This paper has been submitted for presentation at *American Control Conference 2024*.

This is the author's version of an article that has, or will be, published in this journal or conference. Changes were, or will be, made to this version by the publisher prior to publication.

# Blameless and Optimal Control under Prioritized Safety Constraints

Natalia Pavlasek[1], Sarah H.Q. Li[2], Behçet Açıkmeşe[1], Meeko Oishi[3], and Claus Danielson[3]

*Abstract*— In many resource-limited optimal control problems, multiple constraints may be enforced that are jointly infeasible due to external factors such as subsystem failures, unexpected disturbances, or fuel limitations. In this manuscript, we introduce the concept of *blameless optimality* to characterize control actions that a) satisfy the highest prioritized and feasible constraints and b) remain optimal with respect to a mission objective. For a general optimal control problem with jointly infeasible constraints, we prove that a single optimization problem cannot find a blamelessly optimal control sequence. Instead, finding blamelessly optimal control actions requires sequentially solving at least two optimal control problems: one to determine the highest priority level of constraints that is feasible and another to determine the optimal control action with respect to these constraints. We apply our results to a rocket landing scenario in which violating at least one safety-induced landing constraint is unavoidable. Leveraging the concept of blameless optimality, we formulate blamelessly optimal controllers that can autonomously prioritize the constraints most critical to a mission.

## I. INTRODUCTION

Consider a scenario in which a planetary lander is not able to perform its primary landing and must instead autonomously select a landing site. In order to make the best use of resources and accomplish the greatest number of mission goals possible, the lander should select a site by evaluating the potential benefits, such as safety of the landing site, or proximity to sites of interest. Such an evaluation of trade-offs typically rests on an ordering of priorities: first protecting the lander from damage, then attempting to achieve the highest priority mission goals. Designing autonomous systems to adhere to a prioritization that reflects operational choices is an important yet largely unexplored problem [1], that reflects upon the autonomous systems' perceived reliability, trustworthiness, and overall effectiveness [2]. A lander whose actions reflect mis-ordered priorities (i.e., selecting a landing site that results in damage to the vehicle at the cost of being close to a site of interest, for example) would be considered misguided and even blameworthy, as its actions are in conflict with priorities.

In this paper, we propose the design of controllers for autonomous systems, that are both *optimal* and *blameless*. We interpret blamelessness, based on a formal description
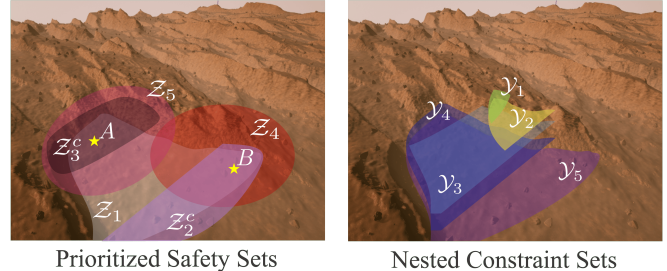
Fig. 1: Optimal controllers for a lander should satisfy constraints in order of their priority, in order to be *blameless*.

in [3], as the ability to avoid actions whose outcomes are inconsistent with an ordered prioritization of constraints. Such mis-prioritizations only appear when not all constraints can be satisfied, that is, when the system operates in regions of the state-space that are infeasible with respect to some constraints. For instance, in the landing scenario in Figure 1, if a landing zone that satisfies all safety constraints exists, an alternative landing site does not have to be considered.

While extensive work has considered finding optimal solutions under constraints, considerably less work has been done on designing solutions under *infeasible* constraints. Unlike standard optimal control problems, in which infeasibility is commonly averted by softening the constraints [4], controller design under infeasibility must focus not only on satisfying as many constraints as possible, but also on satisfying higher priority constraints over ones that are lower priority. That is, ensuring blamelessness in autonomous systems means that controllers must be designed to satisfy constraints in a manner that reflects their underlying prioritization. Further, due to the urgency associated with operation in infeasible scenarios, methods to synthesize blameless controllers should have low computational cost.

**Related work.** Lexicographic optimization, in which a finite number of ordered objective functions are sequentially optimized, such that low priority objectives do not interfere with higher priority objectives [5], is one method to ensure prioritization amongst constraints. It has been used to address applications of prioritized safety [6], [7], as well as to relax constraints in decreasing order of priority under infeasibility [8], [9]. However, the primary disadvantage of lexicographic methods is that even for convex problems, the problem must be solved iteratively, leading to a high computational burden. Typically lexicographic optimization requires solving as many optimization problems as there are prioritized objectives [10].

Related work in reachability has examined conditions under which a controller exists, that ensures feasibility with respect to a known set of constraints [11], [12]. However, these approaches typically presume that all constraints are of the same priority, or rely upon a pre-determined trade-off between safety and performance [13]. Reachability and positive invariance have been integrated into path planning approaches that prioritize safety [14]–[19], but are not readily extendable to multiple priorities and objectives.

**Contributions**. We provide a formal definition of blamelessness and formulate the problem of designing blameless controllers within an optimal control framework, amenable to a wide variety of application domains. We show that for general objective functions and constraints, it is not possible to solve for optimality and blamelessness in a single optimization problem, i.e., that no single continuous objective function can produce a blamelessly optimal action. We further show that it is possible to solve for optimality and blamelessness in exactly two optimization problems; the first determines the highest priority set of constraints that can be solved and the second finds the optimal control actions under these constraints. We demonstrate our algorithm on a real-time rocket landing problem, whose safety requirements are jointly infeasible due to fuel limitations.

*Notation:* We use the shorthand $x_{0:N}$ to denote the sequence of variables $x_0, \ldots, x_N$, for $x_i \in \mathbb{R}^n, i \in \{1, \cdots, N\}$. A sequence of constrained control inputs is denoted as $u_{0:N-1} = u_0, \ldots, u_{N-1}, u_k \in \mathcal{U} \subseteq \mathbb{R}^\ell$, with $u_{0:N-1} \in \mathcal{U}^N$. The notation $[N]$ is used to denote $\{1, \ldots, N\}$. For the set $\mathcal{X}$, the complement is denoted $\mathcal{X}^c$, so that $x \in \mathcal{X}^c$ implies $x \notin \mathcal{X}$.

