Robotica: (2020) volume 38, pp. 2151–2172. © Cambridge University Press 2020
doi:10.1017/S0263574719001905

Robust Motion Planning in Dynamic Environments Based on Sampled-Data Hamilton–Jacobi Reachability

Sébastien Kleff†,‡ and Ning Li†,‡*

†*Department of Automation, Shanghai Jiao Tong University, Shanghai, 200240, China.*
E-mail: sebastien.kleff@eleves.ec-nantes.fr
‡*Key Laboratory of System Control and Information Processing, Ministry of Education of China, Shanghai, 200240, China.*

(Accepted 21 December 2019. First published online: February 14, 2020)

SUMMARY
We propose a novel formal approach to robust motion planning (MP) in dynamic environments based on reachability analysis. While traditional MP methods usually fail to provide formal robust safety and performance guarantees, our approach provably ensures safe task achievement in time-varying and adversarial environments under parametric uncertainty. We leverage recent results on Hamilton–Jacobi (HJ) reachability and differential games in order to compute offline guaranteed motion plans that are compatible with the sampled-data (SD) paradigm. Also, we synthesize online provably robust safety-preserving and target-reaching feedback controls. Unlike earlier applications of reachability analysis to MP, our methodology handles arbitrary time-varying constraints, adversarial agents such as pursuing obstacles or evading targets, and takes into account the robot's configuration. Furthermore, we use HJ projections in order to reduce significantly the computational burden without trading off safety guarantees. The validity of this approach is demonstrated through the case study of a robot arm subject to measurement errors, which is tasked with safely reaching a goal in a known time-varying workspace while avoiding capture by an unpredictable pursuer. Finally, the performance of the approach and research perspectives are discussed.

KEYWORDS: Robust motion planning; Reachability analysis; Sampled-data system; Reach-avoid game; Hamilton–Jacobi–Isaacs Equation.

1. Introduction

1.1. Motivation

As robots are increasingly operating among humans, providing deterministic guarantees of achieving safe tasks under uncertainty becomes crucial. A robot evolving in a cluttered workspace typically does not have perfect knowledge of itself nor of the environment because of sensing, measurement, estimation or modeling errors or due to the presence of unpredictable objects. Planning a task without taking these uncertain factors into account beforehand may result in the mission failure or even harmful accidents. Although there exist many practical and efficient algorithms for motion planning (MP) in dynamic environments,[1] most of them usually fail to provide safety and performance guarantees under uncertainty. As a matter of fact, formal verification in this context is challenging as an infinite number of evolutions must be taken into account: same initial conditions may result in different outcomes depending on the realization of uncertain factors at run-time. Thus, it is possible to perform

* Corresponding author. E-mail: ning_li@sjtu.edu.cn

the verification over sets of possible evolutions rather than checking individually each one of them by modeling uncertain factors explicitly in a worst-case fashion.[2,3] Indeed, given initial conditions and some worst possible realization of uncertainty, it is sufficient to ensure safety and performance along a worst-case evolution which delimits the set of possible evolutions starting from the same initial conditions. This idea of preparing the robot for the worst can be mathematically formalized by differential games.[4] This leads to regard the robust MP problem somehow pessimistically as a game opposing the robot trying to generate *nice* evolutions, to an adversarial agent acting maliciously – rationally and optimally against the robot's objectives.

To this end, reachability analysis is a powerful framework that can address the formal verification of safety-critical control systems. Its main feature is to define a wide range of constructs characterizing attainability and invariance properties of dynamical systems. The most basic reachability construct is the *reachable set*, which contains every state that can reach a predefined target set at a given time. There exists many shades of reachability[5] to characterize more sophisticated combinations of these basic properties. For instance, in the context of robust MP, one may be interested in characterizing the states that can reach a predefined target set (goal configurations) at a certain time while avoiding a given unsafe set at all times (forbidden configurations) regardless of the opposing action of adversarial agents (uncertain factors), that is the *robust reach-avoid set*.[6] Numerous methods have been developed to calculate efficiently reachability constructs,[7–10] and many of them however rely on restrictive assumptions (linear dynamics or constraints convexity) which are unlikely to be verified in the context of MP. A more general approach to characterize reachability constructs is to express them as victory domains of appropriately defined differential games.[11] In this way, it has been shown that a level set representation of the robust reach-avoid set can be computed as the viscosity solution to appropriate Hamilton–Jacobi–Isaacs (HJI) partial differential equation (PDE) in the case of static constraints[12] and time-varying constraints.[13–15]

Besides, most of the real-world cyber-physical systems are not realistically modeled as purely continuous-time (CT) processes. Instead, they consist of a plant evolving continuously over time while being periodically controlled and observed by a digital controller, which is represented by the sampled-data (SD) model. This loss of knowledge and control authority must naturally be taken into account when it comes to safety guarantees: safety must be ensured between sampling times when no state measurement is available and no control adjustment is possible. The definition and characterization of reachability properties under SD models requires to take into account the lack of knowledge and control authority between sampling times: for example, the SD robust reach-avoid set describes initial states that can reach a given target set within N sampling steps while *continuously* avoiding a given unsafe set on the way. It has been shown that computing SD reachability constructs can be done by recursively using CT reachability over sampling intervals in the case of static constraints[16,17] and time-varying constraints.[18]

1.2. Related work

There are already some applications of Hamilton-Jacobi (HJ) reachability to MP. In ref. [11], the set of inevitably unsafe initial states is computed as the solution to a continuous dynamic game in order to ensure aircraft collision avoidance and, in ref. [19], the victory domain of a capture-the-flag game is shown to be a reach-avoid set described by a specific HJI PDE. In refs. [14, 15], the reach-avoid game under time-varying constraints is addressed in CT by solving a Hamilton–Jacobi–Bellman PDE in the state-time space and a time-varying double obstacle HJI PDE in the state space, respectively. However, as mentioned previously, the CT model is somewhat unrealistic.

Hence, reachability-based MP for SD systems has also been investigated. For instance in ref. [20], the set of safe initial states for a mobile robot represented by an SD model evolving in the presence of an unpredictable malicious agent is computed through recursive reachability computations, and a safety-preserving control law is synthesized. The control synthesis problem is specifically addressed in refs. [17, 21] for SD systems. Finally, closer to our work, the authors of ref. [16] introduced a systematic approach based on robust reach-avoid sets to synthesize guaranteed safety-preserving and target-reaching feedback policies for SD systems under bounded disturbances and applied this methodology to Unmanned Aerial Vehicles in ref. [22]. However these results only hold under the assumption of static state constraints and are restricted to point robots.

Other methods have been developed for MP under uncertainty that rely on "robustifying" model predictive control (MPC). For instance, mini-max MPC seeks to minimize online the cost of the

https://doi.org/10.1017/S0263574719001905 Published online by Cambridge University Press

worst-case uncertainty over open-loop controls, while feedback MPC reasons over control policies in closed-loop. However, these approaches are impractical because either overly conservative or computationally prohibitive, thus tube-based MPC emerged, which consists in computing invariant tubes within which the predicted state trajectory is guaranteed to remain regardless of uncertainty. In fact, these tubes are forward reachability constructs, in a sense that they characterize sets of states that *can be reached from a given state or region*, as opposed to backward reachability constructs that characterize states *from which a given state or region is reachable*. An exhaustive comparison of these homologue definitions in the context of safety analysis is available in ref. [5] and a detailed survey on MPC robust variants is available in ref. [23]. Another approach that has been recently developed is that one of ref. [24] which relies on the computation of funnel libraries based on sum-of-square programming and enables online robust MP in previously unknown environments.

1.3. Contributions and outline

In this paper, we show that reachability analysis can be used in a general and flexible manner as a formal approach to the robust MP problem in dynamic environments. *General* because provable deterministic guarantees of performance and safety can be provided that are compatible with the SD framework and hold under arbitrary dynamics and constraints. *Flexible* because changing the model, the task or the environment does not require modifications on the algorithm's core: the main idea is always to solve numerically specific differential games, namely reach-avoid games, in order to synthesize offline a guaranteed feedback controller. In our previous work,[18] we already proposed the extension to time-varying constraints of the SD HJ algorithm originally described in ref. [22] for static constraints. However, we did not provide a formal proof of the main result, neither did we consider an adversarial agent nor we restricted our scope to point robots. We intend to show that SD HJ algorithm can be further extended to more challenging dynamic environments (adversarial environments) through an appropriate differential game formulation and to non-point robots by reasoning in the configuration space instead of the state space. Furthermore, we explain how to mitigate the curse of dimensionality while preserving safety and performance guarantees. Hence our main contributions are:

- development of a formal approach to robust MP in dynamic environments
- extension of the SD HJ algorithm to time-varying constraints
- under-approximation of robust reach-avoid sets through HJ projections
- extension of the methodology to non-point robots.

Section 2 introduces the robust MP problem through a generic example and provides an overview of the SD HJ reachability approach. In Section 3, preliminary definitions are provided that include the robot's SD dynamic model, constraints, basic reachability constructs and set representation tools. Section 4 describes our SD HJ algorithm for dynamic – time-varying and adversarial – environments and the associated control synthesis. Section 5 demonstrates the validity of the approach on a case study and explains how to mitigate computational issues in higher dimensions. Finally, Section 6 provides a discussion on the performance of the methodology as well as research perspectives.

2. Problem Formulation

In this section, we illustrate the class of problems addressed by our method through a concrete but generic example scenario and we provide an overview of the SD HJ reachability methodology.

2.1. Robust MP problem

Consider a robot arm operating in a cluttered workspace and tasked with reaching in finite time a known moving target with its end-effector while avoiding collision with known moving obstacles (see Fig. 1). In addition, the robot must also avoid being captured by an adversarial agent, assumed to be unpredictable and malicious. Moreover, the robot is represented by an SD system subject to bounded external disturbances modeling uncertainty. Under these assumptions, the robust MP problem amounts to answer the following question: *how to steer the robot's end-effector to the target within N sampling steps while continuously avoiding collision with obstacles or capture by the pursuing agent in spite of uncertainties*. The answer to that question must contain:

https://doi.org/10.1017/S0263574719001905 Published online by Cambridge University Press

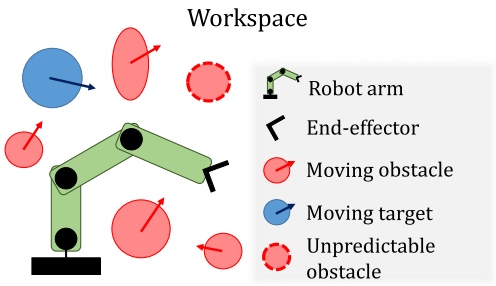

Fig. 1. Robot arm operating in a known time-varying workspace in the presence of an unpredictable agent.

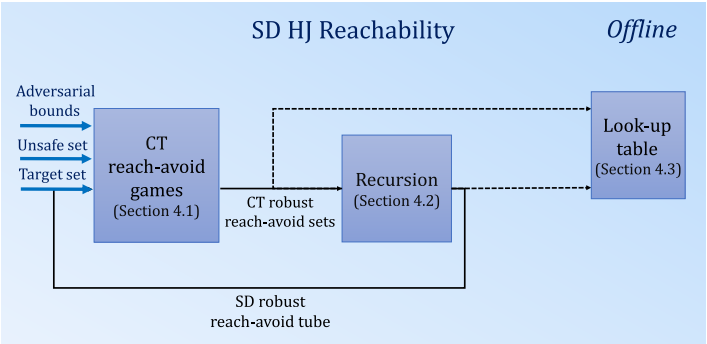

Fig. 2. Offline stage (SD HJ reachability): constraints and adversarial input bounds are used to compute CT reach-avoid constructs over sampling intervals (cf. Section 4.1) which are stored in a look-up table. A recursion formula enables to retrieve the current SD robust reach-avoid construct used as a new goal to reach over next sampling interval (cf. Section 4.2).

- *Analysis*: what is the set of admissible initial conditions?
- *Synthesis*: given an admissible initial condition, how to derive a suitable control policy?

2.2. *Overview of the approach*

First of all, the robot's success and failure configurations are expressed in a joint configuration space $\mathcal{C}$ describing the joint configuration of the robot and the pursuer. Also, the time-varying nature of the environment is taken into account by augmenting $\mathcal{C}$ with an additional time dimension (the configuration-time space, cf. Section 3.2). Then, the *analysis* part of the robust MP problem defined previously is formalized in reachability terms as the *SD robust reach-avoid problem* of which the solution is the set of admissible initial conditions, namely the *N-step robust reach-avoid tube* (cf. Section 3.3). It is calculated offline based on the SD HJ reachability algorithm (see Fig. 2): on each sampling interval, *CT robust reach-avoid sets* are computed by solving appropriate HJI PDEs (cf. Section 4.1) and a recursion is performed over sampling intervals, starting from the target set and propagating backwards the robust performance and safety properties (cf. Section 4.2). All intermediate reach-avoid constructs are stored in a look-up table and used to address the *synthesis* problem (see Fig. 3): a guaranteed feedback control policy is synthesized online through a look-up table exploration algorithm (cf. Section 4.3).

