Department of Electrical and Computer Engineering Technical Reports

Department of Electrical and Computer Engineering

2-5-2018

Approximate Dynamic Programming for Building Control Problems with Occupant Interactions

Donghwan Lee
Illinois, donghwan@illinois.edu

Seungjae Lee
Lyles School of Civil Engineering, Purdue University, West Lafayette, Indiana USA / Center for High Performance Buildings, Ray W. Herrick Laboratories, Purdue University, West Lafayette, Indiana USA, lee1904@purdue.edu

Panagiota Karava
School of Civil Engineering and Division of Construction Engineering and Management, Purdue University, United States of America, pkarava@purdue.edu

Jianghai Hu
Purdue University, jianghai@purdue.edu

Lee, Donghwan; Lee, Seungjae; Karava, Panagiota; and Hu, Jianghai, "Approximate Dynamic Programming for Building Control Problems with Occupant Interactions" (2018). *Department of Electrical and Computer Engineering Technical Reports.* Paper 489.
https://docs.lib.purdue.edu/ecetr/489

Approximate Dynamic Programming for Building Control Problems with Occupant Interactions

Donghwan Lee, Seungjae Lee, Panagiota Karava, and Jianghai Hu[*][†][‡][§]

February 5, 2018

Abstract

The goal of this paper is to study potential applicability and performance of approximate dynamic programming (ADP) for building control problems. It is well known that occupants' stochastic behavior affects the thermal dynamics of building spaces. Incorporating occupant interactions in building control system designs is the main focus of this work. We apply ADP to stochastic optimal control designs for illustrative scenarios of occupant-building interactions and demonstrate its validity through a simulation study.

1 Introduction

Our goal is to study stochastic optimal control designs and its applicability to building climate control scenarios, important stochastic control applications. Recently, there has been a lot of interests in energy consumption and comfort management in buildings [1]. Its main goal is to balance between the energy consumption and occupants' comfort in work environments. The presence of stochastic uncertainties and disturbances, such as weather and occupant interactions, is a major concern in building environment research as they degrade the performance of the control systems.

This paper considers the building control problem with a particular focus on building-occupant interactions. The role of occupants is significant in the thermal dynamics of building spaces [2–5]. In particular, the thermal preferences of occupants induce their actions, which potentially perturb the thermal dynamics of building spaces. It is a special class of stochastic systems in the sense that the statistical behavior of the occupant's actions interact with the system evolution: occupant thermal preference models [6–8] depend on environmental factors, for example, the indoor air temperature. For this reason, developments of effective stochastic control methods become of prime importance.

Literature Review: Model predictive control (MPC) is a popular optimal control scheme in the presence of various constraints and objectives. For this reason, it has been widely used for building

[*] This material is based upon work supported by the National Science Foundation under Grant No. 1539527

[†] D. Lee is with the Department of Mechanical Science and Engineering, University of Illinois, Urbana-Champaign, IL 61801, USA lee1923@purdue.edu,donghwan@illinois.edu.

[‡] S. Lee and P. Karava are with the Department of Civil Engineering, Purdue University, West Lafayette, IN 47906, USA lee1904@purdue.edu, pkarava@purdue.edu.

[§] J. Hu is with the Department of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47906, USA jianghai@purdue.edu.

control problems [9, 10]. However, MPC uses predictions of system's future output trajectories; thereby, its performance is sensitive to uncertainties. To overcome this difficulty, stochastic MPC (SMPC) has been studied for building controls problems [11–14]. Many SMPC approaches assume that the system disturbances are Gaussian. However, Gaussian disturbances cannot describe more complicated behavior of real-world systems, and it is of great importance to develop optimal control designs for systems with more generic stochastic disturbances. In this respect, occupant models based on Markov chains have been studied in [3, 5] for building control problems. Scenario-based (or sample-based) MPC [15,16] can be applied to cope with generic non-Gaussian stochastic disturbances. The approach was successfully applied to many engineering applications, for instance, robot path-planning problems in [15] and the aircraft conflict detection in [17]. For the building problems, the scenario approach was investigated in [18], where samples of the external temperature, solar radiation, and room occupancy are generated by using an empirical statistic model. Approximate dynamic programming (ADP) [19] (or reinforcement learning (RL) [20] from the machine learning context) is another possibility. For building applications, ADP was studied in several researches, for instance [21–24].

Challenges: In building environment research, advanced occupant thermal preference models have been developed, e.g., [6–8], where occupant's thermal preferences are expressed as probability mass functions that depend on environmental factors, for example, the indoor air temperature. However, the existing results did not consider such occupant behavior models that interacts with the building dynamics. The optimal control of such stochastic systems cannot be formulated as easily solvable optimization problems. For instance, the scenario-based control design schemes are problematic in this case. Such cases arise in many applications, for example, hybrid electric vehicle powertrain management problems [25, 26]. One possible approach to solve such complex optimal control problems is to use ADP. For building control problems, ADP has been studied in several researches, for instance [21–24, 27, 28], to find a balance among energy savings, high comfort, and indoor air quality. However, the previous studies did not consider occupant interactions.

Statement of contributions: Motivated by the observations, the first contribution is to study dynamic programming (DP) for special stochastic systems, where the continuous and discrete state-spaces coexist and interact with each other. Such systems include the thermal dynamics of building spaces with occupants. We prove the convergence of the DP. The proposed results build upon the previous work [25], where a DP was investigated for a similar class of stochastic systems. Compared to that of [25], the proposed DP can handle systems with additional random disturbances in the continuous state-space. Another contribution is an application of the RL to building control problems with occupant interactions based on the developed DP. RL[1] is a family of unsupervised learning schemes for agents interacting with unknown environment, and has been widely studied in [29–34]. We assume that a stochastic model of occupant behavior is given and present illustrative scenarios where RL can be applied to building control systems with occupant interactions, assessing potential of RL in those cases. Finally, we note that a disadvantage of the proposed DP method compared to scenario-based MPCs is its inability to deal with constraints, which needs to be further studied in future researches.

[1] RL can be regarded as a class of ADPs.

2 Preliminaries

Throughout the paper, the following notations will be used: $\mathbb{N}$ and $\mathbb{N}_+$: sets of nonnegative and positive integers, respectively; $\mathbb{R}$: set of real numbers; $\mathbb{R}^n$: n-dimensional Euclidean space; $\mathbb{R}^{n\times m}$: set of all $n \times m$ real matrices; A^T: transpose of matrix A; $|S|$: cardinality of finite set S; $\mathbb{E}[\cdot]$: expectation operator; $\mathbb{P}[\cdot]$: probability of event.

