An Iterative Scheme for the Approximate Linear Programming Solution to the Optimal Control of a Markov Decision Process

Alessandro Falsone and Maria Prandini

Abstract— **This paper addresses the computational issues involved in the solution to an infinite-horizon optimal control problem for a Markov Decision Process (MDP) with a continuous state component and a discrete control input. The optimal Markov policy for the MDP can be determined based on the fixed point solution to the Bellman equation, which can be rephrased as a constrained Linear Program (LP) with an infinite number of constraints and an infinite dimensional optimization variable (the optimal value function). To compute an (approximate) solution to the LP, an iterative randomized scheme is proposed where the optimization variable is expressed as a linear combination of basis functions in a given class: at each iteration, the resulting semi-infinite LP is solved via constraint sampling, whereas the number of basis functions is progressively increased through the iterations so as to meet some performance goal. The effectiveness of the proposed scheme is shown on a multi-room heating system example.**

I. INTRODUCTION

The goal of this paper is to develop computationally effective control design methods for large scale systems where continuous dynamics, discrete dynamics, and uncertainty are tightly coupled, [1]. We adopt the quite comprehensive modeling framework of Markov Decision Processes (MDPs, [2]) with a continuous, possibly hybrid, state space, and explore the use of randomized methods to defeat the curse of dimensionality hampering the use of standard control design techniques.

We focus on the problem of designing a state feedback control policy that maximizes an infinite-horizon expected discounted reward function for an MDP with a hybrid state space and a discrete input space. The considered stochastic optimal control problem can be solved –in principle– through dynamic programming by determining the fixed-point of the Bellman equation so as to determine the so-called optimal value function and then the optimal control policy (see e.g. [3], [4]). The optimal value function and policy can be efficiently computed when the state and control spaces are finite and not too large compared with the memory storage capacity. If the state space is infinite but its dimension is small, computations can be performed by gridding the space, approximating the original MDP with a finite state MDP, and determining the (approximate) optimal value function and control policy on the grid points, see e.g [5], [6], [7]. As a result, the practical use of dynamic programming is limited by the exponential growth with the problem dimensions of the computation and storage requirements.

Inspired by [8] and [9], we develop in Section II an approximate dynamic programming method resorting to random sampling of the state space instead of gridding in order to overcome these limits. Scalability of the proposed approach could be further enhanced by taking advantage of model abstraction techniques (see e.g. [10] and the reference therein), which, however, are not considered in this paper.

As in [11] dealing with finite state MDP, we start by rephrasing the Bellman equation as a constrained Linear Program (LP) where the unknown quantity to be determined via optimization is the optimal value function, and it is subject to a set of constraints, one per each state-control input pair. When the state space has a continuous component, the resulting LP is infinite dimensional in both the decision variables and the constraints. By expressing the optimal value function to be determined as a linear combination of a finite number of basis functions, then, the infinite LP is transformed into a semi-infinite LP, i.e., an LP with a finite number of decision variables but an infinite number of constraints, which, in turn, can be tackled via the scenario approach to robust convex optimization. The idea of the scenario approach is to consider a finite number N of constraints only, and solve the resulting finite LP. If N is appropriately chosen, then, the originally infinite constraints are guaranteed to be satisfied probabilistically, with a certain confidence, [12], [13], [14]. Quality of the obtained approximation of the value function (and, hence, of the resulting policy) is strongly affected by the choice of the basis functions. This is indeed a key issue in function approximation as well, where data samples in the form of input and output of an unknown (static) function are available and an approximation of the function through a linear combination of basis functions is looked for. Families of universal approximators have been studied in this context (see e.g. [15], [16], [17]), but an effective method to select an appropriate finite number of functions out of a given family is still to be developed. The present paper represents a first step in this direction within the more challenging framework of approximate dynamic programming, the additional challenge being that input/output data samples are not available for the unknown optimal value function.

To compute an approximate solution to the infinite LP reformulation of the Bellman equation, in Section III an iterative Approximate LP (iALP) scheme is proposed where the optimization variable is expressed as a linear combination of basis functions in a given class. At each iteration, the basis functions are given and the resulting semi-infinite LP can be solved via constraint sampling. Through the iterations, the number of basis functions is increased by adding a suitably chosen basis function at each iteration. The iterative procedure is halted when some performance goal is reached or some upper bound on the number of basis functions is hit. From an implementation point of view, the solution to each scenario LP involves integral calculations, which may

This work is partly supported by the European Commission under project UnCoVerCPS (grant number 643921).
The authors are with Politecnico di Milano, Dipartimento di Elettronica, Informazione e Bioingegneria, Milano, Italy {alessandro.falsone,maria.prandini}@polimi.it

hamper the computational efficiency of the method. However, in certain cases, depending on the stochastic kernel governing the MDP evolution and the chosen basis functions, integrals admit an analytic solution, which makes the overall approach particularly appealing. This is the case of the multi-room heating system example in Section IV.

