

École polytechnique de Louvain

Trajectory-Guided Ellipsoidal Abstractions for Reach-Avoid Control

A Certification Pipeline for Nonlinear Systems

Author: **Florentin CANART**
Supervisor: **Raphaël JUNGERS**
Readers: **Julien CALBERT, Benoît LEGAT**
Academic year 2025–2026
Master [120] in Mathematical Engineering

Acknowledgements

I would first like to thank my *grandmother*¹, whose mathematical spirit first guided my trajectory towards mathematics. Long before this thesis, she taught me to approach difficulties with curiosity, to navigate obstacles with patience, and to never lose the joy of the path.

I am especially grateful to *Julien Calbert*, whose advice, support, and generosity during my summer internship and throughout this thesis profoundly shaped my understanding of research. His guidance helped me grow technically, scientifically, as well as personally.

I would also like to thank *Benoît Legat* and *Raphaël Jungers* for their expertise, trust, and support. Their insights, resources, and demanding perspective provided an exceptional environment in which to develop this work.

I am also grateful to the *Dionysos* team, whose research seminars, group meetings, and scientific discussions offered me many opportunities to learn, exchange ideas, and better understand what research can be.

Finally, I thank my *family* and *friends* for their constant support.

¹A personal note: my grandmother defended her master's thesis in mathematical sciences at UCLouvain in 1970. Her thesis was devoted to set stability in metric spaces, with a focus on the stability and attractivity of sets for dynamical systems.

Abstract

This thesis studies reach-avoid control for nonlinear systems, where the objective is to synthesize feedback controllers that drive a set of initial states to a target set while avoiding unsafe states.

Abstraction-based control provides a systematic way to obtain formal guarantees, but global symbolic abstractions may become large or conservative. Conversely, trajectory optimization and sampling-based model predictive methods often find useful motions, but their outputs are nominal and do not, by themselves, provide robustness guarantees for the resulting control sequence.

The central idea of this thesis is to use nominal trajectories not as certificates, but as guides for constructing local partial abstractions. We formalize this viewpoint through certified local reach-avoid chains: finite sequences of local sets equipped with feedback laws whose robust one-step reachable images are included in the next set.

The proposed pipeline first computes a nominal trajectory using a coarse centered abstraction and MPPI refinement, and then attempts to lift it into a chain of ellipsoidal sets with affine feedback laws using a backward LMI-based certifier. The method is implemented in `Dionysos.jl` and evaluated on nonlinear benchmarks.

The experiments show that nontrivial certified behaviours can be obtained, while also identifying key certifiability bottlenecks: local model conservatism, limited feedback authority near input saturation, numerical conditioning, and insufficient geometric clearance. Overall, the thesis proposes a task-oriented alternative to monolithic abstraction construction: certify relevant behaviours locally, while preserving the soundness of abstraction-based control.

Contents

Abstract	ii
Notation	v
1 Introduction	1
1.1 Motivation	1
1.2 Contributions	5
1.3 Structure of the manuscript	6
2 Background and Related Work	7
2.1 Reach-avoid control and formal guarantees	7
2.2 Abstraction-based control	9
2.2.1 Symbolic models and conservative transitions	10
2.2.2 Controller synthesis and refinement	12
2.3 Local set-based abstractions	13
2.4 Nominal trajectory generation	16
2.4.1 Nominal planning as finite-horizon optimal control	16
2.4.2 Families of nominal planners	16
2.4.3 Model Predictive Path Integral control	17
2.5 Local certification tools	19
2.6 Positioning of this thesis	21
3 Problem Formulation	22
3.1 System and specification	22
3.2 Certified local reach-avoid chains	22
3.3 Standing assumptions	25
3.4 Nominal planning and certification problems	27
4 A Modular Trajectory-Guided Certification Pipeline	29
4.1 Overview of the proposed pipeline	29
4.2 Chosen instantiation of the pipeline	30
4.3 Nominal trajectory generation	31
4.3.1 Centered simulation path warm start	31
4.3.2 MPPI refinement	32
4.4 Backward ellipsoidal certification	34
4.5 Complete algorithm	36
4.6 Soundness, scope, and limitations	37
5 Benchmarks and Experimental Evaluation	38
5.1 Benchmark design and evaluation protocol	38

5.1.1	Evaluation metrics	39
5.2	Main certification results	39
5.2.1	Simple pendulum	39
5.2.2	Double pendulum	43
5.2.3	Articulated vehicle benchmarks	46
5.3	Supplementary animations	60
5.4	Ablation, comparison, and qualitative studies	61
5.4.1	Planning-stage ablation: effect of the warm start	61
5.4.2	Abstraction baseline: comparison with a uniform abstraction	63
5.4.3	Pipeline bottlenecks	66
5.4.4	From certified chains to task libraries	70
6	Conclusion and Future Work	71
6.1	Summary of contributions	71
6.2	Main findings	72
6.3	Future work	74
	References	76
A	Local LMI Certificate	80
A.1	Local transition problem	80
A.2	Local affine model and nonlinear remainder	81
A.3	Convex change of variables	82
A.4	LMI transition certificate	83
A.5	Why the LMIs imply the certificate	84
A.6	Scaling and numerical conditioning	85
A.7	Optional numerical objective	87
B	Software implementation details	89

Notation

Notation	Meaning
Systems and reach-avoid specification	
$\mathcal{X} \subseteq \mathbb{R}^n, \mathcal{U} \subseteq \mathbb{R}^m, \mathcal{W} \subseteq \mathbb{R}^p$	State, input, and disturbance sets.
$\mathcal{X}_I, \mathcal{X}_T, \mathcal{X}_O \subseteq \mathcal{X}$	Initial, target, and obstacle/unsafe sets.
$\mathcal{X}_{\text{safe}} := \mathcal{X} \setminus \mathcal{X}_O$	Safe part of the state space.
$x_{k+1} = f(x_k, u_k, w_k)$	Discrete-time uncertain nonlinear dynamics.
$S = (\mathcal{X}, \mathcal{U}, F)$	Transition system associated with the dynamics.
$F(x, u) := \{f(x, u, w) : w \in \mathcal{W}\}$	Set-valued one-step transition map.
$\mathcal{P} := (f, \mathcal{X}, \mathcal{U}, \mathcal{W}, \mathcal{X}_I, \mathcal{X}_T, \mathcal{X}_O)$	Reach-avoid problem data.
Nominal planning and certified chains	
$(\bar{x}_{0:N}, \bar{u}_{0:N-1})$	Nominal trajectory over horizon N .
Ω_k	Local certified set at step k .
$\kappa_k : \Omega_k \rightarrow \mathcal{U}$	Local feedback law used on Ω_k .
$\mathfrak{C}_N := ((\Omega_k)_{k=0}^N, (\kappa_k)_{k=0}^{N-1})$	Certified local reach-avoid chain.
$\mathcal{X}_I^{\text{cert}}(\mathfrak{C}_N) := \Omega_0 \cap \mathcal{X}_I$	Certified subset of initial states induced by the chain.
$\text{Reach}_f(\Omega, \kappa)$	Closed-loop one-step reachable set $\{f(x, \kappa(x), w) : x \in \Omega, w \in \mathcal{W}\}$.
$\text{Reach}_f(\Omega_k, \kappa_k) \subseteq \Omega_{k+1}$	Certified transition condition.
Ellipsoidal certificates	
$\mathcal{E}(c, P)$	Ellipsoid $\{x \in \mathbb{R}^n : (x - c)^\top P^{-1}(x - c) \leq 1\}$, with $P > 0$.
$\mathcal{E}_k = \mathcal{E}(c_k, P_k)$	Certified ellipsoid at step k , with center c_k and shape matrix P_k .
$\kappa_k(x) = \ell_k + K_k(x - c_k)$	Local affine feedback law on $\mathcal{E}_k$.
ℓ_k, K_k	Feedforward term and local feedback gain.
$\mathcal{H}(c, r)$	Hyperrectangle centered at c with half-lengths r .
$\mathcal{B}_k = \mathcal{Y}_k \times \mathcal{V}_k \times \mathcal{W}$	Local certification region for state, input, and disturbance variables.
$\tilde{f}_k$	Local affine approximation of f around the nominal transition.
$\mathcal{H}_{r,k}$	Sound bound on the nonlinear remainder on $\mathcal{B}_k$.
$\oplus$	Minkowski sum.
Abstraction viewpoint	
$q, R^{-1}(q)$	Abstract state and its concrete region.
$\mathcal{K}$	Set of feedback labels in a state-feedback abstraction.
D_x, D_x^+, D_u	Diagonal state, successor-state, and input scalings used for numerical conditioning.
Linear-algebra and LMI notation	
$M^\top, M^{-1}$	Transpose and inverse of a matrix M , when the inverse exists.
$\mathbb{S}_+^n, \mathbb{S}_{++}^n$	Cones of symmetric positive semidefinite and positive definite matrices.
$M \geq 0, M > 0$	Positive semidefiniteness and positive definiteness.
$\ z\ $	Euclidean norm.
$\text{co}\{w^1, \dots, w^{N_w}\}$	Convex hull of finitely many disturbance vertices.
A_k, B_k, E_k, g_k	Local affine-model matrices in $\tilde{f}_k(x, u, w) = A_k x + B_k u + E_k w + g_k$.

List of Figures

1.1	Constant input and nondeterminism	3
1.2	Uniform-grid growth with dimension	3
1.3	From global abstractions to trajectory-guided certification	4
2.1	Abstraction-based control workflow	9
2.2	Conservative transition in a grid abstraction	11
2.3	Partition versus overlapping concretization covers	14
2.4	Uniform versus local abstractions	15
2.5	Ellipsoidal transition certificate	20
4.1	Instantiated trajectory-guided pipeline	30
4.2	Centered-abstraction warm start	32
4.3	MPPI trajectory refinement	34
4.4	Backward ellipsoidal certification	36
5.1	Simple pendulum: nominal MPPI trajectory	40
5.2	Simple pendulum: certified swing-up chain	41
5.3	Simple pendulum: input-saturation diagnostic	42
5.4	Double pendulum: abstract warm start	44
5.5	Double pendulum: MPPI-refined trajectory	44
5.6	Double pendulum: certified ellipsoidal chain	45
5.7	Articulated vehicle kinematic geometry	47
5.8	Forward manoeuvre: abstract warm start	48
5.9	Forward manoeuvre: certified trajectory and chain	49
5.10	Forward manoeuvre: closed-loop rollouts	50
5.11	Forward manoeuvre: inflation stress test	51
5.12	Reverse manoeuvre: abstract warm start	53
5.13	Reverse manoeuvre: MPPI-refined trajectory	53
5.14	Reverse manoeuvre: certified ellipsoidal chain	54
5.15	Parallel parking: abstract warm start	57
5.16	Parallel parking: MPPI-refined trajectory	57
5.17	Parallel parking: certified ellipsoidal chain	58
5.18	Warm-start ablation in the maze benchmark	62
5.19	Uniform-grid baseline in a trajectory corridor	65
5.20	Composition of certified chain contracts	70
A.1	Scaling invariance of the local certificate	85
A.2	Adaptive local certification boxes	87

List of Tables

2.1	Nominal trajectory generators	17
5.1	Role of the benchmark problems	38
5.2	Experimental evaluation metrics	39
5.3	Summary for the simple pendulum benchmark	41
5.4	Summary for the double pendulum benchmark	45
5.5	Summary for the forward articulated-vehicle benchmark	50
5.6	Summary for the reverse articulated-vehicle benchmark	55
5.7	Summary for the parallel-parking articulated-vehicle benchmark	59
5.8	Supplementary benchmark animations	60
5.9	Uniform-grid abstraction baseline summary	65
B.1	Main software components	90

Introduction

1.1 Motivation

Modern autonomous and robotic systems increasingly make control decisions while interacting with nonlinear, uncertain, and constrained physical dynamics. In such settings, the objective is rarely to produce a single successful simulation. One would like to synthesize controllers whose safety and reachability properties hold for a set of initial states, and remain valid under admissible disturbances, modelling errors, or small deviations from the nominal behaviour [1–4].

