

Smart abstraction based on iterative cover and non-uniform cells

Julien Calbert, Lucas N. Egidio, and Raphaël M. Jungers, *Senior Member, IEEE*

Abstract—We propose a multi-scale approach for computing abstractions of dynamical systems, that incorporates both *local* and *global* optimal control to construct a *goal-specific* abstraction. For a local optimal control problem, we not only design the controller ensuring the transition between every two subsets (*cells*) of the state space but also incorporate the volume and shape of these cells into the optimization process. This integrated approach enables the design of *non-uniform* cells, effectively reducing the complexity of the abstraction. These local optimal controllers are then combined into a digraph, which is globally optimized to obtain the entire trajectory. The global optimizer attempts to *lazily* build the abstraction along the optimal trajectory, which is less affected by an increase in the number of dimensions. Since the optimal trajectory is generally unknown in practice, we propose a methodology based on the RRT* algorithm to determine it incrementally. Finally, we provide a tractable implementation of this algorithm for the optimal control of L -smooth nonlinear dynamical systems.

Index Terms—Formal abstractions, optimal control problem, nonlinear discrete-time systems.

I. INTRODUCTION

ABSTRACTION-BASED techniques have been a popular approach to safety-critical control of cyber-physical systems, enabling complex specifications and dynamics to be taken into account in the control design problem. Generally, these methods involve discretizing both the state space and the input space with uniform hyperrectangles [2]. The curse of dimensionality, however, significantly affects the uniform discretization of the entire state space due to the exponential growth of the number of states with respect to the dimension. Additionally, to account for the quantization error between the actual state and the quantized state, it is required to over-approximate the forward image of the cells under constant discretized inputs. One of the main drawbacks of this approach is that, in the absence of incremental stability (δ -GAS) [3], over-approximation increases the level of non-determinism in

the symbolic system, which could result in an intractable or even unsolvable symbolic problem.

In [4], the authors propose to adapt the size of the abstraction gradually but uniformly over the whole state space. In [5], while the authors provide a strategy to lazily (i.e., postponing heavier numerical operations) build an abstraction along the optimal trajectory, the local control design is performed combinatorially on a discretized input set, which is likely a source of non-determinism. More recently, in [6], while the authors propose to design local feedback controllers between nearby cells to eliminate the non-determinism in the abstraction (i.e., the abstraction is a weighted digraph instead of a hypergraph), it relies on a predefined ellipsoidal cover of the entire state space, which suffers from the curse of dimensionality.

In light of these shortcomings, we propose a new approach that leverages the incremental construction of [5] and the optimal control solution proposed in [6] by optimizing not only the transitions but also the positioning and shape of the cells along the optimal trajectory (see Figure 1). Precisely, our solution relies on a Rapidly-Exploring Random Trees (RRT) algorithm that *lazily* constructs the abstraction on the basis of an *ellipsoidal covering* of the state space and a finite set of *local affine controllers*. Unlike [2] and [4], one of the key advantages of our approach is that, instead of discretizing the input space, we use a finite set of local affine controllers as our symbolic input set. For L -smooth nonlinear systems, the combination of linearization and the Lipschitz constant allows us to locally optimize such controllers using semi-definite programming without the need for extensive discretization. Moreover, the proposed approach differs from classic techniques as the partitioning is designed smartly, building the abstraction iteratively, instead of adopting a predefined uniform partition, which is suboptimal and prone to the curse of dimensionality.

The RRT algorithm [7] is a sampling-based path planning algorithm [8], [9] that generates open-loop trajectories for nonlinear systems. Therefore, these techniques are not robust to perturbations and can only provide probabilistic guarantees [10]. In [11], the authors propose a safety-critical path planner based on RRT and relying on control barrier functions. In this same vein, the authors of [12] propose an abstraction-based technique relying on the RRT algorithm limited to deterministic systems with additional stability assumptions (δ -GAS). In [13], the authors propose a reachable set-based RRT planning algorithm that provides a formal guarantee in the presence of bounded perturbations, does not take into account the optimal control problem and is limited to piecewise constant controllers. Furthermore, their framework does not

JC is a FRIA Research Fellow. RJ is a FNRS honorary Research Associate. This project has received funding from the European Research Council (ERC) under *European Union's Horizon 2020 research and innovation programme* under grant agreement No 864017 - L2C. A version of this paper containing the proofs is available in [1]. The results of the numerical experiments presented in Section V are available in the Dionysos Julia package https://github.com/dionysos-dev/Dionysos.jl/releases/tag/2024_LCSS in the subfolder `utils/CDC2024`.

J. Calbert, L. N. Egidio and R. M. Jungers are with the ICTEAM, UCLouvain, 4 Av. G. Lemaître, 1348 Louvain-la-Neuve, Belgium. {julien.calbert, lucas.egidio, raphael.jungers}@uclouvain.be

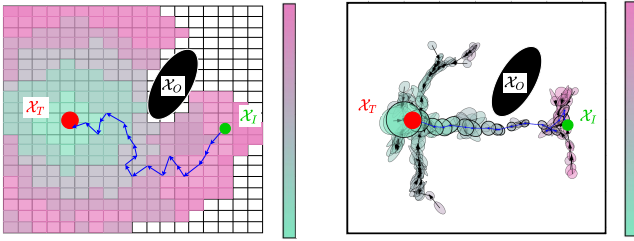


Fig. 1. Comparison between the classic approach and the smart state-feedback abstraction for a planar system with state trajectory (blue line) and value function (color map) obtained for the optimal control problem of departing from \mathcal{X}_I and reaching \mathcal{X}_T while avoiding obstacles \mathcal{X}_O . Left: Classic abstraction approach based on predefined grid discretization of the entire state-space. Non-colored represents a region where no controller could be designed. Right: Smart state-feedback abstraction lazily constructed with locally optimized ellipsoids and controllers.

allow to optimize the shape of new cells. Finally, the authors of [14] propose a forward control strategy for reachability based on ellipsoids, an affine linear approximation with an affine controller via LMIs resolution.

