Non-convex Policy Search Using Variational Inequalities

Yusen Zhan

yusen.zhan@wsu.edu

The School of Electrical Engineering and Computer Science

Washington State University, Pullman, WA 99163, USA

Haitham Bou Ammar

hb71@aub.edu.lb

Department of Computer Science

American University of Beirut, Lebanon

Matthew E. Taylor

taylorm@eecs.wsu.edu

The School of Electrical Engineering and Computer Science

Washington State University, Pullman, WA 99163, USA

Keywords: Reinforcement Learning, Policy Search, Non-Convex Constraints, Non-

Abstract

Policy search is a class of reinforcement learning algorithms for finding optimal policies in control problems with limited feedback. These methods have shown to be successful in high-dimensional problems, such as robotics control. Though successful, current methods can lead to unsafe policy parameters potentially damaging hardware units. Motivated by such constraints, projection based methods are proposed for safe policies.

These methods, however, can only handle convex policy constraints. In this paper, we propose the first safe policy search reinforcement learner capable of operating under *non-convex policy constraints*. This is achieved by observing, for the first time, a connection between non-convex variational inequalities and policy search problems. We provide two algorithms, i.e., Mann and two-step iteration, to solve the above problems and prove convergence in the non-convex stochastic setting. Finally, we demonstrate the performance of the above algorithms on six benchmark dynamical systems and show that our new method is capable of outperforming previous methods under a variety of settings.

1 Introduction

Policy search is a class of reinforcement learning algorithms for finding optimal control policies with no dynamical models. Such algorithms have shown numerous successes in handling high dimensional control problems, especially in the field of robotics (Kober and Peters, 2009). Despite these successes, their use has not been adopted in industrial and real-world applications. As detailed elsewhere (Thomas et al., 2013), one of the main drawbacks of policy search algorithms is their lack of safety guarantees. This can be traced back to the unconstrained nature of the optimization objective, which can lead to searching in regions where policies are known to be dangerous.

Researchers have attempted to remedy such problems of policy search before (Thomas et al., 2013). These approaches vary from methods in control theory (Harker and Pang, 1990) and stability analysis (Tobin, 1986), to methods in constrained optimization and proximal methods (Duchi et al., 2011; Thomas et al., 2013). Most of these approaches, however, can only deal with convex constraints, which can pose restrictions to real-world safety considerations. A robotic arm control has an intuitive justification of our motivation. Suppose we have multiple blocks in the environment and the arm cannot contact or attempt to move through the blocks. The learning problem is, therefore, to optimally move the end effector of the arm to its goal position without touching the blocks. This setting is also applicable if, instead of blocks, humans were in the robot's workspace. Thus, the positions of those obstacles may lead to a non-convex constraint set.

In this paper, we remedy the above problems by proposing the first policy search reinforcement learner which can handle *non-convex safety constraints*. Our approach leverages a novel connection between constrained policy search and non-convex variational inequalities under a special type of non-convex constraint set referred to as r-prox-regular. With this result, we propose adapting two algorithms (i.e., the Mann and two-step iteration method (Noor, 2009)), for solving the non-convex constrained policy search problem. We also generalize other deterministic convergence proofs of such methods to the stochastic setting and show the convergence results. In summary, the contributions of this paper are:

1. Establishing the connection between policy search with non-convex constraints and non-convex variational inequalities;

2. Proposing Mann and two-step iteration approaches to solve the policy search problem;

3. Proving convergence with probability 1 of Mann and two-step iteration under the stochastic setting; and

4. Demonstrating the effectiveness of the proposed methods in a set of experiments on six benchmark dynamical systems, in which our new technique succeeds where others fail, successfully handling non-convex constraint sets.

2 Background

This section briefly introduces reinforcement learning and the non-convex variational inequality problems.

2.1 Reinforcement Learning

In reinforcement learning (RL) an agent must sequentially select actions to maximize its total expected payoff. These problems are typically formalized as Markov decision processes (MDPs) $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$, where $\mathcal{A} \subseteq \mathbb{R}^d$ and $\mathcal{A} \subseteq \mathbb{R}^m$ denote the state and action spaces. $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$ represents the transition probability governing the dynamics of the system, $\mathcal{R} : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ is the reward function quantifying the performance of the agent and $\gamma \in [0, 1)$ is a discount factor specifying the degree to which rewards are discounted over time. At each time step t, the agent is in state $\boldsymbol{s}_t \in \mathcal{S}$ and must choose an action $\boldsymbol{a}_t \in \mathcal{A}$, transitioning it to a successor state $\boldsymbol{s}_{t+1} \sim p(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t, \boldsymbol{a}_t)$ as given by $\mathcal{P}$ and yielding a reward $r_{t+1} = \mathcal{R}(\boldsymbol{s}_t, \boldsymbol{a}_t)$. A policy $\pi : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$ is defined as a probability distribution over state-action pairs, where $\pi\left(\boldsymbol{a}_t|\boldsymbol{s}_t\right)$ denotes the probability of choosing action $\boldsymbol{a}_t$ in state $\boldsymbol{s}_t$.

Policy gradient algorithms (Sutton and Barto, 1998; Kober and Peters, 2009) are a type of reinforcement learning algorithm that has shown successes in solving complex robotic problems. Such methods represent the policy $\pi_{\boldsymbol{\theta}}(\boldsymbol{s}_t|\boldsymbol{a}_t)$ by an unknown vector of parameters $\boldsymbol{\theta} \in \mathbb{R}^d$. The goal is to determine the optimal parameters $\boldsymbol{\theta}^{\star}$ that maximize the expected average payoff:

$$\mathcal{J}(\boldsymbol{\theta}) = \int_{\boldsymbol{\tau}} p_{\boldsymbol{\theta}}(\boldsymbol{\tau}) \mathfrak{R}(\boldsymbol{\tau}) d\boldsymbol{\tau}, \tag{1}$$

where $\boldsymbol{\tau} = [\boldsymbol{s}_{0:T}, \boldsymbol{a}_{0:T}]$ denotes a trajectory over a possibly infinite horizon T. The probability of acquiring a trajectory, $p_{\boldsymbol{\theta}}(\boldsymbol{\tau})$, under the policy parameterization $\pi_{\boldsymbol{\theta}}(\cdot)$ and average per-time-step return $\mathfrak{R}(\boldsymbol{\tau})$ are given by:

$$\begin{aligned} p_{\boldsymbol{\theta}}(\boldsymbol{\tau}) &= p_0(\boldsymbol{s}_0) \prod_{m=0}^{T-1} p\left(\boldsymbol{s}_{m+1}|\boldsymbol{s}_m, \boldsymbol{a}_m\right) \pi_{\boldsymbol{\theta}}(\boldsymbol{a}_m|\boldsymbol{s}_m) \\ \mathfrak{R}(\boldsymbol{\tau}) &= \frac{1}{T} \sum_{m=0}^{T} r_{m+1}, \end{aligned}$$

with an initial state distribution p_0.

Policy gradient methods, such as episodic REINFORCE (Williams, 1992), PoWER (Kober and Peters, 2009), and Natural Actor Critic (Bhatnagar et al., 2009; Peters and Schaal, 2008), typically employ a lower-bound on the expected return $\mathcal{J}(\boldsymbol{\theta})$ for fitting the unknown policy parameters $\boldsymbol{\theta}$. To achieve this, such algorithms generate trajectories using the current policy $\boldsymbol{\theta}$, and then compare performance with a new parameterization $\tilde{\boldsymbol{\theta}}$. As detailed in Kober and Peters (2009), the lower bound on the expected return can be calculated by using Jensen's inequality and the concavity of the logarithm:

$$\log \mathcal{J}\left(\tilde{\boldsymbol{\theta}}\right) = \log \int_{\boldsymbol{\tau}} p_{\tilde{\boldsymbol{\theta}}}(\boldsymbol{\tau})\mathfrak{R}(\boldsymbol{\tau})d\boldsymbol{\tau} \geq \int_{\boldsymbol{\tau}} p_{\boldsymbol{\theta}}(\boldsymbol{\tau})\mathfrak{R}(\boldsymbol{\tau}) \log \frac{p_{\tilde{\boldsymbol{\theta}}}(\boldsymbol{\tau})}{p_{\boldsymbol{\theta}}(\boldsymbol{\tau})} d\boldsymbol{\tau} + \text{constant}$$

$$\propto -\text{KL}\left(p_{\boldsymbol{\theta}}(\boldsymbol{\tau})\mathfrak{R}(\boldsymbol{\tau}) || p_{\tilde{\boldsymbol{\theta}}}(\boldsymbol{\tau})\right) = \mathcal{J}_{\mathcal{L},\boldsymbol{\theta}}\left(\tilde{\boldsymbol{\theta}}\right),$$

where $\text{KL}(p(\boldsymbol{\tau})||q(\boldsymbol{\tau})) = \int_{\boldsymbol{\tau}} p(\boldsymbol{\tau}) \log \frac{p(\boldsymbol{\tau})}{q(\boldsymbol{\tau})} d\boldsymbol{\tau}$.

2.2 Non-Convex Variational Inequality

In this section, we introduce the problem of non-convex variational inequalities which will be used later for improving policy search reinforcement learning. We let $\mathbb{H}$ denote a Hilbert space and $\langle \cdot, \cdot \rangle$ the inner product in $\mathbb{H}$. Given the above, we next define the projection, $\text{Proj}_{\mathcal{K}}[\boldsymbol{u}]$, of a vector $\boldsymbol{u} \in \mathbb{H}$ to a set $\mathcal{K}$ as that vector $\boldsymbol{v}$ acquiring the closest distance to $\boldsymbol{u} \in \mathcal{K}$. Formally:

Definition 1. *The set of projections of $\boldsymbol{u} \in \mathbb{H}$ onto set $\mathcal{K}$ is defined by*

$$\mathit{Proj}_{\mathcal{K}}[\boldsymbol{u}] = \left\{ \boldsymbol{v} \in \mathcal{K} \mid d_{\mathcal{K}}(\boldsymbol{u}) = \inf_{\boldsymbol{v} \in \mathcal{K}} ||\boldsymbol{u} - \boldsymbol{v}|| \right\},$$

where $||\cdot||$ denotes the norm.

Without loss generality, we use $||\cdot||$ to denote general norm in Hilbert space. One can easily use Euclidean norm $||\cdot||_2$ to replace it. To formalize safe policies, we consider specific forms of non-convex sets, referred to as uniformly prox-regular, previously introduced elsewhere (Clarke et al., 2008; Federer, 1959; Poliquin et al., 2000). For their definition, however, the proximal normal cone first needs to be introduced.

