Control and Cybernetics

vol. **44** (2015) No. 3

Optimal control of dynamic systems using a new adjoining cell mapping method with reinforcement learning*

by

Tomás Arribas Navarro[1], Sebastián Sánchez Prieto[2], Mariano Gómez Plaza[1]

[1] SOTICOL Robotics Systems, S.L., Madrid, Spain
[2] Computer Engineering Department, Universidad de Alcalá, Alcalá de Henares, Spain

Abstract: This work aims to improve and simplify the procedure used in the Control Adjoining Cell Mapping with Reinforcement Learning (CACM-RL) technique, for the tuning process of an optimal controller during the pre-learning stage (controller design), making easier the transition from a simulation environment to the real world. Common problems, encountered when working with CACM-RL, are the adjustment of the cell size and the long-term evolution error. In this sense, the main goal of the new approach, developed for CACM-RL that is proposed in this work (CACM-RL*), is to give a response to both problems for helping engineers in defining of the control solution with accuracy and stability criteria instead of cell sizes. The new approach improves the mathematical analysis techniques and reduces the engineering effort during the design phase. In order to demonstrate the behaviour of CACM-RL*, three examples are described to show its application to real problems. In all the examples, CACM-RL* improves with respect to the considered alternatives. In some cases, CACM- RL* improves the average controllability by up to 100%.

Keywords: optimal control, cell mapping, state space, reinforcement learning, stability, nonlinear control, controllability

1. Introduction

Optimal control theory is a mathematical discipline with numerous applications in both science and engineering. It deals with the problem of finding a control law for a given system such that a certain optimality criterion is fulfilled. The popularization of this discipline has also resulted in the need of using powerful computational resources and the development of new mathematical methods in order secure the implementation in the real world.

*Submitted: July 2015; Accepted: January 2016

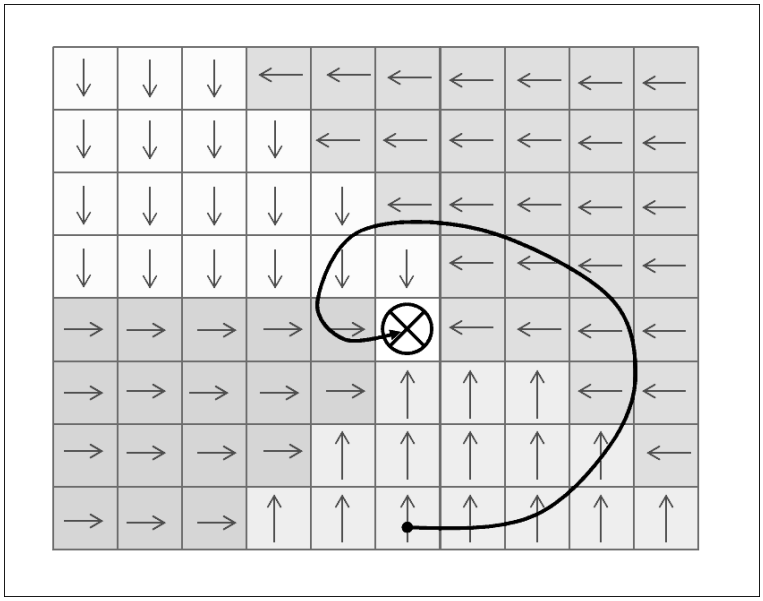


Figure 1. Trajectory that reaches a specific goal. The orientation of each arrow refers to the control action applied

In 1985, C. S. Hsu published the Cell-to-Cell methodology of global analysis for nonlinear systems (see Hsu, 1985). In this work, C. S. Hsu introduced the mathematical foundations to for translating the continuous space into the discrete cell space. These techniques have been successfully applied in numerous fields of mathematical analysis where the conventional methods have many difficulties with finding a solution.

Cell Mapping, together with advancing digital control technologies, have made possible the development of new controllers. In this sense, combining C. S. Hsu's powerful analysis techniques with dynamic programming theory and Bellman's Optimality Principle (see Barto, 1995, or Bellman, 2010), a new field of research in optimal control has been opened using Cell Mapping (CM) techniques as a baseline of control (see Zufiria, 2003, or 1993). Works such as (see Gómez, 2007, 2012, or Papa, 1995, 1997) use Cell Mapping techniques for control purposes. The basic idea is to encode one action per state in order to obtain an optimal controller using state space analysis techniques. In Moore (1995), using game theory and machine learning algorithms, Moore solves path planning and control problems, obtaining optimal solutions. To illustrate how the state-action pair can be used to solve a control problem, Fig. 1 shows a simple 2-D example.

The grid represents a system with two dimensions and four different control actions (arrows). By applying the specific control action in each state, the

system optimally reaches the goal (represented as a X). With this control model, the run time implementation is very simple and the controller complexity is focused on the algorithms in charge of the analysis and definition of the control action in each state. Therefore, the quality of the final controller will depend on the analysis done by these algorithms.

Several authors have carried out modifications and improvements oriented at the reduction of some constraints of CM analysis techniques. On the one hand, errors associated with the state space discretization could be reduced by means of Generalized Cell Mapping (GCM), which is based on a probabilistic formulation for characterizing chaotic dynamic systems (see Mo-Hong, 1993, or Wilhelmus, 1994). On the other hand, Interpolated Cell Mapping (ICM) reduces the long-term evolution errors (Bursal, 1992). In this work, the proposed CACM-RL* technique has been inspired by both, GCM and ICM.

Most of the systems that are subject to analysys in order to be optimally controlled are continuous. The discretization techniques (see Hsu, 1985) are responsible for converting the continuous state space into the discrete state space. The discretization process translates the continuous and multidimensional state space into a discrete state space of dimensions, where K is the number of dimensions that describes the state space. Cell Mapping techniques consider the central point of a cell as a reference for the analysis, and the cell size as the minimum unit of movement. In this context, it is appropriate to introduce the concept of transition. A transition is the change of system's state (cell) when applying a control action on the system during a defined period.

