Control-Oriented Learning on the Fly

Melkior Ornik, Steven Carr, Arie Israel, and Ufuk Topcu

***Abstract*—This paper focuses on developing a strategy for control of systems whose dynamics are almost entirely unknown. This situation arises naturally in a scenario where a system undergoes a critical failure. In that case, it is imperative to retain the ability to satisfy basic control objectives in order to avert an imminent catastrophe. To deal with limitations on the knowledge of system dynamics, we develop a theory of myopic control. At any given time, myopic control optimizes the current direction of the system trajectory, given solely the knowledge about system dynamics obtained from data until that time. We propose an algorithm that uses small perturbations in the control effort to learn system dynamics around the current system state while ensuring that the system moves in a nearly optimal direction, and provide bounds for its suboptimality.**

Notice of previous publication. This manuscript is a substantially extended version of [1]. It provides proofs omitted from [1]. Entirely novel material includes Sections IV, VII-B, VIII, Appendix A, and parts of several other sections.

I. Introduction

In an event of a rapid, unexpected change in system dynamics, the ability to retain certain degree of control over a system is crucial. A notable example of such a catastrophic event is that of an aircraft losing its wing after a mid-air collision [2]. In this event, the pilot managed to retain enough control over the aircraft to be able to safely land. His strategy depended on his intuition and prior experience. This paper seeks to develop a theory of control of a system with unknown dynamics in real time. It formalizes an intuitive approach one might use when trying to operate an unknown vehicle: performing small "wiggles" as control actions to see how the system behaves before choosing a longer-term action.

In the described setting of control of an unknown system, the only data available to extract information on the system dynamics can be obtained during the system run. Data-driven learning of the dynamics of unknown systems has been a subject of significant recent research [3], [4], [5], [6], [7], [8], [9], [10]. However, previously introduced methods are data-intensive and require a significant length of time to collect data, significant a priori knowledge about system dynamics, or provide guarantees only on long-term asymptotic system performance. For example, model-free reinforcement learning [6], [7] succeeds at determining an optimal control policy for a system, but relies on a significant number of system runs or one long run during which the system returns to the same state multiple times. On the other hand, adaptive control [8], [9] relies on having the correct model of the system dynamics up to an unknown parameter. A relevant work [11] considers differential inclusions to assess the safety of a trajectory governed by unknown dynamics, but only considers the case where the control signal has been predetermined.

In the event of a failure, it might be desirable to focus on the system's continued survival. In physical systems, the problem of system survival can naturally be framed as a question of reaching a target set while avoiding obstacles and staying within a safety envelope, i.e., a reach-avoid problem. A standard approach to reach-avoid problems [12] is based on constrained optimal control, where the constraints are given by the geometry of the safety envelope and obstacles, and the time to reach a target set is minimized. The solution to this problem depends on prior knowledge of complete system dynamics. For cases in which such knowledge is not available, we propose a new method, *myopic control*, and use it to approximately solve reach-avoid problems. In this framework, the system uses a control input that seemingly works best *at the given time*, without firm knowledge on whether that input will lead to good results in the future. The notion of "seemingly best" is formalized by a *goodness function*, designed based on the control objective and any prior information on system dynamics. To counterbalance the lack of knowledge about the future, a new myopically optimal control input is repeatedly recomputed. In order to determine the next myopically optimal input, the system continually modifies the previous one by a series of wiggles: small, short-time perturbations in control inputs that act as a mechanism to learn the control dynamics of the system around the current state. In order for this mechanism to function, we make a technical assumption that the underlying system is control-affine.

The proposed algorithm results in an approximate solution to the myopic optimal control problem with a degree of suboptimality dependent on the length of the control signal update interval and the size of wiggles. Additionally, if there are known bounds on regularity of system dynamics, the parameters of the algorithm can be set to ensure any desired bound on the degree of suboptimality.

Notation. Set of all continuous functions from set $\mathcal{A} \subseteq \mathbb{R}^n$ to set $\mathcal{B} \subseteq \mathbb{R}^m$ is denoted by $C(\mathcal{A}, \mathcal{B})$. For a set $\mathcal{A}$, $2^{\mathcal{A}}$ denotes the set of all subsets of $\mathcal{A}$. For $I \subseteq \mathbb{R}$, and $t \in \mathbb{R}$ such that there exists $\eta > 0$ with $(t, t+\eta) \in I$, the right one-sided derivative of function $f : I \to \mathbb{R}^n$ at time t is denoted by $df(t+)/dt$, when such a derivative exists. For $x \in \mathbb{R}^n$, $\|x\|$ denotes its 2-norm, $\|x\|_\infty$ its max-norm, x^T its transpose, and x_i, $i \in \{1, \ldots, n\}$, the i-th component of x. Function $\mathbb{1} : \mathbb{R} \to \mathbb{R}$ takes the value of 1 if its argument is positive and 0 otherwise.

M. Ornik is with the University of Illinois at Urbana-Champaign. S. Carr, A. Israel, and U. Topcu are with the University of Texas at Austin. e-mails: mornik@illinois.edu, stevencarr@utexas.edu, arie@math.utexas.edu, utopcu@utexas.edu. The work presented in this paper was partially funded by grants AFOSR FA9550-19-1-0005 and DARPA FA8750-19-C-0092.

II. Problem Definition

We investigate a control system governed by

$$\dot{x}(t) = f\left(x(t)\right) + \sum_{i=1}^{m} g_i\left(x(t)\right) u_i \tag{1}$$

for $t \geq 0$, evolving on a strongly forward invariant set $\mathcal{X} \subseteq \mathbb{R}^n$, where $f, g_i \in C(\mathcal{X}, \mathcal{X})$ are unknown functions, and $u = (u_1, \ldots, u_m)$ is the control input with values in a bounded set $\mathcal{U} \subseteq \mathbb{R}^m$. For simplicity, we take $\mathcal{U} = \{u \in \mathbb{R}^m \mid \|u\|_\infty \leq 1\}$. A solution to (1) with control input u and initial value x_0 is denoted by $\phi_u(\cdot, x_0)$. Assume that at all times we are able to observe the full state x and the entire control u.

If $\mathcal{X}$ is compact, there exists $M_0 \geq 0$ such that $\|f(x)\| \leq M_0$ and $\|g_i(x)\| \leq M_0$ for all $x \in \mathcal{X}$, $i \in \{1, \ldots, m\}$. We assume the existence of such M_0 throughout the paper. Additionally, we assume that $\|f(x) - f(y)\| \leq M_1\|x - y\|$ and $\|g_i(x) - g_i(y)\| \leq M_1\|x - y\|$ for all $x, y \in \mathcal{X}$, $i \in \{1, \ldots, m\}$.

The class of control-affine systems covers a wide array of physical control systems, including standard linear [13] and a variety of nonlinear aircraft models [14], [15]. Control of control-affine systems has been substantially covered by previous literature (see, e.g., [16]). However, motivated by the scenario of an unexpected failure in a physical system, our setting introduces two additional requirements:

- R1 Functions $f, g_1, \ldots, g_m$ in (1) are unknown at the beginning of the system run. We are allowed to use trajectory data during the system run to extract information on them, but when the run starts, we are aware merely that the system is of form (1) and the constants M_0 and M_1.
- R2 There is only one system run, starting from a single predetermined initial state x_0. All control design needs to be performed during that run, without any repetitions.

We emphasize that requirement R1 is beyond the scope of most existing work on uncertain or noisy systems, e.g., [17], [18], which assumes knowledge of the system dynamics up to some bounded disturbance.

The work of this paper is motivated by attempting to solve the *reach-avoid problem* [19] for systems (1) under requirements R1 and R2:

Reach-Avoid Problem. *Let $x_0 \in \mathcal{X}$, $T \geq 0$, and $\mathcal{T}, \mathcal{B} \subseteq \mathcal{X}$. Find, if it exists, a control signal $u^* : [0, T] \to \mathcal{U}$ such that the following holds:*

(i) $\phi_{u^*}(t, x_0) \notin \mathcal{B}$ for all $t \in [0, T]$,
(ii) there exists $0 \leq T'_{u^*} \leq T$ such that $\phi_{u^*}(t, x_0) \in \mathcal{T}$ for all $t \in [T'_{u^*}, T]$, and
(iii) T'_u from (ii) is the minimal such value for all the control laws $u : [0, T] \to \mathcal{U}$ which satisfy (i) and (ii).

