2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

Non-Stationary Policy Learning for Multi-Timescale Multi-Agent Reinforcement Learning

Patrick Emami, Xiangyu Zhang, David Biagioni and Ahmed S. Zamzam

***Abstract*— In multi-timescale multi-agent reinforcement learning (MARL), agents interact across different timescales. In general, policies for time-dependent behaviors, such as those induced by multiple timescales, are non-stationary. Learning non-stationary policies is challenging and typically requires sophisticated or inefficient algorithms. Motivated by the prevalence of this control problem in real-world complex systems, we introduce a simple framework for learning non-stationary policies for multi-timescale MARL. Our approach uses available information about agent timescales to define and learn periodic multi-agent policies. In detail, we theoretically demonstrate that the effects of non-stationarity introduced by multiple timescales can be learned by a periodic multi-agent policy. To learn such policies, we propose a policy gradient algorithm that parameterizes the actor and critic with phase-functioned neural networks, which provide an inductive bias for periodicity. The framework's ability to effectively learn multi-timescale policies is validated on a gridworld and building energy management environment.**

I. INTRODUCTION

The ability to control multiple interacting components is essential to efficiently manage complex systems. For instance, in power systems applications, flexible loads and distributed generation operate within this complex system with coupling in the dynamical models, constraints, and objectives. Similar multi-component control challenges appear in transportation systems, robotics, etc. Thus, interest in data-driven approaches that try to learn distributed control policies from experience has grown recently. Multi-agent reinforcement learning (MARL) is a promising agent-based sequential decision-making framework for learning complex coordination strategies. Crucially, it does not depend on having access to a model of the (often stochastic and nonlinear) environment dynamics.

This work was authored in part by the National Renewable Energy Laboratory (NREL), operated by Alliance for Sustainable Energy, LLC, for the U.S. Department of Energy (DOE) under Contract No. DE-AC36-08GO28308. This work was supported by the Laboratory Directed Research and Development (LDRD) Program at NREL. The views expressed in the article do not necessarily represent the views of the DOE or the U.S. Government. The U.S. Government retains and the publisher, by accepting the article for publication, acknowledges that the U.S. Government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this work, or allow others to do so, for U.S. Government purposes. A portion of this research was performed using computational resources sponsored by the Department of Energy's Office of Energy Efficiency and Renewable Energy and located at the National Renewable Energy Laboratory.

Patrick Emami, Xiangyu Zhang, and Ahmed Zamzam are with the National Renewable Energy Laboratory, Golden, CO 80401 USA (e-mail: {patrick.emami, xiangyu.zhang, ahmed.zamzam}@nrel.gov.

David Biagioni is with Maplewell Energy, Broomfield, CO 80021, USA (email: dave@maplewelleng.com).

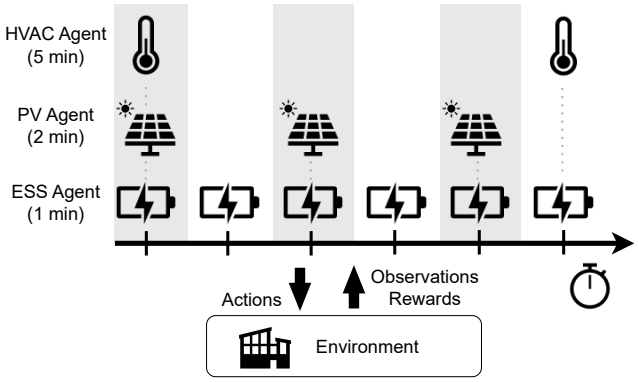

Fig. 1. In **Multi-timescale MARL**, agents that act on different timescales try to learn to coordinate to achieve a goal, e.g., controlling a heterogeneous set of electronic devices for building energy optimization. Learning a *non-stationary* multi-agent policy allows multi-timescale agents to perform complex time-dependent behaviors.

However, applying MARL to real-world problems is challenging because agents typically only receive noisy, partial observations of the environment state and have limited communication with each other. Moreover, from the perspective of each agent, the environment dynamics appear to shift over time as other agents learn to adapt their behaviors. All of these factors contribute to the extreme *non-stationarity* present in MARL, which makes learning good agent policies notoriously challenging [1], [2]. Despite the dedication of significant effort to discover practical MARL algorithms capable of learning policies in the face of non-stationarity [3], [4], there remains a gap between the synthetic environments used for algorithm development and the real world.

This work studies an under-explored MARL setting inspired by real world applications where agents need to coordinate *time-dependent* actions across different timescales. This type of time-dependent coordination, arises, for example, in:

Ex 1: Power systems applications where we wish to learn a coordination strategy between electrical devices that can be controlled at different timescales (e.g., energy storage units, solar panels, and thermostatically controlled loads) connected to the same micro-grid (Fig. 1),

Ex 2: Robotic control tasks where multiple heterogeneous robots try to collaborate to execute a task (e.g., moving an object) while their actuators are controlled at different frequencies.

Introducing a time dependency via multiple timescales to MARL adds *additional* sources of non-stationarity, which makes learning policies particularly challenging.

In this work, we formally define the multi-timescale

decentralized partially observable Markov decision process (Dec-POMDP) setting and propose a framework for learning multi-agent time-dependent (i.e., non-stationary) policies to solve such environments. We show that multiple timescales induces the optimal multi-timescale policy to be periodic in nature. A practical policy gradient method for learning periodic multi-agent policies based on phase-functioned neural networks [5] is provided. Our framework's ability to learn effective multi-timescale policies with fewer environment interactions than key baselines is validated on a gridworld and a building energy management environment.