## II. BLAMELESS OPTIMAL CONTROL

This section introduces the concept of a blamelessly optimal control sequence given user-prioritized constraints. The state trajectory $x_{0:N}$ is given by the discrete-time dynamics,

$$x_{k+1} := f(x_k, u_k) \in \mathbb{R}^n, \forall k \in [N-1], \quad (1)$$

under control sequence $u_{0:N-1} \in \mathbb{R}^{N\ell}$ and initial state $x_0 \in \mathcal{X}_0 \subseteq \mathbb{R}^n$. We assume that the dynamics (1) are subject to a set of *prioritized constraints* that are ranked by importance.

**Assumption 1** (Prioritized safety sets). *The user-defined compact sets $\{\mathcal{Z}_i\}_{1 \leq i \leq m} = \mathcal{Z}_1, \ldots, \mathcal{Z}_m \subseteq \mathcal{R}^p$, with $p \leq n$, are prioritized state constraints on a subset of the state $x_k$, such that $\mathcal{Z}_1$ has the highest priority. For ease of notation, it is assumed that $p = n$, meaning that the full state at a given time is constrained by the safety sets.*

The ordering of the sets $\mathcal{Z}_i$ is user-defined and dictates the prioritization of the constraints to be satisfied in the event that the system dynamics and control constraints cause the problem subject to a subset of the safety constraints to be infeasible. Also note that, without loss of generality, we presume that these constraint sets are terminal constraints, i.e., applicable to only the state at the last time instant, $x_N$. We recast the prioritized safety sets as nested constraint sets.

**Definition 1** (Nested constraint sets). *Let the nested constraint set $\mathcal{Y}_i$ be defined by $\mathcal{Y}_i = \bigcap_{j=1}^{m+1-i} \mathcal{Z}_j$, such that $\mathcal{Y}_1 \subseteq \ldots \subseteq \mathcal{Y}_m$.*

**Example 1.** *Consider a sample-return mission in which an autonomous lander is carrying a rover that will drive from the landing site to sites of interest to collect samples, but can only drive a limited distance. A safe landing area is defined near two points of interest, denoted by A and B. If feasible, the lander should select a landing site from which the rover can collect material from both sites of interest, but should avoid landing too close to the sites and risking contaminating the material. This example is depicted in Figure 1. The prioritized sets represent first landing in the designated landing area ($\mathcal{Z}_1$), then avoiding regions which may damage the scientifically interesting material B and A ($\mathcal{Z}_2$ and $\mathcal{Z}_3$, respectively), then landing in an area from which the interesting sites B and A can be reached by the rover ($\mathcal{Z}_4$ and $\mathcal{Z}_5$, respectively). The nested constraint sets describe landing sites that will not contaminate sites A or B and from which the rover can reach both A and B ($\mathcal{Y}_1$), sites that will not contaminate sites A or B and from which the rover can only reach site B ($\mathcal{Y}_2$), sites that will not contaminate sites A or B, but from which the rover cannot reach either site ($\mathcal{Y}_3$), and sites that avoid contaminating only site B ($\mathcal{Y}_4$), and all sites in the designated landing area ($\mathcal{Y}_5$).*

Nested constraint sets provide an intuitive understanding of blameworthiness, in which it is desirable for an autonomous system to sacrifice satisfaction of low priority constraints to ensure the satisfaction of high priority constraints. For instance, we are willing to sacrifice reaching site A if it ensures that the lander does not damage the area surrounding site B. In short, we wish to maximize the index $i$ such that $\mathcal{Z}_j$ is satisfied for all $j \leq i$, or equivalently, minimize $i$ such that $\mathcal{Y}_i$ is satisfied.

**Remark 1.** *We assume that at least one of the nested constraint sets is non-empty $\mathcal{Y}_i \neq \varnothing$. Thus, $\mathcal{Y}_j \supseteq \mathcal{Y}_i$ are non-empty for $j \in \{i, \ldots, m\}$.*

We formally define the concept of *blameworthiness* of a control sequence with respect to the nested constraint sets.

**Definition 2** (Blameworthy control sequence). *Suppose that the smallest nested constraint set that can be satisfied given the system dynamics, control constraints and initial condition is $\mathcal{Y}_{i^\star}$. A control sequence $u_{0:N-1} \in \mathbb{R}^{N\ell}$ that results in the state trajectory $x_{0:N}$ is blameworthy if $x_N \notin \mathcal{Y}_{i^\star}$.*

A control sequence is blameworthy if there exists an alternative control sequence that reaches a higher priority safety set. In Example 1, a control sequence that causes the lander to land in a region that may damage sites A or B is blameworthy if there is an alternative control sequence that would cause the lander to land in a region in which damage is unlikely to occur. If there exists no such alternative control sequence, it is blameless.
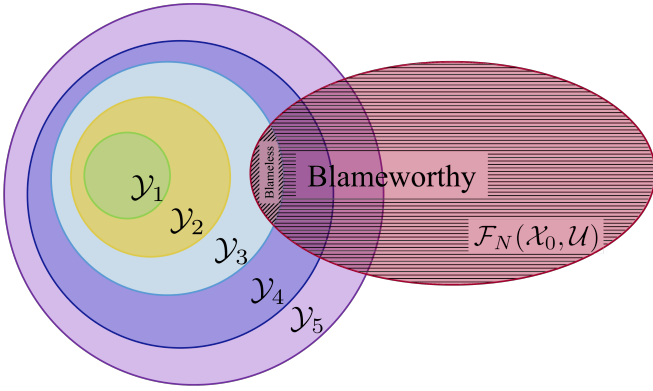
Fig. 2: Illustration of the concepts of blameworthiness and blamelessness.

**Definition 3** (Blameless control sequence). *A control sequence $u_{0:N-1} \in \mathbb{R}^{N\ell}$ is* blameless *if it is not blameworthy.*

A blameless control sequence minimizes the index $i \in [m]$ such that the resulting state sequence $x_{0:N}$ given by (1), satisfies $x_N \in \mathcal{Y}_i$.

For the set of initial states given by $\mathcal{X}_0 \subseteq \mathbb{R}^n$ and the input constraint set by $\mathcal{U} \subseteq \mathbb{R}^\ell$, we define the dynamically feasible set, defined by the dynamics (1) and the control constraints, $u_{0:N-1} \in \mathcal{U}^N$ as follows.