II. OPTIMAL CONTROL & LP REFORMULATION

Consider a discrete time MDP $H = (\mathcal{S},\mathcal{U},T_s)$, with state space $\mathcal{S}$, control space $\mathcal{U}$, and controlled transition probability function given by the stochastic kernel $T_s : \mathcal{B}(\mathcal{S}) \times \mathcal{S} \times \mathcal{U} \to [0,1]$ that assigns to each $s \in \mathcal{S}$ and $u \in \mathcal{U}$ a probability measure $T_s(\cdot|s,u)$ on the Borel space $(\mathcal{S},\mathcal{B}(\mathcal{S}))$. $T_s(\cdot|s,u)$ describes the next state conditional probability distribution given that the current state is s and the control input u is applied, and it is still well-defined when the state space $\mathcal{S}$ is hybrid, i.e., $\mathcal{S} = \mathcal{Q} \times \mathcal{X}$, where $\mathcal{Q} = \{q_1,\dots,q_m\}$ is a finite discrete state space and $\mathcal{X} = \mathbb{R}^n$ is a continuous state space, [18]. Given an MDP H, our goal is to design a state feedback control policy that optimizes some performance index when applied to H. To this purpose, we next recall the notion of Markov policy.

Definition 1 (Markov Policy): A Markov policy π for an MDP $H = (\mathcal{S},\mathcal{U},T_s)$ is a sequence $\pi = (\pi_0,\pi_1,\pi_2,\dots)$ of maps $\pi_k : \mathcal{S} \to \mathcal{U}$, $k = 0,1,2,\dots$, from the state space $\mathcal{S}$ to the control space $\mathcal{U}$. If all the maps π_k are identical, then, the policy is said to be stationary. □

We denote the set of stationary Markov policies as Π_m. In the sequel, we shall consider only stationary policies.

We can now formulate the control problem as that of determining a Markov policy $\pi^\star \in \Pi_m$ that maximizes the average discounted reward:

$$V^\pi(s) = E^\pi_s\left[\sum_{t=0}^{\infty}\gamma^t \ell\big(s(t),\mu(s(t))\big)\right], \tag{1}$$

where $\gamma \in (0,1)$ is the discount factor, $E^\pi_s[\cdot]$ denotes the expectation with respect to the probability measure P^π_s associated with the policy π and the initial condition $s(0) = s$, and $\ell : \mathcal{S}\times\mathcal{U} \to \mathbb{R}$ is the reward per stage function, which is assumed to be bounded. We shall next recall known facts regarding the dynamic programming solution to this control problem, [4].

The reward function (1) associated to a stationary Markov policy $\pi = (\mu,\mu,\mu,\dots) \in \Pi_m$ can be characterized through the recursive equation

$$V^\pi(s) = \ell(s,\mu(s)) + \gamma E_{s'\sim T_s(\cdot|s,\mu(s))}\left[V^\pi(s')\right],$$

where $E_{s'\sim T_s(\cdot|s,u)}[\cdot]$ denotes the expectation with respect to the conditional probability distribution $T_s(\cdot|s,u)$ for s': $E_{s'\sim T_s(\cdot|s,u)}\left[V^\pi(s')\right] = \int_{\mathcal{S}} V^\pi(s')T_s(ds'|s,u)$. The optimal value function $V^\star(s) = \sup_\pi V^\pi(s)$ is the unique fixed point to the *Bellman equation*: $V^\star(s) = \mathcal{T}[V^\star(s)]$, where $\mathcal{T}$ is called Bellman operator and is given by

$$\mathcal{T}\left[V(\cdot)\right](s) = \sup_{u\in\mathcal{U}}\{\ell(s,u) + \gamma E_{s'\sim T_s(\cdot|s,u)}[V(s')]\}.$$

An optimal policy $\mu^\star : \mathcal{S} \to \mathcal{U}$ can be computed from the optimal value function as follows

$$\mu^\star(s) \in \arg\sup_{u\in\mathcal{U}}\{\ell(s,u) + \gamma E_{s'\sim T_s(\cdot|s,u)}[V^\star(s')]\}.$$

Grid-based solutions to the problem of computing the optimal value function and the optimal control policy can be adopted when the state space dimension is low, [7], [19].

The problem of determining $V^\star(\cdot)$ can be rephrased as the following LP:

$$\min_{V(\cdot)} \int_{\mathcal{S}} \Psi(s)V(s)ds \tag{2}$$

subject to:

$$V(s) \geq \ell(s,u) + \gamma\int_{\mathcal{S}} T_s(s'|s,u)V(s'),\ \forall s \in \mathcal{S}, u \in \mathcal{U},$$