A convenient formalization of this requirement is the *reach-avoid* problem [5, 6]. Given an initial set $\mathcal{X}_I$, a target set $\mathcal{X}_T$, and an unsafe set $\mathcal{X}_O$, the objective is to synthesize a feedback controller on a certified subset $\mathcal{X}_I^{\text{cert}} \subseteq \mathcal{X}_I$ such that every admissible closed-loop trajectory satisfies

$$x_0 \in \mathcal{X}_I^{\text{cert}} \implies \exists N \in \mathbb{N} : x_N \in \mathcal{X}_T \quad \text{and} \quad x_k \notin \mathcal{X}_O \quad \text{for all } k = 0, \dots, N-1.$$

This property is simple to state, but difficult to certify for nonlinear systems with continuous state spaces. A finite number of simulations may reveal failures, but it cannot prove that no failure exists. The relevant object is therefore not one trajectory, but a set of behaviours controlled in a worst-case sense.

Abstraction-based control provides a systematic way to reason about such set-based properties. The continuous-state control problem is replaced by a finite symbolic model on which correct-by-construction synthesis can be performed [7–9]. Its central strength is soundness. An abstract state represents a set of concrete states, and an abstract transition conservatively represents all concrete successors compatible with that set and input. If the abstraction contains all admissible concrete behaviours, then a controller synthesized on the symbolic model can be refined into a controller for the original system.

This strength also explains the computational price of classical abstractions. A uniform symbolic abstraction attempts to represent all behaviours in a region of the state space. However, when the system is nonlinear, constrained, and continuous-state, constructing a sufficiently accurate global model may become prohibitively expensive or overly conservative. The limitation is therefore not that uniform abstractions are conceptually weak; it is that they often solve a more ambitious problem than what is needed for a single reach-avoid task.

Trajectory optimization and model predictive control take a complementary route. They optimize directly over the continuous dynamics and often scale well to nonlinear

systems [10–15]. They are effective at finding candidate motions, especially in systems where a uniform abstraction would be too expensive. However, the resulting trajectories are usually nominal. By themselves, they do not certify robust reach-avoid correctness for a set of initial states and disturbances. A nominal trajectory may reach the target in simulation and still leave no room for feedback corrections, obstacle clearance, or local robustness.

The starting point of this thesis is the tension between these two viewpoints. Abstraction-based control provides a soundness guarantee, but may require a large finite model of the system. Trajectory optimization efficiently proposes useful motions, but a nominal motion is not a proof. The question studied here is whether one can keep the soundness of abstraction-based control without paying the price of abstracting all behaviours. The proposed answer is to use nominal trajectories not as certificates, but as guides for constructing local certificates.

Three bottlenecks appear repeatedly in classical grid-based abstractions. They are not flaws of the abstraction-based paradigm; rather, they are the price of building finite conservative models of continuous dynamics.

- **Over-approximation and nondeterminism.** Let q be an abstract state whose concretization $R^{-1}(q)$ is a cell in the continuous state space, and let u be a representative input. The exact set of one-step concrete successors is

$$F(R^{-1}(q), u).$$

Since this set is rarely available exactly, one computes a conservative enclosure $\widehat{F}(R^{-1}(q), u)$ such that

$$F(R^{-1}(q), u) \subseteq \widehat{F}(R^{-1}(q), u).$$

Every grid cell intersecting this enclosure must then be added as a possible abstract successor. This preserves all concrete behaviours, which is the source of soundness. But it may also introduce spurious successors. During synthesis, these successors are treated in a worst-case sense: an input is useful only if all possible abstract successors remain compatible with the specification. Thus, conservative over-approximation may enlarge the transition relation and shrink the winning set.

- **Constant inputs inside cells.** In many grid-based constructions, a single input is associated with the whole cell represented by q :

$$u(x) = \bar{u}_q, \quad x \in R^{-1}(q).$$

This is algorithmically convenient, but it ignores the fact that different states inside the same cell may require different corrective actions. A fixed input must propagate all points in the cell at once, and the corresponding reachable image may intersect many successor cells. A state-dependent feedback law can instead react to the position inside the cell and localize the closed-loop image. Figure 1.1 illustrates this effect: replacing a constant input by feedback can reduce the

nondeterminism induced by the abstraction.

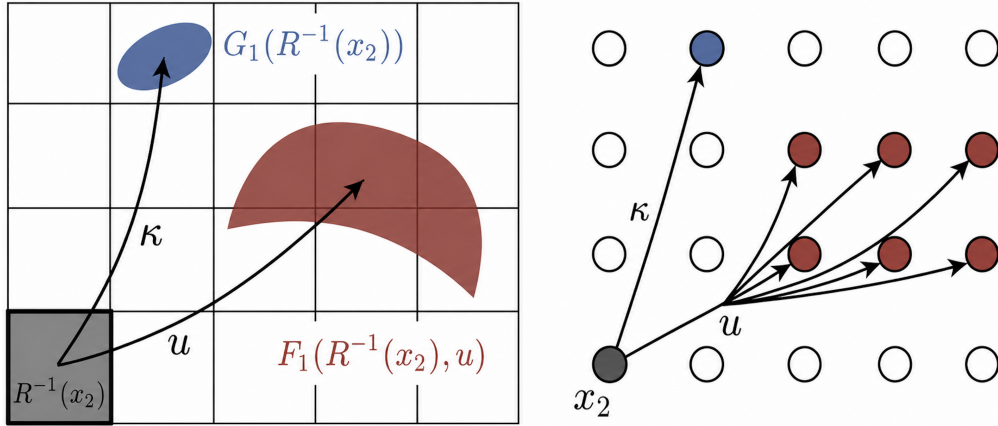


Figure 1.1: Illustration of constant-input conservatism in grid-based abstractions. A fixed input may produce a large reachable image, leading to several possible abstract successors. A state-dependent feedback law can localize the image and reduce nondeterminism. From [9].

- **Curse of dimensionality.** The number of cells in a uniform grid grows exponentially with the state dimension. If η denotes the discretization resolution in dimension n , the number of abstract states typically scales as

$$O(\eta^{-n}).$$

Hence a moderate refinement may already produce a very large finite model. Moreover, this is not only a storage issue: each additional cell comes with transitions, conservative enclosures, and fixed-point synthesis operations. The combinatorial growth of the abstraction is therefore coupled with the conservatism of its transition relation.

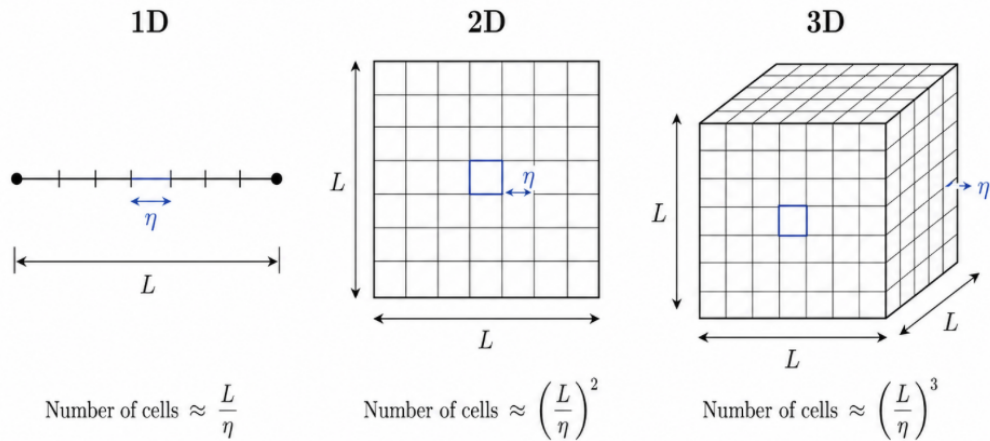


Figure 1.2: Growth of the number of abstract states under uniform discretization.

A natural response is to restrict the abstraction to a corridor around a nominal trajectory. This reduces the number of cells and focuses the computation on the part

of the state space that appears relevant for the task. However, if the corridor is still represented by a local grid with representative inputs and conservative cell-to-cell transitions, then the same mechanisms remain present locally: over-approximation creates nondeterminism, constant inputs may be too weak to stabilize deviations inside a cell, and obstacle clearance may force very fine local partitions. This baseline will be evaluated experimentally in Chapter 5.

The approach developed in this thesis follows a different principle. Instead of using a nominal trajectory to restrict a global grid abstraction, the trajectory is used as a scaffold for constructing a partial feedback abstraction. Around the trajectory, one builds local sets equipped with local feedback controllers, and certifies only the transitions needed to realize the desired behaviour. In this sense, the abstraction is not global and behaviour-complete; it is local, task-oriented, and feedback-labelled.

In the instantiation studied here, a planner first computes a nominal reach-avoid trajectory, namely a sequence of states and inputs that reaches the target and avoids the obstacle in simulation. This trajectory is then passed to a local certifier. Around each nominal state, the method constructs an ellipsoid and equips it with a local affine feedback law,

$$\kappa_k(x) = \ell_k + K_k(x - \bar{x}_k).$$

The ellipsoid describes the set of states certified around the nominal state $\bar{x}_k$, while the feedback law corrects deviations from the nominal motion. The required certificate is a robust one-step inclusion of the form

$$\text{Reach}_f(\mathcal{E}_k, \kappa_k) \subseteq \mathcal{E}_{k+1},$$

together with input and safety constraints. Thus, trajectory optimization indicates where to certify, while local feedback certification provides the set-based guarantee.