The reach-avoid problem can be naturally adapted to an *avoid problem*, where $\mathcal{T} = \mathcal{X}$, and the objective is for the system state to remain outside of set $\mathcal{B}$.

III. Myopic Control

Finding the exact solution to the reach-avoid problem under requirements R1 and R2 might be impossible. By R1 and R2, in order to determine a control signal to be used starting at some time t, we may only use the information obtained from the system trajectory during the interval $[0, t]$. Because the only known relationship between the dynamics at different states is given by the Lipschitz property of functions f and g_i, information obtained from the system trajectory provides increasingly broad bounds on system dynamics at states in the state space the more distant these states are from the system's trajectory. Thus, at time t we can have no certain knowledge of the effect of any control signal at future times.

We propose to replace the reach-avoid problem by a *myopic optimal control problem*, where we desire to design a control signal such that, at every time instance, the trajectory *seems* to behave as well as possible. For instance, in an avoid problem, one possible option is to require that the trajectory at any given instance of time is moving away from the undesirable set $\mathcal{B}$ as fast as possible. This reasoning may not be optimal with respect to solving the reach-avoid problem, as moving away from $\mathcal{B}$ as quickly as possible at some state may bring the system into a state where the dynamics are such that it is forced to enter $\mathcal{B}$. However, since we do not have almost any knowledge of the system dynamics on the entire state space, finding a seemingly optimal direction of movement at any given time represents an intuitive guess about the optimal control signal.

We formalize the above notion of "appearing to behave as well as possible" using a *goodness function* $(\phi, v) \mapsto G(\phi, v)$ with $\phi \in \mathcal{F}$, $v \in \mathbb{R}^n$, and $\mathcal{F} = \cup_{T \geq 0} C([0, T], \mathcal{X})$. In order to emphasize that $\phi \in \mathcal{F}$ has $[0, T]$ as its domain, we will occasionally write such functions as $\phi|_{[0,T]}$. The function G is intended to quantify how well the trajectory appears to be doing at satisfying the reach-avoid specification at a time when its trajectory until time T and velocity at time T are given by ϕ and v, respectively.

Example 1. *Consider a damaged aircraft with the objective of remaining in the air for as long as possible. Let x_1 and x_2 denote the aircraft's horizontal and vertical position, respectively. Then $\mathcal{B} = \{(x_1, x_2) \mid x_2 \leq 0\}$. A possible choice for G is to measure the slope on which the trajectory is moving towards the boundary of $\mathcal{B}$, inferring that the more negative this slope is, the worse the aircraft is doing. Hence, we may take $G(\phi, v) = v_2/v_1$, where we disregard the case of $v_1 \leq 0$. Fig. 1 provides an illustration.*

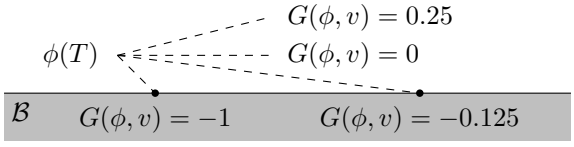

Fig. 1. An illustration of function G from Example 1. Dashed lines represent four example tangents to the trajectory of system (1) at $\phi(T)$. These tangents are evaluated by G depending on their slope towards the undesirable set $\mathcal{B}$.

We now formally introduce the problem of maximizing the system's goodness at every time during the system run.

Myopic Optimal Control Problem. *Let $x_0 \in \mathcal{X}$. Find a control signal $u^* : [0, +\infty) \to \mathcal{U}$ such that, for all $t \geq 0$, then*

$$\begin{aligned} &G\left(\phi_{u^*}(\cdot, x_0)|_{[0,t]}, \frac{d\phi_{u^*}(0+, x)}{dt}\right) \\ &\quad = \max_{u \in \mathcal{U}} G\left(\phi_{u^*}(\cdot, x_0)|_{[0,t]}, \frac{d\phi_u(0+, x)}{dt}\right), \end{aligned} \tag{2}$$

where $x = \phi_{u^*}(t, x_0)$.

With a goodness function G that aims to drive the system to reach a target set and avoid the undesirable set, we use the myopic optimal control problem as an proxy for the reach-avoid problem. We note that in (2) we are slightly abusing notation: for $u \in \mathcal{U}$, $d\phi_u(0+, x)/dt$ denotes the value of the right-hand side of (1) for the particular $x \in \mathcal{X}$ and $u \in \mathcal{U}$. When it causes no confusion, we will omit a formal distinction between a control signal u and a control input $u \in \mathcal{U}$.

In Section V of the paper, we propose an approximate solution to the myopic optimal control problem (Algorithm 3) that satisfies the requirements R1 and R2. The approximation can be made arbitrarily good in the sense that for any $T > 0$ and $\mu > 0$, the algorithm generates a piecewise-constant control signal u^+ that satisfies

$$\left| G\left(\phi_{u^+}(\cdot, x_0)|_{[0,t]}, \frac{d\phi_{u^+}(0+, x)}{dt}\right) - \max_{u \in \mathcal{U}} G\left(\phi_{u^+}(\cdot, x_0)|_{[0,t]}, \frac{d\phi_u(0+, x)}{dt}\right) \right| \leq \mu \tag{3}$$

for all $t \geq T$, where $x = \phi_{u^+}(t, x_0)$. We note that this result does not directly guarantee existence of an *exact* solution to the myopic optimal control problem, nor relate to the solution of the standard optimal control problem of maximizing the total collected goodness (i.e., reward) over a period of time. A brief discussion of sufficient conditions for the existence of a solution to the myopic optimal control problem is contained in Appendix A.

IV. Measuring Goodness

Prior to proposing an algorithm to approximately solve the myopic optimal control problem, let us provide a brief discussion of design of goodness functions to solve the reach-avoid problem.

Let us first revisit the avoid problem. As the sole specification is to not enter the bad set $\mathcal{B}$, one approach would seek the trajectory to be moving away from $\mathcal{B}$ as fast as possible. Without any additional knowledge, a possible approximation of the system state at time Δt after it is at state x is $x + v\Delta t$, where v is the velocity vector at the given time. Let $d_{\mathcal{B}} : \mathcal{X} \to [0, +\infty)$ denote the Euclidean distance from a state $x \in \mathcal{X}$ to set $\mathcal{B}$. Thus, a possible option for G is

$$G\left(\phi|_{[0,t]}, v\right) = \nabla_v d_{\mathcal{B}}\left(\phi(t)\right) = \lim_{\Delta t \to 0+} \frac{d_{\mathcal{B}}\left(\phi(t) + v\Delta t\right) - d_{\mathcal{B}}\left(\phi(t)\right)}{\Delta t}. \tag{4}$$

If $d_{\mathcal{B}}$ is not differentiable at $\phi(t)$, we replace $\lim$ in (4) by $\liminf$.