**Definition 4** (Dynamically feasible set). *The set of states and control sequences achievable from the initial state subject to the dynamics and control constraints is called the dynamically feasible set. It is denoted as*

$$\mathcal{F}(\mathcal{X}_0, \mathcal{U}) = \{(x_{0:N}, u_{0:N-1}) \mid x_{k+1} = f(x_k, u_k),$$
$$u_k \in \mathcal{U}, x_0 \in \mathcal{X}_0, \forall k \in [N-1]\} \subseteq \mathbb{R}^{(N+1)n+N\ell}. \quad (2)$$

The concepts of blameworthiness and blamelessness are depicted in Figure 2. The set $\mathcal{F}_N(\mathcal{X}_0, \mathcal{U}) = \{x_N | x_N \in f$ for some $f \in \mathcal{F}(\mathcal{X}_0, \mathcal{U})\}$ is the set of terminal states in the dynamically feasible set. A controller that results in a terminal state that lies in the intersection of $\mathcal{F}_N(\mathcal{X}_0, \mathcal{U})$ and $\mathcal{Y}_3$ is blameless since $\mathcal{Y}_3$ is the highest priority set that is dynamically feasible. A control sequence that results in a terminal state that is not in $\mathcal{Y}_3$ is blameworthy since a solution that results in the terminal state being in a higher priority set exists.

We assume the user-defined continuous objective, referred to as the *mission objective*, is

$$q(x_{0:N}, u_{0:N-1}) : \mathbb{R}^{Nn} \times \mathbb{R}^{N\ell} \mapsto \mathbb{R}, \quad (3)$$

where $q$ evaluates the cost of each state and control sequence. For a given objective $q$, we can define a blamelessly optimal control sequence as follows.

**Definition 5** (Blameless optimality). *Consider a state and control sequence $(x_{0:N}, u_{0:N-1}) \in \mathcal{F}(\mathcal{X}_0, \mathcal{U})$, where $\mathcal{F}(\mathcal{X}_0, \mathcal{U})$ is given by (2). The control sequence $u_{0:N-1}$ is* blamelessly optimal *if*

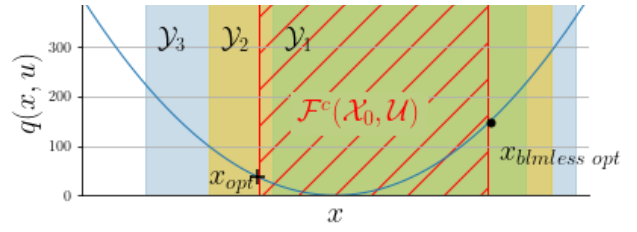*1) It is blameless according to Definition 3, and*



Fig. 3: Visualization of the need for blameless optimality. Point $x_{opt}$ is optimal with respect to the blue objective subject to the dynamics constraints but is in $\mathcal{Y}_2$. Point $x_{blmless\ opt}$ achieves a higher cost than $x_{opt}$, but is blamelessly optimal since is in $\mathcal{Y}_1$.

*2) For all $(\hat{x}_{0:N}, \hat{u}_{0:N-1}) \in \mathcal{F}(\mathcal{X}_0, \mathcal{U})$ where $\hat{u}_{0:N-1}$ is blameless,*

$$q(x_{0:N}, u_{0:N-1}) \leq q(\hat{x}_{0:N}, \hat{u}_{0:N-1}). \quad (4)$$

**Problem 1.** *Given nested constraint sets $\{\mathcal{Y}_i\}_{1 \leq i \leq m}$ (Definition 1), and initial state $x_0 \in \mathcal{X}_0$, find a blamelessly optimal control sequence $u_{0:N-1} \in \mathcal{U}^N$.*

Problem 1 entails finding a control sequence that leads to the satisfaction of the largest number of safety sets possible, while taking into account the prioritization of the sets and optimality with respect to objective $q$ in (3).

**Remark 2.** *Critically, we assume that not all nested constraint sets $\mathcal{Y}_i$ are feasible. Therefore, Problem 1 is equivalent to finding 1) the smallest index $i^\star$ such that $x_N \in \mathcal{Y}_{i^\star}$ is feasible, and 2) the control sequence $u_{0:N-1}$ that is optimal under the constraint $x_N \in \mathcal{Y}_{i^\star}$.*

## III. SOLVING FOR BLAMELESSLY OPTIMAL CONTROL SEQUENCES

Blamelessness and optimality may be competing objectives when the optimal solution with respect to objective $q$ lies outside of the highest priority safety set, as is the case in Figure 3. In this section, we discuss various solution methods for finding blamelessly optimal control sequences.

### A. Need for Blameless Optimality

We illustrate the need for blameless optimality in Figure 3 for a one-dimensional state space and a one-dimensional control space. The objective $q$ given by (3) is a quadratic function of the state, shown by the blue curve. The nested safety sets are by the shaded regions: the green, yellow, and blue regions correspond to $\mathcal{Y}_1$, $\mathcal{Y}_2$, and $\mathcal{Y}_3$, respectively. The red hashed region shows states that are infeasible under $\mathcal{F}(\mathcal{X}_0, \mathcal{U})$ given by (2). Although $x_{\text{blmless-opt}}$ has a higher cost than $x_{\text{opt}}$, it is preferable to $x_{\text{opt}}$ because it satisfies a higher priority nested constraint set, $\mathcal{Y}_1$.

### B. Related Methods

The concept of priority within optimization problems is not novel. Some related concepts the exist in the literature are discussed in the following sections.

*1) Connection with Lexicographic Optimization:* The lexicographic optimization problem

$$\min_{w \in \mathcal{W}} \ (q_1(w), \ldots, q_m(w)),$$

in which $w \in \mathbb{R}^\eta$ is the solution variable, $\mathcal{W}$ is the set of feasible solutions, and $(\cdot, \ldots, \cdot)$ denotes an ordering, is typically addressed by iteratively solving

$$\text{for } j = 1, \ldots, m :$$
$$\min_{w \in \mathcal{W}} \ q_j(w) \quad \text{s.t. } q_i \le q_i^\star, \quad i = 1, \ldots, j-1,$$

where $q_i^\star$ is the optimal value of the $i^{th}$ problem. In lexicographic optimization, multiple ordered objectives are optimized. This is related to Problem 1, in that a notion of priority exists within an optimization problem.

It is possible to construct an algorithm that uses an iterative approach similar to that of lexicographic optimization to solve for a blamelessly optimal control sequence. This algorithm, shown below, iteratively imposes constraints in order of priority.

