An Algorithm for Cooperative Probabilistic Control Design

Miguel Barão

Abstract— **This paper deals with the decentralized closed loop control in a pure probabilistic framework. In this framework, a system is a controlled Markov chain whose transition probabilities depend on the actions of the agents. The agents are also described in a probabilistic way. The objective is to drive the system so that the joint state and agents actions are close to a set of given target probability distributions. The Kullback-Leibler divergence is used as a performance measure. The resulting algorithm uses dynamic programming interleaved with an iterative process that computes the behavior of each agent.**

I. Introduction

Distributed and cooperative control has been the focus of intense research in recent years, many centralized problems being recast into this realm. While many problems are deterministic in what concerns the control action, and deriving their uncertainty from the system or from imperfect communication channels, other problems exist where the control action is itself stochastic by nature. These kinds of randomized problems are tackled with in the most general way by describing the intervening models by probability distributions. The probabilistic control framework arises therefore as a natural way to deal with them and methods are required to design these controllers.

Past work on probabilistic control has been done in [5], [6] leading to explicit formulas that can be applied in controlled Markov chains. These works relate with Markov Decision Processes [8], with the difference that the Kullback-Leibler divergence is used as the cost function instead of arbitrary rewards on the states and actions, as is usually done with MDPs. The result is a probabilistic controller $c(u_t|x_t)$ that generates a randomized actuation variable u_t conditional on the system state x_t.

The main contribution of the paper is the extension to the case where more than one controller acts simultaneously on the same system. The main differences to earlier probabilistic control situations are that the controllers can be different from each other, and can act differently on the system. These controllers may or may not communicate with each other. If they do not communicate, which is the situation considered in this paper, only each others' behaviors are known. This knowledge is the result of the design process and is represented in the form of conditional probability distributions. If they can communicate, then their actual combined actuation can be thought of as if a joint distribution is in place. The multi agent formulation can be looked at from a game theoretic point of view [4], [11], where multiple players cooperate to a common objective.

This framework relates with Markov Games. These deal with the situation where decisions are taken by multiple agents acting on a common system. A particular case is known as Multiagent Markov Decision Processes (MMDPs) where the rewards/costs are shared among the agents [9], [7], [10]. This condition is sometimes referred as the agents being cooperative (see cited papers) although the game played is called noncooperative since they take decisions independently. A similar situation is considered here.

The paper is organized as follows: section II formulates the standard probabilistic control problem and the situation where two independent controllers act simultaneously on the same system; section III proposes a solution based on a two-player iterated policy game, where each controller (player, agent) perceives an equivalent system encompassing the true system and the other controller; section IV illustrates with a toy example, the stochastic JK flip-flop; section V illustrates a different example requiring coordination; and section VI draws conclusions.

II. Problem formulation

In the probabilistic framework all of the knowledge concerning variables and behaviors is represented by probability distributions. For example, a system having two inputs u^1 and u^2 updates its own state x_t according to the conditional distribution

$$s(x_{t+1}|x_t, u_t^1, u_t^2). \tag{1}$$

We assume that the inputs u^1 and u^2 are generated by two controllers (other common terminology is to call them agents or players) having complete access to a common shared state x_t. The decisions taken by the controllers are also assumed to depend only on the shared state, and so their models are the conditional probability distributions

$$c_1(u_t^1|x_t), \quad c_2(u_t^2|x_t). \tag{2}$$

The closed loop connecting the system (1) and controllers (2) is depicted in the diagram 1.

Under the assumption of independence between the actions u_t^1 and u_t^2 at time t, the joint distribution of the state and actions over a time period T is obtained by the product rule of probabilities

$$\begin{aligned} p(x_{1:T}, u^1_{0:T-1}, u^2_{0:T-1}|x_0) = \\ = \prod_{t=0}^{T-1} s(x_{t+1}|x_t, u_t^1, u_t^2) c_1(u_t^1|x_t) c_2(u_t^2|x_t), \end{aligned} \tag{3}$$

This work was supported by FCT (INESC-ID multiannual funding) through the PIDDAC Program funds and project PROBCONTROL - PTDC/EEA-CRO/115038/2009.

M. Barão is with Évora University and INESC-ID Lisboa, Portugal miguel.barao@inesc-id.pt

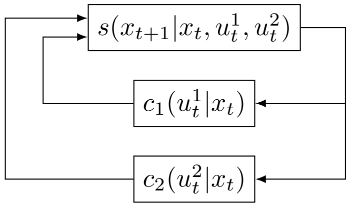


Fig. 1. Agents in closed loop with system s.

where the index notation $x_{1:T} \triangleq (x_1, \ldots, x_T)$ is used.

