Control Frequency Adaptation via Action Persistence in Batch Reinforcement Learning

Alberto Maria Metelli 1  Flavio Mazzolini 1  Lorenzo Biš 1,2  Luca Sabbioni 1,2  Marcello Restelli 1,2

Abstract

The choice of the control frequency of a system has a relevant impact on the ability of reinforcement learning algorithms to learn a highly performing policy. In this paper, we introduce the notion of action persistence that consists in the repetition of an action for a fixed number of decision steps, having the effect of modifying the control frequency. We start analyzing how action persistence affects the performance of the optimal policy, and then we present a novel algorithm, Persistent Fitted Q-Iteration (PFQI), that extends FQI, with the goal of learning the optimal value function at a given persistence. After having provided a theoretical study of PFQI and a heuristic approach to identify the optimal persistence, we present an experimental campaign on benchmark domains to show the advantages of action persistence and proving the effectiveness of our persistence selection method.

1. Introduction

In recent years, Reinforcement Learning (RL, Sutton & Barto, 2018) has proven to be a successful approach to address complex control tasks: from robotic locomotion (e.g., Peters & Schaal, 2008; Kober & Peters, 2014; Haarnoja et al., 2019; Kilinc et al., 2019) to continuous system control (e.g., Schulman et al., 2015; Lillicrap et al., 2016; Schulman et al., 2017). These classes of problems are usually formalized in the framework of the discrete–time Markov Decision Processes (MDP, Puterman, 2014), assuming that the control signal is issued at discrete time instants. However, many relevant real–world problems are more naturally defined in the continuous–time domain (Luenberger, 1979). Even though a branch of literature has studied RL in continuous–time MDPs (Bradtke & Duff, 1994; Munos & Bourgine, 1997; Doya, 2000), the majority of the research has focused on the discrete–time formulation, which appears to be a necessary, but effective, approximation.

Intuitively, increasing the control frequency of the system offers the agent more control opportunities, possibly leading to improved performance as the agent has access to a larger policy space. This might wrongly suggest that we should control the system with the highest frequency possible, within its physical limits. However, in the RL framework, the environment dynamics is unknown, thus, a too fine discretization could result in the opposite effect, making the problem harder to solve. Indeed, any RL algorithm needs samples to figure out (implicitly or explicitly) how the environment evolves as an effect of the agent’s actions. When increasing the control frequency, the advantage of individual actions becomes infinitesimal, making them almost indistinguishable for standard value–based RL approaches (Tallec et al., 2019). As a consequence, the sample complexity increases. Instead, low frequencies allow the environment to evolve longer, making the effect of individual actions more easily detectable. Furthermore, in the presence of a system characterized by a “slowly evolving” dynamics, the gain obtained by increasing the control frequency might become negligible. Finally, in robotics, lower frequencies help to overcome some partial observability issues, like action execution delays (Kober & Peters, 2014).

Therefore, we experience a fundamental trade–off in the control frequency choice that involves the policy space (larger at high frequency) and the sample complexity (smaller at low frequency). Thus, it seems natural to wonder: “what is the optimal control frequency?” An answer to this question can disregard neither the task we are facing nor the learning algorithm we intend to employ. Indeed, the performance loss we experience by reducing the control frequency depends strictly on the properties of the system and, thus, of the task. Similarly, the dependence of the sample complexity on the control frequency is related to how the learning algorithm will employ the collected samples.

In this paper, we analyze and exploit this trade–off in the context of batch RL (Lange et al., 2012), with the goal of enhancing the learning process and achieving higher perfor-
mance. We assume to have access to a discrete–time MDP $M_{\Delta t_0}$, called base MDP, which is obtained from the time discretization of a continuous–time MDP with fixed base control time step $\Delta t_0$, or equivalently, a control frequency equal to $f_0 = \frac{1}{\Delta t_0}$. In this setting, we want to select a suitable control time step $\Delta t$ that is an integer multiple of the base time step $\Delta t_0$, i.e., $\Delta t = k \Delta t_0$ with $k \in \mathbb{N}_{\geq 1}$. Any choice of $k$ generates an MDP $M_k$ obtained from the base one $M_{\Delta t_0}$ by altering the transition model so that each action is repeated for $k$ times. For this reason, we refer to $k$ as the action persistence, i.e., the number of decision epochs in which an action is kept fixed. It is possible to appreciate the same effect in the base MDP $M_{\Delta t_0}$ by executing a (non-Markovian and non-stationary) policy that persists every action for $k$ time steps. The idea of repeating actions has been previously employed, although heuristically, with deep RL architectures (Lakshminarayanan et al., 2017).

The contributions of this paper are theoretical, algorithmic, and experimental. We first prove that action persistence (with a fixed $k$) can be represented by a suitable modification of the Bellman operators, which preserves the contraction property and, consequently, allows deriving the corresponding value functions (Section 3). Since increasing the duration of the control time step $\Delta t_0$ has the effect of degrading the performance of the optimal policy, we derive an algorithm–independent bound for the difference between the optimal value functions of MDPs $M_{\Delta t_0}$ and $M_k$, which holds under Lipschitz conditions. The result confirms the intuition that the performance loss is strictly related to how fast the environment evolves as an effect of the actions (Section 4). Then, we apply the notion of action persistence in the batch RL scenario, proposing and analyzing an extension of Fitted Q-Iteration (FQI, Ernst et al., 2005). The resulting algorithm, Persistent Fitted Q-Iteration (PFQI) takes as input a target persistence $k$ and estimates the corresponding optimal value function, assuming to have access to a dataset of samples collected in the base MDP $M_{\Delta t_0}$ (Section 5). Once we estimate the value function for a set of candidate persistences $K \subset \mathbb{N}_{\geq 1}$, we aim at selecting the one that yields the best performing greedy policy. Thus, we introduce a persistence selection heuristic able to approximate the optimal persistence, without requiring further interactions with the environment (Section 6). After having revised the literature (Section 7), we present an experimental evaluation on benchmark domains, to confirm our theoretical findings and evaluate our persistence selection method (Section 8). We conclude by discussing some open questions related to action persistence (Section 9). The proofs of all the results can be found in Appendix A. The code is available at github.com/albertometelli/pfqi.

2. Preliminaries

In this section, we introduce the notation and the basic notions that we will employ in the remainder of the paper.