Our work differs from the previously mentioned works by the following points. Firstly, our approach not only provides a controller to solve a specific control problem, but also an abstraction of the original system that can be reused as a basis for solving other optimal control problems. Secondly, our RRT algorithm does not act on states but on sets of states, which makes it possible to build a robust controller that provides formal guarantees even in the presence of noise. Thirdly, in contrast with [14], our RRT algorithm operates in a backward manner. In this configuration, the target set serves as the root of the growing tree, offering the advantages of 1) maximizing the volume of newly added ellipsoids, thereby covering more of the state space with a single cell in the abstraction, and 2) effectively handling obstacles in the state space, as discussed later.

Notation: Given two sets A, B , we define a *single-valued map* as $f : A \rightarrow B$, while a *set-valued map* is defined as $f : A \rightarrow 2^B$, where 2^B is the power set of B , i.e., the set of all subsets of B . The image of a subset $\Omega \subseteq A$ under $f : A \rightarrow 2^B$ is denoted $f(\Omega)$. We define $\mathcal{F}(X, U)$ as the set of single-valued functions $f : X \rightarrow U$. We denote by \mathbb{S}_+^n the set of positive definite matrices of dimension n . Also, $\mathbf{A} \succ \mathbf{0}$ represents that $\mathbf{A} \in \mathbb{S}_+^n$ and $\mathbf{A} \succeq \mathbf{0}$, that $\mathbf{A} \in \partial\mathbb{S}_+^n$, i.e., \mathbf{A} is positive semidefinite. The function $\|\cdot\| : \mathbb{R}^n \rightarrow \mathbb{R}$ denotes the euclidean norm. The Minkowski sum of two sets $A, B \subset \mathbb{R}^n$ is $A \oplus B = \{a + b \mid a \in A, b \in B\}$. Given a bounded set A , we define the *Chebyshev center* of A as $\text{CC}(A) = \text{argmin}_{x \in A} \max_{y \in A} \|x - y\|$. An n -dimensional hyperrectangle of center $c \in \mathbb{R}^n$ and half-lengths $h \in \mathbb{R}_+^n$ is denoted as $\text{H}(c, h) := \{x \in \mathbb{R}^n \mid |x_i - c_i| \leq h_i \text{ for } i = 1, \dots, n\}$. An n -ellipsoid with center $c \in \mathbb{R}^n$ and shape defined by $\mathbf{P} \in \mathbb{S}_+^n$ is denoted as $\text{E}(c, \mathbf{P}) := \{x \in \mathbb{R}^n \mid (x - c)^\top \mathbf{P} (x - c) \leq 1\}$. The n -dimensional Euclidean ball of radius $r > 0$ of center c is denoted as $\text{B}(c, r) := \text{E}(c, r^{-2} \mathbf{I}_n)$.

II. PROBLEM FORMULATION

To introduce the symbolic control formalism needed to support the correctness of the proposed strategy, some definitions of the control framework are required.

Definition 1. A transition control system is a tuple $\mathcal{S} := (\mathcal{X}, \mathcal{U}, F)$ where $\mathcal{X} \subseteq \mathbb{R}^{n_x}$ and $\mathcal{U} \subseteq \mathbb{R}^{n_u}$ are respectively the set of states and inputs and the set-valued map $F : \mathcal{X} \times \mathcal{U} \rightarrow 2^{\mathcal{X}}$ where $F(x, u)$ give the set of states that may be reached from a given state x under a given input u .

A tuple $(x, u) \in \mathcal{X}^{[0, T]} \times \mathcal{U}^{[0, T-1]}$ is a trajectory of length $T \in \mathbb{N} \cup \{\infty\}$ starting at $x(0) \in \mathcal{X}$ of the system $\mathcal{S} = (\mathcal{X}, \mathcal{U}, F)$ if $\forall k \in [0, T-1]: u(k) \in \mathcal{U}(x(k))$ and $x(k+1) \in F(x(k), u(k))$. The set of trajectories of \mathcal{S} is called the *behavior* of \mathcal{S} , denoted $\mathcal{B}(\mathcal{S})$.

We consider *static* controllers, where the set of control inputs enabled at a given state depends only on that state.

Definition 2. We define a static controller for a system $\mathcal{S} = (\mathcal{X}, \mathcal{U}, F)$ valid on a subset $\mathcal{X}_v \subseteq \mathcal{X}$ as a single-valued map $\mathcal{C} : \mathcal{X}_v \rightarrow \mathcal{U}$ such that $\forall x \in \mathcal{X}_v : \mathcal{C}(x) \in \mathcal{U}(x)$. We define the controlled system, denoted as $\mathcal{C} \times \mathcal{S}$, as the transition system characterized by the tuple $(\mathcal{X}_v, \mathcal{U}, F_{\mathcal{C}})$ where $x' \in F_{\mathcal{C}}(x, u) \Leftrightarrow (u = \mathcal{C}(x) \wedge x' \in F(x, u))$.

Given a system $\mathcal{S} = (\mathcal{X}, \mathcal{U}, F)$ and sets $\mathcal{X}_I, \mathcal{X}_T, \mathcal{X}_O \subseteq \mathcal{X}$, a *reach-avoid* specification is defined as

$$\Sigma = \{(x, u) \in (\mathcal{X} \times \mathcal{U})^\infty \mid x(0) \in \mathcal{X}_I \Rightarrow \exists N \in \mathbb{Z}_+ : (x(N) \in \mathcal{X}_T \wedge \forall k \in [0, N[: x(k) \notin \mathcal{X}_O])\}, \quad (1)$$

which enforces that all states in the initial set \mathcal{X}_I will reach the target \mathcal{X}_T in finite time while avoiding obstacles in \mathcal{X}_O . We use the abbreviated notation $\Sigma = [\mathcal{X}_I, \mathcal{X}_T, \mathcal{X}_O]$ to denote the specification (1). A system \mathcal{S} together with a specification Σ constitute a *control problem* (\mathcal{S}, Σ) . Additionally, a controller \mathcal{C} is said to *solve* the control problem (\mathcal{S}, Σ) if $\mathcal{X}_I \subseteq \mathcal{X}_v$ and $\mathcal{B}(\mathcal{C} \times \mathcal{S}) \subseteq \Sigma$. We consider an *optimal control problem* whose goal is to design a controller \mathcal{C} enforcing reach-avoid specification Σ while minimizing a *cost function*

$$\mathcal{L}(\mathcal{C}) = \sup_{x_0 \in \mathcal{X}_I} l_{\mathcal{C}}(x_0) \quad (2)$$

where $l_{\mathcal{C}}(x) = 0$ for $x \in \mathcal{X}_T$ and

$$l_{\mathcal{C}}(x) = \mathcal{J}(x, \mathcal{C}(x)) + \sup_{x' \in F(x, \mathcal{C}(x))} l_{\mathcal{C}}(x') \text{ for } x \in \mathcal{X}_v \setminus \mathcal{X}_T,$$

where $\mathcal{J} : \mathcal{X} \times \mathcal{U} \rightarrow \mathbb{R}_+$ is a given *stage cost function*.