The discretization concept and the use of the central point of the cell as the origin of state transitions are responsible for two kinds of undesired effects, both locally and globally. Locally, the accuracy is constrained by the cell size. The resulting error can be reduced as much as desired, just reducing the cell size. The long-term evolution error is associated with the cumulative undesired effects along trajectories. In this later case, the reduction of the cell size does not reduce the error propagation.

Adjoining Cell Mapping (ACM) (Zufiria, 2003, 1993, or Guttalu, 1993), is based on the definition of an adjoining condition between cells. In general, the adjoining property opens new possibilities for developing efficient algorithms in optimal control to be applied to dynamic and non-linear systems. In this way, it contributes to the reduction of the long-term error evolution. Although techniques based on ACM optimally solve control problems, they have a high computational cost due to the reduction of cell size in order to increase the adjoining distance.

To illustrate the long-term error evolution, the trajectory AB of Fig. 2 represents the evolution of a generic trajectory in a continuous system. In a discretized controller, the problem arises because the real transitions that begin at the central point do not finish at the central point of the image cell. In Fig. 2, we can see two vectors: V_1 and V_2. They show two discrete transitions of the system's real response considering the central point of the cell as the origin. On the basis of these two vectors, a new vector, K (dashed line), is defined. It

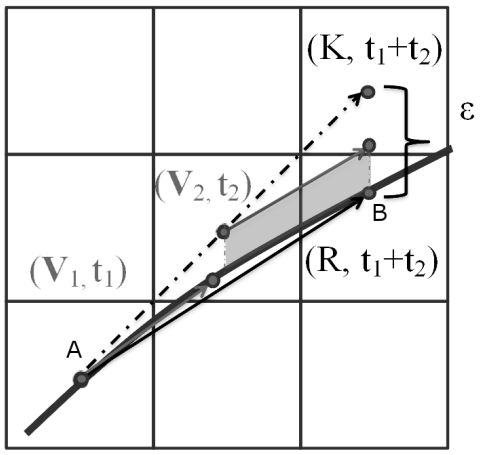


Figure 2. Long-term evolution error in the state space discretization process

represents the transition of the system in time interval t_1+t_2 when discretization is considered. However, R, represents another vector (solid line) that finishes in the trajectory line AB and, therefore, is a vector that falls in the right final state in $t_1 + t_2$. In this way, we can see the error, ϵ, which is generated in the discretization process (vector K). This error, produced by the discretization process, grows when long trajectories are covered.

One approach to solve this problem consists in adjusting the cell size with the aim of bringing the transition from (V_1, t_1) closer to the centre of the cell and thus securing that the discrete approach approximates the continuous solution. This technique can be useful when working with linear systems, but it is not very appropriate for non-linear systems. Adjusting the cell size in non-linear systems requires a lot of time, so that, a trade-off between accuracy and performance must be considered.

The goal of this work is to improve the analytic and mathematical process during the pre-learning stage (controller design) to solve the restrictions associated with the discretization process. As a result, we provide a new method of state space discretization and exploration, which is non-sensitive to the discretization and to the sample period (T_s). The method presented in this work, CACM-RL*, is fully compliant with Cell Mapping and reinforcement learning techniques (Barto, 1998, or Watkins, 1992) and helps engineers in defining the control solution with accuracy and stability criteria instead of cell sizes.

2. Relevant characteristics of CACM-RL*

CACM-RL* is a new optimal control technique based on CACM-RL (see Gómez, 2007, 2012, or 2009) which gives a solution to the issues of the discretization process errors and the long-term evolution error. In the further course of the paper, a specific example of applying CACM-RL with a DC motor is given in order to show up these errors. When using CACM-RL, it is not necessary to take into account the mathematical model of the motor since the method learns from the experience. However, in order to carry out a theoretical analysis of the arising errors, we have introduced the differential equations that define the behaviour of a DC motor (2.1). In this case, we need two state variables (angular position, X_1, and angular velocity, X_2) and therefore, a 2-D map, where the different control actions per state can be represented (see Fig. 3). The objective in this example is to reach the goal at the origin of the coordinates in minimum time from any initial state.

The equations that govern a DC motor are specified in (2.1), with exemplary time constant, $\tau = 0.6$, motor constant, $k = 0.35$ and a voltage, V that has a range between -10 and +10 volts:

$$\begin{aligned} \theta &= \omega \\ \omega' &= \tfrac{-\omega+(k\cdot v)}{\tau}. \end{aligned} \tag{1}$$

When applying CACM-RL (Gómez, 2007, 2012, or 2009) using the dynamic model specified in (1), we obtain an optimal control surface with three control actions (10v, 0v, -10v). We can define a control surface as a representation of the discretization of the state space, where each area or region is associated with a specific control action. In this case, the optimal control surface is represented in Fig. 3 in two areas (10v –left area–, -10v –right area–).

We can see in Fig. 3 how a trajectory reaches the goal (X_1=0, X_2=0) from the origin located at the lower left corner. The goal is reached as follows: first, a maximum acceleration is performed (left area) until reaching the maximum angular velocity. Before reaching the goal, an inverse maximum acceleration (right area) is applied. In all cases, the goal is reached with a null velocity and no oscillations. The two areas in the graph represent the control action value for each state (on the left side: V=10, on the right side: V=-10). The vertical axis is the angular velocity and the horizontal axis is the angular position.

The optimal control surface, obtained in Fig., 3 is a representative example with a perfectly tuned optimal controller. The discretization size used has been tuned after an iterative trial and error process, concerning the adjustment of the cell sizes of the angular velocity and the angular position. However, the more common results obtained are like those 25 surfaces shown in Fig. 4-a. Each of these surfaces is obtained with different cell sizes.

