Online Learning of Feasible Strategies in Unknown Environments

Santiago Paternain and Alejandro Ribeiro

***Abstract*—Define an environment as a set of convex constraint functions that vary arbitrarily over time and consider a cost function that is also convex and arbitrarily varying. Agents that operate in this environment intend to select actions that are feasible for all times while minimizing the cost's time average. Such action is said optimal and can be computed *offline* if the cost and the environment are known a priori. An *online* policy is one that depends causally on the cost and the environment. To compare online policies to the optimal offline action define the fit of a trajectory as a vector that integrates the constraint violations over time and its regret as the cost difference with the optimal action accumulated over time. Fit measures the extent to which an online policy succeeds in learning feasible actions while regret measures its success in learning optimal actions. This paper proposes the use of online policies computed from a saddle point controller. This controller pushes actions along a linear combination of the negative gradients of the constraints and objective, while dynamically adapting the coefficients of this linear combination to find appropriate weightings. It is shown that this controller produces policies with fit and regret that are either bounded or grow at a sublinear rate. These properties provide an indication that the controller finds trajectories that are feasible and optimal in a relaxed sense. Concepts are illustrated throughout with the problem of a shepherd that wants to stay close to all sheep in a herd. Numerical experiments show that the saddle point controller allows the shepherd to do so.**

I. Introduction

A shepherd wants to stay close to a herd of sheep while also staying as close as possible to a preferred sheep. The movements of the sheep, including the preferred, are unknown a priori and arbitrary, perhaps strategic. However, their time varying positions are such that it is possible for the shepherd to stay within a prescribed distance of all of them. The shepherd observes the sheep movements and responds to this online information through a causal dynamical system. This paper shows that an online version of the saddle point algorithm of Arrow and Hurwicz [1] succeeds in keeping the shepherd close to all sheep while maintaining a distance to the preferred sheep that is not much worse that the distance he would maintain had he known the sheep's paths a priori.

More generically, we consider an agent that operates in an environment that we define as a set of time varying functions of the agent's actions – the distance between the sheep and the shepherd – as well as a cost function that is also time varying and dependent on the agent's actions – the distance to the preferred sheep. Since these functions are unknown a priori the agent operates *online* by responding causally to observations of the cost and environment. The goodness of an online policy is determined by comparing to the optimal action chosen *offline* by a clairvoyant agent that has prescient access to cost and environment. We therefore define a viable environment as one in which the constraints, if known, are satisfiable over time – whatever the sheep do, the shepherd can position himself close to all of them – and the fit as the time accumulation of the constraint violations incurred by an online policy – the integrals of the distance of the shepherd to each sheep when making causal decisions. Likewise, we define the optimal action as the one that minimizes the cost aggregated over time – the smallest possible average distance between the shepherd and the preferred sheep when given the benefit of hindsight – and the regret as the time integral of the difference between the cost attained online and this optimal cost – how much worse the shepherd does when not having the benefit of hindsight.

The problem of operating in unknown convex environments with unknown costs generalizes operation in known environments with known costs, which in turn generalizes plain cost minimization. The latter is a canonical problem that can be solved locally with extremum seeking gradient descent controllers that push the agent along the negative gradient of the cost function; see e.g., [2]–[6]. These algorithms converge to local minima in general and to the global minimum when the cost is convex. These costs can represent natural constraints or artificial potentials and are common methodologies to solve, e.g., navigation problems [7]–[12]. If uncertainties are present in the environment, stochastic gradient descent algorithms can be used instead with similar convergence guarantees [13]–[16]. When adding known constraints to cost minimization we can add barrier potentials, or, more germane to the methodology advocated in this paper, pose the problem as the determination of a saddle point of a Lagrangian function. This saddle point can be found with the saddle point algorithm of Arrow and Hurwicz which interprets each constraint as a separate potential and descends on a linear combination of their negative gradients [1]. The coefficients of these linear combinations are multipliers that adapt dynamically so as to push the agent to the optimal solution in the region where all constraints are satisfied. Saddle point algorithms and variations have been widely studied [17]–[19] and used in various domains such as decentralized control [20], [21] and image processing, see e.g. [22]. As in the case of extremum seeking algorithms, saddle point methods require prior knowledge of costs and constraints.

The novelty of this work is to consider constraints and costs

Work in this paper is supported by NSF CCF-0952867 and ONR N00014-12-1-0997. The authors are with the Department of Electrical and Systems Engineering, University of Pennsylvania, 200 South 33rd Street, Philadelphia, PA 19104. Email: {spater, aribeiro}@seas.upenn.edu.

that are unknown a priori and can change arbitrarily over time. In this case, cost minimization can be formulated in the language of regret [23], [24] whereby agents operate online by selecting plays that incur a cost selected by nature. The cost functions are revealed to the agent ex post and used to adapt subsequent plays. It is a quite remarkable fact that an online version of gradient descent is able to find plays whose regret grows at a sublinear rate when the cost is a convex function [25], [26] – therefore suggesting vanishing per-play penalties of online plays with respect to the clairvoyant play. Our main contribution is to show that an online saddle point algorithm that observes costs and constraints ex post succeeds in finding policies with regret and fit that, at worst, grow at a sublinear rate – and may even stay bounded with more stringent hypotheses.

The online learning of strategies that are feasible with respect to an unknown and arbitrarily varying environment is formulated here in the language of fit, which we define as a vector that accumulates the violation of each constraint over time (Section II). To clarify the connection with existing regret literature, we begin the technical part of the paper with the derivation of a projected gradient controller to minimize an unknown cost in an environment without constraints (Section III). We then move on the main part of the paper in which we propose to control fit and regret growth with the use of an online saddle point controller that moves along a linear combination of the negative gradients of the instantaneous constraints and the objective function. The coefficients of these linear combinations are adapted dynamically as per the instantaneous constraint functions as well (Section IV). This online saddle point controller is a generalization of (offline) saddle point in the same sense that an online gradient controller generalizes (offline) gradient descent. We show that if there exists a viable action that can satisfy the environmental constraints at all times, the online saddle point controller achieves bounded fit if optimality is not of interest (Theorem 2). When optimality is considered, the controller achieves bounded regret and the fit grows sublinearly with the time horizon (Theorem 3). Throughout the paper we illustrate concepts with the problem of a shepherd that has to stay close to a herd of sheep (Section II-B). A numerical analysis of this problem closes the paper (Section V) except for concluding remarks (Section VI).

Notation. A multivalued function $f : \mathbb{R}^n \to \mathbb{R}^m$ is defined by stacking the components functions, i.e., $f := [f_1, \dots, f_m]^T$. The notation $\int f(x)dx := [\int f_1(x)dx, \dots, \int f_m(x)dx]^T$ represents a vector stacking each individual integral. An inequality $x \leq y$ between vectors of equal dimension $x, y \in \mathbb{R}^n$ is interpreted componentwise. An inequality $x \leq c$ between a vector $x = [x_1, \dots, x_n]^T \in \mathbb{R}^n$ and a scalar $c \in \mathbb{R}$ means that $x_i \leq c$ for all components of x.

II. Viability, feasibility and optimality

We consider a continuous time environment in which an agent selects an action that results in a time varying set of penalties. Use t to denote time and let $X \subseteq \mathbb{R}^n$ be a closed convex set from which the agent selects action $x \in X$. The penalties incurred at time t for selected action x are given by the value $f(t, x)$ of the vector function $f : \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}^m$. We interpret the vector penalty function f as a definition of the environment. Our interest in this paper is in situations where the agent is faced with an environment f and must choose an action $x \in X$ – or perhaps a trajectory $x(t)$ – that guarantees nonpositive penalties $f(t, x(t)) \leq 0$ for all times t not exceeding a time horizon T. Since the existence of this trajectory depends on the specific environment we start by defining a viable environment as one in which it is possible for the agent to select an action with nonpositive penalty for times $0 \leq t \leq T$ as we formally specify next.

Definition 1 (Viable environment). *We say that a given environment $f : \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}^m$ is viable over the time horizon T for an agent that selects actions $x \in X$ if there exists an action $x^\dagger \in X$ such that*

$$f(t, x^\dagger) \leq 0, \quad \textit{for all } t \in [0, T]. \tag{1}$$

An action $x^\dagger$ satisfying (1) *is said feasible and the set $X^\dagger := \{x^\dagger \in X : f(t, x^\dagger) \leq 0, \textit{ for all } t \in [0, T]\}$ is termed the feasible set of actions.*

Since for a viable environment it is possible to have multiple feasible actions it is desirable to select one that is optimal with respect to some criterion of interest. Introduce then the objective function $f_0 : \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}$, where for a given time $t \in [0, T]$ and action $x \in X$ the agent suffers a loss $f_0(t, x)$. The optimal action is defined as the one that minimizes the accumulated loss $\int_0^T f_0(t, x)\, dt$ among all viable actions, i.e.,

$$\begin{aligned} x^* := \operatorname*{argmin}_{x \in X} & \int_0^T f_0(t, x)\, dt \\ \text{s.t. } & f(t, x) \leq 0, \text{ for all } t \in [0, T]. \end{aligned} \tag{2}$$

For the definition in (2) to be valid the function $f_0(t, x)$ has to be integrable with respect to t. In subsequent definitions and analyses we also require integrability of the environment f as well as convexity with respect to x as we formally state next.

Assumption 1. The functions $f(t, x)$ and $f_0(t, x)$ are integrable with respect to t in the interval $[0, T]$.

Assumption 2. The functions $f(t, x)$ and $f_0(t, x)$ are convex with respect to x for all times $t \in [0, T]$.

If the environment f and functions f_0 are known beforehand, the question of finding the action in a viable environment that minimizes the total aggregate cost is equivalent to solving the constrained convex optimization problem in (2). A number of algorithms are known to solve this problem. Here, we consider the problem of adapting a strategy $x(t)$ when the functions $f(t, x)$ and $f_0(t, x)$ are *arbitrary* and *revealed causally*. I.e., we want to choose the action $x(t)$ using observations of viability $f(t, x)$ and cost $f_0(t, x)$ in the open interval $[0, t)$. This implies that $f(t, x(t))$ and $f_0(t, x(t))$ are not observed before choosing $x(t)$. The action $x(t)$ is chosen ex ante and the corresponding viability $f(t, x(t))$ and cost $f_0(t, x(t))$ are incurred ex post.

A. Regret and fit

We evaluate the performance of trajectories $x(t)$ through the concepts of regret and fit. To define regret we compare the accumulated cost $\int_0^T f_0(t, x(t))\, dt$ incurred by $x(t)$ with the cost that would had been incurred by the optimal action x^* defined in (2),

$$\mathcal{R}_T := \int_0^T f_0(t, x(t))\, dt - \int_0^T f_0(t, x^*)\, dt. \tag{3}$$

Analogously, we define the fit of the trajectory $x(t)$ as the accumulated value of the penalties $f(t, x(t))$ incurred for times $t \in [0, T]$,

$$\mathcal{F}_T := \int_0^T f(t, x(t))\, dt. \tag{4}$$

The regret $\mathcal{R}_T$ and fit $\mathcal{F}_T$ can be interpreted as performance losses associated with online causal operation as opposed to offline clairvoyant operation. If the fit $\mathcal{F}_T$ is positive in a viable environment we are in a situation in which, had the environment f be known a priori, we could have selected an action $x^\dagger$ with $f(t, x^\dagger) \leq 0$. The fit measures how far the trajectory $x(t)$ comes from achieving that goal. Likewise, if the regret $\mathcal{R}_T$ is large we are in a situation in which prior knowledge of environment and cost would had resulted in the selection of the much better action x^* – and in that sense $\mathcal{R}_T$ indicates how much we regret not having had that information available.

A good learning strategy is one in which $x(t)$ approaches x^*. In that case, the regret and fit grow for small T but eventually stabilize or, at worst, grow at a sublinear rate. Considering regret $\mathcal{R}_T$ and fit $\mathcal{F}_T$ separately, this observation motivates the definitions of feasible trajectories, strong feasible trajectories, and strong optimal trajectories that we formally state next.