Consider the discrete-time stochastic system

$$\begin{aligned} &\mathbf{x}(k+1) = f(\mathbf{x}(k), \mathbf{u}(k), \mathbf{w}(k), \mathbf{z}(k)), \quad \mathbf{x}(0) = x_0 \in X, \\ &\mathbf{z}(k+1) \sim p(\mathbf{x}(k)), \quad \mathbf{z}(0) \sim \mu, \end{aligned} \tag{1}$$

where $k \in \mathbb{N}$ is the time step, $\mathbf{x}(k) \in X$ is the state, $X \subset \mathbb{R}^n$ is a compact state space, $\mathbf{u}(k) \in U$ is the control input, $U \subset \mathbb{R}^{m_u}$ is a compact control space, $\mathbf{w}(k) \in \mathbb{R}^{m_w}$ is a random variable representing disturbances and uncertainties, and each $\mathbf{w}(k)$ is independent of other random variables, and $(\mathbf{z}(k))_{k=0}^{\infty}$, is a stochastic process with finite states $S := \{1, 2, \ldots, |S|\}$. $\mathbf{z}(0) \sim \mu$ implies $\mathbb{P}[\mathbf{z}(0) = i] = \mu_i$, and $\mathbf{z}(k+1) \sim p(\mathbf{x}(k))$ implies that the stochastic process $\mathbf{z}(k), k \in \mathbb{N}_+$, evolves according to $\mathbb{P}[\mathbf{z}(k+1) = i|\mathbf{x}(k) = x(k)] = p_i(x(k))$ with the transition probability $p(x(k)) := \begin{bmatrix} p_1(x(k)) & \ldots & p_{|S|}(x(k)) \end{bmatrix}^T$, and $x(k)$ is a realization of $\mathbf{x}(k)$. In other words, the transition probability depends on the current state $x(k)$ of (1).

We note that $(\mathbf{z}(k))_{k=0}^{\infty}$ is a special case of Markov chain. However, the first convergence result of dynamic programming (DP) in this paper holds for the general Markov chain case. Note that (1) is a Markov decision process (MDP) [19], where the continuous and discrete state-spaces coexist and interact with each other.

3 Dynamic Programming

Consider the process $(\mathbf{x}(k))_{k=0}^{\tau(x_0,z_0;\pi)}$, where $\tau(x_0, z_0; \pi)$ is the first time instant the trajectory $\mathbf{x}(k)$ exits X given $\mathbf{x}(0) = x_0$ and $\mathbf{z}(0) = z_0$ under a policy π. For a given nonnegative Lebesgue measurable stage cost function, $g : \mathbb{R}^n \times \mathbb{R}^m \times S \to \mathbb{R}_+$, and control input space $U \subset \mathbb{R}^m$, the cost associated with a given admissible state-feedback control policy $\pi : X \times S \to U$ and initial states $x_0 \in X$, $z_0 \in S$, is

$$J^{\pi}(x_0, z_0) := \mathbb{E}_{x_0,z_0}\left[\sum_{i=0}^{\tau(x_0,z_0;\pi)-1} \alpha^i g(\mathbf{x}(i), \mathbf{u}(i), \mathbf{z}(i))\right], \tag{2}$$

where $\mathbf{u}(i) = \pi(\mathbf{x}(i), \mathbf{z}(i))$, $\alpha \in [0, 1)$ is called the discount factor and $\mathbb{E}_{x_0,z_0}[\cdot]$ is a shorthand notation for $\mathbb{E}[\cdot|\mathbf{x}(0) = x_0, \mathbf{z}(0) = z_0]$. The expectation in (2) is with respect to the trajectory $(\mathbf{x}(i), \mathbf{u}(i), \mathbf{z}(i))_{i=0}^{\tau(x_0,z_0;\pi)-1}$ and $\tau(x_0, z_0; \pi)$.

Remark 1. *In this paper, a compact state space $X \subset \mathbb{R}^n$ is considered. The stage cost g is summed over a time interval where the state trajectory stays inside X. However, we assume that state trajectories of the system (1) outside of X are also defined well for convenience. For simplicity, we also assume that once the state exists X, it never comes back. In [25], a similar cost function was used, where a constraint set is used instead of the entire state space X. The DP in [25] tried to maximize the exit time by incorporating it into the cost function. In this paper, even though the constraint is not considered, the exit time regarding X is still required. By considering the compact* *state space X instead of $\mathbb{R}^n$, the cost function can be more easily approximated by using function approximations, for instance, the linear function approximation [33]. Moreover, the compact state space guarantees the boundedness of the stage cost function g under some mild conditions, for instance, the continuity of g.*

The set of all admissible state-feedback control policies is denoted by Π. In addition, we make the following standard assumption.

Assumption 1. *The cost per stage g satisfies $|g(x,u,i)| \le M$ for all $(x,u,i) \in X \times U \times S$, where M is some scalar.*

Under Assumption 1, the quantity (2) is always finite, and hence well defined. The *optimal cost* is $J^*(x_0,z_0) := \inf_{\pi\in\Pi} J^\pi(x_0,z_0)$. For any bounded function, define the operator

$$(TJ)(x_0,z_0) := \mathbb{I}_X(x_0) \inf_{u\in U} \mathbb{E}_{x_0,z_0}\left[g(x_0,u,z_0) + \alpha \mathbb{I}_X(\mathbf{x}(1)) J(\mathbf{x}(1),\mathbf{z}(1))\right], \tag{3}$$

where $\mathbf{x}(1) = f(x_0,u,\mathbf{w}(0),z_0)$ and $\mathbb{I}_X$ is the indicator function, i.e., $\mathbb{I}_X(x) = 1$ if $x \in X$ and $\mathbb{I}_X(x) = 0$ otherwise for any set $X \subseteq \mathbb{R}^n$. The indicator function is multiplied in (2) so that whenever $\mathbf{x}(1) \notin X$, the second term vanishes. This will be used to prove the convergence of the proposed DP to the optimal cost J^*. The optimal cost J^* satisfies $TJ^* = J^*$, called Bellman equation, and the sequence $(J_k)_{k=0}^\infty$ generated by the dynamic programming (DP) algorithm (value iteration), $J_{k+1} = TJ_k$, $J_0 \equiv 0$, converges uniformly to J^* under Assumption 1.

Theorem 1. *The sequence $(J_k)_{k=0}^\infty$ generated by the DP algorithm*

$$J_{k+1}(x_0,z_0) = (TJ_k)(x_0,z_0), \quad (x_0,z_0) \in X \times S$$

with $J_0 \equiv 0$ converges to J^*.

Proof. See Appendix I. □

Remark 2. *Note that $(\mathbf{z}(k))_{k=0}^\infty$ is a special case of Markov chains, and Theorem 1 can be directly applied to the more general case where $(\mathbf{z}(k))_{k=0}^\infty$ is a Markov chain. A convergence result of DP for MDP, where continuous and discrete state-spaces coexist and interact with each other, was addressed in [25, Theorem 2,Theorem 3]. However, the proof in [25] cannot be directly applied to our case because for the system in (1), the MDP has stochastic disturbances in continuous state spaces.*