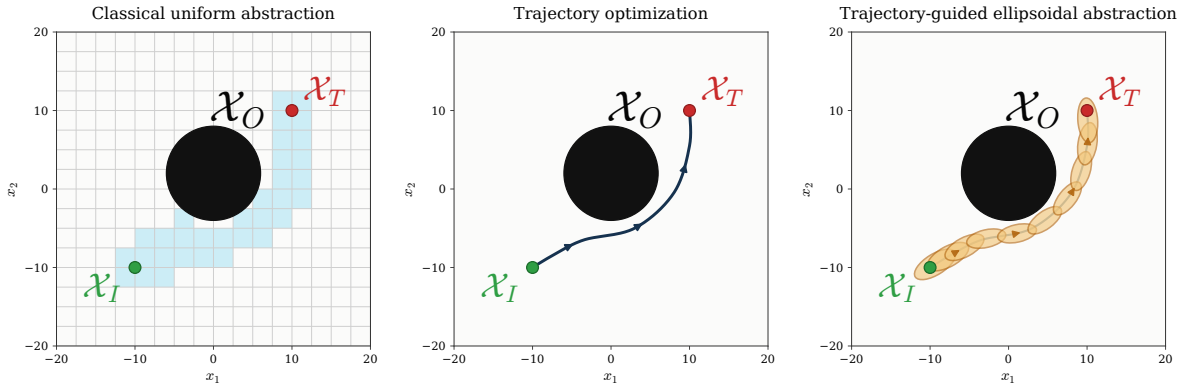


Figure 1.3: From global abstractions to trajectory-guided certification. Uniform symbolic abstractions provide a conservative finite model of many behaviours, but may scale poorly. Trajectory optimization finds a useful motion, but returns a nominal object. The proposed method uses the nominal motion to decide where a local abstraction should be constructed, and uses feedback-labelled ellipsoidal certificates to recover a robust reach-avoid guarantee around that behaviour.

The resulting object is a finite chain of local sets and feedback laws,

$$\Omega_0 \xrightarrow{\kappa_0} \Omega_1 \xrightarrow{\kappa_1} \dots \xrightarrow{\kappa_{N-1}} \Omega_N.$$

When the inclusions

$$\text{Reach}_f(\Omega_k, \kappa_k) \subseteq \Omega_{k+1}, \quad k = 0, \dots, N-1,$$

are certified, this chain is not merely a tube around a trajectory. It is also a partial abstraction of the concrete system. The sets Ω_k are local abstract states, and the feedback laws κ_k are transition labels. This is the main conceptual viewpoint of the thesis: a nominal trajectory selects a behaviour, and a certified chain turns this behaviour into a local abstraction with a robust reach-avoid guarantee.

This viewpoint also suggests a compositional interpretation. A certified chain can be read as a local contract: its entry set is the certified initial region, its exit set is the terminal region, and its guarantee is that all admissible closed-loop trajectories starting in the entry set reach the exit set safely. If the exit set of one chain lies inside the entry set of another, the two contracts can be concatenated. Thus, rather than constructing one monolithic abstraction of all behaviours, one may build libraries of certified behaviours and let a higher-level planner orchestrate them [16]. This thesis only takes a first step in this direction, but the contract viewpoint motivates the proposed local abstraction object.

1.2 Contributions

The contributions of this thesis are the following.

- **A certified-chain formulation for local reach-avoid control.** The thesis introduces certified local reach-avoid chains: finite sequences of local sets and feedback laws satisfying input constraints, obstacle constraints, target inclusion, and robust one-step transition inclusions. Such a chain induces a finite-horizon controller that solves the reach-avoid specification on the certified subset of initial states.
- **A trajectory-guided synthesis problem.** The computation is formulated as a two-stage problem. A nominal planner first proposes a candidate reach-avoid trajectory. A local certifier then attempts to lift this trajectory into a certified chain. The planner decides where certification should be attempted; the certifier is the source of the formal guarantee.
- **A modular planner–certifier pipeline in Dionysos.jl.** The proposed framework is implemented in Dionysos.jl [9]. The instantiation studied in the thesis combines a centered simulation abstraction used as a coarse warm start, an MPPI refinement step used to obtain a dynamically consistent nominal trajectory, and a backward ellipsoidal certifier based on local affine feedback and LMI transition certificates.

- **An experimental study of certifiability on nonlinear benchmarks.** The method is evaluated on pendulum and articulated-vehicle reach-avoid tasks. The experiments show that the proposed pipeline can certify nontrivial local behaviours, while also identifying the main bottlenecks of the current instantiation: nonlinear-model conservatism, input saturation and lack of feedback authority, numerical conditioning, and obstacle-clearance limitations.
- **A contract interpretation of certified chains.** The thesis interprets certified chains as reusable local behaviours with entry sets, exit sets, feedback policies, and reach-avoid guarantees. This provides a first step toward task libraries in which certified behaviours could be selected and concatenated by a higher-level planner.

1.3 Structure of the manuscript

The remainder of the manuscript is organized as follows.

- **Chapter 2** introduces the background used throughout the thesis. It recalls the reach-avoid setting, transition-system models, the soundness principle of abstraction-based control, and the local certification tools used later in the manuscript. It also positions the proposed approach with respect to symbolic control, robust tubes, safe tracking, and funnel-based certification.
- **Chapter 3** formulates the mathematical problem addressed in the thesis. It defines certified local reach-avoid chains, proves their soundness, and states trajectory-guided synthesis as the combination of nominal planning and local robust certification.
- **Chapter 4** presents the proposed algorithmic pipeline. A centered simulation abstraction first provides a non-certified global guide, which is refined by MPPI. A backward ellipsoidal certification procedure then constructs local sets equipped with affine feedback laws along the nominal trajectory.
- **Chapter 5** evaluates the method on nonlinear benchmarks. It studies the role of warm starts, compares the approach with a uniform-corridor abstraction baseline, and identifies the main certification bottlenecks observed in practice.
- **Chapter 6** summarizes the contributions and discusses future work. It highlights certified chains as the central abstraction object and outlines directions such as certification-aware planning, improved certification backends, and libraries of certified behaviours.

Background and Related Work

This chapter recalls the technical ingredients used in the rest of the manuscript. Section 2.1 introduces robust reach-avoid specifications through set-valued transition systems. Section 2.2 recalls the soundness principle of conservative symbolic abstractions. Section 2.3 then moves from uniform cells to local feedback-labelled sets, while Section 2.4 discusses how nominal trajectories can be used to choose where such local sets should be constructed. Section 2.5 introduces the ellipsoidal transition certificates used later in the thesis. The chapter ends with Section 2.6, which positions the proposed trajectory-guided approach with respect to abstraction-based control, robust tubes, safe tracking, and funnel-based certification.

2.1 Reach-avoid control and formal guarantees

We first introduce a transition-system formalism, which provides a common language for deterministic systems, uncertain systems, and finite symbolic abstractions. This formalism is standard in abstraction-based control [7, 8].

Definition 2.1 (Transition system). A *transition system* is a tuple $S = (\mathcal{X}, \mathcal{U}, F)$, where $\mathcal{X}$ is the state set, $\mathcal{U}$ is the input set, and $F : \mathcal{X} \times \mathcal{U} \rightarrow 2^{\mathcal{X}}$ is a set-valued transition map. A one-step transition of S satisfies the condition

$$x_{k+1} \in F(x_k, u_k). \quad (2.1)$$

Given a transition system $S = (\mathcal{X}, \mathcal{U}, F)$, the set of inputs available at a state $x \in \mathcal{X}$ is defined by

$$\mathcal{U}_S(x) := \{u \in \mathcal{U} : F(x, u) \neq \emptyset\}. \quad (2.2)$$

Definition 2.2 (Trajectory and behavior). Let $S = (\mathcal{X}, \mathcal{U}, F)$ be a transition system and let $N \in \mathbb{N} \cup \{\infty\}$. A trajectory of horizon N is a pair of sequences $(x_{0:N}, u_{0:N-1})$ such that, $\forall k \in [0, N-1]$,

$$x_k \in \mathcal{X}, \quad u_k \in \mathcal{U}_S(x_k), \quad x_{k+1} \in F(x_k, u_k). \quad (2.3)$$

If $N = \infty$, the same condition holds for all $k \in \mathbb{N}$. The set of all trajectories of horizon N is denoted by $\mathcal{B}_N(S)$. The *behavior* of S is the set of all its finite and infinite trajectories:

$$\mathcal{B}(S) := \bigcup_{N \in \mathbb{N} \cup \{\infty\}} \mathcal{B}_N(S).$$

Definition 2.3 (Uncertain discrete-time control system). An uncertain discrete-time control system is a tuple $D_d = (\mathcal{X}, \mathcal{U}, \mathcal{W}, f_d)$, where $\mathcal{X} \subseteq \mathbb{R}^n$, $\mathcal{U} \subseteq \mathbb{R}^m$, $\mathcal{W} \subseteq \mathbb{R}^p$ is bounded, and $f_d : \mathcal{X} \times \mathcal{U} \times \mathcal{W} \rightarrow \mathcal{X}$ is a possibly nonlinear transition map. The dynamics are

$$x_{k+1} = f_d(x_k, u_k, w_k), \quad x_k \in \mathcal{X}, \quad u_k \in \mathcal{U}, \quad w_k \in \mathcal{W}. \quad (2.4)$$

The associated transition system is $S_d = (\mathcal{X}, \mathcal{U}, F_d)$, with $F_d : \mathcal{X} \times \mathcal{U} \rightarrow 2^{\mathcal{X}}$ given by

$$F_d(x, u) := \{f_d(x, u, w) : w \in \mathcal{W}\}. \quad (2.5)$$

Continuous-time models will be treated through their sampled-data discretization, so that the certification problem can be formulated directly on a discrete-time transition map of the form above.

Definition 2.4 (Reachable set). Let $S = (\mathcal{X}, \mathcal{U}, F)$ be a transition system. For a set $\Omega \subseteq \mathcal{X}$ and an input $u \in \mathcal{U}$, the one-step reachable set from Ω under u is

$$F(\Omega, u) := \bigcup_{x \in \Omega} F(x, u).$$

Definition 2.5 (Static controller). Let $S = (\mathcal{X}, \mathcal{U}, F)$ be a transition system. A *static controller* for S is a set-valued map $C : \mathcal{X} \rightarrow 2^{\mathcal{U}}$ such that

$$C(x) \subseteq \mathcal{U}_S(x), \quad \forall x \in \mathcal{X}. \quad (2.6)$$

Its domain is defined as $\text{dom}(C) := \{x \in \mathcal{X} : C(x) \neq \emptyset\}$.

The closed-loop system induced by C is the transition system $C \times S = (\mathcal{X}, \mathcal{U}, F_C)$, where

$$F_C(x, u) := \begin{cases} F(x, u), & \text{if } u \in C(x), \\ \emptyset, & \text{otherwise.} \end{cases} \quad (2.7)$$

Since C may be set-valued, closed-loop trajectories include all possible selections $u_k \in C(x_k)$. Therefore, a specification solved by C must hold for every admissible input selection and every successor allowed by F .

Definition 2.6 (Reach-avoid specification on a certified initial set). Let $S = (\mathcal{X}, \mathcal{U}, F)$ be a transition system, and let $\mathcal{X}_I, \mathcal{X}_{\mathcal{T}}, \mathcal{X}_O \subseteq \mathcal{X}$, where $\mathcal{X}_I$ is the initial set, $\mathcal{X}_{\mathcal{T}}$ is the target set, and $\mathcal{X}_O$ is the obstacle set, with $\mathcal{X}_{\mathcal{T}} \cap \mathcal{X}_O = \emptyset$.