Definition 2. *The proximal normal cone of a set $\mathcal{K}$ at a point $\boldsymbol{u} \in \mathcal{K}$ is defined as:*

$$N_{\mathcal{K}}^{P}(\boldsymbol{u}) = \{\boldsymbol{z} \in \mathbb{H} \mid \exists \alpha > 0 \textit{ such that } \boldsymbol{u} \in \textit{Proj}_{\mathcal{K}}[\boldsymbol{u} + \alpha \boldsymbol{z}]\}.$$

With the above definition, we now define the uniformly prox-regular set:

Definition 3. *For a given $r \in (0, \infty]$, a subset $\mathcal{K}_r$ of $\mathbb{H}$ is normalized uniformly r-prox-regular if and only if every nonzero proximal normal to K_r can be realized by an r-ball Equivalently, $\forall \boldsymbol{u} \in \mathcal{K}_r$ and $0 \neq \boldsymbol{z} \in N_{\mathcal{K}_r}^{P}(\boldsymbol{u})$, we have*

$$\left\langle \frac{\boldsymbol{z}}{\|\boldsymbol{z}\|}, \boldsymbol{v} - \boldsymbol{u} \right\rangle \leq \frac{1}{2r} \|\boldsymbol{v} - \boldsymbol{u}\|^2, \qquad \forall \boldsymbol{v} \in K_r. \tag{2}$$

Please note that such a class of normalized uniformly r-prox-regular sets is broad as it includes convex sets, p-convex sets, $C^{1,1}$, sub-manifolds of $\mathbb{H}$ and several other non-convex sets (Clarke et al., 2008; Federer, 1959; Poliquin et al., 2000).[1]

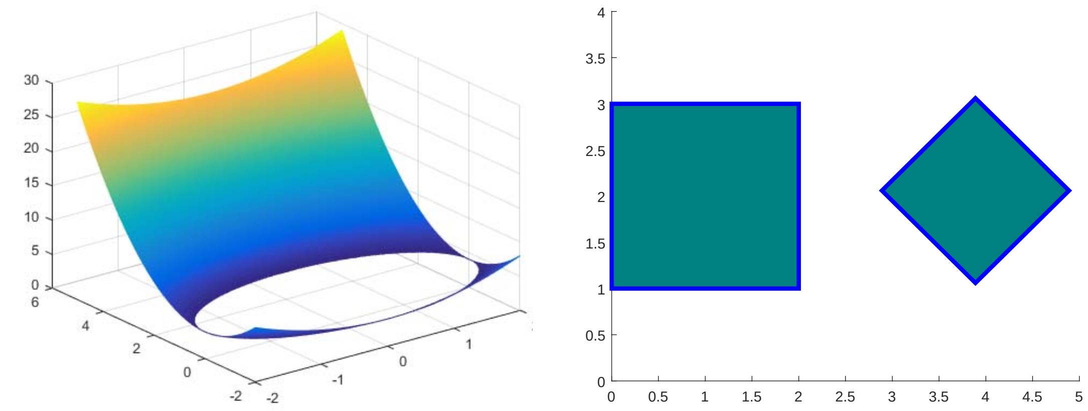
To illustrate, we provide some examples of r-prox-regular sets.

Example 1. *Let $x, y \in \mathbb{R}$ be real numbers, then we have*

$$\mathcal{K} = \left\{x^2 + (y-2)^2 \geq 4 \mid -2 \leq x \leq 2, y \geq -2\right\},$$

which is a subset of the Euclidean plane and a r-prox-regular set $\mathcal{K}_r$.

Example 2. *Let $\mathcal{K}$ be the union of two disjoint squares in a plane, A and B with vertexes at the points $(0,1)$, $(2,1)$, $(2,3)$, $(0,3)$ and at the points $(4,1)$, $(5,2)$, $(4,3)$, $(3,2)$, respectively. Clearly, $\mathcal{K}$ is a r-prox-regular set in $\mathbb{R}^2$.*



(a) Example 1

(b) Example 2

Figure 1: Figure 1a shows the set in 3D and Figure 1b illustrates the set in the Euclidean plane.

To help the readers understand, we provide the graphical examples in Figure 1. It is worth mentioning that normalized uniformly r-prox-regularity adheres to the following properties:

Proposition 1 (Poliquin et al. (2000))**.** *Let $r > 0$ and $\mathcal{K}_r$ be a nonempty closed and r-prox-regular subset of $\mathbb{H}$, then:*

- $\forall \boldsymbol{u} \in K_r$, *Proj*$_{\mathcal{K}_r}[\boldsymbol{u}] \neq \emptyset$,
- $\forall r' \in (0, r)$, *Proj*$_{\mathcal{K}_r}$ *is Lipschitz continuous with constant* $\frac{r}{r-r'}$ *on* $\mathcal{K}_{r'}$*, and*

[1] We will use r-prox-regular sets as the short name for uniformly r-prox-regular sets.

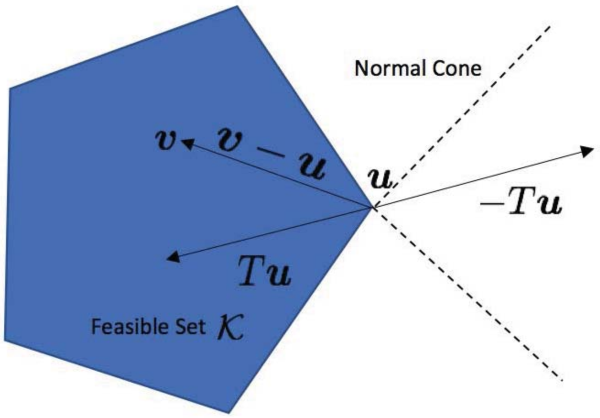


Figure 2: The geometric interpretation of VI is that we try to find a point $\boldsymbol{u}$ in the constraint set $\mathcal{K}$ such that the inner product of $T\boldsymbol{u}$ and $\boldsymbol{v}-\boldsymbol{u}$ is greater or equal to 0 for all $v \in \mathcal{K}$. In the standard VI setting, the constraint set $\mathcal{K}$ must be convex. However, if we replace the convex constraint set $\mathcal{K}$ with r-prox regular set $\mathcal{K}_r$, which is non-convex, it yields the non-convex VI problem. Therefore, the key difference between standard VI and non-convex VI is the constraint set.

- *the proximal normal cone is closed as a set-value mapping.*

Given the above, we now formally present non-convex variational inequalities (Noor, 2009), accompanied with operator properties, as needed for the rest of this paper. For a given nonlinear operator, $\boldsymbol{T}$, the *non-convex variational inequality* problem, $\mathcal{NVI}(\boldsymbol{T}, \mathcal{K}_r)$, is to determine a vector $\boldsymbol{u} \in \mathcal{K}_r$ such that:

$$\langle \boldsymbol{Tu}, \boldsymbol{v}-\boldsymbol{u} \rangle \geq 0, \qquad \boldsymbol{v} \in \mathcal{K}_r \tag{3}$$

To elaborate the standard VI and non-convex VI, we provide the geometrical interpretation in Figure 2. Finally, in our proofs, we require two important properties for operators:

Definition 4. *A nonlinear operator, $\boldsymbol{T} : \mathbb{H} \rightarrow \mathbb{H}$, is said to be strongly monotone if and only if there exists a constant $\alpha > 0$ such that:*

$$\langle \boldsymbol{Tu} - \boldsymbol{Tv}, \boldsymbol{u} - \boldsymbol{v} \rangle \geq \alpha \left|\left|\boldsymbol{u} - \boldsymbol{v}\right|\right|^2, \ \forall \boldsymbol{u}, \boldsymbol{v} \in \mathbb{H}.$$

Definition 5. *A nonlinear operator, $\boldsymbol{T} : \mathbb{H} \rightarrow \mathbb{H}$, is said to be Lipschitz continuous if and only if there exists a constant $\beta > 0$ such that:*

$$\left|\left|\boldsymbol{Tu} - \boldsymbol{Tv}\right|\right| \leq \beta \left|\left|\boldsymbol{u} - \boldsymbol{v}\right|\right|, \ \forall \boldsymbol{u}, \boldsymbol{v} \in \mathbb{H}.$$

3 Non-convex Constraints and Policy Search

Much of the work in reinforcement learning has focused on exploiting convexity to determine solutions to the sequential decision-making problem (Kaelbling et al., 1996). In reality, however, it is easy to construct examples in which the overall optimization problem is non-convex. One such example is the usage of complex constraints on the state-space and/or policy parameters. Examples of such a need arise in a variety of applications including bionics, medicine, energy storage problems, and others. One relative approach is projected natural actor-critic (PNAC) from Thomas et al. (2013). In PNAC, the authors incorporate parameter constraints to policy search and solve the corresponding optimization problem. They show that using their method, algorithms are capable of adhering to safe policy constraints. Our method is similar in spirit to that of PNAC but has some crucial differences. First, PNAC can only handle convex parameter constraints. Second, PNAC provides only closed form projections under relatively restrictive assumptions of the policy sets considered. In this paper, we generalize

PNAC-like methods to the non-convex setting and then show that such projections can be learned using the branch and bound method from non-convex programming.

To achieve this, we first map policy gradients to non-convex variational inequalities and then show that by solving the latter structure, we can acquire a solution to the original objective. Given the above mapping, we then present two algorithms capable of acquiring a solution of the above problem.

3.1 The Mapping to Non-convex Variational Inequalities

Considering the original objective in Equation (1), it is clear that the goal of a policy gradients agent is to determine a local optimum $\boldsymbol{\theta}^{\star}$ which maximizes $\mathcal{J}(\boldsymbol{\theta})$. Equivalently, the goal is to determine $\boldsymbol{\theta}^{\star}$ such that $\nabla_{\boldsymbol{\theta}}\mathcal{J}(\boldsymbol{\theta})\Big|_{\boldsymbol{\theta}^{\star}} = 0$. In the following lemma, we show that the original problem in Equation (1) can be reduced to a non-convex variational inequality with $\mathcal{K}_r = \mathbb{R}^d$:

Lemma 1. *The policy gradients problem in Equation (1) can be reduced to a non-convex variational inequality of the form:* $\mathcal{NVI}\left(\boldsymbol{T}, \mathcal{K}_r\right)$ *with* $\boldsymbol{T} = \nabla_{\boldsymbol{\theta}}\mathcal{J}(\boldsymbol{\theta})$ *and* $\mathcal{K}_r = \mathbb{R}^d$.