The cell size tuning process consists in the identification of anomalies detected in the optimal control surface. When detecting faults or control surfaces not well shaped, as Fig. 4-a, the cell size that is causing the problem is slightly

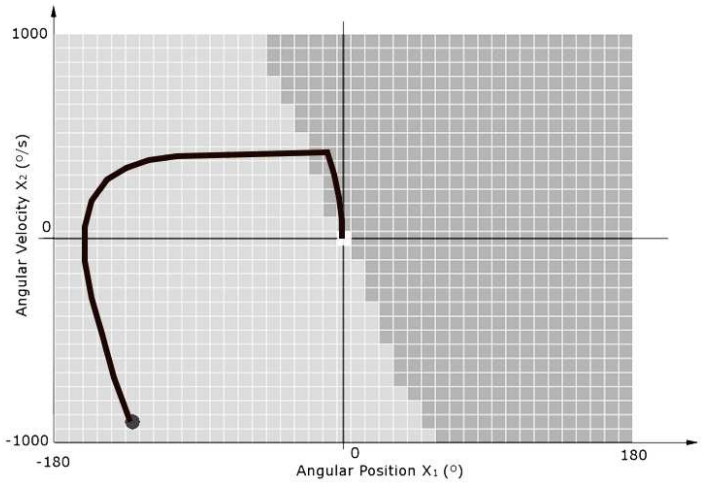


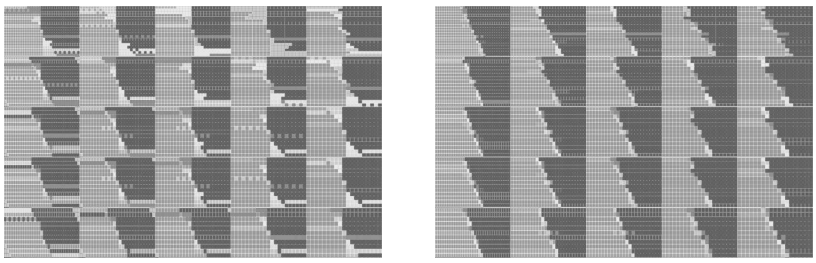
Figure 3. Example of a trajectory that goes from a specific initial state (position, velocity) to the origin of coordinates (0, 0)

adjusted in order to avoid the propagation of the anomalies to other state variables. Fig. 4-a shows a set of 25 faulty control planes. In this sense, we can see the differences with respect to the single control plane shown in Fig. 3.

The different optimal control surfaces of Fig. 4-a show several imperfections due to the long-term evolution error, illustrated in Fig. 2. From the Optimality Principle (see Bellman, 2010), when transiting to the same image cell from the same initial cell for different control actions, the fastest transition is chosen. Choosing the fastest control action is not equivalent obtaining the greatest reward. Therefore, it is important to know the system position inside the image cell because otherwise, some bands and patterns, such as the ones contained in some surfaces in Fig. 4-a, could reduce the controllability and performance. Furthermore, the cell size affects the softness of the frontier regions.

The errors and imperfections, mentioned above, can be solved by means of the implementation of CACM-RL*. The most relevant advantage of CACM-RL*, if compared with CACM-RL, is in the discretization process where after the identification and establishment of the boundaries of each state variable, it is necessary to set the cell size per variable as a function of the needed accuracy and available resources (memory and processing power). It is in this step that the CACM-RL* shows its strength because it removes any constraint associated with the cell sizes. In Fig. 5-a, three transitions from the same initial cell (we suppose to begin from the centre of the cell) for three different control actions (a_1, a_2, a_3) are shown. The lengths of the subsequent transitions are, as it is shown in Fig. 5(a): $t_1 = 0.2$, $t_2 = 0.1$ and $t_3 = 0.3$.

In (2), the overall expression for the reward, acquired by each state when



(a) (b)

Figure 4. Controllability maps of the position control problem of a DC motor using CACM-RL (a) and CACM-RL* (b). The ranges of X_1 [10 - 20], X_2 [10/s - 20/s] are covered in five steps with a sample period of 5 ms

applying a specific control action, is defined. This expression is used in CACM-RL in the interaction with the environment.

$$Q\left(s_t, a_t\right) = Q\left(s_t, a_t\right) + \alpha\left(r_{t+1} + \gamma Q\max\left(s_{t+1}, a\right) - Q\left(s_t, a_t\right)\right). \tag{2}$$

CACM-RL updates the current reward, $Q(s_t, a_t)$, in an iterative way, according to (2). The reward concept is used to search iteratively for the optimal control action associated to each state. Following (3), the formulation specified in (2) converges to (4). CACM-RL considers only the centre of the cell in the reward evaluator (2). With these prerequisites, Fig. 5-a shows how the (a_2, t_2)-pair obtains the maximum reward because the associated transition is the fastest.

However, if we consider the possibility of transit to an area that covers several cells and we associate a specific probability of occurrence to each transition, we can obtain a transition structure as shown in Fig. 5-b. In this way, we have a surface image per control action. For example, the (a_1, t_1)-pair in Fig. 5-b may lead to several transitions to different adjoining cells, each with different probability: $p_{Q2} = 0.1$, $p_{Q3} = 0.4$, $p_{Q5} = 0.1$, $p_{Q6} = 0.4$. Taking into account this new way of proceeding with respect to CACM-RL, it is possible to reformulate (2) with a statistical approach (5) and thereby implement CACM-RL*:

$$Q\left(s_t, a_t\right) = Q\left(s_t, a_t\right)\left(1 - \alpha\right) + \alpha\left(r_{t+1} + \gamma Q\max\left(s_{t+1}, a\right)\right) \tag{3}$$

$$\lim_{t\to\infty} Q\left(s_t, a_t\right) = \left(r_{t+1} + \gamma Q\max\left(s_{t+1}, a\right)\right) \tag{4}$$

$$\bar{Q}\left(s_t, a_t\right) = \sum_{i=1}^{n} p_i \cdot \left[Q\left(s_t, a_t\right) + \alpha\left(r_{t+1} + \gamma Q_{max}\left(s_{t+1}, a\right) - Q\left(s_t, a_t\right)\right)\right] \tag{5}$$

where $\bar{Q}$ is the average weighted value of Q, n is the number of states reachable from the initial state, and p_i is the probability of transition.