In the context of optimal control, we recall the notion of *value function* ([6, Definition 3]), which provides a guaranteed cost for any closed-loop trajectory starting in a given subset $\mathcal{X}_v \subseteq \mathcal{X}$ which decreases with each time step.

Definition 3. A function $v : \mathcal{X} \rightarrow \mathbb{R}$ is a value function with stage cost $\mathcal{J}(x, u)$ for system $\mathcal{S} = (\mathcal{X}, \mathcal{U}, F)$ in $\mathcal{X}_v \subseteq \mathcal{X}$ if $v(x)$ is bounded from below within \mathcal{X}_v and for all $x \in \mathcal{X}_v$ there exists $u \in \mathcal{U}(x)$ fulfilling the Bellman inequality

$$v(x) \geq \mathcal{J}(x, u) + \sup_{x' \in F(x, u)} v(x'). \quad (3)$$

Our abstraction-based approach relies on the notion of *state-feedback abstraction* [6, Definition 1] which is a specific instance of *alternating simulation relation* [15, Definition 4.19] (see [6, Lemma 1]).

Definition 4. Consider a system $\mathcal{S} = (\mathcal{X}, \mathcal{U}, F)$. A state-feedback abstraction $\tilde{\mathcal{S}}$ of \mathcal{S} is a system $\tilde{\mathcal{S}} = (\tilde{\mathcal{X}}, \tilde{\mathcal{U}}, \tilde{F})$ that

satisfies the following conditions: $\tilde{\mathcal{X}} \subseteq 2^{\mathcal{X}}$, $\tilde{\mathcal{U}} \subseteq \mathcal{F}(\mathcal{X}, \mathcal{U})$ and $\forall \xi \in \tilde{\mathcal{X}}, \forall \kappa \in \tilde{\mathcal{U}}: \tilde{F}(\xi, \kappa) = \{\xi_+\}$ where

$$\xi_+ \supseteq \{x' \in F(x, \kappa(x)) \mid x \in \xi\}. \quad (4)$$

A value function $v: \mathcal{X} \rightarrow \mathbb{R}$ with stage cost \mathcal{J} for \mathcal{S} can be derived from a value function $\tilde{v}: \tilde{\mathcal{X}} \rightarrow \mathbb{R}$ for its state-feedback abstraction $\tilde{\mathcal{S}}$ [6, Theorem 1]

$$v(x) = \min\{\tilde{v}(\xi) : \xi \in \tilde{\mathcal{X}}, x \in \xi\} \quad (5)$$

with any cost function $\tilde{\mathcal{J}}$ that verifies

$$\tilde{\mathcal{J}}(\xi, \kappa) \geq \mathcal{J}(x, \kappa(x)), \quad \forall x \in \xi. \quad (6)$$

Based on the fact that any state-feedback abstraction $\tilde{\mathcal{S}}$ is a deterministic system by definition and that the set $\tilde{\mathcal{U}}$ contains static (memoryless) state-feedback controllers, the state-feedback abstraction allows a specific concretization scheme, presented in the following proposition.

Proposition 1. *Let \mathcal{S} be a system and $\tilde{\mathcal{S}}$ a corresponding state-feedback abstraction. Additionally, consider specifications $\Sigma = [\mathcal{X}_I, \mathcal{X}_T, \mathcal{X}_O]$ with stage cost \mathcal{J} for \mathcal{S} and $\tilde{\Sigma} = [\xi_I, \xi_T, \xi_O]$ with stage cost $\tilde{\mathcal{J}}$ for $\tilde{\mathcal{S}}$ such that*

$$\mathcal{X}_I \subseteq \xi_I, \quad \xi_T \subseteq \mathcal{X}_T, \quad \mathcal{X}_O \subseteq \xi_O \quad (7)$$

and satisfying (6). If $\tilde{\mathcal{C}}: \tilde{\mathcal{X}}_v \rightarrow \tilde{\mathcal{U}}$ with $\tilde{\mathcal{X}}_v = \tilde{\mathcal{X}}$ is a controller that solves the control problem $(\tilde{\mathcal{S}}, \tilde{\Sigma})$, then for any value function \tilde{v} of $\tilde{\mathcal{S}}$, the controller $\mathcal{C}: \mathcal{X}_v \rightarrow \mathcal{U}$, defined as

$$\mathcal{C}(x) = \kappa(x), \quad \kappa = \tilde{\mathcal{C}}(\xi^*), \quad \xi^* = \operatorname{argmin}\{\tilde{v}(\xi) : \xi \in \tilde{\mathcal{X}}, x \in \xi\} \quad (8)$$

with $\mathcal{X}_v = \cup_{\xi \in \tilde{\mathcal{X}}_v} \xi$, solves the control problem (\mathcal{S}, Σ) . In addition, the value function v of \mathcal{S} defined according to (5) provides an upper bound on the cost function (2), i.e., $v(x_0) \geq \mathcal{L}(\mathcal{C})$ for any $x_0 \in \mathcal{X}_I$.

In the next section, we present a method to solve the following problem: to lazily build a state-feedback abstraction $\tilde{\mathcal{S}}$ of the original system, addressing an optimal control problem with reach-avoid specification Σ and cost function \mathcal{L} .

