Controller Synthesis for Reward Collecting Markov Processes in Continuous Space

Sadegh Esmaeil Zadeh Soudjani
Max Planck Institute for Software Systems
Kaiserslautern, Germany
sadegh@mpi-sws.org

Rupak Majumdar
Max Planck Institute for Software Systems
Kaiserslautern, Germany
rupak@mpi-sws.org

ABSTRACT

We propose and analyze a generic mathematical model for optimizing rewards in continuous-space, dynamic environments, called Reward Collecting Markov Processes. Our model is motivated by request-serving applications in robotics, where the objective is to control a dynamical system to respond to stochastically generated environment requests, while minimizing wait times. Our model departs from usual discounted reward Markov decision processes in that the reward function is not determined by the current state and action. Instead, a background process generates rewards whose values depend on the number of steps between generation and collection. For example, a reward is declared whenever there is a new request for a robot and the robot gets higher reward the sooner it is able to serve the request. A policy in this setting is a sequence of control actions which determines a (random) trajectory over the continuous state space. The reward achieved by the trajectory is the cumulative sum of all rewards obtained along the way in the finite horizon case and the long run average of all rewards in the infinite horizon case.

We study both the finite horizon and infinite horizon problems for maximizing the expected (respectively, the long run average expected) collected reward. We characterize these problems as solutions to dynamic programs over an augmented hybrid space, which gives history-dependent optimal policies. Second, we provide a computational method for these problems which abstracts the continuous-space problem into a discrete-space collecting reward Markov decision process. Under assumptions of Lipschitz continuity of the Markov process and uniform bounds on the discounting, we show that we can bound the error in computing optimal solutions on the finite-state approximation. Finally, we provide a fixed point characterization of the optimal expected collected reward in the infinite case, and show how the fixed point can be obtained by value iteration.

HSCC'17, April 18 - 20, 2017, Pittsburgh, PA, USA

ISBN 978-1-4503-4590-3/17/04…$15.00

DOI: http://dx.doi.org/10.1145/3049797.3049827

Keywords Reward collecting Markov processes; formal controller synthesis; continuous-space stochastic systems

1. INTRODUCTION

Consider a mobile robot in an environment. The robot receives requests from different users and must serve these requests by traveling to the location of the request. The aim of the robot is to respond to each request as soon as possible. How should the robot plan its actions?

This scenario generalizes many problems studied in the robotics, control, and combinatorial optimization literatures. In the most general form, these problems incorporate: (a) control of continuous-state dynamical systems w.r.t. temporal requirements (the robot must navigate to different locations while maintaining safety), (b) dynamic requests from a stochastic environment (user requests can be modeled as a stochastic process), (c) cumulative reward collection (the robot gets a reward on serving a request, depending on the wait time, and the overall reward is cumulative).

We propose and analyze a generic mathematical model for optimizing rewards in continuous-space, dynamic environments, called Reward Collecting Controlled Markov Processes (RCCMP). Our model is motivated by the above request-serving applications, where the objective is to control a dynamical system to respond to environment requests that are generated stochastically, while minimizing wait times. An RCCMP is defined using (a) a controlled Markov process, (b) a reward process, and (c) a reward functional. The controlled Markov process is defined over a continuous state space in discrete time. That is, the states and inputs form Borel spaces, and the transitions are defined by a conditional stochastic kernel which associates with each state and control input a probability measure over the next states. The reward process is defined over a given finite partition of the state space, and assigns a random reward to the partition at each time step. Finally, we use discounted reward as the classic way to ensure low latency for each request: if the Markov process visits a particular region consecutively at two time instances $t = k$ and $t = k'$, then the cumulative reward is collected, which is the sum of all the rewards associated with the requests generated at this region between times k and k' each discounted depending on the time the request is generated and the time it is served (i.e., k').

A *policy* ascribes a control action to the controlled Markov process at each time step. It determines a random trajectory. The reward achieved by the trajectory is the cumulative sum of all rewards obtained along the trajectory. We study both the finite horizon and infinite horizon problems

for maximizing the expected cumulative reward for finite horizon and the long run average expected cumulative reward for the infinite horizon. We also study how to design a policy that maximizes the expected rewards.

Our first result characterizes these problems as solutions to dynamic programs. Second, we provide a computational method which abstracts the continuous-space problem into a discrete-space cumulative reward Markov decision process. Under assumptions of Lipschitz continuity of the Markov process and uniform bounds on the discounting, we show that we can bound the error in computing optimal solutions on the finite-state approximation. Finally, we provide a fixed point characterization of the optimal expected cumulative reward in the finite case, and show how the fixed point can be obtained by value iteration. We illustrate our results with a simple request-serving robot example.

Related Work. A number of models studied in the optimization and control literature are close to ours. For example, (dynamic) traveling salesman problems, and their analogues such as the stochastic orienteering problem or the vehicle routing problem, study strategies to optimize path costs in a finite graph. In contrast, our model is defined over a general continuous-state stochastic process and defines rewards dynamically and cumulatively.

A second related model is Markov reward models (MRMs) [8] in which the model is deterministic (i.e., with no input) and the reward is a function of state $r(x_t)$. Another related model is Markov decision process (MDPs) over finite or infinite state spaces, in which the rewards are usually defined as fixed functions of the current state and action taken at that state $r(x_t, u_t)$. The reinforcement learning community sometimes works with rewards defined as functions of the tuple (current state, action, next state) $r(x_t, u_t, x_{t+1})$ [15, Chapter 3.6].

Infinite-horizon performance evaluation in MRMs and optimization in MDPs are performed via the following measures: total reward, discounted reward, and average reward. The first measure is just the infinite sum of all rewards associated to the paths of the process, which may not be bounded in general. The other two measures ensures boundedness of the measure by considering respectively the infinite sum of discounted and long-run average rewards. Such problems are throughly studied in [12] for finite and countable space models and in [10] for continuous uncountable space models. The third measure is in fact the long-run average expected value of the rewards, while the paper [3] and related works study a stronger infinite-horizon optimization in MDPs, which is the expected long-run average (the difference is in the position of the expected value operator).

Our model departs from MDPs in that the reward function is not deterministically determined for each state and action once and for all. Instead, a background stochastic process repeatedly generates rewards in the state space, and each generated reward decays over time through a discount factor. The treatment of accumulating rewards through a "double summation" over consecutive visits to a location introduces differences from the MDP model: for an RCCMP, we show that memoryless policies are no longer optimal. The work [13] studies MDPs with functional rewards, in which the reward is also a function of previously collected rewards. However, we cannot define our rewards in their framework: our rewards depend on the time since of the last visit.

Finite-state approximations of continuous-space Markov processes with guarantees on error bounds was studied before [1, 4, 6, 14, 16]. In comparison with [4], the main challenge in our context is that the state space has a countable, unbounded, component tracking the time steps since the last visit to each region. Second, approximations studied in [1, 6, 14] consider finite-horizon temporal specifications with extensions to infinite-horizon ones [16]. These approximations benefit from the fact that the associated value functions are bounded by one uniformly. We study cumulative reward problems over finite and infinite horizons and must modify the approximation construction due to the lack of this bound. Our work is also distinct from the previous related work in that we give such approximation and the error analysis for long-run average criterion.

2. CONTROLLED MARKOV PROCESSES

2.1 Preliminaries

We consider a probability space $(\Omega, \mathcal{F}_\Omega, P_\Omega)$, where Ω is the sample space, $\mathcal{F}_\Omega$ is a sigma-algebra on Ω comprising subsets of Ω as events, and P_Ω is a probability measure than assigns probabilities to events. We assume that random variables introduced in this article are measurable functions of the form $X : (\Omega, \mathcal{F}_\Omega) \to (S_X, \mathcal{F}_X)$. Any random variable X induces a probability measure on its space $(S_X, \mathcal{F}_X)$ as $Prob\{A\} = P_\Omega\{X^{-1}(A)\}$ for any $A \in \mathcal{F}_X$. We often directly discuss the probability measure on $(S_X, \mathcal{F}_X)$ without explicitly mentioning the underlying probability space and the function X itself.

A topological space S is called a Borel space if it is homeomorphic to a Borel subset of a Polish space (i.e., a separable and completely metrizable space). Examples of a Borel space are the Euclidean spaces $\mathbb{R}^n$, its Borel subsets endowed with a subspace topology, as well as hybrid spaces. Any Borel space S is assumed to be endowed with a Borel sigma-algebra, which is denoted by $\mathcal{B}(S)$. We say that a map $f : S \to Y$ is measurable whenever it is Borel measurable.

The following notation is used throughout the paper. We denote the set of nonnegative integers by $\mathbb{N} := \{0, 1, 2, \ldots\}$ and the set of positive integers by $\mathbb{Z}_+ := \{1, 2, 3, \ldots\}$. The bounded set of integers is indicated by $\mathbb{N}[a, b] := \{a, a + 1, \ldots, b\}$ for any $a, b \in \mathbb{N}$, $a \leq b$. For any set A we denote by $A^{\mathbb{N}}$ the Cartesian product of a countable number of copies of A, i.e., $A^{\mathbb{N}} = \prod_{k=0}^{\infty} A$. We denote with $\mathbb{I}(\cdot)$ the indicator function which takes a Boolean-valued expression as an argument and gives 1 if this expression evaluates to true and 0 when it is false.

2.2 Controlled Markov Processes

We adopt the notation from [10] and consider controlled Markov processes (CMP) in discrete time defined over a general state space, characterized by a tuple

$$\mathfrak{S} = (\mathcal{S}, \mathcal{U}, \{\mathcal{U}(s) | s \in \mathcal{S}\}, T_{\mathfrak{s}}),$$

where $\mathcal{S}$ is a Borel space as the state space of the process. We denote by $(\mathcal{S}, \mathcal{B}(\mathcal{S}))$ as the measurable space with $\mathcal{B}(\mathcal{S})$ being the Borel sigma-algebra on the state space. $\mathcal{U}$ is a Borel space as the input space of the process. The set $\{\mathcal{U}(s) | s \in \mathcal{S}\}$ is a family of non-empty measurable subsets of $\mathcal{U}$ with the property that

$$\mathcal{K} := \{(s, u) : s \in \mathcal{S}, u \in \mathcal{U}(s)\}$$

is measurable in $\mathcal{S}\times\mathcal{U}$. Intuitively, $\mathcal{U}(s)$ is the set of inputs that are feasible at state $s\in\mathcal{S}$. $T_{\mathfrak{s}}:\mathcal{B}(\mathcal{S})\times\mathcal{S}\times\mathcal{U}\to[0,1]$, is a conditional stochastic kernel that assigns to any $s\in\mathcal{S}$ and $u\in\mathcal{U}(s)$ a probability measure $T_{\mathfrak{s}}(\cdot|s,u)$ on the measurable space $(\mathcal{S},\mathcal{B}(\mathcal{S}))$ so that for any set $A\in\mathcal{B}(\mathcal{S}), P_{s,u}(A)=\int_A T_s(ds|s,u)$, where $P_{s,u}$ denotes the conditional probability $P(\cdot|s,u)$.