Definition 2. *Given an environment $f : \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}^m$, a cost function $f_0 : \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}$, and a trajectory $x(t)$ we say that:*

Feasibility. *The trajectory $x(t)$ is feasible in the environment if the fit $\mathcal{F}_T$ grows sublinearly with T. I.e., if there exist a function $h(T)$ with $\limsup_{T\to\infty} h(T)/T = 0$ and a constant vector C such that for all times T it holds,*

$$\mathcal{F}_T := \int_0^T f(t, x(t))\, dt \leq C h(T). \tag{5}$$

Strong Feasibility. *The trajectory $x(t)$ is strongly feasible in the environment if the fit $\mathcal{F}_T$ is bounded for all T. I.e., if there exists a constant vector C such that for all times T it holds,*

$$\mathcal{F}_T := \int_0^T f(t, x(t))\, dt \leq C. \tag{6}$$

Strong optimality. *The trajectory $x(t)$ is strongly optimal in the environment if the regret $\mathcal{R}_T$ is bounded for all T. I.e., if there exists a constant C such that for all times T it holds,*

$$\mathcal{R}_T := \int_0^T f_0(t, x(t))\, dt - \int_0^T f_0(t, x^*)\, dt \leq C. \tag{7}$$

Remark 1 (Not every trajectory is strongly feasible). Notice that in definition (6) we are considering the integral of a measurable function in a finite interval, hence the integral will always be bounded by a constant, yet the constant could be dependent on the time horizon T. If this is the case, the trajectory is not strongly feasible, because the integral has to be uniformly bounded by a constant for all time horizons T in order to meet the definition. The same remark is valid for the definitions of strongly optimal and feasible.

Having the regret satisfy $\mathcal{R}_T \leq C$ irrespectively of T is an indication that $f_0(t, x(t))$ is close to $f_0(t, x^*)$ so that the integral stops growing. This is not necessarily so because we can also achieve small regret by having $f_0(t, x(t))$ oscillate above and below $f_0(t, x^*)$ so that positive and negative values of $f_0(t, x(t)) - f_0(t, x^*)$ cancel out. In general, the possibility of having small regret by a trajectory that does not approach x^* is a limitation of the concept of regret. Alternatively, we can think of the optimal offline policy x^* as fixing a budget for cost accumulated across time. An optimal online policy meets that budget within a constant factor C – perhaps by overspending at some times and underspending at some other times.

Likewise, when the fit satisfies $\mathcal{F}_T \leq C$ irrespectively of T, it suggests that $x(t)$ approaches the feasible set. This need not be true as it is possible to achieve bounded fit by having $f(t, x(t))$ oscillate around 0. Thus, as in the case of regret, we can interpret strongly feasible trajectories as meeting the *accumulated* budget $\int_0^T f(t, x(t))\, dt \leq 0$ within a constant factor C. This is in contrast with feasible actions $x^\dagger$ that meet the budget $f(t, x^\dagger) \leq 0$ for all times. Feasible trajectories differ from strongly feasible trajectories in that the fit is allowed to grow at a sublinear rate. This means that feasible trajectories do not meet the *accumulated* budget $\int_0^T f(t, x(t))\, dt \leq 0$ within a constant C but do meet the *time averaged* budget $(1/T)\int_0^T f(t, x(t))\, dt \leq 0$ within that constant. The notion of optimality – as opposed to strong optimality – could have been defined as a case in which regret is bounded by a sublinear function of T. This is not necessary here because all of our results state strong optimality.

In this work we solve three different problems: (i) Finding strongly optimal trajectories in unconstrained environments. (ii) Finding strongly feasible trajectories. (iii) Finding feasible, strongly optimal trajectories. We develop these solutions in sections III, IV-A, and IV-B, respectively. Before that, we clarify concepts with the introduction of an example.

B. The shepherd problem

Consider a target tracking problem in which an agent – the shepherd – follows a group of m targets – the sheep. Specifically, let $z(t) = [z_1(t), z_2(t)]^T \in \mathbb{R}^2$ denote the position of the shepherd at time t. To model smooth paths for the shepherd introduce a polynomial parameterization so that each of the position components $z_k(t)$ can be written as

$$z_k(t) = \sum_{j=0}^{n-1} x_{kj} p_j(t), \tag{8}$$

where $p_j(t)$ are polynomials that parameterize the space of possible trajectories. The action space of the shepherd is then given by the vector $x = [x_{10}, \ldots, x_{1,n-1}, x_{20}, \ldots, x_{2,n-1}]^T \in \mathbb{R}^{2n}$ that stacks the coefficients of the parameterization in (8).

Further define $y_i(t) = [y_{i1}(t), y_{i2}(t)]^T$ as the position of the ith sheep at time t for $i = 1, \ldots, m$ and introduce a maximum allowable distance r_i between the shepherd and each of the sheep . The goal of the shepherd is to find a path $z(t)$ that is within distance r_i of sheep i for all sheep. This can be captured by defining an m-dimensional environment f with each component function f_i defined as

$$f_i(t, x) = \|z(t) - y_i(t)\|^2 - r_i^2 \quad \text{for all} \quad i = 1..m. \tag{9}$$

That the environment defined by (9) is viable means that it is possible to select a vector of coefficients x so that the shepherd's trajectory generated by (8) stays close to all sheep for all times. To the extent that (8) is a loose parameterization – we can approximate arbitrary functions with sufficiently large index n –, this simply means that the sheep are sufficiently close to each other at all times. E.g., if $r_i = r$ for all times, viability is equivalent to having a maximum separation between sheep smaller than $2r$.

As an example of a problem with an optimality criterion say that the first target – the black sheep – is preferred in that the shepherd wants to stay as close as possible to it. We can accomplish that by introducing the objective function

$$f_0(t, x) = \|z(t) - y_1(t)\|^2. \tag{10}$$

Alternatively, we can require the (lazy) shepherd to minimize the work required to follow the sheep. This behavior can be induced by minimizing the integral of the acceleration which in turn can be accomplished by defining the optimality criterion [cf. (2)],

$$f_0(t, x) = \left\|\ddot{z}(t)\right\| = \left\| \left[\sum_{j=0}^{n-1} x_{1j} \ddot{p}_j(t), \ \sum_{j=0}^{n-1} x_{2j} \ddot{p}_j(t) \right] \right\|. \tag{11}$$

Trajectories $x(t)$ differ from actions in that they are allowed to change over time, i.e., the constant values x_{kj} in (8) are replaced by the time varying values $x_{kj}(t)$. A feasible or strongly feasible trajectory $x(t)$ means that the shepherd is repositioning to stay close to all sheep. An optimal trajectory with respect to (10) is one in which he does so while staying as close as possible to the black sheep. An optimal trajectory with respect to (11) is one in which the work required to follow the sheep is minimized. In all three cases we apply the usual caveat that small fit and regret may be achieved with stretches of underachievement following stretches of overachievement.

III. Unconstrained regret in continuous time.

Before considering the feasibility problem we consider the following unconstrained minimization problem. Given an unconstrained environment ($f(t, x) \equiv 0$) our goal is to generate strong optimal trajectories $x(t)$ in the sense of Definition 2, selecting actions from a closed convex set X i.e. $x(t) \in X$ for all $t \in [0, T]$. Given the convexity of the objective function with respect to the action, as per Assumption 2, it is natural to consider a gradient descent controller. To avoid restricting attention to functions that are differentiable with respect to x, we introduce the notion of subgradient that we formally define next.

Definition 3 (Subgradient). *Let $g : X \to \mathbb{R}$, be a convex function where $X \subset \mathbb{R}^n$. Then g_x is a subgradient of g at a point $x \in X$ if*

$$g(y) \geq g(x) + g_x(x)^T (y - x) \quad \textit{for all} \quad y \in X \tag{12}$$

In general, subgradients are defined at all points for all convex functions. At the points where the function f is differentiable the subgradient and the gradient coincide. In the case of vector functions $f : \mathbb{R}^n \to \mathbb{R}^m$ we group the subgradients of each component into a subgradient matrix $f_x(x) \in \mathbb{R}^{n \times m}$ that we define as

$$f_x(x) = \left[\begin{array}{cccc} f_{1,x}(x) & f_{2,x}(x) & \cdots & f_{m,x}(x) \end{array} \right] \tag{13}$$

where $f_{i,x}(x)$ is a subgradient of $f_i(x)$ as per Definition 3. In addition, since the action must always be selected from the set X we define the controller in a way that the actions are the solution of a projected dynamical system over the set X. The solution has been studied in [27], [28] and we define the notion as follow.

Definition 4 (Projected dynamical system)**.** *Let X be a closed convex set.*

Projection of a point. *For any $z \in R^n$, there exits a unique element in X, denoted $P_X(z)$ such that*

$$P_X(z) = \arg \inf_{y \in X} \|y - z\|. \tag{14}$$

Projection of a vector at a point. *Let $x \in X$ and v a vector, we define the projection of v over the set X at the point x, $\Pi_X(x, v)$ as*

$$\Pi_X(x, v) = \lim_{\delta \to 0^+} \left(P_X(x + \delta v) - x \right) / \delta. \tag{15}$$

As it is demonstrated in Lemma 3, the projection of a vector at a point over a set is equivalent to project the vector over the smallest cone containing the set X with vertex at the point x.

Projected dynamical system. *Given a closed convex set X and a vector field $F(t, x)$ which takes elements from $\mathbb{R} \times X$ into $\mathbb{R}^n$ the projected differential equation associated with X and F is defined to be*

$$\dot{x}(t) = \Pi_X \left(x, F(t, x) \right). \tag{16}$$

In the above projection if the point x is in the interior of X then the projection is not different from the original vector field i.e. $\Pi_X(x, F(t, x)) = F(t, x)$. On the other hand if the point x is in the border of X, then the projection is just the component of the vector field that is tangential to the set X at the point x. Let's consider for instance the case where the set X is a box in R^n. Let $X = [a_1, b_1] \times ... \times [a_n, b_n]$ where $a_1..a_n$ and $b_1...b_n$ are real numbers. Then for each component

of the vector field we have that

$$\Pi_X\left(x, F(t,x)\right)_i \begin{cases} 0 \quad \text{if} \quad x_i = a_i \quad \text{and} \quad F(t,x)_i < 0, \\ 0 \quad \text{if} \quad x_i = b_i \quad \text{and} \quad F(t,x)_i > 0, \\ F(t,x)_i \quad \text{otherwise.} \end{cases} \tag{17}$$

Therefore, when the projection is included, the proposed controller takes the form of the following projected dynamical system:

$$\dot{x} = \Pi_X\left(x, -\varepsilon f_{0,x}(t,x)\right). \tag{18}$$

Before stating the first theorem we need a Lemma concerning the relation between the original vector field and the projected vector field. This lemma is used in the proofs of theorems 1, 2 and 3.

Lemma 1. *Let X be a convex set and $x_0 \in X$ and $x \in X$. Then*

$$(x_0 - x)^T \Pi_X(x_0, v) \leq (x_0 - x)^T v. \tag{19}$$

Proof. See Apendix A. ■

Let's define an Energy function $V_{\bar{x}} : \mathbb{R}^n \to \mathbb{R}$ as

$$V_{\bar{x}}(x) = \frac{1}{2}(x - \bar{x})^T (x - \bar{x}). \tag{20}$$

Where $\bar{x} \in X \subset \mathbb{R}^n$ is an arbitrary fixed action. We are now in conditions to present the first theorem, which states that the gradient controller defined in (18) gives origin to strong optimal trajectories i.e. with bounded regret for all T.