Proof. For solving the policy gradients problem, we have:

$$\nabla_{\boldsymbol{\theta}}\mathcal{J}(\boldsymbol{\theta})\Big|_{\boldsymbol{\theta}^{\star}} = 0.$$

A vector $\boldsymbol{u}^{\star}$ is a solution to an $\mathcal{NVI}(\boldsymbol{T}, \mathcal{K}_r)$ if and only if $\boldsymbol{T}\boldsymbol{u}^{\star} = 0$. This is easy to show since if $\boldsymbol{T}\boldsymbol{u}^{\star} = 0$, then $\langle \boldsymbol{T}\boldsymbol{u}, \boldsymbol{v} - \boldsymbol{u}\rangle = 0$ for any vector $\boldsymbol{v}$. Conversely, if $\boldsymbol{u}^{\star}$ solves $\langle \boldsymbol{T}\boldsymbol{u}, \boldsymbol{v} - \boldsymbol{u}\rangle \geq 0$ for any $\boldsymbol{v} \in \mathcal{K}_r$, then by letting $\boldsymbol{v} = \boldsymbol{u}^{\star} - \boldsymbol{T}\boldsymbol{u}^{\star}$, we have:

$$\langle \boldsymbol{T}\boldsymbol{u}^{\star}, -\boldsymbol{T}\boldsymbol{u}\star\rangle \geq 0 \implies -\left|\left|\boldsymbol{T}\boldsymbol{u}^{\star}\right|\right|^2 \geq 0.$$

Consequently, $\boldsymbol{T}\boldsymbol{u}^\star = 0$. Hence, choosing $\boldsymbol{T} = \nabla_{\boldsymbol{\theta}}\mathcal{J}(\boldsymbol{\theta})$ and $\mathcal{K}_r = \mathbb{R}^d$, we can derive:

$$\boldsymbol{T}\boldsymbol{\theta}^\star = \nabla_{\boldsymbol{\theta}}\mathcal{J}(\boldsymbol{\theta})\Big|_{\boldsymbol{\theta}^\star} = 0,$$

if and only if $\boldsymbol{\theta}^\star$ solves the non-convex variational inequality $\mathcal{NVI}\left(\nabla_{\boldsymbol{\theta}}\mathcal{J}(\boldsymbol{\theta}), \mathbb{R}^d\right)$. □

Given the results in Lemma 1, we can now quantify the conditions under which a solution to the non-convex variational inequality (and equivalently, the policy search problem) can be attained. For ensuring the existence of a solution as well as informative policy parameters parameter $\boldsymbol{\theta}$, we start by introducing a smooth regularization into the policy gradient's objective function:

$$-\log \int_{\boldsymbol{\tau}} p_{\tilde{\boldsymbol{\theta}}}(\boldsymbol{\tau})\mathfrak{R}(\boldsymbol{\tau})d\boldsymbol{\tau} + \mu \left\|\boldsymbol{\theta}\right\|_2^2, \tag{4}$$

where $\mu > 0$ is a constant. Typical policy gradient methods (Kober and Peters, 2009) maximize a lower bound to the expected reward, which can be obtained by using Jensen's inequality:

$$\begin{aligned}\log \int_{\boldsymbol{\tau}} p_{\tilde{\boldsymbol{\theta}}}(\boldsymbol{\tau})\mathfrak{R}(\boldsymbol{\tau})d\boldsymbol{\tau} &= \log \sum_{m=1}^{M} p_{\tilde{\boldsymbol{\theta}}}(\boldsymbol{\tau})\mathfrak{R}(\boldsymbol{\tau}) \\ &\geq \log\left[M\right] + \mathbb{E}\left[\sum_{m=0}^{T-1} \log\left[\pi_{\boldsymbol{\theta}}\left(\boldsymbol{a}_m^k \mid \boldsymbol{s}_m^k\right)\right]\right]_{k=1}^{M} + \text{constant}\end{aligned}$$

where M is the number of trajectories and T is the number of steps in each trajectory. Therefore, our goal is to minimize the following objective:[2]

$$\mathcal{J}(\boldsymbol{\theta}) = -\sum_{k=1}^{M}\sum_{m=0}^{T-1} \log\left[\pi_{\boldsymbol{\theta}}\left(\boldsymbol{a}_m^k \mid \boldsymbol{s}_m^k\right)\right] + \mu \left\|\boldsymbol{\theta}\right\|_2^2. \tag{5}$$

[2] Please note that in the above equation we absorbed the reward in the normalization terms of the policy.

Algorithm 1 Mann Iteration for Policy Search

```
Input: $\rho > 0$, $\alpha_t \in [0, 1]$
for $t = 1, \dots, M$ do
    Update $\boldsymbol{\theta}_{t+1} = (1 - \alpha_t)\boldsymbol{\theta_t} + \alpha_t \mathrm{Proj}_{\mathcal{K}_r}\left[\boldsymbol{\theta}_t - \rho \nabla_{\boldsymbol{\theta}} \mathcal{J}(\boldsymbol{\theta})\Big|_{\boldsymbol{\theta}_t}\right]$
end for
```

Algorithm 2 Two-step Iteration for Policy Search

```
Input: $\rho > 0$, $\alpha_t, \beta_t \in [0, 1]$
for $t = 1, \dots, M$ do
    Compute $\boldsymbol{\mu}_t = (1 - \beta_t)\boldsymbol{\theta_t} + \beta_t \mathrm{Proj}_{\mathcal{K}_r}\left[\boldsymbol{\theta}_t - \rho \nabla_{\boldsymbol{\theta}} \mathcal{J}(\boldsymbol{\theta})\Big|_{\boldsymbol{\theta}_t}\right]$
    Update $\boldsymbol{\theta}_{t+1} = (1 - \alpha_t)\boldsymbol{\theta_t} + \alpha_t \mathrm{Proj}_{\mathcal{K}_r}\left[\boldsymbol{\mu}_t - \rho \nabla_{\boldsymbol{\mu}} \mathcal{J}(\boldsymbol{\mu})\Big|_{\boldsymbol{\mu}_t}\right]$
end for
```

Given the above variational inequality, we next adapt two algorithms, the Mann iteration method and the two-step iteration method, for determining an optimum. The Mann iteration method is a one-step method in which we update the parameter $\boldsymbol{\theta}$ once. In each update t, the algorithm collects a trajectory with n steps and computes the gradient $\nabla_{\boldsymbol{\theta}} \mathcal{J}(\boldsymbol{\theta})\Big|_{\boldsymbol{\theta}_t}$. Then, $\boldsymbol{\theta}_{t+1}$ leverages between $\boldsymbol{\theta}_t$ and $\mathrm{Proj}_{\mathcal{K}_r}\left[\boldsymbol{\theta}_t - \rho \nabla_{\boldsymbol{\theta}} \mathcal{J}(\boldsymbol{\theta})\Big|_{\boldsymbol{\theta}_t}\right]$ by a smooth constant α_t. Similarly, the two-step iteration method computes an auxiliary parameter $\boldsymbol{\mu}$ at first and then updates the parameter $\boldsymbol{\theta}$ given $\boldsymbol{\mu}$. The update rules are as same as the Mann iteration method (see Algorithms 1 and 2 for details).

3.2 Construction of The Projection $\mathbf{Proj}_{\mathcal{K}_r}[\boldsymbol{x}]$

In both algorithms, the projection $\mathrm{Proj}_{\mathcal{K}_r}[\boldsymbol{x}]$ to the r-prox-regular set plays a crucial role as it determines a valid solution to the policy search problem within the non-convex set. Such a projection can be interpreted as the solution to the following optimization problem:

$$\begin{aligned} &\min \|\boldsymbol{y} - \boldsymbol{x}\|^2 \\ &\text{s.t.} \quad \boldsymbol{y} \in \mathcal{K}_r. \end{aligned} \tag{6}$$

Notice that if $\mathcal{K}_r$ is convex, any convex programming algorithm can be adapted for determining a solution (Boyd and Vandenberghe, 2004). In fact, the algorithm used in PNAC can be seen as a special case of our method under convex policy constraints, where a closed form solution can be determined. In our setting the r-prox-regular set, $\mathcal{K}_r$, can be non-convex (making the problem more difficult). Fortunately, the projection onto a r-prox-regular set $\mathcal{K}_r$ exists and can be written as:

$$\text{Proj}_{\mathcal{K}_r}[\boldsymbol{x}] = (I + N^P_{\mathcal{K}_r})^{-1}(\boldsymbol{x}) \quad \boldsymbol{x} \in \mathbb{H},$$

where $I(\boldsymbol{x})$ is the identity operation, $N^P_{\mathcal{K}_r}(\boldsymbol{x})$ is the proximal normal cone of $\mathcal{K}_r$ at $\boldsymbol{x}$ and $+$ denotes Minkowski's addition (Noor, 2009; Poliquin et al., 2000).

To construct such projections, the method introduced by Thomas et al. (2013), which maps the computations to a quadratic program, becomes inapplicable due to the non-convexity of our problem.

We instead use non-convex programming methods such as the branch and bound method, which are capable of solving non-convex programming problem with optimality guarantees (Hendrix et al., 2010). The basic idea of branch and bound is to recursively decompose the problem into disjoint subproblems until the solution is found. The method prunes some subproblems, which do not contain the solution, to reduce the search space. Therefore, branch and bound is guaranteed to find the optimal solution even if the constraints are non-convex. Next, we provide the general way to construct the projection based on Algorithm 30 in Hendrix et al. (2010). Algorithm 3 shows the details of our whole method. It generates an approximate solution with error less than δ. This method starts with a set X_1, which contains the r-prox-regular set $\mathcal{K}_r$, and a list L. At each iteration, it removes a subset X from L, and expands it into k disjoint subsets $X_{r+1}, \ldots, X_{r+k}$, with corresponding lower bounds $g^L_{r+1}, \ldots, g^L_{r+k}$. Then, recalculate a upper bound $g^U = g_i$ in feasible set $X_i \cap \mathcal{K}_r$. According to this upper bound, delete all subsets X_j from L such that $g^L_j > g^U$. This is called bounding (pruning). If the lower bound g_i is close to the upper bound g^U, then set the y_i as an approximate solution of the problem (as defined in Equation (6)). $Size(\cdot)$ is a pre-defined function to determine the size of the subset. If the subset becomes smaller than the constant ε, abandon it.

Therefore, Algorithm 3 implements the projection operation $\text{Proj}_{\mathcal{K}_r}[\boldsymbol{x}]$ in Algorithm 1 and Algorithm 2. Although we provide a general method to implement the projection onto the non-convex set above, it is still possible to construct the projection in an easy way under some special circumstances, which we will show in the experimental section.

Computational Complexity Although the branch and bound algorithm solves the projection $\text{Proj}_{\mathcal{K}_r}[\boldsymbol{x}]$ problem, the computational cost is exponential in the worst case. For the Algorithm 1 and Algorithm 2, the primary cost is computing the gradient $\nabla_{\boldsymbol{\theta}} \mathcal{J}(\boldsymbol{\theta})$, which depends on the length of the trajectories T. Consider the M iterations, the computation complexity is $O(TM)$.