2.3 Semantics

The semantics of a CMP is characterized by its *paths* or executions, which reflect both the history of previous states of the system and of implemented control inputs. Paths are used to measure the performance of the system.

Definition 1. *Given a CMP $\mathfrak{S}$, a* finite path *is a sequence*

$$w_n=(s_0,u_0,\ldots,s_{n-1},u_{n-1},s_n),\quad n\in\mathbb{N},$$

where $s_i\in\mathcal{S}$ are state coordinates and $u_i\in\mathcal{U}(s_i)$ are control input coordinates of the path. The space of all paths of length n is denoted by $\mathsf{PATH}_n:=\mathcal{K}^n\times\mathcal{S}$. Further, we denote projections by $w_n[i]:=s_i$ and $w_n(i):=u_i$. An infinite path *of the CMP $\mathfrak{S}$ is the sequence $w=(s_0,u_0,s_1,u_1,\ldots)$, where $s_i\in\mathcal{S}$ and $u_i\in\mathcal{U}(s_i)$ for all $i\in\mathbb{N}$. As above, let us introduce $w[i]:=s_i$ and $w(i):=u_i$. The space of all infinite paths is denoted by $\mathsf{PATH}_\infty:=\mathcal{K}^\infty$.*

Given an infinite path w or a finite path w_n, we assume below that s_i and u_i are their state and control coordinates respectively, unless otherwise stated. For any infinite path $w\in\mathsf{PATH}_\infty$, its n-prefix (ending in a state) w_n is a finite path of length n, which we also call *n-history*. We are now ready to introduce the notion of control policy.

Definition 2. *A* policy *is a sequence $\rho=(\rho_0,\rho_1,\rho_2,\ldots)$ of universally measurable stochastic kernels ρ_n [2], each defined on the input space $\mathcal{U}$ given PATH_n and such that for all $w_n\in\mathsf{PATH}_n$ with $n\in\mathbb{N}$, $\rho_n(\mathcal{U}(s_n)|w_n)=1$. The set of all policies is denoted by Π.*

Given a policy $\rho\in\Pi$ and a finite path $w_n\in\mathsf{PATH}_n$, the distribution of the next control input u_n given by $\rho_n(\cdot|w_n)$ is supported on $\mathcal{U}(s_n)$. A policy ρ is *deterministic* if all stochastic kernels ρ_i, $i\in\mathbb{N}$, are Dirac delta measures, otherwise it is called *randomized*. Among the class of all possible policies, special interest is shown in the literature towards those with a simple structure in that they depend only on the current state, rather than on the whole history.

Definition 3. *A policy $\rho\in\Pi$ is called a Markov policy if for any $n\in\mathbb{N}$ it holds that $\rho_n(\cdot|w_n)=\rho_n(\cdot|s_n)$, i.e., ρ_n depends on the history w_n only through the current state s_n. The class of all Markov policies is denoted by $\Pi_M\subset\Pi$.*

A more restrictive set of policies, which will be used in Section 5, is the class of *stationary* policies $\Pi_S\subset\Pi_M$, which are Markov, deterministic, and time-independent. Namely, there is a function $d:\mathcal{S}\to\mathcal{U}$ such that at any time epoch $n\in\mathbb{N}$, the input u_n is taken to be $d(s_n)\in\mathcal{U}(s_n)$. We denote stationary policies just by $d\in\Pi_S$.

For a CMP $\mathfrak{S}$, any policy $\rho\in\Pi$ together with an initial probability measure $\alpha:\mathcal{B}(\mathcal{S})\to[0,1]$ of the CMP induce a unique probability measure on the canonical sample space of paths [10] denoted by P^ρ_α with the expectation $\mathbb{E}^\rho_\alpha$. In the case when the initial probability measure is supported on a single point, i.e., $\alpha(s)=1$, we write P^ρ_s and $\mathbb{E}^\rho_s$ in place of P^ρ_α and $\mathbb{E}^\rho_\alpha$, respectively. We denote the set of probability measures on $(\mathcal{S},\mathcal{B}(\mathcal{S}))$ by $\mathfrak{D}$.

Example 1. *Consider a robot moving in a 2-dimensional environment $\mathcal{S}=[0,a]\times[0,b]$, surrounded by walls, according to the dynamics:*

$$s_{t+1}=s_t+\alpha_0 g_{\mathfrak{m}}(u_t)+\eta_t,\quad t\in\mathbb{N},\tag{1}$$

where $\{\eta_t,\,t\in\mathbb{N}\}$ are independent identically-distributed (iid) random variables with η_t having normal distribution $\mathcal{N}(0,\Sigma_{\mathsf{r}})$ and models the uncertainty in the movement of the robot. The input space is $\mathcal{U}=\{\mathsf{left},\mathsf{right},\mathsf{up},\mathsf{down}\}$. The parameter α_0 is the length of the nominal move of the robot and the function $g_{\mathfrak{m}}:\mathcal{U}\to\mathbb{R}^2$ indicates the move direction:

$$g_{\mathfrak{m}}(\mathsf{left})=\begin{bmatrix}-1\\0\end{bmatrix},g_{\mathfrak{m}}(\mathsf{right})=\begin{bmatrix}1\\0\end{bmatrix},g_{\mathfrak{m}}(\mathsf{up})=\begin{bmatrix}0\\1\end{bmatrix},g_{\mathfrak{m}}(\mathsf{down})=\begin{bmatrix}0\\-1\end{bmatrix}.$$

The stochastic kernel of the dynamical system (1) is also normal $T_{\mathfrak{s}}(ds_{t+1}|s_t,u_t)\sim\mathcal{N}(s_t+\alpha_0 g_{\mathfrak{m}}(u_t),\Sigma_{\mathsf{r}})$.

3. PROBLEM DEFINITION

3.1 Cumulative Discounted Rewards

A measurable partition of the state space $\mathcal{S}$ is a finite set $\mathcal{D}:=\{D_1,\ldots,D_m\}$ such that each D_j is a non-empty measurable subset of $\mathcal{S}$, the sets are pairwise disjoint, i.e., $D_i\cap D_j=\ $ for $i\neq j$, and the union of the sets is $\mathcal{S}$, i.e., $\mathcal{S}=\cup_i D_i$. We refer to the subsets D_j as *regions*.

Fix a measurable partition $\mathcal{D}$. We associate the following functions with $\mathcal{D}$. A *reward function* $r:\Omega\times\mathbb{N}\times\mathcal{D}\to\mathbb{R}$ is a stochastic process assigning real random reward $r(\cdot,t,D)$ to any region $D\in\mathcal{D}$ at any time $t\in\mathbb{N}$. The *discounting function* $\gamma:\mathcal{D}\to(0,1)$ associates with each region D a discounting factor $\gamma(D)$, which is a real number in the open interval $(0,1)$.

Each state $s\in\mathcal{S}$ belongs to exactly one $D\in\mathcal{D}$. We define the map $\Xi:\mathcal{S}\to\mathcal{D}$ that maps $s\in\mathcal{S}$ with the (unique) region $D\in\mathcal{D}$ s.t. $s\in D$. We also use $\xi:\mathcal{S}\to\{0,1\}^m$ with $\xi(s)$ being a row vector of dimension m with elements $\mathbb{I}(s\in D_i),\ i\in\mathbb{N}[1,m]$.

For a finite path $w_n=(s_0,u_0,\ldots,s_{n-1},u_{n-1},s_n)$, a region $D\in\mathcal{D}$, and an index $k\leq n$, we define $\mathsf{Last}(w_n,D,k)$ as the last time epoch before k that the path w_n visits region D, defining it to be -1 in case w_n does not visit D before time epoch k:

$$\mathsf{Last}(w_n,D,k):=\max\left\{\{j|j<k,w_n[j]\in D\}\cup\{-1\}\right\}.$$

The finite-horizon *cumulative discounted reward* (CDR) is defined as a map from the set of policies Π and set of measures $\mathfrak{D}$ to $\mathbb{R}$ as follows:

$$\mathsf{CDR}_n(\rho,\alpha)=\mathbb{E}\left[\sum_{k=0}^{n}\sum_{t=\mathsf{Last}(w_n,\Xi(s_k),k)+1}^{k}\gamma(\Xi(s_k))^{k-t}r(\varsigma,t,\Xi(s_k))\right],\tag{2}$$

for any $\rho\in\Pi$ and $\alpha\in\mathfrak{D}$, with s_k being the state visited at time epoch k by w_n. Intuitively, the path w_n visits the region $\Xi(s_k)$ at time epoch k. The inner sum gives the discounted reward accumulated at region $\Xi(s_k)$ since the last visit of the region: the reward $r(\varsigma,t,\Xi(s_k))$ generated at time epoch t is discounted by multiplying it with the factor $\gamma(\Xi(s_k))^{k-t}$, which depends on the difference $k-t$ between the time the reward is generated and the time it is collected.

The expected value in (2) is respect to both the canonical sample space of paths and the underlying probability space of the generated rewards $\varsigma \in \Omega$.