Theorem 1. *Let $f_0 : \mathbb{R} \times X \to \mathbb{R}$ be cost function satisfying assumptions 1 and 2, with $X \subseteq \mathbb{R}^n$ convex. The trajectory $x(t)$ generated by the online projected gradient controller in* (18) *is strongly optimal in the sense of Definition 2. In particular, the regret $\mathcal{R}_T$ can be bounded by*

$$\mathcal{R}_T \leq V_{x^*}\left(x(0)\right)/\varepsilon, \quad \textit{for all } T \tag{21}$$

where $V_{\bar{x}}$ is the Energy function in (20)*.*

Proof. Consider an action trajectory $x(t)$, an arbitrary given action $\bar{x} \in X$, and the corresponding energy function $V_{\bar{x}}(x(t))$ as per (20). The derivative $\dot{V}_{\bar{x}}(x(t))$ of the energy function with respect to time is then given by

$$\dot{V}_{\bar{x}}(x(t)) = (x(t) - \bar{x})^T \dot{x}(t). \tag{22}$$

If the trajectory $x(t)$ follows from the online projected gradient dynamical system in (18) we can substitute the trajectory derivative $\dot{x}$ by the vector field value and reduce (22) to

$$\dot{V}_{\bar{x}}(x(t)) = (x(t) - \bar{x})^T \Pi_X\left(x(t), -\varepsilon f_{0,x}(t, x(t))\right). \tag{23}$$

Use now the result in Lemma 1 with $v = -\varepsilon f_{0,x}(t, x(t))$ to remove the projection operator from (23) and write

$$\dot{V}_{\bar{x}}(x(t)) \leq -\varepsilon (x(t) - \bar{x})^T f_{0,x}(t, x(t)). \tag{24}$$

Using the definition of subgradient (c.f. Definition 3), we can upper bound the inner product $-(x(t) - \bar{x})^T f_{0,x}(t, x(t))$ by the difference $f_0(t, \bar{x}) - f_0(t, x(t))$ and transform (24) into

$$\dot{V}_{\bar{x}}(x(t)) \leq \varepsilon \left(f_0(t, \bar{x}) - f_0(t, x(t))\right). \tag{25}$$

Rearranging terms in the preceding inequality and integrating over time yields

$$\int_0^T f_0(t, x(t))\, dt - \int_0^T f_0(t, \bar{x})\, dt \leq -\frac{1}{\varepsilon} \int_0^T \dot{V}_{\bar{x}}(x(t))\, dt. \tag{26}$$

Since the primitive of $\dot{V}_{\bar{x}}(x(t))$ is $V_{\bar{x}}(x(t))$ we can evaluate the integral on the right hand side of (26) and further use the fact that $V_{\bar{x}}(x) \geq 0$ for all $x \in \mathbb{R}^n$ to conclude that

$$-\int_0^T \dot{V}_{\bar{x}}(x(t)) dt \;=\; V_{\bar{x}}(x(0)) - V_{\bar{x}}(x(T)) \;\leq\; V_{\bar{x}}\left(x(0)\right). \tag{27}$$

Combining the bounds in (26) and (27) we have that

$$\int_0^T f_0(t, x(t))\, dt - \int_0^T f_0(t, \bar{x})\, dt \leq V_{\bar{x}}(x(0))/\varepsilon. \tag{28}$$

Since the above inequality holds for an arbitrary point $\bar{x} \in \mathbb{R}^n$ it holds for $\bar{x} = x^*$ in particular. When making $\bar{x} = x^*$ in (28) the left had side reduces to the regret $\mathcal{R}_T$ associated with the trajectory $x(t)$ [cf. (3)] and in the right hand side we have $V_{\bar{x}}(x(0))/\varepsilon = V_{x^*}(x(0))/\varepsilon$. Eq. (21) follows because (28) is true for all times T. This implies that the trajectory is strongly optimal according to (7) in Definition 2. ■

The strong optimality of the online projected gradient controller in (18) that we claim in Theorem 1 is not a straightforward generalization of the optimality of gradient controllers in constant convex potentials. The functions f_0 are allowed to change arbitrarily over time and are not observed until after the cost $f_0(t, x(t))$ has been incurred.

Since the initial value of the Energy function $V_{x^*}(x(0))$ is the square of the distance between $x(0)$ and x^*, the regret bound in (21) shows that the closer we start to the optimal point the smaller the accumulated cost is. Likewise, the larger the controller gain ϵ, the smaller the regret bound is. Theoretically, increasing ε we can make the regret bound arbitrarily small. This is not possible in practice because larger ε entails trajectories with larger derivatives which cannot be implemented in systems with physical constraints. In the example in Section II-B the derivatives of the state $x(t)$ control the speed and acceleration of the shepherd. The physical limits of these quantities along with an upper bound on the cost gradient $f_{0,x}(t, x)$ can be used to estimate the largest allowable gain ε.

Remark 2. In discrete time systems where t is a natural variable and the integrals in (3) are replaced by analogous sums, online gradient descent algorithms are used to reduce regret; see e.g. [25], [26]. The online gradient controller in (18) is a direct generalization of online gradient descent to continuous time. This similarity notwithstanding, the result in Theorem 1 is stronger than the corresponding regret bound in discrete time which states a sublinear growth at a rate not faster than $\sqrt{T}$ if the stepsize of the algorithm is constant [25], and $\log T$ with a variable stepsize [26].

IV. Saddle point algorithm

Given an environment $f(t, x)$ and an objective function $f_0(t, x)$ verifying assumptions 1 and 2 we set our attention

towards two different problems: design a controller that gives origin to strongly feasible trajectories and a controller that gives origin to feasible and strongly optimal trajectories. As already noted, when the environment is known beforehand the problem of finding such trajectories is a constrained convex optimization problem, which we can solve using the saddle point algorithm of Arrow and Hurwicz [1]. Following this idea, let $\lambda \in \Lambda = \mathbb{R}^m_+$, be a multiplier and define the time-varying Lagrangian associated with the online problem as

$$\mathcal{L}(t, x, \lambda) = f_0(t, x) + \lambda^T f(t, x). \tag{29}$$

Saddle point methods rely on the fact that for a constrained convex optimization problem, a pair is a primal-dual optimal solution if and only if the pair is a saddle point of the Lagrangian associated with the problem; see e.g. [29]. The main idea of the algorithm is then to generate trajectories that descend in the opposite direction of the gradient of the Lagrangian with respect to x and that ascend in the direction of the gradient with respect to λ.

Since the Lagrangian is differentiable with respect to λ, we denote by $\mathcal{L}_\lambda(t, x, \lambda) = f(t, x)$ the derivative of the Lagrangian with respect to λ. On the other hand, since the functions $f_0(\cdot, x)$ and $f(\cdot, x)$ are convex, the Lagrangian is also convex with respect to x. Thus, its subgradient with respect to x always exist, let us denote it by $\mathcal{L}_x(t, x, \lambda)$. Let ε be the gain of the controller, then following the ideas in [1] we define a controller that descends in the direction of the subgradient with respect to the action x

$$\begin{aligned} \dot{x} &= \Pi_X\left(x, -\varepsilon \mathcal{L}_x(t, x, \lambda)\right) \\ &= \Pi_X\left(x, -\varepsilon(f_{0,x}(t, x) + f_x(t, x)\lambda)\right), \end{aligned} \tag{30}$$

and that ascends in the direction of the subgradient with respect to the multiplier λ

$$\dot{\lambda} = \Pi_\Lambda\left(\lambda, \varepsilon \mathcal{L}_\lambda(t, x, \lambda)\right) = \Pi_\Lambda\left(\lambda, \varepsilon f(t, x)\right). \tag{31}$$

The projection over the set X in (30) is done to assure that the trajectory is always in the set of possible actions. The projection concerning the dual variable λ in (31) is done to assure that $\lambda(t) \in \mathbb{R}^m_+$ for all times $t \in [0, T]$. An important observation regarding (30) and (31) is that the environment is observed locally in space and causally in time. The values of the environment constraints and its subgradients are observed at the current trajectory position $x(t)$ and the values of $f(t, x(t))$ and $f_x(t, x(t))$ affect the derivatives of $x(t)$ and $\lambda(t)$ only. Notice that if the environment function satisfies $f(t, x) \equiv 0$ we recover the algorithm defined in (18) as a particular case of the saddle point controller.

A block diagram for the controller in (30) - (31) is shown in Figure 1. The controller operates in an environment to which it inputs at time t an action $x(t)$ that results in a penalty $f(t, x(t))$ and cost $f_0(t, x(t))$. The value of these functions and their subgradients $f_x(t, x(t))$ and $f_{0,x}(t, x(t))$ are observed and fed to the multiplier and action feedback loops. The action feedback loop behaves like a weighted gradient descent controller. We move in the direction given by a linear combination of the the gradient of the objective function $f_{0,x}(t, x(t))$ and the constraint subgradients $f_i(t, x(t))$

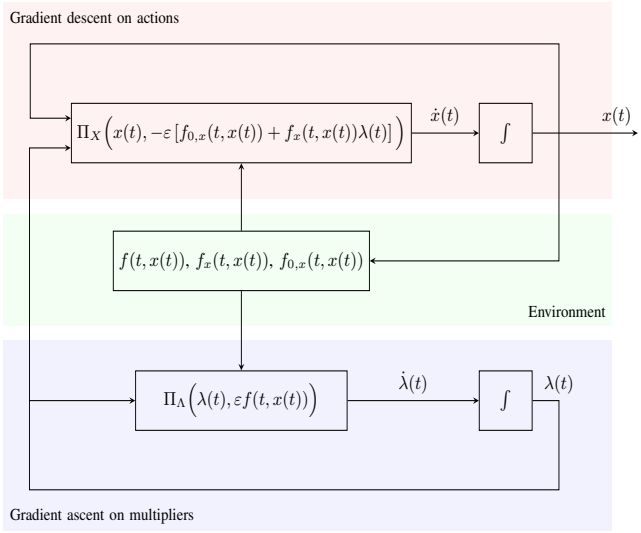


Fig. 1: Block diagram of the saddle point controller. Once that action $x(t)$ is selected at time t, we measure the corresponding values of $f(t, x)$, $f_x(t, x)$ and $f_{0,x}(t, x)$. This information is fed to the two feedback loops. The action loop defines the descent direction by computing weighted averages of the subgradients $f_x(t, x)$ and $f_{0,x}(t, x)$. The multiplier loop uses $f(t, x)$ to update the corresponding weights.

weighted by their corresponding multipliers $\lambda_i(t)$. Intuitively, this pushes $x(t)$ towards satisfying the constraints and to the minimum of the objective function in the set where constraints are satisfied. However, the question remains of how much weight to give to each constraint. This is the task of the multiplier feedback loop. When constraint i is violated we have $f_i(t, x(t)) > 0$. This pushes the multiplier $\lambda_i(t)$ up, thereby increasing the force $\lambda_i(t) f_i(t, x(t))$ pushing $x(t)$ towards satisfying the constraint. If the constraint is satisfied, we have $f_i(t, x(t)) < 0$, the multiplier $\lambda_i(t)$ being decreased, and the corresponding force decreasing. The more that constraint i is violated, the faster we increase the multiplier, and the more we increase the force that pushes $x(t)$ towards satisfying $f_i(t, x(t)) < 0$. If the constraint is satisfied, the force is decreased and may eventually vanish altogether if we reach the point of making $\lambda_i(t) = 0$.

A. *Strongly feasible trajectories*

We begin by studying the saddle point controller defined by (30) and (31) in a problem in which optimality is *not* taken into account. In this case the action descent equation of the controller (30) takes the form:

$$\dot{x} = \Pi_X\left(x, -\varepsilon \mathcal{L}_x(t, x, \lambda)\right) = \Pi_X\left(x, -\varepsilon f_x(t, x)\lambda\right), \tag{32}$$

while the multiplier ascent equation (31) remains unchanged. The bounds to be derived for the fit ensure that the trajectories $x(t)$ are strongly feasible in the sense of Definition 2. To state the result consider an arbitrary fixed action $\bar{x} \in X$ and an arbitrary multiplier $\bar{\lambda} \in \Lambda$ and define the energy function

$$V_{\bar{x},\bar{\lambda}}(x, \lambda) = \frac{1}{2}\left(||x - \bar{x}||^2 + ||\lambda - \bar{\lambda}||^2\right). \tag{33}$$

We can then bound fit in terms of the initial value $V_{\bar{x},\bar{\lambda}}(x(0), \lambda(0))$ of the energy function for properly chosen $\bar{x}$ and $\bar{\lambda}$ as we formally state next.