Let us now discuss a goodness function for the full reach-avoid problem. One naive goodness function uses the following motivation: let us partition the safe set $\mathcal{X}\backslash\mathcal{B}$ into a "boundary zone" $\mathcal{Z}^B$ and a "interior zone" $\mathcal{Z}^I$. If a system state is inside $\mathcal{Z}^B$, the system is considered at risk of leaving the safe set, and the primary objective is to avoid entering $\mathcal{B}$. If a system state is inside $\mathcal{Z}^I$, the system is not at risk, and the primary objective is to reach the target set $\mathcal{T}$. Then, a possible goodness function is given by $G\left(\phi|_{[0,t]}, v\right) = \nabla_v d_{\mathcal{B}}(\phi(t))$ if $\phi(t) \in \mathcal{Z}^B$ and $G\left(\phi|_{[0,t]}, v\right) = -\nabla_v d_{\mathcal{T}}(\phi(t))$ if $\phi(t) \in \mathcal{Z}^I$. This function is not continuous in $\phi(t)$, which will become important when determining bounds for suboptimality of a solution to the myopic optimal control problem in Section VI. A more subtle goodness function may use a measure of how much a system state is at risk (for instance, distance from $\mathcal{B}$), and then use a mixed goodness function between trying to enter set $\mathcal{T}$ and not leave $\mathcal{X}\backslash\mathcal{B}$: $G\left(\phi|_{[0,t]}, v\right) = (1-\lambda)\nabla_v d_{\mathcal{B}} - \lambda\nabla_v d_{\mathcal{T}}$, where $\lambda = d_{\mathcal{B}}\left(\phi(t)\right) / \max_y d_{\mathcal{B}}(y)$.

The choice of the goodness function rests with the designer, and is a natural vessel for the inclusion of side information that we may have about the system. Such side information may, for example, be available from knowing details about the damage that the system sustained. For instance, if the system is known to evolve on a particular subset of $\mathcal{X}$, the goodness function should be chosen in a way that is conscious of such information. Fig. 2 illustrates such a situation. While the naive goodness function described above would choose a direction that points towards the target set $\mathcal{T}$, but does not lead to it, a more informed goodness function would choose a counterintuitive direction that ultimately leads towards $\mathcal{T}$.

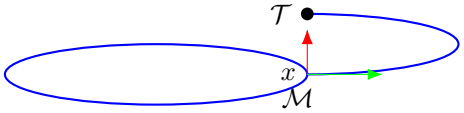

Fig. 2. An example of the role of side information in design of G. The control objective is to reach the set $\mathcal{T}$, and system trajectories are *a priori known* to evolve on the set $\mathcal{M} \subseteq \mathbb{R}^2$ (drawn in blue). Let x be the current system state, where the velocity vectors made possible by different control inputs are given by the red and green arrows. A naive goodness function would prefer the red arrow, but the available side information urges the designer to apply G which favors the direction indicated by the green arrow.

Remark 2. *In* (4) *and the subsequent paragraph, the proposed goodness functions are* memoryless*, i.e., $G(\phi_{[0,T]}, v)$ is solely a function of $(\phi(T), v)$. However, G may incorporate properties of the entire preceding system trajectory. This feature allows G to directly deal with a generalization of the reach-avoid problem where the system is required to reach a sequence of target sets $\mathcal{T}_1, \ldots, \mathcal{T}_n$ in order, and a further generalization to temporal logic specifications [20].*

V. Learn-Control Algorithm

We now approximately solve the myopic optimal control problem by the use of piecewise-constant control inputs. The proposed strategy relies on repeatedly noting the system response given the control input. It then uses this information to learn the approximate system dynamics at states on the observed system trajectory. Finally, it uses these dynamics to compute a control input that is approximately myopically optimal at the current system state. Such an input is then applied and used to note the current system response. This *learn-control cycle* is then repeated throughout the system run.

In order to learn the system dynamics at a single state on the trajectory, the controller can apply any $m + 1$ affinely independent constant inputs for a short period of time ε. Thus, the entire learn-control cycle is of length $\tau = (m + 1)\varepsilon$.

Since system (1) is control-affine, we can use the observed changes in the state during the k-th cycle to approximate the function $u \mapsto v_{\phi(k\tau)}(u)$ defined by $v_{\phi(k\tau)}(u) = f(\phi(k\tau)) + \sum_{i=1}^{m} g_i(\phi(k\tau))u_i$.

Let $\phi_{[0,k\tau]}$ denote the trajectory of the system until time $k\tau$. After obtaining an approximation for $v_{\phi(k\tau)}$, we can determine a myopically optimal control input $u^* \in \mathcal{U}$ by computing $u^* \in \operatorname{argmax}_u G(\phi, v_{\phi(k\tau)}(u))$. The next learn-control cycle uses an affinely independent set of inputs $u^*+\Delta u^0, \ldots, u^*+\Delta u^m$, where Δu^i are small enough — of magnitude δ — to ensure that their application still results in near-optimal goodness. For simplicity, we choose $\Delta u^i = \pm\delta e_i$, where the appropriate sign can always chosen so that $u^* + \Delta u^i \in \mathcal{U}$, with $\Delta u^i = +\delta e_i$ being the default choice. The above description is formalized as Algorithm 3.

Algorithm 3.

1 $u^* := 0$, $t_0 := 0$
2 **repeat**
3 Let $\Delta u^0 = 0$. Let $\Delta u^1 = \pm\delta e_1, \ldots, \Delta u^m = \pm\delta e_m$ with $u^* + \Delta u^1, \ldots, u^* + \Delta u^m \in \mathcal{U}$.
4 **for** $j = 0, \ldots, m$
5 Apply control $u^* + \Delta u^j$ in the interval of time $[t_0 + j\varepsilon, t_0 + (j+1)\varepsilon)$.
6 Let $x^{j+1} = \phi(t_0 + (j+1)\varepsilon)$.
7 **end for**
8 $x := x^{m+1}$
9 Define function $\tilde{v} : \mathcal{U} \to \mathbb{R}^n$ as follows: Let $\lambda_0, \ldots, \lambda_m$ be the unique coefficients with $\sum \lambda_j = 1$ such that $u = \sum \lambda_j (u^* + \Delta u^j)$. Let $\tilde{v}(u) = \sum \lambda_j (x^{j+1} - x^j)/\varepsilon$.
10 Let $u^* \in \operatorname{argmax} G(\phi|_{[0,t_0+\tau]}, \tilde{v}(u))$.
11 $t_0 := t_0 + \tau$
12 **until** the end of system run

As we will show in Section VI, with an appropriate choice of parameters δ and ε, Algorithm 3 produces a control signal that comes arbitrarily close to solving the myopic optimal control problem.

Remark 4. *Algorithm 3 does not distinguish between the learning and control phases of the algorithm: the algorithm learns v_x as a result of performing small perturbations of the previously established optimal control input u^*. These two phases can be decoupled by a minor modification of Algorithm 3. After learning v_x by applying* any $m+1$ *affinely independent control inputs in a short time interval of length $\tau' < \tau$, the system can then apply the u^* computed from these dynamics for the remaining $\tau - \tau'$ time in the learn-control cycle.*

VI. Performance Bounds

Unless otherwise noted, the proofs of technical results in this section are contained in Appendix B. The current section provides the intuition behind these proofs.

A. Learning of System Dynamics

We claim that lines 3–9 of Algorithm 3 produce a good approximation $\tilde{v} : \mathcal{U} \to \mathbb{R}^n$ of the map $v_x : \mathcal{U} \to \mathbb{R}^n$, with $x = x^{m+1}$ defined as in the algorithm. Let us first provide an intuitive explanation of the learning process.

As $\delta \leq 1$, for every control input $u^* \in \mathcal{U} = [-1, 1]^m$, $\{u_i^* - \delta, u_i^* + \delta\} \cap [-1, 1] \neq \emptyset$. Hence, for all $u^* \in \mathcal{U}$ we can indeed choose $\Delta u^i = \delta e_i$ or $\Delta u^i = -\delta e_i$ such that $u^* + \Delta u^i \in \mathcal{U}$, as stipulated by line 3 of Algorithm 3. We note that $u^* + \Delta u^i$, $i \in \{0, \ldots, m\}$, trivially form an affinely independent set. We denote $u^* + \Delta u^i$ by u^i.

For each $j \in \{0, 1, \ldots, m\}$, the vector $(x^{j+1} - x^j)/\varepsilon = (\phi_{u^j}(\varepsilon, x^j) - \phi_{u^j}(0, x^j))/\varepsilon$ approximates $d\phi_{u^j}(\varepsilon, x^j)/dt = v_{x^{j+1}}(u^j)$. Additionally, since x^{j+1} and $x = x^{m+1}$ are not far apart (because x^{m+1} is the state of the trajectory just $(m-j)\varepsilon$ later than x^{j+1}), $v_{x^{j+1}}(u^j) \approx v_x(u^j)$.