We assume that the random variables $r(\cdot, t, D)$ are stochastically independent of the $\mathfrak{S}$ dynamics and their expected value exists, is non-negative, and is denoted by $\lambda(t, D)$. Due to the additive nature of CDR, we can write

$$\mathsf{CDR}_n(\rho, \alpha) = \mathbb{E}^{\rho}_{\alpha}\left[\sum_{k=0}^{n}\sum_{t=\mathsf{Last}(w_n, \Xi(s_k), k)+1}^{k} \gamma(\Xi(s_k))^{k-t}\lambda(t, \Xi(s_k))\right],$$

for all $\rho \in \Pi$ and $\alpha \in \mathfrak{D}$. We define the infinite-horizon CDR as

$$\mathsf{CDR}_\infty(\rho, \alpha) = \liminf_{n\to\infty} \frac{1}{n+1}\mathsf{CDR}_n(\rho, \alpha), \tag{3}$$

which is the *liminf average* of the finite-horizon CDR. Note that limit average does not necessarily exist, thus we have selected the worst case limiting reward accumulated along the path. Alternatively, one may opt for best case limiting reward, i.e *limsup average*. The analysis and results of this paper are valid for both cases with minor modifications.

Definition 4. *A RCCMP is a pair $(\mathfrak{S}, \mathfrak{R})$ with CMP $\mathfrak{S}$ defined in Section 2.2 and the tuple $\mathfrak{R} := (\mathcal{D}, \lambda, \gamma)$, where $\mathcal{D}$ is a measurable partition of the state space, $\lambda : \mathbb{N}\times\mathcal{D} \to \mathbb{R}_{\geq 0}$ is the expected generated rewards, and $\gamma : \mathcal{D} \to (0, 1)$ is the discounting function.*

Remark 1. *The definition of CDR relies on having geometrically discounting factors for each region, i.e., the sequence*

$$\{1, \gamma(D), \gamma(D)^2, \gamma(D)^3, \ldots\},$$

the larger discount the longer takes to collect the reward (the term $\gamma(D)^{k-t}$ in (2)). While we present our results using this familiar notion of discounting, such geometric discounting factors may not be appropriate for applications in which the required time to serve customers takes multiple time steps. The whole analysis of this paper is valid if the discounting is performed with any other non-negative sequence $\{a_t, t \in \mathbb{N}\}$ with bounded total sum $\sum_{t=0}^{\infty} a_t < \infty$.

Example 1 (continued). Suppose that the 2-dimensional state space is partitioned into two offices and one hallway as depicted in Figure 1. Consider discounting factors $\gamma(D_i) := \gamma_i$ in $(0, 1)$ and expected generated reward $\lambda(t, D_i) := \lambda_i$ for all $i \in \{1, 2, 3\}$ and $t \in \mathbb{N}$, such that $\lambda_2 = 0$ and $\lambda_1, \lambda_3 > 0$. For a policy ρ generating a sample path that visits the following regions consecutively:

$$D_2, D_2, D_3, D_2, D_1, D_2, D_2, \ldots,$$

the expected collected reward is

$$\lambda_2 + \lambda_2 + \lambda_3(1 + \gamma_3 + \gamma_3^2) + \lambda_2(1 + \gamma_2) + \\ + \lambda_1(1 + \gamma_1 + \gamma_1^2 + \gamma_1^3 + \gamma_1^4) + \lambda_2(1 + \gamma_2) + \lambda_2 + \ldots.$$

This sum goes to infinity if the path visits either of the regions D_1, D_3 infinitely often. The long-run average reward for the path that visits the regions $(D_2, D_2, D_3, D_2, D_1)$ periodically is

$$\frac{1}{5}\left[\lambda_2(1 + \gamma_2) + \lambda_2 + \frac{\lambda_3(1-\gamma_3^5)}{1-\gamma_3} + \lambda_2(1 + \gamma_2) + \frac{\lambda_1(1-\gamma_1^5)}{1-\gamma_1}\right].$$

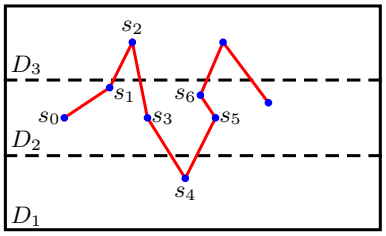

Figure 1: Layout of the 2-dimensional space for the robot's move in Example 1. D_1 and D_3 are offices and D_2 is the hallway. A sample path of the robot is sketched.

3.2 Optimal Policy and Value Problems

For the RCCMP $(\mathfrak{S}, \mathfrak{R})$ described in Section 3.1, we can define different problems depending on whether we are interested in computing optimal collected reward, deciding on the existence of policies generating a minimum collected reward, or synthesizing optimal policies. These problems can be defined for both finite and infinite horizon, and also features exact and approximate versions.

Definition 5. [**Optimal policy problems**] *Let $(\mathfrak{S}, \mathfrak{R})$ be the RCCMP defined in Section 3.1.*

1. *Given $n \in \mathbb{N}$ and initial probability measure $\alpha \in \mathfrak{D}$, the* finite horizon optimal policy problem *asks to compute a policy $\rho = (\rho_0, \rho_1, \ldots, \rho_{n-1})$ of length n, such that for every policy ρ' of length n it holds that $\mathsf{CDR}_n(\rho, \alpha) \geq \mathsf{CDR}_n(\rho', \alpha)$.*

2. *Given an initial probability measure $\alpha \in \mathfrak{D}$, the* infinite horizon optimal policy problem *asks to compute an infinite policy $\rho = (\rho_0, \rho_1, \rho_2, \ldots)$, such that for every infinite policy ρ' in Π it holds that $\mathsf{CDR}_\infty(\rho, \alpha) \geq \mathsf{CDR}_\infty(\rho', \alpha)$.*

*In the ϵ-*optimal policy problem *we require computation of a policy ρ such that for all ρ', we have respectively for finite and infinite horizon, $\mathsf{CDR}_n(\rho, \alpha) \geq \mathsf{CDR}_n(\rho', \alpha) - \epsilon$ and $\mathsf{CDR}_\infty(\rho, \alpha) \geq \mathsf{CDR}_\infty(\rho', \alpha) - \epsilon$.*

Definition 6. [**Value computation problems**] *Let $(\mathfrak{S}, \mathfrak{R})$ be the RCCMP defined in Section 3.1.*

1. *Given $n \in \mathbb{N}$ and initial probability measure $\alpha \in \mathfrak{D}$, the* finite horizon value computation problem *asks to compute the value*

$$\mathsf{CDR}^*_n(\alpha) := \sup_{\rho\in\Pi} \mathsf{CDR}_n(\rho, \alpha). \tag{4}$$

2. *Given an initial probability measure $\alpha \in \mathfrak{D}$, the* infinite horizon value computation problem *asks to compute the value*

$$\mathsf{CDR}^*_\infty(\alpha) := \sup_{\rho\in\Pi} \mathsf{CDR}_\infty(\rho, \alpha). \tag{5}$$

*In the ϵ-*optimal value problem *we require computation of quantities $\mathsf{CDR}^\epsilon_n(\alpha)$ and $\mathsf{CDR}^\epsilon_\infty(\alpha)$ such that $|\mathsf{CDR}^\epsilon_n(\alpha) - \mathsf{CDR}^*_n(\alpha)| \leq \epsilon$ and $|\mathsf{CDR}^\epsilon_\infty(\alpha) - \mathsf{CDR}^*_\infty(\alpha)| \leq \epsilon$.*

Definition 7. [**Value decision problems**] *Let $(\mathfrak{S}, \mathfrak{R})$ be a RCCMP defined in Section 3.1 and $\mathsf{r_d} \in \mathbb{R}$.*

1. *Given $n \in \mathbb{N}$ and initial probability measure $\alpha \in \mathfrak{D}$, the* finite horizon value decision problem *asks to decide if* $\mathsf{CDR}^*_n(\alpha) \geq \mathsf{r_d}$.

2. *Given an initial probability measure $\alpha \in \mathfrak{D}$, the* infinite horizon value decision problem *asks to decide if* $\mathsf{CDR}^*_\infty(\alpha) \geq \mathsf{r_d}$.

In this paper we formulate the solution of optimal policy and value computation problems. Moreover, we discuss abstraction methods for the ϵ-optimal value computation problem[1]. The usual performance measures in the literature (e.g. total, discounted, or long-run average rewards) have an additive structure that results in *dynamic programming* (DP) procedures. Thus Markov policies are sufficient for the optimization under very mild assumptions [2]. The definition of CDR indicates that in general the computation of $\mathsf{CDR}_n, \mathsf{CDR}_\infty$ (and therefore that of $\mathsf{CDR}^*_n, \mathsf{CDR}^*_\infty$) requires the knowledge of history, thus Markov policies are not sufficient for optimizing the expected CDR. Take for instance the robot dynamics in Example 1. The robot should visit both regions D_1 and D_3 to collect the rewards generated in these regions, so the robot's move from D_2 will be towards either of the regions depending not only on its current location but also on the previously visited regions. To tackle this difficulty, we reformulate the optimization problem via an additive reward function in an augmented state space, for which the theory of DP is rather rich. We study finite and infinite horizon cases in Sections 4 and 5, respectively.

4. FINITE-HORIZON CDR

4.1 Dynamic Programming Formulation

Given the RCCMP $(\mathfrak{S}, \mathfrak{R})$ with the reward structure $\mathfrak{R} = (\mathcal{D}, \lambda, \gamma)$, we consider a new CMP

$$\hat{\mathfrak{S}} = \left(\hat{\mathcal{S}}, \mathcal{U}, \{\hat{\mathcal{U}}(s,y) | (s,y) \in \hat{\mathcal{S}}\}, \hat{T}_{\mathfrak{s}}\right)$$

with an augmented state space $\hat{\mathcal{S}} = \mathcal{S} \times \mathbb{Z}^m_+$, where m is the cardinality of $\mathcal{D}$. The states are of the form (s,y) with coordinates being $s \in \mathcal{S}$, $y \in \mathbb{Z}^m_+$. For a given finite path w_n, the i^{th} element of y is in fact the length of the path starting at the previous occurrence of a state being in set D_i, $y_k(i) := k - \mathsf{Last}(w_n, D_i, k)$. The control space $\mathcal{U}$ is the same and we further define $\hat{\mathcal{U}}(s,y) = \mathcal{U}(s)$. The dynamics of $\hat{\mathfrak{S}}$ are given as follows:

$$\begin{cases} s_{n+1} & \sim & T_{\mathfrak{s}}(\cdot | s_n, u_n) \\ y_{n+1} & = & g_{\mathfrak{d}}(s_n, y_n), \end{cases} \tag{6}$$

where $g_{\mathfrak{d}}(s,y) := y + \mathbf{1}_m - \xi(s).y$ with $\mathbf{1}_m$ being a row vector of dimension m with all elements equal to one. $\xi(s)$ is a row vector of dimension m with elements $\mathbb{I}(s \in D_i)$, $i \in \mathbb{N}[1,m]$. The dot in $\xi(s).y$ indicates the element-wise product. Hence the corresponding transition kernel $\hat{T}_{\mathfrak{s}}$ is given by

$$\hat{T}_{\mathfrak{s}}(B \times \{y'\} | s, y, u) := T_{\mathfrak{s}}(B | s, u) \mathbb{I}\left(y' = g_{\mathfrak{d}}(s,y)\right), \tag{7}$$

for all $B \in \mathcal{B}(\mathcal{S})$. In words, the state s is updated stochastically according to $T_{\mathfrak{s}}$ while the state y is updated deterministically by incrementing all its elements by one except the i^{th} element which is set to one.