III. RRT*-BASED ABSTRACTION

The nonlinear nature of the system and of the reach-avoid specification makes it extremely difficult to solve the optimal control problem directly. Therefore, to approximately solve this problem our abstraction-based approach will split it up into several convex subproblems. The approach involves using a RRT (Rapidly-Exploring Random Trees) algorithm that grows a state-feedback abstraction $\tilde{\mathcal{S}}$ of \mathcal{S} with an underlying *tree* structure from the target set to the initial set. The nodes of this tree represent abstract states, each of which is a *set* $\xi \in 2^{\mathcal{X}}$ and transitions from child nodes to their parents are handled by *local state-feedback controllers* $\kappa \in \mathcal{F}(\mathcal{X}, \mathcal{U})$. While the strategy presented is theoretically applicable to any set templates for $\tilde{\mathcal{X}}$ and function templates for $\tilde{\mathcal{U}}$, we propose a practical implementation (Algorithm 1) using *ellipsoids* and *affine functions* templates, respectively, to take advantage of the power of LMIs. The functions that implement Algorithm 1 are described as follows:

- `getAbsSpecification` (Σ) returns abstract specification, i.e., ellipsoids ξ_I, ξ_T, ξ_O satisfying (7).
- `getConState` (\mathcal{X}) returns a candidate state $c \in \mathcal{X}$.
- `getKClosestAbsStates` ($\tilde{\mathcal{X}}, \mathcal{A}, K$) returns the K abstract states of $\tilde{\mathcal{X}}$ closest to the set \mathcal{A} according to Euclidean distance.
- `solveLocalProblem` (c, ξ_+) returns $\xi = E(c, \mathbf{P})$, a controller κ such that $\forall x \in \xi: F(x, \kappa(x)) \subseteq \xi_+$ and $\kappa(\xi) \subseteq \mathcal{U}$, and an upper bound $\tilde{\mathcal{J}}$ on the transition cost from ξ to ξ_+ , i.e., $\tilde{\mathcal{J}} \geq \max_{x \in \xi} \mathcal{J}(x, \kappa(x))$.
- `handleObstacles` (ξ, ξ_O) given $\xi = E(c, \gamma \mathbf{P})$, returns a new ellipsoid $\xi_s = E(c, \gamma \mathbf{P})$ where $\gamma \geq 1$ is the smallest value such that ξ_s does not intersect any obstacles, i.e. $\xi_s \cap \xi_O = \emptyset$.
- `getAbsValueFunction` ($\tilde{\mathcal{S}}, \xi_T$) returns a value function \tilde{v} for $\tilde{\mathcal{S}}$ such that $\tilde{v}(\xi_T) = 0$ and which is strictly positive for the other abstract states.

Algorithm 1: Lazy construction of an ellipsoid-based abstraction $\tilde{\mathcal{S}}$ of $\mathcal{S} = (\mathcal{X}, \mathcal{U}, F)$ and of an abstract value function \tilde{v} that solves the specification $\Sigma = [\mathcal{X}_I, \mathcal{X}_T, \mathcal{X}_O]$.

```

1  $\tilde{\Sigma} \leftarrow \text{getAbsSpecification}(\Sigma)$ ;
2  $\xi_I, \xi_T, \xi_O \leftarrow \tilde{\Sigma}$ ;
3  $\tilde{\mathcal{X}} \leftarrow \{\xi_T\}$ ;
4 while not exists  $\xi \in \tilde{\mathcal{X}}: \xi_I \subseteq \xi$  do
5    $c \leftarrow \text{getConState}(\mathcal{X})$ ;
6    $\xi_+ \leftarrow \text{getKClosestAbsStates}(\tilde{\mathcal{X}}, \{c\}, K=1)$ ;
7    $\xi, \kappa, \tilde{\mathcal{J}} \leftarrow \text{solveLocalProblem}(c, \xi_+)$ ;
8   if not feasible then return to Line 5.;
9    $\xi_s \leftarrow \text{handleObstacles}(\xi, \xi_O)$ ;
10   $\tilde{\mathcal{X}}, \tilde{\mathcal{U}} \leftarrow \tilde{\mathcal{X}} \cup \{\xi_s\}, \tilde{\mathcal{U}} \cup \{\kappa\}$ ;
11   $\tilde{F}(\xi_s, \kappa), \tilde{\mathcal{J}}(\xi_s, \kappa) \leftarrow \{\xi_+\}, \tilde{\mathcal{J}}$ ;
12   $\tilde{\mathcal{S}} \leftarrow (\tilde{\mathcal{X}}, \tilde{\mathcal{U}}, F)$ ;
13   $\tilde{v} \leftarrow \text{getAbsValueFunction}(\tilde{\mathcal{S}}, \xi_T)$ ;
14 return  $\tilde{\mathcal{S}}, \tilde{v}$ ;
```

As mentioned in the introduction, our RRT algorithm works backwards in the sense that the abstraction is initialized with the target set ξ_T , and when designing a new transition, the optimized cells are preceding ones (in terms of the controlled system trajectory). This allows us to maximize the volume of the preceding ellipsoid, covering a larger portion of the state space with a single abstract state—a feature not achievable in the forward approach, which focuses on minimizing the volume of the destination ellipsoid [14]. In addition, the backward approach makes it possible to decouple the design of the preceding ellipsoid ξ and the controller κ from obstacle handling. Indeed, when a preceding ellipsoid ξ returned by `solveLocalProblem` intersects an obstacle, it can be shrunk into $\xi_s \subseteq \xi$ and the same controller κ still ensures a transition to ξ_+ . This decoupling is not feasible in the forward approach.

The function `getConState` is a *heuristic* used to generate a concrete state from which a new ellipsoid is added to the abstraction. Although the specific implementation of this function does not affect the correctness of Algorithm 1, it can be used to guide the exploration of the state space and the

construction of the abstraction in order to reduce the number of abstract states. For example, a simple but effective strategy involves sampling points predominantly in the direction of the initial set \mathcal{X}_I .

The following theorem guarantees the validity of the approach.