Given a nonempty set $\mathcal{X}_I^{\text{cert}} \subseteq \mathcal{X}_I$, a controller C solves the reach-avoid specification on $\mathcal{X}_I^{\text{cert}}$ if every closed-loop trajectory (x, u) of $C \times S$ starting from $x_0 \in \mathcal{X}_I^{\text{cert}}$ satisfies

$$\exists N \in \mathbb{N}_0 \quad \text{such that} \quad x_N \in \mathcal{X}_{\mathcal{T}} \quad \text{and} \quad x_k \in \mathcal{X} \setminus \mathcal{X}_O \quad \forall k = 0, \dots, N-1. \quad (2.8)$$

The set $\mathcal{X}_I^{\text{cert}}$ is called the certified initial set. The certified-set formulation is inspired by the reach-avoid synthesis framework of Fan et al. [6, Secs. III-C IV-C IV-D].

2.2 Abstraction-based control

As discussed in the Introduction, abstraction-based control is a systematic way to synthesize controllers with formal guarantees for continuous or infinite-state systems [7, 8, 15, 17, 18]. In the context of this thesis, it provides a natural framework for reach-avoid specifications of the form introduced in Definition 2.6.

The general abstraction-based control workflow consists of three steps:

1. *Abstraction*: the original control system is approximated by a finite-state transition system;
2. *Abstract controller synthesis*: a discrete controller is synthesized on the finite abstraction;
3. *Concretization*: the abstract controller is refined into a controller for the original system.

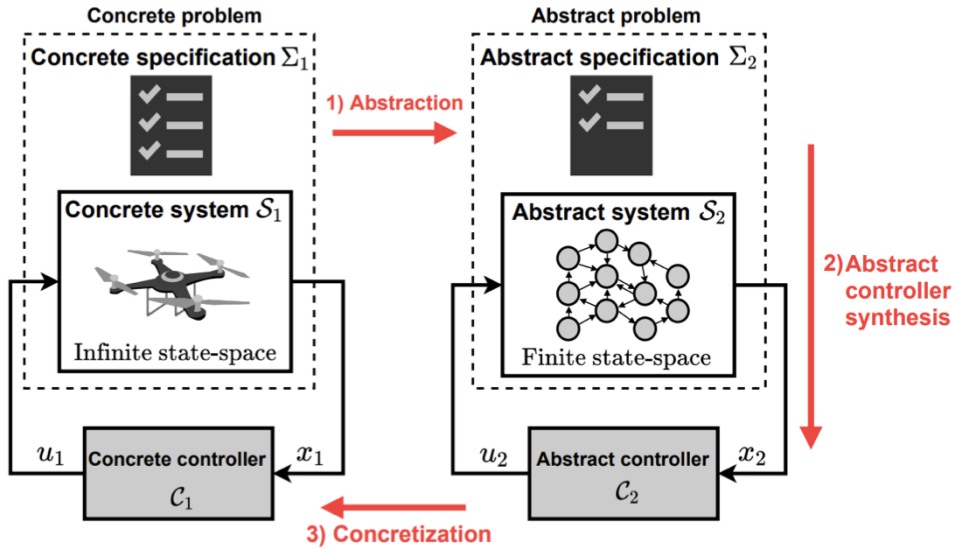


Figure 2.1: The three steps of abstraction-based control illustrated. From [9].

The main mathematical difficulty is to ensure that the controller synthesized on the abstract system remains sound when it is applied to the original system. In the following, the original system will be called the *concrete system*, whereas the finite model used for synthesis will be called the *abstract system*.

Let

$$S_1 = (\mathcal{X}_1, \mathcal{U}_1, F_1) \quad \text{and} \quad S_2 = (\mathcal{X}_2, \mathcal{U}_2, F_2)$$

denote respectively the concrete and abstract transition systems. The objective is to synthesize a concrete controller $C_1 : \mathcal{X}_1 \rightarrow 2^{\mathcal{U}_1}$ from an abstract controller $C_2 : \mathcal{X}_2 \rightarrow 2^{\mathcal{U}_2}$. To formalize this refinement step, one must relate concrete states to abstract states. This is done through a binary relation between $\mathcal{X}_1$ and $\mathcal{X}_2$, which describes which abstract states represent a given concrete state.

2.2.1 Symbolic models and conservative transitions

Throughout this subsection, let $S_1 = (\mathcal{X}_1, \mathcal{U}_1, F_1)$ denote the original transition system. A symbolic model replaces this system by a finite transition system $S_2 = (\mathcal{X}_2, \mathcal{U}_2, F_2)$, whose states are symbols representing sets of concrete states. For simplicity, we identify abstract inputs with concrete representative inputs and assume $\mathcal{U}_2 \subseteq \mathcal{U}_1$.

Definition 2.7 (Binary relation). Given two sets $\mathcal{X}_1$ and $\mathcal{X}_2$ we identify a *binary relation* $R \subseteq \mathcal{X}_1 \times \mathcal{X}_2$ with the set-valued maps $R : \mathcal{X}_1 \rightarrow 2^{\mathcal{X}_2}$ and $R^{-1} : \mathcal{X}_2 \rightarrow 2^{\mathcal{X}_1}$ such that :

$$R(x_1) := \{x_2 \in \mathcal{X}_2 : (x_1, x_2) \in R\}, \quad R^{-1}(x_2) := \{x_1 \in \mathcal{X}_1 : (x_1, x_2) \in R\}. \quad (2.9)$$

The relation R is said to be *strict* if

$$R(x_1) \neq \emptyset, \quad \forall x_1 \in \mathcal{X}_1. \quad (2.10)$$

Definition 2.8 (Quantization relation). Let $S_1 = (\mathcal{X}_1, \mathcal{U}_1, F_1)$ be a transition system and let $\mathcal{X}_2$ be a finite set. A strict relation $R \subseteq \mathcal{X}_1 \times \mathcal{X}_2$ is called a *quantization relation*. For $x \in \mathcal{X}_1$, the set $R(x)$ contains the abstract states that represent x . For $q \in \mathcal{X}_2$, the set $R^{-1}(q)$ is the concretization of q , namely the concrete region represented by the symbolic state q .

Definition 2.9 (Finite symbolic abstraction). Let $S_1 = (\mathcal{X}_1, \mathcal{U}_1, F_1)$ be a transition system. A finite symbolic abstraction of S_1 is a pair (S_2, R) , where $S_2 = (\mathcal{X}_2, \mathcal{U}_2, F_2)$ is a transition system such that $\mathcal{X}_2$ and $\mathcal{U}_2$ are finite, and $R \subseteq \mathcal{X}_1 \times \mathcal{X}_2$ is a strict quantization relation. Each $q \in \mathcal{X}_2$ is a symbolic state representing the concrete set $R^{-1}(q)$.

The role of F_2 is not to reproduce F_1 exactly, but to conservatively represent its behaviours. Thus, the abstract system may contain more transitions than the concrete system, but it should not discard any concrete behaviour.

Definition 2.10 (Hyperrectangle). Let $c \in \mathbb{R}^n$ and $h \in \mathbb{R}_+^n$. The hyperrectangle centered at c with half-lengths h is

$$H(c, h) := \{x \in \mathbb{R}^n : |x_i - c_i| \leq h_i, i = 1, \dots, n\}.$$

In uniform grid abstractions, the set $R^{-1}(q)$ is typically chosen as a hyperrectangle centered at the grid point q .

Conservative transitions. Given $q \in \mathcal{X}_2$ and $u \in \mathcal{U}_2$, the one-step concrete successors from the region represented by q are

$$F_1(R^{-1}(q), u) := \bigcup_{x \in R^{-1}(q)} F_1(x, u). \quad (2.11)$$

Since this set is rarely available exactly, one constructs an enclosure $\widehat{F}_1(R^{-1}(q), u)$ satisfying

$$F_1(R^{-1}(q), u) \subseteq \widehat{F}_1(R^{-1}(q), u). \quad (2.12)$$

Definition 2.11 (Conservative abstract transition). Let (S_2, R) be a finite symbolic abstraction of $S_1 = (\mathcal{X}_1, \mathcal{U}_1, F_1)$. Suppose that, for each $q \in \mathcal{X}_2$ and $u \in \mathcal{U}_2$, an enclosure $\hat{F}_1(R^{-1}(q), u) \subseteq \mathcal{X}_1$ satisfies (2.12). A conservative abstract transition map is any map $F_2 : \mathcal{X}_2 \times \mathcal{U}_2 \rightarrow 2^{\mathcal{X}_2}$ such that

$$\begin{aligned} \{q^+ \in \mathcal{X}_2 : R^{-1}(q^+) \cap \hat{F}_1(R^{-1}(q), u) \neq \emptyset\} &\subseteq F_2(q, u), \\ \forall q \in \mathcal{X}_2, \forall u \in \mathcal{U}_2. \end{aligned} \quad (2.13)$$

Equivalently, every concrete successor is represented by at least one abstract successor:

$$F_1(R^{-1}(q), u) \subseteq R^{-1}(F_2(q, u)).$$

This construction is sound because it does not remove any concrete behaviour. However, the enclosure may intersect symbolic states that are not reached by any true trajectory. The corresponding transitions are called spurious transitions and are a major source of nondeterminism in grid-based symbolic abstractions. This nondeterminism should be interpreted in a worst-case sense. In the abstract synthesis problem, an input is admissible only if all abstract successors that may follow from this input remain compatible with the specification. Hence, even a single spurious successor can invalidate an otherwise useful control action: the controller must be robust not only to true disturbances and initial uncertainty inside the cell, but also to the additional behaviours introduced by conservative over-approximation. This is the fundamental trade-off of conservative symbolic abstractions: soundness is obtained by adding behaviours, but these additional behaviours may shrink the winning set.

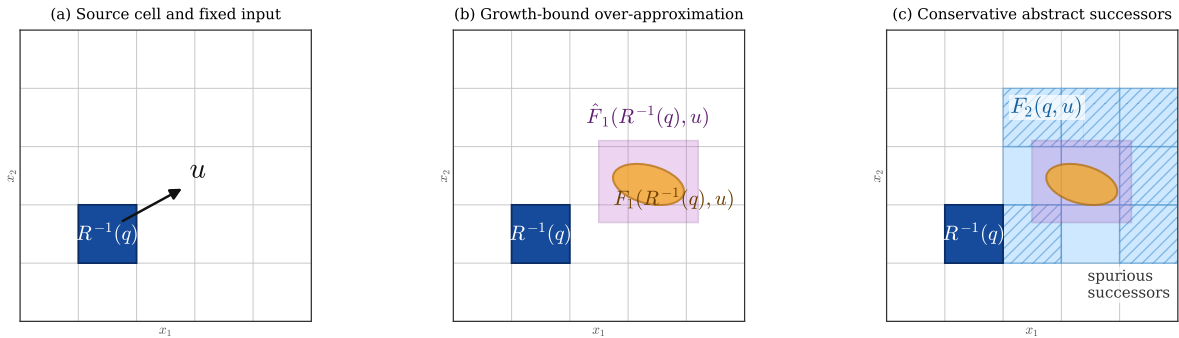


Figure 2.2: Starting from a source cell $R^{-1}(q)$ and a fixed input u , the one-step reachable set $F_1(R^{-1}(q), u)$ is over-approximated by a computable enclosure $\hat{F}_1(R^{-1}(q), u)$, here represented by a component-wise growth-bound box. Every grid cell intersecting this enclosure is added as an abstract successor of q . This preserves all concrete behaviours, but may introduce additional spurious transitions.

This explains the strength and the limitation of classical abstractions. The abstraction is conservative, hence suitable for formal synthesis, but conservative enclosures may enlarge successor sets and increase nondeterminism.