Theorem 2. *Let $f : \mathbb{R} \times X \to \mathbb{R}^m$, satisfying assumptions 1 and 2, where $X \subseteq \mathbb{R}^n$ is a convex set. If the environment is viable, then the controller defined by* (32) *and* (31) *gives origin to strongly feasible trajectories $x(t)$ for all $T > 0$. Specifically, the fit is bounded by*

$$\mathcal{F}_{T,i} \leq \frac{1}{\varepsilon} V_{x^\dagger, e_i}(x(0), \lambda(0)), \tag{34}$$

where $x^\dagger$ is any point that belongs to the feasible set $X^\dagger$, and e_i with $i = 1..m$ are the vectors of the canonical base of $\mathbb{R}^m$.

Proof. Consider action trajectories $x(t)$ and multiplier trajectories $\lambda(t)$ and the corresponding energy function $V_{\bar{x},\bar{\lambda}}(x(t), \lambda(t))$ in (33) for arbitrary given action $\bar{x} \in X$ and multiplier $\bar{\lambda} \in \Lambda$. The derivative $\dot{V}_{\bar{x},\bar{\lambda}}(x(t), \lambda(t))$ of the energy with respect to time is then given by

$$\dot{V}_{\bar{x},\bar{\lambda}}(x(t), \lambda(t)) = (x(t) - \bar{x})^T \dot{x}(t) + (\lambda(t) - \bar{\lambda})^T \dot{\lambda}(t). \tag{35}$$

If the trajectories $x(t)$ and $\lambda(t)$ follow from the saddle point dynamical system given by (32) and (31) we can substitute the action and multiplier derivatives by their corresponding values and reduce(35) to

$$\begin{aligned}\dot{V}_{\bar{x},\bar{\lambda}}(x(t), \lambda(t)) =& (x(t) - \bar{x})^T \Pi_X \left(x, -\varepsilon f_x(t, x(t))\right) \lambda(t) \\ &+ (\lambda(t) - \bar{\lambda})^T \Pi_\Lambda \left(x, \varepsilon f(t, x(t))\right). \end{aligned} \tag{36}$$

Then, using the result of Lemma 1 for both X and Λ, the following inequality holds:

$$\begin{aligned}\dot{V}_{\bar{x},\bar{\lambda}}(x(t), \lambda(t)) \leq& \varepsilon(\bar{x} - x(t))^T f_x(t, x(t))\lambda(t) \\ &+ \varepsilon(\lambda(t) - \bar{\lambda})^T f(t, x(t)). \end{aligned} \tag{37}$$

Notice that $f(t,x)\lambda(t)$ is a convex function with respect to the action, therefore we can upper bound the inner product $(\bar{x} - x(t))^T f_x(t, x(t))\lambda(t)$ by the quantity $f(t, \bar{x})^T \lambda(t) - f(t, x(t))^T \lambda(t)$ and transform (37) into

$$\begin{aligned}\dot{V}_{\bar{x},\bar{\lambda}}(x(t), \lambda(t)) \leq& \varepsilon \left(f(t, \bar{x}) - f(t, x(t))\right)^T \lambda(t) \\ &+ \varepsilon(\lambda(t) - \bar{\lambda})^T f(t, x(t)). \end{aligned} \tag{38}$$

Further note that in the above equation the second and the third term are opposite. Thus, it reduces to

$$\dot{V}_{\bar{x},\bar{\lambda}}(x(t), \lambda(t)) \leq \varepsilon \left[\lambda^T(t) f(t, \bar{x}) - \bar{\lambda}^T f(t, x(t))\right]. \tag{39}$$

Rewriting the above expression and then integrating both sides with respect to time from $t = 0$ to $t = T$ we obtain

$$\begin{aligned}\varepsilon \int_0^T \bar{\lambda}^T f(t, x(t)) - \lambda^T(t) f(t, \bar{x}) dt& \\ \leq -\int_0^T \dot{V}_{\bar{x},\bar{\lambda}}(x(t), \lambda(t)) dt.& \end{aligned} \tag{40}$$

Integrating the right side of the above equation we obtain

$$\begin{aligned}-\int_0^T \dot{V}_{\bar{x},\bar{\lambda}}(x(t),\lambda(t)) dt& \\ = V_{\bar{x},\bar{\lambda}}(x(0), \lambda(0)) - V_{\bar{x},\bar{\lambda}}(x(T), \lambda(T)),& \end{aligned} \tag{41}$$

and then using the fact that $V_{\bar{x},\bar{\lambda}}(x(t)), \lambda(t)) \geq 0$ for all t we have that

$$-\int_0^T \dot{V}_{\bar{x},\bar{\lambda}}(x(t), \lambda(t)) dt \leq V_{\bar{x},\bar{\lambda}}\left(x(0), \lambda(0)\right). \tag{42}$$

Then, combining (40) and (42), we have that

$$\int_0^T \bar{\lambda}^T f(t, x(t)) - \lambda^T(t) f(t, \bar{x}) dt \leq \left(V_{x^\dagger,\bar{\lambda}}(x(0), \lambda(0))\right)/\varepsilon. \tag{43}$$

Since the environment is viable, there exist a fixed action $x^\dagger$ such that $f(t, x^\dagger) \leq 0$ for all $t \geq 0$. Then choosing $\bar{x} = x^\dagger$, since $\lambda(t) \geq 0$ for all t, we have that

$$\lambda^T(t) f(t, x^\dagger)\, dt \leq 0 \; \forall t \in [0, T]. \tag{44}$$

Therefore the left hand side of (43) can be lower bounded by

$$\bar{\lambda}^T \int_0^T f(t, x(t)) dt \leq \left(V_{x^\dagger,\bar{\lambda}}(x(0), \lambda(0)\right)/\varepsilon. \tag{45}$$

Choosing $\bar{\lambda} = e_i$ where e_i is the ith element of the canonical base of $\mathbb{R}^m$, we have that for all $i = 1..m$:

$$\int_0^T f_i(t, x(t)) dt \leq \left(V_{x^\dagger,e_i}(x(0), \lambda(0))\right)/\varepsilon. \tag{46}$$

The left hand side of the above inequality is the ith component of the fit. Thus, since the m components of the fit of the trajectory generated by the saddle point algorithm are bounded for all T, the trajectory is strongly feasible with the specific upper bound stated in (34). ∎

Theorem 2 assures that if an environment is viable for an agent that selects actions over a set X, the controller defined by (32) and (31) gives origin to a trajectory $x(t)$ that is strongly feasible in the sense of Definition 2. This result is not trivial, since the function f that defines the environment is observed causally and can change arbitrarily over time. In particular, the agent could be faced with an adversarial environment that changes the function f in a way that makes the value of $f(t, x(t))$ larger. The caveat is that the choice of the function f must respect the viability condition that there exists a feasible action $x^\dagger$ such that $f(t, x^\dagger) \leq 0$ for all $t \in [0, T]$. This restriction still leaves significant leeway for strategic behavior. E.g., in the shepherd problem of Section II-B we can allow for strategic sheep that observe the shepherd's movement and respond by separating as much as possible. The strategic action of the sheep are restricted by the condition that the environment remains viable, which in this case reduces to the not so stringent condition that the sheep stay in a ball of radius $2r$ if all $r_i = r$.

Since the initial value of the energy function $V_{x^\dagger,e_i}(x(0), \lambda(0))$ is the square of the distance between $x(0)$ and $x^\dagger$ added to a term that depends on the distance between the initial multiplier and e_i, the fit bound in (32) shows that the closer we start to the feasible set the smaller the accumulated constraint violation becomes. Likewise, the larger the gain ε, the smaller the fit bound is. As in section III we observe that increasing ε can make the fit bound arbitrarily small, yet for the same reasons discussed in that section this can't be done.

Further notice that for the saddle point controller defined by (32) and (31) the action derivatives are proportional not only to the gain ε but to the value of the multiplier λ. Thus, to select gains that are compatible with the system's physical constraints we need to determine upper bounds in

the multiplier values $\lambda(t)$. An upper bound follows as a simple consequence of Theorem 2 if the action set is bounded as we state in the following corollary.

Corollary 1. *Given the controller defined by* (32) *and* (31) *and assuming the same hypothesis of Theorem 2, if the set of actions X is bounded in norm by R, then the multipliers λ are bounded for all time by*

$$0 \leq \lambda_i(t) \leq \left(4R^2 + 1\right), \text{ for all } i = 1, \ldots, m. \tag{47}$$

Proof. First of all notice that according to (31) a projection over the positive orthant is performed for the multiplier update. Therefore, for each component of the multiplier we have that $\lambda_i(t) \geq 0$ for all $t \in [0, T]$. On the other hand, since the trajectory of the multipliers is defined by $\dot{\lambda}(t) = \Pi_\Lambda(\lambda(t), \varepsilon f(t, x(t))$, while $\lambda(t) > 0$ we have that $\dot{\lambda}(t) = \varepsilon f(t, x(t))$. Let t_0 be the first time instant for which $\lambda_i(t) > 0$ for a given $i \in \{1, 2, .., m\}$, i.e.

$$t_0 = \arg \inf_{t \in [0,T]} \{\lambda(t) > 0\}. \tag{48}$$

In addition, let T^* be the first time instant greater than t_0 where $\lambda_i(t) = 0$, if this time is larger than T we set $T^* = T$, i.e.

$$T^* = \max \left\{ \operatorname*{argmin}_{t \in (t_0, T]} \{\lambda_i(t) = 0\}, T \right\}. \tag{49}$$

Therefore, we have that, for any $\tau \in (t_0, T^*]$:

$$\begin{aligned} \int_{t_0}^{\tau} \dot{\lambda}_i(t) dt &= \int_{t_0}^{\tau} \Pi_{\lambda_i}(\lambda(t), \varepsilon f(t, x(t))) dt \\ &= \int_{t_0}^{\tau} \varepsilon f_i(t, x(t)) dt. \end{aligned} \tag{50}$$

Notice that the rightmost side of the above equation is, by definition, proportional to the ith component of the fit restricted to the time interval $[t_0, \tau]$. In Theorem 2 it was proved that the ith component of the fit was bounded for all time horizons by $V_{x^\dagger, e_i}(x(t_0), 0)/\varepsilon$. In this particular case we have that

$$V_{x^\dagger, e_i}(x(t_0), 0) = \frac{1}{2}\left((x - x^\dagger)^2 + (0 - e_i)^2\right), \tag{51}$$

and since for any $x \in X$ we have that $\|x\| \leq R$, we conclude

$$V_{x^\dagger, e_i}(x(t_0), 0) \leq \frac{1}{2}\left((2R)^2 + 1^2\right) \tag{52}$$

Therefore, for all $\tau \in (t_0, T^*]$

$$\lambda_i(\tau) \leq \frac{1}{2}\left(4R^2 + 1^2\right). \tag{53}$$

Two cases are possible now, either $T^* = T$ in which case the bound for $\lambda_i(t)$ holds for any $t \in [0, T]$ or $T^* < T$. If the second case holds, then we can repeat the same argument defining a time t_1 such that:

$$t_1 = \arg \inf_{t \in [T^*, T]} \{\lambda_i(t) > 0\}. \tag{54}$$

And using that for times larger than t_1 the multipliers $\lambda(t)_i$ are once again bounded by the equation in (53). ■

The bound in Corollary 1 ensures that action derivatives $\dot{x}(t)$ remain bounded if the subgradients are. This means that action derivatives increase, at most, linearly with ε and is not compounded by an arbitrary increase in the values of the multipliers.

B. Strongly optimal feasible trajectories

This section presents bounds on the growth of the fit and the regret of the trajectories $x(t)$ generated by the saddle point controller defined by (30) and (31). These bounds ensure that the trajectory is feasible and strongly optimal in the sense of Definition 2. To derive these bounds we need to assume that the objective functions $f_0(t, x)$ are lower bounded as we formally explain next.