If J^* is known, then the optimal state-feedback control policy can be computed as

$$u^*(x_0,z_0) := \arg\inf_{u\in U} \mathbb{E}_{x_0,z_0}\left[g(x_0,u,z_0) + \alpha \mathbb{I}_X(\mathbf{x}(1)) J^*(\mathbf{x}(1),\mathbf{z}(1))\right] \tag{4}$$

provided that the infimum is attained, where $\mathbf{x}(1) = f(x_0,u,\mathbf{w}(0),z_0)$. Moreover, Q-factor [35] is defined as

$$Q^*(x_0,z_0,u) := \mathbb{E}_{x_0,z_0}\left[g(x_0,u,z_0) + \alpha \mathbb{I}_X(\mathbf{x}(1)) J^*(\mathbf{x}(1),\mathbf{z}(1))\right], \tag{5}$$

where $\mathbf{x}(1) = f(x_0,u,\mathbf{w}(0),z_0)$. By comparing this definition with (4), the optimal policy can be expressed as $u^*(x_0,z_0) := \arg\inf_{u\in U} Q^*(x_0,z_0,u)$. In addition, one has $J^*(x_0,z_0) = \inf_{u\in U} Q^*(x_0,z_0,u)$. Similarly to T, if we define the operator F

$$(FQ)(x_0,z_0,u) := \mathbb{E}_{x_0,z_0}\left[g(x_0,u,z_0) + \alpha \mathbb{I}_X(\mathbf{x}(1)) \inf_{\bar{u}\in U} Q(\mathbf{x}(1),\mathbf{z}(1),\bar{u})\right],$$

then, (5) can be written as $Q^* = FQ^*$, which is equivalent to the Bellman equation ($TJ^* = J^*$). The Q-value iteration, $Q_{k+1} = FQ_k, Q_0 \equiv 0$, generates sequence $(Q_k)_{k=0}^{\infty}$ that converges to Q^* under the same condition as in the DP. In the building control problem of our interest, $\mathbf{z}(k)$ describes occupant thermal preferences, which may not be measured easily. Therefore, it is practical to assume that $\mathbf{z}(k)$ is not available in real time.

Assumption 2. *$\mathbf{x}(k)$ is measured in real time, but $\mathbf{z}(k)$ cannot be measured.*

To design an optimal control policy under Assumption 2, (2) is modified as

$$J^{\pi}(x_0) := \mathbb{E}_{x_0}\left[\sum_{k=0}^{\tau(x_0;\pi)-1} \alpha^k g(\mathbf{x}(k), \mathbf{u}(k), \mathbf{z}(k))\right],$$

where $\mathbf{z}(0) \sim \mu$, $\tau(x_0;\pi)$ is the first time instant the trajectory $\mathbf{x}(k)$ exits X given $\mathbf{x}(0) = x_0$ and under the policy π. Consider the optimal cost

$$J^*(x) := \inf_{\pi \in \Pi} J^{\pi}(x). \tag{6}$$

In this case, the operator (3) and the Bellman equation cannot be well formed because the next state evolution cannot be entirely determined based on the current state information, i.e., the Markov property does not hold. However, we can construct an augmented system that satisfies the Markov property. In particular, Figure 1 shows a graph which describes the dependencies of random variables. From the figure, it is clear that the augmented state vector $\tilde{\mathbf{x}}(k) = \begin{bmatrix}\mathbf{x}(k) & \mathbf{x}(k+1)\end{bmatrix}^T$ has enough information to determine the distributions of $\mathbf{x}(k+2)$. Define $\tilde{\mathbf{w}}(k) := \mathbf{w}(k+1), \tilde{\mathbf{z}}(k) := \mathbf{z}(k+1), \tilde{\mathbf{u}}(k) := \mathbf{u}(k+1), k \in \{0, 1, \ldots\}$, and define the corresponding stage cost such that $\tilde{g}(\tilde{\mathbf{x}}(k), \tilde{\mathbf{u}}(k), \tilde{\mathbf{z}}(k)) = g(\mathbf{x}(k+1), \mathbf{u}(k+1), \mathbf{z}(k+1)), k \in \{0,\ 1, \ldots\}$. Then, the augmented state satisfies

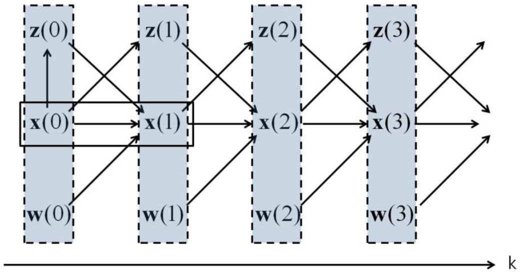


Figure 1: Graph describing dependencies of random variables.

$$\tilde{\mathbf{x}}(k+1) = \tilde{f}(\tilde{\mathbf{x}}(k), \tilde{\mathbf{u}}(k), \tilde{\mathbf{w}}(k), \tilde{\mathbf{z}}(k)) := \begin{bmatrix} \mathbf{x}(k+1) \\ f(\mathbf{x}(k+1), \mathbf{u}(k+1), \mathbf{w}(k+1), \mathbf{z}(k+1)) \end{bmatrix}.$$

Define the corresponding cost function

$$\tilde{J}^{\pi}(\tilde{x}_0) := \mathbb{E}_{\tilde{x}_0}\left[\sum_{k=0}^{\tau(\tilde{x}_0;\pi)-1} \alpha^k \tilde{g}(\tilde{\mathbf{x}}(k), \tilde{\mathbf{u}}(k),\ \tilde{\mathbf{z}}(k))\right],$$

where $\tilde{x}_0 \in X \times X$, and $\tilde{J}^*(\tilde{x}_0) := \inf_{\pi \in \Pi} \tilde{J}^\pi(\tilde{x}_0)$. If the distribution of $\mathbf{z}(k+1)$ depends only on partial coordinates of $\mathbf{x}(k)$, i.e., $P\mathbf{x}(k)$ where P is a projection matrix that projects onto the partial coordinates, then the augmented state can be replaced with $\tilde{\mathbf{x}}(k) = \begin{bmatrix} P\mathbf{x}(k) & \mathbf{x}(k+1) \end{bmatrix}^T$, and the augmented system can be defined as

$$\tilde{\mathbf{x}}(k+1) = \tilde{f}(\tilde{\mathbf{x}}(k), \tilde{\mathbf{u}}(k), \tilde{\mathbf{w}}(k), \tilde{z}(k)) := \begin{bmatrix} P\mathbf{x}(k+1) \\ f(\mathbf{x}(k+1), \mathbf{u}(k+1), \mathbf{w}(k+1), \mathbf{z}(k+1)) \end{bmatrix}.$$

Now, we obtain a system of the form (1), and the result in Theorem 1 can be directly applied.

4 Building Control with Occupant Interactions

4.1 Building Model

In this paper, we consider a 3m × 3m private office space with a 2.5m^2 south facing window, and its RC (resistor-capacitor) circuit analogy is given in Figure 2. To reduce the order of the model, we use one node for air in the room and another node collecting all the thermal mass in the room, where T_a is the indoor air temperature ($^\circ C$), T_o is the outdoor air temperature ($^\circ C$), T_w is the temperature of the aggregated mass node ($^\circ C$), q_solar is the solar radiation (W), q_internal is the internal heat (W), q_HVAC is the heating/cooling rate of the HVAC system (W). All notations which will be used in this section are summarized in Table 2. We assume that the room is conditioned by a VAV system so that q_HVAC directly affects T_a. Since we use low order model, we assume that the air node includes some portion of surfaces in the room which absorb radiative heat and release the heat quickly to the air. To determine appropriate values of the parameters of the circuit, we conducted a building energy simulation with EnergyPlus 8.7.0 in [36], and estimated the parameters minimizing the root-mean-square error between the air temperatures calculated by the EnergyPlus simulation and the low order model. The values of parameters are summarized in Table 1. Moreover, all notations used to describe the building system are presented in Table 2.