---

**Algorithm 1** Brute-Force Blameless Control

---

Set $i = 0$
**while** not feasible **do**
    Loosen constraints $i \leftarrow i + 1$
    Solve
$$\min_{u_{0:N-1}} \ q(x_{0:N}, u_{0:N-1}) \tag{5a}$$
$$\text{s.t. } x_{k+1} = f(x_k, u_k), \ x_0 \in \mathcal{X}_0, \tag{5b}$$
$$u_k \in \mathcal{U}, \ x_N \in \mathcal{Y}_i \tag{5c}$$

**end while**

---

**Proposition 1.** *Under Assumption 1, Algorithm 1 produces a control sequence $u_{0:N-1} \in \mathcal{U}^N$ that is blamelessly optimal (Definition 5).*

*Proof.* Suppose the control sequence found using Algorithm 1 was blameworthy for constraint $\mathcal{Y}_j$. Then, by definition there exists $\hat{u}_{0:N-1} \in \mathcal{U}^N$ such that $\hat{x}_N \in \mathcal{Y}_j$. Thus, problem (5) is feasible and therefore $x_N \in \mathcal{Y}_j$. $\square$

While Algorithm 1 produces a blamelessly optimal control sequence, it is not an ideal solution because of the excessive cost that comes from solving as many optimization problems as there are prioritized constraints. For a scenario with $m$ prioritized safety constraints, Algorithm 1 solves $m-1$ infeasible optimization problems and 1 feasible optimization problem in the worst case. We are interested in finding a more computationally efficient solution to Problem 1, that is amenable to real-time application in safety-critical systems.

*2) Connection with Reachability Analysis:* We can show that blamelessness can equivalently be defined using the successor sets used in reachability analysis [20]. A successor set is defined as follows.

**Definition 6** (Successor Set). *[21, Def.10.3] Consider the set of initial conditions $\mathcal{X}_0 \subseteq \mathbb{R}^n$ and set of inputs $\mathcal{U}$. The N-step successor set under the input constraints $u_{0:N-1} \in \mathcal{U}^N$*

and dynamics (1) is given by

$$Suc(\mathcal{X}_0, \mathcal{U}^N) = \{x_1, \ldots, x_N \mid \exists u_k \in \mathcal{U}, x_0 \in \mathcal{X}_0$$
$$x_{k+1} = f(x_k, u_k), \forall k \in [N]\}. \tag{6}$$

We use the shorthand $Suc(\mathcal{X}_0) = Suc(\mathcal{X}_0, \mathcal{U}^N)$, and denote $n^{th}$ state in $Suc(\mathcal{X}_0)$ as $Suc_n(\mathcal{X}_0)$. For a given input sequence $u_{0:N-1} \in \mathcal{U}^N$, we use the shorthand $Suc(\mathcal{X}_0, u_{0:N-1})$. The following proposition formulates blamelessness using successor sets.

**Proposition 2.** *The control sequence $u_{0:N-1} \in \mathcal{U}^N$ is blameless if $\mathcal{Y}_i \cap Suc_N(\mathcal{X}_0) \ne \varnothing$ implies $Suc_N(\mathcal{X}_0, u_{0:N-1}) \in \mathcal{Y}_i$ for all $i = 1, \ldots, m$.*

*Proof.* If $\mathcal{Y}_i \cap Suc_N(\mathcal{X}_0) \ne \varnothing$, then there is a control sequence $u_{0:N-1}$ such that $x_N \in \mathcal{Y}_i$. Then, $u_{0:N-1}$ is blameless if $Suc_N(\mathcal{X}_0, u_{0:N-1}) \in \mathcal{Y}_i$. $\square$

*C. Blamelessness and Optimality Relationship*

In this section, we show that in general, finding blamelessly optimal control sequences requires the solution of two optimization. We show this by contradiction. We use this result to formulate a framework for solving for blamelessly optimal control sequences that moves the computational complexity involved in the brute force method, (5), offline.

Consider the problem of designing an objective function that produces control sequences that are simultaneously blameless and optimal with respect to objective $q$. We introduce the notation $\hat{q}$ to represent a continuous objective that, when minimized, results in a control sequence that is blameless.

**Problem 2.** *Find a continuous objective function $\hat{q} : \mathbb{R}^{Nn} \times \mathbb{R}^{N\ell} \mapsto \mathbb{R}$ for the optimization problem*

$$\min_{u_{0:N-1} \in \mathbb{R}^{N\ell}} \ \hat{q}(x_{0:N}, u_{0:N-1}) \tag{7a}$$
$$\text{s.t. } (x_{0:N}, u_{0:N-1}) \in \mathcal{F}(\mathcal{X}_0, \mathcal{U}), \tag{7b}$$

*whose solution $u_{0:N-1}^\star$ is blameless, and optimal with respect to the user-defined objective q.*

We define the following set to facilitate the solution to Problem 2. Consider all the dynamically feasible state and control sequences, $(x_{0:N}, u_{0:N-1}) \in \mathcal{F}(\mathcal{X}_0, \mathcal{U})$, that have a cost $\hat{q}(x_{0:N}, u_{0:N-1}) \le \alpha_i$ for $i = 1, \ldots, m$. We denote the set of terminal states attained by these state and control sequences as

$$\mathcal{H}_i(\hat{q}) = \{x_N \mid (x_{0:N}, u_{0:N-1}) \in \mathcal{F}(\mathcal{X}_0, \mathcal{U}),$$
$$\hat{q}(x_{0:N}, u_{0:N-1}) \le \alpha_i\}. \tag{8}$$

The set $\mathcal{H}_i(\hat{q})$ is the set of terminal conditions in the $\alpha_i$ sublevel set of $\hat{q}$.

The following theorem presents the necessary and sufficient conditions for designing an objective $\hat{q}$ that produces blameless control sequences.

**Theorem 1.** *Under the nested constraint sets $\{\mathcal{Y}_i\}_{1 \le i \le m}$ defined in Definition 1, the optimal control problem (7)*

*produces blameless control sequences if and only if the continuous objective $\hat{q}$ satisfies*

$$\mathcal{H}_i(\hat{q}) = \mathcal{Y}_i \cap Suc_N(\mathcal{X}_0), \ \forall\, 1 \le i \le m. \tag{9}$$

*Proof.* Note that $\mathcal{H}_i(\hat{q})$ are compact sets since the dynamics (1) and the objective function $\hat{q}(x_{0:N}, u_{0:N-1})$ are continuous functions and $\mathcal{U}$ is a compact set.