Mathematical Background Let $\mathcal{X}$ be a set with a $\sigma$-algebra $\sigma_\mathcal{X}$, we denote with $\mathcal{P}(\mathcal{X})$ the set of all probability measures and with $\mathcal{B}(\mathcal{X})$ the set of all bounded measurable functions over $(\mathcal{X}, \sigma_\mathcal{X})$. If $x \in \mathcal{X}$, we denote with $\delta_x$ the Dirac measure defined on $x$. Given a probability measure $\rho \in \mathcal{P}(\mathcal{X})$ and a measurable function $f \in \mathcal{B}(\mathcal{X})$, we abbreviate $\rho f = \int f(x) \rho(dx)$ (i.e., we use $\rho$ as an operator). Moreover, we define the $L_p(\rho)$-norm of $f$ as $\|f\|_{p, \rho} = \int |f(x)|^p \rho(dx)$ for $p \geq 1$, whereas the $L_{\infty}$-norm is defined as $\|f\|_{\infty} = \sup_{x \in \mathcal{X}} f(x)$. Let $\mathcal{D} = \{x_i\}_{i=1}^n \subseteq \mathcal{X}$ we define the $L_p(\rho)$ empirical norm as $\|f\|_{p, \mathcal{D}} = \frac{1}{n} \sum_{i=1}^n |f(x_i)|^p$.

Markov Decision Processes A discrete-time Markov Decision Process (MDP, Puterman, 2014) is a $5$-tuple $M = (\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$, where $\mathcal{S}$ is a measurable set of states, $\mathcal{A}$ is a measurable set of actions, $\mathcal{P}: \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{P}(\mathcal{S})$ is the transition kernel that for each state-action pair $(s, a) \in \mathcal{S} \times \mathcal{A}$ provides the probability distribution $\mathcal{P}(|s, a)$ of the next state, $\mathcal{R}: \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{P}(\mathbb{R})$ is the reward distribution $\mathcal{R}(|s, a)$ for performing action $a \in \mathcal{A}$ in state $s \in \mathcal{S}$, whose expected value is denoted by $r(s, a) = \sum_{x \in \mathcal{X}} R(dx|s, a)$ and uniformly bounded by $R_{\max} < +\infty$, and $\gamma \in [0, 1)$ is the discount factor.

A policy $\pi = (\pi_t)_{t \in \mathbb{N}}$ is a sequence of functions $\pi_t: \mathcal{H}_t \rightarrow \mathcal{P}(\mathcal{S})$ mapping a history $\mathcal{H}_t = (S_0, A_0, ..., S_{t-1}, A_{t-1}, S_t)$ of length $t \in \mathbb{N}$ to a probability distribution over $\mathcal{A}$, where $\mathcal{H}_t = (\mathcal{S} \times \mathcal{A})^t \times \mathcal{S}$. If $\pi_t$ depends only on the last visited state $S_t$ then it is called Markovian, i.e., $\pi_t: \mathcal{S} \rightarrow \mathcal{P}(\mathcal{A})$. Moreover, if $\pi_t$ does not depend on explicitly $t$ it is stationary, in this case we remove the subscript $t$. We denote with $\Pi$ the set of Markovian stationary policies. A policy $\pi \in \Pi$ induces a (state-action) transition kernel $P^\pi: \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{P}(\mathcal{S} \times \mathcal{A})$, defined for any measurable set $B \subset \mathcal{S} \times \mathcal{A}$ as (Farahmand, 2011):

$$
(P^\pi)(B|s, a) = \int_{\mathcal{A}} P(ds'|s, a) \int_{\mathcal{A}} \pi(da'|s') \delta_\{s', a'|\}(B).
$$

(1)

The action-value function, or Q-function, of a policy $\pi \in \Pi$ is the expected discounted sum of the rewards obtained by performing action $a$ in state $s$ and following policy $\pi$ thereafter $Q^\pi(s, a) = \mathbb{E} \left[ \sum_{t=0}^{\infty} \gamma^t R_t | S_0 = s, A_0 = a \right]$, where $R_t \sim \mathcal{R}(|S_t, A_t)$, $S_{t+1} \sim \mathcal{P}(|S_t, A_t)$, and $A_{t+1} \sim \pi(|S_{t+1})$ for all $t \in \mathbb{N}$. The value function is the expectation of the Q-function over the actions: $V^\pi(s) = \int_{\mathcal{A}} \pi(da|s) Q^\pi(s, a)$. Given a distribution $\rho \in \mathcal{P}(\mathcal{S})$, we define the expected return as $J^\rho(\pi) = \int_{\mathcal{S}} \rho(ds) V^\pi(s)$. The optimal Q-function is
given by: \( Q^\pi(s,a) = \sup_{\pi \in \Pi} Q^\pi(s,a) \) for all \((s,a) \in S \times A\). A policy \( \pi \) is greedy w.r.t. a function \( f \in \mathcal{B}(S \times A) \) if it plays only greedy actions, i.e., \( \pi(\cdot|s) \in \mathcal{F}(\arg \max_{a \in A} f(s,a)) \). An optimal policy \( \pi^* \in \Pi \) is any policy greedy w.r.t. \( Q^\pi \).

Given a policy \( \pi \in \Pi \), the Bellman Expectation Operator \( T^\pi \): \( \mathcal{B}(S \times A) \to \mathcal{B}(S \times A) \) and the Bellman Optimal Operator \( T^{\pi^*} \): \( \mathcal{B}(S \times A) \to \mathcal{B}(S \times A) \) are defined for a bounded measurable function \( f \in \mathcal{B}(S \times A) \) and \((s,a) \in S \times A\) as (Bertsekas & Shreve, 2004):

\[
(T^\pi f)(s,a) = r(s,a) + (T^\pi f)(s,a),
\]

\[
(T^{\pi^*} f)(s,a) = r(s,a) + \gamma \int S P(ds'|s,a) \max_{a' \in A} f(s',a').
\]

Both \( T^\pi \) and \( T^{\pi^*} \) are \( \gamma \)-contractions in \( L_\infty \)-norm and, consequently, they have a unique fixed point, that are the \( \mathcal{Q} \)-function of policy \( \pi \) \( (T^\pi Q^\pi = Q^\pi) \) and the optimal \( \mathcal{Q} \)-function \( (T^{\pi^*} Q^{\pi^*} = Q^{\pi^*}) \) respectively.

**Lipschitz MDPs** Let \((X,d_X)\) and \((Y,d_Y)\) be two metric spaces, a function \( f: X \to Y \) is called \( L_f \)-Lipschitz continuous \( (L_f \text{-LC}) \), where \( L_f > 0 \), if for all \( x, x' \in X \) we have:

\[
d_Y(f(x), f(x')) \leq L_f d_X(x, x').
\]

Moreover, we define the Lipschitz semi-norm as \( \|f\|_L = \sup_{x, x' \in X} d_Y(f(x), f(x')) / d_X(x, x') \). For real functions we employ Euclidean distance \( d_Y(y, y') = |y - y'|_2 \), while for probability distributions we use the Kantorovich \( (L_1 \text{-Wasserstein}) \) metric defined for \( \mu, \nu \in \mathcal{P}(Z) \) as (Villani, 2008):

\[
d_1(\mu, \nu) = W_1(\mu, \nu) = \sup_{f : \|f\|_L \leq 1} \left| \int Z f(z) (\mu - \nu)(dz) \right|.
\]

We now introduce the notions of Lipschitz MDP and Lipschitz policy that we will employ in the following (Rachelson & Lagoudakis, 2010; Pirotta et al., 2015).