II. Multi-timescale Dec-POMDPs

First, we define a Dec-POMDP [6], which is a multi-agent stochastic game defined by a tuple:

$$\texttt{DEC-POMDP} := (N, S, \mathbf{A}, \mathbf{O}, r, T, dt, \gamma, \rho, p, P). \quad (1)$$

The number of agents is N, S is the state space of the environment, $\mathbf{A} := \times_{i=1}^{N} A^i$ is a joint action space, $\mathbf{O} := \times_{i=1}^{N} O^i$ is a joint observation space, $r(s, a) : S \times \mathbf{A} \to \mathbb{R}$ is a global reward function, T is the (possibly infinite) horizon, dt is the time discretization of the environment, $\gamma \in [0, 1]$ is a discount factor, and $\rho : S \to [0, 1]$ is the initial state distribution. The joint action $\mathbf{a} \in \mathbf{A}$ induces a transition from state s to state s' according to the transition function $p(s, \mathbf{a}) : S \times \mathbf{A} \to S$. We assume that each agent is only given a noisy, partial observation of the state governed by an observation function $P(s, i) : S \times \mathbb{N} \to O^i$, where i is the agent index, and that there is limited or no communication between agents. Each agent learns a stochastic policy function $\pi^i(a^i \mid o^i)$ that maps an observation to a distribution over A^i, and the joint policy is $\boldsymbol{\pi} = \{\pi^i, \ldots, \pi^N\}$.

In this work, we introduce multi-timescale Dec-POMDPs as an extension of Dec-POMDPs:

$$\texttt{MT-DEC-POMDP} := [\texttt{DEC-POMDP}; k, C], \quad (2)$$

where $[;]$ is concatenation. The agents are defined with action frequencies $k := \{k_1, \ldots, k_N\}$ and the environment is defined to be periodic with period $C \geq 1$, i.e., the reward function $r_t(s, \mathbf{a})$ and transition functions $p_t(s, \mathbf{a})$ are C-periodic. When $C = 1$, the environment is aperiodic, as is typically assumed in a DEC-POMDP. Each agent's timescale is defined by the action frequency k_i times the base timescale discretization dt. For example, agent 1 with $k_1 = 2$ acts every two steps (i.e., when $t \bmod 2dt = 0$) and agent 2 with $k_2 = 3$ acts every three steps (i.e., when $t \bmod 3dt = 0$). We assume environments are defined with $dt = 1$ for simplicity in the remainder of the paper. Between actions, "slow" agents take a null action $a^i_{\texttt{null}}$ or repeat the most recently taken action (e.g., when an action represents the setpoint for a device). Our setting differs from the multi-timescale MARL setting considered in Wu et al. [7] as they assume agents can communicate locally with their neighbors and that the environment is aperiodic.

One can observe that the sequencing of actions across agent timescales repeats periodically every $\tilde{K} = \text{LCM}(k_1, \ldots, k_N)$ time steps, where LCM is the least common multiple. Taking into account the periodicity of the environment C, we see that the pattern of action and state transition sequencing repeats every $K = \text{LCM}(\tilde{K}, C)$ steps. When $K > 1$, solving a MT-DEC-POMDP is generally more difficult than solving a DEC-POMDP, as the periodicity introduces a time dependency that compounds on top of the non-stationarity caused by partial observability and limited communication between agents [8], [3].

To help motivate our framework, we now briefly describe a class of MARL environments where the optimal stationary joint policy is sub-optimal (i.e., when the agents ignore time dependencies). In detail, certain MARL environments cause agents to suffer from what we call the *observation aliasing problem*. In Sec. V-A, we provide an example multi-timescale environment exhibiting observation aliasing. This aliasing problem, which is closely related to the state aliasing problem that can arise in single agent RL [9], occurs when agent i receives a specific partial observation o^i that has a different action under non-stationary policy π^i depending on the time step:

$$\exists t, t' \text{ with } t < t' \text{ s.t. } \pi^i_t(o^i_t) \neq \pi^i_{t'}(o^i_{t'}).$$

A *time-unaware* agent is not able to learn a stationary policy that distinguishes between o^i_t and $o^i_{t'}$ due to the impossibility of proper credit assignment. Intuitively, time-unaware agents perceive time-dependent rewards or dynamics as stochasticity in the environment (e.g., from an exogenous source). Consequentially, these agents learn suboptimal policies that try to explain the aliased observation as perceived randomness.

A common heuristic is to make each agent time aware by appending the current time step t to the observation. Alternatively, recurrent neural networks can be used to infer a *belief state* for each agent based on its history [10]. In this work, we explore how information about agent timescales and environment periodicity can be used to more effectively learn non-stationary policies in MT-DEC-POMDPs.

Recent work has introduced custom multi-timescale solutions for power systems problems that simply attempt to learn a stationary policy [11] or use recurrent neural networks to model temporal information for a fast agent and a slow agent [12]. The single agent RL setting with multiple action frequencies is also a closely related setting. Multi-timescale MDPs (MMDPs) [13], [14] involve an agent formulated as a hierarchical MDP with fast and slow actions. MMDPs aim to learn a (top-down) hierarchy over action timescales, in the sense that actions taken on slower timescales influence the actions taken on faster timescales, but not vice versa. Also related is a setting where a factored-action MDP agent can choose actions that persist for various lengths of time [15].

III. K-periodic Non-stationary Joint Policies

We now demonstrate that, under simplifying assumptions, policy iteration in the space of *K-periodic* non-stationary joint policies converges to the optimal multi-timescale policy and value function.

Definition 1: For any $K \geq 1$, a K-periodic non-stationary joint policy satisfies

$$\bar{\boldsymbol{\pi}}(\mathbf{o}_t) = \bar{\boldsymbol{\pi}}(\mathbf{o}_{t+K}). \tag{3}$$

Let $\bar{\boldsymbol{\Pi}}_K = \{\bar{\boldsymbol{\pi}} : \bar{\boldsymbol{\pi}}(\mathbf{o}_t) = \bar{\boldsymbol{\pi}}(\mathbf{o}_{t+K})\}$ be the set of all such policies. We use this to define a *multi-timescale non-stationary joint policy* $\boldsymbol{\pi}_t \in \boldsymbol{\Pi}$ with timescale action frequencies k as the policy *induced* by a K-periodic non-stationary policy:

$$\forall t, \boldsymbol{\pi}_t = \{\pi_t^1, \ldots, \pi_t^N\} \text{ s.t.} \tag{4}$$

$$\pi_t^i := \begin{cases} \bar{\pi}_t^i & \text{if } t \bmod k_i = 0 \\ \delta_{t-(t \bmod k_i)}(a^i) \text{ or } \delta_t(a^i_{\texttt{null}}) & \text{otherwise.} \end{cases}$$

Here, δ_t is the Dirac delta function. In general, the inducing policy for a multi-timescale policy $\boldsymbol{\pi}_t$ does not need to be K-periodic. However, later in this section we prove that policy iteration in the space of K-periodic policies converges to the optimal multi-timescale policy assuming cooperative agents and full observability.