Our aim is to find the distributions $c_1(u_t^1|x_t)$ and $c_2(u_t^2|x_t)$ that minimize the Kullback-Leibler divergence to some previously specified target distributions

$$S(x_{t+1}), \quad C(u_t^i). \tag{4}$$

The Kullback-Leibler divergence is defined generically by

$$D(p\|q) \triangleq \sum_x p(x) \log \frac{p(x)}{q(x)}, \tag{5}$$

and is a measure similar to a distance in what $D(p\|q) = 0$ when $p(x) = q(x)$, and positive otherwise, but is not symmetric and does not satisfy the triangular inequality, two required properties for a metric distance.

To optimize the closed behavior, the Kullback-Leibler divergence between the joint distribution (3) and the target distribution

$$\begin{aligned} q(x_{1:T}, u^1_{0:T-1}, u^2_{0:T-1}|x_0) &= \\ &= \prod_{t=0}^{T-1} S(x_{t+1}) C_1(u_t^1) C_2(u_t^2) \end{aligned} \tag{6}$$

is minimized, where $S(x_{t+1})$ and $C_1(u_t^1), C_2(u_t^2)$ specify the target distributions for x and (u^1, u^2) in an analogous way to the specification of a cost function in optimal control.

The probabilistic control problem has been solved in [6] using an explicit approach, and in [1], [2], [3] using an iterative information geometrical method, for the situation where a single controller $c(u_t|x_t)$ is sought. The solutions found there are based essentially in the factorization of the joint distributions p and q, and then using properties of the Kullback-Leibler divergence to write a Bellman equation, which is then solved by dynamic programming.

The explicit solution found in [6] can be rewritten as follows:

$$\begin{aligned} c(u_t|x_t) = \exp\Big(&\log C(u_t) - D(s_{t+1}\|S_{t+1}) \\ &- E_{s_{t+1}}[-\log \gamma_{t+1}(x_{t+1})] - \log \gamma_t(x_t)\Big) \end{aligned} \tag{7}$$

where

$$\begin{aligned} \gamma_t(x_t) \triangleq \sum_{u_t} \exp\Big(&\log C(u_t) - D(s_{t+1}\|S_{t+1}) \\ &- E_{s_{t+1}}[-\log \gamma_{t+1}(x_{t+1})]\Big) \end{aligned} \tag{8}$$

is a normalization constant.

This formulation could be extended to allow two input signals by doing $u_t \triangleq (u_t^1, u_t^2)$. This means that (u_t^1, u_t^2) would be jointly specified, and therefore u_t^1 and u_t^2 depend on each other, a condition that requires perfect communication between them.

In the formulation considered in this paper, the single controller $c(u_t^1, u_t^2|x)$ is replaced by two separate controllers that may or may not exchange information between them. The situation where they do not directly communicate corresponds to the independence assumption

$$c(u_t^1, u_t^2|x_t) = c_1(u_t^1|x_t) c_2(u_t^2|x_t). \tag{9}$$

Under this constraint, a design process is required to specify each individual controller.

One possibility is to compute the joint optimal controller $c(u_t^1, u_t^2|x_t)$, and then marginalize to obtain controllers $c_1(u_t^1|x_t)$ and $c_2(u_t^2|x_t)$. This is an *adhoc* procedure, however, and does not guarantee by itself to yield an optimal pair of controllers under the independence assumption. A different method is proposed next.

III. Proposed solution

The proposed solution is based on a game theoretic approach where two players, the controllers, try to minimize the overall cost function. An iterative process is employed to achieve a Nash equilibrium solution.

At each time step t of the backward induction (7) and (8), one controller perceives an equivalent system corresponding to the closed loop of the system with the other controller (see figure 2). A similar situation occurs for the second controller albeit the equivalent system is generally different.

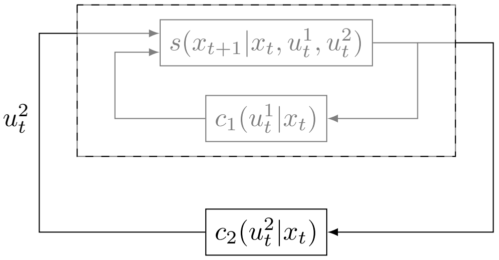


Fig. 2. Equivalent closed loop system as perceived by controller 2, and assuming the controller 1 is known.