Theorem 1. *Consider the system $\tilde{\mathcal{S}} = (\tilde{\mathcal{X}}, \tilde{\mathcal{U}}, \tilde{F})$ and its value function \tilde{v} generated by Algorithm 1. Then 1) $\tilde{\mathcal{S}}$ is a state-feedback abstraction of \mathcal{S} ; 2) there exists a controller $\tilde{\mathcal{C}}$ that solves $(\tilde{\mathcal{S}}, \tilde{\Sigma})$; 3) the controller \mathcal{C} defined according to (8) solves (\mathcal{S}, Σ) ; 4) the function v derived from \tilde{v} according to (5) is a value function for \mathcal{S} with $\mathcal{X}_v = \cup_{\xi \in \tilde{\mathcal{X}}} \xi$.*

Proof. Due to space constraints, we will demonstrate the first item and refer to the extended version of the paper for the proof of the remaining points. 1) The transition map \tilde{F} is constructed according to $\tilde{F}(\xi_s, \kappa) = \{\xi_+\}$, where `solveLocalProblem` guarantees that $\forall x \in \xi : F(x, \kappa(x)) \subseteq \xi_+$, and `handleObstacles` ensures that $\xi_s \subseteq \xi$, which (always) implies $\forall x \in \xi_s : F(x, \kappa(x)) \subseteq \xi_+$. Consequently, \tilde{F} satisfies the condition (4). \square

The abstract value function \tilde{v} is derived in `getAbsValueFunction` by computing the shortest path to ξ_T , which can be done efficiently since ξ_T is the root of a tree. The optimization problem in `handleObstacles` and the inclusion test in Line 4 of Algorithm 1 can be efficiently addressed by solving a convex scalar optimization problem, as outlined in [16, Corollary 2] and [16, Algorithm 1], respectively. Finally, the function `solveLocalProblem` can be implemented efficiently by solving a convex optimization problem as described in the following section.

IV. LOCAL CONTROLLER DESIGN

In this section, we provide a practical and efficient implementation of the function `solveLocalProblem` in Line 7 of Algorithm 1 for the following general class of systems.

We consider nonlinear discrete-time system $\mathcal{S} = (\mathcal{X}, \mathcal{U}, F)$ with bounded disturbances, i.e.,

$$F(x, u) = \{f(x, u, w) \mid w \in \mathcal{W}\} \quad (9)$$

where $\mathcal{X} \subseteq \mathbb{R}^{n_x}$, $\mathcal{U} \subseteq \mathbb{R}^{n_u}$, $f : \mathcal{X} \times \mathcal{U} \times \mathcal{W} \rightarrow \mathcal{X}$ is a continuous nonlinear function whose Jacobian is globally Lipschitz continuous, and $\mathcal{W} \subseteq \mathbb{R}^{n_w}$ is a bounded polytopic set. The control input takes values in the intersection of (possibly degenerated) ellipsoids

$$\mathcal{U} = \bigcap_{k=1, \dots, N_u} \mathcal{U}_k \quad (10)$$

with $\mathcal{U}_k = \{u \in \mathbb{R}^{n_u} : \|\mathbf{U}_k u\| \leq 1\}$ for some matrix \mathbf{U}_k of appropriate dimensions. The exogenous input takes values inside the convex hull of a set of points

$$\mathcal{W} = \text{co}\{w_1, \dots, w_{N_w}\} \quad (11)$$

with $0 \in \mathcal{W}$ and introduces a level of non-determinism to the system, which can either capture non-modeled behaviors

or adversarial disturbances. We consider a general quadratic stage cost function of the form

$$\mathcal{J}(x, u) = \begin{pmatrix} x^\top & u^\top & 1 \end{pmatrix} \mathbf{Q} \begin{pmatrix} x^\top & u^\top & 1 \end{pmatrix}^\top \quad (12)$$

defined for some given matrix $\mathbf{Q} \succ 0$.

Therefore, the objective of this section is to design a controller κ to map all the states of $\xi \in 2^{\mathcal{X}}$ into $\xi_+ \in 2^{\mathcal{X}}$ despite exogenous noise, i.e.,

$$\forall x \in \xi \quad \forall w \in \mathcal{W} : f(x, \kappa(x), w) \in \xi_+ \quad (13)$$

and such that the inputs of the closed loop system lie within the admissible input set \mathcal{U} , i.e.,

$$\kappa(\xi) \subseteq \mathcal{U}. \quad (14)$$

A. Specific approximation scheme

Since optimizing directly on the nonlinear dynamics f is a challenging problem, we rely on the linearized function \tilde{f} around a point $\bar{p} = (\bar{x}, \bar{u}, \bar{w}) \in \mathcal{X} \times \mathcal{U} \times \mathcal{W}$:

$$\tilde{f}(x, u, w) = \mathbf{A}x + \mathbf{B}u + \mathbf{E}w + g \quad (15)$$

where $g = f(\bar{x}, \bar{u}, \bar{w}) - \mathbf{A}\bar{x} - \mathbf{B}\bar{u} - \mathbf{E}\bar{w}$.

We can derive a component-wise bound on the linearization error [17, Lemma 1.2.3], i.e., $\forall (x, u, w) \in \mathcal{X} \times \mathcal{U} \times \mathcal{W}$:

$$f(x, u, w) \in \{\tilde{f}(x, u, w)\} \oplus \Omega(x, u, w) \quad (16)$$

where $\Omega(x, u, w) = \text{H}(0, \frac{1}{2}Lr_{\bar{p}}(x, u, w)^2)$ with $L \in \mathbb{R}^{n_x}$ representing the vector of Lipschitz constants of component functions f_i and

$$r_{\bar{p}}(x, u, w)^2 = \|x - \bar{x}\|^2 + \|u - \bar{u}\|^2 + \|w - \bar{w}\|^2. \quad (17)$$

Therefore, to enforce the condition (13), we can impose the following stronger condition

$$\forall x \in \xi \quad \forall w \in \mathcal{W} : \{\tilde{f}(x, \kappa(x), w)\} \oplus \Omega(x, \kappa(x), w) \subseteq \xi_+. \quad (18)$$

In order to keep the future optimization problem tractable (convex), we use a common error bound corresponding to the "worst case" to impose the more conservative condition

$$\forall x \in \xi \quad \forall w \in \mathcal{W} : \{\tilde{f}(x, \kappa(x), w)\} \oplus \mathcal{H}_r \subseteq \xi_+, \quad (19)$$

with $\mathcal{H}_r := \text{H}(0, \frac{1}{2}Lr^2)$ and $r := \max_{(x, w) \in \xi \times \mathcal{W}} r_{\bar{p}}(x, \kappa(x), w)$.