3. Preliminaries

3.1. *SD model*

As mentioned previously, many real-world cyber-physical systems are CT plants remotely operated through digital controllers updating their signals periodically upon measurements at a certain rate. As a matter of fact, our ability to observe and control such systems is impacted by this design pattern and it is crucial to incorporate it into mathematical models[25] for analysis and control synthesis. But purely

https://doi.org/10.1017/S0263574719001905 Published online by Cambridge University Press

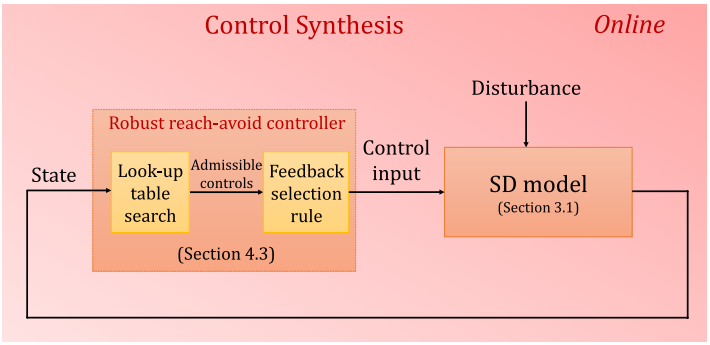

Fig. 3. Online stage (control synthesis): at each sampling step, the controller searches in the look-up table the set of viable controls corresponding to the current state (joint configuration) measurement and a *best* control input is selected based upon a predefined selection rule (cf. Section 4.3).

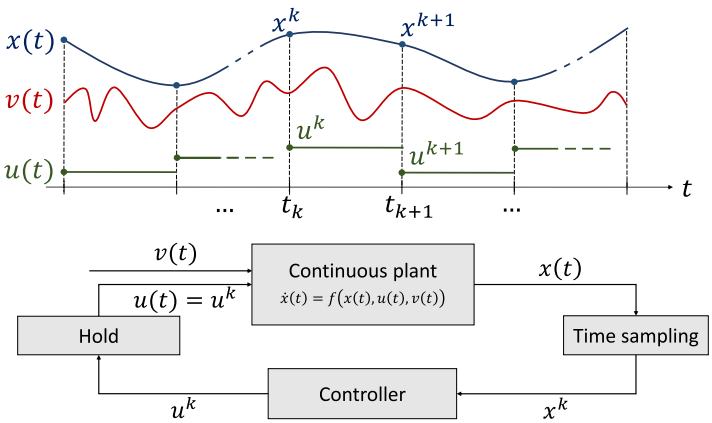

Fig. 4. SD model: the continuous state trajectory $x(.)$ is sampled at a rate δ. At time t_k, the k^{th} measurement x^k is received by the digital controller. An input value u^k is selected and held constant until the next state measurement x^{k+1} is available. Hence the full control signal is piecewise constant over sampling intervals.

CT models assume somehow unrealistically that continuous state knowledge and instantaneous control adjustment are possible, and discrete-time (DT) models simply neglect the plant's inter-sampling behavior. In fact the SD paradigm lays somewhat between CT and DT by assuming that the system's state evolves continuously over time but is only observed at a fixed sampling rate δ, while the control input is updated periodically based on those measurements (cf. Fig. 4). In addition to time-sampling, it is also common to assume that the control input's amplitude is quantized for simplicity and implementation convenience. From a modeling perspective, the SD model assumption enables to capture the system's open-loop behavior within sampling intervals (the control input cannot be adjusted faster than time sampling). From an analysis perspective, it enables to check whether the system violates safety constraints between consecutive measurements, and raises the critical question of operating it with a restricted control authority. Let us formalize this by considering the dynamical system

$$f : \begin{cases} \mathcal{X} \times \mathcal{U} \times \mathcal{V} \to & \mathcal{X} \\ (x, u, v) & \mapsto f(x, u, v) \end{cases} \tag{1}$$

where f is bounded and Lipschitz continuous in the state x, u is the control input, v is an external disturbance modeling uncertainty, $\mathcal{U}$ and $\mathcal{V}$ are compact and bounded. According to the aforementioned SD assumptions, x and v are allowed to vary continuously over time, whereas u is piecewise constant and ranges over a finite set of input values, that is, $\mathcal{U} = \{u_1, \ldots, u_L\}$. Given some state measurement x^k received at step k, the controller generates a constant control signal $u^k(.) \equiv u_j \in \mathcal{U}$ and a 1-step adversarial signal $v^k(.)$ is applied. The resulting open-loop trajectory over $[t_k, t_{k+1})$ is

https://doi.org/10.1017/S0263574719001905 Published online by Cambridge University Press

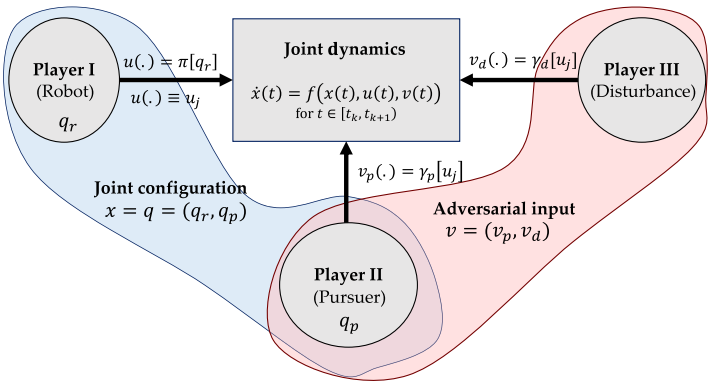

Fig. 5. On the one hand, the robot acts on the joint configuration $q = (q_r, q_p)$ through q_r using input $u(.)$ (chosen upon the policy π). On the other hand, the adversarial team acts on q through q_r and q_p using the disturbance v_d and the pursuer's input v_p (chosen upon the joint strategy $\gamma = (\gamma_p, \gamma_d)$).

the unique continuous solution to the Ordinary Differential Equation

$$\begin{cases} \dot{\phi}(t) = f(\phi(t), u^k(t), v^k(t)) \\ \phi(t_k) = x^k \end{cases} \tag{2}$$

This state trajectory is formally denoted as $\phi^k(.; x^k, u^k(.), v^k(.))$. Let us now formalize how $u^k(.)$ and $v^k(.)$ are selected:

- An *admissible feedback policy (or admissible control law)* π^k is a rule for selecting a 1-step constant control signal based on the k^{th} measurement x^k, that is, $\pi^k[x^k](.)$ is a 1-step constant function $u^k(.) \equiv u_j \in \mathcal{U}$ over $[t_k, t_{k+1})$.
- An *admissible adversarial strategy (or non-anticipative strategy)* γ^k is a rule for selecting a 1-step measurable adversarial signal based on a control input value, that is, γ^k maps an element $u_j \in \mathcal{U}$ to some measurable function $\gamma^k[u_j](.) = v^k(.) : [t_k, t_{k+1}) \to \mathcal{V}$.

Under the light of the above definitions, the SD closed-loop trajectory over $[0, t_N)$ can be fully defined given an initial state measurement x^0, an N-step feedback policy $\Pi_0^N = (\pi^0, \ldots, \pi^{N-1})$, and an N-step adversarial strategy $\Gamma_0^N = (\gamma^0, \ldots, \gamma^{N-1})$:

$$\forall k \in \{0, \ldots, N-1\}\ \forall t \in [t_k, t_{k+1}) \quad x_{cl}(t; x^0, \Pi_0^N, \Gamma_0^N) = \phi^k(.; x^k, u^k(.), v^k(.))$$

where x^k is the state measurement received at time t_k, $u_j = \pi^k[x^k]$, $v(.) = \gamma^k[u_j]$, and $\phi^k(.; x^k, u^k(.), v^k(.))$ is the solution of (2). When there is no ambiguity on the initial state, input signals, policies, or interval we are considering, $x^k, \pi^k, \gamma^k, u^k(.), v^k(.)$ will be abbreviated by $x, \pi, \gamma, u(.), v(.)$, the open-loop trajectory by $x(.)$, and the closed-loop trajectory by $x_{cl}(.)$.

In the rest of the paper, the state $x \in \mathcal{X}$ will actually be replaced by $q \in \mathcal{C}$, representing the robot–pursuer joint configuration $q = (q_r, q_p) \in \mathcal{C} = \mathcal{C}^r \times \mathcal{C}^p$, u will represent robot's control input chosen upon policy π and v will denote the joint adversarial input $v = (v_p, v_d)$ where v_d is the external disturbance modeling uncertainty and v_p is the malicious agent's control input, both selected upon the joint strategy $\gamma = (\gamma_p, \gamma_d)$. This special interpretation of x, u, and v leads to regard our system (1) as a differential game setting between two teams, as illustrated in Fig. 5.

3.2. *Configuration-time constraints*

Consider a workspace $\mathcal{W} \subset \mathbb{R}^2$ or $\mathbb{R}^3$ containing a union of known moving obstacles $\mathcal{OBS}(t)$ and a known moving target $\mathcal{TAR}(t)$. Since the robot and the pursuer are not assumed to be points, their geometry and volume occupied in given configurations must be taken into account in order to characterize safety. We define the *configuration-time* unsafe and target sets as

$$\begin{cases} \mathbb{G} = \bigcup_{t \in [0, t_N]} \mathcal{G}(t) \times \{t\} \\ \mathbb{L} = \bigcup_{t \in \{0, \ldots, t_N\}} \mathcal{L}(t) \times \{t\} \end{cases} \tag{3}$$

https://doi.org/10.1017/S0263574719001905 Published online by Cambridge University Press

where $\mathcal{G}(t)$ is the set of forbidden configurations at time t and $\mathcal{L}(t)$ the set of goal configurations defined as:

$$\begin{cases} \mathcal{G}(t) = \{q \mid \mathcal{R}(q_r) \cap \mathcal{OBS}(t) \neq \emptyset \ \vee \ \mathcal{R}(q_r) \cap \mathcal{P}(q_p) \neq \emptyset\} \\ \mathcal{L}(t) = \{q \mid \mathcal{E}[q_r] \in \mathcal{TAR}(t)\} \end{cases} \tag{4}$$

where $\mathcal{R}(q_r), \mathcal{P}(q_p) \subset \mathcal{W}$ denote the spaces occupied by the robot in configuration q_r and by the pursuer in configuration q_p, respectively, and $\mathcal{E}[q_r] \in \mathcal{W}$ is the end-effector location. It should be observed that $\mathbb{L}$ is a finite sequence as a consequence of the SD assumption: the target's trajectory between sampling times is not important since it should be reached exactly at a sampling time, that is, when a state measurement is available to verify that it has effectively been reached. Thus for any k, $\mathcal{L}(t_k)$ will be simply denoted $\mathcal{L}_k$. On the contrary, the unsafe sets from an infinite sequence over the CT space $[0, t_N]$ since safety should be ensured continuously throughout time – even between sampling times when no state measurement is available.

3.3. SD robust reach-avoid problem over $[t_0, t_N]$

The SD model and the configuration-time constraints being properly defined, we can now formally express the set of admissible initial conditions to the robust MP problem.

Definition 1. The $N-i$-step robust reach-avoid tube is the set of (q, t_k) that can be steered into $\mathbb{L}$ within $N-i$ steps while continuously avoiding $\mathbb{G}$ regardless of the adversarial strategy:

$$\mathbb{RA}_{N-i}(\mathbb{L}, \mathbb{G}) = \{(q, t_k) \mid \exists \Pi_k^m, \ \forall \Gamma_k^m, \ \exists k' \in \{k, \dots, m\} \\ q_{cl}(t_{k'}) \in \mathcal{L}_{k'} \ \wedge \ \forall \tau \in [t_k, t_{k'}] \ q_{cl}(\tau) \notin \mathcal{G}(\tau)\}$$

where $m = \min\{N, k+N-i\}$ and $q_{cl}(.) = q_{cl}(.; q, \Pi_k^m, \Gamma_k^m)$.