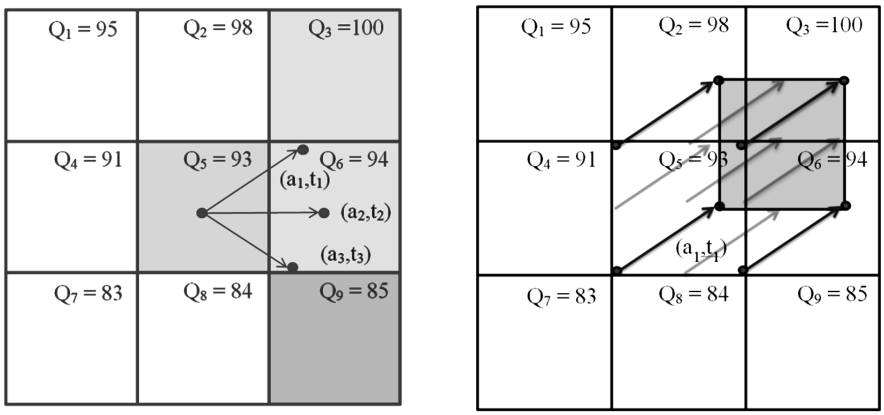


Figure 5. Multiple transitions from an initial cell to the same image cell (a) and to different image cells (b)

Table 1 specifies the transition evaluation when applying (2) and (5) for purposes of comparing CACM-RL and CACM-RL*. It allows to see the convergence of Q(s, a). In this example, the parameters α and γ are equal to 0.6 and 1, respectively.

In (6) and (7) the evaluation of $Q(s_t, a_t)$ and $Q(s_t, a_t)^*$ is performed. In (7), the transition probability is considered according to the overlapping regions, shown in Fig. 5-b. As it can be noted in Table 1, the control action with the highest reward in CACM-RL is a_2 while in CACM-RL* it is a_1.

$$\begin{aligned}
&\lim_{t\to\infty} Q\,(5, a_1) = (-0.2 + 94) = 93.8\\
&\lim_{t\to\infty} Q\,(5, a_2) = (-0.1 + 94) = 93.9\\
&\lim_{t\to\infty} Q\,(5, a_3) = (-0.3 + 94) = 93.7
\end{aligned} \tag{6}$$

$$\begin{aligned}
&\lim_{t\to\infty} Q\,(5, a_1)^* = (-0.2 + 0.1 \cdot 98 + 0.4 \cdot 100 + 0.1 \cdot 93 + 0.4 \cdot 94) = 96.4\\
&\lim_{t\to\infty} Q\,(5, a_2)^* = (-0.1 + 0.01 \cdot 93 + 0.99 \cdot 94) = 93.89\\
&\lim_{t\to\infty} Q\,(5, a_3)^* = (-0.3 + 0.15 \cdot 93 + 0.05 \cdot 84 + 0.45 \cdot 94 + 0.35 \cdot 85) = 89.9.
\end{aligned} \tag{7}$$

There are several methods for evaluating the transition probability from each state-action pair. One approach consists in quantifying the probability of transition by sampling uniformly each cell as it is shown in Fig. 5-b. By doing

this, we can know the number of transitions that fall in each of the different image cells and consequently calculate the probability. The Monte Carlo method could be also used, although it is more appropriate when working with systems characterized by a high number of dimensions.

Table 1: CACM-RL vs CACM-RL*

	γQ_{max}	r_{t+1}	$Q(s_t, a_t)$
CACM-RL			
$Q(5, a_1)$	94.00	-0.2	93.80
$Q(5, a_2)$	94.00	-0.1	93.90
$Q(5, a_3)$	94.00	-0.3	93.70
CACM-RL*			
$Q(5, a_1)^*$	96.60	-0.2	96.40
$Q(5, a_2)^*$	93.99	-0.1	93,89
$Q(5, a_3)^*$	90.20	-0.3	89.90

According to the statistical process, described previously, CACM-RL* drives us to consider non-critical adjoining distances, that is to say, we do not need to set a specific adjoining distance in an explicit way during the learning stage, as was the case with CACM-RL (see Gómez, 2007, 2012, or 2009). CACM-RL* implicitly adapts the adjoining distance in an automatic way, taking into account the sample period and the system's dynamics. In Fig. 6, we can see two trajectories obtained with CACM-RL*, from the same origin to the same goal, for the same motor position control problem, described previously. In order to appreciate the non-uniformity of the adjoining distance, the white trajectory has been traced in a theoretical way, always using the cell centre as the start of any transition. We can see by means of the white points that the adjoining distance is not constant for each state, in order to achieve an adaptation to the system's dynamics. However, the dark trajectory is generated in a real way, starting each new transition just in the previously reached state. If we compare both paths, we can see that the real trajectory tracks the white path and, therefore, it tends to the optimal solution.

In order to analyse how system's controllability evolves when applying CACM-RL* and with due account of the clarity of its results, the previous motor position control problem has been solved using different sample periods, so that we get different controllability maps: $T_{S1} = 5$ ms, $T_{S2} = 10$ ms, $T_{S3} = 20$ ms and $T_{S4} = 30$ ms (see Fig. 7). All controllers, generated for each period, are perfectly valid since the lower the sample period is, the closer the controller is to the optimal solution. On the contrary, when the sample period increases, the frontier between the two control regions becomes wider. This means that the response time of the system is longer because the motor has to begin to brake well in advance, before reaching the goal.