The joint optimization of the controllers at time t is performed by iterating the control design process for each controller separately. At each iteration step, a new equivalent system is computed assuming the controller obtained at the previous iteration. The following algorithm illustrates the process:

- Set controller 1 to some initial candidate controller;
- Using controller 1, compute the equivalent system
$$s_2(x_{t+1}|x_t, u_t^2) = \sum_{u_t^1} s(x_{t+1}|x_t, u_t^1, u_t^2) c_1(u_t^1|x_t) \tag{10}$$
as perceived by controller 2.

- Optimize controller $c_2(u_t^2|x_t)$ with respect to the equivalent system $s_2(x_{t+1}|x_t, u_t^2)$ using equations (7) and (8).
- Using controller 2, compute the equivalent system
$$s_1(x_{t+1}|x_t, u_t^1) = \sum_{u_t^2} s(x_{t+1}|x_t, u_t^1, u_t^2)c_2(u_t^2|x_t) \tag{11}$$
as perceived by controller 1.
- Optimize controller $c_1(u_t^1|x_t)$ using equations (7) and (8).
- Repeat until convergence is achieved.

This procedure leads to a Nash equilibria where neither controller (a controller is a player in this two player game) can obtain a better performance by changing its behavior unilaterally.

If no further information is available to the controllers, their (probabilistic) actions follow the equilibrium controllers behavior.

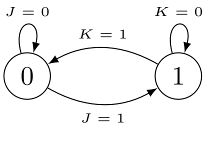
IV. Example I

For illustrative purposes, the solution proposed in the previous section is applied to a very simple toy problem: a stochastic flip-flop.

A JK flip-flop is an electronic device that can be in one of the two states $\{0, 1\}$. Its state Q_t can change as a function of the values present at the J and K inputs. Its truth table is the following:

J	K	Q_{t+1}	description
0	0	Q_t	hold current state
0	1	0	reset
1	0	1	set
1	1	$\bar{Q}_t$	toggle current state

We now consider a stochastic JK flip-flop where the truth table is not strictly followed, but instead the truth table rules above are applied with probability α. This flip-flop can be represented as a two-state controlled Markov chain, where the state is $x_t \triangleq Q_t$, and the transition probabilities depend on the input values $(u^1, u^2) \triangleq (J, K)$. The following graph depicts the Markov chain and the conditions that make the indicated transition probabilities equal to α (arrows do not represent transition probabilities).



The value of α is a property of the system. In our simulations we use a probability value of $\alpha = 0.9$.

Suppose that controllers 1 and 2 act on inputs J and K, respectively. If the target state 0 is to be achieved with probability 0.999 and there is no preference or penalization on the inputs (u^1, u^2), *i.e.*
$$c_1(u_t^1|x_t) = c_2(u_t^2|x_t) = 0.5, \tag{12}$$

TABLE I
Equilibrium controller 1 for the stochastic JK flip-flop.

$c_1(u_t^1|x_t)$	$x_t = 0$	$x_t = 1$
$u_t^1 = J = 0$	0.9977	0.4979
$u_t^1 = J = 1$	0.0023	0.5021

TABLE II
Equilibrium controller 2 for the stochastic JK flip-flop.

$c_2(u_t^2|x_t)$	$x_t = 0$	$x_t = 1$
$u_t^2 = K = 0$	0.9952	0.0013
$u_t^2 = K = 1$	0.0048	0.9987

then the design algorithm leads to the controllers shown in tables I and II, after repeating 20 iterations at each one of the 10 time steps optimization.

These results can be interpreted intuitively as follows:

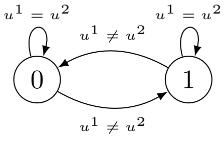
- If the state is $x_t = 0$, then the controllers should use $J = K = 0$. The inherent Markov chain transition probabilities $(\alpha, 1-\alpha) = (0.9, 0.1)$ achieve the desired goal of staying at $x_t = 0$ with probability 0.9, the best possible in these circunstances.
- If the state is $x_t = 1$, then the input K should definitely be $u_t^2 = 1$, while J can have any value: both a reset or a toggle leads to the desired state $x_t = 0$ with probability 0.9.