Intuitively, this set contains good initial conditions from the robot's perspective, that is, the ones for which there exist a robust $N-i$ step target-reaching and safety-preserving piecewise feedback policy. Using m enables to exclude the states that would reach the target later than step N. A natural convention for $i = N$ is to set $\mathbb{RA}_0(\mathbb{L}, \mathbb{G}) = \mathbb{L}$. As a matter of fact, solving the *analysis* part of the robust MP problem consists in calculating the N-step robust reach-avoid tube $\mathbb{RA}_N(\mathbb{L}, \mathbb{G})$ – as it characterizes exactly the admissible states for the full horizon problem. This is a purpose of the SD HJ reachability offline algorithm illustrated in Fig. 2 and described in the next section.

3.4. CT reachability constructs

Throughout the offline SD HJ reachability algorithm, the N-step robust reach-avoid tube defined above will be calculated from simpler CT reachability constructs. Let $\mathcal{S} \subset \mathcal{C}$, $u_j \in \mathcal{U}$.

Definition 2. The reach set of $\mathcal{S}$ at t_{k+1} associated with u_j

$$\underline{\mathcal{R}}_k^j(\mathcal{S}) = \{q \mid \forall \gamma \ q(t_{k+1}) \in \mathcal{S}\}$$

contains every configuration that can be driven by u_j into $\mathcal{S}$ in exactly one step starting from t_k regardless of the adversarial strategy γ (cf. Fig. 6). Note that $q(.)$ represents the open-loop trajectory starting from q and driven by u_j and $\gamma[u_j]$.

Definition 3. The avoid tube of $\mathbb{G}$ over $[t_k, t_{k+1}]$ associated with u_j

$$\mathcal{A}_k^j(\mathbb{G}) = \{q \mid \forall \gamma \ \forall t \in [t_k, t_{k+1}] \ q(t) \notin \mathcal{G}(t)\}$$

contains every configuration that can continuously avoid entering $\mathcal{G}(t)$ between t_k and t_{k+1} under u_j and regardless of the adversarial strategy γ. This definition naturally holds for any static unsafe set $\mathcal{G}$ (replacing $\mathbb{G}$ and $\mathcal{G}(t)$ by $\mathcal{G}$ in the above expression).

Definition 4. The CT robust reach-avoid set of $\mathcal{S}$, $\mathbb{G}$ over $[t_k, t_{k+1}]$ associated with u_j

$$\underline{\mathcal{RA}}_k^j(\mathcal{S}, \mathbb{G}) = \underline{\mathcal{R}}_k^j(\mathcal{S}) \bigcap \mathcal{A}_k^j(\mathbb{G})$$

https://doi.org/10.1017/S0263574719001905 Published online by Cambridge University Press

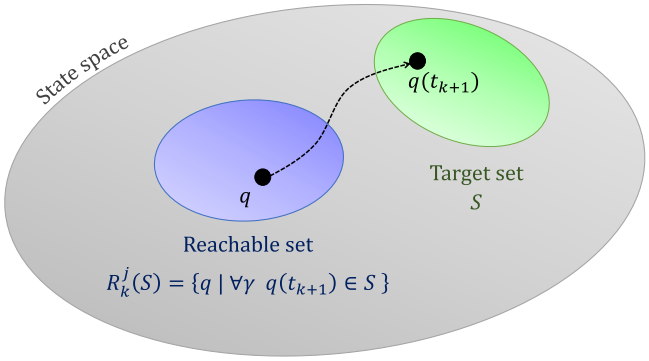

Fig. 6. The reach set of $\mathcal{S}$ at t_{k+1} associated with u_j: any q that belongs to this set can be steered into $\mathcal{S}$ by u_j in one step.

contains every configuration that can be driven into $\mathcal{S}$ exactly at time t_{k+1} while continuously avoiding $\mathcal{G}(t)$ between t_k and t_{k+1} under u_j and regardless of the adversarial strategy γ.

3.5. Level set representation

Definition 5. A level set (LS) function for the static set $\mathcal{A} \subset \mathbb{R}^n$ is a function $a(.) : \mathbb{R}^n \mapsto \mathbb{R}$ such that $\mathcal{A} = \{x \in \mathbb{R}^n \mid a(x) \leq 0\}$.

If $a(.)$, $b(.)$ are respective LS functions for $\mathcal{A}, \mathcal{B} \subset \mathbb{R}^n$, then we state the following properties:

$x \mapsto \min\{a(x), b(x)\}$	is a LS function for	$\mathcal{A} \cup \mathcal{B}$
$x \mapsto \max\{a(x), b(x)\}$	is a LS function for	$\mathcal{A} \cap \mathcal{B}$
$x \mapsto -a(x)$	is a LS function for	$\bar{\mathcal{A}}$

Any time-varying set $\mathbb{A} \subset \mathbb{R}^n \times [t_0, t_N]$ is represented by an LS function $a(., .)$ such that $\mathbb{A} = \{(x, t) \in \mathbb{R}^n \times [t_0, t_N] : a(x, t) \leq 0\}$

4. General Approach

We describe now the methodology introduced in Section 2.2: Sections 4.1 and 4.2 correspond to the offline stage (analysis) and Section 4.3 to the online stage (synthesis).

4.1. CT reach-avoid game over a sampling interval (offline analysis)

We first consider over the sampling interval $[t_k, t_{k+1}]$ the reach-avoid game opposing the robot with input u to the adversarial team with input $v = (v_p, v_d)$. The goal of the robot is to steer q into some static target set $\mathcal{S} = \{q \mid s(q) \leq 0\}$ exactly at time t_{k+1} while continuously avoiding $\mathbb{G} = \{(q, t) \mid g(q, t) \leq 0\}$, whereas the adversarial team must prevent the robot from winning the game. Starting from q, the cost for the robot of playing $\pi[q] \equiv u_j$ against adversarial strategy γ yields

$$V^j(q(.)) = \max\{s(q(t_{k+1})), \max_{t_k \leq \tau \leq t_{k+1}} -g(q(\tau), \tau)\} \tag{5}$$

Obviously, the robot wins the game when $V^j(q(.)) \leq 0$. Therefore, the goal of the adversarial team is to maximize $V^j(q(.)) \leq 0$ by adjusting its strategy γ so as to get the optimal cost for this game:

$$V^j(q) = \sup_{\gamma} V^j(q(.)) \tag{6}$$

It can be observed that the robot's victory domain is precisely the CT robust reach-avoid set, that is, $\underline{\mathcal{RA}}_k^j(\mathcal{S}, \mathbb{G}) = \{q \mid V^j(q) \leq 0\}$ (see Fig. 7).

https://doi.org/10.1017/S0263574719001905 Published online by Cambridge University Press

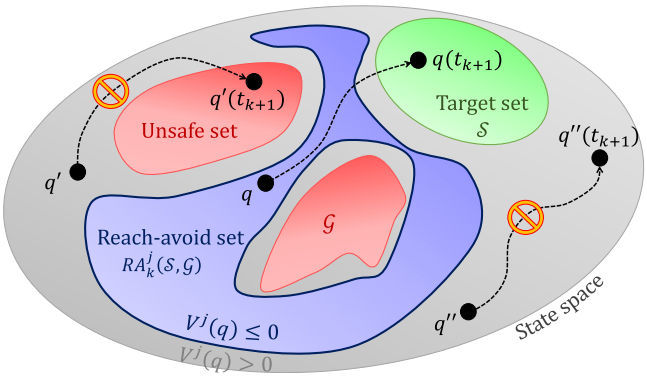

Fig. 7. Initial states q' and q'' are located outside of the reach-avoid set; thus, there exists a defeating adversarial strategy. On the contrary, q belongs to the reach-avoid set, thus it can be driven safely into $\mathcal{S}$ by u_j despite any opposing strategy. The boundary of the reach-avoid set corresponds to the zero-level of the cost function $V^j(.)$.

Proposition 1. *The optimal cost verifies* $V^j(q) = \Phi(q, t_k)$, *where* $\Phi(.,.) : \mathcal{C} \times [t_k, t_{k+1}] \to \mathbb{R}$ *is the viscosity solution to the HJI PDE:*

$$\begin{cases} \max\{-g(x,\tau) - \Phi(x,\tau), \frac{\partial \Phi}{\partial t}(x,\tau) + H(x, \frac{\partial \Phi}{\partial x}(x,\tau))\} = 0 \\ H(x,p) = \sup_{v \in \mathcal{V}} p^T f(x, u_j, v) \\ \Phi(x, t_{k+1}) = s(x) \end{cases}$$

This result has been proven in the case of static state constraints in ref. [12] and extended to time-varying constraints in ref. [15]. The present HJI PDE is in fact a particular case of the one studied in ref. [15] in which the state is only required to reach the target *exactly* at the end of the time interval instead of *within* the time interval.

4.2. SD HJ reachability recursion (offline analysis)

We show now how the N-step robust reach-avoid tube can computed offline by using a sequence of nested sets. This sequence relies on the 1-*step robust reach-avoid set operator* which is formally defined for any $\mathbb{S} \subset \mathcal{C} \times \{t_0, \ldots, t_{N-1}\}$ as

$$\underline{\mathbb{RA}}_1(\mathbb{S}, \mathbb{G}) = \{(q, t_k) \mid \exists \Pi_k^{k+1},\ \forall \Gamma_k^{k+1},\ q_{cl}(t_{k+1}) \in \mathcal{S}_{k+1} \ \wedge\ \forall \tau \in [t_k, t_{k+1}]\ q_{cl}(\tau) \notin \mathcal{G}(\tau)\} \quad (7)$$

The set $\underline{\mathbb{RA}}_1(\mathbb{S}, \mathbb{G})$ contains every initial (q, t_k) that can robustly reach $\mathbb{S}$ while avoiding $\mathbb{G}$ in exactly 1 step. Conceptually, the SD HJ reachability algorithm consists in applying this operator recursively starting from $\mathbb{L}$ in order to back-propagate the 1-step robust reach-avoidance property over sampling intervals. To this end, we state Theorems 1 and 2 – the main result of this section. Their technical proofs are available in Appendices A.1 and A.2. These theorems, respectively indicate that

- The $N - i$-step robust reach-avoid tube can be calculated by propagating the 1-step robust reach-avoid set operator in $\mathcal{C} \times \{t_0, \ldots, t_N\}$
- Applying the 1-step robust reach-avoid set operator amounts to compute CT robust reach-avoid sets in $\mathcal{C}$

Theorem 1. *Let us consider the following recursion:*

$$\begin{cases} \mathbb{S}^N = \mathbb{L} \\ \mathbb{S}^i = \underline{\mathbb{RA}}_1(\mathbb{S}^{i+1}, \mathbb{G}) \bigcup \mathbb{S}^{i+1} \quad \forall N - 1 \geq i \geq 0 \end{cases}$$

Then for any $i \in \{0, \ldots, N\}$, $\mathbb{S}^i = \mathbb{RA}_{N-i}(\mathbb{L}, \mathbb{G})$.

Theorem 2. *For any* $i \in \{0, \ldots, N-1\}$ *and* $k \in \{0, \ldots, i\}$,

$$\begin{cases} \mathcal{S}_k^i = \mathcal{S}_k^k & \text{if } k \in \{i, \ldots, N\} \\ \mathcal{S}_k^i = \mathcal{S}_k^{i+1} \bigcup_{j=1}^{L} \underline{\mathcal{RA}}_k^j(\mathcal{S}_{k+1}^{i+1}, \mathbb{G}) & \text{otherwise} \end{cases}$$

https://doi.org/10.1017/S0263574719001905 Published online by Cambridge University Press

In particular, Theorem 2 not only provides a practical formula to calculate the recursion of Theorem 1 (see Algorithm 1) but also implies that the calculation of $\mathbb{S}^N, \ldots, \mathbb{S}^0$ can be performed in $\mathcal{C}$ by solving $r\frac{N(N+1)}{2}$ HJI PDEs like the one of Proposition 1.