where $\Psi : \mathcal{S} \to \mathbb{R}_+$ is a weighting function on the state space $\mathcal{S}$. Indeed, a feasible solution $V(\cdot)$ to (2) satisfies: $V(s) \geq \mathcal{T}[V(\cdot)](s)$, $\forall s \in \mathcal{S}$, and, since the Bellman operator $\mathcal{T}$ is a monotonic contraction mapping, it holds that $V(s) \geq \mathcal{T}[V(\cdot)](s) \geq \mathcal{T}^2[V(\cdot)](s) \geq \dots \quad \dots \geq \mathcal{T}^n[V(\cdot)](s) \geq \dots \geq V^\star(s)$, $\forall s \in \mathcal{S}$, which shows that $V(\cdot)$ is an upper bound on $V^\star(\cdot)$, and, hence, minimizing $\int_{\mathcal{S}}\Psi(s)V(s)ds$ is equivalent to minimizing $\int_{\mathcal{S}}\Psi(s)|V(s)-V^\star(s)|ds$ which is the Ψ-weighted 1-norm of the error. This gives an interpretation of $\Psi(\cdot)$ as a state-relevance weight, which can be chosen so as to give higher weight to regions of the state space in which we would like a better approximation. If we set $\int_{\mathcal{S}}\Psi(s)ds = 1$, then $\Psi(\cdot)$ can be interpreted as a probability distribution over the state space $\mathcal{S}$, and $\int_{\mathcal{S}}\Psi(s)V(s)ds$ as the expected value of $V(\cdot)$ with respect to $\Psi(\cdot)$: $E_{s\sim\Psi(\cdot)}[V(s)]$.

The LP (2) is hard to solve since optimization is performed over an infinite dimensional functional space and the number of constraints is infinite. In the following section we shall provide an approximate solution to (2) for the case when the state $\mathcal{S}$ is hybrid, i.e., $\mathcal{S} = \mathcal{Q}\times\mathcal{X}$, under the assumption that

Assumption 1: The control space $\mathcal{U}$ is a finite set. □

This assumption is quite standard in practice, since when $\mathcal{U}$ is not finite some gridding is typically performed to find a numerical solution to the problem.

III. ALP SOLUTION

Given a finite set of pre-specified basis functions $\Phi_k = \{g_i : \mathcal{X} \to \mathbb{R}\}_{i=0}^k$ defined on the continuous state space $\mathcal{X}$, we approximate the optimization variable $V(\cdot)$ in (2) as follows:

$$V_k^w(q,x) = \sum_{i=0}^{k} w_{iq}g_i(x),\ (q,x) \in \mathcal{S}, \tag{3}$$

with the coefficients of the linear combination that depend on the discrete state component $q \in \mathcal{Q}$.

By plugging (3) into (2), and recalling that $\int_{\mathcal{S}}\Psi(s)V(s)ds = E_{s\sim\Psi(\cdot)}[V(s)]$, the resulting Approximate Linear Program (ALP) is given by:

$$\min_{\{w_{iq},\, i=0,\dots,k\}_{q\in\mathcal{Q}}} E_{(q,x)\sim\Psi(\cdot)}\left[\sum_{i=0}^{k} w_{iq}g_i(x)\right] \tag{4}$$

subject to:

$$\sum_{i=0}^{k} w_{iq}g_i(x) \geq \ell(s,u) + \gamma E_{(q',x')\sim T_s(\cdot|s,u)}\left[\sum_{i=0}^{k} w_{iq'}g_i(x')\right],$$

$\forall s = (q,x) \in \mathcal{S},\ u \in \mathcal{U}$.

Since the state space $\mathcal{S} = \mathcal{Q} \times \mathcal{X}$ has infinite cardinality, (4) is a semi-infinite LP problem and it is hard to solve. We then head for a suitable relaxation of the problem so as to make its solution computable. This relaxation consists in reducing the constraints to a finite number while retaining some (probabilistic) guarantees on the satisfaction of the constraints that are not considered when computing the solution. This is achieved through the so-called scenario approach [14], which is briefly explained next.

A. Scenario-based ALP

Consider the semi-infinite convex optimization problem:

$$\begin{aligned} \min_{w \in \mathbb{R}^d} \quad & c^T w \\ \text{subject to:} \quad & f_\delta(w) \leq 0, \ \forall \delta \in \Delta \end{aligned} \tag{5}$$

where δ is some uncertainty parameter taking values in a set Δ according to some probability distribution P. Let us formulate a relaxed version of problem (5):

$$\begin{aligned} \min_{w \in \mathbb{R}^d} \quad & c^T w \\ \text{subject to:} \quad & f_{\delta^{(i)}}(w) \leq 0, \ i = 1, 2, ..., N, \end{aligned} \tag{6}$$

where $\delta^{(i)}$, $i = 1, \dots, N$, are independently extracted at random from Δ according to P. Then, the following theorem holds.

Theorem 1 ([14]): Suppose that, for any $\delta \in \Delta$, $f_\delta(w)$ is convex as a function of $w \in \mathbb{R}^d$, and that problem (6) is feasible for every N. Denote as $w_N^\star$ its solution. Select a 'violation parameter' $\epsilon \in (0,1)$ and a 'confidence parameter' $\beta \in (0,1)$. If N satisfies:

$$\sum_{i=0}^{d-1} \binom{N}{i} \epsilon^i (1-\epsilon)^{N-i} \leq \beta, \tag{7}$$

where d is the number of optimization variables, then, with probability $1 - \beta$, $w_N^\star$ satisfies all constraints in (5) except for a fraction of probability at most ϵ. □

By interpreting $s \in \mathcal{S}$ as the uncertainty parameter δ distributed over $\Delta = \mathcal{S}$ according to $P = \Psi$, and including the constant function $g_0(x) = 1$, $x \in \mathcal{X}$, in the basis functions set $\Phi_k = \{g_i : \mathcal{X} \to \mathbb{R}, i = 0, \dots, k\}$ (so as to guarantee the feasibility condition), the ALP problem (4) can be approximately solved via Algorithm 1.