[1]The proposed methods can be used iteratively to answer the value decision problem in Definition 7 with termination guarantees for any $\mathsf{r_d} \neq \mathsf{CDR}^*_n, \mathsf{CDR}^*_\infty$.

We construct a space of policies $\hat{\Pi}$ and for each $\hat{\rho} \in \hat{\Pi}$, a probability measure $\hat{P}^{\hat{\rho}}$ with the expectation $\hat{\mathbb{E}}^{\hat{\rho}}$. We denote by $\hat{\Pi}_M \subset \hat{\Pi}$ the corresponding class of Markov policies for $\hat{\mathfrak{S}}$. The reward structure consists of reward functions $\mathsf{rew} : \mathbb{N} \times \hat{\mathcal{S}} \to \mathbb{R}$, given by

$$\mathsf{rew}(k,s,y) := \sum_{t=0}^{\xi(s)y^T - 1} \gamma(\Xi(s))^t \lambda(k-t, \Xi(s)), \tag{8}$$

and additive functional $\widehat{\mathsf{CDR}}^{\hat{\rho}}_n(s,y) := \hat{\mathbb{E}}^{\hat{\rho}}_{s,y}\left[\sum_{k=0}^{n} \mathsf{rew}(k, s_k, y_k)\right]$ and its long-run average $\widehat{\mathsf{CDR}}^{\hat{\rho}}_\infty(s,y) := \liminf_{n\to\infty} \frac{1}{n+1} \widehat{\mathsf{CDR}}^{\hat{\rho}}_n(s,y)$.

In order to relate $(\hat{\mathfrak{S}}, \mathsf{rew})$ to the original formulation defined over the RCCMP $(\mathfrak{S}, \mathfrak{R})$, we first have to establish an explicit relationship between classes of strategies Π and $\hat{\Pi}$. Clearly, we can treat Π as a subset of $\hat{\Pi}$ as any policy $\rho \in \Pi$ for the CMP $\mathfrak{S}$ serves also as a policy for the CMP $\hat{\mathfrak{S}}$. We let $\iota : \Pi \to \hat{\Pi}$ be the inclusion map. On the other hand, we define the projection map $\theta : \hat{\Pi} \to \Pi$ by

$$\theta_j(\rho)(du_j | w_j) := \hat{\rho}_j(du_j | s_0, y_0, u_0, \ldots, s_j, y_j), \tag{9}$$

with $w_j = (s_0, u_0, s_1, u_1, \ldots, s_j)$, $y_k(i) = k - \mathsf{Last}(w_j, D_i, k)$, for all $i \in \mathbb{N}[1,m]$ and $k \in \mathbb{N}[0,j]$. The following result relates the two optimization problems.

THEOREM 1. *For any $n \in \mathbb{N}$, $\rho \in \Pi$ and $\hat{\rho} \in \hat{\Pi}$, it holds that*

$$\widehat{\mathsf{CDR}}^{\hat{\rho}}_n(s, \mathbf{1}_m) = \mathsf{CDR}_n(\theta(\hat{\rho}), s),\ \mathsf{CDR}_n(\rho, s) = \widehat{\mathsf{CDR}}^{\iota(\rho)}_n(s, \mathbf{1}_m),$$

$$\widehat{\mathsf{CDR}}^{\hat{\rho}}_\infty(s, \mathbf{1}_m) = \mathsf{CDR}_\infty(\theta(\hat{\rho}), s),\ \mathsf{CDR}_\infty(\rho, s) = \widehat{\mathsf{CDR}}^{\iota(\rho)}_\infty(s, \mathbf{1}_m).$$

□

Theorem 1 has several important corollaries. First of all, it can be used to prove that Markov policies $\hat{\Pi}_M$ are sufficient for the finite-horizon optimal value problem of RCCMP $(\mathfrak{S}, \mathfrak{R})$ in the *augmented* state space $\hat{\mathcal{S}}$. At the same time, the optimal policy may depend on time and thus is not necessary stationary. Let us further define $\widehat{\mathsf{CDR}}^*_n(s,y) := \sup_{\hat{\rho} \in \hat{\Pi}} \widehat{\mathsf{CDR}}^{\hat{\rho}}_n(s,y)$ and $\widehat{\mathsf{CDR}}^*_\infty(s,y) := \sup_{\hat{\rho} \in \hat{\Pi}} \widehat{\mathsf{CDR}}^{\hat{\rho}}_\infty(s,y)$.

COROLLARY 1. *For any $n \in \mathbb{N}$ and $s \in \mathcal{S}$, it holds that $\mathsf{CDR}^*_n(s) = \widehat{\mathsf{CDR}}^*_n(s, \mathbf{1}_m)$ and $\mathsf{CDR}^*_\infty(s) = \widehat{\mathsf{CDR}}^*_\infty(s, \mathbf{1}_m)$.* □

Finally, we can exploit DP recursions for the additive functionals $\widehat{\mathsf{CDR}}^*_n$ to compute the finite-horizon optimal value problem of RCCMP $(\mathfrak{S}, \mathfrak{R})$. Let us introduce the following time-dependent operators

$$\mathfrak{J}_k f(s,y,u) := \mathsf{rew}(k,s,y) + \\ + \sum_{y' \in \mathbb{Z}^m_+} \int_{\mathcal{S}} f(s', y') \hat{T}_{\mathfrak{s}}(ds' \times \{y'\} | s, y, u), \tag{10}$$

and $\mathfrak{J}^*_k f(s,y) = \sup_{u \in \hat{\mathcal{U}}(s,y)} \mathfrak{J}_k f(s,y,u)$, which act on the space of bounded universally measurable functions. These operators can be used to compute optimal value functions recursively, as the following result states.

COROLLARY 2. *For any $n \in \mathbb{N}$, consider value functions $V_k : \hat{\mathcal{S}} \to \mathbb{R}$, $k \in \mathbb{N}[0,n]$ defined recursively as*

$$V_k = \mathfrak{J}^*_k V_{k+1}, \quad V_n(s,y) = \mathsf{rew}(n,s,y).$$

These value functions are universally measurable. Moreover, $\widehat{\mathsf{CDR}}^*_n(s,y) = V_0(s,y)$. □

Note that the operators in (10) can be further simplified to

$$\mathfrak{J}_k f(s,y,u) := \mathsf{rew}(k,s,y) + \int_{\mathcal{S}} f(s', g_{\mathfrak{d}}(s,y)) T_{\mathfrak{s}}(ds'|s,u). \tag{11}$$

4.2 Approximate Abstractions

Since the recursion in Corollary 2 does not admit a closed-form solution, we introduce an abstraction procedure, which results in numerical methods for the computation of such functions. Moreover, we provide an explicit upper bound on the error caused by the abstraction. We focus on finite-horizon optimal value problem in this section and then present the results for the infinite-horizon case in Section 5.

The abstraction algorithm initially proposed in [1] and further developed in [5, 17] are not directly applicable to our problem. First, in these works the state space of the process is considered to be continuous or hybrid. Applying such techniques to our problem that has $\mathcal{S} \times \mathbb{Z}_+^m$ as its state space results in a *countable unbounded* abstract space, which is difficult to deal with computationally. Second, the error of these abstraction algorithms are computed with respect to formal synthesis of policies for satisfaction of a given specification, while in our case we are optimizing collected rewards. In this section we present the abstraction algorithm adapted to our problem and then show how to solve the ϵ-optimal value problem.

Algorithm 1 presents the procedure for abstracting RCCMP $(\mathfrak{S}, \mathfrak{R})$ to a finite-state RCCMP $(\mathfrak{M}, \mathfrak{R}_{\mathfrak{d}})$. It works directly on the CMP $\mathfrak{S}$ and computes MDP $\mathfrak{M}$ as its abstraction. It also gives the construction of collecting reward structure $\mathfrak{R}_{\mathfrak{d}}$ on the MDP $\mathfrak{M}$. Here the state space $\mathcal{S}$ is partitioned such that the partition refines $\mathcal{D}$, i.e., for any $i \in \mathbb{N}[1, m_{\mathfrak{s}}]$ there is a $D \in \mathcal{D}$ such that $S_i \subset D$. Then representative points z_i are selected and the state space of $\mathfrak{M}$ is constructed in Step 3. The input sets $\mathcal{U}(z_i)$ are also partitioned in Step 4 and arbitrary representative points are selected in Step 5. Step 6 defines the set of valid discrete inputs at each state and finally Step 7 gives the transition probabilities of the MDP $\mathfrak{M}$.

In this algorithm $\Xi_{\mathfrak{s}} : \mathcal{S} \to 2^{\mathcal{S}}$ is a set-valued map that assigns any state $s \in \mathcal{S}$ to the partition set it belongs to, i.e., $\Xi_{\mathfrak{s}}(s) = S_i$ whenever $s \in S_i$. Step 8 constructs the collecting reward structure by defining discrete regions $\mathcal{D}_{\mathfrak{d}}$ (intersection of elements of $\mathcal{D}$ with the discrete space $\mathcal{S}_{\mathfrak{d}}$) and then restricting functions λ, γ to $\mathcal{D}_{\mathfrak{d}}$. Moreover, it selects a different discounting sequence in which the power of discounting factor is saturated with a constant $\ell \in \mathbb{N}$.