If the target probabilities for u^1 and u^2 are modified to give preference to one action over the other, then the solution is mostly the same except for $c_1(u_t^1|x_t = 1)$, the second column of table I, where the action will reflect the desired probabilities, since the flip-flop state outcome is independent of the u^1 value.

V. Example II

In this example two controllers can act on the same system using binary inputs, and in the exact same way, *i.e.* the problem is symmetric in what concerns the controllers, as opposed to the previous problem where both acted on different inputs.

Again a binary state is considered. The state toggles with probability 0.9 when the actions of the controllers are different, and holds the state when they are equal. This system can be described by the following two-state controlled Markov chain



where the indicated transitions occur with probability 0.9.

The same design algorithm is tested here, where a high probability is assigned to the desired state $x_t = 1$.

If there are no predefined preferences for the control action, then $C(u_t^1) = C(u_t^1) = 0.5$ is selected. If the design

TABLE III

EQUILIBRIUM CONTROLLER 1 FOR EXAMPLE II.

$c_1(u_t^1\|x_t)$	$x_t = 0$	$x_t = 1$
$u_t^1 = 0$	0.0012	0.9988
$u_t^1 = 1$	0.9988	0.0012

TABLE IV

EQUILIBRIUM CONTROLLER 2 FOR EXAMPLE II.

$c_2(u_t^2\|x_t)$	$x_t = 0$	$x_t = 1$
$u_t^2 = 0$	0.9988	0.0012
$u_t^2 = 1$	0.0012	0.9988

algorithm is initialized using a uniform distribution for the first tentative controller $c(u|x_t) = 0.5$, then the solution is to decide any action with probability 0.5, a situation that could be described as a stall: there is absolutely no information that promotes one action over its opposite. However, a small change in one of the probabilities $C(u_t^i)$, or to the initial tentative controller, leads to the completely distinct probabilities shown in tables III and IV. In this case, controllers take opposite actions in order to achieve the desired goal. There is therefore a discontinuity in the controller probabilities seen as a function of either the problem specification or the initial conditions for the iterative game played.

VI. CONCLUSIONS

This paper dealt with a probabilistic control situation where two controllers act simultaneously on the same system. A solution is proposed that includes a game theoretic approach to get to an equilibrium at each time step of the design phase. The iteration then achieves a Nash equilibrium where neither controller can do better unilaterally, an assumption that is appropriate when no communication between the controllers is possible. The problem can be further extended to a higher number of controllers by assuming equivalent systems from each controllers perspective, although that is not pursued here. To illustrate the procedure, two very simple problems were setup: in the first, a stochastic JK flip-flop was introduced and the obtained controllers were analyzed; the second problem, formulated as a symmetric problem that required cooperation, led to a discontinuous design function, a somewhat surprising result.

REFERENCES

[1] M. Barão. Optimization on discrete probability spaces and applications to probabilistic control design. In *European Control Conference*, Budapest, Hungary, August 2009.

[2] M. Barão. Probabilistic control design using an information geometric framework. In *1st IFAC Workshop on Estimation and Control of Networked Systems*, Venice, Italy, September 2009.

[3] M. Barão and J. M. Lemos. An efficient Kullback-Leibler optimization algorithm for probabilistic control design. In *16th Mediterranean Conference on Control and Automation*, pages 198–203, June 2008.

[4] R. Gibbons. *A Primer in Game Theory*. Prentice-Hall, 1992.

[5] M. Kárný. Towards fully probabilistic control. *Automatica*, 32(12):1719–1722, 1996.

[6] E. Nováková and M. Kárný. Fully probabilistic control design for Markov chains. In *European Control Conference*, 1997.

[7] L. Peshkin, K.-E. Kim, N. Meuleau, and L. P. Kaelbling. Learning to cooperate via policy search. In *Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence*, UAI'00, pages 489–496, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.

[8] M. L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley, 2005.

[9] P. Vrancx. *Decentralized Reinforcement Learning in Markov Games*. PhD thesis, Vrije Universiteit Brussel, 2010.

[10] P. Vrancx, K. Verbeeck, and A. Nowé. Optimal convergence in multi-agent mdps. In B. Apolloni, R. Howlett, and L. Jain, editors, *Knowledge-Based Intelligent Information and Engineering Systems*, volume 4694 of *Lecture Notes in Computer Science*, pages 107–114. Springer Berlin / Heidelberg, 2007.

[11] J. N. Webb. *Game Theory: Decisions, Interaction and Evolution*. Springer, 2007.