Since $u^0, \ldots, u^m$ are affinely independent, for every u there exist unique $\lambda_0, \ldots, \lambda_m \in \mathbb{R}$ such that $u = \lambda_0 u^0 + \ldots + \lambda_m u^m$ and $\lambda_0 + \ldots + \lambda_m = 1$. Then, $v_x(u) = \sum \lambda_j v_x(u^j)$ by the definition of v_x. Since we already have approximations for $v_x(u^j)$, we can thus approximate $v_x(u)$ by taking $v_x(u) \approx \sum_{j=0}^{m} \lambda_j (x^{j+1} - x^j)/\varepsilon$. Theorem 6 shows that such an approximation produces an error no worse than linear in ε. It is preceded by Lemma 5, which follows directly from the definition of ϕ_u as a solution to (1), definitions of M_0 and M_1, and triangle inequality.

Lemma 5. *Let u be the control signal produced in one repetition of lines 2–12 of Algorithm 3. Let x^j, $j \in \{0, \ldots, m+1\}$, be the corresponding system states during the same repetition, as defined in Algorithm 3. Let $x = x^{m+1}$. Then, the following holds:*

(i) For all $t_1, t_2 \in [0, (m+1)\varepsilon)$, $\|\phi_u(t_1, x^0) - \phi_u(t_2, x^0)\| \leq M_0(m+1)|t_1 - t_2|$. In particular, $\|x^j - x^k\| \leq M_0(m+1)|j-k|\varepsilon$ for all $j, k \in \{0, \ldots, m+1\}$.

(ii) For all $j \in \{0, \ldots, m\}$, $\|(x^{j+1} - x^j)/\varepsilon - v_{x^{j+1}}(u^j)\| \leq M_0 M_1 (m+1)^2 \varepsilon/2$.

(iii) For all $j \in \{0, \ldots, m\}$, $\left\|v_{x^{j+1}}(u^j) - v_x(u^j)\right\| \leq M_0 M_1 (m+1)^3 \varepsilon$.

Theorem 6. *Let $x^0, \ldots, x^{m+1} = x$ be as in Algorithm 3, and let $u \in \mathcal{U}$. Let $\lambda_0, \ldots, \lambda_m \in \mathbb{R}$ be such that $u = \lambda_0 u^0 + \ldots + \lambda_m u^m$ and $\lambda_0 + \ldots + \lambda_m = 1$. Then, $\|v_x(u) - \sum_{j=0}^{m} \lambda_j (x_{j+1} - x_j)/\varepsilon\| \leq 2M_0 M_1 (m+1)^3 \varepsilon(4m^{\frac{3}{2}} + \delta)/\delta$.*

B. Myopically Optimal Control Input

From Theorem 6, Algorithm 3 can calculate $v_{x^{m+1}}(u)$ for any $u \in \mathcal{U}$ with arbitrary precision. Thus, we are able to accurately calculate $G(\phi|_{[0,t_0+m+1\varepsilon]}, v_{x^{m+1}}(u))$ for any $u \in \mathcal{U}$. Additionally, for a fixed x and ϕ, $u \mapsto G(\phi, v_x(u))$ is a real function of a bounded variable $u \in \mathcal{U}$. The computational complexity of determining $\operatorname{argmax}_u G\big(\phi, \tilde{v}(u)\big)$ depends on G. For efficiency, one may choose G to be, e.g., convex or affine.

As the system dynamics around the current state constantly change with the change in state, control input that is optimal at the time when the system is at state x^{m+1} is not necessarily optimal at any later time. However, if the function G is "tame enough", u^* will still be a good approximation of the optimal control input even after the system leaves x^{m+1}. We first introduce a measure of tameness of G.

Definition 7. *Let* $\phi_1|_{[0,T_1]}, \phi_2|_{[0,T_2]} \in \mathcal{F}$ *and* $v_1, v_2 \in \mathbb{R}^n$. *Define* $d(\phi_1|_{[0,T_1]}, \phi_2|_{[0,T_2]}) = |T_1 - T_2| + \max_{t\in[0,\min(T_1,T_2)]} \|\phi_1(t) - \phi_2(t)\|$. *The function* $G : \mathcal{F} \times \mathbb{R}^n \to \mathbb{R}$ *has a* Lipschitz constant L *if* $|G(\phi_1|_{[0,T_1]}, v_1) - G(\phi_2|_{[0,T_2]}, v_2) \leq L(d(\phi_1|_{[0,T_1]}, \phi_2|_{[0,T_2]}) + \|v_1 - v_2\|)$ *for all* $\phi_1|_{[0,T_1]}, \phi_2|_{[0,T_2]} \in \mathcal{F}$ *and* $v_1, v_2 \in \mathbb{R}^n$.

While Definition 7 describes the standard notion of a Lipschitz constant with a distance function d, we note that d, as defined above, is not a metric on $\mathcal{F}$.

Theorem 8. *Let* $\phi_1|_{[0,T_1]}, \phi_2|_{[0,T_2]} \in \mathcal{F}$, $x = \phi_1(T_1)$, $y = \phi_2(T_2)$ *and* $\nu > 0$. *Let* L *be the Lipschitz constant of* G *and let* u^* *satisfy* $G(\phi_1|_{[0,T_1]}, \tilde{f} + \sum_{i=1}^m \tilde{g}_i u_i^*) = \max_{u\in\mathcal{U}} G(\phi_1|_{[0,T_1]}, \tilde{f} + \sum_{i=1}^m \tilde{g}_i u_i)$, *where* $\|\tilde{f} + \sum_{i=1}^m \tilde{g}_i u_i - (f(x) + \sum_{i=1}^m g_i(x)u_i)\| \leq \nu$ *for all* $u \in \mathcal{U}$. *Then,* $|\max_u G(\phi_2|_{[0,T_2]}, v_y(u)) - G(\phi_2|_{[0,T_2]}, v_y(u^*))| \leq 2Ld(\phi_1|_{[0,T_1]}, \phi_2|_{[0,T_2]}) + 2LM_1(m+1)\|x - y\| + 2L\nu$.

Theorem 8 provides a bound on the suboptimality of the control input u^*. However, in Algorithm 3 we do not apply just u^* as the control; we modify this u^* by some small Δu^i. Corollary 9 deals with this issue.

Corollary 9. *Assume the same notation as in Theorem 8. Let* $0 < \delta \leq 1$ *and let* $\tilde{u} \in \mathbb{R}^m$, $\|\tilde{u}\| \leq \delta$. *Then,* $|\max_u G(\phi_2|_{[0,T_2]}, v_y(u)) - G(\phi_2|_{[0,T_2]}, v_y(u^* + \tilde{u}))| \leq 2Ld(\phi_1|_{[0,T_1]}, \phi_2|_{[0,T_2]}) + 2LM_1(m+1)\|x - y\| + 2L\nu + LM_0(m+1)\delta$.

Finally, Theorem 10 provides a bound on the degree of suboptimality of Algorithm 3 in terms of algorithm parameters ε and δ. Solely for notational purposes, we assume that the system run is of infinite length.

Theorem 10. *Let* $u^+ : [0, +\infty) \to \mathcal{U}$ *be the control signal used in Algorithm 3. Let* L *be the Lipschitz constant of* G. *Then, for all* $t \geq (m+1)\varepsilon$, (3) *holds for* $\mu = 6L(M_0 + 1)(M_1 + 1)(m+1)^3(1 + (4m\sqrt{m})/\delta)\varepsilon + LM_0(m+1)\delta$.

Theorem 10 is the central result of the theoretical discussions of this paper. It shows that, for any $\mu > 0$, if ε and δ are properly chosen, Algorithm 3 will result in a control signal that approximately satisfies (2), with error no larger than μ.

