEN AGENTS, AND INFORMATION FOR AGENT 7 WAS USED FOR THE REMAINING AGENTS.

Agent	1	2	3	5	6	8
Percentage	82	7	83	82	6	5

The reason why even agents had better performance than odd ones can be understood by analyzing the sensing parameters in Tables VIII, IX, X and XI. The greater the number of possible configurations, the more likely the sensing capability can be used successfully in any position of the target shape. The problem is that the greater the number of configurations, the larger the state space with its corresponding learning problems. (see [5]).

The next step was to obtain the hits percentages using policies and sensing capabilities from even agents in odd agents and vice versa. Results are shown in Table XVI.

All 8 agents were tested leaving Agent 4 docked as a seed. In that case policy and sensing capability from Agent 0 was used for even agents, and policy and sensing capability for Agent 7 was used for odd agents, giving a fitness value of 52%. The results shown in Table XVII let conclude that it is enough to train as many agents as distinguishable positions there can be found in the goal shape.

Table XVII
FITNESS FOR EACH AGENT USING INFORMATION FROM AGENT 0 AND 7.

Agent	0	1	2	3	5	6	7	8
Percentage	97	89	95	89	86	95	92	94

In this problem it is possible to distinguish two different behaviors. One for the agents who occupy the corners and another for agents who occupy the remaining positions. In that sense, the method was tested looking for behaviors only for Agents 0 and 7. The method gave two different sensing capabilities, one for Agent 0 shown in Table XVIII, and a distinct one for Agent 7, shown in Table XIX. The fitness value was 82%.

Table XVIII
PARAMETER VALUES OBTAINED FOR AGENT 0.

	β	δ_1	δ_r
Front	4	2	1
Right	1	1	1
Back	2	2	1
Left	1	1	2

Table XIX
PARAMETER VALUES OBTAINED FOR AGENT 7.

	β	δ_1	δ_r
Front	4	3	1
Right	2	1	2
Back	1	4	1
Left	1	2	1

All the experiences made with the behaviors obtained for 4 agents were repeated with the information obtained after training only Agent 0 and 7. Table XX shows the fitness value for each agent using the policy and sensing capability from Agent 0 for the even positions and using the policy and sensing capability for Agent 7 for odd positions. Remember that only Agent 0 and 7 where trained.

Table XX
FITNESS VALUES FOR EACH AGENT.

Agent	0	1	2	3	5	6	7	8
Percentage	99	92	100	86	88	98	87	99

When 8 agents were tested together the fitness value was 56%, that was greater than the fitness obtained with the information gathared in the previous case (52%).

IV. CONCLUSION

This work introduces a method that efficiently allows finding the sensing capability that agents should have in order to complete a mission. This is a subject that has not been researched in depth yet, according with the reviewed literature. The results reached show that the proposed method is efficient in obtaining a sensing capability that allows learning a solution for a given problem.

It is worth mentioning that the grouping problem could require an appropriate docking of the agents (for example, if the agents must group forming a geometric shape). This strongly depends on a precise sensing capability. Furthermore, it is clear to see that the policies obtained from the experiments are very sensitive to the region shape to be completed.

The relevance of these results lies in the fact that the obtained sensing capability is achieved in an automatic way and faster than that reached by trial and error (as in [1]).

In the first experience the relevance of the results is reflected in the fact that the parameter values were obtain after one day processing. Whereas several weeks of trial and error were needed to obtain the values used in [1], with a considerable decrease in the number of configurations.

In the second experiment, besides the detection of differentiation, it was showed that it was enough to learn only two different policies and to generalize them to the other agents in order to solve the problem.

REFERENCES

[1] Cristina Parpaglione and Juan Miguel Santos, *Obtención de Comportamientos Emergentes en Sistemas Multiagente mediante Aprendizaje por Refuerzo*, ASAI2009, 38JAIIO, 24 y 25 de Agosto de 2009, Mar del Plata, Argentina.

[2] Richard Sutton and Andrew Barto, *Reinforcement Learning: An Introduction*, MIT Press, Cambridge, Massachusetts, 1998.

[3] David E. Goldberg, *Genetic Algorithms in Search, Optimization & Machine Learning*, Addison Wesley, 1989, ISBN 0-201-15767-5.

[4] Rosana Matuk Herrera and Juan Miguel Santos, *The Clustering aliasing problem in Reinforcement Learning for robots*, Proceedings of th Fifth European Workshop on Reinforcement Learning, pp. 33-35, Utrecht, The Netherlands, 2001.

[5] Ana Julia Villar and Juan Miguel Santos, *Q-Function Kernel Smoother: a New Approach for Opened Issues in Huge State*, Anales del X Argentine Symposium on Artificial Intelligence, 24 y 25 de Agosto de 2009, Mar del Plata, Argentina.