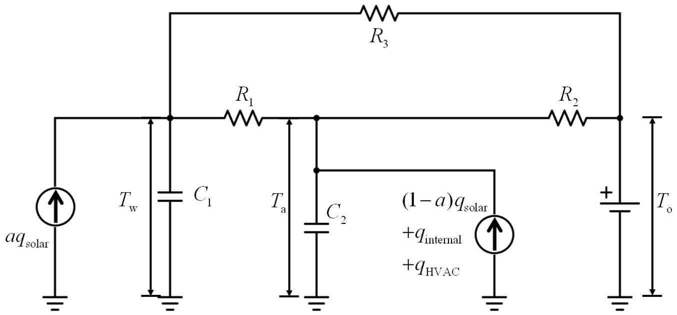


Figure 2: RC circuit analogy

The dynamic system model is given as

$$C_2\dot{T}_\text{a}(t) = \frac{T_\text{o}(t) - T_\text{a}(t)}{R_2} + \frac{T_\text{w}(t) - T_\text{a}(t)}{R_1} + (1-a)q_\text{solar}(t) + q_\text{HVAC}(t) + q_\text{internal}(t),$$
$$C_1\dot{T}_\text{w}(t) = \frac{T_a(t) - T_w(t)}{R_1} + \frac{T_\text{o}(t) - T_\text{w}(t)}{R_3} + aq_\text{solar}(t).$$

Table 1: Values of the parameters of the circuit in Figure 2

Parameter	Value	Unit
R_1	0.0084197	$°C/W$
R_2	0.044014	$°C/W$
R_3	4.38	$°C/W$
C_1	9861100	$J/°C$
C_2	128560	$J/°C$
a	0.55	–

Table 2: Notations

Notation	Meaning
T_a	Indoor air temperature ($°C$)
T_o	Outdoor air temperature ($°C$)
T_w	Temperature of the aggregated mass node ($°C$)
q_solar	Solar radiation (W)
q_internal	Internal heat (W)
q_HVAC	Heating/cooling rate of the HVAC system (W)
Δt	Sampling time (min)
T_ref	Current reference signal ($°C$)
M	Occupant's control input ($°C$)

A discrete time representation can be obtained by using the Euler discretization with a sampling time of Δt

$$\begin{aligned}
T_\mathrm{a}(k+1) - T_\mathrm{a}(k) =& \frac{\Delta t}{C_2 R_2}(T_\mathrm{o}(k) - T_\mathrm{a}(k)) + \frac{\Delta t}{C_2 R_1}(T_\mathrm{w}(k) - T_\mathrm{a}(k)) + \frac{\Delta t(1-a)}{C_2} q_\mathrm{solar}(k) \\
&+ \frac{\Delta t}{C_2} q_\mathrm{HVAC}(k) + \frac{\Delta t}{C_2} q_\mathrm{internal}(k), \\
T_\mathrm{w}(k+1) - T_\mathrm{w}(k) =& \frac{\Delta t}{C_1 R_1}(T_\mathrm{a}(k) - T_\mathrm{w}(k)) + \frac{\Delta t}{C_1 R_3}(T_\mathrm{o}(k) - T_\mathrm{w}(k)) + \frac{\Delta t a}{C_1} q_\mathrm{solar}(k),
\end{aligned}$$

where $k \in \mathbb{N}$ is the discrete time step. In this paper, we consider $\Delta t = 10$min sampling time. In the building control literature, the time step is usually chosen to be $\Delta t = 30$min. The reason we consider finer time steps is for quicker responses to occupant's actions.

4.2 Occupant Model

We assume that there is an occupant in the room, and the occupant's stochastic behavior affects the system dynamics. In particular, define the stochastic process $(\mathbf{z}(k))_{k=0}^{\infty}$ with the state space $S = \{1, 2, 3\}$, which represents the occupant's feeling of cold, comfort, and hot, respectively. Its probability depends on the current indoor temperature $T_\mathrm{a}(k)$, and its probability mass function $p_\mathbf{z}(z; T_\mathrm{a})$ is obtained by the Bayesian modelling approach in [8]. The values of the probability for different T_a are depicted in Figure 3. Consider some probability space $(\Omega, \mathcal{F}, \mathbb{P})$. Let $\mathcal{A}$ be a

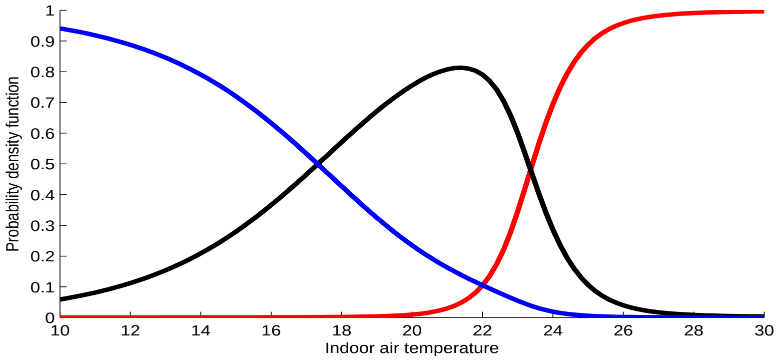


Figure 3: The probability mass function $p_{\mathbf{z}}(1; T_{\rm a})$ (blue), $p_{\mathbf{z}}(2; T_{\rm a})$ (black), $p_{\mathbf{z}}(3; T_{\rm a})$ (red) for different $T_{\rm a}$

space of occupant's actions, and let $\mathcal{I}$ be some information space. The information space $\mathcal{I}$ is a set of variables that affect occupant actions. For example, the values of $\mathbf{z}(k)$ can be an element of $\mathcal{I}$ because it is used to induce occupant actions. The occupant's actions are modelled as a map $M : \mathcal{I} \times \Omega \to \mathcal{A}$. We consider two possible scenarios of occupant's actions described as follows.

Occupancy (M_1): The occupant arrives at the room at time $\mathbf{w}_1$ uniformly distributed within $\{48, \ldots, 54\}$ (between 8am and 9am), and leaves the room at time $\mathbf{w}_2$ uniformly distributed within $\{96, \ldots, 114\}$ (between 4pm and 7pm). The map $M_1(k, \mathbf{w}_1, \mathbf{w}_2) \in \{0, 1\}$ is

$$M_1 = \begin{cases} 0, & \text{if } k < \mathbf{w}_1 \text{ or } k > \mathbf{w}_2 \\ 1, & \text{otherwise} \end{cases}.$$

Occupant's overriding on current temperature set point (M_2): The occupant can use a control panel to increase, decrease, or maintain the current temperature. The reference signal has the dynamic equation $T_{\rm ref}(k+1) = T_{\rm ref}(k) + M_2$, where $T_{\rm ref}(k)$ is the current reference signal ($°C$) and M_2 is occupant's control input. In particular, if $\mathbf{z}(k) = 1$, then

$$M_2 = \begin{cases} 0, & \text{w.p.} \quad 0.4 \\ M_1, & \text{w.p.} \quad 0.3 \\ 2M_1, & \text{w.p.} \quad 0.2 \\ 3M_1, & \text{w.p.} \quad 0.1 \end{cases},$$

if $\mathbf{z}(k) = 2$, then $M_2 = 0$, and if $\mathbf{z}(k) = 3$, then

$$M_2 = \begin{cases} 0, & \text{w.p.} \quad 0.4 \\ -M_1, & \text{w.p.} \quad 0.3 \\ -2M_1, & \text{w.p.} \quad 0.2 \\ -3M_1, & \text{w.p.} \quad 0.1 \end{cases}.$$

The temperature set point is assumed to vary within the range $15 \le T_{\rm ref}(k) \le 30$.