The quality of the solution obviously depends on the set Φ_k of basis functions that are chosen to approximate the optimal value function, which also has some impact on the computational effort involved in the ALP solution and, hence, its scalability. This issue is addressed next through an iterative approach.

B. Iterative ALP (iALP)

The optimization variable $V(\cdot)$ in (2) is approximated through $V_k^w(\cdot)$ in (3) which is a linear combination of the basis functions in the set $\Phi_k = \{g_i : \mathcal{X} \to \mathbb{R}\}_{i=0}^k$ of cardinality $k+1$. In this section, we propose to solve the problem of defining the set Φ_k, i.e., deciding how many basis functions and which basis functions to include, through an iterative procedure, where at each iteration j one basis function is added to Φ_j, starting from the set Φ_0 that contains only the constant function $g_0(x) = 1$, $x \in \mathcal{X}$, needed for the constrained LP problem (4) and its scenario version (8) to be always feasible.

Algorithm 1 Scenario-based ALP

```
Fix $\epsilon \in (0,1)$ and $\beta \in (0,1)$;
Compute $N$ satisfying (7) where $d = (k+1)|Q|$;
Extract $N$ values $s_1, \dots, s_N$ from $\mathcal{S}$ according to $\Psi(\cdot)$.
    Set $S_N = \{s_1, \dots, s_N\}$;
Solve the relaxed ALP problem:
    $$\min_{\{w_{iq},\, i=0,\dots,k\}_{q \in \mathcal{Q}}} E_{(q,x)\sim\Psi(\cdot)} \left[ \sum_{i=0}^k w_{iq} g_i(x) \right] \qquad (8)$$
    subject to:
    $$\sum_{i=0}^k w_{iq} g_i(x) \geq \ell(s,u)$$
    $$+ \gamma E_{(q',x')\sim T_s(\cdot|s,u)} \left[ \sum_{i=0}^k w_{iq'} g_i(x') \right],$$
    $$\forall s = (q,x) \in S_N,\ u \in \mathcal{U}.$$
```

More precisely, we consider a family of basis functions that are finitely parameterized via some meta-parameter vector $\boldsymbol{\vartheta}$: $\mathcal{F} = \{f(\cdot;\boldsymbol{\vartheta}) : \mathcal{X} \to \mathbb{R},\ \boldsymbol{\vartheta} \in \Theta\}$. For instance, $f(\cdot;\boldsymbol{\vartheta})$ can be a Gaussian basis function parameterized through its mean and covariance matrix.

The iterative procedure works as follows.

We set a maximum number K of basis functions to be added to Φ_0, and generate a sampled space S_N, with N computed according to Theorem 1, using K in place of k to determine d (i.e., $d = (K+1)|Q|$). The solution to (8) computed based on Φ_0 is considered as baseline.

At each iteration $j \leq K$, a new set of basis functions is constructed as follows:

$$\Phi_j(\boldsymbol{\vartheta}_j) = \{g_i : \mathcal{X} \to \mathbb{R}\}_{i=0}^{j-1} \cup \{f(\cdot, \boldsymbol{\vartheta}_j) : \mathcal{X} \to \mathbb{R}\},$$

where $g_i : \mathcal{X} \to \mathbb{R}$, $i = 1, \dots, j-1$, were defined in the previous $j-1$ iterations. The meta-parameter $\boldsymbol{\vartheta}_j$ of the newly introduced basis function is tuned at iteration j by solving the following optimization problem:

$$\min_{\boldsymbol{\vartheta}_j} H_j(\boldsymbol{\vartheta}_j), \tag{9}$$

with $H_j(\boldsymbol{\vartheta}_j)$ defined as follows

$$H_j(\boldsymbol{\vartheta}_j) = E_{(q,x)\sim\Psi(\cdot)} \left[\sum_{i=0}^{j-1} w_{iq}(\boldsymbol{\vartheta}_j) g_i(x) + w_{jq}(\boldsymbol{\vartheta}_j) f(x, \boldsymbol{\vartheta}_j) \right],$$

where the coefficients $w_{iq}(\boldsymbol{\vartheta}_j)$, $i = 0, 1, \dots, j$, $q \in \mathcal{Q}$, of the linear combination of basis functions depend on $\boldsymbol{\vartheta}_j$ since they are obtained by solving the scenario LP (8) with $\Phi_j(\boldsymbol{\vartheta}_j)$ as set of basis functions.

The optimization problem above is nonlinear, but since we are tuning only a single basis function, the tuning of $\boldsymbol{\vartheta}_j$ can be efficiently implemented by standard nonlinear optimization routines, possibly starting from an initial random guess. Once a solution $\boldsymbol{\vartheta}_j^\star$ is found, then, we set $g_j(\cdot) = f(\cdot;\boldsymbol{\vartheta}_j^\star)$.

The iterative procedure stops when either a certain condition related to the quality of the solution, evaluated possibly

in terms of the Bellman residual, is met, or the maximum number of iterations K is reached. When it does stop, a new set S_N of hybrid states is sampled (with N computed according to Theorem 1 using the actual number of basis function that were included) and (8) is solved once again so that the probabilistic guarantees given by the scenario theory are retained.
The iterative ALP (iALP) is summarized in Algorithm 2.