Compatibility with reach-avoid sets. For a reach-avoid problem with concrete sets $\mathcal{X}_I, \mathcal{X}_T, \mathcal{X}_O \subseteq \mathcal{X}_1$, a conservative symbolic specification can be chosen by selecting abstract sets $\mathcal{X}_{I2}, \mathcal{X}_{T2}, \mathcal{X}_{O2} \subseteq \mathcal{X}_2$ such that

$$\mathcal{X}_I \subseteq R^{-1}(\mathcal{X}_{I2}), \quad R^{-1}(\mathcal{X}_{T2}) \subseteq \mathcal{X}_T, \quad \mathcal{X}_O \subseteq R^{-1}(\mathcal{X}_{O2}).$$

The first inclusion ensures that every concrete initial state is represented abstractly, the second prevents declaring as target an abstract state containing non-target concrete states, and the third ensures that every unsafe concrete state is covered by an abstract obstacle state.

2.2.2 Controller synthesis and refinement

Once a finite abstraction has been constructed, the reach-avoid problem becomes a finite-state synthesis problem. The abstract controller must select inputs whose possible successors all remain inside a set of states from which the target can still be reached. This synthesis step is meaningful only because the abstract controller is later concretized into a controller for the original system, through a suitable control architecture or interface [19].

Definition 2.12 (Robust predecessor). Let $S = (\mathcal{X}, \mathcal{U}, F)$ be a transition system and let $Y \subseteq \mathcal{X}$. The robust predecessor of Y is

$$\text{Pre}_S(Y) := \{x \in \mathcal{X} : \exists u \in \mathcal{U}_S(x) \text{ such that } F(x, u) \subseteq Y\}. \quad (2.14)$$

The inclusion $F(x, u) \subseteq Y$ is robust: every possible successor induced by the input u must belong to Y .

Definition 2.13 (Reach-avoid winning set). Let $S = (\mathcal{X}, \mathcal{U}, F)$ be a finite transition system, and let $\mathcal{X}_T, \mathcal{X}_O \subseteq \mathcal{X}$ be respectively the target and obstacle sets, with $\mathcal{X}_T \cap \mathcal{X}_O = \emptyset$. Define

$$\mathcal{G}(Y) := \mathcal{X}_T \cup (\text{Pre}_S(Y) \cap (\mathcal{X} \setminus \mathcal{X}_O)). \quad (2.15)$$

Starting from $Y_0 = \mathcal{X}_T$, define recursively $Y_{i+1} = \mathcal{G}(Y_i)$. Since $\mathcal{X}$ is finite, the sequence stabilizes after finitely many steps. Its limit Y^* is called the reach-avoid winning set.

The set Y^* contains exactly the abstract states from which the target can be reached while avoiding the obstacle, under the worst-case interpretation of the transition map.

Definition 2.14 (Symbolic reach-avoid controller). Let Y^* be the reach-avoid winning set, and let $\rho(q) := \min\{i \in \mathbb{N} : q \in Y_i\}$ be the fixed-point rank of $q \in Y^*$. A symbolic reach-avoid controller is the set-valued map $C : \mathcal{X} \rightarrow 2^{\mathcal{U}}$ defined, for $q \in Y^* \setminus \mathcal{X}_T$, by

$$C(q) := \{u \in \mathcal{U}_S(q) : F(q, u) \subseteq Y_{\rho(q)-1}\}. \quad (2.16)$$

For $q \in \mathcal{X}_T$, the reach-avoid objective is already satisfied. For $q \notin Y^*$, we set $C(q) = \emptyset$.

The rank condition ensures progress toward the target: outside $\mathcal{X}_T$, each allowed input sends all possible successors to a lower-rank set.

Definition 2.15 (Memoryless concretization relation). Following [20, Def. 8]. Let $S_i = (\mathcal{X}_i, \mathcal{U}_i, F_i)$, $i \in \{1, 2\}$, be transition systems, and let $R \subseteq \mathcal{X}_1 \times \mathcal{X}_2$ be a strict relation, meaning that $R(x_1) \neq \emptyset$ for every $x_1 \in \mathcal{X}_1$. We say that R is a *memoryless concretization relation* from S_1 to S_2 , and write $S_1 \leq_R^{\text{MCR}} S_2$, if, for every $(x_1, x_2) \in R$ and every abstract input $u_2 \in \mathcal{U}_{S_2}(x_2)$, there exists a concrete input $u_1 \in \mathcal{U}_{S_1}(x_1)$ such that

$$R(F_1(x_1, u_1)) \subseteq F_2(x_2, u_2). \quad (2.17)$$

Here, for any subset $\Omega \subseteq \mathcal{X}_1$, we use the notation $R(\Omega) := \bigcup_{x_1 \in \Omega} R(x_1)$.

Definition 2.16 (Interface). Following [20, Def. 6]. Let R be a memoryless concretization relation from S_1 to S_2 . An interface associated with R is a set-valued map $I_R : \mathcal{X}_1 \times \mathcal{X}_2 \times \mathcal{U}_2 \rightarrow 2^{\mathcal{U}_1}$ such that, for every $(x_1, x_2) \in R$ and every $u_2 \in \mathcal{U}_{S_2}(x_2)$,

$$\emptyset \neq I_R(x_1, x_2, u_2) \subseteq \{u_1 \in \mathcal{U}_{S_1}(x_1) : R(F_1(x_1, u_1)) \subseteq F_2(x_2, u_2)\}.$$

Definition 2.17 (Memoryless concretized controller). Following [20, Def. 7]. Let $C_2 : \mathcal{X}_2 \rightarrow 2^{\mathcal{U}_2}$ be an abstract controller. The *memoryless concretization* of C_2 through R and I_R is the concrete controller $C_1 : \mathcal{X}_1 \rightarrow 2^{\mathcal{U}_1}$ defined by

$$C_1(x_1) := \bigcup_{x_2 \in R(x_1)} I_R(x_1, x_2, C_2(x_2)). \quad (2.18)$$

Equivalently, using the composition notation, $C_1 = C_2 \circ^{I_R} R$. The controller is memoryless because its admissible inputs are determined only from the current concrete state x_1 .

Connection with this thesis. The method developed in this thesis follows the same soundness principle: all admissible concrete behaviours are over-approximated before a guarantee is claimed. The difference is that the abstraction is not built uniformly over the whole state space. It is constructed locally around a nominal trajectory, using ellipsoidal sets and local feedback controllers. Thus, the global synthesis problem is replaced by a sequence of local transition certificates.

2.3 Local set-based abstractions

The previous section described symbolic abstractions from the viewpoint of finite transition systems and conservative transitions. Classical constructions often rely on uniform grids and representative constant inputs. In this section, we introduce a complementary viewpoint: abstract states may be arbitrary local sets, possibly non-uniform, and transitions may be labelled by local feedback controllers rather than by constant inputs.

This viewpoint is motivated by recent work on smart, hierarchical, and state-feedback abstractions [21–25].

Definition 2.18 (Local cover). Let $X_{\text{rel}} \subseteq \mathcal{X}$ be a region of interest. A local cover of X_{rel} is a finite family of sets $\mathcal{C} := \{X_q \subseteq \mathcal{X} : q \in Q\}$ such that $X_{\text{rel}} \subseteq \bigcup_{q \in Q} X_q$. Each index $q \in Q$ is interpreted as an abstract state and X_q as its concretization.

Unlike a uniform grid, the sets X_q need not have the same shape or size. They may overlap, be refined locally, or be constructed only in the part of the state space relevant for the specification [21, 22].

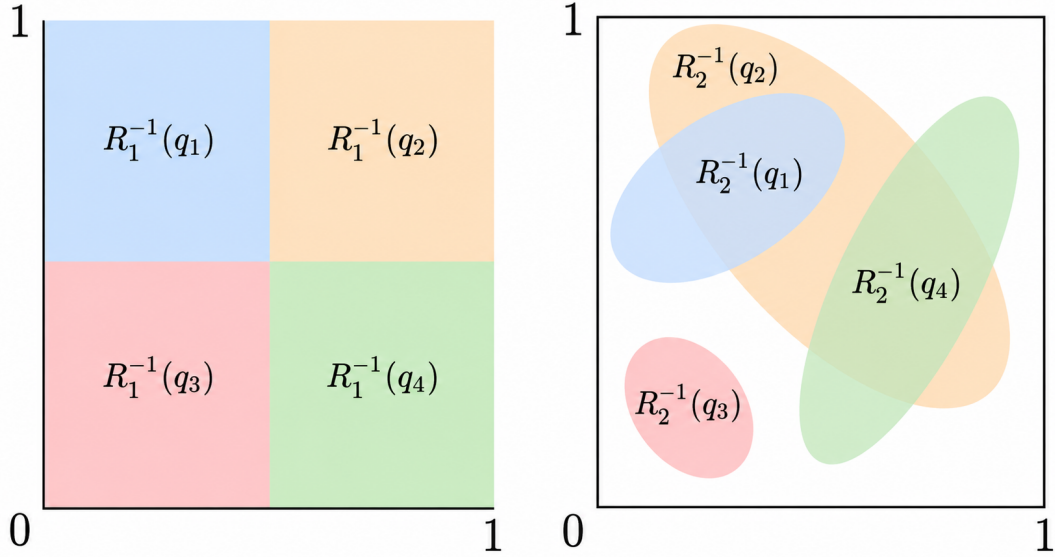


Figure 2.3: Partition and cover induced by concretization relations. Left: a strict relation R_1 induces a partition of the concrete state space into disjoint regions $R_1^{-1}(q_i)$. Right: a set-valued relation R_2 may induce an overlapping cover by concrete regions $R_2^{-1}(q_i)$. This illustrates the local-cover viewpoint used in Definition 2.18: an abstract state $q \in Q$ is represented by a concrete set $X_q \subseteq \mathcal{X}$, and these sets need not form a uniform partition of the whole state space. From [20].

Definition 2.19 (Feedback-labelled local transition). Let $S_1 = (\mathcal{X}, \mathcal{U}, F_1)$ be a transition system, and let $X_q, X_{q^+} \subseteq \mathcal{X}$ be two local sets. Let $\kappa : X_q \rightarrow \mathcal{U}$ be a feedback law. We denote the one-step closed-loop reachable set from X_q under κ by

$$F_1(X_q, \kappa) := \bigcup_{x \in X_q} F_1(x, \kappa(x)). \quad (2.19)$$

We say that κ certifies a local transition from X_q to X_{q^+} if

$$F_1(X_q, \kappa) \subseteq X_{q^+}. \quad (2.20)$$

Definition 2.20 (State-feedback abstraction). Let $S_1 = (\mathcal{X}, \mathcal{U}, F_1)$ be a transition system and let $\mathcal{C} = \{X_q : q \in Q\}$ be a local cover. A state-feedback abstraction is a finite transition system $S_2 = (Q, \mathcal{K}, F_2)$, where $\mathcal{K}$ is a finite set of local feedback controllers. A transition labelled by $\kappa \in \mathcal{K}$ is allowed from q to q^+ only if the corresponding

closed-loop concrete transition is certified, namely

$$q^+ \in F_2(q, \kappa) \implies F_1(X_q, \kappa) \subseteq X_{q^+}. \quad (2.21)$$

Definition 2.21 (Ellipsoidal abstract state). Let $c \in \mathbb{R}^n$ and let $P \in \mathbb{S}_{++}^n$. The ellipsoid centered at c with shape matrix P is $\mathcal{E}(c, P) := \{x \in \mathbb{R}^n : (x - c)^\top P^{-1}(x - c) \leq 1\}$. An ellipsoidal abstract state is an abstract state q whose concretization is an ellipsoid: $X_q = \mathcal{E}(c_q, P_q)$.