Definition 2: The projection operator for joint actions $\Gamma_t^k(\mathbf{a}_t)$ is defined as

$$\Gamma_t^k(\mathbf{a}_t) = \{\bar{a}_t^1, \ldots, \bar{a}_t^N\}, \text{ where} \tag{5}$$

$$\bar{a}_t^i := \begin{cases} a_t^i & \text{if } t \bmod k_i = 0 \\ \bar{a}_{t-1}^i \text{ or } a^i_{\texttt{null}} & \text{otherwise.} \end{cases} \tag{6}$$

Notice that Γ_t^k is $\tilde{K}$-periodic. That is, $\Gamma_t^k(\mathbf{a}) = \Gamma_{t+\tilde{K}}^k(\mathbf{a})$ for any t and joint action $\mathbf{a}$.

Definition 3: The projection operator for the reward and transition function is defined as

$$\Theta_t^C(f_t) = \begin{cases} f_t^0 & \text{if } t \bmod C = 0 \\ \vdots & \\ f_t^{C-1} & \text{if } t \bmod C = C-1. \end{cases} \tag{7}$$

Notice that Θ_t^C is C-periodic, i.e., $\Theta_t^C(r(s,\mathbf{a})) = \Theta_{t+C}^C(r(s,\mathbf{a}))$ for any t and $(s,\mathbf{a})$.

The agents aim to learn an optimal multi-timescale policy

$$\boldsymbol{\pi}^* = \arg\max_{\boldsymbol{\pi}\in\Pi} \mathbb{E}_{\boldsymbol{\pi}, \Theta_t^C(p_t(s,a))}\left[\sum_{t=0}^{T} \gamma^t \Theta_t^C\big(r_t(s_t, \Gamma_t^k(\mathbf{a}_t))\big)\right], \tag{8}$$

and the optimal multi-timescale value function is

$$Q_t^{\boldsymbol{\pi}^*}(a,s) = \mathbb{E}_{\mathbf{a_{t+1}}:\mathbf{a_T}, s_{t+1}:s_T}\big[R_t | \Gamma_t^k(\mathbf{a_t}), s_t, \boldsymbol{\pi}^*\big], \tag{9}$$

where $R_t = \sum_{\tau=t}^{T} \gamma^{\tau-t} \Theta_\tau^C r_\tau(s_\tau, \mathbf{a}_\tau)$. The following theorem establishes convergence to the optimal multi-timescale value function via *policy iteration in the space of K-periodic non-stationary policies* assuming agent cooperation and full observability.

Assumption 1: The agents are cooperative and have full observability of the environment.

Theorem 1: Under assumption 1, for any $\boldsymbol{\pi}^0 \in \Pi$ induced by K-periodic non-stationary policy $\bar{\boldsymbol{\pi}}^0 \in \bar{\Pi}_K$, $K = \text{LCM}(\tilde{K}, C)$, the sequence of value functions $Q^{\boldsymbol{\pi}^n}$ and improved policies $\boldsymbol{\pi}^{n+1}$ due to policy iteration converges to the optimal multi-timescale value function and optimal multi-timescale policy $\boldsymbol{\pi}^*$, i.e., $Q^{\boldsymbol{\pi}^*}(s,a) = \lim_{n\to\infty} Q^{\boldsymbol{\pi}^n}(s,a) \geq Q^{\boldsymbol{\pi}}(s,a)$. Moreover, the optimal multi-timescale value function always exists and is induced by a K-periodic non-stationary policy.

Lemma 1: Under assumption 1, MT-DEC-POMDP reduces to a multi-timescale multi-agent MDP [16]. Furthermore, a multi-timescale multi-agent MDP is equivalent to an action-persistent factored action (FA) MDP [15].

Proof A multi-agent MDP can be described by the tuple $\{S, N, \mathbf{A}, p, r\}$ with elements defined as in Sec. II [16]. A multi-timescale multi-agent MDP is defined similarly to MT-DEC-POMDP (Eq. (2)), i.e., by expanding a multi-agent MDP with k and C. Assuming agent cooperation, MT-DEC-POMDP has a single reward shared by all agents. Assuming full observability, the observation function $P(s,i)$ in a MT-DEC-POMDP is the identity mapping. Therefore, each agent's policy in MT-DEC-POMDP becomes $\pi^i(a^i|s)$, which completes the reduction. A factored action (FA) MDP with a fully factorized policy over N-dimensional actions $\pi(a^1, \ldots, a^N|s) = \prod_{n=1}^{N} \pi^n(a^n|s)$ is a single agent MDP that can be thought of as equivalently having N agents, i.e., a multi-agent MDP. A k-persistent FA MDP assumes that action a^i is decided every k_i steps and repeated otherwise [15], and can similarly be extended to periodic environments with period C. By modifying the persistence property to allow action a^i to be repeated k_i times *or* for a null action $a^i_{\texttt{null}}$ to be subsequently taken $k_i - 1$ times, the k-persistent FA MDP is equivalent to the multi-timescale multi-agent MDP setting, completing the proof. ■

Given Lemma 1, our proof for Theorem 1 follows the same proof technique used to prove Theorem 3 in Lee et al. [15]. Differently, our proof handles C-periodic environments. To that end, we define here the one-step multi-timescale Bellman *optimality* operator $\bar{\mathcal{T}}_t^*$ induced by $\bar{\boldsymbol{\pi}}$ for $t \in \{0, \ldots, K-1\}$:

$$(\bar{\mathcal{T}}_t^* Q)(s,\mathbf{a}) := \tag{10}$$
$$\Theta_t^C(r_t(s_t,\mathbf{a}_t)) + \gamma \mathbb{E}_{s_{t+1}\sim\Theta_t^C(p_t)}\big[\max_{\mathbf{a}_{t+1}} Q(s_{t+1}, \Gamma_t^k(\mathbf{a}_{t+1})\big].$$