First, we prove by contradiction that if $\hat{q}$ satisfies (9), then (7) produces blameless control sequences. Suppose $\hat{q}$ satisfies (9), but the result of (7) is blameworthy. This means if the solution to $\hat{q}$ is $(x^\star_{0:N}, u^\star_{0:N-1}) \in \mathcal{F}(\mathcal{X}_0, \mathcal{U})$ with $x^\star_N \notin \mathcal{Y}_i$, there exists another state and control sequence pair, $(x^\dagger_{0:N}, u^\dagger_{0:N-1}) \in \mathcal{F}(\mathcal{X}_0, \mathcal{U})$ with $x^\dagger_N \in \mathcal{Y}_i$. Then, by (9), we have $\hat{q}(x^\dagger_{0:N}, u^\dagger_{0:N-1}) \le \alpha_i < \hat{q}(x^\star_{0:N}, u^\star_{0:N-1})$, which contradicts the definition of an optimal solution of (7).

We will prove the reverse implication directly. Suppose (7) produces blameless control sequences. Then, by construction,

$$\max_{u_{0:N-1}} \ \hat{q}(x_{0:N}, u_{0:N-1}) \le \alpha_i = \inf_{u_{0:N-1}} \ \hat{q}(x_{0:N}, u_{0:N-1})$$
$$\text{s.t. } \mathcal{F}(\mathcal{X}_0, \mathcal{U}) \qquad\qquad \text{s.t. } \mathcal{F}(\mathcal{X}_0, \mathcal{U})$$
$$x_N \in \mathcal{Y}_i \qquad\qquad\qquad x_N \notin \mathcal{Y}_i$$

since, otherwise, there exists a solution to (7), $(x^\star_{0:N}, u^\star_{0:N-1}) \in \mathcal{F}(\mathcal{X}_0, \mathcal{U})$ with $x^\star_N \notin \mathcal{Y}_i$, for which $\hat{q}(x^\star_{0:N}, u^\star_{0:N-1}) \le \hat{q}(x^\dagger_{0:N}, u^\dagger_{0:N-1})$ for some $(x^\dagger_{0:N}, u^\dagger_{0:N-1}) \in \mathcal{F}(\mathcal{X}_0, \mathcal{U})$ with $x^\dagger_N \in \mathcal{Y}_i$. That is, the control sequence $u^\star_{0:N-1}$ found by solving (7) is blameworthy, which results in a contradiction. Thus, if $\hat{q}(x_{0:N}, u_{0:N-1}) \le \alpha_i$ then $x_N \in \mathcal{Y}_i$, and we have $x_N \in \mathcal{H}_i(\hat{q})$ implies $x_N \in \mathcal{Y}_i$, i.e. $\mathcal{H}_i(\hat{q}) \subseteq \mathcal{Y}_i \cap \mathrm{Suc}_N(\mathcal{X}_0)$. Conversely, if $z \in \mathcal{Y}_i \cap \mathrm{Suc}_N(\mathcal{X}_0)$ then there exists $(x_{0:N}, u_{0:N-1}) \in \mathcal{F}(\mathcal{X}_0, \mathcal{U})$ with $x_N = z$. By definition of $\alpha_i$, we have $\hat{q}(x_{0:N}, u_{0:N-1}) \le \alpha_i$. Thus, $z \in \mathcal{H}_i(\hat{q})$ i.e. $\mathcal{Y}_i \cap \mathrm{Suc}_N(\mathcal{X}_0) \subseteq \mathcal{H}_i(\hat{q})$.
$\square$

Theorem 1 provides the necessary and sufficient conditions (9) for constructing an objective function $\hat{q}$ for which (7) produces blameless control sequences. Next, we consider whether it is possible to construct the objective function $\hat{q}$ such that it produces control sequences that are both blameless and optimal with respect to the original user-defined objective function $q$. The following corollary shows that there are continuous objectives $q$ for which all continuous objectives that produce blameless control sequences, $\hat{q}$ produce sub-optimal control sequences. That is, there does not exist an objective satisfying (9) that produces control sequences that are optimal with respect to $q$.

**Corollary 1.** *There exists a continuous objective $q$ such that there is no objective $\hat{q}$ that produces blameless control sequences that are also optimal with respect to $q$.*

*Proof.* We will prove by construction that there exists an objective $q$ whose optimal solution does not correspond with the optimal of any objective $\hat{q}$ satisfying (9).

Consider a continuous objective $\hat{q}(x_{0:N}, u_{0:N-1})$ that satisfies (9). We will show $\hat{q}(x_{0:N}, u_{0:N-1}) = \alpha_i$ for all $x_N$ on the boundary of $\mathcal{Y}_i$, denoted $\delta\mathcal{Y}_i$. If $\hat{q}(x_{0:N}, u_{0:N-1}) > \alpha_i$ for $x_N \in \delta\mathcal{Y}_i$, then by continuity of $\hat{q}$, we have $H_i \not\subseteq \mathcal{Y}_i$, and likewise if $\hat{q}(x_{0:N}, u_{0:N-1}) < \alpha_i$ for $x_N \in \delta\mathcal{Y}_i$ then by continuity, $\mathcal{Y}_i \not\subseteq H_i$. Thus, by contradiction with (8), we have $\hat{q}(x_{0:N}, u_{0:N-1}) = \alpha_i$ for all $x_N \in \delta\mathcal{Y}_i$. Furthermore, by (8), for any $(x^\dagger_{0:N}, u^\dagger_{0:N-1}) \in \mathcal{F}(\mathcal{X}_0, \mathcal{U})$ with $x^\dagger_N$ in the interior of $\mathcal{Y}_i$ and $(x^\ddagger_{0:N}, u^\ddagger_{0:N-1}) \in \mathcal{F}(\mathcal{X}_0, \mathcal{U})$ with $x^\ddagger_N \in \delta\mathcal{Y}_i$, we have $\hat{q}(x^\dagger_{0:N}, u^\dagger_{0:N-1}) \le \hat{q}(x^\ddagger_{0:N}, u^\ddagger_{0:N-1})$.

Next, consider the objective $q(x_{0:N}, u_{0:N-1}) = \|x_N - z\|$ where $z \in \delta\mathcal{Y}_i$ is some point on the boundary of $\mathcal{Y}_i$ with a reachable neighborhood $N(z) = \{y \in \mathcal{Y}_i : \|y - z\| \le \epsilon\}$ for some $\epsilon$. By construction, this objective is continuous and minimized by the feasible solution $x^\star_N = z$.