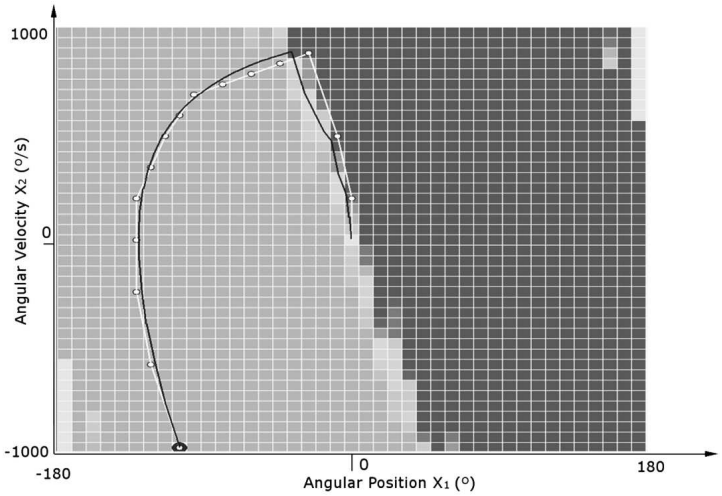


Figure 6. CACM-RL* adaptive adjoining trajectory with T_s equal to 20 ms

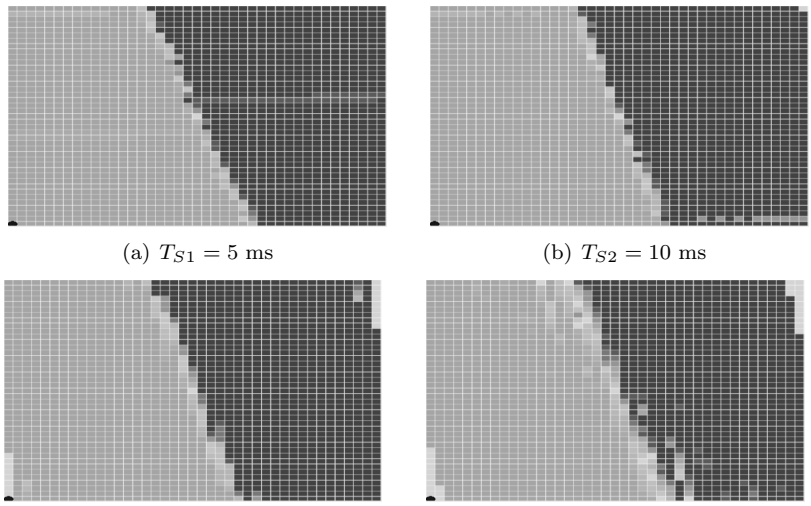
3. Results

In order to evaluate CACM-RL*, two examples have been considered: position control of a DC motor, which has been compared with CACM-RL and PID controllers, and the "Car on the Hill" problem (see Moore, 1995, or 1990), for which the comparison with CACM-RL has been performed. In both examples, one can observe the feasibility and stability of the solution over the entire range of cell sizes, making CACM-RL* a robust solution in comparison with CACM-RL or PID controllers. CACM-RL* allows the engineer to focus only on the accuracy required to solve a control problem when establishing the cell size.

3.1. CACM-RL* vs CACM-RL in the position control of a real DC motor

To show the feasibility of CACM-RL*, a 2-D controllability map with 25 control planes has been calculated. We have selected the angular velocity and angular position as state variables. In Fig. 4-a, the 2-D map is shown for position control of a DC motor using CACM-RL and in Fig. 4-b the same map for CACM-RL* with a $T_{S1} = 5$ ms.

The quality of a controller can be quantified by several criteria like controllability (see Song, 2002), stability, reliability or feasibility. When the controllability achieved by the two controllers is similar, one indicator used to compare the quality is the "time to goal" (see Papa, 1995, 1997). In this way, a histogram test is performed for each controller. It quantifies the average time spent on transiting from controllable cells to the objective cell. The dispersion associ-



(a) $T_{S1} = 5$ ms (b) $T_{S2} = 10$ ms

(c) $T_{S3} = 20$ ms (d) $T_{S4} = 30$ ms

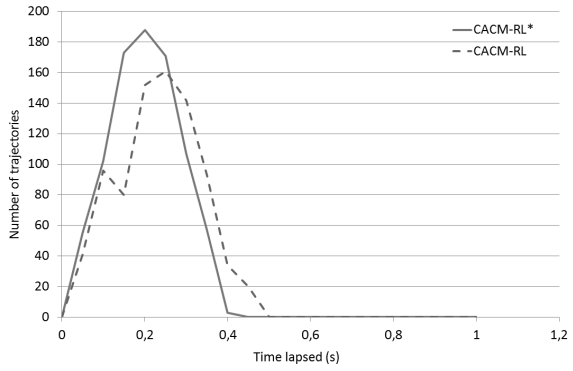
Figure 7. Controllability maps of the position control problem of a DC motor, where controllability equals 83% in all cases and four sample periods: Ts_1=5 ms, Ts_2=10 ms, Ts_3=20 ms, Ts_4=30 ms, are accounted

ated to the histogram graph is an indicator of the quality of the controller. The smaller the dispersion, the better the quality controller. Fig. 8 shows the two histograms, one for CACM-RL and the other one for CACM-RL*.

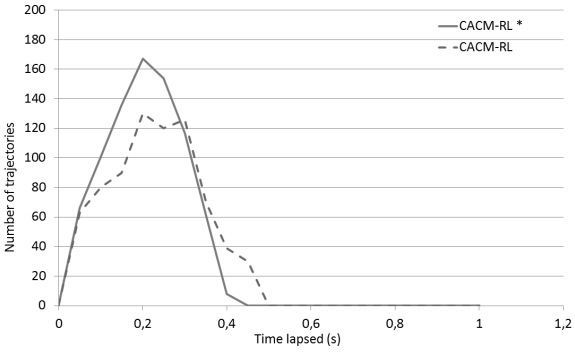
Fig. 4-a shows several controllability maps, where the frontier between different control actions is not well formed and some random effects are generated due to the mutual influence of the two variables. Therefore, this adversely affects the quality of the controller in terms of controllability and stability. Worse even, these effects make it difficult to find a suitable combination of cells. However, the results achieved with CACM-RL*, as shown in Fig. 4-b, are perfectly defined by a clear frontier, obtaining in all cases an optimal solution with a constant controllability, and where the accuracy just depends on the cell size. In this sense, the flexibility of CACM-RL* shown in Fig. 4-b allows to discretize the variables without any kind of constraint and perform an automatic adjoining process between cells.

According to Fig. 8, CACM-RL* provides the best solution for both the high resolution case and for the low resolution case. Another remarkable characteristic of CACM-RL* is that the memory resources can be reduced thanks to its cell size tolerance. As an example, with the lowest resolution and using 50% of the memory resources needed by the highest resolution case, we obtain the same controllability.

Table 2 shows the higher controllability with CACM-RL* than with CACM-RL (100% vs 72% in the best case).