Algorithm 3 Branch and Bound Algorithm for $\mathrm{Proj}_{\mathcal{K}_r}$

```
Input: Two constants $\delta, \epsilon$, a $r$-prox-regular set $\mathcal{K}_r$ and a set $X_1$ enclose the set $\mathcal{K}_r$
Compute a lower bound $g_1^L$ on $X_1$ and a feasible point $y_1 \in X_1 \cap \mathcal{K}_r$
if no feasible point then
    STOP
else
    Set upper bound $g^U = \|y_1 - x\|^2$; Put $X_i$ in a list $L$; counter $r = 1$
end if
while $L \neq \emptyset$ do
    Remove a subset from $X$ from $L$ and expand it into $k$ disjoint subsets $X_{r+1}, \dots, X_{r+k}$; Compute the corresponding lower bounds $g_{r+1}^L, \dots, g_{r+k}^L$
    for $i = r + 1, \dots, r + k$ do
        if $X_i \cap \mathcal{K}_r$ has no feasible point then
            $g_i^L = \infty$
        end if
        if $g_i^L < g^U$ then
            Determine a feasible point $y_i$ and $g_i = \|y_i - x\|^2$
            if $g_i < g^U$ then
                $g^U = g_i$
                Eliminate all $X_j$ from $L$ such hat $g_j^L > g^U$ {pruning}
                Continue
            end if
            if $g_i^L > g^U - \delta$ then
                $g^U = g_i$; Save $y_i$ as an approximation solution
            else if $Size(X_i) \geq \epsilon$ then
                store $X_i$ in $L$
            end if
        end if
    end for
    $r = r + k$
end while
```

3.3 Comments for the Extragradient Method

This section is devoted to providing some discussion regarding the extragradient method proposed by Noor (2009) and we point out the algorithm is flawed.

Noor does not provide the convergence proofs of extragradient method in (Noor, 2009) and only cites prior work (Noor, 2004) as a reference. After thoroughly reading the reference paper, we do not find the corresponding proofs. Fortunately, Noor *et. al.* indeed show the proofs of extragradient methods under the non-convex VI setting (Noor et al., 2011).

After a thorough investigation of their proofs, we identified a key mistake. In the proofs of the Theorem 3.1 (Noor et al., 2011), they claim that

$$\langle \rho T \boldsymbol{u}_{t+1} + \boldsymbol{u}_{t+1} - \boldsymbol{u}_t, \boldsymbol{u} - \boldsymbol{u}_{t+1} \rangle \geq 0, \tag{7}$$

where $\boldsymbol{u}_{t+1}$ is the solution of extragradient algorithm at $t+1$ step and $\boldsymbol{u} \in \mathcal{K}_r$ is the solution of the non-convex VI problem. However, Equation (7) only holds if $\mathcal{K}_r$ is convex (Khobotov, 1987). It does not hold when $\mathcal{K}_r$ is non-convex, indicating that the proofs are incorrect.

When trying to amend the proofs, we discovered that the extragradient method actually solves another problem instead of the non-convex VI problem in Equation (3). We also discover that researchers point out that some results in Noor's work are incorrect. See Ansari and Balooee (2013) for details.

4 Convergence Results

In this section, we show, with probability 1, convergence under the r-prox-regular non-convex sets for both Algorithms 1 and 2. Our proof strategy can be summarized in the following steps.

Proof Strategy: The first step is to prove that Lipschitz and monotone properties of the policy search loss function hold under the broad class of log-concave policy distributions. This guarantees the existence of a solution for the non-convex variational inequality problem. The next step is to make use of the supermartingale convergence theorem. Finally, we show that the change between policy parameter updates under r-prox-regular sets abides by the supermartingale properties and thus show convergence in expectation with probability 1.

We have show convergence results, with probability 1, for Algorithms 1 and 2.

Theorem 1. *If the gradient $\nabla_{\boldsymbol{\theta}} \mathcal{J}(\boldsymbol{\theta})$ satisfies the log-concave distribution assumption, $\alpha_t \in [0, 1]$, $\sum_t \alpha_t = \infty, t \geq 0$, then Algorithm 1 converges to the solution of $\mathcal{NVI}(\nabla_{\boldsymbol{\theta}} J'_{\boldsymbol{\theta}}, \mathcal{K}_r)$ with probability* 1.

Theorem 2. *If the gradient $\nabla_{\boldsymbol{\theta}} \mathcal{J}_{\boldsymbol{\theta}}$ satisfies the log-concave distribution assumption, $\alpha_t, \beta_t \in [0, 1]$, $\sum_t \alpha_t = \infty$ and $\sum_t \beta_t = \infty, t \geq 0$, then Algorithm 2 converges to the solution of $\mathcal{NVI}(\nabla_{\boldsymbol{\theta}} J'_{\boldsymbol{\theta}}, \mathcal{K}_r)$ with probability* 1.

We provide the full proofs in the Appendix, following the proof strategy.

5 Experimental Results

To empirically validate the performance of our methods, we applied the Mann and two-step iteration algorithms to control a variety of benchmark dynamical systems, including cart pole (CP), double inverted pendulum (DIP), bicycle (BK), simple mass (SM), robotic arm (RA) and double mass (DM). These systems have been previously introduced (Bou Ammar et al., 2015; Sutton and Barto, 1998).

Cart Pole: The cart pole system is described by the cart's mass m_c in kg, the pole's mass m_p in kg and the pole's length l in meters. The state is given by the cart's position and velocity v, as well as the pole's angle θ and angular velocity $\dot{\theta}$. The goal is to train a policy that controls the pole in an upright position.

Double Inverted Pendulum: The double inverted pendulum (DIP) is an extension of the cart pole system. It has one cart m_0 in kg and two poles in which the corresponding lengths are l_1 and l_2 in meters. We assume the poles have no mass and that there are two masses m_1 and m_2 in kg on the top of each pole. The state consists of the cart's position x_1 and velocity v_1, the lower pole's angle θ_1 and angular velocity $\dot{\theta}_1$, as well as the upper pole's angle, θ_2, and angular velocity $\dot{\theta}_2$. The goal is also to learn a policy to control the two poles in a specific state.

Bicycle: The bicycle model assumes a fixed rider and is characterized by eight parameters. The goal is to keep the bike balanced as it rolls along the horizontal plane.

Simple Mass: The simple mass (SM) system is characterized by the spring constant k in N/m, the damping constant d in Ns/m and the mass m in kg. The system's state is given by the position x and the velocity v of the mass. The goal is to train a policy for guiding the mass to a specific state.

Double Mass: The double mass (DM) is an extension of the simple mass system. It has two masses m_1, m_2 in kg and two springs in which the corresponding springs constants are given by k_1 and k_2 in N/m, as well as the damping constant d_1 and d_2 in Ns/m. The state consists of the big mass's position x_1 and velocity v_1, as well as the small mass's position x_2 and velocity v_2. The goal is also to learn a policy to control the two mass in a specific state.

Robotic Arm: The robotic arm (RA) is a system of two arms connected by a joint. The upper arm has weight m_1 in Kg and length l_1 in meters, and the lower arm has weight m_2 in Kg and length l_2 in meter. The state consists of the upper arm's position x_1 and angle θ_1, as well as the lower arm' position x_1 and velocity θ_2. The goal is also to learn a policy to control the two mass in a specific state.

For reproducibility, we summarize the parameters that we used for each experiment in Table 1. The reward $r_t = -\sqrt{\|x_t - x_g\|^2}$ was set similarly in all experiments where x_t is the current state at time t and x_g is the goal state.

Table 1: Parameter ranges used in the experiments

CP	DIP	BK
$m_c, m_p \in [0, 1]$ $l \in [0.2, 0.8]$	$m_0 \in [1.5, 3.5]$ $m_1, m_2 \in [0.055, 0.1]$ $l_1 \in [0.4, 0.8], l_2 \in [0.5, 0.9]$	$m \in [10, 14], a \in [0.2, 0.6]$ $h \in [0.4, 0.8], b \in [0.4, 0.8]$ $c \in [0.1, 1], \lambda \in [\pi/3, 8\pi/18]$
SM	RA	DM
$m \in [3, 5]$ $k \in [1, 7]$	$l_1, l_2 \in [3, 5]$ $m_1, m_2 \in [0, 1]$	$m_1 \in [1, 7], m_2 \in [1, 10]$ $k_1 \in [1, 5], k_2 \in [1, 7]$ $d_1, d_2 \in [0.01, 0.1]$

5.1 Experimental Protocol

We generate 10 tasks for each domain by varying the system parameters (Table 1) to ensure the diversity of the domain tasks and optimal policies. We run each task for a total of 1000 iterations. At each iteration, the learner used its policy to generate 50 trajectories of 150 steps and updated its policy. We used eNAC (Peters and Schaal, 2008), a standard PG algorithm, as the base learner.

We compare our Mann iteration and two-step iteration algorithms to Projected Natural Actor-Critic (PNAC) (Thomas et al., 2013), which suffers when handling non-convex constraints. We also provide other parameters for our algorithms: $\alpha_t = 0.8$, $\beta_t = 0.8$ and $\rho = 0.9$. Our constraint set was defined as follows:

$$\begin{aligned} \|\boldsymbol{\theta}\|_2^2 &\geq k \\ \theta(1)^2 + \theta(2)^2 + \cdots + \theta(n-1)^2 &\leq k \\ \theta(n) &\geq -\sqrt{k}, \end{aligned}$$