Algorithm 2 Iterative ALP

```
Fix $\epsilon \in (0,1)$, $\beta \in (0,1)$, and $K$;
Compute $N$ satisfying (7) where $d = (K+1)|Q|$;
Start from $\Phi_0$ containing the constant function;
Extract $N$ values $s_1, \dots, s_N$ from $\mathcal{S}$ according to $\Psi(\cdot)$.
  Set $S_N = \{s_1, \dots, s_N\}$;
$j \leftarrow 1$
repeat
    $\boldsymbol{\vartheta}_j^\star \in \arg\min_{\boldsymbol{\vartheta}_j} \; H_j(\boldsymbol{\vartheta}_j)$ involving the solution to
    (8) with $\Phi_j(\boldsymbol{\vartheta}_j)$ as set of basis functions;
    Set $\Phi_j = \Phi_{j-1} \cup \{f(\cdot, \boldsymbol{\vartheta}_j^\star) : \mathcal{X} \to \mathbb{R}\}$
    $j \leftarrow j+1$
until $j > K$ or the accuracy is met
$k \leftarrow j-1$
Run Algorithm 1 with the set of basis functions $\Phi_k$
```

IV. Numerical Example

We consider the multi-room heating system example that was originally proposed in [20] and further elaborated in [21], [18], [7]. The problem consists in regulating the temperatures of n rooms. Each room has one heater, but only one heater at a time can be active. The goal is to maintain the temperature of each room inside a prescribed range $[x_l, \; x_u]$. The optimal policy is a switching strategy that, at each time step, decides based on the current temperatures of all rooms and current status of the heaters whether heating a room or not, and, in the former case, which room should be heated.

The multi-room heating system can be modeled as an MDP process on a hybrid state space $\mathcal{S} = \mathcal{Q} \times \mathcal{X}$. The hybrid state $s = (q, x)$ is composed of a discrete state component $q \in \mathcal{Q} = \{1, \dots, n+1\}$ that specifies the room that is being heated (no room if $q = n+1$) and a continuous state component $x = (x_1, \dots, x_n) \in \mathcal{X} = \mathbb{R}^n$ that represents the average temperatures of all n rooms. The control space $\mathcal{U} = \mathcal{Q}$ is the set of all possible actions that correspond to heating the u^{th}-room if $u < n+1$ or none of them if $u = n+1$.

For a given discrete state q, the evolution of the continuous state component x_i is governed by the stochastic difference equation

$$x_i(k+1) = x_i(k) + \big[b_i(x_a - x_i(k)) + c_i h_i(k) + \sum_{j=1,\dots,n; j \neq i} a_{ij}(x_j(k) - x_i(k))\big]\Delta t + \eta_i(k), \; i = 1, \dots, n, \tag{10}$$

obtained by discretizing the corresponding continuous time dynamic, using the constant-step Euler-Maruyama method with discretization step Δt. Parameter a_{ij} represents the heat exchange coefficient between room i and room j, b_i is the heat loss rate of room i to the ambient, and c_i is the heat rate of the heater in the i^{th} room. They are all non-negative constants and are all normalized with respect to the average thermal capacity of room i. The term $h_i(k)$ is a boolean function that assumes the value 1 if $q(k) = i$ (*i.e.* when the i^{th} room is heated), and 0 otherwise. The ambient temperature x_a is considered to be constant, and the disturbance η_i affecting the temperature of room i is assumed to be a sequence of i.i.d. Gaussian random variables with zero mean and variance $\nu^2 \Delta t$, independent of η_j, $\forall j \neq i$.
System (10) defines on the Borel space $(\mathcal{X}, \mathcal{B}(\mathcal{X}))$ a Gaussian transition function $T_x : \mathcal{B}(\mathcal{X}) \times \mathcal{S} \to [0,1]$ given by

$$T_x(\,\cdot \,|\, (q,x)) = \mathcal{N}(\,\cdot \,|\, \Xi x + \Gamma(q), \nu^2 \Delta t I_n), \; (q,x) \in \mathcal{S}, \tag{11}$$

where Ξ is a square matrix of size n, $\Gamma(q)$ is an n-dimensional column vector, and I_n is the identity matrix of size n. Each elements of Ξ and $\Gamma(q)$ can be obtained from the parameters in (10).
The evolution of the discrete state is driven by the transition function $T_q \; : \; Q \times Q \times U \; \to \; [0,1]$, with $T_q(q'|q,u)$ representing the probability of a transition from $q \in \mathcal{Q}$ to $q' \in \mathcal{Q}$ when the control input u is applied. Here we take

$$T_q(q'|q,u) = \begin{cases} 1, & u = q = q' \\ 1-\alpha, & u \neq q, q' = q \\ \alpha, & u = q', q' \neq q \end{cases} , \tag{12}$$