Assumption 3. The objective functions $f_0(t, x)$ are lower bounded on the action space X. In particular, there is a finite constant K independent of the time horizon T such that for all t in the interval $[0, T]$.

$$K \geq f_0(t, x) - \min_{x \in X} f_0(t, x). \tag{55}$$

The existence of the bound in (55) is a mild requirement. Since the functions $f_0(t, x)$ are convex, a lower bound exists for each function $f_0(t, x)$ if the action space X is bounded, as is the case in most applications of practical interest. The only restriction imposed in this case is that $\min_{x \in X} f_0(t, x)$ does not become progressively smaller with time so that a uniform bound K holds for all times t. The bound can still hold if X is not compact as long as the span of the functions $f_0(t, x)$ is not unbounded below.

A consequence of Assumption 3 is that the regret cannot *decrease* faster than a linear rate as we formally state in the following lemma.

Lemma 2. *Let $X \subset \mathbb{R}^n$ be a convex set of actions. If Assumption 3 hold, then the regret defined in* (3) *is lower bounded by $-KT$ where K is the constant defined in* (55) *i.e*

$$\mathcal{R}_T \geq -KT. \tag{56}$$

Proof. See Appendix B. ■

Observe that regret is a quantity that we want to make small and, therefore, having negative regret is a desirable outcome. The result in Lemma 2 puts a floor on how much we can succeed in making regret negative.

Using the bound in (56) and the definition of the energy function in (33) we can write down regret and fit bounds for an action trajectory $x(t)$ that follows the saddle point dynamics defined by (30) and (31). We state these bounds in the following theorem.

Theorem 3. *Let $f : \mathbb{R} \times X \to \mathbb{R}^m$ and $f_0 : \mathbb{R} \times X \to \mathbb{R}$, where f and f_0 and 3 where $X \subset \mathbb{R}^n$ is a convex set. If the environment is viable, then the controller defined by* (30) *and* (31) *produces trajectories $x(t)$ that are feasible and strongly optimal for all time horizons $T > 0$. In particular, the fit is bounded by*

$$\mathcal{F}_{T,i} \leq \left(\frac{1}{\varepsilon} V_{x^*, \left[\int_0^T f(t,x)\, dt\right]^+}(x(0), \lambda(0)) + KT\right)^{1/2}, \tag{57}$$

and the regret is bounded by

$$\mathcal{R}_T \leq \frac{1}{\varepsilon} V_{x*,0}\left(x(0), \lambda(0)\right), \tag{58}$$

where $V_{\bar{x},\bar{\lambda}}(x,\lambda)$ *is the energy function defined in* (33)*,* x^* *is the solution to the problem in* (2) *and* K *is the constant defined in* (55)*.*

Proof. See Appendix C ∎

Theorem 3 assures that if the environment is viable for an agent selecting actions from a bounded set X, the saddle point controller defined in (30)-(31) gives origin to trajectories that are feasible and strongly optimal. The fit bounds in theorems 2 and 3 prove a trade off between optimality and feasibility. If optimality of the trajectory is not of interest it is possible to get strongly feasible trajectories with fit that is bounded by a constant independent of the time horizon T (cf. Theorem 2). When an optimality criterion is added to the problem, its satisfaction may come at the cost of a fit that may increase as $\sqrt{T}$. An important consequence of this difference is that even if we could set the gain ε to be arbitrarily large, the fit bound cannot be made arbitrarily small. The fit would still grow as $\sqrt{KT}$. The result in Theorem 3 also necessitates Assumption 3 which is not needed for Theorem 2.

As in the cases of theorems 1 and 2 it is possible to have the environment and objective function selected strategically. Further note that, again, similar to theorems 1 and 2, the initial value of the energy function used to bound both regret and fit is related with the square of the distance between the initial action and the optimal offline solution of problem (2). Therefore, the closer we start from this action the smaller the bound of regret and fit will be.

V. Numerical experiments

We evaluate performance of the saddle point algorithm defined by (30)-(31) in the solution of the shepherd problem introduced in Section II-B. We determine sheep paths using a perturbed polynomial characterization akin to the one in (8). Specifically, letting $p_j(t)$ be elements of a polynomial basis, the path $y_i(t) = [y_{i1}(t), y_{i2}(t)]^T$ followed by the ith sheep is given by the expression

$$y_{ik}(t) = \sum_{j=0}^{n_i-1} y_{ikj} p_j(t) + w_{ik}(t), \tag{59}$$

where $k = 1, 2$ denotes different path components, n_i the total number of polynomials that parameterize the path followed by sheep i, and y_{ikj} represent the corresponding n_i coefficients. The noise terms $w_{ik}(t)$ are Gaussian white with zero mean, standard deviation σ, and chosen independently across components and sheep. Their purpose is to obtain more erratic paths.

To determine y_{ikj} we make $w_{ik}(t) = 0$ in (59) and require all sheep to start at position $y_i(0) = [0,0]^T$ and finish at position $y_i(T) = [1,1]^T$. A total of L random points $\{\tilde{y}_l\}_{l=1}^L$ are then drawn independently and uniformly at random in the unit box $[0,1]^2$. Sheep $i = 1$ is required to pass trough points $\tilde{y}_l$ at times $lT/(L+1)$, i.e., $y_1(lT/(L+1)) = \tilde{y}_l$. For each of the other sheep $i \neq 1$ we draw L random offsets $\{\Delta\tilde{y}_{il}\}_{l=1}^L$ uniformly at random from the box $[-\Delta, \Delta]^2$ and require the ith sheep path to satisfy $y_i(lT/(L+1)) = \tilde{y}_l + \Delta\tilde{y}_{il}$. Paths $y_i(t)$ are then chosen as those that minimize the path integral of the acceleration squared subject to the constraints of each individual path, i.e.,

$$\begin{aligned} y_i^* = \operatorname{argmin} & \int_0^T \|\ddot{y}_i(t)\|^2 dt, \\ \text{s.t.} \quad & y_i(0) = [0,0]^T, \quad y_i(T) = [1,1]^T, \\ & y_i(lT/(L+1)) = \tilde{y}_l + \Delta\tilde{y}_{il}, \end{aligned} \tag{60}$$

where, by construction $\Delta\tilde{y}_{il} = 0$ for $i = 1$. The minimum acceleration paths in (60) can be computed as solutions of a quadratic program [30]. Let $y_i^*(t)$ be the trajectory given by (59) when we set $y_{ikj} = y_{ikj}^*$. We obtain the paths $y_{ik}(t)$ by adding $w_{ik}(t)$ to $y_i^*(t)$.

In subsequent numerical experiments we consider $m = 5$ sheep, a time horizon $T = 1$, and set the proximity constraint in (9) to $r_i = 0.3$. We use the standard polynomial basis $p_j(t) = t^j$ in both, (8) and (59). The number of basis elements in both cases is set to $n = n_i = 30$. To generate sheep paths we consider a total of $L = 3$ randomly chosen intermediate points, set the variation parameter to $\Delta = 0.1$, and the perturbation standard deviation to $\sigma = 0.1$. These problem parameters are such that the environment is most likely viable in the sense of Definition 1. We check that this is true by solving the offline feasibility problem. If the environment is not viable a new one is drawn before proceeding to the implementation of (30)-(31).

We emphasize that even if the complete trajectory of the sheep is known to us, the information is not used by the controller. The controller is only fed information of the position of the sheep at the current time, which it uses to evaluate the environment functions $f_i(t,x)$ in (9), their gradients $f_{ix}(t,x)$ and the gradient of $f_0(t,x)$. In the first problem considered $f_0(t,x)$ is identically zero, in the second takes the form of (10) and in the last problem the form of (11).

A. Strongly feasible trajectories

We first consider a problem without optimality criterion in which case (30)-(31) simplifies to (32)-(31) and the strong feasibility result in Theorem 2 applies.

The system's behavior is illustrated in Figure 2 when the gain is set to $\varepsilon = 50$. A qualitative examination of the sheep and shepherd paths shows that the shepherd succeeds in following the herd. A more quantitative evaluation is presented in Figure 3 where we plot the instantaneous constraint violation $f_i(t, x(t))$ with respect to each sheep for the trajectories $x(t)$ from (32)-(31). Observe the oscillatory behavior that has the constraint violations $f_i(t, x(t))$ hovering at around $f_i(t, x(t)) = 0$. When the constraints are violated, i.e., when $f_i(t, x(t)) > 0$, the saddle point controller drives the shepherd towards a position that makes him stay within r_i of all sheep. When a constraint is satisfied we have $f_i(t, x(t)) < 0$. This drives the multiplier $\lambda_i(t)$ towards 0 and removes the force that pushes the shepherd towards the sheep (c.f. Figure 3). The absence of this force makes the constraint violation grow and eventually surpass the maximum tolerance $f_i(t, x(t)) = 0$. At

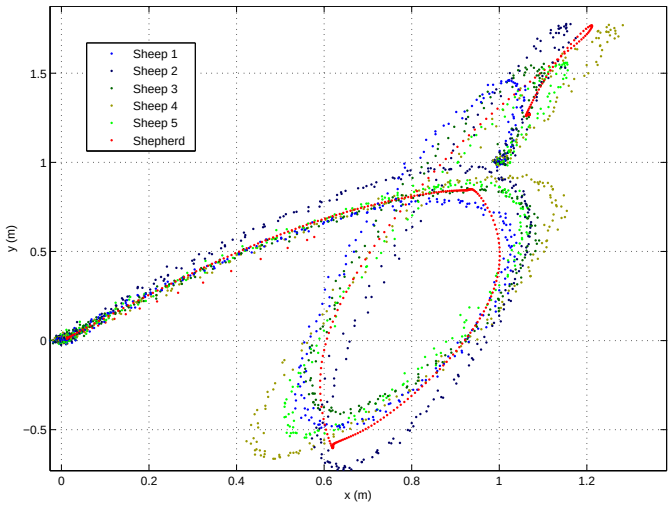
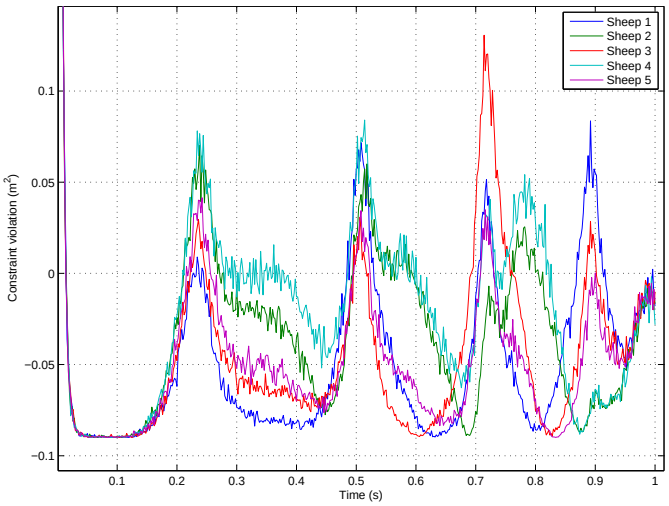
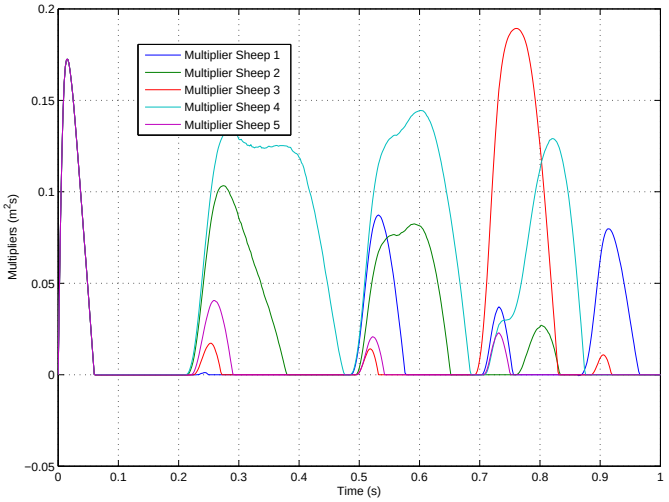


Fig. 2: Path of the sheep and the shepherd for the feasibility-only problem (Section V-A) when the gain of the saddle point controller is set to be $\varepsilon = 50$. The shepherd succeed in following the herd since its path – in red – is close to the path of all sheep.