**Assumption 2.1** (Lipschitz MDP). **Let \( M \) be an MDP. \( M \) is called \( (L_f, L_r) \)-LC if for all \((s, a),(\pi, \bar{\pi}) \in S \times A\):

\[
W_1(P(\cdot|s, a), P(\cdot|\pi, \bar{\pi})) \leq L_f d_S(s, a)(\pi, \bar{\pi}),
\]

\[
|r(s, a) - r(\pi, \bar{\pi})| \leq L_r d_S(s, a)(\pi, \bar{\pi}).
\]

**Assumption 2.2** (Lipschitz Policy). **Let \( \pi \in \Pi \) be a Markovian stationary policy. \( \pi \) is called \( L_\pi \)-LC if for all \( s, \pi \in S\):

\[
W_1(\pi(\cdot|s), \pi(\cdot|\pi)) \leq L_\pi d_S(s, \pi).
\]

### 3. Persisting Actions in MDPs

By the phrase “executing a policy \( \pi \) at persistence \( k \)”, with \( k \in \mathbb{N}_{\geq 1} \), we mean the following type of agent-environment interaction. At decision step \( t = 0 \), the agent selects an action according to its policy \( A_0 \sim \pi(\cdot|S_0) \). Action \( A_0 \) is kept fixed, or persisted, for the subsequent \( k - 1 \) decision steps, i.e., actions \( A_1, \ldots, A_{k-1} \) are all equal to \( A_0 \). Then, at decision step \( t = k \), the agent queries again the policy \( A_k \sim \pi(\cdot|S_k) \) and persists action \( A_k \) for the subsequent \( k - 1 \) decision steps and so on. In other words, the agent employs its policy only at decision steps \( t \) that are integer multiples of the persistence \( k \) \( (t \mod k = 0) \). Clearly, the usual execution of \( \pi \) corresponds to persistence 1.

### 3.1. Duality of Action Persistence

Unsurprisingly, the execution of a Markovian stationary policy \( \pi \) at persistence \( k > 1 \) produces a behavior that, in general, cannot be represented by executing any Markovian stationary policy at persistence 1. Indeed, at any decision step \( t \), such a policy needs to remember which action was taken at the previous decision step \( t - 1 \) (thus it is non-Markovian with memory 1) and has to understand whether to select a new action based on \( t \) (so it is non-stationary).

**Definition 3.1** (\( k \)-persistent policy). **Let \( \pi \in \Pi \) be a Markovian stationary policy. For any \( k \in \mathbb{N}_{\geq 1} \), the \( k \)-persistent policy induced by \( \pi \) is a non–Markovian non–stationary policy, defined for any measurable set \( B \subseteq A \) and \( t \in \mathbb{N} \) as:

\[
\pi_{t,k}(B|H_t) = \begin{cases} 
\pi(B|S_t) & \text{if } t \mod k = 0 \\
\delta_{A_{t-1}}(B) & \text{otherwise} \end{cases}
\]

Moreover, we denote with \( H_k = \{(\pi_{t,k})_{t \in \mathbb{N}} : \pi \in \Pi\} \) the set of the \( k \)-persistent policies.

Clearly, for \( k = 1 \) we recover policy \( \pi \) as we always satisfy the condition \( t \mod k = 0 \) i.e., \( \pi = \pi_{t,1} \) for all \( t \in \mathbb{N} \). We refer to this interpretation of action persistence as *policy view*.

A different perspective towards action persistence consists in looking at the effect of the original policy \( \pi \) in a suitably modified MDP. To this purpose, we introduce the (state-action persistent transition probability kernel \( P^\delta:S \times A \to \mathcal{P}(S \times A) \) defined for any measurable set \( B \subseteq S \times A \) as:

\[
(P^\delta)(s,a) = \int S P(ds'|s,a) \delta_{(s',a)}(B).
\]

The crucial difference between \( P^\pi \) and \( P^\delta \) is that the former samples the action \( a' \) to be executed in the next state \( s' \) according to \( \pi \), whereas the latter replicates in state \( s' \) action \( a \). We are now ready to define the \( k \)-persistent MDP.

**Definition 3.2** (\( k \)-persistent MDP). **Let \( M \) be an MDP. For any \( k \in \mathbb{N}_{\geq 1} \), the \( k \)-persistent MDP is the following MDP \( M_{k} = (S, A, P_k, R_k, \gamma^k) \), where \( P_k \) and \( R_k \) are the \( k \)-persistent transition model and reward distribution respectively, defined for any measurable sets \( B \subseteq S \), \( C \subseteq \mathbb{R} \) and state-action pair \((s,a) \in S \times A\) as:

\[
P_k(B|s,a) = \left( P^\delta \right)^{k-1}(B|s,a),
\]

\[
R_k(C|s,a) = \sum_{i=0}^{k-1} \gamma^i \left( P^\delta \right)^{i} R(C|s,a),
\]

and \( r_k(s,a) = \int_S R_k(ds|s,a) = \sum_{i=0}^{k-1} \gamma^i \left( P^\delta \right)^{i} r(s,a) \) is the expected reward, uniformly bounded by \( R_{\max} \frac{1 - \gamma^k}{1 - \gamma} \).

The \( k \)-persistent transition model \( P_k \) keeps action \( a \) fixed for \( k-1 \) steps while making the state evolve according to \( P \).
Similarly, the k-persistent reward \( R_k \) provides the cumulative discounted reward over \( k \) steps in which \( a \) is persisted. We define the transition kernel \( P^\pi_k \), analogously to \( P^\pi \), as in Equation (1). Clearly, for \( k = 1 \) we recover the base MDP, i.e., \( M = M_1 \). Therefore, executing policy \( \pi \) in \( M_k \) at persistence 1 is equivalent to executing policy \( \pi \) at persistence \( k \) in the original MDP \( M \). We refer to this interpretation of persistence as environment view (Figure 1). Thus, solving the base MDP \( M \) in the space of k-persistent policies \( \Pi_k \) (Definition 3.1), thanks to this duality, is equivalent to solving the k-persistent MDP \( M_k \) (Definition 3.2) in the space of Markovian stationary policies \( \Pi \).