Condition (19) can be reformulated as

$$q(\xi) \oplus \mathbf{E}\mathcal{W} \oplus \mathcal{H}_r \subseteq \xi_+ \quad (20)$$

where $q(x) := \tilde{f}(x, \kappa(x), \bar{w})$. This provides a geometric interpretation of the different sources of conservatism. The term $q(\xi)$ corresponds to the closed-loop image of ξ in the linearized model with nominal noise (\bar{w}), the second term $\mathbf{E}\mathcal{W}$ corresponds to the exogenous noise, and the third term \mathcal{H}_r is responsible for the linearization error.

We first propose the following lemma, which provides sufficient conditions to design κ satisfying (13).

Lemma 1. *Let a system (9) with exogenous input set (11), a controller κ , two sets ξ and ξ_+ and a linearization point \bar{p} . If the following conditions are satisfied*

$$\xi \subseteq \text{B}(\bar{x}, \sqrt{\delta_X}), \quad (21)$$

$$\kappa(\xi) \subseteq \text{B}(\bar{u}, \sqrt{\delta_U}), \quad (22)$$

$$\forall x \in \xi \forall w \in \mathcal{W} : \{\tilde{f}(x, \kappa(w), w)\} \oplus \mathcal{H}_r \subseteq \xi_+, \quad (23)$$

with $r^2 = \delta_X + \delta_U + \delta_W$ for some $\delta_X, \delta_U \geq 0$ and $\delta_W := \max_{w \in \mathcal{W}} \|w - \bar{w}\|^2$, then (13) holds.

The linearization point that minimizes the linearization error (r) is determined by $\bar{p}^* = \text{CC}(\xi \times \kappa(\xi) \times \mathcal{W})$, the Chebyshev center of $\xi \times \kappa(\xi) \times \mathcal{W}$. However, given that the controller κ is part of the optimization process, we opt for the choice: $\bar{p} = \text{CC}(\xi \times \mathcal{U} \times \mathcal{W})$ which represents the optimal choice for minimizing the linearization error when κ is arbitrary.

B. Discretization and controller templates

In this paper, we discretize the state space using ellipsoids, i.e., $\xi = \text{E}(c, \mathbf{P})$, $\xi_+ = \text{E}(c_+, \mathbf{P}_+)$ with $c, c_+ \in \mathbb{R}^{n_x}$ and $\mathbf{P}, \mathbf{P}_+ \in \mathbb{S}_+^{n_x}$, and we consider affine controllers of the form

$$\kappa(x) = \mathbf{K}(x - c) + l \quad (24)$$

where $\mathbf{K} \in \mathbb{R}^{n_u \times n_x}$ and $l \in \mathbb{R}^{n_u}$.

Therefore, in this setting, we obtain $\bar{x} = \text{CC}(\xi) = c$, and, for the sake of clarity throughout the rest of the paper, we assume that $\bar{w} = \text{CC}(\mathcal{W}) = 0$.

The following theorem provides conditions based on Lemma 1 to guarantee the existence of a valid controller satisfying (13) and (14).

Theorem 2. *The system (9) with control input set (10) and exogenous input set (11) under the constrained affine control law (24) satisfies the condition that $x(k+1) \in \xi_+ = \text{E}(c_+, \mathbf{P}_+)$ for all $x(k) \in \xi = \text{E}(c, \mathbf{P})$, if, given the linearized system (15) around the point $\bar{p} = (c, \bar{u}, 0)$ with $\bar{u} \in \mathcal{U}$, there exist $\mathbf{L} \in \mathbb{S}_+^{n_x}$, $\mathbf{F} \in \mathbb{R}^{n_x \times n_u}$ and scalars $\delta_X \geq 0, \delta_U \geq 0, \phi \geq 0, \beta_{ij} \geq 0, \tau_k \geq 0$ such that*

$$\begin{pmatrix} \mathbf{I} & \mathbf{L} \\ \bullet & \delta_X \mathbf{I} \end{pmatrix} \succeq \mathbf{0}, \begin{pmatrix} \phi \mathbf{I} & 0 & \mathbf{F}^\top \\ \bullet & \delta_U - \phi & (l - \bar{u})^\top \\ \bullet & \bullet & \mathbf{I} \end{pmatrix} \succeq \mathbf{0}, \quad (25)$$

$$\begin{pmatrix} \beta_{ij} \mathbf{I} & 0 & (\mathbf{A}\mathbf{L} + \mathbf{B}\mathbf{F})^\top \\ \bullet & 1 - \beta_{ij} & \mu^\top + V_i^\top + (\mathbf{E}w_j)^\top \\ \bullet & \bullet & \mathbf{P}_+^{-1} \end{pmatrix} \succeq \mathbf{0},$$

$$\text{for } i \in \{1, \dots, 2^{n_x}\}, j \in \{1, \dots, N_w\} \quad (26)$$

$$\begin{pmatrix} \tau_k \mathbf{I} & 0 & \mathbf{F}^\top \mathbf{U}_k^\top \\ \bullet & 1 - \tau_k & l^\top \mathbf{U}_k^\top \\ \bullet & \bullet & \mathbf{I} \end{pmatrix} \succeq \mathbf{0}, \text{ for } k \in \{1, \dots, N_u\} \quad (27)$$

where $\mu = g + \mathbf{A}c + \mathbf{B}l - c_+$, V_i for $i = 1, \dots, 2^{n_x}$ are the vertices of $\text{H}(0, \frac{1}{2}Lr^2)$ with $r^2 = \delta_X + \delta_U + \delta_W$ and $\delta_W = \max_{i=1, \dots, N_w} \{\|w_i\|^2\}$. From which we have

$$\mathbf{P} = \mathbf{L}^{-2}, \mathbf{K} = \mathbf{F}\mathbf{L}^{-1}.$$

The extension presented in Theorem 2 builds upon [6, Theorem 2] by incorporating the linearization error while

maintaining convexity when considering the parameters of the initial ellipsoids ξ as optimization variables. It is worth noting that in [6, Theorem 2], the problem is no longer an LMI if we consider \mathbf{P} as a variable. However, as demonstrated by Theorem 2, the optimization problem can be rendered convex through a congruent transformation when considering the square root of \mathbf{P}^{-1} as the variable of interest. If f is an affine function, the LMIs from Theorem 2 can be simplified to the ones presented in [6, Theorem 2] by applying a congruent transformation.