with $\alpha \in [0,1]$ being the success probability of the thermostat control action. If a discrete transition from q to $q' \neq q$ occurs at time step k, then, the continuous state is reset according to the dynamic equations (10) with the discrete state set to q. Finally, using on (11) and (12), the transition probability function $T_s : \mathcal{B}(\mathcal{S}) \times \mathcal{S} \times \mathcal{U} \to [0,1]$ can be defined as

$$T_s(dx, q' \,|\, q, x, u) = T_x(dx \,|\, q', x) T_q(q' \,|\, q, u). \tag{13}$$

Let $A = \mathcal{Q} \times A_x$, with $A_x := \{(x_1, \dots, x_n) \in \mathbb{R}^n : x_i \in [x_l, x_u], \; i = 1, \dots, n\}$. Then, the reward function to be maximized can be expressed as

$$V^\pi(s_0) = E_{s_0}^\pi \left[\sum_{k=0}^{\infty} \gamma^k \rho(s(k), s(k+1)) \right], \tag{14}$$

where $\gamma \in (0,1)$ is the discount factor, and $\rho : \mathcal{S} \times \mathcal{S} \to \{-1, 0, 1\}$ is the reward per stage, which is set equal to

$$\rho(s,s') = \begin{cases} -1, & s \in A \text{ and } s' \notin A \\ 1, & s \notin A \text{ and } s' \in A \\ 0, & \text{otherwise} \end{cases}$$

(see [7] for more details).

By setting $\ell(s,u) = \int_{\mathcal{S}} \rho(s,s') T_s(ds'|s,u) ds$, the reward function (14) associated to a stationary Markov policy $\pi = (\mu, \mu, \mu, \dots) \in \Pi_m$ can be rewritten in the form (1).

A. Comparison between iALP and AVI

In order to compare the performance of the iALP solution with that of the Approximate Value Iteration (AVI) scheme described in [7], we applied both algorithms to the multi-room heating system problem. We considered two different setup: a) two rooms and b) three rooms, numbered sequentially in a row. As for the parameters entering the problem description, coefficients a_{ij} were set equal to 0.33 if either $j = i+1$ or $j = i-1$, and 0 otherwise. Coefficients b_i and c_i were set equal to 0.25 and 12, respectively, for all rooms. α was set to 0.8, $\Delta t = 1/10$, $x_a = 6$ and $\nu^2 = 1$. The "safe" temperature range for each room was set equal to $[x_l, x_u] = [17.5, 22]$, and the discount factor γ in (14) to 0.95. The parameters of the scenario approach were $\beta = 10^{-5}$ and $\epsilon = 0.01$, and the maximum number of basis functions in Algorithm 2 was $K = 20$, leading to a number of scenario realisations $N = 10632$ for the two-room case and $N = 13363$ for the three-room case. As for the AVI, we used $m = 40$ grid points within A_x for the two-room case and $m = 15$ for the three-room case.

Function $\Psi((q, \cdot))$ over $\mathcal{X}$ was set equal to a n-variate Gaussian density function with independent components, each one with mean $\mu_\Psi = \frac{x_l + x_u}{2}$ and variance $\sigma_\Psi^2 = 5$, for any $q \in Q$. As for functions $g_i : \mathcal{X} \to \mathbb{R}$ entering the approximate optimal value function expression in (3), we used n-variate Gaussian function with independent components and with support limited to either A_x or $\bar{A}_x$ depending on the fact that the basis function is used to approximate the optimal value function over A_x or $\bar{A}_x$. Additionally, we include the indicator functions over A_x and $\bar{A}_x$.

In the implementation of the iALP algorithm, in order to decide if taking a basis function with support limited to A_x or $\bar{A}_x$, we toss a coin. Since the optimal value function is much simpler over A_x, we set the probability of obtaining a basis function with support limited to A_x equal to 0.25.
As a stopping criterion we monitored the mean square Bellman residual (MSBR) computed over the scenario realisations, *i.e.*

$$\mathrm{MSBR}_j = \frac{1}{N} \sum_{s \in S_N} (V_j^{w^\star}(s) - \mathcal{T}[V_j^{w^\star}(s)])^2,$$

where $V_j^{w^\star}(\cdot)$ is the (approximate) optimal value function obtained via the scenario-based ALP at iteration j in Algorithm 2.

We next show the results obtained using the iALP procedure and compare them with those obtained with the AVI approach. We first present the two-room case, and then the three-room case.

In the two-room case, the iALP Algorithm stopped after two iterations only, assigning no basis function over A_x and 2 basis function over $\bar{A}_x$, besides the constant functions needed to ensure feasibility. In Figure 1 we can compare the optimal value function computed using the AVI scheme and that obtained using the scenario-based ALP. As expected, the ALP solution is an upper bound for the optimal value function (here considered to be the solution of the AVI algorithm as a reference). Interestingly, despite the fact that graphically the two functions are not so similar, the approximation found by the iALP approach achieves a quite small value for the mean square Bellman residual ($\mathrm{MSBR} = 3.95 \cdot 10^{-4}$ with $\frac{1}{N}\sum_{s \in S_N} V_j^{w^\star}(s)^2 = 0.5004$).