(a) Instantaneous constraint value.



(b) Temporal evolution of the multipliers.

Fig. 3: Relationship between the instantaneous value of the constraints and their corresponding Lagrange multipliers for the feasibility-only problem (Section V-A). At the times in which the value of a constraint is positive, its corresponding multiplier increases. When the value of the multipliers is large enough a decrease of the value of the constraint function is observed. Once the constraint function is negative the corresponding multiplier decreases until it reaches zero.

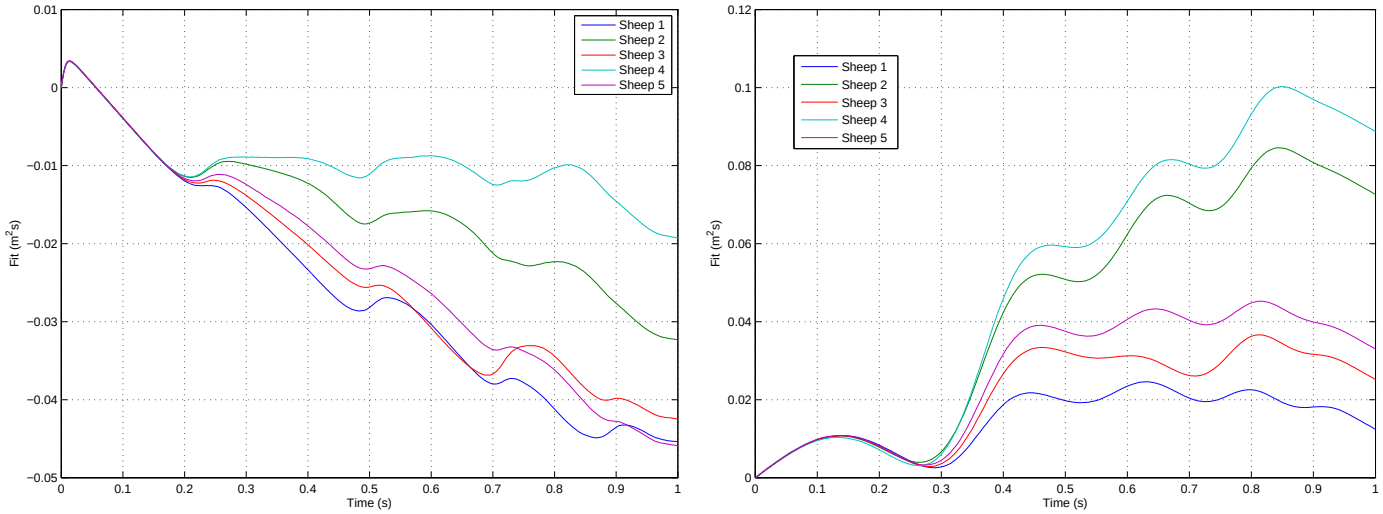
this point the multipliers start to grow and, as a consequence, to push the shepherd back towards proximity with the sheep.

The behavior observed in Figure 3 does not contradict the result in Theorem 2 which gives us a guarantee on fit, not on instantaneous constraint violations. The components of the fit are shown in Figure 4a where we see that they are indeed bounded. Thus, the trajectory is feasible in the sense of Definition 2, even if the constraints are being violated at specific time instances. Further note that the fit is not only bounded but actually becomes negative. This is a consequence of the relatively large gain $\varepsilon = 50$ which helps the shepherd to respond quickly to the sheep movements. The fit for a second experiment in which the gain is reduced to $\varepsilon = 5$ is shown in Figure 4b. In this case the fit stabilizes at a positive value. This behavior is expected because reducing ε decreases the speed with which the shepherd can adapt to changes in the sheep paths. More to the point, the fit bound in Theorem 2 is inversely proportional to the gain ε. The paths and instantaneous constraints violations for $\varepsilon = 5$ are not shown but they are qualitatively similar to the ones shown for $\varepsilon = 50$ in figures 2 and 3.

B. Preferred sheep problem

Besides satisfying the constraints defined in (9), the shepherd is interested in following the first (black) sheep as close as possible. This translates into the optimality criterion defined in (10). Since we construct sheep trajectories that are viable the hypotheses of Theorem 3 hold. Thus, if the shepherd follows the dynamics described by (30) and (31), the resulting action trajectory is feasible and strongly optimal.

Given that the trajectory is guaranteed to be feasible, we expect to have the fit bounded by a sublinear function of T. This does happen, as can be seen in the fit trajectories illustrated in Figure 5 where a gain $\varepsilon = 50$ is used. In fact, the fit does not grow and is actually bounded by a constant for all time horizons T. The trajectory is therefore not only feasible but strongly feasible. This does not contradict Theorem 3 because strong feasibility implies feasibility. The reason why it's reasonable to see bounded fit here is that the objective function pushing the shepherd closer to the sheep is, in a sense, redundant with the constraints that push the shepherd to stay closer to all sheep. This redundancy can be also observed in the fact that the fit in this problem (c.f. Figure 5) is smaller than the fit in the problem of Section V-A (c.f. Figure 4a). To explain why this may happen, focus on the value of the multipliers in Figure 3b between, e.g., times $0.7\text{s} < t < 2.1\text{s}$. During this time all the multipliers are equal to zero because the shepherd is satisfying all constraints and,



(a) Experiment with gain $\varepsilon = 50$.

(b) Experiment with gain $\varepsilon = 5$.

Fig. 4: Fit $\mathcal{F}_T$ for two different controller gains in the feasibility-only problem (Section V-A). Fit is bounded in both cases as predicted by Theorem 2. As is also predicted by Theorem 2, the larger the value of the gain ε the smaller the fit of the shepherd's trajectory.

as a consequence, the Lagrangian subgradient with respect to the action is identically zero in the time interval. In turn, this implies that the action is constant and no effort is made to reduce the value of the constraints. If the optimality criterion were present, the shepherd would still be pushed towards the black sheep and fit would be further reduced.

The regret trajectory for this experiment with $\varepsilon = 50$ is shown in Figure 6. Since the trajectory is strongly optimal as per Theorem 3, we expect regret to be bounded. This is the case in Figure 6 where regret is actually negative for all times $t \in [0, T]$. Negative regret implies that the trajectory of the shepherd is incurring a total cost that is smaller than the one associated with the optimal solution. Notice that while the optimal fixed action minimizes the total cost as defined in (2) it does not minimize the objective at all times. Thus, by selecting different actions the shepherd can suffer smaller instantaneous losses than the ones associated with the optimal action. If this is the case, regret – which is the integral of the difference between these two losses – can be negative.

The path of the shepherd is not shown for this experiment as it is qualitatively analogous to the one in Figure 2 for the feasibility-only problem considered in Section V-A.

C. Minimum acceleration problem

We consider, as in sections V-A and V-B, an environment defined by the distances between the shepherd and the sheep given by (9), but add the minimum acceleration objective defined in (11). Since the construction of the target trajectories gives a viable environment we satisfy, again, the hypotheses of Theorem 3. Hence, for a shepherd following the dynamics given by (30) and (31), the action trajectory is feasible and strongly optimal. For the simulation in this section the gain of the controller is set to $\varepsilon = 50$.

A feasible trajectory implies that the fit must be bounded by a function that grows sub linearly with the time horizon T. Notice that this is the case in Figure 8. Periods of growth of the fit are observed, yet the presence of inflection points is an evidence of the growth being controlled. The fit in this problem is larger than the one in problem V-B (c.f figures 5 and 8). This result is predictable since the constraints and the objective function push the action in different directions. For instance, suppose that all constraints are satisfied and that the Lagrange multipliers are zero. Then, the subgradient of the Lagrangian is equal to the subgradient of the objective function. Hence the action will be modified trying to minimize the acceleration without taking the constraints (distance with the sheep) into account. Hence, pushing the action to the boundary of the feasible set. In this problem, this translates into the fact that the shepherd does not follow the sheep as closely as in the problems in sections V-A and V-B (c.f Figure 7).

Since the trajectory is strongly optimal, we should observe a regret bounded by a constant. This is the case in Figure 9. Notice that regret increases since the initial action differs from the optimal. However, as in the case of the fit, the inflection point at the end of the simulation is the evidence that the regret is being controlled. Compared with the regret of the black sheep problem (c.f Figure 6), the regret in this problem is larger. This is again explained by the fact that in this problem objective and constraints can push the action in different directions while in the problem in Section V-B the objective and the constraints point in the same general direction.

VI. Conclusion

We considered a continuous time environment in which an agent must select actions to satisfy a set of constraints. These constraints are time varying and the agent does not have information regarding their future evolution. We defined a viable environment as one in which there is a fixed action that verifies all the constraints at all times. We defined the concept

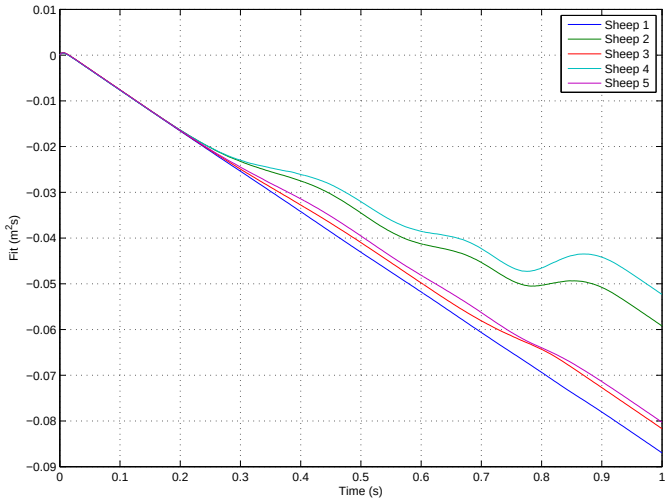


Fig. 5: Fit $\mathcal{F}_T$ for the preferred sheep problem (Section V-B) when the gain of the saddle point controller is set to be $\varepsilon = 50$. As predicted by Theorem 3 the trajectory is feasible since the fit is bounded, and, in fact, appears to be strongly feasible. Since the subgradient of the objective function is the same as the subgradient of the first constrain the fit is smaller than in the pure feasibility problem (c.f Figure 4).

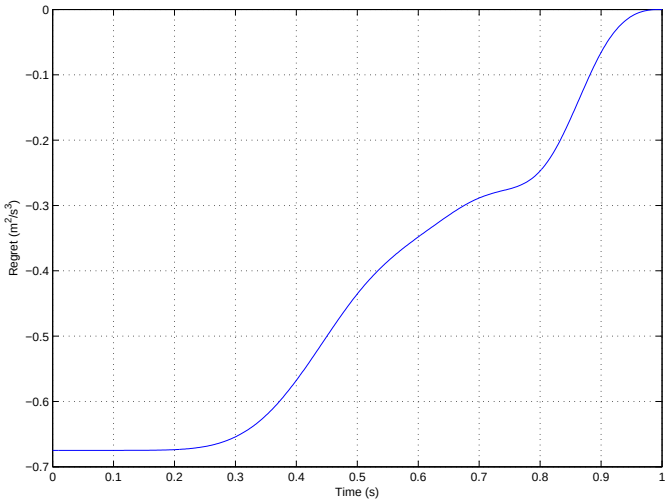


Fig. 6: Regret $\mathcal{R}_T$ for the preferred sheep problem (Section V-B) when the gain of the saddle point controller is set to be $\varepsilon = 50$. The trajectory is strongly feasible, as predicted by Theorem 3.