Making use of the bound in Theorem 10 requires the ability to make ε and δ arbitrarily small, i.e., an arbitrarily fine time and actuation resolution of the control input. While these resolutions depend on the physical properties of the system actuator, developing actuators with fine resolutions for both time and actuation is a topic of significant recent research [21], [22].

Remark 11. *We note that Algorithm 3 is based on the system learning the system dynamics at a state anew during each learn-control cycle. We revisit this property in a later discussion on future work. A beneficial consequence of such a property is the ability of Algorithm 3 to account for the presence of time-varying disturbances in system dynamics. Consider a disturbance function* $D : [0, +\infty) \times \mathcal{X} \to \mathbb{R}^n$ *such that* $\dot{x} = f(x) + \sum_{i=1}^m g_i(x)u_i + D(t, x)$. *We assume that* D *is a Lipschitz continuous function with known bounds on its size and Lipschitz constant. By appending an additional variable* x_{n+1} *given by the dynamics* $dx_{n+1}/dt = 1$ *and* $x_{n+1}(0) = 0$, *and with an obvious abuse of notation, we obtain dynamics*

$$\dot{x} = \begin{bmatrix} f(x) + D(x) \\ 1 \end{bmatrix} + \sum_{i=1}^m \begin{bmatrix} g_i(x) \\ 0 \end{bmatrix} u_i. \tag{5}$$

Assuming that the system governed by dynamics (5) *still evolves on* $\mathcal{X}$, *Algorithm 3, with updated bounds* M_0', M_1', *remains valid.*

VII. Simulation Results

A. Damaged Aircraft Example

The simulation work presented in this section describes the scenario of a damaged aircraft that is attempting to retain a safe altitude. We consider a Boeing 747-200 that lost 33% of its right wing. The dynamics of such an aircraft were developed in [23], and we use the nonlinear model contained therein to simulate aircraft behavior. While this motivating example demonstrates the myopic control procedure, a number of the assumptions that go into establishing the bounds in Theorem 10 do not hold for the dynamics at hand. Thus, we include this indicative simulation to demonstrate the efficacy of myopic control in a real-world application. We further emphasize that we do not use these dynamics at any point to decide on an appropriate myopic control signal: the controller is ignorant of the true system dynamics and bases its decisions on learned system dynamics as described in Algorithm 3.

The state variables $x = [v,\ w,\ q,\ \theta,\ \beta,\ p,\ r,\ \phi,\ z]^T$ that we consider are forward velocity, vertical velocity, pitch rate, pitch angle, sideslip angle, roll rate, yaw rate, roll angle, and altitude, respectively. The control inputs $u = [\delta_e,\ \delta_a,\ \delta_r]$ are the elevator, aileron, and rudder deflections in degrees from the wings-level trim condition for an undamaged aircraft, respectively. In the remainder of the paper, we denote the appropriate coordinates of x by $x_1, \ldots, x_9$, and analogously for u. The bounds on allowed control inputs $u \in \mathcal{U}$ are set to $u_1 = \delta_e \in [-15, 15]$, $u_2 = \delta_a \in [-25, 25]$ and $u_3 = \delta_r \in [-10, 10]$, roughly informed by [24].

1) Initial state and specifications: The initial flight conditions equal the trim conditions of a Boeing 747-200 at 283000 kg, 500 knots and a nominal altitude of 500 m [25]. The setting of this example is that the aircraft suffered damage while in flight, during a banked turn, at an angle of 28.6 degrees (0.5 radians). Thus, the initial system state is $x(0) = [257.22, -0.7818, 0, 2.5, 0, 0, 0, 28.6, 500]^T$ (all values are in meters, seconds and degrees). The aircraft's primary objective (O1) is to remain in the air: $x_9(t) > 0$ for all $t \geq 0$. We also desire the aircraft to reach and remain within safe altitude bounds, and remain nearly horizontal: objective (O2) is $x_9(t) \in [1900, 2100]$ for all $t \geq T$, and (O3) is $x_8(t) \in [-5, 5]$ for all $t \geq T$, with $T > 0$ as small as possible. Additionally, we require adherence to the following safety specifications based upon physical constraints of an aircraft [26]: objective (O4a) is $x_4(t) \in [-10, 10]$ for all $t \geq 0$, (O4b) is $x_2(t) \in [-50, 50]$ for all $t \geq 0$. We impose the following importance ranking of specifications, in descending order: (O1), (O4a), (O4b), (O3), (O2).

2) Goodness function design: The following conditional function describes the ideal aircraft behavior, given objectives (O1)–(O4).

if $z = x_9 < 100$:	x_9 should quickly increase,
else if $\lvert x_4\rvert > 10$ or $\lvert x_2\rvert > 50$:	$\lvert x_2\rvert$, $\lvert x_4\rvert$ or both (as needed) should quickly decrease,
else if $\lvert x_8\rvert > 5$:	x_8 should quickly approach 0,
else if $x_9 \notin [1900, 2100]$:	x_9 should quickly approach 2000,
else:	$\lvert x_4\rvert$ and $\lvert x_2\rvert$ should remain small.

Defining a goodness function G to achieve the above behavior would be simple if we had full knowledge of the system dynamics. However, the only a priori knowledge about system dynamics comes from physical laws and basic understanding of aircraft control inputs. In particular, we know the following:

(i) u_1 will have the most effect on x_2 and x_3 (by the design of the elevator);
(ii) u_2 and u_3 will have the most effect on x_5, x_6 and x_7 (by the design of the ailerons and rudder);
(iii) $\dot{x}_4 = x_3$ and $\dot{x}_8 = x_6$ (by definition);
(iv) an increase in either x_2 or x_4 will increase $\dot{x}_9$ (by longitudinal force definitions) and a decrease in x_6 leads to an increase in $\dot{x}_8$ (by lateral force definitions);
(v) u_1 does not directly influence x_4 and x_9, but instead acts on them through x_2 and x_3 (by Newton's second law on the longitudinal forces), and u_2 and u_3 do not directly influence x_8, but instead act on it through x_6 (by Newton's second law on the lateral forces).

While we omit further details, G can now be formally designed using the above facts as follows:

$$G(x|_{[0,t]}, v) = \begin{cases} v_2 + v_3, & x_9(t) < 100, \\ m_1 + m_2 + m_3, & x(t) \in \mathcal{P}, \\ (v_2 + v_3)|v_8| \operatorname{sign}(c - x_9(t)) & \text{otherwise}, \end{cases}$$

where $c = 2000$, $\mathcal{P} = \{x \in \mathbb{R}^9 \mid x_9 \geq 100\} \cap (\{x \in \mathbb{R}^9 \mid |x_4| > 10\} \cup \{x \in \mathbb{R}^9 \mid |x_2| > 50\} \vee \{x \in \mathbb{R}^9 \mid |x_8| > 5\} \vee \{x \in \mathbb{R}^9 \mid |x_9 - 2000| \leq 100\})$, $m_1 = -v_2 \operatorname{sign}(x_2(t)) \mathbb{1}(|x_2(t)| - 1)$, $m_2 = -v_2 \operatorname{sign}(x_4) \mathbb{1}(|x_4(t)| - 0.5)$, and $m_3 = -v_6 \operatorname{sign}(x_8) \mathbb{1}(|x_8(t)| - 0.5)$.

We emphasize that the above goodness function G is not the only possible one. The proposed function has the benefit of being simple to design, but does not have a Lipschitz constant in the sense of Definition 7. Thus, we are unable to directly use the results of Section VI to determine good choices of parameters ε and δ. Instead, we describe the correct choice of ε and δ in the following paragraph. Additionally, instead of applying Algorithm 3 directly, we employ it as described in Remark 4.