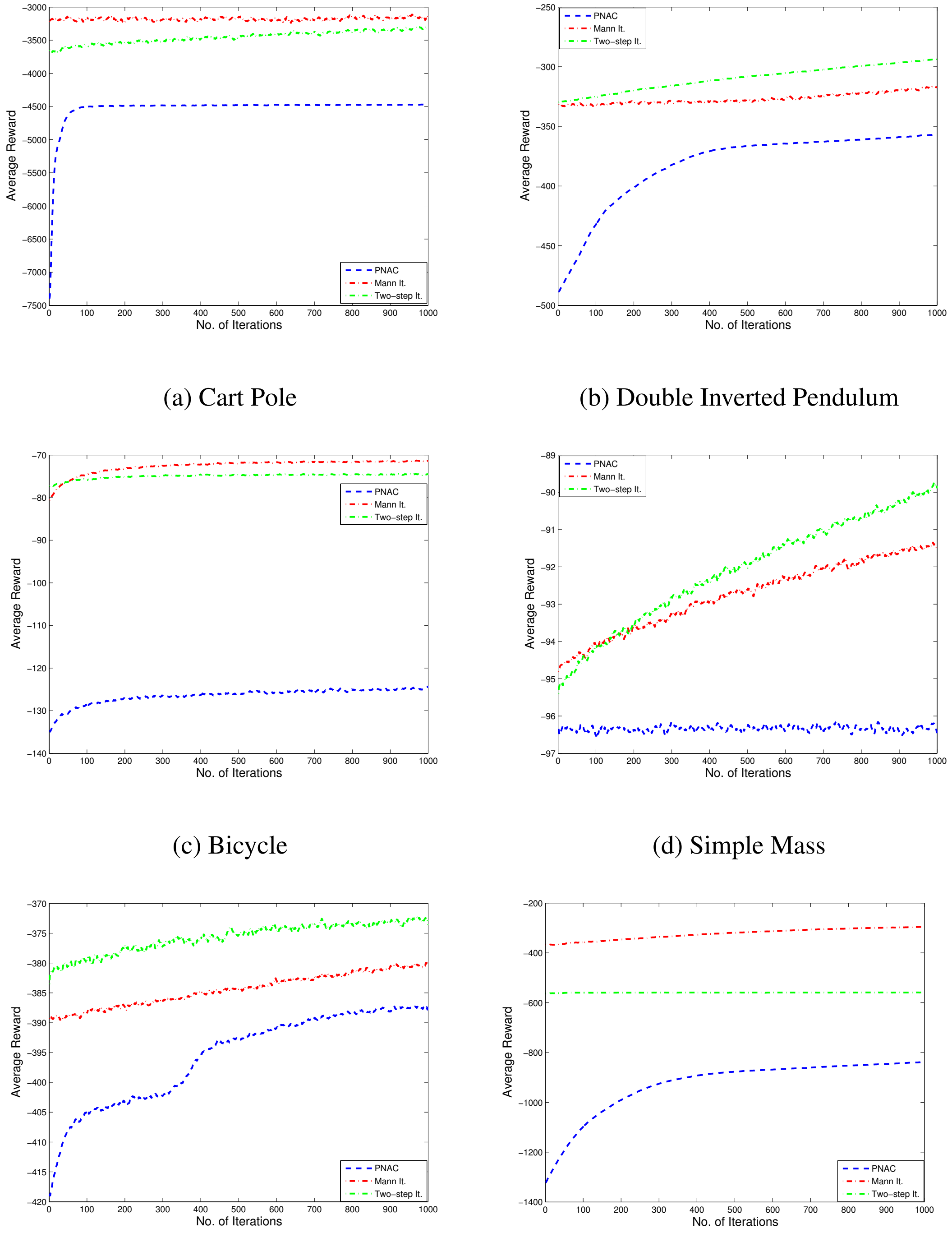
where $\theta(i)$ is the i^{th} scalar of the vector $\boldsymbol{\theta}$, n is the dimension of $\boldsymbol{\theta}$ and k is a constant. This set can be verified as a $\mathcal{K}_r$ set which is non-convex and can be regarded as an example construction of a generic non-convex r-prox-regular sets. As mentioned before, there are a variety of non-convex r-prox-regular sets such as p-convex sets, $C^{1,1}$ and sub-manifolds of $\mathbb{H}$ (Clarke et al., 2008; Poliquin et al., 2000). PNAC is not capable of tackling any non-convex set as the projection becomes difficult to construct. When the constraints are violated, PNAC receives an extra punishment reward: $-10 * \sqrt{\|x_t - x_g\|^2}$, where x_t is the current state and x_g is the goal state. Unlike PNAC, our method can adapt non-convex programming algorithms for computing the projection. In the above setting, however, we propose a decomposition method to solve this projection which circumvents the need to use non-convex optimization techniques. The construction is as follows: Given a $\boldsymbol{\theta}_t$, if $\|\boldsymbol{\theta}_t\|_2^2 < k$, then we solve the following convex programming problem:

$$\begin{aligned} &\min \|y - \boldsymbol{\theta}_t\|_2^2 \\ &\text{s.t.} \quad \|y\|_2^2 = k. \end{aligned}$$

In the cases of $\theta_t(1)^2 + \theta_t(2)^2 + \cdots + \theta_t(n-1)^2 > k$ or $\theta_t(n) < -2k$, we solve following convex programming problem:

$$\begin{aligned} &\min \|y - \boldsymbol{\theta}_t\|_2^2 \\ &s.t. \quad y(1)^2 + y(2)^2 + \cdots + y(n-1)^2 \leq k \\ &\qquad\quad y(n) \geq -2k. \end{aligned}$$

Otherwise, $\boldsymbol{\theta}_t \in \mathcal{K}_r$, the projection is not required. With this construction, it is easy to verify that the feasible solution set is equivalent to the original non-convex $\mathcal{K}_r$, allowing



(a) Cart Pole

(b) Double Inverted Pendulum

(c) Bicycle

(d) Simple Mass

(e) Robotic Arm

(f) Double Mass

Figure 3: Average reward results on a variety of benchmark dynamical systems show that our method is capable of: 1) handling non-convex safety constraints, and 2) significantly outperforming state-of-the-art methods.

the circumvention of non-convex programming.

Experimental Results Results for the six systems are shown in Figure 1, where the performance of the learners is averaged over 10 randomly generated constrained tasks from the settings used in Table 1. We run each experiment for 1000 iterations. At each iteration, the learner used its policy to generate 50 trajectories of 150 steps and updated its policy. As expected, our methods outperform PNAC due to the non-convex constraints, both in terms of initial performance and learning speeds. Generally, the results demonstrate that our approach is capable of outperforming comparisons in terms of both initial performance and learning speeds under non-convex constraints.

6 Related Work

Bhatnagar et al. (2009) introduced Projected Natural Actor-Critic (PNAC) algorithms for the average reward setting. However, they do not provide the construction of the projection and their method can not handle non-convex constraints. Thomas et al. (2013) proposed a new projection called projected natural gradient to improve the performance of PNAC, but their method is still limited to convex constraints.

The Variational Inequality (VI) problem plays an important role in optimization, economics and game theory (Mahadevan et al., 2014). It also provides a useful framework for reinforcement learning. Bertsekas (2009) established the connection between the VI problem and temporal difference methods and solves the VI problem with Galerkin methods (Gordon, 2013). In contrast, we show the connection between the (non-convex) VI problem and the policy search problem.

Safe reinforcement learning has drawn a lot of attention in the machine learning community (García and Fernández, 2015). Ammar et al. (2015) incorporated safe reinforcement learning into lifelong learning setting. Torrey and Taylor (2013) introduce policy advice from an expert teacher to avoid unsafe actions. Liu et al. (2015) show that VIs can be used to design safe stable proximal gradient TD methods, and provided the first finite sample bounds for a linear TD method in the reinforcement learning literature. Mahadevan et al. (2014) study the mirror descent and mirror proximal method for convex VIs. However, non-convex constraints are not considered in any of the above works.

Conclusions

In this paper, we established the connection between policy search with non-convex constraints and non-convex variational inequalities. We used the Mann iteration method and the two-step iteration approach to solve the policy search problem. We must point out that the results we derived in this paper can be easily extended to three-step iterative methods proposed by Noor (2000) and Noor (2004). We also proved the convergence of these two methods in the stochastic setting. Finally, we empirically showed that our method is capable of dealing with non-convex constraints, a property current techniques can not handle. Future work will test these methods on more complex systems, including physical robots, and investigate when the Mann iteration will outperform the two-step iteration method (and vice versa). Also, we will consider studying the convergence rate or sample complexity in the future work.

Acknowledgments

We thank the anonymous reviewers for their useful suggestions and comments. This research has taken place in part at the Intelligent Robot Learning (IRL) Lab, Washington State University. IRL's support includes NASA NNX16CD07C, NSF IIS-1149917, NSF IIS-1643614, and USDA 2014-67021-22174.

References

Ammar, H. B., Tutunov, R., and Eaton, E. (2015). Safe policy search for lifelong reinforcement learning with sublinear regret. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 2361–2369.

Ansari, Q. H. and Balooee, J. (2013). Predictor–corrector methods for general regularized nonconvex variational inequalities. *Journal of Optimization Theory and Applications*, 159(2):473–488.

Bertsekas, D. P. (2009). Projected equations, variational inequalities, and temporal difference methods. *Lab. for Information and Decision Systems Report LIDS-P-2808, MIT*.

Bhatnagar, S., Sutton, R. S., Ghavamzadeh, M., and Lee, M. (2009). Natural actor–critic algorithms. *Automatica*, 45(11):2471–2482.

Bou Ammar, H., Eaton, E., Luna, J. M., and Ruvolo, P. (2015). Autonomous cross-domain knowledge transfer in lifelong policy gradient reinforcement learning. In

Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-15).

Boyd, S. and Vandenberghe, L. (2004). *Convex optimization*. Cambridge university press.

Clarke, F. H., Ledyaev, Y. S., Stern, R. J., and Wolenski, P. R. (2008). *Nonsmooth analysis and control theory*, volume 178. Springer Science & Business Media.

Duchi, J., Hazan, E., and Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*, 12:2121–2159.

Federer, H. (1959). Curvature measures. *Transactions of the American Mathematical Society*, 93(3):418–491.

García, J. and Fernández, F. (2015). A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*, 16:1437–1480.

Gordon, G. J. (2013). Galerkin methods for complementarity problems and variational inequalities. *arXiv preprint arXiv:1306.4753*.

Harker, P. T. and Pang, J.-S. (1990). Finite-dimensional variational inequality and nonlinear complementarity problems: a survey of theory, algorithms and applications. *Mathematical programming*, 48(1-3):161–220.

Hendrix, E. M., Boglárka, G., et al. (2010). *Introduction to nonlinear and global optimization*. Springer New York.

Kaelbling, L. P., Littman, M. L., and Moore, A. W. (1996). Reinforcement learning: A survey. *Journal of artificial intelligence research*, pages 237–285.

Khobotov, E. N. (1987). Modification of the extra-gradient method for solving variational inequalities and certain optimization problems. *USSR Computational Mathematics and Mathematical Physics*, 27(5):120–127.

Kober, J. and Peters, J. R. (2009). Policy search for motor primitives in robotics. In *Advances in neural information processing systems*, pages 849–856.

Liu, B., Liu, J., Ghavamzadeh, M., Mahadevan, S., and Petrik, M. (2015). Finite-sample analysis of proximal gradient td algorithms. In *Conference on Uncertainty in Artificial Intelligence*. Citeseer.

Mahadevan, S., Liu, B., Thomas, P., Dabney, W., Giguere, S., Jacek, N., Gemp, I., and Liu, J. (2014). Proximal reinforcement learning: A new theory of sequential decision making in primal-dual spaces. *arXiv preprint arXiv:1405.6757*.