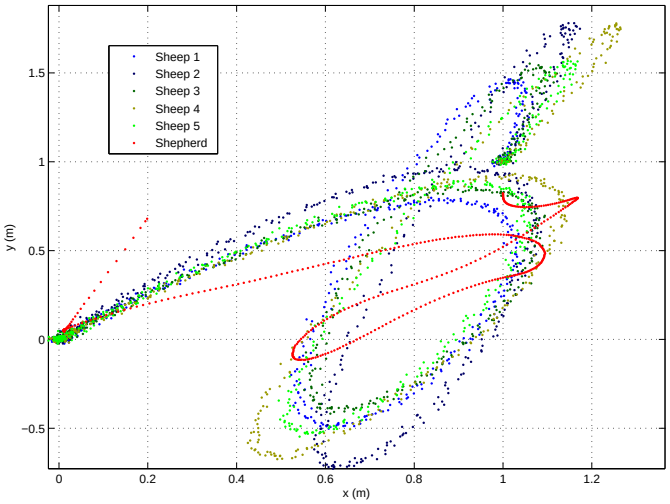


Fig. 7: Path of the sheep and the shepherd for the minimum acceleration problem (Section V-C) when the gain of the saddle point controller is set to be $\varepsilon = 50$. Observe that the shepherd path – in red – is not as close to the path of the sheep as in Figure 2. This is reasonable because the objective function and the constraints push the shepherd in different directions.

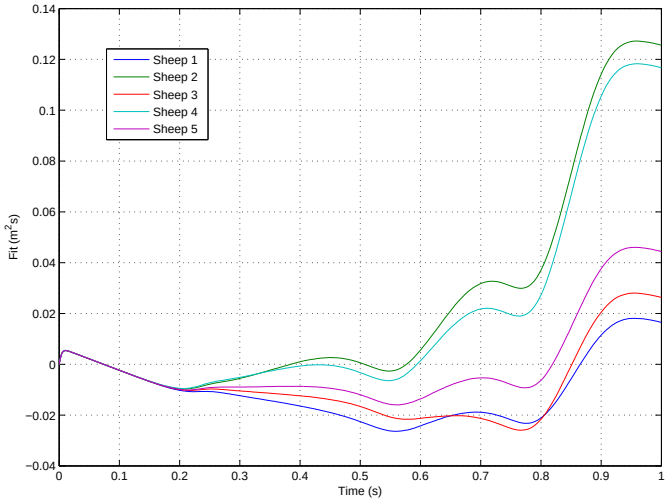


Fig. 8: Fit $\mathcal{F}_T$ for the minimum acceleration problem (Section V-C) when the gain of the saddle point controller is set to $\varepsilon = 50$. Since the fit is bounded, the trajectory is feasible in accordance with Theorem 3. Since the gradient of the objective function and the gradient of the feasibility constraints tend to point in different directions, the fit is larger than in the preferred sheep problem (c.f Figure 5).

of fit as the total constraint violation and the notions of feasible and stronlgy feasible trajectories. In feasible trajectories the fit is bounded by a constant that is independent of the time horizon, and in strongly feasible trajectories the fit is bounded by a sub linear function of the time horizon. An objective function was considered as well to select a strategy that meets an optimality criterion from the set of strategies that satisfy the constraints. We defined regret in continuous time as the difference between the integral of the cost the agent incurs and the minimum total cost that a clairvoyant agents would suffer. We then defined strongly optimal trajectories as those for which the regret is bounded by a constant that is independent of the time horizon.

We proposed an online version of the saddle point controller of Arrow-Hurwicz to generate trajectories with small fit and regret. We showed that for any viable environment the trajectories that follow the dynamics of this controller are: (i) Strongly feasible if no optimality criterion is considered. (ii) Feasible and strongly optimal when an optimality criterion is considered. Numerical experiments on a shepherd that tries to follow a herd of sheep support these theoretical results.

APPENDIX

A. *Proof of Lemma 1*

In order to develop this proof we need to define the concept of tangent cone and to state Lemma 3 relating the projection

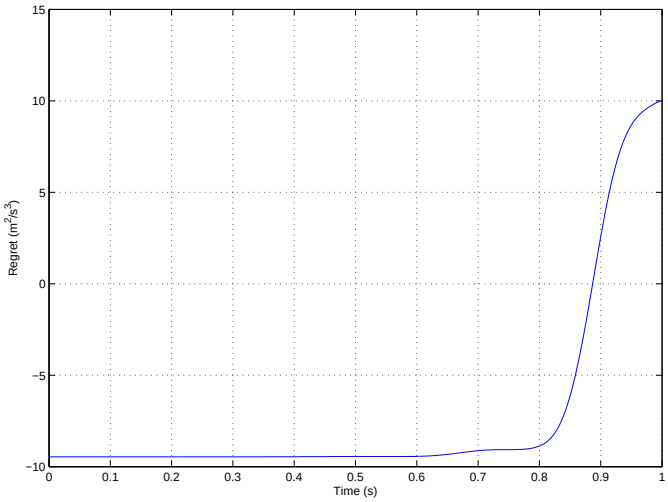


Fig. 9: Regret $\mathcal{R}_T$ for the minimum acceleration problem (Section V-C) when the gain of the saddle point controller is set to be $\varepsilon = 50$. The trajectory is strongly optimal as predicted by Theorem 3. Since the gradient of the objective function and the gradient of the feasibility constraints tend to point in different directions, regret is larger than the regret of the preferred sheep problem (c.f Figure 5).

of a vector over a set with the projection over the tangent cone.

Definition 5 (Tangent cone)**.** *Let $X \subset \mathbb{R}^n$ be a closed convex set. We define the tangent cone to X at x_0 as*

$$T_X(x_0) = \overline{\bigcup_{\theta>0, x\in X} \theta(x - x_0)}. \tag{61}$$

The above union is over all the points of the set X and over all the positive reals θ. Notice that the $\bigcup_{\theta>0} \theta(x-x_0)$ is the ray from x_0 and intersecting the point x. Thus, the tangent cone is then the closure of the cone formed by all rays emanating from x_0 and intersecting at least one point $x \in X$ different from x_0.

Lemma 3. *For arbitrary δ and v the projection over the set X can be written as*

$$P_X(x_0 + \delta v) = x_0 + \delta P_{T_X(x_0)}(v) + \mathcal{O}(\delta), \tag{62}$$

where $\mathcal{O}(\delta)$ is a function such that $\lim_{\delta\to 0} \mathcal{O}(\delta)/\delta = 0$.

Proof. See [28] Lemma 4.6 page 300. ■

Corollary 2. *Let $X \in \mathbb{R}^n$ be a closed convex set, let $x_0 \in X$ and let $v \in \mathbb{R}^n$. Then the projection of v over the set X at x_0 defined in* (14) *is*

$$\Pi_X(x_0, v) = P_{T_X(x_0)}(v). \tag{63}$$

Proof. The proof is trivial from lemma 3. ■

Proof of Lemma 1. Consider the case in which $x_0 \in int(X)$. Then, for any v there exits a small enough $\delta > 0$ such that $x_0 + \delta v \in X$. Hence $P_X(x_0 + \delta v) = x_0 + \delta v$, and then we have that

$$P_X(x_0 + \delta v) - x_0 = v\delta. \tag{64}$$

Thus $\Pi_X(x, v) = v$ and then we have trivially that

$$(x_0 - x)^T \Pi_X(x_0, v) = (x_0 - x)^T v. \tag{65}$$

Let's now consider the case in which x_0 is in the border of X, here two case are posible: either $x_0 + \delta v \in T_X(x_0)$ for small enough $\delta > 0$ or $x_0 + \delta v \notin T_X(x_0)$ for all $\delta > 0$. Because of this distinction is that the result of Corollary 2 is important. In the first case we trivially have that

$$\Pi_X(x_0, v) = P_{T_X(x_0)}(v) = v. \tag{66}$$

And therefore (65) holds in this case as well. Let us now consider the case in which $x_0 \in \partial X$ and $x_0 + \delta v \notin T_X(x_0)$. Because X is a convex set there exists a vector $a \in \mathbb{R}^n$ with $\|a\| = 1$ defining a supporting hyperplane at x_0

$$\mathcal{H} = \{x \in \mathbb{R}^n : a^T(x - x_0) = 0\}. \tag{67}$$

Since it $\mathcal{H}$ is a supporting hyperplane, for all $x \in X$ we have that

$$a^T(x - x_0) \leq 0. \tag{68}$$

If the set is smooth at x_0 then the border of the tangent cone at the point x_0 is contained in the hyperplane $\mathcal{H}$, therefore $\Pi_X(x_0, v) \subset \mathcal{H}$. Thus, $a^T \Pi_X(x_0, v) = 0$. In this case we have as well that $a^T v \geq 0$, otherwise there must exists a $\delta > 0$ such that $x_0 + \delta v \in T_X(x_0)$. On the other hand if there is a corner at x_0 there are infinite supporting hyperplanes. One of them verifies that $a^T v \geq 0$ and contains the border of the tangent cone, thus $a^T \Pi_X(x_0, v) = 0$. Since $\Pi_X(x_0, v)$ is the projection of v over the tangent cone, we have that: $\Pi_X(x_0, v) = P_{T_X(x_0)}(v) = (a_\perp^T v) a_\perp$, where $a_\perp \in \mathbb{R}^n$ and verifies that $a^T a^\perp = 0$ and $\|a^\perp\| = 1$. Projecting the vectors $x_0 - x$ and v over a and $a^\perp$, we have

$$(x_0 - x)^T v = (x_0 - x)^T a v^T a + (x_0 - x)^T a_\perp v^T a_\perp. \tag{69}$$

Because of the previous discusion the above equation reduces to

$$(x_0 - x)^T v = (x_0 - x)^T a v^T a + (x_0 - x)^T \Pi_X(x_0, v). \tag{70}$$

Using the fact that a is the vector director of a supporting hyperplane (68) and using the fact that $v^T a \geq 0$ the following inequality holds

$$(x_0 - x)^T v \geq (x_0 - x)^T \Pi_X(x_0, v). \tag{71}$$

Hence we have proved the lemma for all posible cases. ■

B. Proof of Lemma 2

If Assumption 3 holds, for any $x \in X$ we have that

$$K \geq f_0(t, x) - \min_{x\in X} f_0(t, x). \tag{72}$$

In particular, set $x = x^*$, where x^* is the solution to the offline convex optimization problem defined in (2). Rearrange the terms in the above equation to get

$$\min_{x\in X} f_0(t, x) - f_0(t, x^*) \geq -K. \tag{73}$$

Since for any $x \in X$ we have that $f_0(t, x) \geq \min_{x\in X} f_0(t, x)$, in particular

$$f_0(t, x(t)) - f_0(t, x^*) \geq -K, \tag{74}$$

where $x(t)$ is the action at time t when the agent follows the dynamics defined by (30) and (31). Integrate both sides of the above equation in the interval $[0, T]$

$$\int_0^T f_0(t, x(t))dt - \int_{t=0}^T f_0(t, x^*)dt \geq -KT \tag{75}$$

Since the left hand side of the above equation is the definition of regret up to time T defined in (3) we proved the lower bound stated on the Lemma.