We can visualize the terms of the Minkowski sum in (20) for our specific setting in Figure 2.

C. Optimal control

Since our aim is not only to design κ to minimize a cost function, but also to optimize the shape \mathbf{P} of the starting ellipsoid ξ , we introduce a performance objective that involves minimizing a cost function (12) and maximizing the hypervolume of the preceding ellipsoid ξ . As a reminder [18, Section 2.2.4], we can maximize the volume of an ellipsoid $\xi = \text{E}(c, \mathbf{P})$ by minimizing the convex function $-\log(\det(\mathbf{L}))$ where \mathbf{L} is the square root of \mathbf{P}^{-1} , i.e., $\mathbf{P}^{-1} = \mathbf{L}^2$.

The correctness and efficiency of the global RRT* algorithm (Algorithm 1) are guaranteed by the following result.

Corollary 1. *For a given point c and target set $\xi_+ = \text{E}(c_+, \mathbf{P}_+)$, with a stage cost function \mathcal{J} (12) where $\mathbf{Q} = \mathbf{S}^\top \mathbf{S}$ and $\lambda \in [0, 1]$, the solution of the convex optimization problem*

$$\inf_{\substack{\mathbf{L} \succ \mathbf{0}, \mathbf{F}, l, \delta_X \geq 0, \delta_U \geq 0, \\ \phi \geq 0, \beta_{ij} \geq 0, \tau_k \geq 0, \gamma \geq 0, \tilde{\mathcal{J}}} } \lambda \tilde{\mathcal{J}} + (1 - \lambda)(-\log(\det(\mathbf{L}))) \quad (28)$$

s.t. (25), (26), (27),

$$\begin{pmatrix} \gamma \mathbf{I} & 0 & [\mathbf{L}, \mathbf{F}^\top, 0] \mathbf{S}^\top \\ \bullet & \tilde{\mathcal{J}} - \gamma & [c^\top, l^\top, 1] \mathbf{S}^\top \\ \bullet & \bullet & \mathbf{I} \end{pmatrix} \succeq \mathbf{0} \quad (29)$$

satisfies

$$\tilde{\mathcal{J}} \geq \max_{x \in \xi} \mathcal{J}(x, \kappa(x)) \quad (30)$$

where the controller $\kappa(x) := \mathbf{K}(x - c) + l \in \mathcal{U}$ with $\mathbf{K} = \mathbf{F}\mathbf{L}^{-1}$ ensures a transition from $\xi = \text{E}(c, \mathbf{P})$ with $\mathbf{P} = \mathbf{L}^{-2}$ to ξ_+ , as in Theorem 2.

The parameter λ governs the weight assigned to each criterion and also represents an exploitation/exploration trade-off, as maximizing the volume allows us to explore a larger portion of the state space, while minimizing the cost provides better (finer) solutions for already explored areas. Because of that, λ can be chosen adaptively throughout the execution. A comprehensive study on how to fine-tune this meta-parameter is beyond the scope of this paper. However, a basic approach is to start with a smaller λ to encourage designing larger cells (potentially leading faster to a feasible controller), and then increase its value to favor smaller cells (with eventually lower transition costs).

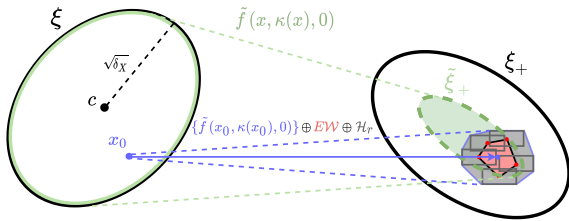


Fig. 2. Local transition from ellipsoid ξ to ellipsoid ξ_+ . The affine controller κ enforces that $\forall x_0 \in \xi$ and $\forall w \in \mathcal{W}$: $f(x_0, \kappa(x_0), w) \in \{\tilde{f}(x_0, \kappa(x_0), 0)\} \oplus \mathcal{E}\mathcal{W} \oplus \mathcal{H}_r \subseteq \xi_+$ where \mathcal{W} is the polytopic noise and $\mathcal{H}_r = \mathbf{H}(0, \frac{1}{2}Lr^2)$ is the noise resulting from the linearization (19) where $r^2 = \delta_X + \delta_U + \delta_W$. The set $\tilde{\xi}_+ = q(\xi)$ with $q(x) = \tilde{f}(x, \kappa(x), 0)$ (see (20)) is an ellipsoid since an affine transformation of an ellipsoid is also an ellipsoid and the set $\mathcal{E}\mathcal{W}$ is a polytope since the linear transformation of a polytope is a polytope.

V. NUMERICAL EXPERIMENTS

We consider the nonlinear dynamical system (9) given by

$$f(x, u, w) = \begin{pmatrix} 1.1x_1 - 0.2x_2 - \rho x_2^3 + u_1 + w_1 \\ 0.2x_1 + 1.1x_2 + \rho x_1^3 + u_2 + w_2 \end{pmatrix} \quad (31)$$

where $\rho = 0.0005$. The control input u is constrained by the set $\mathcal{U} = \mathcal{U}_1 \cap \mathcal{U}_2 \cap \mathcal{U}_3$ with

$$\mathcal{U}_1 = \mathbf{H}\left(0, \begin{pmatrix} 4 \\ 5 \end{pmatrix}\right), \mathcal{U}_2 = \mathbf{B}(0, 5), \mathcal{U}_3 = \mathbf{H}\left(0, \begin{pmatrix} 0.05 & 0 \\ 0 & 0.033 \end{pmatrix}\right)$$

and the exogenous input w by the set $\mathcal{W} = \mathbf{H}(0, (\omega_{\max}, \omega_{\max})^\top)$ with $\omega_{\max} = 0.01$. The specification Σ is clear from Figure 3 and we consider the quadratic cost stage cost function (12) with $\mathbf{Q} = \text{diag}(\mathbf{I}, \mathbf{I}, 1)$, i.e., $\mathcal{J}(x, u) = x^\top x + u^\top u + 1$.