As for the control policies, we focus on the discrete state $q = 1$ and report in Figure 2 the iALP policy (left) and the AVI policy (right). Different colors represents different actions to execute when the system is in a particular state, in our case red is $u = 1$, green is $u = 2$ and blue is $u = 3$. The black rectangle represents the "safe" set. Despite the lack of representation capabilities due to the small number of basis function, the obtained policy captures the essential strategy to control the multi-room system. Notice that the policies are more similar close to the center of the "safe" set, and they are more different outside the "safe" set. Yet, the iALP policy reproduces the essential behavior. To better justify this statement, we run a set of Monte Carlo simulations. Each simulation consists of two trajectories that start from the same random point in the hybrid state space and then follows the two policies while being subject to the same disturbances realizations. For each run and for both simulations per run, we recorded the reward collected by the system along a 300-step time horizon and computed the empirical estimate of the expected reward. The number $N_{\mathrm{mc}} = 57565$ of runs was derived from Hoeffding's inequality, $N_{\mathrm{mc}} > \frac{2}{\epsilon^2} \log\left(\frac{1}{\delta}\right)$, so as to guarantee an accuracy $\epsilon = 0.01$ of the estimated reward with confidence $1 - \delta = 1 - 10^{-5}$. The obtained empirical mean is reported in Table I under the "2-rooms" column.

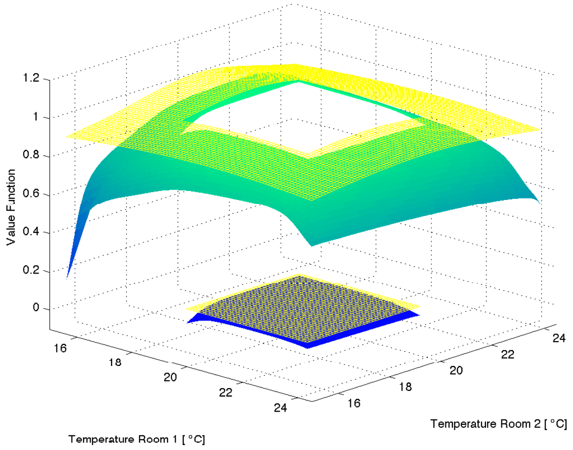

Fig. 1. Value function. iALP: gridded yellow. AVI: blue to green.

In the 3-room case, the iALP algorithm started from a situation where only the two constant functions were present, and stopped after including 10 basis functions: 1 over A_x and 9 over $\bar{A}_x$. The mean square Bellman residual was $\mathrm{MSBR} = 3.65 \cdot 10^{-4}$ with $\frac{1}{N}\sum_{s \in S_N} V_j^{w^\star}(s)^2 = 0.6395$, revealing again that the obtained solution almost satisfies the Bellman equation. A comparative evaluation between the iALP and AVI policies was performed via Monte Carlo simulations and is reported in the "3-rooms" column of Table I. A sample of the simulated temperature trajectories is reported in Figure 3, which shows that there is no apparent difference between the two approaches in controlling the system.

	2-rooms	3-rooms
iALP	0.6718	0.6853
AVI	0.6785	0.6873

TABLE I
iALP AND AVI PERFORMANCE COMPARISON.

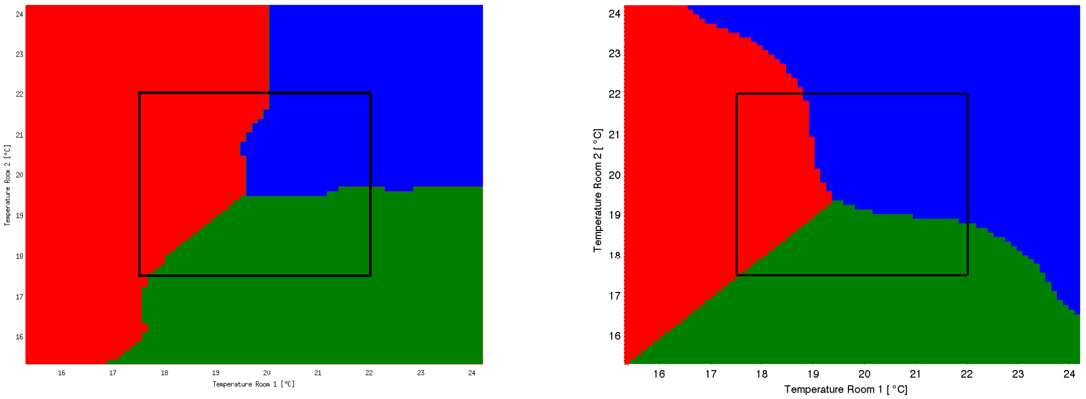

Fig. 2. Optimal policy - Left: iALP. Right: AVI.

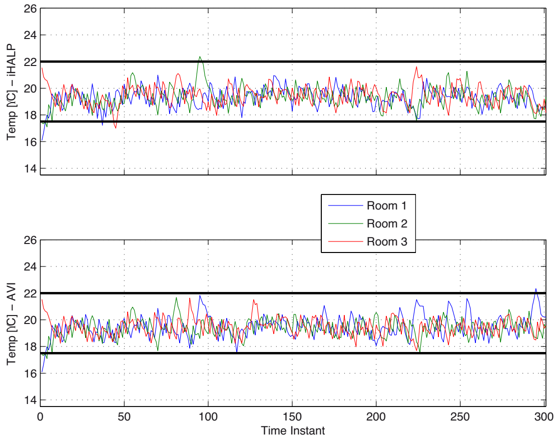

Fig. 3. System evolution in an MC trial - Top: iALP. Bottom: AVI.