Definition 2.22 (Local affine feedback). Let $X_q \subseteq \mathbb{R}^n$ be a local abstract state equipped with a reference point c_q . A local affine feedback on X_q is a controller of the form

$$\kappa_q(x) = \ell_q + K_q(x - c_q), \quad x \in X_q,$$

where $\ell_q \in \mathbb{R}^m$, $K_q \in \mathbb{R}^{m \times n}$ is a feedback gain.

Remark 2.23 (Relation to this thesis). The definitions of this subsection should be understood as a thesis-specific adaptation of the local, non-uniform, and state-feedback abstraction viewpoints reviewed above. They are not meant to reproduce a single existing formalism verbatim, but to isolate the minimal notions needed for the trajectory-guided certificates used in this manuscript.

In this thesis, we use a partial local abstraction rather than a full abstraction of the state space. The local sets are ellipsoids, and each transition is certified by a local affine feedback law. The basic certificate is therefore

$$\text{Reach}(\mathcal{E}_k, \kappa_k) \subseteq \mathcal{E}_{k+1}.$$

Section 2.5 explains how such certificates are constructed in order to verify the inclusion condition introduced in Definitions 2.18–2.19–2.20.

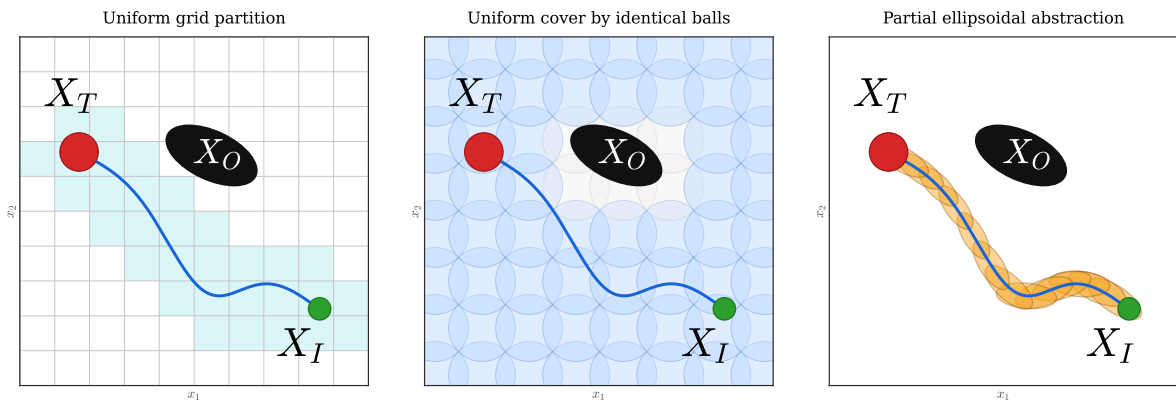


Figure 2.4: From uniform grid symbolic abstractions to local ellipsoidal abstractions. Left: a uniform grid partition of the state space. Middle: a non-rectangular cover, where abstract states need not form a partition. Right: the partial ellipsoidal viewpoint used in this thesis, where local sets are constructed only along a relevant trajectory.

2.4 Nominal trajectory generation

The previous section introduced local set-based abstractions, where abstract states are no longer required to form a uniform partition of the whole state space. A natural question is then where such local sets should be constructed.

Definition 2.24 (Nominal reach-avoid trajectory). Let $D_d = (\mathcal{X}, \mathcal{U}, \mathcal{W}, f)$ be a discrete-time uncertain control system, and let $\mathcal{X}_I, \mathcal{X}_T, \mathcal{X}_O \subseteq \mathcal{X}$ define a reach-avoid specification. Assume that $0 \in \mathcal{W}$. A finite sequence $(\bar{x}_{0:N}, \bar{u}_{0:N-1}) \in \mathcal{X}^{N+1} \times \mathcal{U}^N$ is called a nominal reach-avoid trajectory if

$$\bar{x}_{k+1} = f(\bar{x}_k, \bar{u}_k, 0), \quad k = 0, \dots, N-1, \quad (2.22)$$

and

$$\bar{x}_0 \in \mathcal{X}_I, \quad \bar{x}_N \in \mathcal{X}_T, \quad \bar{x}_k \in \mathcal{X} \setminus \mathcal{X}_O, \quad \bar{u}_k \in \mathcal{U}, \quad (2.23)$$

for all $k = 0, \dots, N-1$.

A nominal trajectory describes only one disturbance-free motion. Therefore, it does not solve the robust reach-avoid problem of Definition 2.6.

2.4.1 Nominal planning as finite-horizon optimal control

Nominal trajectory generation can be written as a finite-horizon optimal control problem.

$$\begin{aligned} \min_{x_{0:N}, u_{0:N-1}} \quad & \Phi(x_N) + \sum_{k=0}^{N-1} \ell(x_k, u_k) \\ \text{s.t.} \quad & x_{k+1} = f(x_k, u_k, 0), \quad k = 0, \dots, N-1, \\ & x_0 \in \mathcal{X}_I, \quad x_N \in \mathcal{X}_T, \\ & x_k \in \mathcal{X}_{safe}, \quad k = 0, \dots, N-1, \\ & u_k \in \mathcal{U}, \quad k = 0, \dots, N-1. \end{aligned} \quad (2.24)$$

The terminal cost Φ encourages reaching the target, while the running cost ℓ may encode path length, control effort, clearance from obstacles, smoothness, or proximity to prescribed waypoints. This formulation is broad enough to include direct trajectory optimization, nonlinear model predictive control, mixed-integer formulations for hybrid systems, and sampling-based model predictive methods [10, 12, 13, 26, 27].

2.4.2 Families of nominal planners

Several families of algorithms can be used to generate the nominal trajectory. Their outputs play the same role in the architecture of this thesis: they provide a candidate motion, while the formal guarantee is deferred to the certification step.

Method	Principle	Advantages
Sampling-based planning	Explore the state space through random samples and forward simulations.	Efficient exploration of complex spaces; probabilistic completeness for RRT-type methods; asymptotic optimality for RRT* variants [28, 29].
Direct trajectory optimization and nonlinear MPC	Solve a constrained finite-horizon optimal control problem.	Constraints and smoothness objectives can be encoded explicitly; natural framework for performance optimization [10, 12, 26].
Hybrid and mixed-integer formulations	Encode dynamics, logic, and constraints through mixed logical dynamical or mixed-integer models.	Useful when mode switches, logical constraints, or piecewise-affine dynamics are central to the problem [27].
Sampling-based MPC and MPPI	Sample many perturbed control sequences, simulate rollouts, and update the nominal sequence from low-cost samples.	Derivative-free, easy to implement for nonlinear dynamics, parallelizable, and well suited to warm starts [13, 30–32].
External trajectories	Use a trajectory produced by a simulator, a hand-designed manoeuvre, a learned policy, or an experiment.	Keeps the certification pipeline modular; useful when a good candidate motion is already available.

Table 2.1: Possible nominal trajectory generators. In the proposed architecture, the nominal planner can be replaced without changing the certification principle: the planner proposes a candidate motion, whereas the local certifier proves the robust reach-avoid property.

2.4.3 Model Predictive Path Integral control

Model Predictive Path Integral control is a sampling-based model predictive control method derived from path-integral and information-theoretic viewpoints on stochastic optimal control [13, 14, 30–32]. In its practical finite-horizon form, MPPI maintains a nominal control sequence

$$U := (\bar{u}_0, \dots, \bar{u}_{H-1}) \in \mathcal{U}^H, \quad (2.25)$$

where H is the planning horizon. At each iteration, M perturbation sequences are sampled:

$$\varepsilon_{0:H-1}^{(i)}, \quad \varepsilon_k^{(i)} \sim \mathcal{N}(0, \Sigma), \quad i = 1, \dots, M. \quad (2.26)$$

The perturbed controls are then applied to the nominal dynamics:

$$\begin{aligned} u_k^{(i)} &= \Pi_{\mathcal{U}} \left(\bar{u}_k + \varepsilon_k^{(i)} \right), \\ x_{k+1}^{(i)} &= f(x_k^{(i)}, u_k^{(i)}, 0), \quad k = 0, \dots, H-1, \end{aligned} \quad (2.27)$$

where $\Pi_{\mathcal{U}}$ denotes projection or saturation onto the admissible input set. Each rollout receives a cost

$$S_i = \Phi(x_H^{(i)}) + \sum_{k=0}^{H-1} \ell(x_k^{(i)}, u_k^{(i)}). \quad (2.28)$$

Low-cost rollouts are given larger weights through an exponential transformation:

$$\omega_i = \frac{\exp \left(-\frac{1}{\lambda} (S_i - S_{\min}) \right)}{\sum_{j=1}^M \exp \left(-\frac{1}{\lambda} (S_j - S_{\min}) \right)}, \quad S_{\min} := \min_{1 \leq j \leq M} S_j. \quad (2.29)$$

The subtraction of $S_{\min}$ is a numerical stabilization and does not change the normalized weights. The parameter $\lambda > 0$ is the temperature: small values concentrate the update on the best rollouts, whereas larger values average over a wider set of samples.

The nominal control sequence is updated by the weighted perturbations:

$$\bar{u}_k \leftarrow \Pi_{\mathcal{U}} \left(\bar{u}_k + \sum_{i=1}^M \omega_i \varepsilon_k^{(i)} \right), \quad k = 0, \dots, H-1. \quad (2.30)$$

Thus, MPPI can be viewed as a stochastic search procedure in control-sequence space. Its main computational advantage is that the rollout simulations in (2.27) are independent, which makes the method naturally parallelizable [13, 32].

Warm starts can be incorporated in MPPI by initializing the nominal control sequence, or the sampling distribution, around a previously computed trajectory. This reduces the amount of exploration needed to find a useful motion. For instance, RRT-guided MPPI uses an RRT path to guide the MPPI sampling process [33]. In this thesis, the warm start is provided by a coarse centered simulation abstraction.

Remark 2.25 (Role in this thesis). The nominal planner used in this thesis is only a proposal mechanism. It selects a candidate trajectory $(\bar{x}_{0:N}, \bar{u}_{0:N-1})$ and therefore a candidate region where a partial abstraction may be constructed. The concrete instantiation of the nominal planner, including the centered-abstraction warm start and the MPPI refinement, is described in Chapter 4.

2.5 Local certification tools

The previous sections introduced local set-based abstractions and feedback-labelled transitions. We now recall the local certificate used later in the thesis to verify such transitions when the local sets are ellipsoids. The presentation follows the ellipsoidal state-feedback construction of [21], but only keeps the certificate needed in the main text. The LMI realization of this certificate is given in Appendix A.