The DP formulation in Section 4.1 is also applicable to the RCCMP $(\mathfrak{M}, \mathfrak{R}_{\mathfrak{d}})$. Due to the particular choice of discounting sequence $\gamma_{\mathfrak{d}}$ in Step 8 of Algorithm 1, the augmented MDP $\hat{\mathfrak{M}}$ will have the finite state space $\hat{S}_{\mathfrak{d}} = S_{\mathfrak{d}} \times \mathbb{N}[1, \ell]^m$. Its transition probabilities are also defined as

$$\hat{T}_{\mathfrak{d}}(z', w'|z, w, v) = T_{\mathfrak{d}}(z'|z, v) \mathbb{I}(w' = g_\ell(z, w)),$$

which requires that the second coordinate of the state $(z, w) \in \hat{S}_{\mathfrak{d}}$ is deterministically updated according to $w' = g_\ell(z, w)$ with

$$g_\ell(z, w) := \min\{w + \mathbf{1}_m - \xi(z).w, \ell\}.$$

Next we present the DP recursion for computation of ϵ-

Algorithm 1 Abstraction of RCCMP $(\mathfrak{S}, \mathfrak{R})$ by finite-state RCCMP $(\mathfrak{M}, \mathfrak{R}_{\mathfrak{d}})$

Require: input model $\mathfrak{S} = (\mathcal{S}, \mathcal{U}, \{\mathcal{U}(s) | s \in \mathcal{S}\}, T_{\mathfrak{s}})$ and reward structure $\mathfrak{R} = (\mathcal{D}, \lambda, \gamma)$

1: Select a finite partition $\{S_1, \ldots, S_{m_{\mathfrak{s}}}\}$ of $\mathcal{S}$ which refines $\mathcal{D}$

2: For each S_i, select a single representative point $z_i \in S_i$

3: Define $S_{\mathfrak{d}} = \{z_i \mid i \in \mathbb{N}[1, m_{\mathfrak{s}}]\}$ as the state space of the MDP $\mathfrak{M}$

4: For each $i \in \mathbb{N}[1, m_{\mathfrak{s}}]$, select a finite partition of the input set $\mathcal{U}(z_i)$ as $\mathcal{U}(z_i) = \cup_{j=1}^{m_{\mathfrak{u}_i}} U_{ij}$ where $m_{\mathfrak{u}_i}$ represents the cardinality of the partition of $\mathcal{U}(z_i)$

5: For each U_{ij}, select single representative point $v_{ij} \in U_{ij}$

6: Define $\mathcal{U}_{\mathfrak{d}} = \{v_{ij} \mid j \in \mathbb{N}[1, m_{\mathfrak{u}_i}], i \in \mathbb{N}[1, m_{\mathfrak{s}}]\}$ as the finite input space of the MDP $\mathfrak{M}$, $\mathcal{U}_{\mathfrak{d}}(z_i) = \{v_{ij} \mid j \in \mathbb{N}[1, m_{\mathfrak{u}_i}]\}$ as the set of feasible inputs when $\mathfrak{M}$ is at any state $z_i \in \mathcal{S}_{\mathfrak{d}}$

7: Compute the state transition matrix $\hat{T}_{\mathfrak{d}}$ for $\mathfrak{M}$ as:

$$T_{\mathfrak{d}}(z'|z, v) = \hat{T}_{\mathfrak{s}}(\Xi_{\mathfrak{s}}(z')|z, v), \tag{12}$$

for any $z, z' \in \mathcal{S}_{\mathfrak{d}}$ and $v \in \mathcal{U}_{\mathfrak{d}}(z)$

8: Define discrete regions $\mathcal{D}_{\mathfrak{d}} := \{D_i \cap S_{\mathfrak{d}} \mid i \in \mathbb{N}[1, m]\}$, discounting sequence $\gamma_{\mathfrak{d}} := \{1, \gamma, \gamma^2, \ldots, \gamma^{\ell-1}, \gamma^\ell, \gamma^\ell, \ldots\}$, and $\lambda_{\mathfrak{d}} := \lambda|_{\mathcal{D}_{\mathfrak{d}}}$

Ensure: output finite-state RCCMP $(\mathfrak{M}, \mathfrak{R}_{\mathfrak{d}})$ with $\mathfrak{M} = (\mathcal{S}_{\mathfrak{d}}, \mathcal{U}_{\mathfrak{d}}, \{\mathcal{U}_{\mathfrak{d}}(z) | (z) \in \mathcal{S}_{\mathfrak{d}}\}, T_{\mathfrak{d}})$ and $\mathfrak{R}_{\mathfrak{d}} = (\mathcal{D}_{\mathfrak{d}}, \lambda_{\mathfrak{d}}, \gamma_{\mathfrak{d}})$

---

value and policy problem. Define operators

$$\bar{\mathfrak{J}}_k f_{\mathfrak{d}}(z, w, v) := \mathsf{rew}_{\mathsf{a}}(k, z, w) + \sum_{(z', w') \in \hat{\mathcal{S}}_{\mathfrak{d}}} f_{\mathfrak{d}}(z', w') \hat{T}_{\mathfrak{d}}(z', w'|z, w, v),$$

and $\bar{\mathfrak{J}}_k^* f_{\mathfrak{d}}(z, w) = \max_{v \in \hat{\mathcal{U}}_{\mathfrak{d}}(z,w)} \bar{\mathfrak{J}}_k f_{\mathfrak{d}}(z, w, v)$. The functions $\mathsf{rew}_{\mathsf{a}}$ are defined as

$$\mathsf{rew}_{\mathsf{a}}(k, z, w) = \sum_{t=0}^{\xi(z) w^T - 1} \gamma(\Xi_{\mathfrak{d}}(z))^t \lambda(k - t, \Xi_{\mathfrak{d}}(z)),$$

where the function $\Xi_{\mathfrak{d}} : \mathcal{S}_{\mathfrak{d}} \to \mathcal{D}_{\mathfrak{d}}$ assigns to any discrete state $z \in D_i$ the discrete region $\Xi_{\mathfrak{d}}(z) = D_i \cap \mathcal{S}_{\mathfrak{d}}$. (cf. the reward function in (8)).

The discrete value functions are computed using the recursion $\bar{V}_k = \bar{\mathfrak{J}}_k^* \bar{V}_{k+1}$ with $\bar{V}_n(z, w) = \mathsf{rew}_{\mathsf{a}}(n, z, w)$. Then the approximate solution of the finite-horizon CDR will be $\overline{\mathsf{CDR}}^*(z, w) = \bar{V}_0(z, w)$. The finite-state MDP $\mathfrak{M}$ can be computed using software tool FAUST^2 [7] and the value functions can be computed with numerically efficient methods [11]. We discuss in the next section how the error of the abstraction algorithm 1 can be quantified based on suitable assumptions on the RCMCP.

4.3 Error Computation

Since $\mathcal{S}$ and $\mathcal{U}$ are Borel spaces they are metrizable topological spaces. Let $d_{\mathfrak{s}}$ and $d_{\mathfrak{u}}$ be metrics on $\mathcal{S}$ and $\mathcal{U}$ respectively, which are consistent with the given topologies of the underlying spaces. Define the *diameter* of a set $A \subset \mathcal{S}$ as

$$\operatorname{diam}_{\mathfrak{s}}(A) := \sup \left\{ d_{\mathfrak{s}}(s, s') | \; s, s' \in \mathcal{S} \right\},$$

likewise for subsets of $\mathcal{U}$. Also define diameter of the partition $\mathcal{S} = \cup_{i=1}^{m_{\mathfrak{s}}} S_i$ as the maximum diameter of its elements $\delta_{\mathfrak{s}} := \max_i \operatorname{diam}_{\mathfrak{s}}(S_i)$, and $\delta_{\mathfrak{u}} := \max_{i,j} \operatorname{diam}_{\mathfrak{u}}(U_{ij})$. We as-

sume that the selected partition sets S_i refine $\mathcal{D}$, i.e., for any $i \in \mathbb{N}[1, m_{\mathfrak{s}}]$ and $D \in \mathcal{D}$ either $S_i \subset D$ or $S_i \cap D = \emptyset$.

The error quantification of the MDP abstraction approach, presented in Algorithm 1, requires the study of the family of sets $\mathcal{U}(s)$ as a function of state. For this purpose, we assign the Hausdorff distance to the family of non-empty subsets of $\mathcal{U}$, which is defined as

$$d_H(X, Y) := \max \left\{ \sup_{x \in X} \inf_{y \in Y} d_{\mathfrak{u}}(x, y), \sup_{y \in Y} \inf_{x \in X} d_{\mathfrak{u}}(x, y) \right\},$$

for all $X, Y \subset \mathcal{U}$. The next assumption poses a regularity condition on state-dependent input sets.

ASSUMPTION 1. *There exists a constant $h_{\mathfrak{u}} \in \mathbb{R}$ such that the family of state-dependent input sets $\{\mathcal{U}(s) | s \in \mathcal{S}\}$ satisfies the Lipschitz inequality*

$$d_H(\mathcal{U}(s), \mathcal{U}(s')) \leq h_{\mathfrak{u}} d_{\mathfrak{s}}(s, s') \quad \forall s, s' \in \mathcal{S}.$$

The error quantification also requires a regularity assumption on the stochastic kernel $T_{\mathfrak{s}}(\bar{s}|s, u)$. Given a function $f : \mathcal{S} \to \mathbb{R}$, we define $T_{\mathfrak{s}} f : \mathcal{K} \to \mathbb{R}$ as

$$T_{\mathfrak{s}} f(s, u) = \int_{\mathcal{S}} f(\bar{s}) T_{\mathfrak{s}}(\bar{s}|s, u),$$

provided that the corresponding integrals are well defined and finite. We pose the following assumption on the stochastic kernel $T_{\mathfrak{s}}$ of the process.