(a)



(b)

Figure 8. Histograms for the position control of a DC motor: using high resolution cells (a) and low resolution cells (b)

Table 2 CACM-RL vs CACM-RL* in the position control of a real DC motor

	Controllability	Histogram
CACM-RL high res	72%	dashed line - Figure 8a
CACM-RL low res	5%	dashed line - Figure 8b
CACM-RL* high res	100%	solid line - Figure 8a
CACM-RL* low res	95%	solid line - Figure 8b

3.2. CACM-RL* vs CACM-RL in the *"Car on the Hill" problem*

The "Car on the Hill" problem, shown in Fig. 9 and defined in Moore (1995, 1990), is a 2-D problem, where the state variables are: X_1 is the position (m) and X_2 is the velocity (m/s). The control action vector is composed of only three values, F=[-4, 0, +4] N. The goal of the problem is to reach the goal of X_1=1 and X_2=0. The difficulty of this control problem is that the force does not suffice to drive the vehicle to the goal from the valley to the hill, and the controller has to increment the kinetic energy of the vehicle iteratively, up to exactly reach the goal on the hill.

It is important to emphasize here that the "Car on the Hill" problem is a theoretical non-linear problem, which aims to verify the performance of controllers. Because of its difficulty, it is a good example for comparing CACM-RL with CACM-RL*.

The dynamics equations that define the behaviour of the problem are the following:

$$x' = v$$
$$H' = \left\{ \begin{array}{c} x\,(x+1)\,; x < 0 \\ \frac{K_1 x}{\sqrt{1+K_2 x^2}}; x \geq 0 \end{array} \right\}, \tag{8}$$

where x is the car position, v is the car velocity that can be written as:

$$v = \frac{F}{M\sqrt{\left(1+H'^2\right)}} - \frac{gH'}{1+H'^2} \tag{9}$$

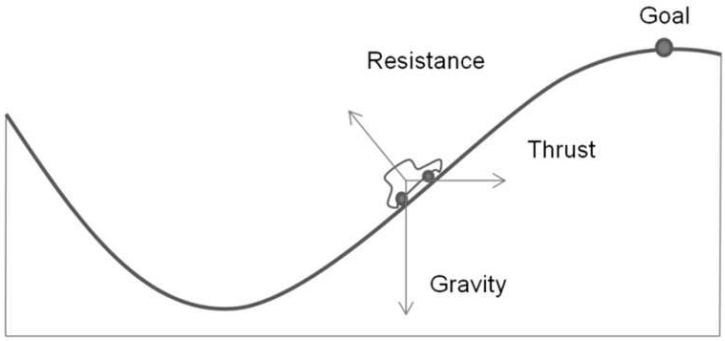


Figure 9. "Car on the Hill" problem

M is the car mass, and g is gravity. K_1, K_2 and M are constants whose values are 1, 5 and 1, respectively.

In Fig. 10, an optimal control surface of the problem is shown. A random subset of 1000 trajectories has been generated from different initial cells. The

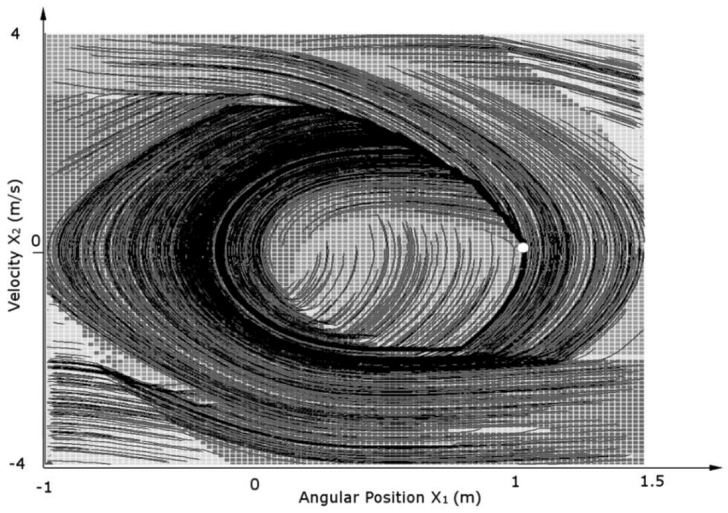


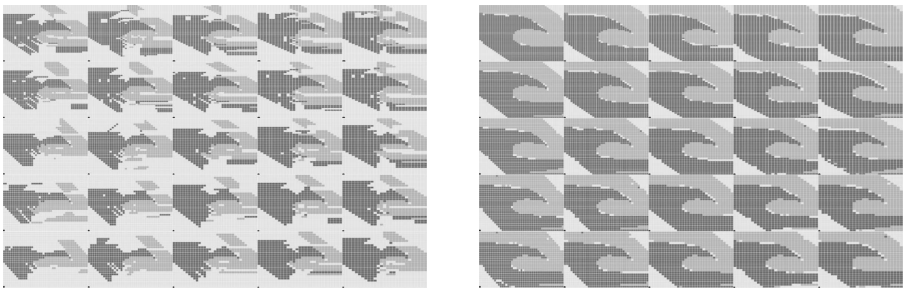
Figure 10. Optimal trajectories from different initial cells to the goal in the "Car on the Hill" problem

triangle-shaped regions at the top right and at the bottom left corners are the non-controllable areas. In these zones close to the frontier of the state space, the velocity is so high that the system is not able to stop the car and to lead the vehicle to the attractor. This is the reason why this problem is not 100% controllable in the entire state space.

As we did with the previous example, which was focused on the position control of a DC motor, in order to compare the results achieved with CACM-RL and CACM-RL*, a set of 25 controllability maps (using different cell sizes in each one) has been generated. Fig. 11-a shows the controllability maps for CACM-RL and Fig. 11-b for CACM-RL*. In this case, the comparison is clear: when using any cell size with CACM-RL, the achieved controllability maps are non-continuous and non-uniform. At the CACM-RL*, as shown in Fig. 11-b, the maps are perfectly defined with a marked frontier between control actions areas.

In the histograms of Fig. 12, we obtain a better controllability and better average time to reach the goal when applying CACM-RL* than with CACM-RL. The narrower the histogram, the better the controller from the optimality point of view. Furthermore, the more cells reach the goal in the average time, the narrower histogram is and therefore, the controller is closer to its optimal implementation.