3) Parameter selection: In selecting parameters ε and δ, we seek to account for scenario-specific issues such as the non-minimum phase of response of x_2 to control u_1. As most aircraft are natural low-pass filters [27], the learning period has to be of sufficient length to observe the system response. On the other hand, the learning period must not be too long, or the wiggle size δ too large, that they impact the performance of the control signal in the middle of the learn-control cycle. In this simulation, we decoupled the control of the longitudinal modes of motion from the lateral [28], and chose the length of the learn-control cycle to be $\varepsilon = 1$ s and $\varepsilon = 0.1$ s with the learning period, as defined in Remark 4, equal to $\varepsilon' = 0.1$ s and $\varepsilon' = 10^{-2}$ s for the longitudinal and lateral cases, respectively. We chose $\delta = 5$.

4) Results: The simulation results are presented in Fig. 3. The myopic control strategy performs exactly as desired, even though the true dynamics are not control-affine. The peaks and troughs that result in the trajectory leaving the desired bounds can be avoided by a more careful design of the goodness function. We did not make such changes in order to emphasize that the controller performs as intended: as soon as the system state leaves the desired bounds, the controller takes immediate corrective action.

The aircraft's oscillatory behavior after reaching the altitude bounds is due to the phugoid and Dutch roll aircraft dynamic modes. Designing a goodness function to control for such behavior is possible, but is outside of the scope of this paper.

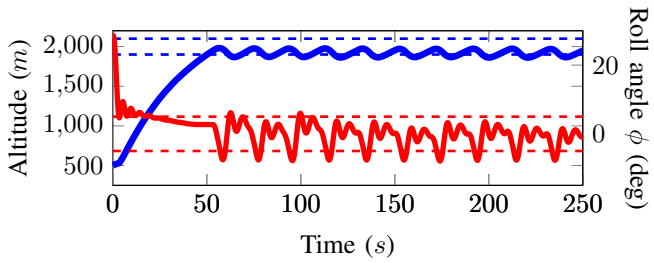

Fig. 3. The results of a damaged aircraft simulation. The altitude and roll angle are denoted by blue and red lines, respectively. The corresponding desired bounds are denoted by dashed lines. A shortened video of the simulation is available at https://goo.gl/a4qYAU.

B. Van der Pol Oscillator

We consider the system given by $\dot{x}_1 = x_2$, $\dot{x}_2 = -x_1 - 2(1 - x_1^2)x_2 + u$, with the initial value $x(0) = (1, -2)$ and control input bounds $u \in [-2, 2]$. This control system models a Van der Pol oscillator with control input [29].

The control specification in this example is to drive and retain the value of x_2 around 0. The design of goodness function G proceeds similarly to Section VII-A: we obtain $G(x|_{[0,t]}, v) = -v_2 \operatorname{sign}(x_2(t))$. However, as the results in Section VI require G to be continuous, we mollify this function and take $G(x|_{[0,t]}, v) = -v_2 \arctan(x_2(t))$.

While the system is evolving on all of $\mathbb{R}^2$, we will assume that we are only interested in the dynamics that take place in $\mathcal{X} = [-5, 5]^2$; even though $\mathcal{X}$ is not strongly forward invariant, we can use the developed theoretical results while the system remains in such this set. For such $\mathcal{X}$, it is not difficult to show that $M_0 \leq 250$, $M_1 \leq 99$, $L \leq 29$ and, by plugging these values into Theorem 10, obtain the bound on the myopic suboptimality of the control signal generated by Algorithm 3. Using this bound, simple calculations show that in order to guarantee that $x_2(t)$ will even start moving in the right direction (after the first ε period of time), it is necessary

to have $\varepsilon < 10^{-7}$ and $\delta < 10^{-4}$. However, these bounds are very conservative; simulations show that ε and δ can be larger by several orders of magnitude without impacting the performance of the system. Fig. 4 shows the simulation results with $\varepsilon = 10^{-4}$ and $\delta = 10^{-3}$.

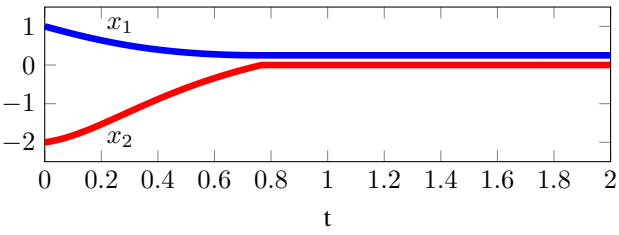

Fig. 4. Graphs of the simulated trajectory from the setting of Section VII-B. The blue and red line represent the values of x_1 and x_2 over time, respectively.

The myopic control strategy performs exactly as desired. We also note that x_1 also remains within 10^{-3} of a particular value. This feature was not guaranteed nor intended by the design of the goodness function, but is a consequence of the fact that x_2 remains around 0 and repeatedly switches signs.

VIII. Future Work

We conclude this paper by listing some remaining open issues and possible future directions of work based.

- The paper makes an assumption on the control-affine structure (1) of the control system. A possible extension of the work in this paper would be to adapt Algorithm 3 to other systems with a structure that allows learning of control dynamics at a state from a small number of tested control inputs; e.g., systems polynomial in control [30].
- The case of partial or noisy observations of the system trajectory is of significant interest. A possible approach is to use $u^* + \Delta u^0, \ldots, u^* + \Delta u^m$ in such a way that, while each control only reveals an incomplete system state, the information gathered from all the control inputs combined allows the controller to reconstruct the full state.
- Algorithm 3 relies on continually relearning the system dynamics at the current system state, and discarding the previously learned dynamics. We are interested in developing a control strategy that uses knowledge of previously learned system dynamics to obtain guarantees for long-term system behavior and reduce the amount of learning necessary.
- The primary question still left partly unexplored by this paper concerns the choice of G, potentially in the presence of side information about the system. It would be useful to provide a formal relationship between a solution to the myopic optimal control problem, for a particular function G, and solutions to the reach-avoid problem.

Appendix A
Existence of a Myopically Optimal Control Signal

Let $\tilde{\mathcal{S}} : \mathcal{F} \to 2^{\mathcal{U}}$ be a set-valued map defined by $\tilde{\mathcal{S}}\left(\phi|_{[0,t]}\right) = \operatorname{argmax}_{u\in\mathcal{U}} G\left(\phi|_{[0,t]}, v_{\phi(t)}(u)\right)$. Then, the myopic optimal control problem has a solution if and only if the differential inclusion $\dot{x}(t) \in \{f(x(t)) + \sum_{i=1}^m g_i(x(t))u_i \mid u \in \tilde{\mathcal{S}}(x|_{[0,t]})\}$, $x(0) = x_0$, admits a solution.

Let us define a set-valued map $S : \mathcal{F} \to 2^{\mathbb{R}^n}$ as the right-hand side of the above differential inclusion. Then, a solution to the myopic optimal control problem exists if and only if there exists a solution to $\dot{x}(t) \in S(x|_{[0,t]})$, $x(0) = x_0$. The latter inclusion is a delay differential inclusion in the sense of [31]. Sufficient conditions for the existence of a solution to this differential inclusion have been previously identified. In particular, Theorem 12 presents an adaptation of a result in [32]. We define the norm on $C([0,t],\mathbb{R}^n)$ by $\|\phi\| = \max_{\tau\in[0,t]} \|\phi(\tau)\|$, and point the reader to [32] for definitions of hemicontinuity and Lebesgue measurability of set-valued maps.

Theorem 12. *Let $\gamma > 0$, $\mathcal{X} = \mathbb{R}^n$. Assume the following:*

(i) $S(\phi|_{[0,t]}) \subseteq \mathbb{R}^n$ is a convex set for all $t \in [0,\gamma]$ and all continuous maps $\phi|_{[0,t]} : [0,t] \to \mathcal{X}$.

(ii) For any fixed $t \in [0,\gamma]$, the restriction of the map S to $C([0,t],\mathcal{X})$ is upper hemicontinuous.

(iii) For any fixed $\phi \in C([0,\gamma],\mathcal{X})$, $S_\phi : [0,\gamma] \to 2^{\mathbb{R}^n}$ defined by $S_\phi(t) = S(\phi_{[0,t]})$ is a Lebesgue-measurable map.