Noor, M. A. (2000). New approximation schemes for general variational inequalities. *Journal of Mathematical Analysis and applications*, 251(1):217–229.

Noor, M. A. (2004). Some developments in general variational inequalities. *Applied Mathematics and Computation*, 152(1):199–277.

Noor, M. A. (2009). Projection methods for nonconvex variational inequalities. *Optimization Letters*, 3(3):411–418.

Noor, M. A., Al-Said, E., Noor, K. I., and Yao, Y. (2011). Extragradient methods for solving nonconvex variational inequalities. *Journal of Computational and Applied Mathematics*, 235(9):3104–3108.

Peters, J. and Schaal, S. (2008). Natural actor-critic. *Neurocomputing*, 71(7):1180–1190.

Poliquin, R., Rockafellar, R., and Thibault, L. (2000). Local differentiability of distance functions. *Transactions of the American Mathematical Society*, 352(11):5231–5249.

Robbins, H. and Siegmund, D. (1985). A convergence theorem for non negative almost supermartingales and some applications. In *Herbert Robbins Selected Papers*, pages 111–135. Springer.

Sutton, R. S. and Barto, A. G. (1998). *Reinforcement learning: An introduction*. MIT press.

Thomas, P. S., Dabney, W. C., Giguere, S., and Mahadevan, S. (2013). Projected natural actor-critic. In *Advances in Neural Information Processing Systems*, pages 2337–2345.

Tobin, R. L. (1986). Sensitivity analysis for variational inequalities. *Journal of Optimization Theory and Applications*, 48(1):191–204.

Torrey, L. and Taylor, M. (2013). Teaching on a budget: Agents advising agents in reinforcement learning. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, pages 1053–1060. International Foundation for Autonomous Agents and Multiagent Systems.

Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256.

A Convergence Proofs

In this appendix, we show convergence results, with probability 1, for Algorithms 1 and 2. Starting with the first step, we prove:

Lemma 2. *For policies satisfying log-concave distributions of the form $\pi_{\boldsymbol{\theta}}\left(\boldsymbol{a}_m^{(k)}|\boldsymbol{s}_m^{(k)}\right)$, for $\boldsymbol{s}_m^{(k)} \in \mathcal{S}$ and $\boldsymbol{a}_m^{(k)} \in \mathcal{A}$, the gradient of $\mathcal{J}(\boldsymbol{\theta})$ is strongly monotone and Lipschitz, i.e., for any $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$:*

$$\left\|\nabla_{\boldsymbol{\theta}}\mathcal{J}(\boldsymbol{\theta})\Big|_{\boldsymbol{\theta}} - \nabla_{\boldsymbol{\theta}}\mathcal{J}(\boldsymbol{\theta})\Big|_{\boldsymbol{\theta}'}\right\| \leq \left(MT\boldsymbol{\Gamma}_{\mathbf{max}} + 2\mu\right)\|\boldsymbol{\theta} - \boldsymbol{\theta'}\|,$$

with M, T, and $\boldsymbol{\Gamma}_{\max}$ denoting the total number of trajectories, the length of each trajectory, and the upper-bound on the state-action basis functions and

$$\left(\nabla_{\boldsymbol{\theta}}\mathcal{J}(\boldsymbol{\theta})\Big|_{\boldsymbol{\theta}} - \nabla_{\boldsymbol{\theta}}\mathcal{J}(\boldsymbol{\theta})\Big|_{\boldsymbol{\theta}'}\right)^{\mathsf{T}}\left(\boldsymbol{\theta} - \boldsymbol{\theta'}\right) \geq 2\mu\|\boldsymbol{\theta} - \boldsymbol{\theta'}\|,$$

respectively.

Proof. First, we show that the gradient of $\mathcal{J}(\boldsymbol{\theta})$ is Lipschitz continuous. Assume the policy π is characterized by a log-concave distribution of the following form:

$$\pi_{\boldsymbol{\theta}}\left(\boldsymbol{a}_m^k \mid \boldsymbol{s}_m^k\right) = \exp\{\varphi_{\boldsymbol{\theta}}\left(\boldsymbol{a}_m^k \mid \boldsymbol{s}_m^k\right)\},$$

where $\varphi_{\boldsymbol{\theta}}$ is concave and parameterized by vector $\boldsymbol{\theta}$. Consequently, the policy search objective can be written as:

$$\mathcal{J}(\boldsymbol{\theta}) = -\sum_{k=1}^{M}\sum_{m=0}^{T-1}\varphi_{\boldsymbol{\theta}}\left(\boldsymbol{a}_m^k \mid \boldsymbol{s}_m^k\right) + \mu\|\boldsymbol{\theta}\|_2^2. \tag{8}$$

Hence, the gradient can be determined using:

$$\nabla_{\boldsymbol{\theta}}\mathcal{J}(\boldsymbol{\theta}) = -\sum_{k=1}^{M}\sum_{m=0}^{T-1}\nabla_{\boldsymbol{\theta}}\varphi_{\boldsymbol{\theta}}\left(\boldsymbol{a}_m^k \mid \boldsymbol{s}_m^k\right) + 2\mu\boldsymbol{\theta}. \tag{9}$$

Thus, we have

$$
\begin{aligned}
&\left\|\nabla_{\boldsymbol{\theta}}\mathcal{J}(\boldsymbol{\theta})\Big|_{\boldsymbol{\theta}} - \nabla_{\boldsymbol{\theta}}\mathcal{J}(\boldsymbol{\theta})\Big|_{\boldsymbol{\theta}'}\right\| \\
&= \|-\sum_{k=1}^{M}\sum_{m=1}^{T-1}\nabla_{\boldsymbol{\theta}}\varphi_{\boldsymbol{\theta}}\left(\boldsymbol{a}_m^k \mid \boldsymbol{s}_m^k\right) + -\sum_{k=1}^{M}\sum_{m=0}^{T-1}\nabla_{\boldsymbol{\theta}'}\varphi_{\boldsymbol{\theta}'}\left(\boldsymbol{a}_m^k \mid \boldsymbol{s}_m^k\right) + 2\mu\boldsymbol{\theta} - 2\mu\boldsymbol{\theta'}\| \\
&\leq \left\|-\sum_{k=1}^{M}\sum_{m=0}^{T-1}\left(\nabla_{\boldsymbol{\theta}}\varphi_{\boldsymbol{\theta}}\left(\boldsymbol{a}_m^k \mid \boldsymbol{s}_m^k\right) - \nabla_{\boldsymbol{\theta}'}\varphi_{\boldsymbol{\theta}'}\left(\boldsymbol{a}_m^k \mid \boldsymbol{s}_m^k\right)\right)\right\| + 2\mu\left\|\boldsymbol{\theta} - \boldsymbol{\theta'}\right\|.
\end{aligned}
$$

Asume that $\nabla_{\boldsymbol{\theta}}\varphi_{\boldsymbol{\theta}}\left(\boldsymbol{a}_m^k \mid \boldsymbol{s}_m^k\right) - \nabla_{\boldsymbol{\theta}'}\varphi_{\boldsymbol{\theta}'}\left(\boldsymbol{a}_m^k \mid \boldsymbol{s}_m^k\right) = (\boldsymbol{\theta} - \boldsymbol{\theta'})^T\Gamma\left(\boldsymbol{a}_m^k, \boldsymbol{s}_m^k\right)$

$$
\begin{aligned}
&= \left\|-\sum_{k=1}^{M}\sum_{m=0}^{T-1}(\boldsymbol{\theta} - \boldsymbol{\theta'})^T\Gamma\left(\boldsymbol{a}_m^k, \boldsymbol{s}_m^k\right)\right\| + 2\mu\left\|\boldsymbol{\theta} - \boldsymbol{\theta'}\right\| \\
&\leq \sum_{k=1}^{M}\sum_{m=0}^{T-1}\left\|(\boldsymbol{\theta} - \boldsymbol{\theta'})^T\right\|\left\|\Gamma\left(\boldsymbol{a}_m^k, \boldsymbol{s}_m^k\right)\right\| + 2\mu\left\|\boldsymbol{\theta} - \boldsymbol{\theta'}\right\| \\
&\leq MT\max_{k,m}\left\{\left\|\Gamma\left(\boldsymbol{a}_m^k, \boldsymbol{s}_m^k\right)\right\|\right\}\left\|\boldsymbol{\theta} - \boldsymbol{\theta'}\right\| + 2\mu\left\|\boldsymbol{\theta} - \boldsymbol{\theta'}\right\| \\
&= \left(MT\boldsymbol{\Gamma_{max}} + 2\mu\right)\left\|\boldsymbol{\theta} - \boldsymbol{\theta'}\right\|,
\end{aligned}
$$

where $\boldsymbol{\Gamma_{max}} = \max_{k,m}\left\{\left\|\Gamma\left(\boldsymbol{a}_m^k, \boldsymbol{s}_m^k\right)\right\|\right\}$. Therefore, we obtain the bound

$$
\left\|\nabla_{\boldsymbol{\theta}}\mathcal{J}(\boldsymbol{\theta})\Big|_{\boldsymbol{\theta}} - \nabla_{\boldsymbol{\theta}}\mathcal{J}(\boldsymbol{\theta})\Big|_{\boldsymbol{\theta}'}\right\| \leq \left(MT\boldsymbol{\Gamma_{max}} + 2\mu\right)\left\|\boldsymbol{\theta} - \boldsymbol{\theta'}\right\|,
$$

which is finalizes the Lipschitz continuity proof of the lemma.