In general, the "Car on the Hill" problem, when solved with CACM-RL*, shows a better controllability than when solved with CACM-RL (74% vs 64% in the best case, see Table 3). It is also important to highlight that CACM-



(a) (b)

Figure 11. Controllability maps of the "Car on the Hill" problem using CACM-RL (a) and CACM-RL* (b). The ranges of X_1 [0.05m - 0.1m], X_2 [0.16m/s - 0.32m/s] are covered in five steps with a sample period of 5ms

RL* provides the same controllability in all the studied cases (independently of the cell sizes). This means that for the proposed problem it is not needed to increase the resolution in order to improve the controllability. According to Table 3, CACM-RL* obtains not only the highest controllability, but also it is constant despite the changes in the grid resolution.

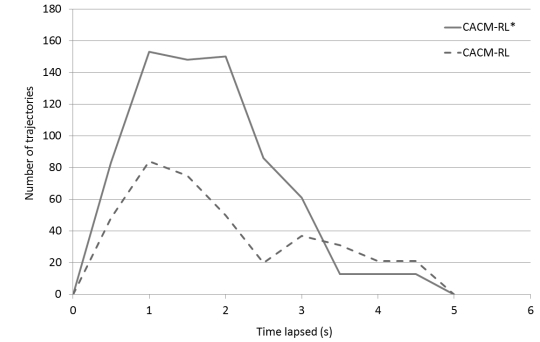
Table 3: CACM-RL vs CACM-RL* in the "Car on the Hill" problem

	Controllability	Histogram
CACM-RL high res	64%	dashed line - Figure 12a
CACM-RL low res	54%	dashed line - Figure 12b
CACM-RL* high res	74%	solid line - Figure 12a
CACM-RL* low res	64%	solid line - Figure 12b

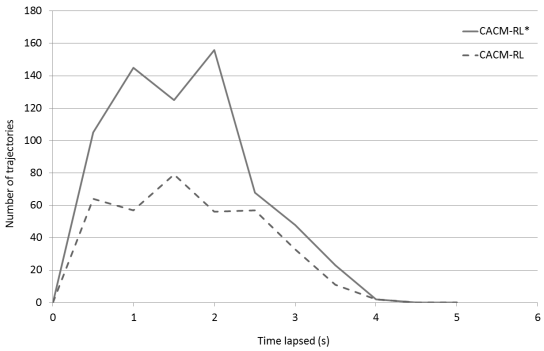
3.3. CACM-RL* vs PID in the position control of a real DC motor

In order to study CACM-RL* in a real scenario, we have chosen the position control problem of a DC motor and a PID controller for comparing results. According to Section 3.1, the position control of a DC motor requires the use of two state variables in order to ensure the optimal control: angular position (X_1) and angular velocity (X_2). In this case and for CACM-RL*, the grid used has been of 21 x 21 cells, obtaining cell sizes of 17.14° and 47.6°/s, respectively.

The main goal during the learning stage of CACM-RL* is to select and try control actions. CACM-RL* has to try actions that have not been selected before (see Gómez, 2012, 2009, 2011). In this sense, exploration is the key to learning the optimal control action. In order to secure a good learning, it is necessary that all control actions for each cell (state) be applied during the exploration. The more times the controller chooses a control action, the better



(a)



(b)

Figure 12. Histograms for the "Car on the Hill" problem: using high resolution cells and low resolution cells

the learning will be. There are several strategies used for exploration. In our case, we use random selection, because it gives good performance; all actions are chosen enough number of times for achieving the best long-term effect from the real plant (see Gómez, 2011).

After about 60 seconds of exploration, the DC motor completes the learning stage. With this learning, the controllability is 96.8% (the cells located at the top-right and bottom-left corners are not controllable) as it can be seen in Fig. 13. Also, Fig. 13 shows an optimal control surface together with a set of trajectories generated from different initial cells, after finishing the learning stage.

The PID controller, in comparison with CACM-RL*, requires a critical tuning and always will be working between the overdamped and underdamped behaviours. Furthermore, the static or stationary error associated with PID

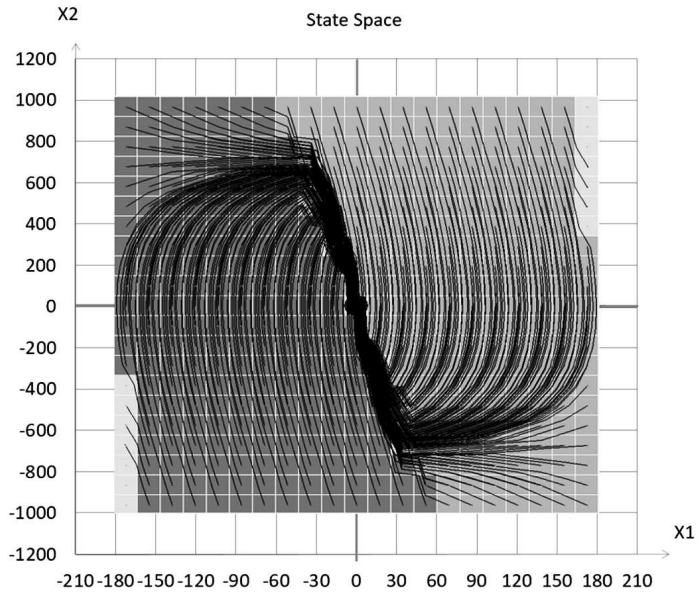


Figure 13. Optimal trajectories, leading from different initial cells to the goal (centre coordinate) in the position control problem of a DC motor using CACM-RL*

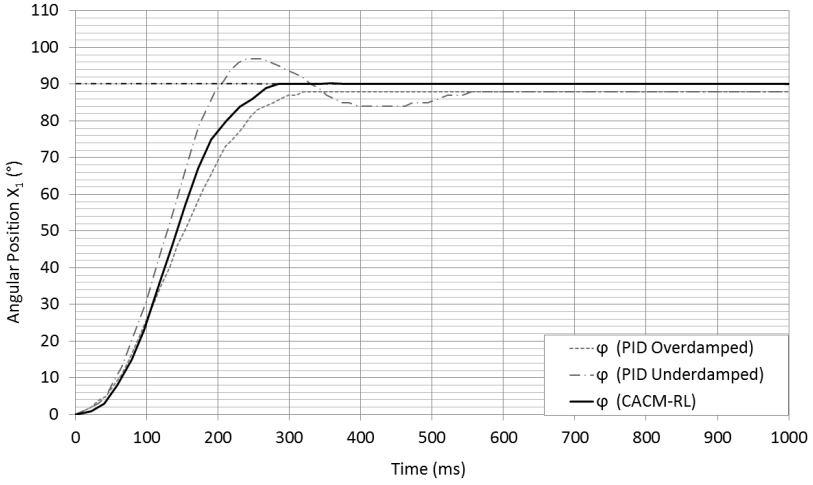


Figure 14. CACM-RL* vs PID responses to the position control of a DC motor

cannot be efficiently reduced in time. Also, PID is not as accurate as CACM-RL*. On the other hand, as it can be seen in Fig. 14, with CACM-RL*, the instabilities and the static error are avoided.