By construction, we have $q(x_{0:N}, u_{0:N-1}) > q(x^\star_{0:N}, u^\star_{0:N-1})$ for any feasible sequence $x_{0:N}$ with terminal condition $x_N \ne z \in N(z)$. In contrast, $\hat{q}(x_{0:N}, u_{0:N-1}) \le \hat{q}(x^\star_{0:N}, u^\star_{0:N-1}) = \alpha_i$ for any terminal condition $x_N \in N(z)$. Thus, there exists $x_N \in N(z)$ that is optimal with respect to $\hat{q}$, $\hat{q}(x_{0:N}, u_{0:N-1}) \le \hat{q}(x^\star_{0:N}, u^\star_{0:N-1})$, but is sub-optimal with respect to $q$, i.e. $q(x_{0:N}, u_{0:N-1}) > q(x^\star_{0:N}, u^\star_{0:N-1})$.
$\square$

As a consequence of Corollary 1, it is possible for the user to define a continuous mission objective $q$ for which it is impossible to formulate a single continuous optimal control problem that produces control sequences that are both blameless and optimal. In other words, Problem 2 does not necessarily have a solution.

It follows that a blamelessly optimal control sequence must be found by solving at least two sub-problems. We propose an algorithm for finding blamelessly optimal control sequences using exactly two sub-problems, rather than up to $m$ problems like in Algorithm 1. The proposed algorithm first finds the highest priority set for which a feasible solution exists. Then, the optimal control sequence with respect to the mission objective subject to the highest priority constraint is found. The algorithm is presented in Algorithm 2. For further details on how $i^\star$ is found in Algorithm 2, see [22].

---

**Algorithm 2** Two-Stage Blameless Control

---

Minimize $i^\star$ such that $\mathcal{Y}_{i^\star} \cap \mathrm{Suc}_N(\mathcal{X}_0) \ne \varnothing$
Solve
$$\min_{u_{0:N-1}} \ q(x_{0:N}, u_{0:N-1}) \tag{10a}$$
$$\text{s.t. } (x_{0:N}, u_{0:N-1}) \in \mathcal{F}(\mathcal{X}_0, \mathcal{U}), \tag{10b}$$
$$x_N \in \mathcal{Y}_{i^\star} \tag{10c}$$

---

**Proposition 3.** *The solution to Algorithm 2 solves Problem 1.*

*Proof.* We will prove that the solution to Algorithm 2 is blameless directly. Assume the solution to Algorithm 2 is $u^\star_{0:N-1}$. Then, the resulting state sequence $x^\star_{0:N}$ has $x_N \in \mathcal{Y}_{i^\star}$, so that $\mathcal{Y}_{i^\star} \cap \mathrm{Suc}_N(\mathcal{X}_0) \ne \varnothing$. It follows directly from Proposition 2 that the control sequence is blameless.

We will prove by contradiction that the solution to Algorithm 2 is blamelessly optimal. Assume the solu-

tion to Algorithm 2 is $(x^\star_{0:N}, u^\star_{0:N-1})$, but there exists a blameless solution $(x^\dagger_{0:N}, u^\dagger_{0:N-1})$ with $q(x^\dagger_{0:N}, u^\dagger_{0:N-1}) < q(x^\star_{0:N}, u^\star_{0:N-1})$. Then, $(x^\star_{0:N}, u^\star_{0:N-1})$ is sub-optimal with respect to Problem (10), contradicting the assumption that $(x^\star_{0:N}, u^\star_{0:N-1})$ is the solution to Algorithm 2. Thus, the optimal solution to Algorithm 2 is blamelessly optimal. $\square$

Algorithm 2 takes advantage of the fact that the prioritized sets are nested to find blamelessly optimal control sequences. Specifically, since high priority sets are contained in low priority sets, the problem of imposing priority within an optimal control algorithm can be posed as two decoupled problems: a set inclusion problem that finds the highest priority set that has a nonempty intersection with the dynamically feasible set, and an optimal control problem that finds the optimal control sequence that is in the set found in the first problem. The result is an algorithm that requires solving two optimization problems, rather than the $m$ required by lexicographic optimization. This becomes computational beneficial when the number of prioritized sets, $m$, is large.

## IV. LANDING UNDER PRIORITIZED SETS

We consider the problem of an autonomous lander selecting a landing site, subject to control limits and limited power. Prioritized sets are defined to dictate the most desirable landing sites. This problem is illustrated in Figure 1 in three-dimensional space. For ease of presentation, the results consider the problem restriction to two-dimensional space, so that the landing sites are restricted to a line.

*Lander Dynamics.:* We model the lander as a linear system. The state consists of the velocity and position of the lander, $x = \begin{bmatrix} \dot{r}^x & \dot{r}^y & r^x & r^y \end{bmatrix}^\mathsf{T}$. The control input is acceleration $u = \begin{bmatrix} a^x & a^y \end{bmatrix}^\mathsf{T}$. The affine continuous-time dynamics are $\dot{x} = Ax + Bu + Cg$ where

$$A = \begin{bmatrix} \mathbf{0}_{2\times2} & \mathbf{0}_{2\times2} \\ I_{2\times2} & \mathbf{0}_{2\times2} \end{bmatrix}, \quad B = \begin{bmatrix} I_{2\times2} \\ \mathbf{0}_{2\times2} \end{bmatrix}, \quad C = \begin{bmatrix} 0 \\ 1 \\ \mathbf{0}_{2\times1} \end{bmatrix},$$

and $g = 9.81\frac{\text{m}}{\text{s}^2}$ is the acceleration due to gravity. The control is subject to the box constraints

$$a^x \in \begin{bmatrix} -10, 10 \end{bmatrix}\frac{\text{m}}{\text{s}^2} \text{ and } a^y \in \begin{bmatrix} 9, 30 \end{bmatrix}\frac{\text{m}}{\text{s}^2}.$$

The lander has initial condition

$$x_0 = \begin{bmatrix} -10\frac{\text{m}}{\text{s}} & -5\frac{\text{m}}{\text{s}} & -130\text{m} & 100\text{m} \end{bmatrix}^\mathsf{T},$$

and sufficient power to last time $T = 12$ seconds.