Algorithm 1 Calculate $\mathbb{S}^i$ for all $0 \leq i \leq N$

$\mathbb{S}^N = (\mathcal{S}_0^N, \ldots, \mathcal{S}_N^N) \leftarrow (\mathcal{L}_0 \setminus \mathcal{G}(t_0), \ldots, \mathcal{L}_N \setminus \mathcal{G}(t_N))$
for $i = N-1$ to 0 **do**
  for $k = 0$ to i **do**
    $\mathcal{S}_k^i \leftarrow (\bigcup_{u_j \in \mathcal{U}} \underline{\mathbb{RA}}_{k \to k+1}^{u_j}(\mathcal{S}_{k+1}^{i+1}, \mathbb{G})) \bigcup \mathcal{S}_k^{i+1}$
  end for
  for $k = i+1$ to N **do**
    $\mathcal{S}_k^i \leftarrow \mathcal{S}_k^k$
  end for
  $\mathbb{S}^i \leftarrow (\mathcal{S}_0^i, \ldots, \mathcal{S}_N^i)$
end for
Return $(\mathbb{S}^N, \ldots, \mathbb{S}^0)$

4.3. SD robust reach-avoid controller (online synthesis)

Supposing that we retained the sets $\underline{\mathcal{RA}}_k^j(\mathcal{S}_{k+1}^{i+1}, \mathbb{G})$ during the recursion, these can be used to derive a safe and robust feedback policy. Starting at time 0 from an initial configuration $q^0 \in \mathcal{S}_0^0$, first search the largest k_0 such that $q^0 \in \mathcal{S}_{k_0}^0$, then record every u_j such that $q^0 \in \underline{\mathcal{RA}}_{k_0}^j(\mathcal{S}_{k_0+1}^1, \mathbb{G})$ such an input always exists since by Proposition 2, $\mathcal{S}_{k_0}^0$ is included in $\bigcup_{j=1}^L \underline{\mathcal{RA}}_{k_0}^j(\mathcal{S}_{k_0+1}^1, \mathbb{G})$. This leads to the set-valued feedback $\mathcal{F}^{\mathcal{U}}(q^0) = \{u_j \in \mathcal{U} \mid q^0 \in \underline{\mathcal{RA}}_{k_0}^j(\mathcal{S}_{k_0+1}^1, \mathbb{G})\}$ from which a *best* input u^* can be selected upon a predefined rule. Once a input is selected, q^0 is driven safely and robustly to some $q^1 \in \mathcal{S}_{k_0+1}^1$ exactly at step 1. The procedure is then repeated for q^1, that is, search the largest k_1 such that $q^1 \in \mathcal{S}_{k_1}^1$, then calculate the set-valued feedback $\mathcal{F}^{\mathcal{U}}(q^1) = \{u_j \in \mathcal{U} \mid q^1 \in \underline{\mathcal{RA}}_{k_1}^j(\mathcal{S}_{k_1+1}^2, \mathbb{G})\}$ and select an input. This procedure is repeated until $\mathbb{L}$ is reached (cf. Algorithm 2). Figure 8 illustrates the algorithm.

Algorithm 2 Online control synthesis

$q \leftarrow q_0,\ i \leftarrow 0$
while $(q, t_i) \notin \mathbb{L}$ **do**
  $k_i \leftarrow N+1$
  repeat
    $k_i \leftarrow k_i - 1$
  until $q \in \mathcal{S}_{k_i}^i$ or $k_i = 0$
  if $k_i = 0$ **then**
    STOP ($\mathbb{L}$ has been reached)
  else
    Record every u_j such that $q \in \underline{\mathcal{RA}}_{k_i}^j(\mathcal{S}_{k_i+1}^{i+1}, \mathbb{G})$
    Select an input u^*
    Apply u^* and update q
  end if
end while

5. Case Study with the 2-DoF Robot Arm

In this section, we demonstrate the validity of our approach by solving the robust MP problem for a simulated 2-degrees of freedom robot arm. The robot operates in a cluttered environment containing unpredictable malicious obstacles along with known moving obstacles and a target. We provide safety and performance guarantees by applying the SD HJ algorithm in order to synthesize an SD target-reaching and safety-preserving control policy that is provably robust against any adversarial policies.

https://doi.org/10.1017/S0263574719001905 Published online by Cambridge University Press

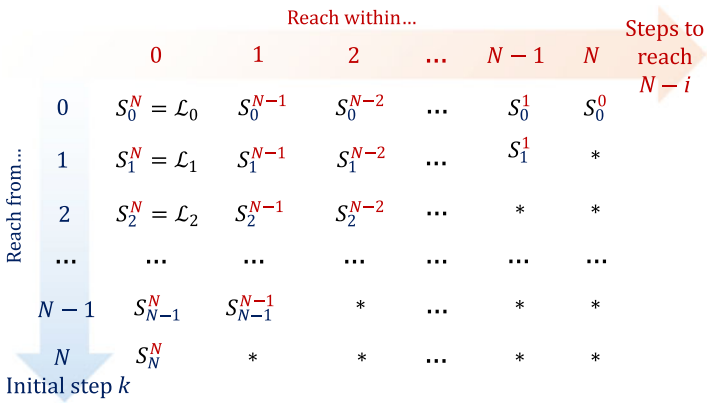

Fig. 8. The look-up table of SD robust reach-avoid tubes $\mathcal{S}_k^i$. At step k, the controller looks in the k^{th} row for the *leftest* set that contains the current state and pushes it down left. Thus the action of the controller can be viewed as driving the state from up right to down left *as left as possible*, in order to reach the first column ($\mathbb{L}$) in a minimal number of steps.

5.1. Computational efficiency

The present algorithm requires to solve numerically HJI PDEs which becomes extremely challenging in high dimensions. Tackling the curse of dimensionality arising in HJ reachability is subject to active research[26,27] and scalable techniques such as complexity reduction or fast numerical schemes for HJ-like equations have emerged recently in the literature. However, they often impose restrictive assumptions unlikely to be satisfied in practice[28–30] (e.g., prismatic constraints, CT model) or have not been extended yet to the kind of HJI PDEs arising in reach-avoid games.[31–33]

In order to be able to apply the SD HJ algorithm of Section 4 to higher dimensional systems, we propose to mitigate the algorithm complexity by projecting HJI PDEs onto lower dimensional spaces (cf. Appendices A.3, A.4, A.5 for technical details). In fact, we adapted the technique originally developed in ref. [34] in order to calculate under-approximations of reach-avoid sets with a reduced computational cost. The main idea consists in projecting the full dimensional constraints onto a lower dimensional subspace and to evolve the associated HJI PDE in this subspace with modified Hamiltonian treating unmodeled dimensions as external disturbances. An under-approximation of the original solution can be retrieved by back-projection into the original space, as shown later in Section 6, which maintains robust safety and performance guarantees.

In this context, we decided to study the case of a 2-DoF robot operating in the presence of a 2-dimensional pursuer so that the corresponding 4-dimensional MP problem is easily tractable using our algorithm and a reference solution is available. This way, we can evaluate the correctness and performance of the original SD HJ algorithm as well as verify the conservativeness, the accuracy, and the computation time improvement of the HJ projection-based approximation. As emphasized in ref. [34], the main challenge to use HJ projections with higher dimensional models is to choose appropriate subspaces on which to project. As a matter of fact, poorly chosen projections may result in overly conservative solutions. In practice, projecting along the dynamics' natural directions works well, and performing multiple projections in several directions can help reducing the final approximation's conservativeness through intersections. Finally, nothing theoretically prevents from using low dimensional projection subspaces – by projecting several times – in order to achieve a better computation time. The main challenges in doing so may be the implementation complexity and the risk of excessive conservativeness due to loss of information when projecting.

5.2. Model

The robot is a 2-DoF planar robot arm with links of length $l = [l_1, l_2]$, configuration $q_r = (q_{r,1}, q_{r,2}) \in \mathcal{C}^r = [-\pi, +\pi]^2$, and end-effector position $\mathcal{E}[q_r] \in \mathcal{W} \subset \mathbb{R}^2$. Its direct geometric model yields

$$q_r = (q_{r,1}, q_{r,2}) \mapsto \begin{pmatrix} l_1 \cdot \cos(q_{r,1}) + (l_1 + l_2) \cdot \cos(q_{r,2}) \\ l_1 \cdot \sin(q_{r,1}) + (l_1 + l_2) \cdot \sin(q_{r,2}) \end{pmatrix} \tag{8}$$

https://doi.org/10.1017/S0263574719001905 Published online by Cambridge University Press

The robot's dynamic model $f_r : (q_r, u, v_d) \mapsto f_r(q_r, u, v_d)$ is a simple integrator subject to external bounded disturbances:

$$\begin{pmatrix} \dot{q_{r,1}} \\ \dot{q_{r,2}} \end{pmatrix} = \begin{pmatrix} u_1 + v_{d,1} \\ u_2 + v_{d,2} \end{pmatrix} \tag{9}$$

where $u = (u_1, u_2)$ is the robot's control input and $v_d = (v_{d,1}, v_{d,2})$ the external disturbance modeling uncertainty. The inputs u and v_d are bounded as $\|u\|_\infty \leq u^{\max}$, $\|v_d\|_\infty \leq v_d^{\max}$. In accordance with the SD model, the control space is sampled $\mathcal{U} = \{-u^{\max}, \ldots, +u^{\max}\}^2$ and we set the number of samples per dimension to 5 so that $L = 25$.

The pursuer is assumed to be a disk of radius r_p moving in $\mathcal{W}$. The pursuer's center's position $p = (p_x, p_y) \in \mathcal{W}$ is equal to its configuration $q_p = (q_{p,1}, q_{p,2}) \in \mathcal{C}^p$ for simplicity (although nothing prevents from considering the case where $\mathcal{C}^p \neq \mathcal{W}$). The space occupied by the pursuer in configuration q_p is $\mathcal{P}(q_p) = \mathcal{D}(p, r_p)$, where $\mathcal{D}(p, r_p)$ represents the disk centered in p with radius r_p. The pursuer's actuation is modeled as a simple integrator as well:

$$\dot{q_p} = v_p \tag{10}$$

where $v_p = (v_{p,1}, v_{p,2})$ is the pursuer's bounded control input such that $\|v_p\|_\infty \leq v_p^{\max}$.

5.3. Obstacles and target

Target and unsafe sets are defined according to Section 3.2. For implementation simplicity we made the assumption that the workspace target and obstacles are disks and moving along straight lines at constant speed, but the present framework enables to consider any kind of shapes and motion patterns. The workspace moving target is a collection of $\mathcal{TAR}(t) \subset \mathcal{W}$ such $\mathcal{TAR}(t) = \mathcal{TAR}(0) + v_{\mathcal{TAR}} \cdot t$. The moving obstacle set is a collection of obstacle sets $\mathcal{OBS}(t)$ for $t \in [0, t_N]$. In Figure 9(a), the 2D workspace is represented. The unsafe regions are represented in red and include one static obstacle, two moving obstacles with circular and linear trajectories at constant speed, and the unpredictable obstacle as a bold dotted black contour. The successive target regions are represented in blue. The white arrows represent the initial velocities of the moving objects. In Fig. 9(b), the robot's 2D configuration space $\mathcal{C}^r$ is represented (it is actually a 2D slice of the 4D joint configuration space $\mathcal{C}$ at some fixed pursuer's configuration q_p). The initial $\mathcal{C}$-unsafe and $\mathcal{C}$-target sets are represented in red and blue, respectively and the bold dotted contoured set contains the configurations corresponding to a collision with the pursuer in $\mathcal{W}$.

5.4. Implementation

The algorithm was implemented using MATLAB and the LS Toolbox.[35] All sets are represented on a fixed grid by their LS function and whenever a function must be evaluated between grid nodes, a linear interpolation is used. The general terms of HJI PDEs are calculated using a Lax–Friedrich approximation, the spatial derivative in the Hamiltonian is approximated with a second-order upwind scheme and all constrained ordinary differential equations are integrated forward in time using a second-order accurate total variation diminishing Runge–Kutta integrator (cf. ref. [35], Sections 3.4, 3.5 and 3.6).