It is worth noting that the persistence \( k \in \mathbb{N}_{\geq 1} \) can be seen as an environmental parameter (affecting \( P \), \( R \), and \( \gamma \)), which can be externally configured with the goal to improve the learning process for the agent. In this sense, the MDP \( M_k \) can be seen as a Configurable Markov Decision Process with parameter \( k \in \mathbb{N}_{\geq 1} \) (Metelli et al., 2018; 2019).

Furthermore, a persistence of \( k \) induces a k-persistent MDP \( M_k \) with smaller discount factor \( \gamma^k \). Therefore, the effective horizon in \( M_k \) is \( \frac{1}{1-\gamma^k} < \frac{1}{1-\gamma} \). Interestingly, the end effect of persisting actions is similar to reducing the planning horizon, by explicitly reducing the discount factor of the task (Petrík & Scherrer, 2008; Jiang et al., 2016) or setting a maximum trajectory length (Farahmand et al., 2016).

### 3.2. Persistent Bellman Operators

When executing policy \( \pi \) at persistence \( k \) in the base MDP \( M \), we can evaluate its performance starting from any state-action pair \( (s,a) \in S \times A \), inducing a Q-function that we denote with \( Q^\pi_k \) and call k-persistent action-value function of \( \pi \). Thanks to duality, \( Q^\pi_k \) is also the action-value function of policy \( \pi \) when executed in the k-persistent MDP \( M_k \). Therefore, \( Q^\pi_k \) is the fixed point of the Bellman Expectation Operator of \( M_k \), i.e., the operator defined for any \( f \in \mathcal{B}(S \times A) \) as \( (T^\pi_k f)(s,a) = r_k(s,a) + \gamma^k P^\pi_k f(s,a) \), that we call k-persistent Bellman Expectation Operator. Similarly, again thanks to duality, the optimal Q-function in the space of k-persistent policies \( \Pi_k \), denoted by \( Q^* \) and called k-persistent optimal action-value function, corresponds to the optimal Q-function of the k-persistent MDP, i.e., \( Q^*_{k}(s,a) = \sup_{\pi \in \Pi} Q^\pi_k(s,a) \) for all \( (s,a) \in S \times A \). As a consequence, \( Q^*_{k} \) is the fixed point of the Bellman Optimal Operator of \( M_k \), defined for \( f \in \mathcal{B}(S \times A) \) as \( (T^k f)(s,a) = r_k(s,a) + \gamma^k \sum_{s' \in A} \max_{a' \in A} f(s',a') \), that we call k-persistent Bellman Optimal Operator. Clearly, both \( T^\pi_k \) and \( T^k \) are \( \gamma^k \)-contractions in \( L_\infty \)-norm.

We now prove that the k-persistent Bellman operators are obtained as composition of the base operators \( T^\pi \) and \( T^k \).

**Theorem 3.1.** Let \( M \) be an MDP, \( k \in \mathbb{N}_{\geq 1} \) and \( M_k \) be the k-persistent MDP. Let \( \pi \in \Pi \) be a Markovian stationary policy. Then, \( T^\pi_k \) and \( T^k \) can be expressed as:

\[
T^\pi_k = (T^\pi)^{k-1} T^\pi \quad \text{and} \quad T^k = (T^\delta)^{k-1} T^k,
\]

where \( T^\pi : \mathcal{B}(S \times A) \rightarrow \mathcal{B}(S \times A) \) is the Bellman Persistent Operator, defined for \( f \in \mathcal{B}(S \times A) \) and \( (s,a) \in S \times A \):

\[
(T^\delta f)(s,a) = r(s,a) + \gamma \sum_{s' \in A} P^\pi f(s',a).
\]

The fixed point equations for the k-persistent Q-functions become: \( Q^\pi_k = (T^\pi)^{k-1} T^\pi Q^\pi_k \) and \( Q^k = (T^k)^{k-1} T^k Q^k \).

### 4. Bounding the Performance Loss

Learning in the space of k-persistent policies \( \Pi_k \) can only lower the performance of the optimal policy, i.e., \( Q^*(s,a) \geq Q^*_{k}(s,a) \) for all \( (s,a) \in S \times A \). The goal of this section is to bound \( ||Q^* - Q^*_{k}||_{p,p} \) as a function of the persistence \( k \in \mathbb{N}_{\geq 1} \). To this purpose, we focus on \( ||Q^* - Q^*_{k}||_{p,p} \) for a fixed policy \( \pi \in \Pi \), since denoting with \( \pi_k \) an optimal policy of \( M_k \) and with \( \pi_k \) an optimal policy of \( M_k \), we have that:

\[
Q^* - Q^*_{k} = Q^* - Q^*_{k} \leq Q^* - Q^*_{k},
\]

since \( Q^*_{k}(s,a) \geq Q^*_{k}(s,a) \). We start with the following result which makes no assumption about the structure of the MDP and then we particularize it for the Lipschitz MDPs.

**Theorem 4.1.** Let \( M \) be an MDP and \( \pi \in \Pi \) be a Markovian stationary policy. Let \( Q_k = \{(T^\delta)^{t-1} T^k Q^k \} : l \in \{0, \ldots, k-2\} \} \) and for all \( (s,a) \in S \times A \) let us define:

\[
d^*_{Q_k}(s,a) = \sup_{f \in \mathcal{B}} \left\| \int_{A} (P^* (ds',da')|s,a) - P^k (ds',da'|s,a) f(s',a') \right\|.
\]

Figure 1. Agent-environment interaction without (top) and with (bottom) action persistence, highlighting duality. The transition generated by the k-persistent MDP \( M_k \) is the cyan dashed arrow, while the actions played by the k-persistent policy are inside the cyan rectangle.
Then, for any $\rho \in \mathcal{P}(S \times A)$, $p \geq 1$, and $k \in \mathbb{N}_{\geq 1}$, it holds that:

$$\|Q^\pi - Q^\rho\|_{p, \rho} \leq \gamma (1 - \gamma^{k+1}) \frac{\|d_{Q_k}^\rho\|_{p, \rho}}{\rho(\gamma)^{-1}}$$

where $\eta_k^\rho \in \mathcal{P}(S \times A)$ is a probability measure defined for any measurable set $B \subseteq S \times A$ as:

$$\eta_k^\rho(B) = \frac{(1 - \gamma)(1 - \gamma^k)}{(1 - \gamma^{k+1})} \sum_{i \mod k = 0}^\infty \gamma^i \rho(\gamma^{-1})^{-1}(B).$$

The bound shows that the Q-function difference depends on the discrepancy $d_{Q_k}^\rho$ between the transition-kernel $P^\pi$ and the corresponding persistent version $P^\delta$, which is a form of integral probability metric (Müller, 1997), defined in terms of the set of $Q_k$. This term is averaged with the distribution that if an action is good in a state, it will also be almost good for states encountered in the near future. Although the environment state changes slowly w.r.t. to time and the policy must play similar actions in similar states. This means that if an action is good in a state, it will also be almost good for states encountered in the near future. Although the dependence on $k$ is represented in the term $\frac{1 - \gamma^k}{1 - \gamma^{k+1}}$. When $k \rightarrow 1$ this term displays a linear growth in $k$, being asymptotic to $(k - 1)\log \frac{1}{k}$, and, clearly, vanishes for $k = 1$. Instead, when $k \rightarrow \infty$ this term tends to 1.