Remark 3. *1) Although only the reference signal is changed by the occupant, it can be regarded as an external signal or disturbance that perturbs the nominal dynamics of the system, where the nominal dynamics implies the system in which the reference signal keeps constant. 2) To consider weather changes, 24 hours real weather histories $(T_{\rm o}(k), q_{\rm solar}(k))_{k=0}^{143}$ over 31 days were collected during July, 2017, in West Lafayette, Indiana, USA, and they were used to construct their Markov chain models. Details are omitted here due to the space limit.*

In summary, one obtains a state-space model $x(k+1) = Ax(k) + Bu(k) + Dw(k)$ with $u(k) = q_{\text{HVAC}}(k)$,

$$x(k) = \begin{bmatrix} T_{\text{a}}(k) \\ T_{\text{a}}(k+1) \\ T_{\text{w}}(k+1) \\ T_{\text{ref}}(k+1) \end{bmatrix}, w(k) = \begin{bmatrix} q_{\text{solar}}(k+1) \\ q_{\text{internal}}(k+1) \\ T_{\text{o}}(k+1) \\ M_2 \end{bmatrix},$$

and

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 - \frac{\Delta t}{C_2 R_2} - \frac{\Delta t}{C_2 R_1} & \frac{\Delta t}{C_2 R_1} & 0 \\ 0 & \frac{\Delta t}{C_1 R_1} & 1 - \frac{\Delta t}{C_1 R_3} - \frac{\Delta t}{C_1 R_1} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ \frac{\Delta t}{C_2} \\ 0 \\ 0 \end{bmatrix}, \quad D = \begin{bmatrix} 0 & 0 & 0 & 0 \\ \frac{\Delta t(1-a)}{C_2} & \frac{\Delta t}{C_2} & \frac{\Delta t}{C_2 R_2} & 0 \\ \frac{\Delta t a}{C_1} & 0 & \frac{\Delta t}{C_1 R_3} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

We assume that $q_{\text{solar}}(k+1)$ and $T_{\text{o}}(k+1)$ in $w(k)$ can be measured/observed. Therefore, these signals are augmented into the state vector so that they can be used in the state-feedback control policy (4). Thus, the information structure of the proposed control policy is $(x(k), q_{\text{solar}}(k+1), T_{\text{o}}(k+1))$. The internal heat is $q_{\text{internal}}(k) = 75 + 70M_1$ (W), where the first term, 75, is internal heat due to electronic appliances, and the second term, $70M_1$, indicates the heat produced by the occupant's body. The stage cost function is set to be $g(\mathbf{x}(k),\, \mathbf{u}(k)) = (\mathbf{T}_{\text{a}}(k+1) - \mathbf{T}_{\text{ref}}(k+1))^2 + 0.00001\mathbf{u}(k)^2$. In addition, we consider the space space $\mathbf{T}_{\text{a}}(k) \in [10,\, 30], \mathbf{T}_{\text{a}}(k+1) \in [10,\, 30], \mathbf{T}_{\text{w}}(k+1) \in [10,\, 30], \mathbf{T}_{\text{ref}}(k+1) \in [15,\, 30]$.

4.3 Approximate dynamic programming

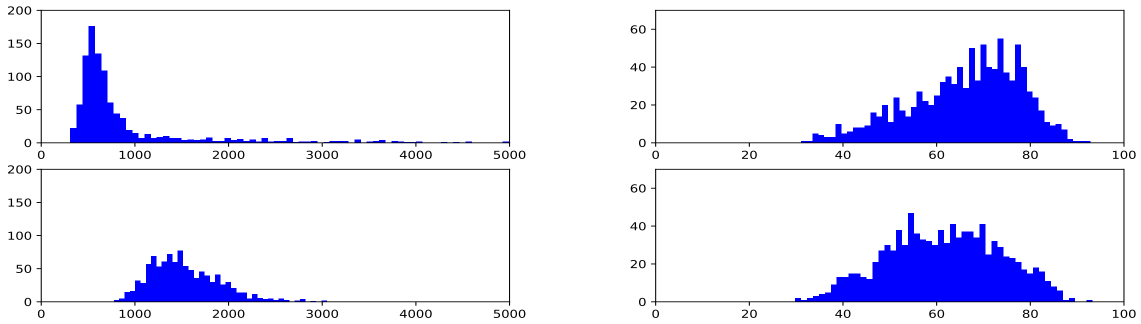
For many important problems, the computational requirements of DP algorithms are overwhelming because they require to find a solution of the Bellman equation in the space of functions or the number of discrete states and control inputs are very large. Approximate dynamic programming (ADP) is a computational approach to approximate the optimal value function. Recent RL approaches address this issue by using neural networks [30] to approximate value functions in high-dimensional spaces, and have demonstrated high performance on various complicated tasks [30]. In this paper, we consider Algorithm 1 in Appendix II, which is a modification of the Q-learning algorithm in [30]. Details of II and a brief introduction of the Q-learning algorithm are also given in Appendix II. Implementation details are provided in Appendix III. We implemented Algorithm 1, and the training took approximately 15 hours. For a comparative analysis, a greedy-type control policy is considered, where the policy minimizes a cost function based on the one-step-ahead prediction of the system trajectory. In particular, given the information $(x(k), q_{\text{solar}}(k+1), T_{\text{o}}(k+1))$, the greedy control policy $u(k) = \pi(x(k), q_{\text{solar}}(k+1), T_{\text{o}}(k+1))$ minimizes the cost function $\alpha x(k+1)^T Q x(k+1) + u(k)^T Q u(k)$ by solving the optimization

$$\pi(x(k)) := \mathop{\arg\min}_{|u| \le 1000} \left\{ \begin{array}{c} \alpha(Ax(k) + Bu + D\eta(k))^T Q \\ \times (Ax(k) + Bu + D\eta(k)) + u^T R u \end{array} \right\}, \tag{7}$$

where $R = 0.00001$, $Q = \begin{bmatrix} 1 & 0 & -1 & 0 \end{bmatrix}^T \begin{bmatrix} 1 & 0 & -1 & 0 \end{bmatrix}$, and

$$\eta(k) := \begin{bmatrix} q_{\text{solar}}(k+1) & 70 & T_{\text{o}}(k+1) & 0 \end{bmatrix}^T.$$

Note that to use an equivalent information structure to the proposed method, only $q_{\text{solar}}(k+1)$ and $T_{\text{o}}(k+1)$ are known in $w(k)$, and $q_{\text{internal}}(k+1)$ and M_2 are arbitrarily set to be 70 and 0, respectively. Figure 4 depicts histograms of the (a) tracking error $\sum_{k=0}^{144\times 7} \mathbf{x}(k)^T Q\mathbf{x}(k)$ and the (b) input energy $\sum_{k=0}^{144\times 7} \mathbf{u}(k)^T R\mathbf{u}(k)$ obtained with 1000 simulations over the one week time interval. The upper figures are the results of the proposed approach, and the lower figures are those of the greedy control policy in (7). The results suggest that the proposed RL approach outperforms in terms of the tracking error, while the input energy is comparable with the greedy policy.