Notice that $\bar{\mathcal{T}}_t^*$ is K-periodic due to the $\tilde{K}$-periodic action projection Γ_t^k and the C-periodic projection operator Θ_t^C. Thus, $\bar{\mathcal{T}}_t^* Q = \bar{\mathcal{T}}_{t+K}^* Q$ for any t and Q. Now, we define the K-step multi-timescale Bellman optimality operator $\bar{H}_t^*$ by composing one-step Bellman optimality operators as follows:

$$\begin{aligned} (\bar{H}_0^* Q)(s,a) &:= (\bar{\mathcal{T}}_0^* \bar{\mathcal{T}}_1^* \cdots \bar{\mathcal{T}}_{K-2}^* \bar{\mathcal{T}}_{K-1}^* Q)(s,a) \\ (\bar{H}_1^* Q)(s,a) &:= (\bar{\mathcal{T}}_1^* \bar{\mathcal{T}}_2^* \cdots \bar{\mathcal{T}}_{K-1}^* \bar{\mathcal{T}}_K^* Q)(s,a) \\ &\vdots \\ (\bar{H}_{K-1}^* Q)(s,a) &:= (\bar{\mathcal{T}}_{K-1}^* \bar{\mathcal{T}}_K^* \cdots \bar{\mathcal{T}}_{K-3}^* \bar{\mathcal{T}}_{K-2}^* Q)(s,a). \end{aligned} \tag{11}$$

The next lemma establishes that K-step multi-timescale Bellman optimality operators are a contraction mapping.

Lemma 2: For all $t \in \{0, \ldots, K-1\}$, the K-step multi-timescale Bellman optimality operator $\bar{H}_t^*$ is a γ^K-contraction with respect to infinity norm with $\bar{H}_t^* Q_t^* = Q_t^*$ as the unique fixed point solution. That is, for any Q_t^0, define

$Q_t^{n+1} = \bar{H}_t^* Q_t^n$. Then, the sequence Q_t^n converges to the t^{th} multi-timescale optimal value function as $n \to \infty$.

Proof Without loss of generality, it is sufficient to prove the case $t = 0$.

For any Q_1, Q_2 and $s_0 \in \mathcal{S}, \mathbf{a}_0 \in \mathbf{A}$,

$$
\begin{aligned}
&|(\bar{H}_0^* Q_1)(s_0, \mathbf{a}_0) - (\bar{H}_0^* Q_2(s_0, \mathbf{a}_0)| \\
&= |(\bar{\mathcal{T}}_0^* \cdots \bar{\mathcal{T}}_{K-1}^* Q_1)(s_0, \mathbf{a}_0) - (\bar{\mathcal{T}}_0^* \cdots \bar{\mathcal{T}}_{K-1}^* Q_2)(s_0, \mathbf{a}_0)| \\
&= \Bigg| \mathbb{E}_{s_1 \sim \Theta_0^C(p_0(s_0, \mathbf{a}_0))} \Big[\Theta_0^C(r_0(s_0, \mathbf{a}_0)) \\
&\quad + \gamma \max_{\mathbf{a}_1} (\bar{\mathcal{T}}_1^* \cdots \bar{\mathcal{T}}_{K-1}^* Q_1)(s_1, \Gamma_{1,\mathbf{a}_0}^k(\mathbf{a}_1)) \Big] \\
&\quad - \mathbb{E}_{s_1 \sim \Theta_0^C(p_0(s_0, \mathbf{a}_0))} \Big[\Theta_0^C(r_0(s_0, \mathbf{a}_0)) \\
&\quad + \gamma \max_{\mathbf{a}_1} (\bar{\mathcal{T}}_1^* \cdots \bar{\mathcal{T}}_{K-1}^* Q_2)(s_1, \Gamma_{1,\mathbf{a}_0}^k(\mathbf{a}_1)) \Big] \Bigg| \\
&= \gamma \Bigg| \mathbb{E}_{\Theta_0^C(p_0)} \Big[\max_{\mathbf{a}_1} (\bar{\mathcal{T}}_1^* \cdots \bar{\mathcal{T}}_{K-1}^* Q_1)(s_1, \Gamma_{1,\mathbf{a}_0}^k(\mathbf{a}_1)) \\
&\quad - \max_{\mathbf{a}_1} (\bar{\mathcal{T}}_1^* \cdots \bar{\mathcal{T}}_{K-1}^* Q_2)(s_1, \Gamma_{1,\mathbf{a}_0}^k(\mathbf{a}_1)) \Big] \Bigg| \\
&\leq \gamma \Bigg| \mathbb{E}_{\Theta_0^C(p_0)} \Big[(\bar{\mathcal{T}}_1^* \cdots \bar{\mathcal{T}}_{K-1}^* Q_1)(s_1, \mathbf{a}_1^*) \\
&\quad - (\bar{\mathcal{T}}_1^* \cdots \bar{\mathcal{T}}_{K-1}^* Q_2)(s_1, \mathbf{a}_1^*) \Big] \Bigg| \\
&\quad \text{where } \mathbf{a}_1^* = \arg\max_{\mathbf{a}} \Big[(\bar{\mathcal{T}}_1^* \cdots \bar{\mathcal{T}}_{K-1}^* Q_1)(s_1, \Gamma_{1,\mathbf{a}_0}^k(\mathbf{a}_1)) \\
&\quad - (\bar{\mathcal{T}}_1^* \cdots \bar{\mathcal{T}}_{K-1}^* Q_2)(s_1, \Gamma_{1,\mathbf{a}_0}^k(\mathbf{a}_1)) \Big] \\
&\leq \gamma \max_{s,\mathbf{a}} \Big| (\bar{\mathcal{T}}_1^* \cdots \bar{\mathcal{T}}_{K-1}^* Q_1)(s, \mathbf{a}) \\
&\quad - (\bar{\mathcal{T}}_1^* \cdots \bar{\mathcal{T}}_{K-1}^* Q_2)(s, \mathbf{a}) \Big|.
\end{aligned}
$$