In theory, the sets $\mathcal{L}(t)$ and $\mathcal{G}(t)$ can be constructed by exhaustive $\mathcal{C}$-space explorations, however, this is costly and impractical. Actually, the $\mathcal{C}$-target can be obtained by exploring only $\mathcal{C}^r$ since it does not depend on q_p. Furthermore, we only need $\mathcal{L}(t)$ at sampling times t_k; thus, only N $\mathcal{C}^r$-space explorations are required. The computation of the unsafe set is somehow trickier, since an infinite number of $\mathcal{C}$-space exploration is required. By using the decomposition introduced in Appendix A.3, the number of explorations is reduced to 1 for the static set $\mathcal{G}^2$, but is still infinite for the time-varying set $\mathbb{G}^1$. In order to circumvent this issue, we use the view-time approach[36] which consists in over-approximating $\mathcal{OBS}(t)$ by the volume it swept over the current sampling interval, that is:

$$\mathcal{OBS}_{[t_k, t_{k+1}]} = \bigcup_{\tau \in [t_k, t_{k+1}]} \mathcal{OBS}(\tau)$$

This volume defines a virtual static obstacle containing the successive volumes occupied by the real moving obstacle during $[t_k, t_{k+1}]$. Then one simply needs to explore the $\mathcal{C}^r$-space N times in order to

https://doi.org/10.1017/S0263574719001905 Published online by Cambridge University Press

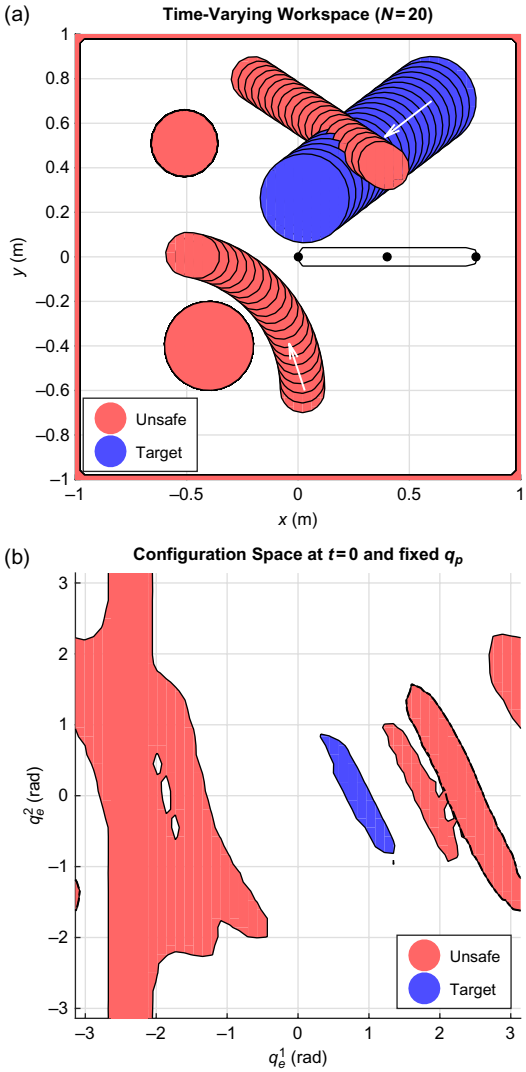

Fig. 9. Workspace and configuration space (cf. Section 3.2). (a) Known time-varying 2D workspace $\mathcal{W}$ with an unpredictable obstacle. (b) Robot's 2D configuration space $\mathcal{C}^r$.

construct static $\mathcal{C}^r$-unsafe sets $(\mathcal{G}^1_{[t_k,t_{k+1}]})_{k=0,..,N-1}$ corresponding to $(\mathcal{OBS}_{[t_k,t_{k+1}]})_{k=0,..,N-1}$. The full $\mathcal{C}$-unsafe set is retrieved by back-projection into $\mathcal{C}$. Finally, in the HJI PDE of Proposition 1, the time-dependent LS function $g(.,t)$ of $\mathcal{G}(t)$ is replaced by the static LS function $g_{[t_k,t_{k+1}]}(.)$ of $\mathcal{G}_{[t_k,t_{k+1}]}$.

5.5. Results

In this simulation, the sampling period was chosen to be $\delta = 0.1$ s (i.e., the feedback controller runs at 10 Hz) and the 4D joint configuration space $\mathcal{C}$ was discretized uniformly into a grid of 25^4 nodes. The corresponding discretization steps in the robot's configuration space $\mathcal{C}^r$ and the pursuer's space $\mathcal{C}^p$ are about 0.2 rad and 0.03 m, respectively. Figure 10 shows a 2D slice of the SD reach-avoid tubes computed through HJ reachability (Section 4.2). For clarity, we did not display every $\mathcal{S}^i_k$ but rather all $\mathcal{S}^i_0$, that is, sets of states that can reach the moving target within $N-i$ steps starting from step 0. Computations took approximately 92 min with an Intel Core i5-6500 CPU.

Figure 11 shows the same sequence of sets computed through HJ projections onto 3D subspaces which took less than 8 min. Games 2 and 3 were projected in $q_{p,2}$ and $q_{p,1}$ directions according to the theoretical results of Appendices A.3, A.4, A.5 and the full dimensional reachability constructs retrieved by back-projection and intersection. We performed simulations of the feedback controller by starting from various initial conditions and observed that in every case, regardless of the disturbance and malicious agent's actions, the robot succeeds in safely reaching the target provided that

https://doi.org/10.1017/S0263574719001905 Published online by Cambridge University Press

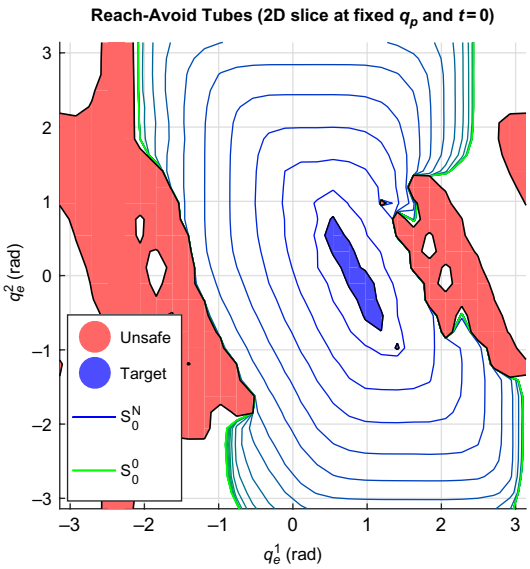

Fig. 10. Robust reach-avoid tubes sequence (cf. Section 4.2).

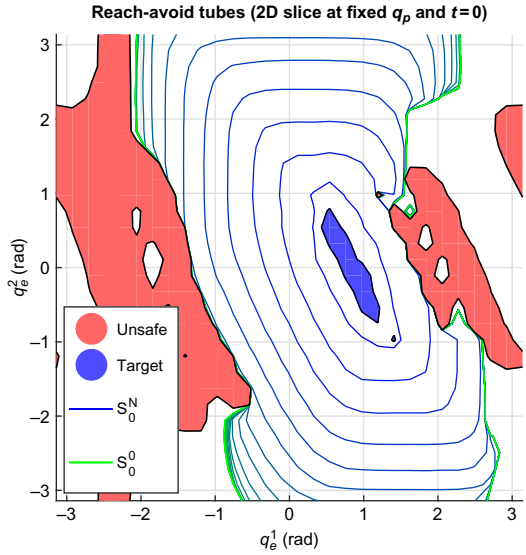

Fig. 11. Under approximation of the robust reach-avoid tubes sequence based on HJ projections.

its initial configuration is admissible. Figure 12 shows a trajectory generated by the feedback controller in the workspace (Fig. 12(a)) and in the configuration space (Fig. 12(b)) that reaches the target within 7 steps (0.7 s). In this simulation, we assumed that the adversarial strategy consisted in selecting piecewise constant randomly valued signals, that is, a random v is chosen within adversarial bounds at each step and kept constant over one sampling period.

6. Discussion

6.1. Simulation results

We observed in our simulations that any initial configuration within the SD robust reach-avoid tube can be driven by the feedback controller safely and robustly to the target within N steps despite the uncertain factors. It should be emphasized that due to the game-theoretic basis of our approach, safety and performance are ensured regardless of the adversarial strategy. For instance, let us assume that the pursuer acts more aggressively, for example, instead of selecting randomly its input value at each sampling time, it effectively attempts to enter in collision with the robot by choosing those

https://doi.org/10.1017/S0263574719001905 Published online by Cambridge University Press

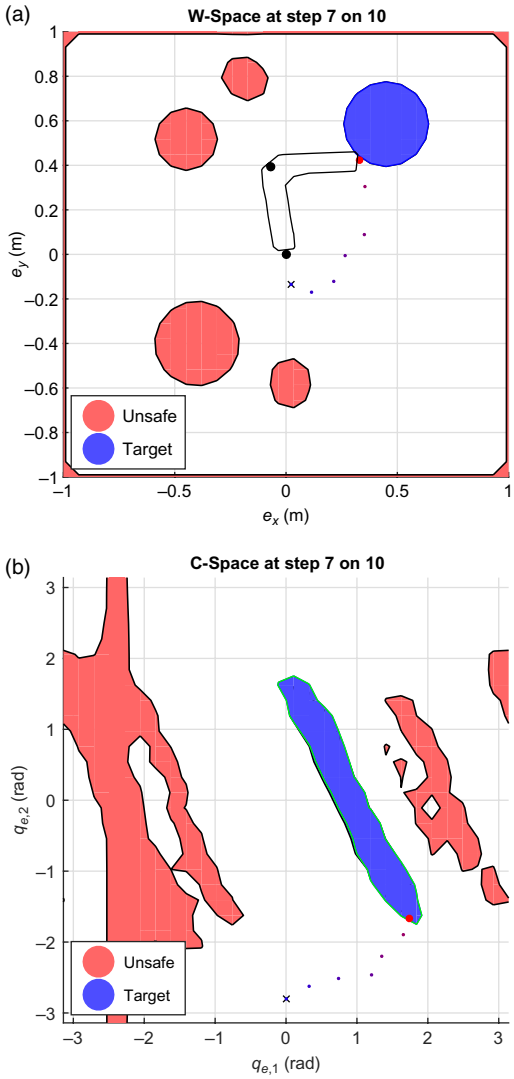

Fig. 12. 2-D views in $\mathcal{C}$-space and $\mathcal{W}$-space of a trajectory generated by the controller. (a) Trajectory of the end-effector in $\mathcal{W}$. (b) Trajectory of the robot's in $\mathcal{C}$ (2D slice view).

inputs that minimize their relative distance. As verified in simulations, this change of adversarial strategy would not affect safety nor performance because the feedback controller has been designed to ensure safety and performance under worst-case scenarios, that is, by optimizing the cost over the space of admissible adversarial strategies.

Furthermore, it should be noted that changing the nature of the adversarial agent does not require significant modifications of the algorithm: all the theoretical results hold if an unpredictable "evading" target is considered instead of an unpredictable pursuing obstacle: it suffices to change the constraints definition of Section 3.2. Concretely, the new *evader* is incorporated into the definition of the target set, whereas the unsafe set is somehow simplified and contains only known obstacles. The decompositions of Appendix A.3 consist in an n_r-dimensional time-varying "avoid" part and an n-dimensional "reach" part that can be under-approximated using the HJ projection method of Appendices A.4 and A.5. Figure 13 shows that the reach-avoid tubes obtained by HJ projections are indeed under-approximations of the reach-avoid tubes obtained by full dimensional computations, which are conservative from both safety and performance perspectives.

Figure 14 shows the performance of the original SD HJ algorithm against the HJ projections-based one. The HJ projection method enables to compute the sequence of reach-avoid tubes in a reasonable time (a few minutes) on moderately dense grids, whereas the original algorithm becomes almost intractable (a few hours). When gradually increasing the number of nodes per dimension, the

https://doi.org/10.1017/S0263574719001905 Published online by Cambridge University Press

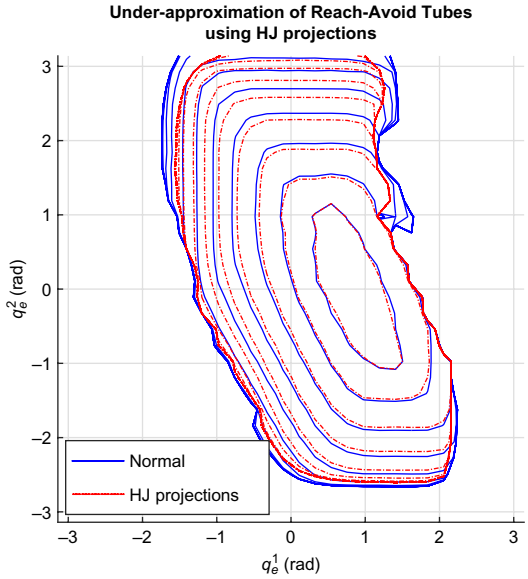

Fig. 13. Comparison of original reach-avoid tubes and their under-approximation based on HJ projections in directions q_p^1 and q_p^2 (cf. Appendices A.4 and A.5).

# nodes per dimension	25	30	35	40
Original	586	1540	3093	5615
HJ projections	105	190	347	549

Fig. 14. Computation times (in seconds) of both algorithms for various grid densities using an Intel Xeon(R) W-2145 CPU. The computation time includes the initialization time (construction of configuration space constraints) and the SD HJ recursion (HJI PDEs solving and sets operations).

ratio of the original algorithm's computation time over the HJ projection method's computation time increases from 5 to 10, that is, the latter scales much better with grid density than the former. This confirms that solving more HJI PDEs in lower dimensional subspaces is computationally beneficial.