If no structure on the MDP/policy is enforced, the dissimilarity term $d_{Q_k}^\rho$ may become large enough to make the bound vacuous, i.e., larger than $\frac{\|d_{Q_k}^\rho\|_{p, \rho}}{\rho(\gamma)^{-1}}$, even for $k = 2$ (see Appendix B.1). Intuitively, since the persistence will execute old actions in new states, we need to guarantee that the environment state changes slowly w.r.t. to time and the policy must play similar actions in similar states. This means that if an action is good in a state, it will also be almost good for states encountered in the near future. Although the condition on $\pi$ is directly enforced by Assumption 2.2, we need a new notion of regularity over time for the MDP.

**Assumption 4.1.** Let $\mathcal{M}$ be an MDP. $\mathcal{M}$ is $L_T$-Time-Lipschitz Continuous ($L_T$-TLC) if for all $(s, a) \in S \times A$:

$$W_1(P(\cdot | s, a), \delta_s) \leq L_T.$$  

(10)

This assumption requires that the Kantorovich distance between the distribution of the next state $s'$ and the deterministic distribution centered in the current state $s$ is bounded by $L_T$, i.e., the system does not evolve “too fast” (see Appendix B.3). We can now state the following result.

**Theorem 4.2.** Let $\mathcal{M}$ be an MDP and $\pi \in \Pi$ be a Markovian stationary policy. Under Assumptions 2.1, 2.2, and 4.1, if $\gamma \max\{L_{P_1} + L_{P_1}(1 + L_T)\} < 1$ and if $\rho(s,a) = \rho_S(s)(a|s)$ with $\rho_S \in \mathcal{P}(S)$, then for any $k \in \mathbb{N}_{\geq 1}$:

$$\left\|d_{Q_k}^\rho\right\|_{p, \rho} \leq L_{Q_k}(L_{P_1} + L_T + \sigma_p),$$

where $\sigma_p = \sup_{s \in S} \int_s d_A(a, a') \pi_d(a|s) \pi_d(da'|s)$. and $\eta_k^\rho$ resembles the $\gamma$-discounted state-action distribution (Sutton et al., 1999a), but ignoring the decision steps multiple of $k$.

$$L_{Q_k} = \frac{\gamma^k}{1 - \gamma^k} \left\|L_{P_1} + L_{P_1}(1 + L_T)\right\|_{p, \rho}.$$ Thus, the dissimilarity $d_{Q_k}^\rho$ between $P^\pi$ and $P^\delta$ can be bounded with four terms. i) $L_{Q_k}$ is (an upper-bound of) the Lipschitz constant of the functions in the set $Q_k$. Indeed, under Assumptions 2.1 and 2.2 we can reduce the dissimilarity term to the Kantorovich distance (Lemma A.5):

$$d_{Q_k}(s, a) \leq L_{Q_k}(P^\pi(\cdot | s, a), P^\delta(\cdot | s, a)),$$

ii) $(L_{P_1} + 1)$ accounts for the Lipschitz continuity of the policy, i.e., policies that prescribe similar actions in similar states have a small value of this quantity. iii) $L_T$ represents the speed at which the environment state evolves over time. iv) $\sigma_p$ denotes the average distance (in $L_p$-norm) between two actions prescribed by the policy in the same state. This term is zero for deterministic policies and can be related to the maximum policy variance (Lemma A.6). A more detailed discussion on the conditions requested in Theorem 2.4 is reported in Appendix B.4.

5. Persistent Fitted Q-Iteration

In this section, we introduce an extension of Fitted Q-Iteration (FQI, Ernst et al., 2005) that employs the notion of persistence.4 Persistent Fitted Q-Iteration (PFQI($k$)) takes as input a target persistence $k \in \mathbb{N}_{\geq 1}$ and its goal is to approximate the $k$-persistent optimal action-value function $Q_k^\pi$. Starting from an initial estimate $Q^{(0)}$, at each iteration we compute the next estimate $Q^{(j+1)}$ by performing an approximate application of $k$-persistent Bellman optimal operator to the previous estimate $Q^{(j)}$, i.e., $Q^{(j+1)} = T_k^Q Q^{(j)}$. In practice, we have two sources of approximation in this process: i) the representation of the Q-function; ii) the estimation of the $k$-persistent Bellman optimal operator. (i) comes from the necessity of using functional space $F \subset \mathcal{H}(S \times A)$ to represent $Q^{(j)}$ when dealing with continuous state spaces. (ii) derives from the approximate computation of $T_k^Q$ which needs to be estimated from samples.

Clearly, with samples collected in the $k$-persistent MDP $\mathcal{M}_k$, the process described above reduces to the standard FQI. However, our algorithm needs to be able to estimate $Q_k^\pi$ for different values of $k$, using the same dataset of samples collected in the base MDP $\mathcal{M}$ (at persistence 1).5 For this purpose, we can exploit the decomposition $T_k^Q = (T_0)^{k-1} T_k^Q$ of Theorem 3.1 to reduce a single application of $T_k^Q$ to a sequence of $k$ applications of the 1-persistent operators. Specifically, at each iteration $j$ with $j \mod k = 0$, given the current estimate $Q^{(j)}$, we need to perform (in this order) a single application of $T_k^Q$ followed by $k-1$ application...
Algorithm 1 Persistent Fitted Q-Iteration PFQI($k$).