Next, we show the gradient $\nabla_{\boldsymbol{\theta}}\mathcal{J}(\boldsymbol{\theta})$ in Equation (9) is strongly monotone. First, note that $\mathcal{J}(\boldsymbol{\theta}) - \mu\left\|\boldsymbol{\theta}\right\|_2^2 = -\sum_{k=1}^{M}\sum_{m=0}^{T-1}\varphi_{\boldsymbol{\theta}}\left(\boldsymbol{a}_m^k \mid \boldsymbol{s}_m^k\right)$ is convex since $\varphi_{\boldsymbol{\theta}}$ is concave. Due to the convexity of $\mathcal{J}(\boldsymbol{\theta}) - \mu\left\|\boldsymbol{\theta}\right\|_2^2$, we can write:

$$
\mathcal{J}(\boldsymbol{\theta}) - \mu\left\|\boldsymbol{\theta}\right\|_2^2 - \mathcal{J}(\boldsymbol{\theta}') + \mu\left\|\boldsymbol{\theta}'\right\|_2^2 \geq \left(\nabla_{\boldsymbol{\theta}}\mathcal{J}(\boldsymbol{\theta})\Big|_{\boldsymbol{\theta}'} - 2\mu\boldsymbol{\theta}'\right)^T(\boldsymbol{\theta} - \boldsymbol{\theta'}). \tag{10}
$$

And,

$$
\mathcal{J}(\boldsymbol{\theta}') - \mu\left\|\boldsymbol{\theta}'\right\|_2^2 - \mathcal{J}(\boldsymbol{\theta}) + \mu\left\|\boldsymbol{\theta}\right\|_2^2 \geq \left(\nabla_{\boldsymbol{\theta}}\mathcal{J}(\boldsymbol{\theta})\Big|_{\boldsymbol{\theta}} - 2\mu\boldsymbol{\theta}\right)^T(\boldsymbol{\theta'} - \boldsymbol{\theta}) \tag{11}
$$

Combining Equation (10) and Equation (11), we obtain:

$$\left(\nabla_{\boldsymbol{\theta}}\mathcal{J}(\boldsymbol{\theta})\Big|_{\boldsymbol{\theta}} - \nabla_{\boldsymbol{\theta}}\mathcal{J}(\boldsymbol{\theta})\Big|_{\boldsymbol{\theta}'}\right)^T (\boldsymbol{\theta} - \boldsymbol{\theta}') \geq 2\mu \left\|\boldsymbol{\theta} - \boldsymbol{\theta}'\right\|.$$

Thus, $\nabla_{\boldsymbol{\theta}}\mathcal{J}(\boldsymbol{\theta})$ is strongly monotone. □

Before diving into these details, we need to introduce the supermartingale convergence theorem by Robbins and Siegmund (1985):

Theorem 3. *Let $\{X_t\}$, $\{Y_t\}$, $\{Z_t\}$ and $\{W_t\}$ be sequences of nonnegative random variables such that*

$$\mathbb{E}[X_{t+1}|\mathcal{F}_t] \leq (1 + W_t)X_t + Y_t - Z_t, \qquad t \geq 0 \textit{ with probability } 1,$$

where $\mathcal{F}_t$ denotes all the history information $\{\{X_{t'}\}_{t'\leq t}, \{Y_{t'}\}_{t'\leq t}, \{Z_{t'}\}_{t'\leq t}\}$. If

$$\sum_{t=0}^{\infty} W_t < \infty \textit{ and } \sum_{t=0}^{\infty} Y_t < \infty,$$

then, X_t converges to a limit with probability 1 and $\sum_{t=0}^{\infty} Z_t < \infty$.

We also make use of the following lemma implying that the non-convex variational inequality problem in Equation (3) is equivalent to the determining the fixed-point solution of Equation (12):

Lemma 3 (Noor (2009))**.** *If $\boldsymbol{\theta}^\star \in \mathcal{K}_r$ is a solution of the $\mathcal{NVI}(\boldsymbol{T}, \mathcal{K}_r)$, if and only if $\boldsymbol{\theta}^\star$ satisfies*

$$\boldsymbol{\theta}^\star = \textit{Proj}_{\mathcal{K}_r}[\boldsymbol{\theta}^\star - \rho \boldsymbol{T}\boldsymbol{\theta}^\star], \tag{12}$$

where $\textit{Proj}_{\mathcal{K}_r}$ is the projection of $\mathbb{H}$ onto the uniformly r-prox-regular set $\mathcal{K}_r$.

Finally, we derive the following result concerning operators:

Lemma 4. *Let operator $\boldsymbol{T}$ be strongly monotone with constant $\alpha > 0$ and Lipschitz continuous with $\beta > 0$, then for $\boldsymbol{u} \neq \boldsymbol{v} \in \mathcal{K}_r$,*

$$\|\boldsymbol{u} - \boldsymbol{v} - \rho(\boldsymbol{T}\boldsymbol{u} - \boldsymbol{T}\boldsymbol{v})\| \leq \sqrt{1 - 2\rho\alpha + \rho^2\beta^2}\, \|\boldsymbol{u} - \boldsymbol{v}\| .$$

Proof. Due to the property of norm, we have

$$\|\boldsymbol{u} - \boldsymbol{v} - \rho(\boldsymbol{T}\boldsymbol{u} - \boldsymbol{T}\boldsymbol{v})\|^2 = \|\boldsymbol{u} - \boldsymbol{v}\|^2 - 2\rho\langle \boldsymbol{T}\boldsymbol{u} - \boldsymbol{T}\boldsymbol{v}, \boldsymbol{u} - \boldsymbol{v}\rangle + \rho^2 \|\boldsymbol{T}\boldsymbol{u} - \boldsymbol{T}\boldsymbol{v}\|^2$$

$\boldsymbol{T}$ is strongly monotone with constant $\alpha > 0$ and Lipschitz continuous with $\beta > 0$

$$\begin{aligned} &\leq \|\boldsymbol{u} - \boldsymbol{v}\|^2 - 2\rho\alpha \|\boldsymbol{u} - \boldsymbol{v}\|^2 + \rho^2\beta^2 \|\boldsymbol{u} - \boldsymbol{v}\|^2 \\ &\leq (1 - 2\rho\alpha + \rho^2\beta^2) \|\boldsymbol{u} - \boldsymbol{v}\|^2 \end{aligned}$$

Then, we have

$$\|\boldsymbol{u} - \boldsymbol{v} - \rho(\boldsymbol{T}\boldsymbol{u} - \boldsymbol{T}\boldsymbol{v})\| \leq \sqrt{1 - 2\rho\alpha + \rho^2\beta^2}\, \|\boldsymbol{u} - \boldsymbol{v}\| .$$

□

Now, we are ready to present our main convergence theorem:

Proof of Theorem 1. According to Lemma 2, let the gradient $\nabla_{\boldsymbol{\theta}}\mathcal{J}_{\boldsymbol{\theta}}$ be strongly monotone with constant $\alpha > 0$ and Lipschitz continuous with $\beta > 0$. Further, let the projection $\mathrm{Proj}_{\mathcal{K}_r}$ be Lipschitz continuous with $\delta > 0$, and $\boldsymbol{\theta}^\star \in \mathbb{H}$ be a solution of the non-convex variational inequality (Equation (3)). Then, by Lemma 3, we have:

$$\boldsymbol{\theta}^\star = (1 - \alpha_t)\boldsymbol{\theta}^\star + \alpha_t \mathrm{Proj}_{\mathcal{K}_r}\left[\boldsymbol{\theta}^\star - \rho\nabla_{\boldsymbol{\theta}}\mathcal{J}(\boldsymbol{\theta})\Big|_{\boldsymbol{\theta}^\star}\right], \tag{13}$$

where $\alpha_t \in [0, 1]$ and ρ are constant. Using Equation (13), we obtain:

$$\begin{aligned}
&\|\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}^\star\| \\
&= \left\| (1-\alpha_t)(\boldsymbol{\theta}_t - \boldsymbol{\theta}^\star) + \alpha_t \{ \mathrm{Proj}_{\mathcal{K}_r} \left[\boldsymbol{\theta}_t - \rho \nabla_{\boldsymbol{\theta}} \mathcal{J}(\boldsymbol{\theta}) \Big|_{\boldsymbol{\theta}_t} \right] - \mathrm{Proj}_{\mathcal{K}_r} \left[\boldsymbol{\theta}^\star - \rho \nabla_{\boldsymbol{\theta}} \mathcal{J}(\boldsymbol{\theta}) \Big|_{\boldsymbol{\theta}^\star} \right] \} \right\| \\
&\leq (1-\alpha_t) \|\boldsymbol{\theta}_t - \boldsymbol{\theta}^\star\| + \alpha_t \left\| \mathrm{Proj}_{\mathcal{K}_r} \left[\boldsymbol{\theta}_t - \rho \nabla_{\boldsymbol{\theta}} \mathcal{J}(\boldsymbol{\theta}) \Big|_{\boldsymbol{\theta}_t} \right] - \mathrm{Proj}_{\mathcal{K}_r} \left[\boldsymbol{\theta}^\star - \rho \nabla_{\boldsymbol{\theta}} \mathcal{J}(\boldsymbol{\theta}) \Big|_{\boldsymbol{\theta}^\star} \right] \right\|
\end{aligned}$$

Using the Lipschitz continuity of the projection $\mathrm{Proj}_{\mathcal{K}_r}$, we get:

$$\leq (1-\alpha_t) \|\boldsymbol{\theta}_t - \boldsymbol{\theta}^\star\| + \alpha_t \delta \left\| \boldsymbol{\theta}_t - \boldsymbol{\theta}^\star - \rho \left(\nabla_{\boldsymbol{\theta}} \mathcal{J}(\boldsymbol{\theta}) \Big|_{\boldsymbol{\theta}_t} - \nabla_{\boldsymbol{\theta}} \mathcal{J}(\boldsymbol{\theta}) \Big|_{\boldsymbol{\theta}^\star} \right) \right\|$$

By Lemma 4 with, $\boldsymbol{T} = \nabla_{\boldsymbol{\theta}} \mathcal{J}(\boldsymbol{\theta})$, we have:

$$\leq (1-\alpha_t) \|\boldsymbol{\theta}_t - \boldsymbol{\theta}^\star\| + \alpha_t \delta \sqrt{1 - 2\alpha\rho + \beta^2\rho^2} \|\boldsymbol{\theta}_t - \boldsymbol{\theta}^\star\|$$

By letting $\varepsilon = \delta\sqrt{1 - 2\alpha\rho + \beta^2\rho^2} \neq 0$, we obtain:

$$= [1 - (1-\varepsilon)\alpha_t] \|\boldsymbol{\theta}_t - \boldsymbol{\theta}^\star\| \tag{14}$$

Taking conditional expectation on both sides of Equation (14), we obtain:

$$\mathbb{E}[\|\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}^\star\| \,|\mathcal{F}_t] \leq \|\boldsymbol{\theta}_t - \boldsymbol{\theta}^\star\| - (1-\varepsilon)\alpha_t \|\boldsymbol{\theta}_t - \boldsymbol{\theta}^\star\| . \tag{15}$$

Applying Theorem 3 to Equation (15), we find that $\|\boldsymbol{\theta}_t - \boldsymbol{\theta}^\star\|$ converges to a limit with probability 1 and:

$$\sum_{t=0}^{\infty} (1-\varepsilon)\alpha_t \|\boldsymbol{\theta}_t - \boldsymbol{\theta}^\star\| < \infty$$