As a consequence of our model of uncertainty, if the adversarial team is allowed to use inputs of larger magnitude, the robot's victory domain shrinks until some point where the set of admissible initial condition gets empty. Whether this conservativeness is an advantage or a drawback depends on the application scenario: in a safety-critical scenario where deterministic guarantees are mandatory, that is, where no trade-off is allowed between safety and performance, this is obviously a strength. However, if the safety constraints can be somehow relaxed to benefit performance, then this solution may be unsuited because too rigid. Nevertheless, increasing the number of levels for control input quantization enables the robot to capture a wider range of behaviors, which can slightly reduce the conservatism on the robot's victory domain (informally, the robot has "more options"), but this comes naturally at the price of an increased computational cost since the number of HJI PDEs to solve scales linearly with L. Also, since the feedback is set-valued, a particular behavior can be prioritized through the feedback selection rule by optimizing some running cost, such as the distance to the target or to the obstacles. It should also be noted that the constraints do not necessarily describe collision with an external objects but can actually characterize various robot's restrictions (joint limits, robot's self-collision configurations, etc.).

6.2. Future prospects

6.2.1. View-time approximation. Computing the swept volume of arbitrary shapes along arbitrary trajectories is unfortunately not a trivial task.[37] Therefore, we decided to approximate each swept

https://doi.org/10.1017/S0263574719001905 Published online by Cambridge University Press

volume by a finite union of obstacle sets over sub-sampling steps, that is:

$$\mathcal{OBS}^{\downarrow}_{[t_k,t_{k+1}]} = \bigcup_{\tau'} \mathcal{OBS}(\tau')$$

where τ' ranges discretely over $[t_k, t_{k+1}]$. This results naturally in an under-approximation of the true swept volume, but provided that the sub-sampling step is small enough, the finite union gets *big* enough so that it does not jeopardize safety (which is confirmed by our simulations where we choose the number of sub-samples to be 20). A practical implementation maintaining with the theoretical safety guarantees would require ideally a fast $\mathcal{C}$-obstacle construction in order to access $g(., \tau)$ efficiently, or at least a provable over-approximation of the swept volume.

6.2.2. Flexibility of the motion plan. As discussed in the simulation results, one of the consequences of the game-theoretic basis of our approach is the conservativeness of the resulting MP. While this is an obvious advantage in safety-critical situations, this may be too rigid for some applications where flexibility is required, such as in human–robot interaction (HRI) or collaboration (HRC) (as discussed in the introduction of ref. [38], for example). In these cases, flexible motion plans – that is, that are allowed to trade off safety against performance – are then preferable since they allow the robot to exploit more possibilities. Besides, one can "soften" reachability through stochastic definitions of reach-avoid sets[39] that allow to guarantee probabilistic safety up to a certain – arbitrary – level against stochastic uncertainty. Other approaches consist in learning disturbance bounds through experience, so as to reduce conservativeness of the reach-avoid set and thereby to enable a less restrictive behavior of the robot. We believe that a promising lead for this behavior "relaxation" may be the use of reinforcement learning (RL) algorithms: instead of assuming the worst (i.e., other agent's full maliciousness), the robot could learn adversarial policies through interaction with its environment and thereby learn some optimal behavior in a specific context. Conversely, a big challenge in RL is to ensure safety during the learning process while not ruling out huge parts of the state space of which the exploration could benefit performance. This lead has been explored in refs. [40,41].

6.2.3. Experimental validation. Another lead of improvement is naturally the integration of this approach on a real experimental platform. While the theoretical correctness makes no doubt and the approach has been validated through simulations, the next logical step would be to implement it on a real robot in an actual hardware experiment in order to evaluate its practicality and to identify the aspects that must be improved.

7. Conclusion

We presented a comprehensive approach based on SD HJ reachability to address the robust MP problem of a non-point robot operating in adversarial dynamic environments. We constructed offline a look-up table of reach-avoid constructs and designed a set-valued feedback controller guaranteeing robust performance and safety. We also proposed a decomposition of the reach-avoid game combined with projective approximations in order to obtain a conservative solution at a reduced computational cost. Our approach enjoys the flexibility of HJ reachability which does not make any restrictive assumption on the constraints nor on the system's dynamics. Also, the robot's geometry is considered since we reasoned in a configuration space, and the impractical construction of time-varying configuration obstacles has been circumvented by the view-time approximation. Moreover, costly computations are done offline and accurate projective under-approximation can be calculated at a reduced cost when the state's dimension is too high (up to 10 times faster in our 4D numerical example). Finally, this method complies with the SD nature of the dynamics, which is useful for practical implementation. The main challenge is to further reduce the computational cost of solving HJI PDE in order to apply this method to high-dimensional systems and to carry out hardware experiments.

Acknowledgment

This work is supported by National Key R&D Program of China (2018YFB1305902); National Nature Science Foundation of China (61773260, 61590925).

References

1. A. S. Matveev, A. V. Savkin, M. Hoy and C. Wang, *Safe Robot Navigation Among Moving and Steady Obstacle* (Butterworth-Heinemann, Elsevier, Oxford, UK, 2015). ISBN: 978-0-12-803730-0
2. T. Lozano-Pérez, M. T. Mason and R. H. Taylor, "Automatic synthesis of fine-motion strategies for robots," *Int. J. Robot. Res.* **3**(1), 3–24 (1984).
3. J. C. Latombe, "Motion Planning with Uncertainty: On the Preimage Backchaining Approach," **In:** *The Robotics Review* (MIT Press, Cambridge, MA, 1989) pp. 55–69.
4. R. Isaacs, *Differential Games. A Mathematical Theory with Applications to Warfare and Pursuit, Control and Optimization* (John Wiley and Sons, New York, 1965).
5. I. M. Mitchell, "Comparing Forward and Backward Reachability as Tools for Safety Analysis," **In:** *Hybrid Systems: Computation and Control* (Springer, Berlin, Heidelberg, 2007) pp. 428–443.
6. J. Tomlin, J. Lygeros and S. S. Sastry, "Controller Synthesis for Hybrid Systems the Hamilton–Jacobi Approach," *Proceedings of the AAAI Spring Symposium*, Stanford (1999).
7. S. Kaynama, "Scalable Techniques for the Computation of Viable and Reachable Sets," *Ph.D. Dissertation* (The University Of British Columbia, 2012).
8. J. N. Maidens, S. Kaynama, I. M. Mitchell, M. M. Oishi and G. A. Dumont, "Lagrangian methods for approximating the viability kernel in high-dimensional systems," *Automatica* **49**(7), 2017–2029 (2013).
9. C. Le Guernic and A. Girard, "Reachability analysis of linear systems using support functions," *Nonlinear Anal. Hybrid Syst.* **4**(2), 250–262 (2010).
10. O. Maler, "Computing reachable sets: an introduction," pp. 1–8 (2008). Available: https://pdfs.semanticscholar.org/2999/49aef669b547a36c091b768cade091d35532.pdf.
11. I. Mitchell, A. Bayen and C. Tomlin, "A time-dependent Hamiliton–Jacobi formulation of reachable sets for continuous dynamic games," *IEEE Trans. Autom. Control* **50**(7), 947–957 (2005).
12. K. Margellos and J. Lygeros, "Hamilton–Jacobi formulation for reach-avoid differential games," *IEEE Trans. Autom. Control* **56**(8), 1849–1861 (2011).
13. O. Bokanowski and H. Zidani, "Minimal Time Problems with Moving Targets and Obstacles," *IFAC Proceedings Volumes (IFAC-PapersOnline)*, vol. 18, PART 1 (Elsevier, 2011) pp. 2589–2593.
14. O. Bokanowski, P. C. France and A. Desilles, "HJB Approach for Motion Planning and Reachability Analysis," *Proceedings of the 5th International ICST Conference on Performance Evaluation Methodologies and Tools*, vol. 1 (2011) pp. 28–36.
15. J. F. Fisac, M. Chen, C. J. Tomlin and S. S. Sastry, "Reach-Avoid Problems with Time-Varying Dynamics, Targets and Constraints," *Proceedings of the 18th International Conference on Hybrid Systems Computation and Control - HSCC'15* (2015) pp. 11–20.
16. J. Ding and C. J. Tomlin, "Robust Reach-Avoid Controller Synthesis for Switched Nonlinear Systems," *49th IEEE Conference on Decision and Control (CDC)* (2010).
17. I. M. Mitchell, M. Chen and M. Oishi, "Ensuring Safety of Nonlinear Sampled Data Systems Through Reachability," *IFAC Proceedings Volumes (IFAC-PapersOnline)* (2012) pp. 108–114.
18. S. Kleff and N. Li, "A Sampled-Data Hamilton-Jacobi Reachability Approach to Safe and Robust Motion Planning," *Proceedings of the 30th Chinese Control and Decision Conference (2018 CCDC)* (2018) pp. 3386–3392.
19. H. Huang, J. Ding, W. Zhang and C. J. Tomlin, "Automation-assisted capture-the-flag: A differential game approach," *IEEE Trans. Control Syst. Technol.* **23**(3), 1014–1028 (2015).
20. C. Dabadie, S. Kaynama and C. J. Tomlin, "A Practical Reachability-Based Collision Avoidance Algorithm for Sampled-Data Systems: Application to Ground Robots," *IEEE International Conference on Intelligent Robots and Systems* (2014) pp. 4161–4168.
21. I. M. Mitchell, S. Kaynama, M. Chen, and M. Oishi, "Safety preserving control synthesis for sampled data systems," *Nonlinear Anal. Hybrid Syst.* Special Issue related to IFAC Conference on Analysis and Design of Hybrid Systems (ADHS 12), **10**, 1–174 (2013).
22. J. Ding, J. Gillula, H. Huang, M. P. Vitus, W. Zhang and C. J. Tomlin, "Toward Reachability-Based Controller Design for Hybrid Systems in Robotics," *International Symposium on Artificial Intelligence, Robotics and Automation in Space* (2011) pp. 1–9.
23. D. Q. Mayne, "Model predictive control: Recent developments and future promise," *Automatica* **50**(12), 2967–2986 (2014).
24. A. Majumdar, "Funnel libraries for real-time robust feedback motion planning," *Int. J. Robot. Res.* **36**(8), 947–982 (2017).
25. D. Nesic and R. Postoyan, "Nonlinear sampled-data systems," **In:** *Encyclopedia of Systems and Control* (J. Bailleul and T. Samad, eds.) (Springer-Verlag, Berlin, Germany, 2014). Available: https://hal.archives-ouvertes.fr/hal-01104990.
26. S. Bansal, M. Chen, S. Herbert and C. J. Tomlin, "Hamilton-Jacobi Reachability: A Brief Overview and Recent Advances," *56th Conference on Decision and Control (CDC)* (IEEE, 2017).
27. M. Chen, "High Dimensional Reachability Analysis Addressing the Curse of Dimensionality in Formal Verification," *Ph.D. Dissertation* (Electrical Engineering and Computer Sciences, University of California at Berkeley, 2017).
28. M. Chen, S. L. Herbert, M. Vashishtha, S. Bansal and C. J. Tomlin, "Decomposition of reachable sets and tubes for a class of nonlinear systems," *IEEE Trans. Autom. Control* **63**(11), 3675–3688 (2018).
29. J. F. Fisac and S. S. Sastry, "The Pursuit-Evasion-Defense Differential Game in Dynamic Constrained Environments," *2015 IEEE 54th Annual Conference on Decision and Control (CDC)* (2015) pp. 4549–4556.

https://doi.org/10.1017/S0263574719001905 Published online by Cambridge University Press

30. Z. Zhou, J. Ding, H. Huang, R. Takei and C. Tomlin, "Efficient path planning algorithms in reach-avoid problems," *Automatica* **89**, 1–446 (2018).
31. Y. T. Chow, J. Darbon, S. Osher and W. Yin, "Algorithm for overcoming the curse of dimensionality for time-dependent non-convex Hamilton-Jacobi equations arising from optimal control and differential games problems," *J. Sci. Comput.* **73**(2–3), 617–643 (2017).
32. O. Bokanowski, J. Garcke, M. Griebel and I. Klompmaker, "An adaptive sparse grid semi-lagrangian scheme for first order Hamilton-Jacobi bellman equations," *J. Sci. Comput.* **55**(3), 575–605 (2013).
33. W. Kang and L. Wilcox, "A causality free computational method for HJB equations with application to rigid body satellites," *AIAA Guidance, Navigation, and Control Conference* (2015) pp. 1–10.
34. I. M. Mitchell and C. J. Tomlin, "Overapproximating reachable sets by Hamilton-Jacobi projections," *J. Sci. Comput.* **19**(1–3), 323–346 (2003).
35. I. M. Mitchell, A Toolbox of Level Set Methods Version 1.1 Beta (2005). Available: https://www.cs.ubc.ca/~mitchell/ToolboxLS/index.html.
36. N. Y. Ko, B. H. Leet and M. S. Ko, "An approach to robot motion planning for time-varying obstacle avoidance using the view-time concept," *Robotica* **11**(4), 315–327 (1993).
37. K. Abdel-Malek, J. Yang, D. Blackmore and K. I. Joy, "Swept volumes: Foundation, perspectives, and applications," *Int. J. Shape Model.* **12**(1), 87–127 (2006).
38. C. Liu and M. Tomizuka, "Designing the Robot Behavior for Safe Human–Robot Interactions," **In:** *Trends in Control and Decision-Making for Human-Robot Collaboration Systems* (W. Yue, ed.) (Springer International Publishing, Cham, 2017) pp. 241–270.
39. J. Gleason, A. P. Vinod and M. M. K. Oishi, "Underapproximation of Reach-Avoid Sets for Discrete-Time Stochastic Systems via Lagrangian Methods," *IEEE 56th Annual Conference on Decision and Control (CDC)* (2017) pp. 4283–4290.
40. A. Akametalu, S. Kaynama, J. Fisac, M. N. Zeilinger, J. H. Gillula and C. J. Tomlin, "Reachability-Based Safe Learning with Gaussian Processes," *Proceedings of the IEEE Conference on Decision and Control (CDC)*, 02 (2015) pp. 1424–1431.
41. J. H. Gillula and C. J. Tomlin, "Reducing conservativeness in safety guarantees by learning disturbances online: Iterated guaranteed safe online learning," **In:** *RSS* (MIT Press, Cambridge, 2012) pp. 81–88.
42. I. Mitchell, A. M. Bayen and C. J. Tomlin, "Validating a Hamilton–Jacobi Approximation to Hybrid System Reachable Sets," **In:** *Hybrid Systems: Computation and Control* (M. D. Di Benedetto and A. Sangiovanni-Vincentelli, eds.), vol. 2034 (Springer Verlag, Berlin, Heidelberg, 2001) pp. 418–432.