ASSUMPTION 2. *There exists a constant $h_{\mathfrak{T}} > 0$ such that for every (s, u) and (s', u') in $\mathcal{K}$, and bounded function $f : \mathcal{S} \to \mathbb{R}$, with Lipschitz constant h_f,*

$$|T_{\mathfrak{s}} f(s, u) - T_{\mathfrak{s}} f(s', u')| \leq h_{\mathfrak{T}} h_f \left[d_{\mathfrak{s}}(s, s') + d_{\mathfrak{u}}(u, u') \right].$$

The kernel $T_{\mathfrak{s}}$ is said to be $h_{\mathfrak{T}}$-Lipschitz continuous. We also assume that $T_{\mathfrak{s}}(D_i|s, u)$ is Lipschitz continuous, i.e., there exists a constant $h_{\mathfrak{c}} > 0$ such that for all $i \in \mathbb{N}[1, m]$,

$$|T_{\mathfrak{s}}(D_i|s, u) - T_{\mathfrak{s}}(D_i|s', u')| \leq h_{\mathfrak{c}} \left[d_{\mathfrak{s}}(s, s') + d_{\mathfrak{u}}(u, u') \right].$$

The following lemma (1) provides an upper bound on the value functions V_k, $k \in \mathbb{N}[0, n]$ and (2) under Assumptions 1 and 2, establishes piecewise continuity properties of the value functions. Note that the reward functions $\mathsf{rew}(k, s, y)$ are piecewise constant with continuity regions $D_i \in \mathcal{D}, i \in \mathbb{N}[1, m]$.

LEMMA 1. *1. Assume there is a constant $\lambda_{\mathfrak{m}} \in \mathbb{R}_{\geq 0}$ such that $\lambda(t, D) \leq \lambda_{\mathfrak{m}}$ for all $D \in \mathcal{D}$ and $t \in \mathbb{N}$. Let $\kappa := \lambda_{\mathfrak{m}}/(1 - \gamma_{\mathfrak{m}})$ with $\gamma_{\mathfrak{m}} := \max_i \gamma(D_i)$. Then the reward functions are bounded $\mathsf{rew}(k, s, y) \leq \kappa$ and the value functions are bounded by*

$$V_k(s, y) \leq (n + 1 - k)\kappa, \quad \forall (s, y) \in \hat{\mathcal{S}},\ k \in \mathbb{N}[0, n],$$

2. Under Assumptions 1 and 2, the value functions V_k are piecewise Lipschitz continuous with continuity regions D_i, and their Lipschitz constants are L_k, computed recursively with $L_n = 0$ and

$$L_k = (1 + h_{\mathfrak{u}}) \left[h_{\mathfrak{T}} L_{k+1} + m h_{\mathfrak{c}} \kappa \right],\ k \in \mathbb{N}[0, n-1], \quad (13)$$

where m is the cardinality of $\mathcal{D}$ and κ and $\gamma_{\mathfrak{m}}$ are defined as above.

□

Piecewise continuous value functions enable us to quantify the abstraction error of Algorithm 1 induced on the respective value functions. Define the function $\xi_{\mathfrak{d}}$ on $\hat{\mathcal{S}}$ such that $\xi_{\mathfrak{d}}(s, y)$ assigns the associated representative point (z, w) to (s, y) as selected in Algorithm 1. Then we have the following theorem.

THEOREM 2. [**Finite-horizon ε-optimal value problem**] *Suppose Assumptions 1 and 2 hold. Define $\mathcal{L}_{\mathfrak{u}} := \sum_{k=0}^{n-1} L_k$, $\mathcal{L}_{\mathfrak{s}} := h_{\mathfrak{T}} \sum_{k=1}^{n} L_k$, and $\epsilon(\ell) := (n+1)\kappa\gamma_{\mathfrak{m}}^{\ell}$, where L_k, κ, and $\gamma_{\mathfrak{m}}$ are defined as in Lemma 1. The abstraction error of Algorithm 1 on the computed optimal finite-horizon CDR is*

$$|\widehat{\mathsf{CDR}}_n^*(s, y) - \overline{\mathsf{CDR}}_n^*(\xi_{\mathfrak{d}}(s, y))| \leq \mathcal{L}_{\mathfrak{u}} \delta_{\mathfrak{u}} + \mathcal{L}_{\mathfrak{s}} \delta_{\mathfrak{s}} + \epsilon(\ell), \quad (14)$$

for all $(s, y) \in \hat{\mathcal{S}}$. □

The error bound in Theorem 2 can be used to solve the ε-optimal policy problem as follows.

COROLLARY 3. [**Finite-horizon ε-optimal policy problem**] *Suppose Assumptions 1 and 2 hold. If we synthesize an optimal policy $\bar{\rho}^* = (\bar{\rho}_0^*, \bar{\rho}_1^*, \ldots)$ for $\mathfrak{M}$ and apply the policy $\rho = (\rho_0, \rho_1, \ldots)$ with $\rho = \theta \bar{\rho}^* \xi_{\mathfrak{d}}(\cdot)$ to $\mathfrak{S}$, with θ being the policy projection map defined in (9), then the error will be*

$$|\mathsf{CDR}_n(\rho, s) - \mathsf{CDR}_n^*(s)| \leq 2(\mathcal{L}_{\mathfrak{u}} \delta_{\mathfrak{u}} + \mathcal{L}_{\mathfrak{s}} \delta_{\mathfrak{s}} + \epsilon(\ell)).$$

Note that the approximate optimal policy is computed as follows: compute (s_j, y_j) for a given path $(s_0, u_0, \ldots, s_j)$; find the discrete representative state $(z_j, w_j) \in \hat{\mathcal{S}}_{\mathfrak{d}}$; compute $v_j = \bar{\rho}_j^*(z_j, w_j) \in \mathcal{U}(s_j)$ and apply it to $\mathfrak{S}$.

The abstraction error in (14) has three terms: the first term is related to discretization of the input space; the second term reflects the effect of discretizing the state space; and the last term is related to the choice of discounting sequence in step 8 of the abstraction algorithm. The error can be tuned by proper selection of partition diameters $\delta_{\mathfrak{u}}$ and $\delta_{\mathfrak{s}}$ and the choice of ℓ.

REMARK 2. *The above error computation is distinct from the one from [1, 6, 16] in there is no requirement on having a bounded state space or on value functions being in the interval $[0, 1]$.*

EXAMPLE 2. *Consider a nonlinear dynamical system with additive noise*

$$s_{t+1} = f_{\mathfrak{m}}(s_t, u_t) + \eta_t,$$

where $\{\eta_t,\ t \in \mathbb{N}\}$ are iid with the distribution $\eta_t \sim T_\eta(\cdot)$. Suppose $f_{\mathfrak{m}}$ is Lipschitz continuous with constant $h_{f_{\mathfrak{m}}}$. Then Assumption 2 holds for this system with the same constant $h_{\mathfrak{T}} = h_{f_{\mathfrak{m}}}$ no matter what the distribution of noise $T_\eta(\cdot)$ is. In contrast, previous error analysis in [1, 6] requires continuity of $T_\eta(\cdot)$.

The behavior of the error in Theorem 2(2) as a function of horizon n depends on the constant $(1+h_{\mathfrak{u}})h_{\mathfrak{T}}$ in recursion (13): the error grows exponentially if $(1+h_{\mathfrak{u}})h_{\mathfrak{T}} > 1$; it grows quadratically if $(1 + h_{\mathfrak{u}})h_{\mathfrak{T}} = 1$; and it diverges linearly if $(1 + h_{\mathfrak{u}})h_{\mathfrak{T}} < 1$. Thus the error analysis of the abstraction method is useful for the infinite-horizon CDR only in the last case, i.e., $(1+h_{\mathfrak{u}})h_{\mathfrak{T}} < 1$: the linearly growing error will be normalized by the horizon and gives a bounded tunable error. We study the infinite-horizon CDR in the next section based on the limiting behavior of the CMP.

5. INFINITE-HORIZON CDR

Recall the definition of infinite-horizon CDR $\mathsf{CDR}_\infty(\rho,\alpha)$ in (3) for a policy $\rho \in \Pi$ and initial distribution $\alpha \in \mathfrak{D}$. For the sake of succinct presentation of the theoretical results, with a slight abuse of notation, we construct the augmented process $\hat{\mathfrak{S}}$ based on the modified discounting sequence $\{1,\gamma,\gamma^2,\ldots,\gamma^{\ell-1},\gamma^\ell,\gamma^\ell,\ldots\}$. Thus the dynamics of $\hat{\mathfrak{S}}$ are

$$\begin{cases} s_{n+1} & \sim & T_{\mathfrak{s}}(\cdot|s_n,u_n) \\ y_{n+1} & = & g_\ell(s_n,y_n), \end{cases} \tag{15}$$

where $g_\ell(s,y) := \min\{y+\mathbf{1}_m-\xi(s).y,\ell\}$ (cf. dynamics in (6)) with ℓ being a properly chosen value (cf. Theorem 5). Based on our discussion in Section 4.3, the induced error on the infinite-horizon CDR is upper bounded by

$$|\,\mathsf{CDR}^*_\infty(s) - \widehat{\mathsf{CDR}}^*_\infty(s,\mathbf{1}_m)| \le \epsilon_1 := \kappa\gamma_{\mathfrak{m}}^\ell,$$

where $\widehat{\mathsf{CDR}}^*_\infty(s,y)$ is the optimal long-run average reward function over the augmented process $\hat{\mathfrak{S}}$ with dynamics (15),

$$\widehat{\mathsf{CDR}}^*_\infty(s,y) = \sup_{\hat\rho} \liminf_{n\to\infty} \frac{1}{n+1}\hat{\mathbb{E}}^{\hat\rho}_{s,y}\left[\sum_{k=0}^{n}\mathsf{rew}(k,s_k,y_k)\right],$$

and the reward function $\mathsf{rew}(k,s_k,y_k)$ is defined in (8). We also assume that the expected reward λ is stationary (it is only a function of regions and does not depend on time). Therefore $\mathsf{rew}(k,s,y)$ will be denoted by $\mathsf{rew}(s,y)$. The quantity $\widehat{\mathsf{CDR}}^*_\infty(s,y)$ depends on the limiting behavior of the process $\hat{\mathfrak{S}}$ and its computational aspect varies depending on the structural properties of the process [10, 12]. For instance ergodicity of the process under any stationary policy ensures that the optimal average reward $\widehat{\mathsf{CDR}}^*_\infty(s,y)$ is independent of the initial state (s,y). We present in Section 5.1 an optimality equation whose solution gives the optimal average reward. We provide an assumption on the original process $\mathfrak{S}$ under which the optimality equation has a solution. In Section 5.2, we discuss value iteration for the computation of the solution of the optimality equation and provide an approximation procedure based on abstraction with guaranteed error bounds.