A first feasible solution $\tilde{\mathcal{S}}_1$ (see Figure 3) was found after 18 seconds with only 8 cells and 7 transitions created. We can continue to expand the tree to explore other paths in the state-space as illustrated with abstraction $\tilde{\mathcal{S}}_2$ in Figure 3 (right). As anticipated, the controller exhibits a preference for solutions that pass near the origin while circumventing the obstacle.

Given $x_0 = (-10, -10) \in \mathcal{X}_I$, the guaranteed total cost by $\tilde{\mathcal{S}}_1$ is $v_1(x_0) = 1732$ whereas the true total cost of this specific trajectory is 1337. For $\tilde{\mathcal{S}}_2$, the guaranteed total cost is $v_2(x_0) = 992$ whereas the true total cost is 875. The control cost guaranteed by $\tilde{\mathcal{S}}_2$ is better than by $\tilde{\mathcal{S}}_1$ for two main reasons: 1) we continued to improve the abstraction using the RRT* variant, and 2) because by considering smaller cells, the cost of transitioning from one cell to another in the worst case is lower. Nevertheless, a similar part of the state space is covered by both abstractions, which is achieved with far fewer cells for $\tilde{\mathcal{S}}_1$.

VI. CONCLUSION

We provided a tractable algorithm for the optimal control of L -smooth nonlinear dynamical systems. Firstly, the proposed algorithm circumvents the necessity of discretizing the input space by employing a set of local feedback controllers. This is done to ensure deterministic transitions, thereby eliminating the non-determinism associated with abstraction, a common limitation in classic approaches. Secondly, the combined use of ellipsoid-based covering and affine local controllers leverages

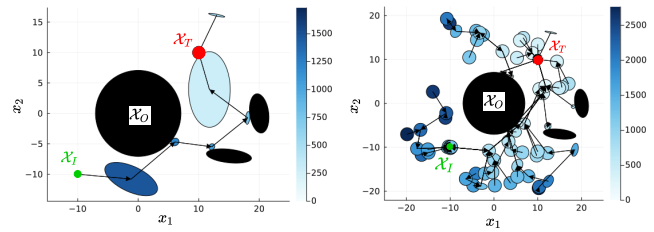


Fig. 3. (Section V) Solution obtained from Algorithm 1 (left) and from the improved RRT* version (right). The initial set \mathcal{X}_I , target set \mathcal{X}_T and obstacles \mathcal{X}_O are respectively in green, red and black. The color map illustrates the value function v for \mathcal{S} . Left: Construction of $\tilde{\mathcal{S}}_1$ and v_1 stops as soon as a feasible solution is found. Right: Continue to extend and improve the abstraction $\tilde{\mathcal{S}}_2$ and the value function v_2 . We added a bound on the cell volume to easily visualize the entire tree.

the power of LMIs and convex optimization, allowing the creation of larger and non-standard cells. Thirdly, the use of a lazy approach and non-uniform cells significantly reduces the complexity of the abstraction, i.e., the number of abstract states, when addressing a specific control problem.

As future work, we plan to demonstrate the efficiency of this approach on higher-dimensional nonlinear dynamical systems.

REFERENCES

- [1] J. Calbert, L. N. Egidio, and R. M. Jungers, "Smart abstraction based on iterative cover and non-uniform cells," <https://arxiv.org/abs/2403.02190>, (Extended version), 2024.
- [2] M. Rungger and M. Zamani, "Scots: A tool for the synthesis of symbolic controllers," in *Proceedings of the 19th international conference on hybrid systems: Computation and control*, 2016, pp. 99–104.
- [3] H. Khalil, *Nonlinear Systems*. Prentice Hall, 1996.
- [4] K. Hsu, R. Majumdar, K. Mallik, and A.-K. Schmuck, "Lazy abstraction-based controller synthesis," in *International Symposium on Automated Technology for Verification and Analysis*. Springer, 2019.
- [5] J. Calbert, B. Legat, L. N. Egidio, and R. Jungers, "Alternating simulation on hierarchical abstractions," in *IEEE Conference on Decision and Control (CDC)*, 2021, pp. 593–598.
- [6] L. N. Egidio, T. A. Lima, and R. M. Jungers, "State-feedback abstractions for optimal control of piecewise-affine systems," in *IEEE 61st Conference on Decision and Control (CDC)*, 2022.
- [7] S. M. LaValle *et al.*, "Rapidly-exploring random trees: A new tool for path planning," 1998.
- [8] J. J. Kuffner and S. M. LaValle, "Rrt-connect: An efficient approach to single-query path planning," in *IEEE International Conference on Robotics and Automation. Symposia Proceedings*, vol. 2, 2000.
- [9] S. M. LaValle and J. J. Kuffner Jr, "Randomized kinodynamic planning," *The international journal of robotics research*, vol. 20, 2001.
- [10] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The international journal of robotics research*, vol. 30, no. 7, pp. 846–894, 2011.
- [11] A. Manjunath and Q. Nguyen, "Safe and robust motion planning for dynamic robotics via control barrier functions," in *60th IEEE Conference on Decision and Control (CDC)*, 2021.
- [12] W. Ren and D. V. Dimarogonas, "Dynamic quantization based symbolic abstractions for nonlinear control systems," in *IEEE 58th Conference on Decision and Control (CDC)*, 2019, pp. 4343–4348.
- [13] A. Wu, T. Lew, K. Solovey, E. Schmerling, and M. Pavone, "Robust-rrt: Probabilistically-complete motion planning for uncertain nonlinear systems," *arXiv preprint arXiv:2205.07728*, 2022.
- [14] L. Asselborn, D. Gross, and O. Stursberg, "Control of uncertain nonlinear systems using ellipsoidal reachability calculus," *IFAC Proceedings Volumes*, vol. 46, no. 23, pp. 50–55, 2013.
- [15] P. Tabuada, *Verification and control of hybrid systems: a symbolic approach*. Springer Science & Business Media, 2009.
- [16] J. Calbert, L. N. Egidio, and R. M. Jungers, "An efficient method to verify the inclusion of ellipsoids," *IFAC-PapersOnLine*, vol. 56, no. 2, pp. 1958–1963, 2023.
- [17] Y. Nesterov *et al.*, *Lectures on convex optimization*. Springer, 2018, vol. 137.
- [18] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear matrix inequalities in system and control theory*. SIAM, 1994.