(a) Tracking error

(b) Input energy

Figure 4: Histograms of the (a) tracking error $\sum_{k=0}^{144\times 7} \mathbf{x}(k)^T Q\mathbf{x}(k)$ and the (b) input energy $\sum_{k=0}^{144\times 7} \mathbf{u}(k)^T R\mathbf{u}(k)$ over one week simulations. The upper figures are the results of the proposed RL approach, and the lower figures are the results of the greedy control policy in (7).

Conclusion

In this paper, we have studied DP and RL algorithms for building control problems with occupant interactions. Through simulation studies, we have demonstrated potential applicability of RL for those problems. A future research agenda is to consider scenarios with multiple rooms and multiple occupants. The problem has larger state spaces and higher uncertainties. Applications of the proposed results to these problems are possible future research directions.

References

[1] A. I. Dounis and C. Caraiscos, “Advanced control systems engineering for energy and comfort management in a building environment–A review,” *Renewable and Sustainable Energy Reviews*, vol. 13, no. 6, pp. 1246–1261, 2009.

[2] A. Aswani, N. Master, J. Taneja, D. Culler, and C. Tomlin, “Reducing transient and steady state electricity consumption in hvac using learning-based model-predictive control,” *Proceedings of the IEEE*, vol. 100, no. 1, pp. 240–253, 2012.

[3] J. Page, D. Robinson, N. Morel, and J.-L. Scartezzini, “A generalised stochastic model for the simulation of occupant presence,” *Energy and buildings*, vol. 40, no. 2, pp. 83–98, 2008.

[4] F. Oldewurtel, D. Sturzenegger, and M. Morari, “Importance of occupancy information for building climate control,” *Applied energy*, vol. 101, pp. 521–532, 2013.

[5] J. R. Dobbs and B. M. Hencey, “Model predictive hvac control with online occupancy model,” *Energy and Buildings*, vol. 82, pp. 675–684, 2014.

[6] W. Liu, Z. Lian, and B. Zhao, “A neural network evaluation model for individual thermal comfort,” *Energy and Buildings*, vol. 39, no. 10, pp. 1115–1122, 2007.

[7] D. Daum, F. Haldi, and N. Morel, “A personalized measure of thermal comfort for building controls,” *Building and Environment*, vol. 46, no. 1, pp. 3–11, 2011.

[8] S. Lee, I. Bilionis, P. Karava, and A. Tzempelikos, “A bayesian approach for probabilistic classification and inference of occupant thermal preferences in office buildings,” *Building and Environment*, vol. 118, pp. 323–343, 2017.

[9] Y. Ma, A. Kelman, A. Daly, and F. Borrelli, “Predictive control for energy efficient buildings with thermal storage,” *IEEE Control system magazine*, vol. 32, no. 1, pp. 44–64, 2012.

[10] Y. Ma, F. Borrelli, B. Hencey, B. Coffey, S. Bengea, and P. Haves, “Model predictive control for the operation of building cooling systems,” *IEEE Transactions on Control Systems Technology*, vol. 20, no. 3, pp. 796–803, 2012.

[11] Y. Ma, S. Vichik, and F. Borrelli, “Fast stochastic MPC with optimal risk allocation applied to building control systems,” in *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, 2012, pp. 7559–7564.

[12] Y. Ma and F. Borrelli, “Fast stochastic predictive control for building temperature regulation,” in *American Control Conference (ACC), 2012*, 2012, pp. 3075–3080.

[13] F. Oldewurtel, A. Parisio, C. N. Jones, M. Morari, D. Gyalistras, M. Gwerder, V. Stauch, B. Lehmann, and K. Wirth, “Energy efficient building climate control using stochastic model predictive control and weather predictions,” in *American control conference (ACC), 2010*, 2010, pp. 5100–5105.

[14] F. Oldewurtel, A. Parisio, C. N. Jones, D. Gyalistras, M. Gwerder, V. Stauch, B. Lehmann, and M. Morari, “Use of model predictive control and weather forecasts for energy efficient building climate control,” *Energy and Buildings*, vol. 45, pp. 15–27, 2012.

[15] L. Blackmore, M. Ono, A. Bektassov, and B. C. Williams, “A probabilistic particle-control approximation of chance-constrained stochastic predictive control,” *IEEE transactions on Robotics*, vol. 26, no. 3, pp. 502–517, 2010.

[16] G. C. Calafiore and L. Fagiano, “Stochastic model predictive control of LPV systems via scenario optimization,” *Automatica*, vol. 49, no. 6, pp. 1861–1866, 2013.

[17] M. Prandini, J. Hu, J. Lygeros, and S. Sastry, “A probabilistic approach to aircraft conflict detection,” *IEEE Transactions on intelligent transportation systems*, vol. 1, no. 4, pp. 199–220, 2000.

[18] A. Parisio, M. Molinari, D. Varagnolo, and K. H. Johansson, “A scenario-based predictive control approach to building HVAC management systems,” in *Automation Science and Engineering (CASE), 2013 IEEE International Conference on*, 2013, pp. 428–435.

[19] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-dynamic programming.* Athena Scientific Belmont, MA, 1996.

[20] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction.* MIT Press, 1998.

[21] L. Yang, Z. Nagy, P. Goffin, and A. Schlueter, "Reinforcement learning for optimal control of low exergy buildings," *Applied Energy*, vol. 156, pp. 577–586, 2015.

[22] S. Liu and G. P. Henze, "Experimental analysis of simulated reinforcement learning control for active and passive building thermal storage inventory: Part 2: Results and analysis," *Energy and buildings*, vol. 38, no. 2, pp. 148–161, 2006.

[23] Z. Yu and A. Dexter, "Online tuning of a supervisory fuzzy controller for low-energy building system using reinforcement learning," *Control Engineering Practice*, vol. 18, no. 5, pp. 532–539, 2010.

[24] Z. Cheng, Q. Zhao, F. Wang, Y. Jiang, L. Xia, and J. Ding, "Satisfaction based Q-learning for integrated lighting and blind control," *Energy and Buildings*, vol. 127, pp. 43–55, 2016.

[25] I. V. Kolmanovsky, L. Lezhnev, and T. L. Maizenberg, "Discrete-time drift counteraction stochastic optimal control: Theory and application-motivated examples," *Automatica*, vol. 44, no. 1, pp. 177–184, 2008.

[26] L. Johannesson, M. Asbogard, and B. Egardt, "Assessing the potential of predictive control for hybrid vehicle powertrains using stochastic dynamic programming," *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, no. 1, pp. 71–83, 2007.

[27] G. P. Henze and J. Schoenmann, "Evaluation of reinforcement learning control for thermal energy storage systems," *HVAC&R Research*, vol. 9, no. 3, pp. 259–275, 2003.

[28] R. Hafner and M. Riedmiller, "Reinforcement learning in feedback control," *Machine learning*, vol. 84, no. 1-2, pp. 137–169, 2011.

[29] M. Hausknecht and P. Stone, "Deep recurrent Q-learning for partially observable MDPSs," *arXiv preprint arXiv:1507.06527*, 2015.

[30] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[31] Z. Wang, T. Schaul, M. Hessel, H. Van Hasselt, M. Lanctot, and N. De Freitas, "Dueling network architectures for deep reinforcement learning," *arXiv preprint arXiv:1511.06581*, 2015.

[32] M. Riedmiller, "Neural fitted Q iteration–first experiences with a data efficient neural reinforcement learning method," in *European Conference on Machine Learning*, 2005, pp. 317–328.