C. Proof of Theorem 3

Consider action trajectories $x(t)$ and multiplier trajectories $\lambda(t)$ and the corresponding energy function $V_{\bar{x},\bar{\lambda}}(x(t), \lambda(t))$ in (33), for arbitrary given action $\bar{x} \in \mathbb{R}^n$ and multiplier $\bar{\lambda} \in \Lambda$. The derivative $\dot{V}_{\bar{x},\bar{\lambda}}(x(t), \lambda(t))$ of the energy with respect to time is then given by

$$\dot{V}_{\bar{x},\bar{\lambda}}(x(t), \lambda(t)) = (x(t) - \bar{x})^T \dot{x}(t) + (\lambda(t) - \bar{\lambda})^T \dot{\lambda}(t). \tag{76}$$

If the trajectories $x(t)$ and $\lambda(t)$ follow from the saddle point dynamical system defined by (30) and (31) respectively we can substitute the action and multiplier derivatives by their corresponding values and reduce (76) to

$$\begin{aligned}\dot{V}_{\bar{x},\bar{\lambda}}(x(t), \lambda(t)) = (x(t) - \bar{x})^T \Pi_X(x, -\varepsilon(f_{0,x}(t, x(t)) \\ + f_x(t, x(t))\lambda(t)) + (\lambda(t) - \bar{\lambda})^T \Pi_\Lambda(x, \varepsilon f(t, x(t))).\end{aligned} \tag{77}$$

Then, use Lemma 1 for both X and Λ to write

$$\begin{aligned}\dot{V}_{\bar{x},\bar{\lambda}}(x(t), \lambda(t)) \leq \varepsilon[-(x(t) - \bar{x})^T (f_{0,x}(t, x(t)) \\ + f_x(t, x(t))\lambda(t)) + (\lambda(t) - \bar{\lambda})^T f(t, x(t))].\end{aligned} \tag{78}$$

Notice that $\mathcal{L}(t, x(t), \lambda(t)) = f_0(t, x(t)) + \lambda(t)^T f(t, x(t))$ is a convex function with respect to the actions since it is a sum of convex functions with respect to x. Then, using the definition of subgradient (c.f. Definition 3) we can upper bound the inner product

$$\begin{aligned}-(x(t) - \bar{x})^T (f_{0,x}(t, x(t)) + f_x(t, x(t))\lambda(t)) \\ = -(x(t) - \bar{x})^T \mathcal{L}_x(t, x(t), \lambda(t))\end{aligned} \tag{79}$$

by the difference $\mathcal{L}(t, \bar{x}, \lambda(t)) - \mathcal{L}(t, x(t), \lambda(t))$. Then, we can upper bound the right hand side of the equation 78 and obtain

$$\begin{aligned}\dot{V}_{\bar{x},\bar{\lambda}}(x(t), \lambda(t)) \leq \varepsilon[f_0(t, \bar{x}) + \lambda^T(t) f(t, \bar{x}) - f_0(t, x(t)) \\ - \lambda^T(t) f(t, x(t)) + (\lambda(t) - \bar{\lambda})^T f(t, x(t))].\end{aligned} \tag{80}$$

Notice that on the right hand side of the above inequality the fourth and the fifth term are opposite. Thus we can reduce the above equation to

$$\begin{aligned}\dot{V}_{\bar{x},\bar{\lambda}}(x(t), \lambda(t)) \leq \varepsilon[f_0(t, \bar{x}) + \lambda^T(t) f(t, \bar{x}) \\ - f_0(t, x(t)) - \bar{\lambda}^T f(t, x(t))].\end{aligned} \tag{81}$$

Rewriting the above equation and then integrating both sides with respect to the time from time $t = 0$ to $t = T$, we obtain

$$\begin{aligned}\int_0^T f_0(t, x(t)) - f_0(t, \bar{x}) + \bar{\lambda}^T f(t, x(t)) - \lambda^T(t) f(t, \bar{x}) dt \\ \leq -\frac{1}{\varepsilon} \int_0^T \dot{V}_{\bar{x},\bar{\lambda}}(x(t), \lambda(t)) dt.\end{aligned} \tag{82}$$

Using the result (42) the above equation reduces to

$$\begin{aligned}\int_0^T f_0(t, x(t)) - f_0(t, \bar{x}) + \bar{\lambda}^T f(t, x(t)) - \lambda^T(t) f(t, \bar{x}) dt \\ \leq \frac{1}{\varepsilon} V_{\bar{x},\bar{\lambda}}(x(0), \lambda(0)).\end{aligned} \tag{83}$$

Since (83) holds for any $\bar{x} \in X$ and any $\bar{\lambda} \in \Lambda$, it holds for the particular choice $\bar{x} = x^*$, $\bar{\lambda} = 0$. Since $\lambda^T(t) f(t, x^*)\, dt \leq 0$ $\forall t \in [0, T]$ we can lower bound the left hand side of (83) to obtain:

$$\int_0^T f_0(t, x(t)) - f_0(t, x^*) dt \leq \frac{1}{\varepsilon} V_{x^*,0}(x(0), \lambda(0)). \tag{84}$$

Notice that the left hand side of the above equation is the definition of regret given in 3. Thus, we have shown that the upper bound for the regret is the one stated in (58). And since the right hand side of the above equation is a constant for all $T > 0$ we proved that the trajectory generated by the saddle point controller is strongly optimal. It remains to prove that the trajectory generated is feasible. In order to do so, choose $\bar{x} = x^*$, and use the result of Lemma 2, to transform (83) into

$$\begin{aligned}\int_0^T \bar{\lambda}^T f(t, x(t)) - \lambda^T(t) f(t, x^*)\, dt \\ \leq \frac{1}{\varepsilon} V_{x^*,\bar{\lambda}}(x(0), \lambda(0)) + KT.\end{aligned} \tag{85}$$

Since $\lambda^T(t) f(t, x^*)\, dt \leq 0 \;\; \forall t \in [0, T]$ we can again lower bound the left hand side of the above equation by $\bar{\lambda}^T \int_0^T f(t, x(t))$ and obtain

$$\bar{\lambda}^T \int_0^T f(t, x(t)) dt \leq \left(V_{x^*,\bar{\lambda}}(x(0), \lambda(0))\right)/\varepsilon + KT. \tag{86}$$

Now let's choose $\bar{\lambda} = \left[\int_0^T f(t, x(t))\, dt\right]^+$. The projection on the positive orthant is needed because $\bar{\lambda} \in \mathbb{R}^m_+$. Let $I = \{i = 1..m | \int_0^T f_i(t, x(t))\, dt \geq 0)\}$. Notice that if $i \notin I$, then $\bar{\lambda}_i \int_0^T f_i(t, x(t))\, dt = 0$. On the other hand, if $i \in I$, $\bar{\lambda}_i \int_0^T f_i(t, x(t))\, dt = \left(\int_0^T f_i(t, x(t))\, dt\right)^2 \geq 0$. Therefore, for all $i \in I$ we have:

$$\begin{aligned}\left(\int_0^T f_i(t, x(t))\, dt\right)^2 \\ \leq \frac{1}{\varepsilon} V_{x^*,\left[\int_0^T f(t,x)\, dt\right]^+}(x(0), \lambda(0)) + KT.\end{aligned} \tag{87}$$

Notice that, the left hand side of the above equation is the square of the ith component of the fit. Thus for all $i \in I$ it is clear that:

$$\mathcal{F}_{T,i} \leq \left(\frac{1}{\varepsilon} V_{x^*,\left[\int_0^T f(t,x)\, dt\right]^+}(x(0), \lambda(0)) + KT\right)^{1/2}. \tag{88}$$

Now if $i \notin I$ it means that $\mathcal{F}_{T,i} < 0$ therefore it is also smaller than the established bound in (88). Which proves that the trajectories generated by the saddle point controller defined by (30) and (31) are feasible since they are bounded by a sublinear function of the time horizon for all T.

REFERENCES

[1] K. J. Arrow and L. Hurwicz, *Studies in linear and nonlinear programming*. CA: Stanford University Press, 1958.

[2] M. W. Hirsch, S. Smale, and R. L. Devaney, *Differential equations, dynamical systems, and an introduction to chaos*, vol. 60. Academic press, 2004.

[3] M. Krstić and H.-H. Wang, "Stability of extremum seeking feedback for general nonlinear dynamic systems," *Automatica*, vol. 36, no. 4, pp. 595–601, 2000.

[4] K. B. Ariyur and M. Krstic, *Real-time optimization by extremum-seeking control*. John Wiley & Sons, 2003.

[5] Y. Tan, D. Nešić, and I. Mareels, "On non-local stability properties of extremum seeking control," *Automatica*, vol. 42, no. 6, pp. 889–903, 2006.

[6] W. H. Moase, C. Manzie, and M. J. Brear, "Newton-like extremum-seeking part i: theory," in *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*, pp. 3839–3844, IEEE, 2009.

[7] E. Rimon and D. E. Koditschek, "Exact robot navigation using artificial potential functions," *Robotics and Automation, IEEE Transactions on*, vol. 8, no. 5, pp. 501–518, 1992.

[8] C. W. Warren, "Global path planning using artificial potential fields," in *Robotics and Automation, 1989. Proceedings., 1989 IEEE International Conference on*, pp. 316–321, IEEE, 1989.

[9] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *Int. J. Rob. Res.*, vol. 5, pp. 90–98, Apr. 1986.

[10] S. A. Masoud and A. A. Masoud, "Constrained motion control using vector potential fields," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 30, no. 3, pp. 251–272, 2000.

[11] S. S. Ge and Y. J. Cui, "New potential functions for mobile robot path planning," *IEEE Transactions on robotics and automation*, vol. 16, no. 5, pp. 615–620, 2000.

[12] P. Vadakkepat, K. C. Tan, and W. Ming-Liang, "Evolutionary artificial potential fields and their application in real time robot path planning," in *Evolutionary Computation, 2000. Proceedings of the 2000 Congress on*, vol. 1, pp. 256–263, IEEE, 2000.

[13] S.-i. Azuma, M. S. Sakar, and G. J. Pappas, "Nonholonomic source seeking in switching random fields," in *Decision and Control (CDC), 2010 49th IEEE Conference on*, pp. 6337–6342, IEEE, 2010.

[14] S.-i. Azuma, M. S. Sakar, and G. J. Pappas, "Stochastic source seeking by mobile robots," *Automatic Control, IEEE Transactions on*, vol. 57, no. 9, pp. 2308–2321, 2012.

[15] N. Atanasov, J. Le Ny, N. Michael, and G. J. Pappas, "Stochastic source seeking in complex environments," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pp. 3013–3018, IEEE, 2012.

[16] S.-J. Liu and M. Krstic, "Stochastic source seeking for nonholonomic unicycle," *Automatica*, vol. 46, no. 9, pp. 1443 – 1453, 2010.

[17] A. Nedić and A. Ozdaglar, "Subgradient methods for saddle-point problems," *Journal of optimization theory and applications*, vol. 142, no. 1, pp. 205–228, 2009.

[18] H. Uzawa, "Iterative methods for concave programming," *Studies in linear and nonlinear programming*, vol. 6, 1958.

[19] D. Maistroskii, "Gradient methods for finding saddle points," *Matekon*, vol. 14, no. 1, pp. 3–22, 1977.

[20] S. H. Low and D. E. Lapsley, "Optimization flow control- i: basic algorithm and convergence," *IEEE/ACM Transactions on Networking (TON)*, vol. 7, no. 6, pp. 861–874, 1999.

[21] M. Chiang, S. H. Low, A. R. Calderbank, and J. C. Doyle, "Layering as optimization decomposition: A mathematical theory of network architectures," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 255–312, 2007.

[22] A. Chambolle and T. Pock, "A first-order primal-dual algorithm for convex problems with applications to imaging," *Journal of Mathematical Imaging and Vision*, vol. 40, no. 1, pp. 120–145, 2011.

[23] S. Shalev-Shwartz, "Online learning and online convex optimization," *Foundations and Trends in Machine Learning*, vol. 4, no. 2, pp. 107–194, 2011.

[24] V. Vapnik, *The nature of statistical learning theory*. Springer, 2000.

[25] M. Zinkevich, "Online convex programming and generalized infinitesimal gradient ascent," in *ICML*, pp. 928–936, 2003.

[26] E. Hazan, A. Agarwal, and S. Kale, "Logarithmic regret algorithms for online convex optimization," *Machine Learning*, vol. 69, no. 2-3, pp. 169–192, 2007.

[27] M.-G. Cojocaru and L. Jonker, "Existence of solutions to projected differential equations in hilbert spaces," *Proceedings of the American Mathematical Society*, vol. 132, no. 1, pp. 183–193, 2004.

[28] D. Zhang and A. Nagurney, "On the stability of projected dynamical systems," *J. Optim. Theory Appl.*, vol. 85, pp. 97–124, Apr. 1995.

[29] D. P. Bertsekas, *Nonlinear Programming*. Belmont, MA: Athena Scientific, 1999.

[30] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, May 2011.