We can continue to expand the inequality in a similar manner:

$$
\begin{aligned}
&\forall s_0, \mathbf{a}_0, \\
&\quad |(\bar{H}_0^* Q_1)(s_0, \mathbf{a}_0) - (\bar{H}_0^* Q_2)(s_0, \mathbf{a}_0)| \\
&\leq \gamma \max_{s,\mathbf{a}} \big| (\bar{\mathcal{T}}_1^* \cdots \bar{\mathcal{T}}_{K-1}^* Q_1)(s, \mathbf{a}) - (\bar{\mathcal{T}}_1^* \cdots \bar{\mathcal{T}}_{K-1}^* Q_2)(s, \mathbf{a}) \big| \\
&\leq \gamma^2 \max_{s,\mathbf{a}} \big| (\bar{\mathcal{T}}_2^* \cdots \bar{\mathcal{T}}_{K-1}^* Q_1)(s, \mathbf{a}) - (\bar{\mathcal{T}}_2^* \cdots \bar{\mathcal{T}}_{K-1}^* Q_2)(s, \mathbf{a}) \big| \\
&\quad \vdots \\
&\leq \gamma^K \max_{s,\mathbf{a}} \big| Q_1(s, \mathbf{a}) - Q_2(s, \mathbf{a}) \big|,
\end{aligned}
$$

which implies $||\bar{H}_0^* Q_1 - \bar{H}_0^* Q_2||_\infty \leq \gamma^K ||Q_1 - Q_2||_\infty$. Therefore $\bar{H}_t^*$ is a γ^K-contraction with respect to infinity norm, and by the Banach fixed-point theorem, $\bar{H}_t^* Q_t^* = Q_t^*$ is the unique fixed point solution for all t. ■

It follows that the fixed points of $\bar{H}_0^*, \ldots, \bar{H}_{K-1}^*$ together make up the optimal multi-timescale value function, which is represented by the K values $Q_0^{\pi^*}, \ldots, Q_{K-1}^{\pi^*}$. Next, we show that these fixed points have the largest value compared to any other multi-timescale value function for any history-dependent policy $\bar{\pi} \in \bar{\Pi}$.

Lemma 3: Let $\bar{H}_{t \bmod K}^* = \bar{\mathcal{T}}_{t \bmod K}^* \cdots \bar{\mathcal{T}}_{(t+K-1) \bmod K}^*$ be the K-step multi-timescale Bellman optimality operator and $Q_{t \bmod K}^{\pi^*}$ be its fixed point. Then, for any history-dependent policy $\bar{\pi} \in \bar{\Pi}$, $Q_{t \bmod K}^{\pi^*}(s, \mathbf{a}) \geq Q_t^{\bar{\pi}}(s, \mathbf{a})$.

Proof For any $\bar{\pi} \in \bar{\Pi}$, $t, s, \mathbf{a}$ and Q, the following inequality holds:

$$
\begin{aligned}
&(\bar{\mathcal{T}}_t^{\bar{\pi}} Q)(s_t, \mathbf{a}_t) \\
&:= \Theta_t^C(r_t(s_t, \mathbf{a}_t)) \\
&\quad + \gamma \mathbb{E}_{s_{t+1} \sim \Theta_t^C(p_t), \mathbf{a}_{t+1} = \bar{\pi}} \big[Q(s_{t+1}, \Gamma_{t+1,\mathbf{a}_t}^k(\mathbf{a}_{t+1})) \big] \\
&\leq \Theta_t^C(r_t(s_t, \mathbf{a}_t)) \\
&\quad + \gamma \max_{\mathbf{a}_{t+1}} \mathbb{E}_{s_{t+1} \sim \Theta_t^C(p_t)} \big[Q(s_{t+1}, \Gamma_{t+1}^k(\mathbf{a}_{t+1})) \big] \\
&= (\bar{\mathcal{T}}_{t \bmod K}^* Q)(s_t, \mathbf{a}_t).
\end{aligned}
$$

This implies

$$
\begin{aligned}
&(\bar{\mathcal{T}}_t^{\bar{\pi}} \bar{\mathcal{T}}_{t+1}^{\bar{\pi}} \cdots \bar{\mathcal{T}}_{t+K-1}^{\bar{\pi}} Q)(s, \mathbf{a}) \\
&\leq (\bar{\mathcal{T}}_{t \bmod K}^* \bar{\mathcal{T}}_{(t+1) \bmod K}^* \cdots \bar{\mathcal{T}}_{(t+K-1) \bmod K}^* Q)(s, \mathbf{a}) \\
&= (\bar{H}_{t \bmod K}^* Q)(s, \mathbf{a}).
\end{aligned}
$$

Therefore,

$$
\begin{aligned}
&Q_t^{\bar{\pi}}(s, \mathbf{a}) \\
&= \lim_{n \to \infty} (\bar{\mathcal{T}}_t^{\bar{\pi}} \bar{\mathcal{T}}_{t+1}^{\bar{\pi}} \cdots \bar{\mathcal{T}}_{t+Kn-1}^{\bar{\pi}} Q)(s, \mathbf{a}) \\
&\leq \lim_{n \to \infty} ((\bar{H}_{t \bmod K}^*)^n Q)(s, \mathbf{a}) = Q_t^{\pi^*}(s, \mathbf{a})
\end{aligned}
$$

holds, which concludes the proof. ■

Finally, to prove the main claim in Theorem 1, by Lemma 3 it is sufficient to show $\lim_{n \to \infty} Q_t^{\pi^n} = Q_t^{\pi^*}$ for all $t \in \{0, \ldots, K-1\}$. To prove this, first, we must establish monotonic improvement of multi-timescale policies induced by $\bar{\pi}^n$ during policy iteration, i.e., $Q_t^{\pi^{n+1}}(s, \mathbf{a}) \geq Q_t^{\pi^n}(s, \mathbf{a})$ always holds for all $t, s, \mathbf{a}, n$. This result follows by proving a similar result to Theorem 2 in Lee et al. [15] in the same manner as the proofs for Lemmas 2 and 3, i.e., by projecting the reward and transition functions with Θ_t^C. We refer to [15] for the details. When the policy is no longer improving, i.e., $\pi^{n+1} = \pi^n$ and $Q_t^{\pi^{n+1}} = Q_t^{\pi^n}$, then by definition, $Q_t^{\pi^n}$ satisfies the multi-timescale Bellman optimality equation. By Lemma 2, the multi-timescale Bellman optimality equation has a unique solution Q_t^*, implying $Q_t^* = Q_t^{\pi^n}$. Thus, π^n is the optimal multi-timescale policy. Moreover, at every step of policy iteration, we have a policy $\bar{\pi}^n$ which is in $\bar{\Pi}$ and which induces a multi-timescale policy π^n that is guaranteed to eventually converge to the optimal multi-timescale policy. Therefore, the optimal multi-timescale policy always exists.