[33] A. Geramifard, T. J. Walsh, S. Tellex, G. Chowdhary, N. Roy, J. P. How *et al.*, "A tutorial on linear function approximators for dynamic programming and reinforcement learning," *Foundations and Trends® in Machine Learning*, vol. 6, no. 4, pp. 375–451, 2013.

[34] Y. Engel, S. Mannor, and R. Meir, "Bayes meets Bellman: The Gaussian process approach to temporal difference learning," in *ICML*, vol. 20, no. 1, p. 154.

[35] D. P. Bertsekas, *Dynamic programming and optimal control.* Athena Scientific Belmont, MA, 1995, vol. 1, no. 2.

[36] U. D. of Energy, “Energyplustm 8.7.0 documentation,” https://energyplus.net/documentation, [Online Available].

[37] L.-J. Lin, “Reinforcement learning for robots using neural networks,” Carnegie-Mellon Univ Pittsburgh PA School of Computer Science, Tech. Rep., 1993.

Appendix I: Proof of Theorem 1

We first find another characterization of $J_k(x_0,\, z_0)$ in terms of a sum of stage cost functions.

Lemma 1. *For any fixed $k \geq 1$, J_k is described as*

$$J_k(x_0, z_0) = \inf_{\pi \in \Pi} \mathbb{E}_{x_0,z_0} \left[\sum_{i=0}^{\min\{\tau(x_0,z_0;\pi)-1,k-1\}} \alpha^i g(\mathbf{x}(i), \mathbf{u}(i), \mathbf{z}(i)) \right],$$

where $\mathbf{u}(i) = \pi(\mathbf{x}(i), \mathbf{z}(i))$.

Proof. The claim will be proved by an induction argument. Since $J_0 \equiv 0$, $J_1(x_0, z_0)$ is given by

$$J_1(x_0, z_0) = (TJ_0)(x_0, z_0) = \inf_{u \in U} \mathbb{E}_{x_0,z_0} [g(x_0, u, z_0)] = \inf_{\pi \in \Pi} \mathbb{E}_{x_0,z_0} \left[\sum_{i=0}^{\min\{\tau(x_0,z_0;\pi)-1,0\}} \alpha^i g(\mathbf{x}(i), \mathbf{u}(i), \mathbf{z}(i)) \right].$$

Now, suppose for $k \geq 2$

$$J_{k-1}(x_0, z_0) = \inf_{\pi \in \Pi} \mathbb{E}_{x_0,z_0} \left[\sum_{i=0}^{\min\{\tau(x_0,z_0;\pi)-1,k-2\}} \alpha^i g(\mathbf{x}(i), \mathbf{u}(i), \mathbf{z}(i)) \right] \tag{8}$$

holds. Then,

$$J_k(x_0, z_0) = (TJ_{k-1})(x_0, z_0) = \inf_{u \in U} \mathbb{E}_{x_0,z_0} [g(x_0, u, z_0) + \alpha \mathbb{I}_X(\mathbf{x}(1)) J_{k-1}(\mathbf{x}(1), \mathbf{z}(1))].$$

By conditioning on the exit time $\tau(x_0, z_0; \pi)$, the expectation in the last equation is expressed as

$$\mathbb{E}_{x_0,z_0}[g(x_0, u(0), z_0) | \tau(x_0, z_0; \pi) = 1] \mathbb{P}[\tau(x_0, z_0; \pi) = 1] + \mathbb{E}_{x_0,z_0}[\chi | \tau(x_0, z_0; \pi) \geq 2] \mathbb{P}[\tau(x_0, z_0; \pi) \geq 2], \tag{9}$$

where

$$\chi := g(x_0, u(0), z_0) + \sum_{i=1}^{\min\{\tau(\mathbf{x}(1),\mathbf{z}(1);\pi)-1,k-2\}+1} \alpha^i g(\mathbf{x}(i), \mathbf{u}(i), \mathbf{z}(i)),$$

and the second term is obtained by the induction hypothesis (8). In the second term, the quantity $\min\{\tau(\mathbf{x}(1), \mathbf{z}(1); \pi) - 1, k-2\} + 1$ is rewritten as

$$\min\{\tau(\mathbf{x}(1), \mathbf{z}(1); \pi) - 1, k-2\} + 1 = \min\{\tau(\mathbf{x}(1), \mathbf{z}(1); \pi), k-1\} = \min\{\tau(x_0, z_0; \pi) - 1, k-1\}.$$

Therefore, (9) is identical to

$$\mathbb{E}_{x_0,z_0} \left[\sum_{i=0}^{\min\{\tau(x_0,z_0;\pi)-1,k-1\}} \alpha^i g(\mathbf{x}(i), \mathbf{u}(i), \mathbf{z}(i)) \right],$$

and the desired result follows. □

Proof of Theorem 1: For any fixed $\pi \in \Pi$, define

$$J_k^{\pi}(x_0, z_0) := \mathbb{E}_{x_0,z_0}\left[\sum_{i=0}^{\min\{\tau(x_0,z_0;\pi)-1,k-1\}} \alpha^i g(\mathbf{x}(i), \mathbf{u}(i), \mathbf{z}(i))\right],$$

where $\mathbf{u}(i) = \pi(\mathbf{x}(i), \mathbf{z}(i))$. Let π be an arbitrary policy and let $u(i) = \pi(x(i), z(i))$. For any $k \geq 1$ and sample path $(x(i), z(i))_{i=0}^{\infty}$, we have

$$\left|\sum_{i=0}^{\min\{\tau(x_0,z_0;\pi)-1,k-1\}} \alpha^i g(x(i), u(i), z(i))\right| \leq \left|\sum_{i=0}^{\min\{\tau(x_0,z_0;\pi)-1,k-1\}} \alpha^i M\right| \leq M\left|\sum_{i=0}^{\infty} \alpha^i\right| \leq M\frac{1}{1-\alpha}. \tag{10}$$

Therefore, $J_k^{\pi}(x_0, z_0)$ is bounded. Since $J_k^{\pi}(x_0, z_0)$ is non-decreasing in k, the point-wise limit $\lim_{k\to\infty} J_k^{\pi}(x_0, z_0)$ exists. Choose a ε-suboptimal control policy $\pi^{\varepsilon} \in \Pi$ such that $J^{\pi^{\varepsilon}}(x_0, z_0) \leq J^*(x_0, z_0) + \varepsilon$. By the dominated convergence theorem, we have

$$\begin{aligned}
&\lim_{k\to\infty} J_k^{\pi^{\varepsilon}}(x_0, z_0) \\
&= \lim_{k\to\infty} \mathbb{E}_{x_0,z_0}\left[\sum_{i=0}^{\min\{\tau(x_0,z_0;\pi^{\varepsilon})-1,k-1\}} \alpha^i g(\mathbf{x}(i), \mathbf{u}(i), \mathbf{z}(i))\right] \\
&= \mathbb{E}_{x_0,z_0}\left[\lim_{k\to\infty} \sum_{i=0}^{\min\{\tau(x_0,z_0;\pi^{\varepsilon}\})-1,k-1\}} \alpha^i g(\mathbf{x}(i), \mathbf{u}(i), \mathbf{z}(i))\right] \\
&= \mathbb{E}_{x_0,z_0}\left[\lim_{k\to\infty} \sum_{i=0}^{\tau(x_0,z_0;\pi^{\varepsilon})-1} \alpha^i g(\mathbf{x}(i), \mathbf{u}(i), \mathbf{z}(i))\right] \\
&= J^{\pi^{\varepsilon}}(x_0, z_0).
\end{aligned} \tag{11}$$