5.1 Optimality Equation

Define $\mathbb{B}(\hat{\mathcal{S}})$ as the Banach space of real-valued bounded measurable functions $f:\hat{\mathcal{S}}\to\mathbb{R}$ with the supremum norm $\|f\| := \sup_{\hat{s}\in\hat{\mathcal{S}}}|f(\hat{s})|$. Under the assumption of generated reward being stationary in expectation (thus having bounded time-independent $\mathsf{rew}(s,y)$), the following theorem presents optimality equation for the infinite-horizon CDR.

THEOREM 3. *Suppose the generated reward is stationary in expectation. If there is a constant g and a function v^* in $\mathbb{B}(\hat{\mathcal{S}})$ such that for all $(s,y)\in\hat{\mathcal{S}}$,*

$$g+v^*(s,y) = \sup_{u\in\mathcal{U}(s)}\left\{\mathsf{rew}(s,y)+\int_{\hat{\mathcal{S}}} v^*(s',y')\hat{T}_{\mathfrak{s}}(ds',dy'|s,y,u)\right\}, \tag{16}$$

then $\widehat{\mathsf{CDR}}^*_\infty(s,y)\le g$ for all $(s,y)\in\hat{\mathcal{S}}$. If $d^*\in\hat{\Pi}_S$ is a stationary policy such that $d^*:\hat{\mathcal{S}}\to\mathcal{U}$ and $d^*(s,y)\in\mathcal{U}(s)$ maximizes the right-hand side of optimality equation (16), then d^* is optimal and $\widehat{\mathsf{CDR}}^{d^*}_\infty(s,y)=g$ for all $(s,y)\in\hat{\mathcal{S}}$. □

If g and $v^*\in\mathbb{B}(\hat{\mathcal{S}})$ are as in Theorem 3, it is then said that $\{g,v^*\}$ is a *solution* to the optimality equation (OE) (16). The OE (16) is sometimes called the average-reward dynamic programming equation [10]. We also define the DP operator

$$\mathfrak{J}^*f(s,y) := \mathsf{rew}(s,y)+\sup_{u\in\mathcal{U}(s)}\int_{\mathcal{S}} f(s',g_\ell(s,y))T_{\mathfrak{s}}(ds'|s,u), \tag{17}$$

which is the time-independent version of (11) adapted to the dynamics (15). Using the DP operator $\mathfrak{J}^*$ in (17) we can write the OE (16) as

$$g+v^*(s,y)=\mathfrak{J}^*v^*(s,y),\quad \forall(s,y)\in\hat{\mathcal{S}}.$$

Note that the solution of OE is not unique in general if it exists at all. In fact, if $\{g,v^*\}$ is a solution to the OE, so if $\{g,v^*+\varrho\}$ for any $\varrho\in\mathbb{R}$. Even if there is a solution $\{g,v^*\}$, it is not guaranteed to get a stationary policy maximizing the right-hand side of OE. The following lemma guarantees existence of such a policy.

LEMMA 2. *Suppose $\mathcal{U}(s)$ is a (non-empty) compact subset of $\mathcal{U}$ for each state $s\in\mathcal{S}$ and the generated reward is stationary in expectation. Then under Assumption 2, if the OE (16) has a solution, there exists a stationary policy $d^*\in\hat{\Pi}_S$ that achieves the optimal value g.* □

The next thing to look at is the existence of a solution for the OE (16). Ergodicity conditions for continuous space processes are discussed in [9] and structural conditions for countable space processes are presented in [12]. We adapt the assumption from [9] to the CMP $\hat{\mathfrak{S}}$.

ASSUMPTION 3. *For any stationary policy $\hat{d}\in\hat{\Pi}_S$ with $\hat{d}:\hat{\mathcal{S}}\to\mathcal{U}$ there exists a probability measure $\hat{p}_{\hat{d}}$ on $\hat{\mathcal{S}}$ such that*

$$\|\hat{T}^k_{\hat{d}}(\cdot|s,y)-\hat{p}_{\hat{d}}(\cdot)\|\le\hat{\mathfrak{c}}_k,\quad\forall(s,y)\in\hat{\mathcal{S}},\ k\in\mathbb{N},$$

where the sequence $\{\hat{\mathfrak{c}}_k,\,k\in\mathbb{N}\}$ is independent of (s,y) and of $\hat{d}$, and $\sum_k\hat{\mathfrak{c}}_k<\infty$. Here $\hat{T}^k_{\hat{d}}(\cdot|s,y)$ denotes the $k-$step transition probability measure of the Markov process $\hat{\mathfrak{S}}$ when the stationary policy $\hat{d}\in\hat{\Pi}_S$ is used, given that the initial state is (s,y). The norm $\|\cdot\|$ denotes the total variation norm for signed measures.

For probability measures P_1 and P_2 on $(\hat{\mathcal{S}},\mathcal{B}(\hat{\mathcal{S}}))$, recall that P_1-P_2 is a finite signed measure and its total variation is given by

$$\|P_1-P_2\| = 2\sup_{B\in\mathcal{B}(\hat{\mathcal{S}})}|P_1(B)-P_2(B)|.$$

If P_1 and P_2 have densities p_1 and p_2 with respect to some sigma-finite measure μ on $\hat{\mathcal{S}}$, then

$$\|P_1-P_2\|=\int_{\hat{\mathcal{S}}}|p_1-p_2|d\mu.$$

THEOREM 4. *Under Assumption 3, the optimal average reward $\widehat{\mathsf{CDR}}^*_\infty(s,y)$ is independent of the initial state (s,y) and the optimality equation (16) has a solution.* □

Assumption 3 puts a restriction on the augmented CMP $\hat{\mathfrak{S}}$. It is possible to check satisfaction of this assumption by looking at the CMP $\mathfrak{S}$. More precisely, Assumption 3 holds

if the same condition is true for $\mathfrak{S}$ with a larger class of policies, namely history-dependent policies with finite memory:

$$\|T_d^k(\cdot|s) - p_d(\cdot)\| \leq \mathfrak{c}_k, \quad \forall s \in \mathcal{S},\ k \in \mathbb{N},$$

for all deterministic policies $(d, d, \ldots) \in \Pi$ with $d(w_n)$ being only a function of $(s_{n-\ell+1}, \ldots, s_{n-1}, s_n)$.

Existence of a solution for the OE (16) is ensured by Assumption 3. In the next section we study value iteration method for the approximate computation of the solution with guaranteed error bounds.

5.2 Value Iteration

In this section we discuss how the solution of OE (16) can be obtained using value iteration under proper assumptions on the operator $\mathfrak{J}^*$. We define the value iteration functions $W_k \in \mathbb{B}(\hat{\mathcal{S}})$ by

$$W_{n+1} = \mathfrak{J}^* W_n = \mathfrak{J}^{*n+1} W_0,\ n \in \mathbb{N}, \tag{18}$$

where $W_0(s, y) \in \mathbb{B}(\hat{\mathcal{S}})$ is arbitrary. As we observed in Section 4, $W_n(s, y)$ can be interpreted as the maximal expected reward for finite horizon n when the initial state is $(s_0, y_0) = (s, y)$ if the initial value function $W_0(s, y) = \mathsf{rew}(s, y)$ is selected. Clearly, as $n \to \infty$, W_n might not converge to a function in $\mathbb{B}(\hat{\mathcal{S}})$. We put the following assumption that ensures appropriate transformations of W_n do converge.

Assumption 4. *The DP operator* (17) *is a span-contraction operator, i.e.,*

$$sp\left(\mathfrak{J}^* f_1 - \mathfrak{J}^* f_2\right) \leq \alpha_{\mathfrak{J}}\, sp(f_1 - f_2), \quad \forall f_1, f_2 \in \mathcal{B}(\hat{\mathcal{S}}),$$

for some $\alpha_{\mathfrak{J}} < 1$. *The* span semi-norm *of a function is defined as* $sp(f) := \sup_{\hat{s}} f(\hat{s}) - \inf_{\hat{s}} f(\hat{s})$.

Banach's fixed point theorem for contraction operators on complete metric spaces [9] implies that under Assumption 4, $\mathfrak{J}^*$ has a span-fixed-point, i.e., there is a function $v^* \in \mathbb{B}(\hat{\mathcal{S}})$ such that $sp(\mathfrak{J}^* v^* - v^*) = 0$. Equivalently, $\mathfrak{J}^* v^* - v^*$ is a constant function. Thus, the OE has a solution.

Remark 3. *Assumption 4 may be generalized by requiring multi-step span-contraction, i.e., there exists an positive integer* ϑ *such that* $\mathfrak{J}^{*\vartheta}$ *is a span-contraction operator. The following results are also valid for such operators.*

Let us define a sequence of functions e_n in $\mathbb{B}(\hat{\mathcal{S}})$ by

$$e_n(s, y) := \mathfrak{J}^{*n} W_0(s, y) - \mathfrak{J}^{*n} v^*(s, y) = W_n(s, y) - v^*(s, y) - ng,$$

for all $(s, y) \in \hat{\mathcal{S}}$ and $n \in \mathbb{N}$. We also define $v_n^+ := \sup(W_n - W_{n-1})$ and $v_n^- := \inf(W_n - W_{n-1})$.

Lemma 3. *The sequence* v_n^+ *is non-increasing,* v_n^- *is non-decreasing, and both sequences converge exponentially fast to* g*; namely, for all* $n \in \mathbb{Z}_+$,

$$-\alpha_{\mathfrak{J}}^{n-1} sp(e_0) \leq v_n^- - g \leq v_n^+ - g \leq \alpha_{\mathfrak{J}}^{n-1} sp(e_0). \qquad \square$$

Lemma 3 provides a uniform approximation to the optimal average reward g.