Assume that $\|\boldsymbol{\theta}_t - \boldsymbol{\theta}^\star\|$ converges to a non-zero limit, then

$$\sum_{t=0}^{\infty} (1-\varepsilon)\alpha_t \|\boldsymbol{\theta}_t - \boldsymbol{\theta}^\star\| = \infty,$$

which is a contradiction. To conclude, $\|\boldsymbol{\theta}_t - \boldsymbol{\theta}^\star\|$ converges to 0 with probability 1, or equivalently, $\boldsymbol{\theta}_t$ converges the optimal solution $\boldsymbol{\theta}^\star$ with probability 1. □

Proof of Theorem 2. According to Lemma 2, let the gradient $\nabla_{\boldsymbol{\theta}}\mathcal{J}_{\boldsymbol{\theta}}$ be strongly monotone with constant $\alpha > 0$ and Lipschitz continuous with $\beta > 0$. Further, let the projection $\text{Proj}_{\mathcal{K}_r}$ be Lipschitz continuous with $\delta > 0$, and $\boldsymbol{\theta}^\star \in \mathbb{H}$ be a solution of the non-convex variational inequality $\mathcal{NVI}(\nabla_{\boldsymbol{\theta}} J'_{\boldsymbol{\theta}}, \mathcal{K}_r)$. Then, by Lemma 3 and Algorithm 2, we have:

$$\begin{aligned}
\boldsymbol{\theta}^\star &= (1-\beta_t)\boldsymbol{\theta}^\star + \beta_t\boldsymbol{\theta}^\star \\
&= (1-\beta_t)\boldsymbol{\theta}^\star + \beta_t\text{Proj}_{\mathcal{K}_r}\left[\boldsymbol{\theta}^\star - \rho\nabla_{\boldsymbol{\theta}}\mathcal{J}(\boldsymbol{\theta})\Big|_{\boldsymbol{\theta}^\star}\right] \\
&= \boldsymbol{\mu}^\star
\end{aligned}$$

where $\boldsymbol{\mu}^\star \in K_r$. Thus,

$$\begin{aligned}
\boldsymbol{\theta}^\star &= (1-\alpha_t)\boldsymbol{\theta}^\star + \alpha_t\text{Proj}_{\mathcal{K}_r}\left[\boldsymbol{\theta}^\star - \rho\nabla_{\boldsymbol{\theta}}\mathcal{J}(\boldsymbol{\theta})\Big|_{\boldsymbol{\theta}^\star}\right] \\
&= (1-\alpha_t)\boldsymbol{\theta}^\star + \alpha_t\text{Proj}_{\mathcal{K}_r}\left[\boldsymbol{\mu}^\star - \rho\nabla_{\boldsymbol{\mu}}\mathcal{J}(\boldsymbol{\mu})\Big|_{\boldsymbol{\mu}^\star}\right]
\end{aligned}$$

Let $\boldsymbol{\mu}_t$ is generated by Algorithm 2 at time t, it yields

$$\begin{aligned}
&\|\boldsymbol{\mu}_t - \boldsymbol{\mu}^\star\| \\
&= \left\|(1-\beta_t)(\boldsymbol{\theta}_t - \boldsymbol{\theta}^\star) + \beta_t\{\text{Proj}_{\mathcal{K}_r}\left[\boldsymbol{\theta}_t - \rho\nabla_{\boldsymbol{\theta}}\mathcal{J}(\boldsymbol{\theta})\Big|_{\boldsymbol{\theta}_t}\right] - \text{Proj}_{\mathcal{K}_r}\left[\boldsymbol{\theta}^\star - \rho\nabla_{\boldsymbol{\theta}}\mathcal{J}(\boldsymbol{\theta})\Big|_{\boldsymbol{\theta}^\star}\right]\}\right\| \\
&\leq (1-\beta_t)\|\boldsymbol{\theta}_t - \boldsymbol{\theta}^\star\| + \beta_t\left\|\text{Proj}_{\mathcal{K}_r}\left[\boldsymbol{\theta}_t - \rho\nabla_{\boldsymbol{\theta}}\mathcal{J}(\boldsymbol{\theta})\Big|_{\boldsymbol{\theta}_t}\right] - \text{Proj}_{\mathcal{K}_r}\left[\boldsymbol{\theta}^\star - \rho\nabla_{\boldsymbol{\theta}}\mathcal{J}(\boldsymbol{\theta})\Big|_{\boldsymbol{\theta}^\star}\right]\right\|
\end{aligned}$$

Using the Lipschitz continuity of the projection $\text{Proj}_{\mathcal{K}_r}$, we get:

$$\leq (1-\beta_t)\|\boldsymbol{\theta}_t - \boldsymbol{\theta}^\star\| + \beta_t\delta\left\|\boldsymbol{\theta}_t - \boldsymbol{\theta}^\star - \rho\left(\nabla_{\boldsymbol{\theta}}\mathcal{J}(\boldsymbol{\theta})\Big|_{\boldsymbol{\theta}_t} - \nabla_{\boldsymbol{\theta}}\mathcal{J}(\boldsymbol{\theta})\Big|_{\boldsymbol{\theta}^\star}\right)\right\|$$

By Lemma 4 with $\boldsymbol{T} = \nabla_{\boldsymbol{\theta}}\mathcal{J}(\boldsymbol{\theta})$, we have:

$$\leq (1-\beta_t)\|\boldsymbol{\theta}_t - \boldsymbol{\theta}^\star\| + \beta_t\delta\sqrt{1-2\alpha\rho+\beta^2\rho^2}\,\|\boldsymbol{\theta}_t - \boldsymbol{\theta}^\star\|$$

By letting $\varepsilon = \delta\sqrt{1-2\alpha\rho+\beta^2\rho^2} \neq 0$, we obtain:

$$= [1-(1-\varepsilon)\beta_t]\,\|\boldsymbol{\theta}_t - \boldsymbol{\theta}^\star\| \tag{16}$$

Then, we show that $\boldsymbol{\theta}_{t+1}$ converges to $\boldsymbol{\theta}^\star$. Similarly, we have

$$
\begin{aligned}
&\|\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}^\star\| \\
&= \left\|(1-\alpha_t)(\boldsymbol{\theta}_t - \boldsymbol{\theta}^\star) + \alpha_t\{\mathrm{Proj}_{\mathcal{K}_r}\left[\boldsymbol{\mu}_t - \rho\nabla_{\boldsymbol{\mu}}\mathcal{J}(\boldsymbol{\mu})\Big|_{\boldsymbol{\mu}_t}\right] - \mathrm{Proj}_{\mathcal{K}_r}\left[\boldsymbol{\mu}^\star - \rho\nabla_{\boldsymbol{\mu}}\mathcal{J}(\boldsymbol{\mu})\Big|_{\boldsymbol{\mu}^\star}\right]\}\right\| \\
&\leq (1-\alpha_t)\|\boldsymbol{\theta}_t - \boldsymbol{\theta}^\star\| + \alpha_t\left\|\mathrm{Proj}_{\mathcal{K}_r}\left[\boldsymbol{\mu}_t - \rho\nabla_{\boldsymbol{\mu}}\mathcal{J}(\boldsymbol{\mu})\Big|_{\boldsymbol{\mu}_t}\right] - \mathrm{Proj}_{\mathcal{K}_r}\left[\boldsymbol{\mu}^\star - \rho\nabla_{\boldsymbol{\mu}}\mathcal{J}(\boldsymbol{\mu})\Big|_{\boldsymbol{\mu}^\star}\right]\right\|
\end{aligned}
$$

Using Lipschitz continuity of the projection $\mathrm{Proj}_{\mathcal{K}_r}$, we get:

$$
\leq (1-\alpha_t)\|\boldsymbol{\theta}_t - \boldsymbol{\theta}^\star\| + \alpha_t\delta\left\|\boldsymbol{\mu}_t - \boldsymbol{\mu}^\star - \rho\left(\nabla_{\boldsymbol{\mu}}\mathcal{J}(\boldsymbol{\mu})\Big|_{\boldsymbol{\mu}_t} - \nabla_{\boldsymbol{\mu}}\mathcal{J}(\boldsymbol{\mu})\Big|_{\boldsymbol{\mu}^\star}\right)\right\|
$$

By Lemma 4, with $\boldsymbol{T} = \nabla_{\boldsymbol{\mu}}\mathcal{J}(\boldsymbol{\mu})$, we have:

$$
\leq (1-\alpha_t)\|\boldsymbol{\theta}_t - \boldsymbol{\theta}^\star\| + \alpha_t\delta\sqrt{1-2\alpha\rho+\beta^2\rho^2}\,\|\boldsymbol{\mu}_t - \boldsymbol{\mu}^\star\|
$$

By letting $\varepsilon = \delta\sqrt{1-2\alpha\rho+\beta^2\rho^2} \neq 0$ and using Equation (16), we obtain:

$$
\begin{aligned}
&\leq (1-\alpha_t)\|\boldsymbol{\theta}_t - \boldsymbol{\theta}^\star\| + \alpha_t\varepsilon[1-(1-\varepsilon)\beta_t]\,\|\boldsymbol{\theta}_t - \boldsymbol{\theta}^\star\| \\
&= \{1-(1-\varepsilon[1-(1-\varepsilon)\beta_t])\alpha_t\}\,\|\boldsymbol{\theta}_t - \boldsymbol{\theta}^\star\| \qquad (17)
\end{aligned}
$$

Taking the conditional expectation on both sides of Equation (17), we obtain:

$$
\mathbb{E}[\|\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}^\star\|\,|\mathcal{F}_t] \leq \|\boldsymbol{\theta}_t - \boldsymbol{\theta}^\star\| - (1-\varepsilon[1-(1-\varepsilon)\beta_t])\alpha_t\,\|\boldsymbol{\theta}_t - \boldsymbol{\theta}^\star\|. \qquad (18)
$$

Applying Theorem 3 to Equation (18), we find that $\|\boldsymbol{\theta}_t - \boldsymbol{\theta}^\star\|$ converges to a limit with probability 1 and:

$$
\sum_{t=0}^{\infty}(1-\varepsilon[1-(1-\varepsilon)\beta_t])\,\|\boldsymbol{\theta}_t - \boldsymbol{\theta}^\star\| < \infty,
$$

Assume that $\|\boldsymbol{\theta}_t - \boldsymbol{\theta}^\star\|$ converges to a non-zero limit, then

$$
\sum_{t=0}^{\infty}(1-\varepsilon[1-(1-\varepsilon)\beta_t])\,\|\boldsymbol{\theta}_t - \boldsymbol{\theta}^\star\| = \infty.
$$

which is a contradiction. To conclude, $\|\boldsymbol{\theta}_t - \boldsymbol{\theta}^\star\|$ converges to 0 with probability 1, or equivalently, $\boldsymbol{\theta}_t$ converges the optimal solution $\boldsymbol{\theta}^\star$ with probability 1. □