A. Appendix

A.1. Proof of Theorem 1

The proof is done by backward induction on i and is very similar to the one for static environments provided in ref. [22]. First, we prove the equality for $i = N$. By convention, $\mathbb{RA}_0(\mathbb{L}, \mathbb{G}) = \mathbb{L}$ so $\mathbb{RA}_0(\mathbb{L}, \mathbb{G}) = \mathbb{S}^N$. Suppose now that there exist $i \in \{1, \ldots, N-1\}$ such that $\mathbb{RA}_{N-i}(\mathbb{L}, \mathbb{G}) = \mathbb{S}^i$. We will show that $\mathbb{RA}_{N-i+1}(\mathbb{L}, \mathbb{G}) = \mathbb{S}^{i-1}$. By definition

$$\mathbb{S}^{i-1} = \underline{\mathbb{RA}}_1(\mathbb{S}^i, \mathbb{G}) \bigcup \mathbb{S}^i$$

By induction hypothesis

$$\mathbb{S}^{i-1} = \underline{\mathbb{RA}}_1(\mathbb{RA}_{N-i}(\mathbb{L}, \mathbb{G}), \mathbb{G}) \bigcup \mathbb{RA}_{N-i}(\mathbb{L}, \mathbb{G})$$

Let $(q, t_k) \in \mathbb{S}^{i-1}$.

- If $(q, t_k) \in \mathbb{RA}_{N-i}(\mathbb{L}, \mathbb{G})$, then there exists a feedback policy Π_k^m (where $m = \min\{N, k + N - i\}$) steering q safely into $\mathbb{L}$ within $N - i$ steps from step k regardless of the adversarial strategy Γ_k^m so trivially $(q, t_k) \in \mathbb{RA}_{N-i+1}(\mathbb{L}, \mathbb{G})$.
- If $(q, t_k) \in \underline{\mathbb{RA}}_1(\mathbb{RA}_{N-i}(\mathbb{L}, \mathbb{G}), \mathbb{G})$, then there exists a one-step feedback policy π steering q safely into $\mathbb{RA}_{N-i}(\mathbb{L}, \mathbb{G})$ within 1 step from step k regardless of the adversarial strategy γ. Let $q' = q_{cl}(t_{k+1})$, then $(q', t_{k+1}) \in \mathbb{RA}_{N-i}(\mathbb{L}, \mathbb{G})$. Hence, q' can be steered by some feedback policy $\Pi_{k+1}^{m'}$ (where $m' = \min\{N, k + 1 + N - i\}$) safely into $\mathbb{L}$ within $N - i$ steps from step t_{k+1} regardless of the adversarial strategy $\Gamma_{k+1}^{m'}$. Then the concatenated feedback policy $\Pi_k^{m'} = (\pi, \Pi_{k+1}^{m'})$ can steer the q safely into $\mathbb{L}$ within $N - i + 1$ steps from step k despite any strategy $\Gamma_k^{m'}$, so by definition $(q, t_k) \in \mathbb{RA}_{N-i+1}(\mathbb{L}, \mathbb{G})$.

Therefore, $\mathbb{S}^{i-1} \subset \mathbb{RA}_{N-i+1}(\mathbb{L}, \mathbb{G})$.

https://doi.org/10.1017/S0263574719001905 Published online by Cambridge University Press

Conversely, if suppose $(q, t_k) \notin \mathbb{S}^{i-1}$, then $(q, t_k) \notin \underline{\mathbb{RA}}_1(\mathbb{RA}_{N-i}(\mathbb{L}, \mathbb{G}), \mathbb{G})$ and $(q, t_k) \notin \mathbb{RA}_{N-i}(\mathbb{L}, \mathbb{G})$. Since $(q, t_k) \notin \underline{\mathbb{RA}}_1(\mathbb{RA}_{N-i}(\mathbb{L}, \mathbb{G}), \mathbb{G})$, then for any one-step feedback strategy π there exists a disturbance γ such that q cannot safely reach $\mathbb{RA}_{N-i}(\mathbb{L}, \mathbb{G})$ from step k to $k+1$, that is, such that $q' = q_{cl}(t_{k+1}) \notin \mathbb{RA}_{N-i}(\mathbb{L}, \mathbb{G})$. Consequently, for any feedback policy $\Pi_{k+1}^{m'}$ there exists a disturbance $\Gamma_{k+1}^{m'}$ preventing (q', t_{k+1}) from reaching safely $\mathbb{L}$ within $N-i$ steps (where $m' = \min\{N, k+1+N-i\}$). Hence, for any feedback policy $\Pi_k^{m'}$, we can construct by concatenation the disturbance strategy $\Gamma_k^{m'} = (\gamma, \Gamma_{k+1}^{m'})$ which prevents (q, t_k) from reaching $\mathbb{L}$ within $N-i+1$ steps. Therefore, $(q, t_k) \notin \mathbb{RA}_{N-i+1}(\mathbb{L}, \mathbb{G})$. This implies $\overline{\mathbb{S}^{i-1}} \subset \overline{\mathbb{RA}_{N-i+1}(\mathbb{L}, \mathbb{G})}$ hence $\mathbb{RA}_{N-i+1}(\mathbb{L}, \mathbb{G}) \subset \mathbb{S}^{i-1}$. Therefore, $\mathbb{S}^{i-1} = \mathbb{RA}_{N-i+1}(\mathbb{L}, \mathbb{G})$ and the result is true for any $i \in \{0, \dots, N\}$.

A.2. Proof of Theorem 2

We first prove that for any $(\mathbb{S}, \mathbb{G}) \subset (\mathcal{C} \times \{0, \dots, t_N\}) \times (\mathcal{C} \times [0, \dots, t_N))$

$$\underline{\mathbb{RA}}_1(\mathbb{S}, \mathbb{G}) = \bigcup_{j=1}^{L} \bigcup_{k=0}^{N-1} \underline{\mathcal{RA}}_k^j(\mathcal{S}_{k+1}, \mathbb{G}) \times \{t_k\}$$

The proof is straightforward from the definition of this operator: for any $(q, t_{k'}) \in \mathcal{C} \times \{0, \dots, t_N\}$

$$\begin{aligned}
(q, t_{k'}) \in \underline{\mathbb{RA}}_1(\mathbb{S}, \mathbb{G}) &\iff \exists \Pi_{k'}^{k'+1}\ \forall \Gamma_{k'}^{k'+1} \\
&\quad (q_{cl}(t_{k'+1}), t_{k'+1}) \in \mathbb{S} \wedge \forall t \in [t_{k'}, t_{k'+1}]\ (q_{cl}(t), t) \notin \mathbb{G} \\
(q, t_{k'}) \in \underline{\mathbb{RA}}_1(\mathbb{S}, \mathbb{G}) &\iff \exists j \in \{1, \dots, r\}\ \forall \Gamma_{k'}^{k'+1} \\
&\quad x_{cl}(t_{k'+1}) \in \mathcal{S}_{k'+1} \wedge \forall t \in [t_{k'}, t_{k'+1}]\ x_{cl}(t) \notin \mathcal{G}(t) \\
(q, t_{k'}) \in \underline{\mathbb{RA}}_1(\mathbb{S}, \mathbb{G}) &\iff \exists j \in \{1, \dots, r\}\ x \in \underline{\mathcal{RA}}_{k'}^j(\mathcal{S}_{k'+1}, \mathbb{G}) \\
(q, t_{k'}) \in \underline{\mathbb{RA}}_1(\mathbb{S}, \mathbb{G}) &\iff (q, t_{k'}) \in \textstyle\bigcup_{j=1}^{L} \bigcup_{k=0}^{N-1} \underline{\mathcal{RA}}_k^j(\mathcal{S}_{k+1}, \mathbb{G}) \times \{t_k\}
\end{aligned}$$

This alternative expression of $\underline{\mathbb{RA}}_1(\mathbb{S}, \mathbb{G})$ enables to expand the recursion formula of Theorem 1. For any $i \in \{0, \dots, N-1\}$, by writing out $\mathbb{S}^i$, $\mathbb{S}^{i+1}$ explicitly as finite unions and by using the above result, the recursion formula becomes:

$$\underbrace{\bigcup_{k=0}^{N} \mathcal{S}_k^i \times \{t_k\}}_{\mathbb{S}^i} = \underbrace{\bigcup_{j=1}^{L} \bigcup_{k=0}^{N-1} \underline{\mathcal{RA}}_k^j(\mathcal{S}_{k+1}^{i+1}, \mathbb{G}) \times \{t_k\}}_{\underline{\mathbb{RA}}_1(\mathbb{S}^{i+1}, \mathbb{G})} \underbrace{\bigcup_{k=0}^{N} \mathcal{S}_k^{i+1} \times \{t_k\}}_{\mathbb{S}^{i+1}}$$

For a fixed $k \in \{0, \dots, i\}$, the term $\mathcal{S}_k^i$ can be identified as

$$\mathcal{S}_k^i = \mathcal{S}_k^{i+1} \bigcup_{j=1}^{L} \underline{\mathcal{RA}}_k^j(\mathcal{S}_{k+1}^{i+1}, \mathbb{G})$$

If $k \in \{i, \dots, N\}$, it can be easily seen in Definition 1 that $m = \min\{N, k+N-i\} = N$, so in this case

$x \in \mathcal{S}_k^i$	$\iff (q, t_k) \in \mathbb{S}^i$	By definition of $\mathbb{S}^i$
$x \in \mathcal{S}_k^i$	$\iff (q, t_k) \in \mathbb{RA}_{N-i}(\mathbb{L}, \mathbb{G})$	Recursion formula of Theorem 1
$x \in \mathcal{S}_k^i$	$\iff (q, t_k) \in \mathbb{RA}_{N-k}(\mathbb{L}, \mathbb{G})$	From Definition 1 with $m = N$
$x \in \mathcal{S}_k^i$	$\iff (q, t_k) \in \mathbb{S}^k$	Recursion formula of Theorem 2
$x \in \mathcal{S}_k^i$	$\iff q \in \mathcal{S}_k^k$	By definition of $\mathbb{S}^k$

A.3. Decomposition of the reach-avoid game

$\mathcal{G}(t)$ can be decomposed into an n_r-dimensional time-dependent part and an n-dimensional static part as

$$\mathcal{G}(t) = \underbrace{\{q_r \in \mathcal{C}^r \mid \mathcal{R}(q_r) \bigcap \mathcal{G}^{\mathcal{W}}(t) \neq \emptyset\}}_{\mathcal{G}^1(t) \text{ time-dependent (collision with an obstacle at } t)} \bigcup \underbrace{\{q \in \mathcal{C} \mid \mathcal{R}(q_r) \bigcap \mathcal{P}(q_p) \neq \emptyset\}}_{\mathcal{G}^2 \text{ static (capture by the pursuer at } t)}$$

https://doi.org/10.1017/S0263574719001905 Published online by Cambridge University Press