Theorem 5. *Suppose we select* $\ell \in \mathbb{N}$ *sufficiently large such that Assumption 4 is still valid and* $\kappa \gamma_{\mathfrak{m}}^{\ell} \leq \epsilon_1$. *Suppose the horizon* $n \in \mathbb{N}$ *is also sufficiently large, such that* $4\kappa \alpha_{\mathfrak{J}}^{n-1} \leq \epsilon_2$. *If we compute* $\overline{\mathsf{CDR}}_n^*(z, w)$ *using abstraction algorithm in Section 4.2 with error* ϵ_3, *then*

$$|\,\overline{\mathsf{CDR}}_n^*(z, w) - \overline{\mathsf{CDR}}_{n-1}^*(z, w) - g| \leq \epsilon_1 + \epsilon_2 + 2\epsilon_3,$$

which gives $\overline{\mathsf{CDR}}_n^*(z, w) - \overline{\mathsf{CDR}}_{n-1}^*(z, w)$ *as an approximation of* $g = \widehat{\overline{\mathsf{CDR}}}_{\infty}^*(s, y)$ *with error* $\epsilon_1 + \epsilon_2 + 2\epsilon_3$. $\square$

6. CASE STUDY

We apply our results to the model of the robot in Example 1. The state space $\mathcal{S} = [0, 4] \times [0, 9]$ is partitioned into three regions, $D_i = [0, 4] \times [3(i-1), 3i]$, $i = 1, 2, 3$, as depicted in Figure 1. The process noise is normally distributed with covariance matrix $\Sigma_{\mathsf{r}} = \mathrm{diag}(\sigma_1^2, \sigma_2^2)$. With this selection, the dynamics of the robot in each dimension are independent and the layout is symmetric. Therefore the solution of the problem should only depend on the dynamics along the vertical axis, i.e., the actions should be either up or down independent of the history of the robot's horizontal locations. This fact is confirmed by the simulations.

Since the set of valid inputs to the system is independent from the current state, Assumption 1 holds with $h_{\mathfrak{u}} = 0$. The input set is already discrete thus there is no need for discretization and so $\delta_{\mathfrak{u}} = 0$. As we discussed in Example 2, Assumption 2 holds with $h_{\mathfrak{T}} = 1$ and also $h_{\mathfrak{c}} = 1/\sigma_2\sqrt{2\pi}$. Therefore, the Lipschitz constants in Lemma 2 are $L_k = 3(n-k)h_{\mathfrak{c}}\kappa$ and the error grows quadratically with n as

$$\varepsilon = \kappa \left[\frac{n(n-1)}{2} 3h_{\mathfrak{c}}\delta_{\mathfrak{s}} + (n+1)\gamma_{\mathfrak{m}}^{\ell}\right].$$

The required memory usage and the computational complexity of the proposed approach depend on the size of the CMP $\mathfrak{M}$, i.e., the number of discrete inputs and states, and on the parameter ℓ for truncating the required history. Suppose state and input spaces of CMP $\mathfrak{S}$ has dimensions $d_{\mathfrak{s}}$ and $d_{\mathfrak{u}}$, respectively, and we take $n_{\mathfrak{s}}$ and $n_{\mathfrak{u}}$ partition sets along each dimension. Then the augmented MDP $\hat{\mathfrak{M}}$ has $n_{\mathfrak{s}}^{d_{\mathfrak{s}}} \ell^m$ discrete states with $n_{\mathfrak{u}}^{d_{\mathfrak{u}}}$ discrete actions, which are exponential is dimension of the process and in ℓ but polynomial is the required accuracy ε. This complexity can be reduced in the following ways. First, we do not need the whole state space $\hat{S}_{\mathfrak{d}} = S_{\mathfrak{d}} \times \mathbb{N}[1, \ell]^m$ but its subset that is reachable from $\xi_{\mathfrak{d}}(s, \mathbf{1}_m)$ for any s in the set of initial states of the CMP $\mathfrak{S}$. Second, the transition probability matrix of $\hat{\mathfrak{M}}$ is quite sparse, which enables us to utilize more efficient data structures to have a tradeoff between computational time and memory usage. Finally, adaptive girdding techniques proposed in [6] can also be used in this setting to reduce the required number of discrete states for a given accuracy.

The expected generated rewards $\lambda_1 = 5$ and $\lambda_3 = 7$ are chosen. The discounting factors are $\gamma_1 = 0.99 = \gamma_3 = 0.99$. For the hallway the expected generated reward is zero $\lambda_2 = 0$ and the discounting factor can be any quantity with no influence on the outcome: we set $\gamma_2 = 0.9$. We select $n_{\mathfrak{s}} = 45$ partition sets and $\ell = 6$. Standard deviation of the process noise is $\sigma_2 = 1.2$ and step size of the robot $\alpha_0 = 1.5$.

Figure 2 shows the approximate computation of $\mathsf{CDR}_n^*(s)$ as a function of initial state s and for different values of horizon n. As it is expected, these functions are piecewise continuous with continuity regions D_1, D_2, D_3. The difference $\mathsf{CDR}_n^*(s) - \mathsf{CDR}_{n-1}^*(s)$ converges to 10.30 after 30 iterations, which gives an approximation for $\mathsf{CDR}_{\infty}^*(s)$. Sample paths of the robot under the approximate optimal policy is shown in Figure 3 with the robot being initially in the hallway. Despite the robot being initially closer to region D_1, it decides to move up to visit D_3 since the expected value of the generated reward at D_3 is higher. After visiting D_3 the robot takes the action down to visit D_1. This clearly shows that the actions taken by the robot depend not only on its current location but also on the previously visited regions.

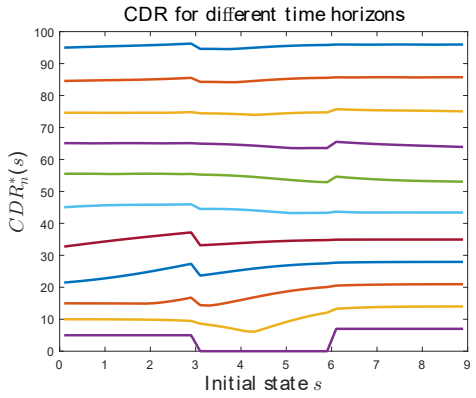

Figure 2: Approximate computation of CDR $\mathsf{CDR}_n^*(s)$ as a function of initial state s and for different values of horizon n in Example 1.

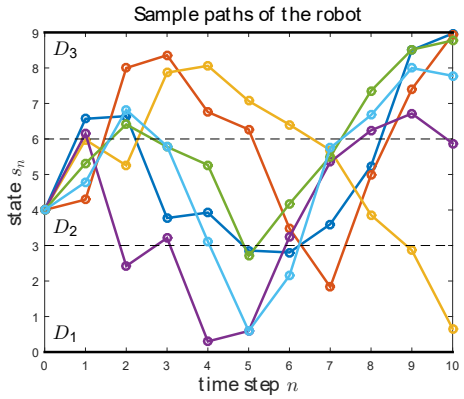

Figure 3: Sample paths of the robot in Example 1 as a function of time. The robot is initially in the hallway and moves towards the region with higher expected generated reward.

7. CONCLUDING REMARKS

We have proposed a mathematical model for optimizing rewards in dynamic environments, called *Reward Collecting Markov Processes*. Our model is motivated by request-serving applications, where a background process generates rewards whose values depend on the number of steps between generation and collection. We studied both the finite and infinite horizon synthesis problems for maximizing the collected reward. We characterized these problems as solutions to dynamic programs over an augmented hybrid space. We also provided a computational method for these problems with guaranteed error bounds based on abstracting the continuous-space problem into a discrete one.

8. REFERENCES

[1] A. Abate, J.-P. Katoen, J. Lygeros, and M. Prandini. Approximate model checking of stochastic hybrid systems. *European J. Control*, 6:624–641, 2010.

[2] D. Bertsekas and S. Shreve. *Stochastic Optimal Control: The Discrete-Time Case*. Athena Scientific, 1996.

[3] T. Brázdil, V. Brozek, K. Chatterjee, V. Forejt, and A. Kucera. Two views on multiple mean-payoff objectives in Markov decision processes. In *LICS*, pages 33–42, 2011.

[4] F. Dufour and T. Prieto-Rumeau. Approximation of Markov decision processes with general state space. *Journal of Mathematical Analysis and Applications*, 388(2):1254 – 1267, 2012.

[5] S. Esmaeil Zadeh Soudjani. *Formal Abstractions for Automated Verification and Synthesis of Stochastic Systems*. PhD thesis, Technische Universiteit Delft, The Netherlands, November 2014.

[6] S. Esmaeil Zadeh Soudjani and A. Abate. Adaptive and sequential gridding procedures for the abstraction and verification of stochastic processes. *SIAM Journal on Applied Dynamical Systems*, 12(2):921–956, 2013.

[7] S. Esmaeil Zadeh Soudjani, C. Gevaerts, and A. Abate. FAUST^2: Formal abstractions of uncountable-state stochastic processes. In *TACAS'15*, volume 9035 of *Lecture Notes in Computer Science*, pages 272–286. Springer, 2015.

[8] A. Gouberman and M. Siegle. Markov reward models and Markov decision processes in discrete and continuous time: Performance evaluation and optimization. In *Stochastic Model Checking. Rigorous Dependability Analysis Using Model Checking Techniques for Stochastic Systems: International Autumn School, ROCKS 2012*, pages 156–241. Springer, 2014.

[9] O. Hernández-Lerma. *Adaptive Markov control processes*. Applied mathematical sciences. Springer, New York, 1989.

[10] O. Hernández-Lerma and J. B. Lasserre. *Discrete-time Markov control processes*, volume 30 of *Applications of Mathematics*. Springer, 1996.

[11] A. Hinton, M. Kwiatkowska, G. Norman, and D. Parker. PRISM: A tool for automatic verification of probabilistic systems. In *TACAS*, volume 3920 of *Lecture Notes in Computer Science*, pages 441–444. Springer, 2006.

[12] M. L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., New York, NY, USA, 1st edition, 1994.

[13] O. Spanjaard and P. Weng. Markov decision processes with functional rewards. In *Multi-disciplinary Trends in Artificial Intelligence MIWAI 2013*, pages 269–280. Springer, 2013.

[14] S. Summers and J. Lygeros. Verification of discrete time stochastic hybrid systems: A stochastic reach-avoid decision problem. *Automatica*, 46(12):1951–1961, 2010.

[15] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.

[16] I. Tkachev and A. Abate. Characterization and computation of infinite-horizon specifications over Markov processes. *Theoretical Computer Science*, 515(0):1–18, 2014.

[17] I. Tkachev, A. Mereacre, J. Katoen, and A. Abate. Quantitative automata-based controller synthesis for non-autonomous stochastic hybrid systems. In *Hybrid Systems: Computation and Control*, pages 293–302. ACM, 2013.