(iv) There exists $k > 0$ such that, for every absolutely continuous function $\phi \in C([0,\gamma],\mathcal{X})$, every $t \in [0,\gamma]$ such that $d\phi(t)/dt$ exists, and every $y \in S(\phi|_{[0,t]})$, it holds that $y^T(d\phi(t)/dt) \le k(1 + \|\phi|_{[0,t]}\|^2)$.

Then, $\dot{x}(t) \in S(x|_{[0,t]})$, $x(0) = x_0$ has a solution on $[0,\gamma]$.

The proof of Theorem 12 follows immediately from the result in [32]. A slightly more permissive, but significantly less elegant, sufficient condition has been identified in [31].

Conditions of Theorem 12 are expressed in terms of the set-valued map S. We note that S is related to the problem data, i.e., functions f, g_i, and G, by the discussion at the beginning of this appendix.

Appendix B
Proofs of Performance Bounds

Proof of Theorem 6. From $u = \sum \lambda_j u^j = \sum \lambda_j (u^* + \Delta u^j)$ and $\sum \lambda_j = 1$ we obtain $u = u^* + \lambda_1 \Delta u^1 + \ldots + \lambda_m \Delta u^m$. By definition of Δu^j from line 3 of Algorithm 3, we have $(u - u^*)_j = \pm\lambda_j \delta$ for all $j \ge 1$. Hence, $|\pm\lambda_j\delta| < \|u - u^*\|$ for all $j \ge 1$. Since $u, u^* \in \mathcal{U} = [-1,1]^m$, we obtain $|\lambda_j| \le 2\sqrt{m}/\delta$ for all $j \ge 1$. By using $\sum \lambda_j = 1$ and triangle inequality, we get $|\lambda_0| \le 1 + 2m\sqrt{m}/\delta$.

Since $v_x(u) = \sum \lambda_j v_x(u^j)$, we have $\|v_x(u) - \sum \lambda_j (x^{j+1} - x^j)/\varepsilon\| \le \sum |\lambda_j| \|v_x(u^j) - (x^{j+1} - x^j)/\varepsilon\|$. Combining parts (ii) and (iii) of Lemma 5, the right hand side of this inequality can be bounded by $\sum |\lambda_j| M_0 M_1 ((m+1)^2/2 + (m+1)^3)\varepsilon \le (1 + 4m\sqrt{m}/\delta) 2 M_0 M_1 (m+1)^3 \varepsilon$, where the last inequality was obtained by using above bounds on λ_j. □

Proof of Theorem 8. Let $\overline{u}$ satisfy $G(\phi_2|_{[0,T_2]}, v_y(\overline{u})) = \max_{u\in\mathcal{U}} G(\phi_2|_{[0,T_2]}, v_y(u))$. We note that $\|v_y(\overline{u}) - \tilde{f} - \sum_{i=1}^m \tilde{g}_i \overline{u}_i\| \le \|v_y(\overline{u}) - v_x(\overline{u})\| + \|v_x(\overline{u}) - \tilde{f} - \sum_{i=1}^m \tilde{g}_i \overline{u}_i\| \le (m+1) M_1 \|y - x\| + \nu$. Hence, $|G(\phi_2|_{[0,T_2]}, v_y(\overline{u})) - G(\phi_1|_{[0,T_1]}, \tilde{f} + \sum_{i=1}^m \tilde{g}_i \overline{u}_i)| \le L(d(\phi_1|_{[0,T_1]}, \phi_2|_{[0,T_2]}) + \|v_y(\overline{u}) - \tilde{f} - \sum_{i=1}^m \tilde{g}_i \overline{u}_i\|) \le L d(\phi_1|_{[0,T_1]}, \phi_2|_{[0,T_2]}) +$

$LM_1(m+1)\|x-y\| + L\nu$. Analogously, $|G(\phi_1|_{[0,T_1]}, \tilde{f} + \sum_{i=1}^m \tilde{g}_i u_i^*) - G(\phi_2|_{[0,T_2]}, v_y(u^*))| \leq Ld(\phi_1|_{[0,T_1]}, \phi_2|_{[0,T_2]}) + L(m+1)M_1\|x-y\| + L\nu$.

Now, if $|G(\phi_2|_{[0,T_2]}, v_y(\overline{u})) - G(\phi_1|_{[0,T_1]}, \tilde{f} + \sum_{i=1}^m \tilde{g}_i u_i^*)| > Ld\left(\phi_1|_{[0,T_1]}, \phi_2|_{[0,T_2]}\right) + L(m+1)M_1\|x-y\| + L\nu$, by combining this inequality with the previous two inequalities, we would obtain that $G(\phi_1|_{[0,T_1]}, \tilde{f} + \sum_{i=1}^m \tilde{g}_i \overline{u}_i) > G(\phi_1|_{[0,T_1]}, \tilde{f} + \sum_{i=1}^m \tilde{g}_i u_i^*)$ or $G(\phi_2|_{[0,T_2]}, v_y(u^*)) > G(\phi_2|_{[0,T_2]}, v_y(\overline{u}))$, which are both contradictions with the definitions of u^* and $\overline{u}$, respectively. Thus, $|G(\phi_2|_{[0,T_2]}, v_y(\overline{u})) - G(\phi_1|_{[0,T_1]}, \tilde{f} + \sum_{i=1}^m \tilde{g}_i u_i^*)| \leq Ld\left(\phi_1|_{[0,T_1]}, \phi_2|_{[0,T_2]}\right) + L(m+1)M_1\|x-y\| + L\nu$. By combining the above inequality with the previously obtained bound on $|G(\phi_1|_{[0,T_1]}, \tilde{f} + \sum_{i=1}^m \tilde{g}_i u_i^*) - G(\phi_2|_{[0,T_2]}, v_y(u^*))|$, as well as $|G(\phi_2|_{[0,T_2]}, v_y(\overline{u})) - G(\phi_2|_{[0,T_2]}, v_y(u^*))| \leq |G(\phi_2|_{[0,T_2]}, v_y(\overline{u})) - G(\phi_1|_{[0,T_1]}, \tilde{f} + \sum_{i=1}^m \tilde{g}_i u_i^*)| + |G(\phi_1|_{[0,T_1]}, \tilde{f} + \sum_{i=1}^m \tilde{g}_i u_i^*) - G(\phi_2|_{[0,T_2]}, v_y(u^*))|$, we obtain the theorem claim. □

Proof of Corollary 9. Using the same notation as in the proof of Theorem 8, we first obtain $|G(\phi_2|_{[0,T_2]}, v_y(\overline{u})) - G(\phi_2|_{[0,T_2]}, v_y(u^* + \tilde{u}))| \leq |G(\phi_2|_{[0,T_2]}, v_y(\overline{u})) - G(\phi_2|_{[0,T_2]}, v_y(u^*))| + |G(\phi_2|_{[0,T_2]}, v_y(u^*)) - G(\phi_2|_{[0,T_2]}, v_y(u^* + \tilde{u}))|$. From Theorem 8, the first summand on the right hand side of the above inequality is bounded by $2Ld\left(\phi_1|_{[0,T_1]}, \phi_2|_{[0,T_2]}\right) + 2LM_1(m+1)\|x-y\| + 2L\nu$. By the definition of v_y and L, the second summand is bounded by $L\|\sum_{i=1}^m g_i(y)\tilde{u}_i\| < LM_0(m+1)\delta$. □

Proof of Theorem 10. Let $t \geq (m+1)\varepsilon$. Then, t is contained in one repetition of lines 2–12 of Algorithm 3, and this repetition is not the first one. Let $x^0, \ldots, x^{m+1}$ be the x^j's used in that repetition. From Theorem 6, the previous repetition resulted in an approximation $\tilde{f} + \sum \tilde{g}_i u_i$ of $v_{x^0}(u)$ which satisfies $\|v_{x^0}(u) - (\tilde{f} + \sum \tilde{g}_i u_i)\| \leq 2M_0M_1(m+1)^3(1 + 4m^{3/2}/\delta)\varepsilon$ for all $u \in \mathcal{U}$ and in a control input $u^* \in \mathcal{U}$ optimal for dynamics $u \to \tilde{f} + \sum \tilde{g}_i u_i$.