As a straightforward consequence, the CT robust reach-avoid set definition leads to:

$$\underline{\mathcal{RA}}_k^j(\mathcal{S}, \mathbb{G}) = \underline{\mathcal{R}}_k^j(\mathcal{S}) \bigcap (\mathcal{A}_k^j(\mathbb{G}^1) \times \mathcal{C}^p) \bigcap \mathcal{A}_k^j(\mathcal{G}^2)$$

where $\mathbb{G}^1$ denotes the unsafe set in the absence of pursuer, and $\mathcal{A}_k^j(\mathcal{G}^2) = \mathcal{A}_0^j(\mathcal{G}^2)$ for any k ($\mathcal{G}^2$ is static). This decomposition corresponds to the intersection of victory domain of 3 familiar differential games

- Game 1 in $\mathcal{C}^r$ (robot-disturbance): q_r must avoid $\mathcal{G}^1(t)$ for all $t \in [t_k, t_{k+1}]$ despite v_d
- Game 2 in $\mathcal{C}$ (robot-pursuer-disturbance): q must avoid $\mathcal{G}^2$ over $[t_k, t_{k+1}]$ despite v
- Game 3 in $\mathcal{C}$ (robot-pursuer-disturbance): q must reach $\mathcal{S}$ at time t_{k+1} despite v.

of which the optimal costs under constant u_j are

$$\begin{cases} V_1^j(q_r) &= \sup_{\gamma_d} \max_{t_k \le \tau \le t_{k+1}} -g^1(q_r(\tau), \tau) \\ V_2^j(q) &= \sup_{\gamma} \max_{t_k \le \tau < t_{k+1}} -g^2(q(\tau)) \\ V_3^j(q) &= \sup_{\gamma} s(q(t_{k+1})) \end{cases}$$

where $g^1(., \tau)$, $g^2(.)$, and $s(.)$ are LS functions representing, respectively, $\mathcal{G}^1(\tau)$, $\mathcal{G}^2$, and $\mathcal{S}$. The above cost functions are respective LS functions of $\mathcal{A}_k^j(\mathbb{G}^1)$, $\mathcal{A}_k^j(\mathcal{G}^2)$, and $\underline{\mathcal{R}}_k^j(\mathcal{S})$ and they each satisfy a special HJI PDE.[11,15,42]

A.4. Projective over-approximation of unsafe sets

The projection of $q \in \mathcal{C}$ in direction $i \in \{0, \ldots, n\}$ is denoted as $p_i[q] \in \mathbb{R}^{n-1}$, the projection of $\mathcal{A} \subset \mathcal{C}$ in direction i is $p_i[\mathcal{A}] = \{p_i[q] \mid q \in \mathcal{A}\}$, the back-projection of $x \in p_i[\mathcal{C}]$ in direction i is $p_i^{-1}[x] = \{q \in \mathcal{C} \mid p_i[q] = x\}$, the back-projection of $\mathcal{B} \subset p_i[\mathcal{C}]$ in direction i is $p_i^{-1}[\mathcal{B}] = \{q \in \mathcal{C} \mid p_i[q] \in \mathcal{B}\}$, the i^{th} dimension of $\mathcal{C}$ is denoted $\mathcal{C}_i \subset \mathbb{R}$, and for any $\omega \in \mathcal{C}_i$ we denote by $p_i^{-1}[x; \omega]$ the element of $p_i^{-1}[x]$ of which the i^{th} coordinate is equal to ω. The projective dynamics in direction i is defined as

$$f^i : \begin{cases} p_i[\mathcal{C}] \times \mathcal{U} \times (\mathcal{V} \times \mathcal{C}_i) \to & p_i[\mathcal{C}] \\ (x, u, (v, \omega)) \mapsto & p_i[f(p_i^{-1}[x; \omega], u, v)] \end{cases}$$

where the $(v, \omega) \in \mathcal{V} \times \mathcal{C}_i$ is the adversarial input augmented with a disturbance $\omega \in \mathcal{C}_i$, and $p_i^{-1}[x; \omega] \in p_i^{-1}[x]$ with i^{th} component equal to ω. Given $\mathcal{G}, \mathcal{S} \subset \mathcal{C}$, we are interested in calculating the unsafe sets

$$\begin{cases} \mathcal{R}_k^j(p_i[\mathcal{G}]) = \{x \in p_i[\mathcal{C}] \mid \exists \gamma, \psi \ \exists t \in [t_k, t_{k+1}] \ x(t) \in p_i[\mathcal{G}]\} \\ \underline{\mathcal{A}}_k^j(p_i[\mathcal{S}]) = \{x \in p_i[\mathcal{C}] \mid \exists \gamma, \psi, \ x(t_{k+1}) \notin p_i[\mathcal{S}]\} \end{cases}$$

These sets can be characterized by solving two differential games dramatically similar to Game 2 and Game 3 of Appendix A.3 of which the optimal costs under u_j are, respectively, $V_{2,i}^j(x) = \sup_{\gamma,\psi} \max_{t_k \le \tau \le t_{k+1}} -g^{p_i}(x(\tau))$ and $V_{3,i}^j(x) = \sup_{\gamma,\psi} s^{p_i}(q(t_{k+1}))$ where $g^{p_i}(x) = \min_{w \in \mathcal{C}_i} g(p_i^{-1}[x; \omega])$ and $s^{p_i}(x) = \min_{w \in \mathcal{C}_i} s(p_i^{-1}[x; \omega])$. These costs are LS functions of the complements of our sets of interest, namely $\mathcal{R}_k^j(\mathcal{G})$ and $\underline{\mathcal{A}}_k^j(p_i[\mathcal{S}])$ and can be computed from the viscosity solutions $\Phi_{2,i}(., .)$, $\Phi_{3,i}(., .)$ to the following HJI PDEs:

$$\begin{cases} \frac{\partial \Phi}{\partial t}(x, \tau) + \min\{0, H(x, \frac{\partial \Phi}{\partial x}(x, \tau))\} = 0 \\ \Phi(x, t_{k+1}) = g^{p_i}(x) \end{cases} \quad \begin{cases} \frac{\partial \Phi}{\partial t}(x, \tau) + H(x, \frac{\partial \Phi}{\partial x}(x, \tau)) = 0 \\ \Phi(x, t_{k+1}) = s^{p_i}(x) \end{cases}$$

with Hamiltonian $H(x, p) = \inf_{v,\omega} p^T f^i(x, u_j, (v, \omega))$. We assume that ω ranges over $\mathcal{C}_i$ – which is to be necessarily bounded in practice – such that $V_{2,i}^j(x) = \Phi_{2,i}(x, t_k)$ and $V_{3,i}^j(x) = \Phi_{3,i}(x, t_k)$. In theory however, tight bounds for ω's input set can be obtained by intersecting back-projections of HJI PDEs' solutions in multiple dimensions.[34]

https://doi.org/10.1017/S0263574719001905 Published online by Cambridge University Press

A.5. Projective under-approximation of safe sets

We use now the results of Appendix A.4 to under-approximate the victory domains of Game 2 and Game 3. We are looking for some sets $\underline{\mathcal{A}}^{\downarrow}, \underline{\mathcal{R}}^{\downarrow} \subset \mathcal{C}$ such that: $\underline{\mathcal{A}}^{\downarrow} \subset \overline{\mathcal{A}_k^j(\mathcal{G}^2)}$ and $\underline{\mathcal{R}}^{\downarrow} \subset \underline{\mathcal{R}_k^j}(\mathcal{S})$, which is equivalent to find $\mathcal{R}^{\uparrow}$, $\underline{\mathcal{A}}^{\uparrow}$ such that $\overline{\mathcal{A}_k^j(\mathcal{G}^2)} = \mathcal{R}_k^j(\mathcal{G}^2) \subset \mathcal{R}^{\uparrow}$ and $\underline{\mathcal{R}_k^j}(\mathcal{S}) = \underline{\mathcal{A}_k^j}(\mathcal{S}) \subset \underline{\mathcal{A}}^{\uparrow}$. We state now two lemmas:

Lemma 1. $\forall \mathcal{G} \subset \mathcal{C}\ \mathcal{R}_k^j(\mathcal{G}) \subset p_i^{-1}[\mathcal{R}_k^j(p_i[\mathcal{G}])]$

Proof. For convenience, let us denote $\mathcal{G}^{\uparrow} = p_i[\mathcal{G}^2]$. Let first prove $\mathcal{R}_k^j(p_i^{-1}[\mathcal{G}^{\uparrow}]) \subset p_i^{-1}[\mathcal{R}_k^j(\mathcal{G}^{\uparrow})]$. Let us define $q \in \mathcal{R}_k^j(p_i^{-1}[\mathcal{G}^{\uparrow}])$ and show that $p_i[q] \in \mathcal{R}_k^j(\mathcal{G}^{\uparrow})$. There exist $\tilde{v}$ and $t \in [t_k, t_{k+1}]$ such that $q(t) \in p_i^{-1}[\mathcal{G}^{\uparrow}]$, where $q(.)$ denotes the solution to the ODE $\dot{\phi}(t) = f(\phi(t), u_j, v(t))$ over $[t_k, t_{k+1}]$ with initial condition $\phi(t_k) = q$. Let $q_i(.)$ denote the i^{th} component of $q(.)$ and $x(.)$ denote its projection in direction i, namely $x(.) : t \in [t_k, t_{k+1}] \mapsto x(t) = p_i[q(t)]$. We only need to show that $x(.)$ is a possible trajectory under the projected dynamics and that $x(.)$ enters $\mathcal{G}^{\uparrow}$ within $[t_k, t_{k+1}]$. Showing that $x(.)$ is a valid trajectory under f^i is not difficult since

$$
\begin{array}{ll}
\dot{x}(.) = p_i[\dot{q}(.)] = p_i[\dot{q}(.)] & \\
\dot{q}(.) = f(q(.), u_j, v(.)) & \text{and by definition of } q(.) \\
\dot{x}(.) = p_i[f(q(.), u_j, v(.))] & \\
\dot{x}(.) = f^i(x(.), u_j, (v(.), q_i(.))) & \text{by definition of } f^i
\end{array}
$$

Showing that $x(.)$ enters $\mathcal{G}^{\uparrow}$ at time t is obvious since $q(t) \in p_i^{-1}[\mathcal{G}^{\uparrow}]$ implies that $x(t) = p_i[q(t)] \in \mathcal{G}^{\uparrow}$. Therefore, $p_i[q] \in \mathcal{R}_k^j(\mathcal{G}^{\uparrow})$ so $q \in p_i^{-1}[\mathcal{R}_k^j(\mathcal{G}^{\uparrow})]$. Now going back to the main proof, since $\mathcal{G}^2 \subset p^{-1}[\mathcal{G}^{\uparrow}]$):

$$
\begin{array}{ll}
\mathcal{R}_k^j(\mathcal{G}^2) \subset \mathcal{R}_k^j(p_i^{-1}[\mathcal{G}^{\uparrow}]) & \text{complement of Definition 3} \\
\mathcal{R}_k^j(\mathcal{G}^2) \subset p_i^{-1}[\mathcal{R}_k^j(\mathcal{G}^{\uparrow})] & \text{by Lemma 2}
\end{array}
$$

□

Lemma 2. $\forall \mathcal{S} \subset \mathcal{C}\ \underline{\mathcal{A}_k^j}(\mathcal{S}) \subset p_i^{-1}[\underline{\mathcal{A}_k^j}(\overline{p_i[\overline{\mathcal{S}}]})]$

Proof. The proof is very similar to the one of Lemma 1. □

According to Lemmas 1 and 2, $\underline{\mathcal{A}}^{\downarrow}$ and $\underline{\mathcal{R}}^{\downarrow}$ can be chosen, respectively, as complements of $p_i^{-1}[\mathcal{R}_k^j(p_i[\mathcal{G}^2])]$ and $p_i^{-1}[\underline{\mathcal{A}_k^j}(\overline{p_i[\overline{\mathcal{S}}]})]$. Furthermore, these results can be generalized and it can be shown that using collection of projection directions $\mathcal{I} = \{i_1, \ldots, i_m\}$ leads to tighter under-approximations:

$$
\begin{cases}
\bigcap_{i \in \mathcal{I}} \overline{p_i^{-1}[\mathcal{R}_k^j(p_i[\mathcal{G}^2])]} \subset \mathcal{A}_k^j(\mathcal{G}^2) \\
\bigcap_{i \in \mathcal{I}} p_i^{-1}[\underline{\mathcal{A}_k^j}(\overline{p_i[\overline{\mathcal{S}}]})] \subset \underline{\mathcal{R}_k^j}(\mathcal{S})
\end{cases}
$$

https://doi.org/10.1017/S0263574719001905 Published online by Cambridge University Press