Let $S = (\mathcal{X}, \mathcal{U}, F)$ be the transition system associated with the uncertain discrete-time dynamics introduced in Definition 2.3. Thus, for a state x and an input u , $F(x, u)$ denotes the set of all one-step successors generated by the admissible disturbances.

Definition 2.26 (Local transition certificate). Let $\xi, \xi^+ \subseteq \mathcal{X}$ be two local sets, and let $\kappa : \xi \rightarrow \mathcal{U}$ be a feedback law. We say that κ certifies a local transition from ξ to ξ^+ if

$$\{x^+ \in F(x, \kappa(x)) : x \in \xi\} \subseteq \xi^+. \quad (2.31)$$

Equivalently, in the uncertain dynamics notation,

$$f(x, \kappa(x), w) \in \xi^+, \quad \forall x \in \xi, \quad \forall w \in \mathcal{W}. \quad (2.32)$$

In the state-feedback abstraction viewpoint, a certified transition is therefore a deterministic abstract transition labelled by a feedback controller: applying κ from any state in the source set ξ sends all admissible concrete successors into the target set ξ^+ . In a reach-avoid problem, this transition is useful only when it is also compatible with the constraints, namely

$$\xi \cap \mathcal{X}_O = \emptyset, \quad \kappa(\xi) \subseteq \mathcal{U}. \quad (2.33)$$

The local construction used in [21] considers a fixed successor ellipsoid ξ^+ and a candidate predecessor center c . The goal is to design both a predecessor ellipsoid

$$\xi = \mathcal{E}(c, P) \quad (2.34)$$

and a local affine feedback law

$$\kappa(x) = K(x - c) + \ell, \quad x \in \xi, \quad (2.35)$$

such that the transition condition (2.32) holds. Thus, the geometry of the cell and the feedback law acting inside it are designed together.

The exact nonlinear inclusion is usually not checked directly. Instead, the discrete map f is approximated locally around a point $\bar{p} = (c, \bar{u}, 0)$ by an affine map

$$\tilde{f}(x, u, w) = Ax + Bu + Ew + g, \quad (2.36)$$

where A , B , and E are the Jacobians of f with respect to x , u , and w at $\bar{p}$. Substituting the feedback law gives the nominal closed-loop affine image

$$q(x) := \tilde{f}(x, \kappa(x), 0). \quad (2.37)$$

The nonlinear part neglected by the affine model is bounded by a common hyperrectangle $\mathcal{H}_r$. Under the Lipschitz-Jacobian assumption used in [21], one obtains a bound of the form

$$\mathcal{H}_r := H\left(0, \frac{1}{2}L_f r^2\right), \quad r^2 = \delta_X + \delta_U + \delta_W, \quad (2.38)$$

where $L_f \in \mathbb{R}_+^n$ collects componentwise Lipschitz bounds, and where δ_X , δ_U , and δ_W bound respectively the state deviation, the input deviation induced by the controller, and the disturbance magnitude on the local certification region.

Consequently, for all $x \in \xi$ and $w \in \mathcal{W}$,

$$f(x, \kappa(x), w) \in \{q(x)\} \oplus E\mathcal{W} \oplus \mathcal{H}_r. \quad (2.39)$$

The local nonlinear transition is therefore certified by the stronger set inclusion

$$q(\xi) \oplus E\mathcal{W} \oplus \mathcal{H}_r \subseteq \xi^+. \quad (2.40)$$

This inclusion is the geometric core of the certificate. The set $q(\xi)$ is the closed-loop image of the source ellipsoid through the local affine model, $E\mathcal{W}$ accounts for bounded disturbances, and $\mathcal{H}_r$ accounts for the linearization error. Hence, if (2.40) holds, then the true nonlinear transition condition (2.32) holds.

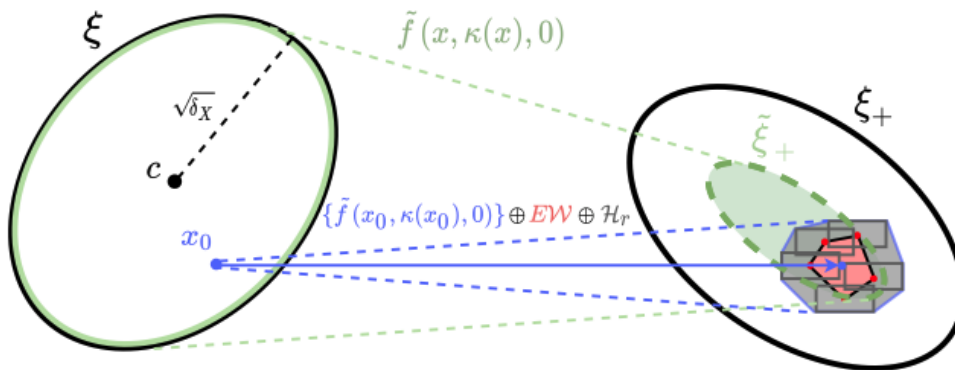


Figure 2.5: Local ellipsoidal transition certificate, from [21]. A predecessor ellipsoid ξ is equipped with a local affine feedback law κ . Under the local affine model, its closed-loop image is $q(\xi)$. The disturbance contribution $E\mathcal{W}$ and the linearization-error bound $\mathcal{H}_r$ are then added through a Minkowski sum. Certifying $q(\xi) \oplus E\mathcal{W} \oplus \mathcal{H}_r \subseteq \xi^+$ implies that every admissible nonlinear successor $f(x, \kappa(x), w)$, with $x \in \xi$ and $w \in \mathcal{W}$, belongs to the successor ellipsoid ξ^+ .

In practice, the inclusion (2.40), the input constraint $\kappa(\xi) \subseteq \mathcal{U}$, and the consistency conditions defining the local linearization region are enforced by sufficient LMIs obtained from the S-procedure and the Schur complement [34]. These algebraic details are collected in Appendix A. In the rest of the thesis, a feasible solution of this local SDP is interpreted as a valid certificate for the transition $\xi \rightarrow \xi^+$.

2.6 Positioning of this thesis

This thesis lies at the intersection of two lines of work: abstraction-based control and trajectory-centered robust certification.

- **Abstraction-based control.** Classical abstraction-based control replaces a continuous-state control problem by a finite transition system on which correct-by-construction synthesis can be performed [7, 8]. Its central strength is soundness: a symbolic controller can be refined only if the abstraction conservatively represents all admissible concrete behaviours. Recent smart and state-feedback abstractions reduce conservatism by using non-uniform cells, local covers, and feedback-labelled transitions [21, 23]. This thesis follows this local viewpoint, but does not attempt to construct a complete abstraction of the whole state space. Instead, it builds only the part of the abstraction needed along a candidate motion.
- **Trajectory-centered robust certification.** Tube-based robust MPC, reachability-based optimal control, safe tracking, and funnel methods start from a nominal trajectory and certify a set of behaviours around it [11, 12, 25, 26, 35–39]. Depending on the method, the certified local sets may be ellipsoids, boxes, polytopes, zonotopes, backward reachable sets, or Lyapunov sublevel sets. The corresponding certificates may be obtained by reachability analysis, linear or semidefinite programming, mixed-integer optimization, Hamilton–Jacobi methods, or sum-of-squares programming. This thesis adopts the trajectory-centered philosophy, but interprets each local certificate as a transition of a partial feedback abstraction.

Several related approaches are close in spirit. Fan et al. [6] combine reference trajectory generation, tracking control, and ellipsoidal reachable-set over-approximations for linear reach-avoid problems. Schäfer and Althoff [35] combine nominal optimal control with set-based reachability analysis to enforce robust constraint satisfaction. Serry et al. [36] compute zonotopic backward reachable sets for safe tracking of discrete-time nonlinear uncertain systems. Funnel-based methods instead certify neighbourhoods of nominal trajectories using Lyapunov or sum-of-squares arguments [37].

The distinction is that this thesis keeps the abstraction-based interpretation explicit. The certified ellipsoids are not only a tracking tube around a trajectory; they are local abstract states. The local affine controllers are not only stabilizing feedback laws; they are transition labels. The output is therefore a finite chain of certified feedback transitions, which can be read both as a robust trajectory-centered certificate and as a partial local abstraction solving the reach-avoid specification on a certified subset of initial states.

Problem Formulation

This chapter fixes the mathematical problem addressed in the rest of the thesis. The objective is to certify a robust reach-avoid controller on a subset of the initial set, using local ellipsoidal sets and local affine feedback laws.

3.1 System and specification

We consider a discrete-time uncertain nonlinear control system

$$x_{k+1} = f(x_k, u_k, w_k), \quad x_k \in \mathcal{X}, \quad u_k \in \mathcal{U}, \quad w_k \in \mathcal{W}, \quad (3.1)$$

where $\mathcal{X} \subseteq \mathbb{R}^n$, $\mathcal{U} \subseteq \mathbb{R}^m$, and $\mathcal{W} \subseteq \mathbb{R}^p$. The associated transition system is $S = (\mathcal{X}, \mathcal{U}, F)$, with $F(x, u) := \{f(x, u, w) : w \in \mathcal{W}\}$.

The reach-avoid specification is given by an initial set $\mathcal{X}_I \subseteq \mathcal{X}$, a target set $\mathcal{X}_T \subseteq \mathcal{X}$, and an obstacle set $\mathcal{X}_O \subseteq \mathcal{X}$, satisfying $\mathcal{X}_T \cap \mathcal{X}_O = \emptyset$.

We collect the problem data as

$$\mathcal{P} := (f, \mathcal{X}, \mathcal{U}, \mathcal{W}, \mathcal{X}_I, \mathcal{X}_T, \mathcal{X}_O). \quad (3.2)$$

If the original model is continuous-time, then f denotes the sampled discrete-time map with sampling time $\tau > 0$. In the implementation, this map is computed with a fourth-order Runge–Kutta scheme; all certificates are stated for the resulting discrete-time model.

3.2 Certified local reach-avoid chains

For a set $\Omega \subseteq \mathcal{X}$ and a feedback law $\kappa : \Omega \rightarrow \mathbb{R}^m$, define the one-step closed-loop reachable set

$$\text{Reach}_f(\Omega, \kappa) := \{f(x, \kappa(x), w) : x \in \Omega, w \in \mathcal{W}\}. \quad (3.3)$$

Definition 3.1 (Certified local reach-avoid chain). A certified local reach-avoid chain of horizon N is a tuple

$$\mathfrak{C}_N := ((\Omega_k)_{k=0}^N, (\kappa_k)_{k=0}^{N-1}), \quad (3.4)$$

where $\Omega_k \subseteq \mathcal{X}$ and $\kappa_k : \Omega_k \rightarrow \mathbb{R}^m$, such that

$$\Omega_0 \cap \mathcal{X}_I \neq \emptyset, \quad (3.5)$$

$$\Omega_k \cap \mathcal{X}_O = \emptyset, \quad k = 0, \dots, N-1, \quad (3.6)$$

$$\Omega_N \subseteq \mathcal{X}_T, \quad (3.7)$$

$$\kappa_k(\Omega_k) \subseteq \mathcal{U}, \quad k = 0, \dots, N-1, \quad (3.8)$$

$$\text{Reach}_f(\Omega_k, \kappa_k) \subseteq \Omega_{k+1}, \quad k = 0, \dots, N-1. \quad (3.9)$$

The certified initial set induced by the chain is $\mathcal{X}_I^{\text{cert}}(\mathfrak{C}_N) := \Omega_0 \cap \mathcal{X}_I$.