V. CONCLUSIONS

In this paper, we propose an iterative approximate linear programming approach to the solution of an optimal control problem for an MDP with a hybrid state space and a discrete control space. Iterations are introduced to progressively define the set of basis functions that is used to approximate the optimal value function. A randomized approach is adopted at each iteration to overcome the curse of dimensionality issue due to the presence of a continuous component of the state. Preliminary results obtained in a multi-room heating system example appear promising. Still simulation results were limited to low dimensional instances and further tests are needed to assess the scalability of the approach.

REFERENCES

[1] J. Lygeros and M. Prandini, "Stochastic hybrid systems: a powerful framework for complex, large scale applications," *European Journal of Control*, vol. 16, no. 6, pp. 583–594, 2010.

[2] C. H. Papadimitriou, *Computational complexity*, C. H. Papadimitriou, Ed. Addison-Wesley, 1994.

[3] L. Busoniu, R. Babuska, B. De Schutter, and D. Ernst, *Reinforcement Learning and Dynamic Programming Using Function Approximators*.

[4] D. Bertsekas, *Dynamic Programming and Optimal Control*. Athena Scientific, 1995.

[5] A. Abate, S. Amin, M. Prandini, J. Lygeros, and S. Sastry, "Computational approaches to reachability analysis of stochastic hybrid systems," in *Hybrid Systems: Computation and Control*, ser. Lecture Notes in Computer Sciences, A. Bemporad, A. Bicchi, and G. Buttazzo, Eds. Berlin: Springer-Verlag, 2007, vol. 4416, pp. 4–17.

[6] A. Abate, J. Katoen, J. Lygeros, and M. Prandini, "Approximate model checking of stochastic hybrid systems," *European Journal of Control*, vol. 16, no. 6, pp. 624–641, 2010.

[7] M. Prandini and L. Piroddi, "A self-recovery approach to the probabilistic invariance problem for stochastic hybrid systems," in *51st IEEE Conference on Decision and Control*, 2012.

[8] A. Petretti and M. Prandini, "An approximate linear programming solution to the probabilistic invariance problem for stochastic hybrid systems," in *53rd IEEE Conference on Decision and Control*, Los Angeles, USA.

[9] N. Kariotoglou, S. Summers, T. Summers, M. Kamgarpour, and J. Lygeros, "Approximate dynamic programming for stochastic reachability," in *European Control Conference*, 2013.

[10] M. Prandini, S. Garatti, and R. Vignali, "Performance assessment and design of abstracted models for stochastic hybrid systems through a randomized approach," *Automatica*, vol. 50, no. 11, pp. 2852–2860, 2014.

[11] D. de Farias and B. V. Roy, "The linear programming approach to approximate dynamic programming," *Oper. Res.*, vol. 51, no. 6, pp. 850–856, 2003.

[12] G. Calafiore and M. Campi, "Uncertain convex programs: randomized solutions and confidence levels," *Mathematical Programming*, vol. 102, no. 1, pp. 25–46, 2005.

[13] ——, "The scenario approach to robust control design," *IEEE Transactions on Automatic Control*, vol. 51, no. 5, pp. 742–753, 2006.

[14] M. Campi, S. Garatti, and M. Prandini, "The scenario approach for systems and control design," *Annual Reviews in Control*, vol. 33, no. 2, pp. 149–157, 2009.

[15] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of Control, Signals and Systems*, vol. 2, no. 4, pp. 303–314, 1989. [Online]. Available: http://dx.doi.org/10.1007/BF02551274

[16] J. Park and I. W. Sandberg, "Universal approximation using radial-basis-function networks," *Neural Comput.*, vol. 3, no. 2, pp. 246–257, Jun. 1991.

[17] L. Breiman, "Hinging hyperplanes for regression, classification, and function approximation," *IEEE Transactions on Information Theory*, vol. 39, no. 3, pp. 999–1013, May 1993.

[18] A. Abate, M. Prandini, J. Lygeros, and S. Sastry, "Probabilistic reachability and safety for controlled discrete time stochastic hybrid systems," *Automatica*, vol. 44, no. 11, pp. 2724–2734, 2008.

[19] W. Powell, *Approximate Dynamic Programming: Solving the Curses of Dimensionality*. John Wiley & Sons, 2007.

[20] A. Fehnker and F. Ivančić, "Benchmarks for hybrid systems verifications," in *Hybrid Systems: Computation and Control*, ser. LNCS 2993, R. Alur and G. Pappas, Eds. Springer Verlag, 2004, pp. 326–341.

[21] A. Abate, S. Amin, M. Prandini, J. Lygeros, and S. Sastry, "Computational approaches to reachability analysis of stochastic hybrid systems," in *Hybrid Systems: Computation and Control*, ser. Lecture Notes in Computer Science 4416, A. Bemporad, A. Bicchi, and G. Buttazzo, Eds. Springer Verlag, 2007, pp. 4–17.