At time t, $u^+(t) = u^* + \Delta u^j$, for some $j \in \{0, \ldots, m\}$. Thus, we can apply Corollary 9 for $\nu = 2M_0M_1(m+1)^3(1+4m^{3/2}/\delta)\varepsilon$. Hence, $|\max_u G(\phi_{u^+}(\cdot, x_0)|_{[0,t]}, v_x(u)) - G(\phi_{u^+}(\cdot, x_0)|_{[0,t]}, v_x(u^+))| \leq LM_0(m+1)\delta + 4LM_0M_1(m+1)^3(1+4m^{3/2}/\delta)\varepsilon + 2Ld(\phi_{u^+}(\cdot, x_0)|_{[0,t]}, \phi_{u^+}(\cdot, x_0)|_{[0,t_0]}) + 2LM_1(m+1)\|x - x^0\|$, where t_0 is the time at the beginning of the current repetition of lines 2–12 of Algorithm 3.

By Definition 7, $d\left(\phi_{u^+}(\cdot, x_0)|_{[0,t]}, \phi_{u^+}(\cdot, x_0)|_{[0,t_0]}\right) = t - t_0 \in [0, (m+1)\varepsilon)$. By part (i) of Lemma 5, $\|x - x^0\| \leq M_0(m+1)^2\varepsilon$. Hence, from the bound in the previous paragraph, $|\max_u G(x, v_x(u)) - G(x, v_x(u^+))| \leq 6L(M_0 + 1)(M_1+1)(m+1)^3(1+4m^{3/2}/\delta)\varepsilon + LM_0(m+1)\delta$. □

References

[1] M. Ornik, S. Carr, A. Israel, and U. Topcu, "Myopic control of systems with unknown dynamics," in *American Control Conference*, 2019, pp. 1064–1071.

[2] S. Aloni, *Israeli F-15 Eagle Units in Combat*. Osprey Publishing, 2006.

[3] H.-H. Yeh, S. S. Banda, and P. J. Lynch, "Control of unknown systems via deconvolution," *Dyn. Control*, vol. 13, no. 3, pp. 416–423, 1990.

[4] D. P. Solomatine and A. Ostfeld, "Data-driven modelling: some past experiences and new approaches," *J. Hydroinf.*, vol. 10, no. 1, pp. 3–22, 2008.

[5] M. Schmidt and H. Lipson, "Distilling free-form natural laws from experimental data," *Science*, vol. 324, no. 5923, pp. 81–85, 2009.

[6] H. van Hasselt, "Reinforcement learning in continuous state and action spaces," in *Reinforcement Learning: State-of-the-Art*, M. Wiering and M. van Otterlo, Eds. Springer, 2012, pp. 207–251.

[7] A. Faust, P. Ruymgaart, M. Salman, R. Fierro, and L. Tapia, "Continuous action reinforcement learning for control-affine systems with unknown dynamics," *IEEE/CAA J. Autom. Sin.*, vol. 1, no. 3, pp. 323–336, 2014.

[8] N. T. Nguyen, K. S. Krishnakumar, J. T. Kaneshige, and P. P. Nespeca, "Flight dynamics and hybrid adaptive control of damaged aircraft," *J. Guid. Control Dyn.*, vol. 31, no. 3, pp. 751–764, 2008.

[9] Y. Liu, G. Tao, and S. M. Joshi, "Modeling and model reference adaptive control of aircraft with asymmetric damage," *J. Guid. Control Dyn.*, vol. 33, no. 5, pp. 1500–1517, 2010.

[10] S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Discovering governing equations from data by sparse identification of nonlinear dynamical systems," *Proc. Natl. Acad. Sci. U.S.A.*, vol. 113, no. 15, pp. 3932–3937, 2016.

[11] M. Ahmadi, A. Israel, and U. Topcu, "Safety assessemt based on physically-viable data-driven models," in *56th IEEE Conference on Decision and Control*, 2017, pp. 6409–6414.

[12] Z. Zhou, R. Takei, H. Huang, and C. J. Tomlin, "A general, open-loop formulation for reach-avoid games," in *51st IEEE Conference on Decision and Control*, 2012, pp. 6501–6506.

[13] B. L. Stevens, F. L. Lewis, and E. N. Johnson, *Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems*. Wiley, 2016.

[14] W. L. Garrard and J. M. Jordan, "Design of nonlinear automatic flight control systems," *Automatica*, vol. 13, no. 5, pp. 497–505, 1977.

[15] A. Mokhtari, A. Benallegue, and Y. Orlov, "Exact linearization and sliding mode observer for a quadrotor unmanned aerial vehicle," *Int. J. Rob. Autom.*, vol. 21, no. 1, pp. 39–49, 2006.

[16] F. Fadaie and M. E. Broucke, "A viability problem for control affine systems with application to collision avoidance," in *45th IEEE Conference on Decision and Control*, 2006, pp. 5998–6003.

[17] R. C. Dorf and R. H. Bishop, *Modern Control Systems*. Prentice Hall, 2011.

[18] H. Kwakernaak and R. Sivan, *Linear Optimal Control Systems*. Wiley, 1972.

[19] P. M. Esfahani, D. Chatterjee, and J. Lygeros, "The stochastic reach-avoid problem and set characterization for diffusions," *Automatica*, vol. 70, pp. 43–56, 2016.

[20] Z. Manna and A. Pnueli, *The Temporal Logic of Reactive and Concurrent Systems*. Springer, 1992.

[21] H.-A. Eckel, S. Scharring, S. Karg, C. Illg, and J. Peter, "Overview of laser ablation micropropulsion research activities at DLR Stuttgart," in *International Symposium on High Power Laser Ablation and Beamed Energy Propulsion*, 2014.

[22] D. Krejci, F. Mier-Hicks, R. Thomas, T. Haag, and P. Lozano, "Emission characteristics of passively fed electrospray microthrusters with propellant reservoirs," *J. Spacecr. Rockets*, vol. 54, no. 2, pp. 447–458, 2017.

[23] M. Arruda, "Dynamic inverse resiliant control for damaged asymmetric aircraft: Modeling and simulation," Ph.D. dissertation, Wichita State University, 2009.

[24] S. Ganguli, A. Marcos, and G. Balas, "Reconfigurable LPV control design for Boeing 747-100/200 longitudinal axis," in *American Control Conference*, 2002, pp. 3612–3617.

[25] T. T. Ogunwa and E. J. Abdullah, "Flight dynamics and control modelling of damaged asymmetric aircraft," *IOP Conference Series: Materials Science and Engineering*, vol. 152, no. 1, p. 012022, 2016.

[26] S. S. Mulgund and R. F. Stengel, "Target pitch angle for the microburst escape maneuver," *Journal of Aircraft*, vol. 30, no. 6, pp. 826–832, 1993.

[27] T. R. Yechout, *Introduction to aircraft flight mechanics*. AIAA, 2003.

[28] B. Etkin and L. D. Reid, *Dynamics of Flight: Stability and Control*. Wiley, 1995.

[29] M. S. Dutra, A. C. de Pina Filho, and V. F. Romano, "Modeling of a bipedal locomotor using coupled nonlinear oscillators of Van der Pol," *Biol. Cybern.*, vol. 88, no. 4, pp. 286–292, 2003.

[30] E. Moulay and W. Perruquetti, "Stabilization of nonaffine systems: A constructive method for polynomial systems," *IEEE Trans. Autom. Control*, vol. 50, no. 4, pp. 520–526, 2005.

[31] L. Boudjenah, "Existence of solutions to differential inclusions with delayed arguments," *Electr. J. Differ. Equ.*, vol. 2010, no. 175, pp. 1–8, 2010.

[32] B. I. Anan'ev, "An existence theorem for a differential inclusion with variable lag," *Differentsial'nye uravneniya*, vol. 11, no. 7, pp. 1155–1158, 1975.