*Prioritized constraints.:* Five prioritized safety constraints are imposed on the states $\dot{r}^x_N$ and $r^x_N$, jointly denoted $x^x_N = \begin{bmatrix} \dot{r}^x_N & r^x_N \end{bmatrix}$. Namely, the position at which the lander lands and the velocity in the direction parallel to the ground at the time of landing are constrained by the prioritized safety constraints. The resulting nested constraint sets are shown in
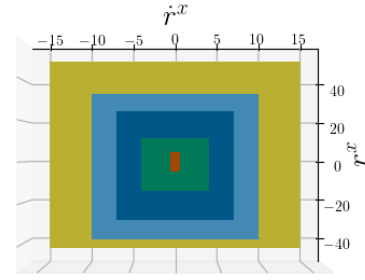


Fig. 4: Sets $\mathcal{Y}_1, \ldots, \mathcal{Y}_5$.

Figure 4 and defined as follows

$$\mathcal{Y}_1 = \{(\dot{r}^x_N, r^x_N) \mid \dot{r}^x_N \in [-0.5, 0.5]\,\frac{\text{m}}{\text{s}}, r^x_N \in [-5, 5]\,\text{m}\},$$

$$\mathcal{Y}_2 = \{(\dot{r}^x_N, r^x_N) \mid \dot{r}^x_N \in [-4, 4]\,\frac{\text{m}}{\text{s}}, r^x_N \in [-15, 12]\,\text{m}\},$$

$$\mathcal{Y}_3 = \{(\dot{r}^x_N, r^x_N) \mid \dot{r}^x_N \in [-7, 7]\,\frac{\text{m}}{\text{s}}, r^x_N \in [-30, 26]\,\text{m}\},$$

$$\mathcal{Y}_4 = \{(\dot{r}^x_N, r^x_N) \mid \dot{r}^x_N \in [-10, 10]\,\frac{\text{m}}{\text{s}}, r^x_N \in [-40, 35]\,\text{m}\},$$

$$\mathcal{Y}_5 = \{(\dot{r}^x_N, r^x_N) \mid \dot{r}^x_N \in [-15, 15]\,\frac{\text{m}}{\text{s}}, r^x_N \in [-45, 52]\,\text{m}\}.$$

*Objective Function.:* The objective function (3) is the quadratic objective $q(x_{0:N}, u_{0:N-1}) = \sum_{k=1}^N q_k(x_k, u_{k-1})$, with

$$q_k(x_k, u_{k-1}) = u_{k-1}^\mathsf{T} R u_{k-1} + (x^x_N - c_i)^\mathsf{T} Q(x^x_N - c_i), \quad (11)$$

where $R \in \mathbb{R}^{2\times2}$ is the input cost matrix, $Q \in \mathbb{R}^{2\times2}$ is the regulator cost matrix, and $c_i \in \mathbb{R}^2$ is the center of the safety set $\mathcal{Y}_i$. The weights, $Q$ and $R$ are tuned by the user.

Blamelessly optimal control sequences are found by solving Algorithm 2. First, an objective function is generated that satisfies the necessary and sufficient conditions for an objective function that produces blameless, but not necessarily optimal, control sequences, according to Theorem 1. This objective is used to determine the smallest index $i^\star$ such that $x^x_N \in \mathcal{Y}_{i^\star}$. Then, the control sequence $u^\star_{0:N-1}$ that is optimal with respect to (11) subject to dynamics (2) and terminal state constraint $x^x_N \in \mathcal{Y}_{i^\star}$ is found.

The results of Algorithm 2 are compared to the results of Algorithm 1, and an optimal control algorithm that is optimal with respect to the quadratic objective $q(x_{0:N}, u_{0:N-1}) = \sum_{k=1}^N q_k(x_k, u_{k-1})$, with

$$q_k(x_k, u_{k-1}) = u_{k-1}^\mathsf{T} R u_{k-1} + (x^x_N - c_1)^\mathsf{T} Q(x^x_N - c_1), \quad (12)$$

where $c_1$ is the center of the highest priority safety set. This algorithm does not guarantee that the resulting control sequences are blameless.

The weighting matrices are $R = \mathbf{diag}(0.5, 1)^2\frac{1}{\text{m}^2}$ and $Q = 5^2 I \frac{\text{s}^2}{\text{m}^2}$ for the blamelessly optimal and brute force algorithms. The optimal algorithm uses $R = \mathbf{diag}(0.5, 1)^2\frac{1}{\text{m}^2}$ and is tested with two values of the input weighting matrix, $Q = 5^2 I \frac{\text{s}^2}{\text{m}^2}$ and $Q = 0.15^2 I \frac{\text{s}^2}{\text{m}^2}$.
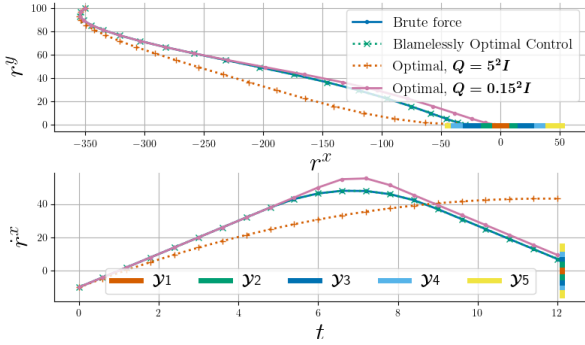
Fig. 5: Comparison of blameless optimality, brute force and optimal control algorithm for 12 second trajectory.

*Results:* Figure 5 shows the trajectories and velocity in the $x$-direction generated using the brute force algorithm (Algorithm 1), the blamelessly optimal control algorithm (Algorithm 2), and the algorithm optimal with respect to (12) with two weights. The brute force and blamelessly optimal trajectories are identical, and both result in solutions that have terminal states in the set $\mathcal{Y}_3$. The algorithm optimal with respect to (12) with $Q = 5^2 I \frac{\text{s}^2}{\text{m}^2}$ results in a terminal position in $\mathcal{Y}_4$ and a terminal velocity outside of the safety sets, and is therefore ultimately outside of the safety sets. The trajectory resulting from the same problem with input cost matrix $Q = 0.15^2 I \frac{\text{s}^2}{\text{m}^2}$ has a terminal position in $\mathcal{Y}_2$ and a terminal velocity in $\mathcal{Y}_4$, and therefore ultimately a terminal state in $\mathcal{Y}_4$.