**Input:** $k$ persistence, $J$ number of iterations ($J \mod k = 0$), $Q^{(0)}$ initial action-value function, $\mathcal{F}$ functional space, $\mathcal{D} = \{(S_i, A_i, S'_i, R_i^k)\}_{i=1}^n$ batch samples

**Output:** greedy policy $\pi^{(j)}$ for $j = 0, \ldots, J - 1$

```
for $j = 0, \ldots, J - 1$ do
    if $j \mod k = 0$ then
        $Y^{(j)}_i = T^* Q^{(j)}(S_i, A_i), \quad i = 1, \ldots, n$
    else
        $Y^{(j)}_i = \tilde{T}^Q(j)(S_i, A_i), \quad i = 1, \ldots, n$
    end if
end for

$Q^{(j+1)} = \arg\min_{f \in \mathcal{F}} \|f - Y^{(j)}\|_2^2$ for $j = 0, \ldots, J - 1$

**Error Propagation** We now consider the error propagation in PFQI($k$). Given the sequence of Q-functions estimates $(Q^{(j)}_{j=0}^{(k)}) \subset \mathcal{F}$ produced by PFQI($k$), we define the approximation error at each iteration $j = 0, \ldots, J - 1$ as:

$$
\epsilon^{(j)} = \begin{cases} 
T^* Q^{(j)} - Q^{(j+1)} & \text{if } j \mod k = 0 \\
T^* Q^{(j)} - Q^{(j+1)} & \text{otherwise}
\end{cases}
$$

The goal of this analysis is to bound the distance between the $k$-persistent optimal Q-function $Q^s_k$ and the Q-function $Q^{(j)}_k$ of the greedy policy $\pi^{(j)}$ w.r.t. $Q^{(j)}$, after $J$ iterations of PFQI($k$). The following result extends Theorem 3.4 of Farahmand (2011) to account for action persistence.

**Theorem 5.1** (Error Propagation for PFQI($k$)). Let $p \geq 1$, $k \in \mathbb{N}_{\geq 1}$, $J \in \mathbb{N}_{\geq 1}$ with $J \mod k = 0$ and $\rho \in \mathcal{P}(\mathcal{S} \times \mathcal{A})$. Then for any sequence $(Q^{(j)}_{j=0}^{(k)}) \subset \mathcal{F}$ uniformly bounded by $Q_{\max} \leq \frac{R_{\max}}{1 - \gamma}$, the corresponding $(\epsilon^{(j)}_{j=0}^{(k)})$ defined in Equation (12) and for any $r \in [0, 1]$ and $q \in [1, +\infty]$ it holds that:

$$
\left\| Q^{s}_k - Q^{(j)}_k \right\|_{p, \rho} \leq \frac{2\gamma^k}{(1 - \gamma)(1 - \gamma^k)} \left[ \frac{2}{1 - \gamma} R_{\max} \right. \\
+ C_{\mathcal{F}, \mathcal{P}, \mathcal{D}}(J, r, q)\mathcal{E}(\epsilon_0, \ldots, \epsilon_{(J-1)}, r, q) \right]
$$

The expression of $C_{\mathcal{F}, \mathcal{P}, \mathcal{D}}(J, r, q)$ and $\mathcal{E}(\cdot; r, q)$ can be found in Appendix A.3.

We immediately observe that for $k = 1$ we recover Theorem 3.4 of Farahmand (2011). The term $C_{\mathcal{F}, \mathcal{P}, \mathcal{D}}(J, r, q)$ is defined in terms of suitable concentrability coefficients (Definition A.1) and encodes the distribution shift between the sampling distribution $\nu$ and the one induced by the greedy policy sequence $(\pi^{(j)}_{j=0}^{(k)})$ encountered along the execution of PFQI($k$). $\mathcal{E}(\cdot; r, q)$ incorporates the approximation errors $(\epsilon^{(j)}_{j=0}^{(k)})$.

In principle, it is hard to compare the values of these terms for different persistences $k$ since both the greedy policies and the regression problems are different. Nevertheless, it is worth noting that the multiplicative term $rac{\gamma^k}{1 - \gamma}$ decreases in $k \in \mathbb{N}_{\geq 1}$. Thus, other things being equal, the bound value decreases when increasing the persistence.

Thus, the trade-off in the choice of control frequency, which motivates action persistence, can now be stated more formally. We aim at finding the persistence $k \in \mathbb{N}_{\geq 1}$ that, for a fixed $J$, allows learning a policy $\pi^{(j)}$ whose Q-function $Q^{(j)}_k$ is the closest to $Q^s$. Consider the decomposition:

$$
\left\| Q^s - Q^{(j)}_k \right\|_{p, \rho} \leq \left\| Q^s - Q^s_k \right\|_{p, \rho} + \left\| Q^s_k - Q^{(j)}_k \right\|_{p, \rho}.
$$

The term $\left\| Q^s - Q^s_k \right\|_{p, \rho}$ accounts for the performance degradation due to action persistence: it is algorithm-independent, and it increases in $k$ (Theorem 4.1). Instead, the second term $\left\| Q^s - Q^{(j)}_k \right\|_{p, \rho}$ decreases with $k$ and depends on the algo-
Algorithm 2 Heuristic Persistence Selection.

**Input:** batch samples $D = \{(S_0, A_0, \ldots, S_{H_t-1}, A_{H_t-1}, S_H_t)\}_{t=1}^m$, set of persistences $\mathcal{K}$, set of Q-function $\{Q_k:k \in \mathcal{K}\}$, regressor $\tilde{R}$

**Output:** approximately optimal persistence $\tilde{k}$

for $k \in \mathcal{K}$ do

$J_k = \frac{1}{m} \sum_{i=1}^m V_k(S_i)$

Use the Reg to get an estimate $\tilde{Q}_k$ of $T_k^\gamma Q_k$

$||\tilde{Q}_k - Q_k||_{1,D} = \frac{1}{1-\gamma^\rho} ||\tilde{Q}_k - Q_k||_{1,D}$

end for

$k^* \in \arg\max_{k \in \mathcal{K}} J_k \rho, \pi$,

In principle, we could execute $\pi_k$ in $M_k$ to get an estimate of $J_k^{\pi,\gamma}$ and employ it to select the persistence $k$. However, in the batch setting, further interactions with the environment might not be allowed. On the other hand, directly using the estimated Q-function $Q_k$ is inappropriate, since we need to take into account how well $Q_k$ approximates $Q_k^{\pi_k}$. This trade-off is encoded in the following result, which makes use of the expected Bellman residual.

**Lemma 6.1.** Let $Q \in \mathcal{R}(S \times A)$ and $\pi$ be a greedy policy w.r.t. $Q$. Let $J^\rho = \sum(s \in S) V(s)$, with $V(s) = \max_{a \in A} Q(s,a)$ for all $s \in S$. Then, for any $k \in \mathbb{N}_+ \setminus 1$, it holds that:

$$J_k^{\rho,\gamma} \geq J^\rho - \frac{1}{1-\gamma^\rho} \left( \mathbb{T}_k^\gamma Q - Q \right)_{1,\eta,\gamma,\pi},$$

where $\eta = (1-\gamma) \rho \pi (\mathbb{I} - \gamma^\rho P_k^\pi)^{-1}$, is the $\gamma$-discounted stationary distribution induced by policy $\pi$ and distribution $\rho$ in MDP $M_k$.

To get a usable bound, we need to make some simplifications. First, we assume that $D \sim \nu$ is composed of $m$ trajectories, i.e., $D = \{(S_0, A_0, \ldots, S_{H_t-1}, A_{H_t-1}, S_H_t)\}_{t=1}^m$, where $H_t$ is the trajectory length and the initial states are sampled as $S_0 \sim \rho$. In this way, $J^\rho$ can be estimated from samples as $J^\rho = \frac{1}{m} \sum_{i=1}^m V(S_i)$. Second, since we are unable to compute expectations over $\eta$, we replace it with the sampling distribution $\nu$.\footnote{For instance, the $Q_k$ can be obtained by executing PFQI($k$) with different persistences $k \in \mathcal{K}$.} Lastly, estimating the expected Bellman residual is problematic since its empirical version is biased (Antos et al., 2008). Thus, we resort to an approach similar to (Farahmand & Szepesvári, 2011), assuming to have a regressor $\tilde{R}$ able to output an approximation $\tilde{Q}_k$ of $T_k^\gamma Q$. In this way, we replace $\|T_k^\gamma Q - Q\|_{1,\nu}$ with $\|\tilde{Q}_k - Q\|_{1,D}$ (details in Appendix C). In practice, we set $Q = Q^{J(\cdot)}$ and we obtain $\tilde{Q}_k$ running PFQI($k$) for $k$ additional iterations, setting $\tilde{Q}_k = Q^{J(\cdot+k)}$. Thus, the procedure (Algorithm 2) reduces to optimizing the index:

$$\tilde{k} \in \arg\max_{k \in \mathcal{K}} B_k = \tilde{J}_k^\rho - \frac{1}{1-\gamma^\rho} ||\tilde{Q}_k - Q_k||_{1,D}.$$ (15)

### 7. Related Works

In this section, we revise the works connected to persistence, focusing on continuous–time RL and temporal abstractions.

**Continuous–time RL** Among the first attempts to extend value–based RL to the continuous–time domain there is advantage updating (Bradtke & Duff, 1994), in which Q–learning (Watkins, 1989) is modified to account for infinitesimal control timesteps. Instead of storing the Q–function, the advantage function $A(s,a) = Q(s,a) - V(s)$ is recorder. The continuous time is addressed in Baird (1994) by means of the semi–Markov decision processes (Howard, 1963) for finite–state problems. The optimal control literature has extensively studied the solution of the Hamilton–Jacobi–Bellman equation, i.e., the continuous–time counterpart of the Bellman equation, when assuming the knowledge of the environment (Bertsekas, 2005; Fleming & Soner, 2006). The model–free case has been tackled by resorting to time (and space) discretizations (Peterson, 1993), with also convergence guarantees (Munos, 1997; Munos & Bourgine, 1997), and coped with function approximation (Dayan & Singh, 1995; Doya, 2000). More recently, the sensitivity of deep RL algorithms to the temporal discretization has been analyzed in Tallec et al. (2019), proposing an adaptation of advantage updating to deal with small time scales, that can be employed with deep architectures.

**Temporal Abstractions** The notion of action persistence can be seen as a form of temporal abstraction (Sutton et al., 1999b; Precup, 2001). Temporally extended actions have been extensively used in the hierarchical RL literature to model different time resolutions (Singh, 1992a,b), subgoals (Dietrich, 1998), and combined with the actor–critic architectures (Bacon et al., 2017). Persisting an action is a particular instance of a semi–Markov option, always lasting $k$ steps. According to the flat option representation (Precup, 2001), we have as initiation set $\mathcal{Z} = \mathcal{S}$ the set of all states, as internal policy the policy that plays deter
Control Frequency Adaptation via Action Persistence in Batch Reinforcement Learning

Table 1. Results of PFQI in different environments and persistences. For each persistence \( k \), we report the sample mean and the standard deviation of the estimated return of the last policy \( J_{\rho,\pi}^k \). For each environment, the persistence with highest average performance and the ones not statistically significantly different from that one (Welch’s t-test with \( p < 0.05 \)) are in bold. The last column reports the mean and the standard deviation of the performance loss \( \delta \) between the optimal persistence and the one selected by the index \( B_k \) (Equation (15)).

<table>
<thead>
<tr>
<th>Environment</th>
<th>Expected return at persistence ( k ) (( J_{\rho,\pi}^k ), mean ± std)</th>
<th>Performance loss (( \delta ) mean ± std)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( k=1 )</td>
<td>( k=2 )</td>
</tr>
<tr>
<td>Cartpole</td>
<td>169.9 ± 5.8</td>
<td>176.5 ± 5.0</td>
</tr>
<tr>
<td>MountainCar</td>
<td>-111.1 ± 1.5</td>
<td>-103.6 ± 1.6</td>
</tr>
<tr>
<td>LunarLander</td>
<td>-165.8 ± 50.4</td>
<td>-12.8 ± 4.7</td>
</tr>
<tr>
<td>Pendulum</td>
<td>-116.7 ± 16.7</td>
<td>-113.1 ± 16.3</td>
</tr>
<tr>
<td>Acrobat</td>
<td>-89.2 ± 1.1</td>
<td>-82.5 ± 1.7</td>
</tr>
<tr>
<td>Swimmer</td>
<td>21.3 ± 1.1</td>
<td>25.2 ± 0.8</td>
</tr>
<tr>
<td>Hopper</td>
<td>58.6 ± 4.8</td>
<td>61.9 ± 4.2</td>
</tr>
<tr>
<td>Walker 2D</td>
<td>61.6 ± 5.5</td>
<td>37.6 ± 4.0</td>
</tr>
</tbody>
</table>

minimistically the action taken when the option was initiated, i.e., the \( k \)-persistent policy, and as termination condition whether \( k \) timesteps have passed after the option started, i.e., \( \beta(H_t) = 1 \mod k = 0 \). Interestingly, in Mann et al. (2015) an approximate value iteration procedure for options lasting at least a given number of steps is proposed and analyzed. This approach shares some similarities with action persistence. Nevertheless, we believe that the option framework is more general and usually the time abstractions are related to the semantic of the tasks, rather than based on the modification of the control frequency, like action persistence.

8. Experimental Evaluation

In this section, we provide the empirical evaluation of PFQI, with the threefold goal: i) proving that a persistence \( k > 1 \) can boost learning, leading to more profitable policies, ii) assessing the quality of our persistence selection method, and iii) studying how the batch size influences the performance of PFQI policies for different persistences. Refer to Appendix D for detailed experimental settings.

We train PFQI, using extra-trees (Geurts et al., 2006) as a regression model, for \( J \) iterations and different values of \( k \), starting with the same dataset \( D \) collected at persistence 1. To compare the performance of the learned policies \( \pi_k \) at the different persistences, we estimate their expected return \( J_{\rho,\pi}^k \) in the corresponding MDP \( \mathcal{M}_k \). Table 1 shows the results for different continuous environments and different persistences averaged over 20 runs and highlighting in bold the persistence with the highest average performance and the ones that are not statistically significantly different from that one. Across the different environments we observe some common trends in line with our theory: i) persistence 1 rarely leads to the best performance; ii) excessively increasing persistence prevents the control at all. In Cartpole (Barto et al., 1983), we easily identify a persistence \( k=4 \) that outperforms all the others. In the Lunar Lander (Brockman et al., 2016) persistences \( k \in \{4,8\} \) are the only ones that lead to positive return (i.e., the lander does not crash) and in the Acrobat domain (Geramifard et al., 2015) we identify \( k \in \{2,4\} \) as optimal persistences. A qualitatively different behavior is displayed in Mountain Car (Moore, 1991), Pendulum (Brockman et al., 2016), and Swimmer (Coulom, 2002), where we observe a plateau of three persistences with similar performance. An explanation for this phenomenon is that, in those domains, the optimal policy tends to persist actions on its own, making the difference less evident. Intriguingly, the more complex Mujoco domains, like Hopper and Walker 2D (Erickson et al., 2019), seem to benefit from the higher persistences.