The known inequality $Q_{\text{Dec-POMDP}}^* \leq Q_{\text{MDP}}^*$ establishes that the optimal value function of a multi-timescale multi-agent MDP is an upper bound of the optimal value function for `MT-DEC-POMDP`. This implies that the optimal value

function obtained by policy iteration under Assumption 1 may overestimate the true optimal value function. See Theorem 5.1 in Oliehoek et al. [8] for proof details.

IV. Phasic Policy Gradient Method

The theory from the previous section suggests encoding K-periodicity into learning algorithms for `MT-DEC-POMDP` to encourage learning the optimal policy when k and C are known. Let the integer $\triangle_t \in \{0, \dots, K-1\}$ indicate the current *phase*, i.e., $t \bmod K$. A straightforward approach is to encode this phase as a one-hot vector of size K, $\texttt{O.H.}(\triangle)$, which can be concatenated to each agent's observation $[o_t^i; \texttt{O.H.}(\triangle_t)]$. However, when the mapping between the phase-augmented observation and the optimal action is complex, encoding the phase as a one-hot vector may not be sufficient to provide a good inductive bias for K-periodicity.

Alternatively, we can parameterize each agent with phase-functioned neural networks [5] (PFNNs). PFNNs are spline-based neural architectures whose weights smoothly vary as a function of the current phase. This provides an inductive bias of using similar weights for adjacent phases and reusing the same network weights at time steps separated by a specified period. **Favorably, the number of parameters in PFNNs scales proportionally with the number of spline control points (a constant) and *not* with the period** K. The use of PFNNs in RL is under-explored, with only one known previous use for training single agents in cyclic environments [17]. Each layer l of a PFNN has a weight matrix α computed by a *phase function* $\alpha_l = \Theta(\beta_l; 2\pi\triangle_t/K)$ conditioned on learnable weight matrices β_l and phase $2\pi\triangle_t/K \in [0, 2\pi]$. Following [5], we use a Catmull-Rom spline for Θ, which is a cubic spline with **four** learnable spline control points $\beta_l = [\beta_l^0, \beta_l^1, \beta_l^2, \beta_l^3]$. The weight for layer l is

$$\begin{aligned}\alpha_l = \beta_l^{x_1} &+ w(\frac{1}{2}\beta_l^{x_2} - \frac{1}{2}\beta_l^{x_0}) \\ &+ w^2(\beta_l^{x_0} - \frac{5}{2}\beta_l^{x_1} + 2\beta_l^{x_2} - \frac{1}{2}\beta_l^{x_3}) \\ &+ w^3(\frac{3}{2}\beta_l^{x_1} - \frac{3}{2}\beta_l^{x_2} + \frac{1}{2}\beta_l^{x_3} - \frac{1}{2}\beta_l^{x_0}),\end{aligned}$$

where $w = 4\triangle_t/K \pmod 1$ and $x_n = \lfloor 4\triangle_t/K \rfloor + n - 1 \pmod 4$. The bias for layer l is computed in a similar fashion. The start and end control points for each layer are the same, making each PFNN layer cyclic. In this work, we adapt the actor-critic policy gradients method COMA [4] by using PFNNs with period $K = \text{LCM}(\tilde{K}, C)$ for the actor and critic networks.

V. Experiments

A. The Move Box Problem

Setup: We adapted a gridworld environment called Move Box [18] to create a toy multi-timescale environment with a time-dependent optimal policy. That is, the two agents need to coordinate their actions to push a green box to a goal location ("G") within a maximum of 20 steps. In the **easy** version, one agent (red) is a "fast" agent that uses $k_1 = 1$ and one agent (blue) is a "slow" agent that acts every two steps $k_2 = 2$. The period provided to time-aware agents is therefore $K := \texttt{LCM}(1, 2, C = 1) = 2$. For the **hard** version of this task, the fast agent uses $k_1 = 2$ and slow agent uses $k_2 = 3$, thus $K := \texttt{LCM}(2, 3, C = 1) = 6$. Each agent receives a 4D partial observation consisting of its own position and the position of the box. To avoid the need for a complex exploration strategy, we restrict the action space to 3 discrete actions: move up, move down, or null (do nothing). To push the green box up or down, the agents have to be on either side of the box and move in the same direction at the same time. Agents receive a reward of +1 whenever the box moves towards the goal and a reward of +20 once the box reaches the goal. The easy version can be solved with 7 actions while the hard version requires 19 actions; there is a significant increase in exploration difficulty between the easy and hard versions. To implement multiple timescales, the only legal action available to the slow agent between steps is the null action. *Move Box is designed so that a time-unaware fast agent suffers from observation aliasing* (Sec. II). A time-unaware fast agent's limited information means it cannot precisely determine whether to move up or do nothing to synchronize its actions with the slow agent.

TABLE I

Move Box results. Success (%) is the fraction out of 8 random seeds that the agents learned to move the box to the goal. The optimal return is 23.0.

	Move Box [Easy]		Move Box [Hard]	
COMA	Success (%)	Avg. return	Success (%)	Avg. return
Basic	0	$1.0_{\pm 0.0}$	0	$1.0_{\pm 0.0}$
Recurrent	75	$17.5_{\pm 10.2}$	0	$3.75_{\pm 7.8}$
O.H. phase-aware	**100**	**23.0**$_{\pm 0.0}$	0	$3.45_{\pm 7.9}$
Phasic (4)	**100**	**23.0**$_{\pm 0.0}$	**50**	$12.0_{\pm 11.8}$
Phasic (15)	**100**	**23.0**$_{\pm 0.0}$	**50**	$12.0_{\pm 11.8}$

Agents: We train four COMA-based agents:

- `Basic COMA`, a time-unaware agent that uses a feed-forward neural network for the actor and critic networks.
- `Recurrent COMA`, a time-aware agent that uses an RNN for the actor and critic networks to condition on the full history up to the current time step [10].
- `One-Hot (O.H.) phase-aware COMA`, an agent whose observations are augmented with a one-hot encoding of the current phase $\triangle_t$.
- `Phasic COMA`, an agent whose actor and critic networks are PFNNs with weights indexed by $2\pi\triangle_t/K$.