## A. Implications of Blameless Optimal Control

The brute force algorithm (Algorithm 1), and the blamelessly optimal algorithm (Algorithm 2) result in equivalent solutions. However, the brute force algorithm requires solving up to $m$ optimization problems given $m$ safety constraints. The blameless optimization problem requires solving two optimization problems, regardless of the number of defined prioritized safety constraints. The blamelessly optimal algorithm is therefore increasingly beneficial when several safety constraints are defined.

In safety critical applications, guarantees on the blamelessness of a control sequence are required to enable trust of autonomous systems. The algorithm optimal with respect to objective (12) requires parameter tuning to find the safest possible solution. Moreover, the difficulty of tuning the input weighting matrix and regulator weighting matrix increases with the number of states being constrained. When problem parameters such as initial conditions and problem horizon are uncertain, tuning the weighting matrices is impractical and could lead to blameworthy control sequences.

## V. CONCLUSIONS

This work develops the concept of a blamelessly optimal control sequence. We show that there is a tradeoff between optimality and blamelessness, which motivates the need for blameless optimality. An algorithm is presented to solve for blamelessly optimal control sequences and results are pre-

sented on a rocket landing problem. Future work will expand the idea of blameless optimality to stochastic systems.

REFERENCES

[1] P. Wu, B. W. Israelsen, K. Srivastava, S. Wu, and R. Grabowski, "A tiered approach for ethical AI evaluation metrics," in *Assoc. Adv. Artif. Intell. (AAAI) Spring Symp. Ser.*, March 2022.
[2] S. Chancellor, "Toward practices for human-centered machine learning," *Commun. ACM*, vol. 66, no. 3, p. 78–85, feb 2023.
[3] J. Halpern and M. Kleiman-Weiner, "Towards formal definitions of blameworthiness, intention, and moral responsibility," *Proc. AAAI Conf. Artif. Intell.*, vol. 32, no. 1, Apr. 2018.
[4] A. Weiss, M. Baldwin, R. S. Erwin, and I. Kolmanovsky, "Model predictive control for spacecraft rendezvous and docking: Strategies for handling constraints and case studies," *IEEE Trans. Control Syst. Technol.*, vol. 23, no. 4, pp. 1638–1647, 2015.
[5] H. Isermann, "Linear lexicographic optimization," in *Operations-Research-Spektrum*, vol. 4. Springer, 12 1982, pp. 223–228.
[6] K. Lesser and A. Abate, "Multiobjective optimal control with safety as a priority," *IEEE Trans. Control Syst. Technol.*, vol. 26, no. 3, pp. 1015–1027, 2018.
[7] D. Dueri, F. Leve, and B. Açıkmeşe, "Minimum error dissipative power reduction control allocation via lexicographic convex optimization for momentum control systems," *IEEE Trans. Control Syst. Technol.*, vol. 24, no. 2, pp. 678–686, 2016.
[8] J. Vada, O. Slupphaug, T. A. Johansen, and B. A. Foss, "Linear MPC with optimal prioritized infeasibility handling: application, computational issues and stability," *Automatica*, vol. 37, no. 11, pp. 1835–1843, 2001.
[9] T. Miksch and A. Gambier, "Fault-tolerant control by using lexicographic multi-objective optimization," in *8th Asian Control Conf. (ASCC)*, 2011, pp. 1078–1083.
[10] J. Marques-Silva, J. Argelich, A. Graça, and I. Lynce, "Boolean lexicographic optimization: algorithms & applications," in *Ann. Math. Artif. Intell.*, vol. 62. Springer, 05 2011, pp. 317–343.
[11] C. Tomlin, J. Lygeros, and S. Shankar Sastry, "A game theoretic approach to controller design for hybrid systems," *Proc. IEEE*, vol. 88, no. 7, pp. 949–970, 2000.
[12] S. Bansal, M. Chen, S. Herbert, and C. J. Tomlin, "Hamilton-Jacobi reachability: A brief overview and recent advances," in *IEEE 56th Annu. Conf. Decis. Control (CDC)*, 2017, pp. 2242–2253.
[13] A. P. Vinod and M. M. Oishi, "Optimal trade-off analysis for efficiency and safety in the spacecraft rendezvous and docking problem," *IFAC Workshop Netw. Auton. Air Space Syst.*, vol. 51, no. 12, pp. 136–141, 2018.
[14] J. A. Starek, E. Schmerling, G. D. Maher, B. W. Barbee, and M. Pavone, "Fast, safe, propellant-efficient spacecraft motion planning under Clohessy–Wiltshire–Hill dynamics," *J. Guid. Control Dyn.*, vol. 40, no. 2, pp. 418–438, 02 2017.
[15] K. Leung, E. Schmerling, M. Zhang, M. Chen, J. Talbot, J. C. Gerdes, and M. Pavone, "On infusing reachability-based safety assurance within planning frameworks for human–robot vehicle interactions," *Int. J. Robot. Res.*, vol. 39, no. 10-11, pp. 1326–1345, 2020.
[16] A. P. Vinod, S. Rice, Y. Mao, M. M. K. Oishi, and B. Açıkmeşe, "Stochastic motion planning using successive convexification and probabilistic occupancy functions," in *Proc. Conf. Decis. Control (CDC)*. IEEE, 2018, pp. 4425–4432.
[17] H.-T. Chiang, N. Malone, K. Lesser, M. Oishi, and L. Tapia, *Aggressive Moving Obstacle Avoidance Using a Stochastic Reachable Set Based Potential Field*. Cham: Springer International Publishing, 2015, pp. 73–89.
[18] C. Danielson, A. Weiss, K. Berntorp, and S. Di Cairano, "Path planning using positive invariant sets," *55th Conf. Decis. Control (CDC)*, pp. 5986–5991, 2016.
[19] S. Dixit, U. Montanaro, S. Fallah, M. Dianati, D. Oxtoby, T. Mizutani, and A. Mouzakitis, "Trajectory planning for autonomous high-speed overtaking using MPC with terminal set constraints," in *21st Int. Conf. Intell. Transp. Syst. (ITSC)*, 2018, pp. 1061–1068.
[20] M. Althoff, G. Frehse, and A. Girard, "Set propagation techniques for reachability analysis," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 4, no. 1, pp. 369–395, 2021.
[21] F. Borrelli, A. Bemporad, and M. Morari, *Predictive control for linear and hybrid systems*. Cambridge University Press, 2017.
[22] https://github.com/nataliapavlasek16/blameless-optimal-control.