4. Conclusions

In this work, a new algorithm called CACM-RL*, derived from CACM-RL, has been presented. CACM-RL* is a robust and efficient technique that reduces the engineering effort, required to achieve an optimal controller, when compared with CACM-RL. With CACM-RL*, the memory resources can be considerably reduced thanks to its flexibility in supporting any cell size. This property allows for avoiding increase of the number of cells to achieve the accuracy requirements. For this reason, CACM-RL* lets us discretize the state variables without any kind of constraint, and, implemented as a part of the internal behaviour, also the automation of the performed adjoining distances.

The sample period and the time-delay control loop are very influential parameters for the controllability of a system. With the use of CACM-RL, the controllability of a system may happen to be reduced due to a bad cell size selection. However, CACM-RL* adapts the adjoining distance to the sample period and, in this way, the controllability is not affected by the design parameter constraints.

With CACM-RL*, the design problems derived from the state space discretization and the critical relationship between dimensions have been satisfactorily resolved. In addition, the optimality and efficiency of performance rely exclusively on the cell size and dedicated resources. The tedious trial and error process meant to adjust the cell size in CACM-RL has been overcome. With this new algorithm, engineers only have to focus on accuracy and stability criteria, instead of the cell sizes.

References

BARTO, A. (1998) *Reinforcement Learning: An Introduction.* MIT Press.

BARTO, A., BRADTKE, S. J. and SINGH, S. P. (1995) Learning to act using Real-Time Dynamic Programming. *Artificial Intelligence, Special Volume on Computational Research on Interaction and Agency*, **72** (1), 81-138.

BELLMAN, R. E. (2010) *Dynamic Programming.* Princeton University Press.

BURSAL, F. H. and TONGUE, B. H. (1992) A New Method of Nonlinear System Identification using Interpolated Cell Mapping. *American Control Conference, Chicago, USA*, 3160-3164. IEEE.

GÓMEZ, M. (2009) Planificación óptima de movimiento y aprendizaje por refuerzo en vehículos móviles autónomos. PhD thesis, Universidad de Alcalá.

GÓMEZ, M., ARRIBAS, T. and SÁNCHEZ, S. (2012) Optimal Control based on CACM-RL in a Two-Wheeled Inverted Pendulum. *International Journal of Advanced Robotic Systems*, **9** (1), 1-8.

GÓMEZ, M., GONZÁLEZ, R. V., MARTÍNEZ-MARÍN, T., MEZIAT, D. and SÁNCHEZ, S. (2011) Optimal Motion Planning by Reinforcement Learning in Autonomous Mobile Vehicles. *Robotica*, **30** (2), 159-170.

GÓMEZ, M., MARTÍNEZ-MARÍN, T., SÁNCHEZ, S. and MEZIAT, D. (2007) Optimal control applied to Wheeled Mobile Vehicles. *Proc. IEEE International Symposium on Intelligent Signal Processing*, Alcalá de Henares, Madrid, Spain, 83-88. IEEE.

GUTTALU, R. S. and ZUFIRIA, P. J. (1993) The adjoining cell mapping and its recursive unraveling, Part II: Application to selected problems. *Nonlinear Dynamics*, **4** (4), 309-336.

HSU C.S., (1985) A discrete method of optimal control based upon the cell state space concept. *Journal of Optimization Theory and Applications*, **46** (4), 547-569.

MO-HONG, C. (1993) A modified cell-to-cell mapping method for nonlinear systems. *Computers & Mathematics with Applications*, **25** (8), 47-57.

MOORE, A. (1990) Efficient Memory-Based Learning for Robot Control. PhD thesis, University of Cambridge.

MOORE, A. and ATKESON, C. (1995) The parti-game algorithm for variable resolution reinforcement learning in multidimensional state space. *Machine Learning*, **21** (3), 199-233.

PAPA, M., HENG-MING, T. and SHENOI, S. (1995) Evaluation of cell state techniques for optimal controller design. *Proc. Int. Joint Conference of the Fourth IEEE International Conference on Fuzzy systems and The Second International Fuzzy Engineering Symposium*, Yokohama, Japan, **3**, 1331-1338. IEEE.

PAPA, M., HENG-MING, T. and SHENOI, S. (1997) Cell mapping for controller design and evaluation. *IEEE Control Systems*, **17** (2), 52-65.

SONG, F. and SMITH, S. M. (2002) Cell-state-space-based search. *IEEE Control Systems*, **22** (4), 42-56.

TONGUE, B. H. (1987) On the Global Analysis of Nonlinear Systems through Interpolated Cell Mapping. *Physica D*, **28**, 401-408.

WATKINS, C. J. C. H. and DAYAN, P. (1992) Technical note: Q-learning. *Machine Learning*, **8** (1), 279-292.

WILHELMUS, J. A. (1994) Cell Mapping methods: modifications and extensions. PhD thesis, Eindhoven University of Technology.

ZUFIRIA, P. J. and GUTTALU, R. S. (1993) The adjoining cell mapping and its recursive unraveling, Part I: Description of adaptive and recursive algorithms. *Nonlinear Dynamics*, **4** (3), 207-226.

ZUFIRIA, P. J. and MARTÍNEZ-MARÍN, T. (2003) Improved Optimal Control Methods based upon the Adjoining Cell Mapping Technique. *Journal of Optimization Theory and Applications*, **118** (3), 657-680.