We also run a variant of PFNN with period 4 instead of K, `Phasic COMA (4)`, to explore performance sensitivity to this hyperparameter. In the easy environment $4 > K$ and in the hard environment $4 < K$. All actor and critic networks share parameters. We take the standard approach of providing a one-hot agent ID as an auxiliary input to distinguish between agents.

Results: Table I shows quantitative results and Fig. 3 compares test return as a function of environment steps. Both `O.H. phase-aware COMA` and `Phasic COMA` learn to reliably solve the easy Move Box environment across all

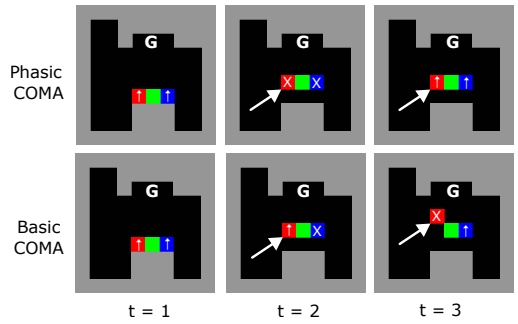

Fig. 2. **Move Box qualitative analysis**. In the easy setting, the red agent acts every step ($k_1 = 1$) and the blue agent acts every two steps ($k_2 = 2$). The joint action is shown in white on each agent. The time-unaware `Basic COMA` red agent tries to move the box up at $t = 2$, which causes it to drop the box at $t = 3$. The time-aware `Phasic COMA` red agent learns to take a null action at $t = 2$. The hard setting requires more sophisticated coordination between agents. Best viewed in color.

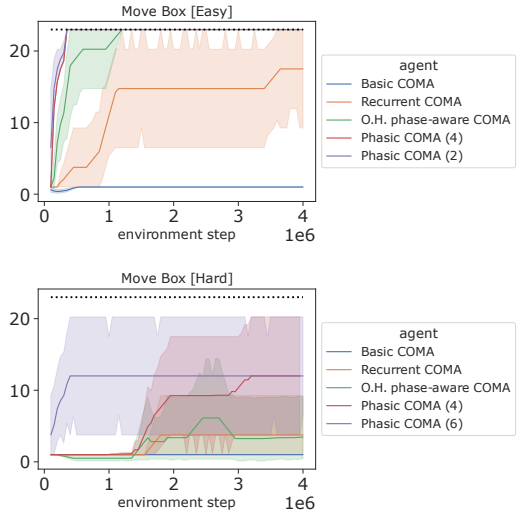

Fig. 3. **Move Box return vs. steps**. Mean return over 8 random seeds of the learned greedy policy at various steps during training (shaded region is the 95% confidence interval (CI)). Best possible return shown by black dotted line. The PFNN-based agent `Phasic COMA` with correct periods $K = 2, 6$ achieves the highest reward in the fewest environment steps.

random seeds, with `Phasic COMA` demonstrating a small advantage in terms of efficiency. `Recurrent COMA` needs more steps to achieve a good test return yet ultimately performs less reliably. `Basic COMA` fails on this environment as expected due to observation aliasing (Fig. 2). In the hard version (Fig. 3), only `Phasic COMA` learns to solve the environment on just 50% of the training runs. The PFNN-based actor and critic is robust to a slightly smaller or larger period than K, although it appears to require more environment steps.

B. Building Energy Management

Setup: In this environment, agents attempt to coordinate the control of HVAC and energy storage (ES) for a five-zone small office building. See Biagioni et al. [19] for details about the reduced order model used to simulate the building. There are 7 agents: an HVAC agent per zone able to change the

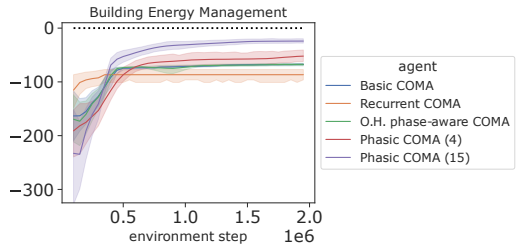

Fig. 4. **BEM results.** Mean return over 8 random seeds of the learned greedy policy at various steps during training (shaded region is the 95% CI). Best possible is the black dotted line. The `Phasic COMA` agent with correct period $K = 15$ outperforms non-phasic variants by a wide margin.

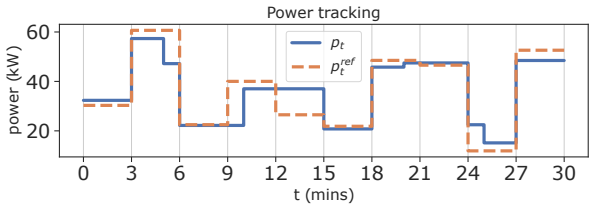

Fig. 5. **BEM qualitative results**. Total power vs. power reference signal from one of the `Phasic COMA` training runs.

mass flow rate (kg/s) every 5 mins, an HVAC chiller agent that can change the discharge air temperature (°C) every 5 mins, and an ES agent that can change its charging or discharging power (kW) every 1 minute. The goal is for the agents to coordinate their total power consumption to track a reference power signal that changes every 3 mins while minimizing discomfort to building occupants. The control horizon is set to 30 mins (dt = 1min). The global reward function at time step t is defined as

$$r_t = -\sum_{\text{zone}_i} \left((T_t^i - \overline{T})^+ + (\underline{T} - T_t^i)^+ \right)^2 - \alpha (p_t - p_t^{\text{ref}})^2,$$

where $\alpha = 0.01$, $\overline{T} := 26$ °C and $\underline{T} := 24$ °C is the thermal comfort band, T_t^i is zone i's temperature, and p_t is the total power. To implement multiple timescales, agents repeat their previous action between steps. The period used for learning periodic non-stationary policies is $K := \texttt{LCM}(1, 5, C = 3) = 15$, where C encodes the cyclic power reference signal.