To test the quality of our persistence selection method, we compare the performance of the estimated optimal persistence, i.e., the one with the highest estimated expected return \( \hat{k} = \arg \max_k J_{\rho,\pi}^k \), and the performance of the persistence \( \hat{k} \) selected by maximizing the index \( B_k \) (Equation (15)). For each run \( i=1,\ldots,20 \), we compute the performance loss \( \delta_i = J_{\rho,\pi_{\hat{k}}} - J_{\rho,\pi_i} \) and we report it in the last column of Table 1. In the Cartpole experiment, we observe a zero loss, which means that our heuristic always selects the optimal persistence \( (k=4) \). Differently, non–zero loss occurs in the other domains, which means that sometimes the index \( B_k \) mispredicts the optimal persistence. Nevertheless, in almost all cases the average performance loss is significantly smaller than the magnitude of the return, proving the effectiveness of our heuristics.

In Figure 2, we show the learning curves for the Cartpole experiment, highlighting the components that contribute to the index \( B_k \). The first plot reports the estimated expected return \( J_{\rho,\pi}^k \), obtained by averaging 10 trajectories executing \( \pi_k \) in the environment \( \mathcal{M}_k \), which confirms that \( k=4 \) is the optimal persistence. The second plot shows the estimated return \( J_{\rho}^k \) obtained by averaging the Q-function \( Q_k \) learned with PFQI(\( k \)), over the initial states sampled from \( \rho \). We can see that for \( k \in \{1,2\} \), PFQI(\( k \)) tends to overestimate the return, while for \( k=4 \) we notice a slight underestimation. The overestimation phenomenon can be explained by the fact that with small persistences we perform a large number
For instance, in Mountain Car, high persistences increase the amount of samples is limited, PFQI can exploit higher market prices, this environment is very noisy, thus, when the batch size is small ($n \in \{10, 30, 50\}$), higher persistences ($k \in \{2, 4, 8\}$) result in better performances, while for larger batch sizes, $k = 1$ becomes the best choice. Since data is taken from real market prices, this environment is very noisy, thus, when the amount of samples is limited, PFQI can exploit higher persistences to mitigate the poor estimation.

To analyze the effect of the batch size, we run PFQI on the Trading environment (see Appendix D.4) varying the number of sampled trajectories. In Figure 3, we notice that the performance improves as the batch size increases, for all persistences. Moreover, we observe that if the batch size is small ($n \in \{10, 30, 50\}$), higher persistences ($k \in \{2, 4, 8\}$) result in better performances, while for larger batch sizes, $k = 1$ becomes the best choice. Since data is taken from real market prices, this environment is very noisy, thus, when the amount of samples is limited, PFQI can exploit higher persistences to mitigate the poor estimation.

9. Open Questions

Improving Exploration with Persistence We analyzed the effect of action persistence on FQI with a fixed dataset, collected in the base MDP $M$. In principle, samples can be collected at arbitrary persistence. We may wonder how well the same sampling policy (e.g., the uniform policy over $A$), executed at different persistences, explores the environment. For instance, in Mountain Car, high persistences increase the probability of reaching the goal, generating more informative datasets (preliminary results in Appendix E.1).

Learn in $M_k$ and execute in $M_k'$deploying each policy $\pi_k$ in the corresponding MDP $M_k$ allows for some guarantees (Lemma 6.1). However, we empirically discovered that using $\pi_k$ in an MDP $M_k$ with smaller persistence $k'$ sometimes improves its performance. (preliminary results in Appendix E.2). We wonder what regularity conditions on the environment are needed to explain this phenomenon.

Persistence in On–line RL our approach focuses on batch off–line RL. However, the on–line framework could open up new opportunities for action persistence. Specifically, we could dynamically adapt the persistence (and so the control frequency) to speed up learning. Intuition suggests that we should start with a low frequency, reaching a fairly good policy with few samples, and then increase it to refine the learned policy.

10. Discussion and Conclusions

In this paper, we formalized the notion of action persistence, i.e., the repetition of a single action for a fixed number $k$ of decision epochs, having the effect of altering the control frequency of the system. We have shown that persistence leads to the definition of new Bellman operators and that using $\pi_k$ in an MDP $M_k$ with smaller persistence $k'$ sometimes improves its performance. (preliminary results in Appendix E.2). We wonder what regularity conditions on the environment are needed to explain this phenomenon. The experimental evaluation justifies the introduction of persistence, since reducing the control frequency can lead to an improvement when dealing with a limited number of samples. Furthermore, we introduced a persistence selection heuristic, which is able to identify good persistence in most cases. We believe that our work makes a step towards understanding why repeating actions may be useful for solving complex control tasks. Numerous questions remain unanswered, leading to several appealing future research directions.
Acknowledgements

The research was conducted under a cooperative agreement between ISI Foundation, Banca IMI and Intesa Sanpaolo Innovation Center.

References


Coulom, R. *Reinforcement Learning Using Neural Networks, with Applications to Motor Control. (Apprentissage par renforcement utilisant des réseaux de neurones, avec des applications au contrôle moteur)*. PhD thesis, Grenoble Institute of Technology, France, 2002.


Control Frequency Adaptation via Action Persistence in Batch Reinforcement Learning