Since $J_k(x_0, z_0) = \inf_{\pi\in\Pi} J_k^{\pi}(x_0, z_0)$ by Lemma 1, we have $J_k(x_0, z_0) \leq J_k^{\pi^{\varepsilon}}(x_0, z_0)$. Combining it with (11) leads to

$$\lim_{k\to\infty} J_k(x_0, z_0) \leq \lim_{k\to\infty} J_k^{\pi^{\varepsilon}}(x_0, z_0) = J^{\pi^{\varepsilon}}(x_0, z_0) \leq J^*(x_0, z_0) + \varepsilon.$$

Since $\varepsilon > 0$ is arbitrary, we have $\lim_{k\to\infty} J_k(x_0, z_0) \leq J^(x_0, z_0)$. To prove the reversed direction, note that

$$J^*(x_0, z_0) \leq J_k(x_0, z_0) + \mathbb{E}_{x_0,z_0}\left[\sum_{i=\min\{\tau(x_0,z_0;\pi)-1,k-1\}+1}^{\tau(x_0,z_0;\pi)-1} \alpha^i g(\mathbf{x}(i), u(i), \mathbf{z}(i))\right],$$

where $\pi \in \Pi$ is arbitrary. Taking the limit $k \to \infty$ on the right-hand side yields $J^*(x_0, z_0) \leq \lim_{k\to\infty} J_k(x_0, z_0)$. This completes the proof.

Appendix II: Q-learning algorithm [30]

This section briefly introduces the Q-learning algorithm developed in [30]. For notational simplicity, consider any MDP (Markov decision process) characterized by $(X, U, \mathcal{T}, g, \gamma)$. At time-step k, the agent with state $x \in X$ executes an action $u \in U$ using the policy $\pi : X \to U$, receives the

Algorithm 1 Deep Q-learning [30]

```
1:  Initialize replay memory $\mathcal{D}$ to capacity $|\mathcal{D}|$.
2:  Initialize $Q$-factor, $Q$, with random weights $\theta$.
3:  Initialize target $Q$-factor's weights $\theta^-$ with $\theta^- = \theta$.
4:  Set $k = 0$.
5:  repeat
6:      With probability $\varepsilon > 0$, select a random control action $u \in U$. Otherwise, select $u = \min_{u \in U} Q(x, u; \theta)$.
7:      Execute the control action $u$ at $k$ in simulator and observe $g$ and the next state $x'$.
8:      Store the transition $(x, u, g, x')$ in $\mathcal{D}$.
9:      Sample a random mini-batch $\mathcal{M}$ of transitions $(x, u, g, x')$ from $\mathcal{D}$.
10:     for $(x, u, g, x') \in \mathcal{M}$ do
11:         Set $y = g + \gamma \min_{u' \in U} Q(x', u'; \theta^-)$.
12:         Perform a gradient descent step on $(y - Q(x, u; \theta))^2$ with respect to $\theta$.
13:     end for
14:     $x \leftarrow x'$ and $k \leftarrow k + 1$
15:     Every $K$ steps, reset $\theta^- = \theta$.
16: until $k > k_{\max}$
```

cost/reward g (deterministic or random and depending on (x, u, x')), and transitions to state $x' \in X$ with probability $\mathbb{P}(x'|x, u) = \mathcal{T}(x, u, x')$. The RL in [30], named deep Q-learning, modified for our control purposes is summarized in Algorithm 1. In Algorithm 1, the neural network $Q(\cdot, \cdot; \theta)$, called deep Q-network (DQN), with parameters θ is used to approximate the optimal Q-factor, where we optimize the following sequence of loss functions at iteration i:

$$L(\theta_i) := \mathbb{E}_{x,u,g,x'}[(y_i - Q(x, u; \theta_i))^2],$$

where $\mathbb{E}_{x,u,g,x'}$ is the expectation with respect to (x, u, g, x'), $y_i = g + \gamma \inf_{u' \in U} Q(x', u'; \theta_i^-)$, x is the current state sampled in any way, x' is the state of the next time step given the current state x, u is the current control input, θ^- represents the parameters of a fixed and separate DQN, called target DQN. To distinguish it with the original Q, the original DQN is called online DQN. In other words, $Q(x', u'; \theta_i^-)$ is the target DQN, while $Q(x', u'; \theta_i)$ is the online DQN. The control input u is driven by the ε-greedy:

$$u^{\varepsilon}(x) := \begin{cases} \inf_{u' \in U} Q(x, u'; \theta_i) \text{ w.p. } 1 - \varepsilon \\ \text{Select a random control in } U \text{ w.p. } \varepsilon \end{cases}$$

In one of standard Q-learning algorithms, stochastic gradient descent-type algorithms are used to minimize $L(\theta_i)$ with respect to θ_i at each step. However, this approach is known to be unstable or even to diverge [30] when a nonlinear function approximator, e.g., a neural network, is used. Two modifications made in [30] were empirically proved to improve the stability of the algorithm [31]. First, the parameters θ_i^- of the target DQN $Q(x', u'; \theta_i^-)$ are freezed for a fixed number K of iterations while updating the online DQN $Q(x; u; \theta_i)$ by stochastic gradient descent (line 12 in Algorithm 1). Another key ingredient is the experience replay [37]. During the learning process, the agent accumulates a dataset $\mathcal{D} = \{e_1, e_2, \ldots, e_{|\mathcal{D}|}\}$ of experiences $e_k := (x, u, g, x')_k$ from system trajectories (line 8 in Algorithm 1). When training the DQN, instead only using the current experience as prescribed by standard Q-learning, the DQN is trained by sampling mini-batches $\mathcal{M}$ of experiences from $\mathcal{D}$ uniformly at random (line 9 in Algorithm 1). Therefore, the losses at step i takes the form

$$L(\theta_i) := \mathbb{E}_{(x,u,g,x')\sim\mathcal{U}(\mathcal{D})}[(y_i - Q(x,u;\theta_i))^2],$$

where $\mathbb{E}_{(x,u,g,x')\sim\mathcal{U}(\mathcal{D})}$ indicates the expectation with respect to the tuple (x,u,g,x') which has the uniform distribution on $\mathcal{D}$ (from line 10 to line 13 in Algorithm 1). The experience replay increases data efficiency through re-use of experience samples in multiple updates and, importantly, it reduces variance as uniform sampling from the replay buffer reduces the correlation among the samples used in the update [31].

Appendix III: Implementation details

We implemented Algorithm 1 using Python running on a Linux 16.04 LTS PC with 64 bit Intel Core i7-4790 3.60GHz×8 CPU and 16 GB RAM. This work uses a neural network with four layers of width $(500, 300, 100, 50)$, which is the size chosen by experiments to balance between the computational efficiency and performance. We also consider $U = \{-1000, -500, -200, -100, 0, 100, 200, 500, 1000\}$, $\alpha = 0.8$, $\varepsilon = 0.05$, $|\mathcal{D}| = 1000$, $|\mathcal{M}| = 100$, $K = 1$, and $k_{\max} = 10000000$.