Results: Out of all agents, only `Phasic COMA` is able to reliably learn a near-optimal joint policy (Fig. 4). The variant with arbitrary PFNN period $4 \ll K$ is the second best agent. We visualize the control actions selected by the joint policy from one of the `Phasic COMA` runs in Fig. 6. The slow HVAC agents have successfully learned to coordinate with the faster ES agent to track the reference signal (Fig. 5) without violating the thermal comfort band; for example, by increasing their power consumption between 10-15 mins while the ES agent is already maximally discharging.

VI. CONCLUSIONS

In this work, we proposed a multi-timescale MARL framework for learning policies that can represent complex time-dependent behaviors. We introduced the multi-timescale non-

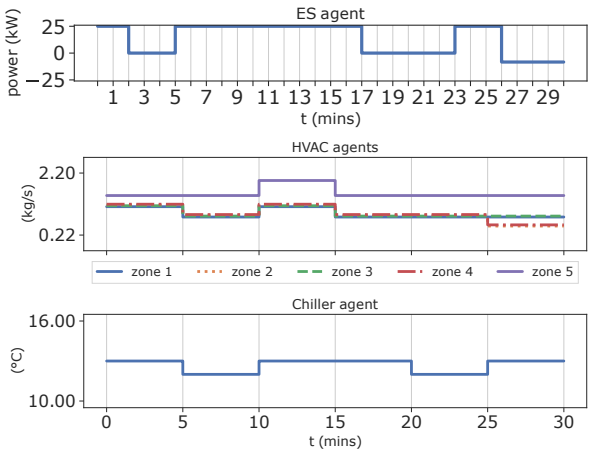

Fig. 6. **BEM qualitative results**. Agent actions from one of the `Phasic COMA` training runs.

stationary joint policy as the policy induced by a K-periodic non-stationary joint policy, where the period K is given by knowledge about agent timescales and cyclic environment components, both of which are typically known *a priori*. We use phase-functioned neural networks to introduce an inductive bias for learning a periodic non-stationary joint policy. Our results on grid world and building energy management environments establish the effectiveness of our framework, suggesting that follow-up work could explore using it to solve more advanced power systems problems.

REFERENCES

[1] G. Papoudakis, F. Christianos, A. Rahman, and S. V. Albrecht, "Dealing with non-stationarity in multi-agent deep reinforcement learning," *arXiv preprint arXiv:1906.04737*, 2019.

[2] P. Hernandez-Leal, M. Kaisers, T. Baarslag, and E. M. de Cote, "A survey of learning in multiagent environments: Dealing with non-stationarity," *ArXiv preprint*, vol. abs/1707.09183, 2017.

[3] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, 2017, pp. 6379–6390.

[4] J. N. Foerster, G. Farquhar, T. Afouras, N. Nardelli, and S. Whiteson, "Counterfactual multi-agent policy gradients," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, 2018, pp. 2974–2982.

[5] D. Holden, T. Komura, and J. Saito, "Phase-functioned neural networks for character control," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 1–13, 2017.

[6] D. S. Bernstein, R. Givan, N. Immerman, and S. Zilberstein, "The complexity of decentralized control of markov decision processes," *Mathematics of operations research*, vol. 27, no. 4, pp. 819–840, 2002.

[7] J. Wu, K. Li, and Q.-S. Jia, "Decentralized multi-agent reinforcement learning with multi-time scale of decision epochs," in *2020 59th IEEE Conference on Decision and Control (CDC)*, 2020, pp. 578–584, ISSN: 2576-2370.

[8] F. A. Oliehoek, M. T. J. Spaan, and N. Vlassis, "Optimal and approximate q-value functions for decentralized POMDPs," *JAIR*, vol. 32, pp. 289–353, 2008.

[9] F. Pardo, A. Tavakoli, V. Levdik, and P. Kormushev, "Time limits in reinforcement learning," in *International Conference on Machine Learning*. PMLR, 2018, pp. 4045–4054.

[10] R. E. Wang, M. Everett, and J. P. How, "R-maddpg for partially observable environments and limited communication," *arXiv preprint arXiv:2002.06684*, 2020.

[11] Z. Wu, Y. Li, W. Gu, Z. Dong, J. Zhao, W. Liu, X.-P. Zhang, P. Liu, and Q. Sun, "Multi-timescale voltage control for distribution system based on multi-agent deep reinforcement learning," *International Journal of Electrical Power & Energy Systems*, vol. 147, p. 108830, 2023.

[12] T. Ochoa, E. Gil, A. Angulo, and C. Valle, "Multi-agent deep reinforcement learning for efficient multi-timescale bidding of a hybrid power plant in day-ahead and real-time markets," *Applied Energy*, vol. 317, p. 119067, 2022.

[13] Hyeong Soo Chang, P. Fard, S. Marcus, and M. Shayman, "Multi-time scale markov decision processes," *IEEE Trans. Automat. Contr.*, vol. 48, no. 6, pp. 976–987, 2003.

[14] M. He, S. Murugesan, and J. Zhang, "A markov decision process approach to multi-timescale scheduling and pricing in smart grids with integrated wind generation," in *2011 4th IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, 2011, pp. 125–128.

[15] J. Lee, B.-J. Lee, and K.-E. Kim, "Reinforcement learning for control with multiple frequencies," *Advances in Neural Information Processing Systems*, vol. 33, pp. 3254–3264, 2020.

[16] C. Boutilier, "Planning, learning and coordination in multiagent decision processes," in *TARK*, vol. 96. Citeseer, 1996, pp. 195–210.

[17] A. Sharma and K. M. Kitani, "Phase-parametric policies for reinforcement learning in cyclic environments," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, 2018, pp. 6540–6547.

[18] S. Jiang and C. Amato, "Multi-agent reinforcement learning with directed exploration and selective memory reuse," in *Proceedings of the 36th Annual ACM Symposium on Applied Computing*, 2021, pp. 777–784.

[19] D. Biagioni, X. Zhang, C. Adcock, M. Sinner, P. Graf, and J. King, "From model-based to model-free: Learning building control for demand response," *arXiv preprint arXiv:2210.10203*, 2022.