Definition 3.2 (Ellipsoidal certified chain). An ellipsoidal certified chain is a certified local reach-avoid chain for which $\Omega_k = \mathcal{E}_k = \mathcal{E}(c_k, P_k)$, and each transition is labelled by a local affine feedback law $\kappa_k(x) = \ell_k + K_k(x - c_k)$.

Proposition 3.3 (Soundness of a certified local chain). Let $\mathfrak{C}_N = ((\Omega_k)_{k=0}^N, (\kappa_k)_{k=0}^{N-1})$ be a certified local reach-avoid chain. Then, for every $x_0 \in \mathcal{X}_I^{\text{cert}}(\mathfrak{C}_N)$ and every disturbance sequence $(w_0, \dots, w_{N-1}) \in \mathcal{W}^N$, the trajectory defined by

$$x_{k+1} = f(x_k, \kappa_k(x_k), w_k), \quad k = 0, \dots, N-1, \quad (3.10)$$

is well-defined and satisfies

$$x_k \in \Omega_k, \quad k = 0, \dots, N, \quad (3.11)$$

$$x_k \notin \mathcal{X}_O, \quad k = 0, \dots, N, \quad (3.12)$$

$$x_N \in \mathcal{X}_T. \quad (3.13)$$

Consequently, the finite-horizon feedback policy $\pi = (\kappa_0, \dots, \kappa_{N-1})$, applied as $u_k = \kappa_k(x_k)$, solves the robust reach-avoid specification on the certified initial set $\mathcal{X}_I^{\text{cert}}(\mathfrak{C}_N)$.

Proof. Let $x_0 \in \Omega_0 \cap \mathcal{X}_I$ and let $(w_0, \dots, w_{N-1}) \in \mathcal{W}^N$. We prove by induction that $x_k \in \Omega_k$ for every $k = 0, \dots, N$. The property is true for $k = 0$ by definition. Assume that $x_k \in \Omega_k$ for some $k < N$. Since κ_k is defined on Ω_k and $\kappa_k(\Omega_k) \subseteq \mathcal{U}$, the input $\kappa_k(x_k)$ is well-defined and admissible. Moreover, by definition of the reachable set,

$$x_{k+1} = f(x_k, \kappa_k(x_k), w_k) \in \text{Reach}_f(\Omega_k, \kappa_k). \quad (3.14)$$

The certified transition condition gives

$$\text{Reach}_f(\Omega_k, \kappa_k) \subseteq \Omega_{k+1}, \quad (3.15)$$

hence $x_{k+1} \in \Omega_{k+1}$. This proves the induction.

The safety condition follows from $\Omega_k \cap \mathcal{X}_O = \emptyset$ for $k = 0, \dots, N-1$. Terminal reachability follows from $\Omega_N \subseteq \mathcal{X}_T$. Since $\mathcal{X}_T \cap \mathcal{X}_O = \emptyset$, we also have $x_N \notin \mathcal{X}_O$. Therefore the trajectory reaches $\mathcal{X}_T$ at time N while avoiding $\mathcal{X}_O$ for all $k = 0, \dots, N$. Since the argument holds for every disturbance sequence in $\mathcal{W}^N$, the guarantee is robust. $\square$

Remark 3.4 (Abstraction viewpoint). A certified local reach-avoid chain can be read from an abstraction-based control viewpoint. The sets Ω_k play the role of local abstract states, and the feedback laws κ_k play the role of transition labels. However, this abstraction is not global: it only represents the behaviours certified along the chain, on the covered domain $\bigcup_{k=0}^N \Omega_k$. It is therefore a partial, local, and task-oriented abstraction of the concrete system.

Proposition 3.5 (Certified local chain as a partial state-feedback abstraction). *Let $\mathfrak{C}_N = ((\Omega_k)_{k=0}^N, (\kappa_k)_{k=0}^{N-1})$ be a certified local reach-avoid chain in the sense of Definition 3.1. Associate with each set Ω_k an abstract state q_k , and consider the finite path*

$$q_0 \xrightarrow{\kappa_0} q_1 \xrightarrow{\kappa_1} \dots \xrightarrow{\kappa_{N-1}} q_N. \quad (3.16)$$

Define $Q_{\mathfrak{C}} := \{q_0, \dots, q_N\}$, the local cover $X_{q_k} := \Omega_k$, and the finite set of feedback labels $\mathcal{K}_{\mathfrak{C}} := \{\kappa_0, \dots, \kappa_{N-1}\}$. Define the abstract transition map by

$$F_{\mathfrak{C}}(q_k, \kappa_k) := \{q_{k+1}\}, \quad k = 0, \dots, N-1, \quad (3.17)$$

with all other transitions empty. Then $S_{\mathfrak{C}} := (Q_{\mathfrak{C}}, \mathcal{K}_{\mathfrak{C}}, F_{\mathfrak{C}})$ is a partial state-feedback abstraction of the concrete system, in the sense of Definition 2.20, on the domain covered by the chain.

Proof. For every $k = 0, \dots, N-1$, the certified-chain condition gives

$$\text{Reach}_f(\Omega_k, \kappa_k) \subseteq \Omega_{k+1}. \quad (3.18)$$

Since $X_{q_k} = \Omega_k$ and $X_{q_{k+1}} = \Omega_{k+1}$, this is exactly

$$\text{Reach}_f(X_{q_k}, \kappa_k) \subseteq X_{q_{k+1}}. \quad (3.19)$$

Thus every transition $q_k \xrightarrow{\kappa_k} q_{k+1}$ satisfies the feedback-labelled transition condition of Definition 2.20. Since the construction only contains the states and transitions of the certified chain, the resulting abstraction is partial. $\square$

Concretization. The previous proposition shows that the certified chain defines a partial state-feedback abstraction. We now relate this partial abstraction to the concrete system through the memoryless concretization relation introduced in Definition 2.15. The only technical point is that the concrete state is augmented with the time index k , so that the finite-horizon feedback sequence becomes memoryless.

Proposition 3.6 (MCR interpretation of a certified chain). *Let $\mathfrak{C}_N$ be a certified local reach-avoid chain, and let $S_{\mathfrak{C}}$ be the partial state-feedback abstraction defined in Proposition 3.5. Define the time-augmented certified domain*

$$\mathcal{Z}_{\mathfrak{C}} := \bigcup_{k=0}^N \Omega_k \times \{k\}. \quad (3.20)$$

Associate each $(x, k) \in \mathcal{Z}_\mathbb{C}$ with the abstract state q_k , namely

$$((x, k), q_j) \in R_\mathbb{C} \iff j = k. \quad (3.21)$$

Then $R_\mathbb{C}$ is a memoryless concretization relation on the certified time-augmented domain. An associated interface is given by

$$I_{R_\mathbb{C}}((x, k), q_k, \kappa_k) := \{\kappa_k(x)\}, \quad k = 0, \dots, N-1. \quad (3.22)$$

Therefore, the abstract path controller $C_\mathbb{C}(q_k) = \{\kappa_k\}$ is concretized as the concrete finite-horizon feedback law

$$C_\mathbb{C}^c(x, k) = \{\kappa_k(x)\}. \quad (3.23)$$

Proof. The relation $R_\mathbb{C}$ is strict on $\mathcal{Z}_\mathbb{C}$, since every $(x, k) \in \mathcal{Z}_\mathbb{C}$ is related to q_k .

Let $((x, k), q_k) \in R_\mathbb{C}$, with $k < N$. The only abstract input available at q_k in the path abstraction is the feedback label κ_k . The interface selects the concrete input $u = \kappa_k(x)$. Since $\mathbb{C}_N$ is certified,

$$F(x, \kappa_k(x)) \subseteq \Omega_{k+1}. \quad (3.24)$$

Hence every concrete successor $x^+ \in F(x, \kappa_k(x))$ satisfies $((x^+, k+1), q_{k+1}) \in R_\mathbb{C}$. Therefore,

$$R_\mathbb{C}(F(x, \kappa_k(x)) \times \{k+1\}) \subseteq \{q_{k+1}\} = F_\mathbb{C}(q_k, \kappa_k). \quad (3.25)$$

This is precisely the MCR condition of Definition 2.15, restricted to the certified time-augmented domain. The expression of $C_\mathbb{C}^c$ follows directly from the definition of a memoryless concretized controller, Definition 2.17. $\square$

3.3 Standing assumptions

We use the following assumptions throughout the manuscript. Assumptions (A1)–(A3) define the general reach-avoid setting, while Assumptions (A4)–(A9) are specific to the ellipsoidal LMI-based certification step used in the implementation.

(A1) State, input, and disturbance sets. The state, input, and disturbance sets satisfy

$$\mathcal{X} \subseteq \mathbb{R}^n, \quad \mathcal{U} \subseteq \mathbb{R}^m, \quad \mathcal{W} \subseteq \mathbb{R}^p,$$

and are nonempty. The sets $\mathcal{U}$ and $\mathcal{W}$ are compact, and $0 \in \mathcal{W}$.

(A2) Dynamics. The discrete-time transition map f is well-defined on an open neighbourhood of every region used for planning and certification. The associated set-valued transition map is

$$F(x, u) := \{f(x, u, w) : w \in \mathcal{W}\}. \quad (3.26)$$

(A3) Specification sets. The reach-avoid specification is defined by sets

$$\mathcal{X}_I, \mathcal{X}_T, \mathcal{X}_O \subseteq \mathcal{X},$$

where $\mathcal{X}_I$ is the initial set, $\mathcal{X}_T$ is the target set, and $\mathcal{X}_O$ is the obstacle set. We assume

$$\mathcal{X}_T \cap \mathcal{X}_O = \emptyset. \quad (3.27)$$

These sets are closed, or at least represented in a form for which all inclusions and disjointness conditions used by the certificate can be checked soundly. We denote the safe set by $\mathcal{X}_{\text{safe}} := \mathcal{X} \setminus \mathcal{X}_O$.

(A4) Local regularity. For each certified transition $k = 0, \dots, N - 1$, there exists a compact local certification region

$$\mathcal{B}_k := \mathcal{Y}_k \times \mathcal{V}_k \times \mathcal{W}, \quad \mathcal{Y}_k \subseteq \mathcal{X}, \quad \mathcal{V}_k \subseteq \mathcal{U}, \quad (3.28)$$

on which f is continuously differentiable. On this region, the Jacobian of f admits a sound componentwise Lipschitz bound.

(A5) Local consistency. The linearization point used for the k -th local certificate satisfies

$$(c_k, \bar{u}_k, 0) \in \mathcal{B}_k. \quad (3.29)$$

Moreover, any returned ellipsoidal certificate must be consistent with the local region on which the model and the bounds are valid:

$$\mathcal{E}_k \subseteq \mathcal{Y}_k, \quad \kappa_k(\mathcal{E}_k) \subseteq \mathcal{V}_k. \quad (3.30)$$

(A6) Disturbance representation. For the SDP certificate, the disturbance set is represented as a convex polytope

$$\mathcal{W} = \text{co}\{w^1, \dots, w^{N_w}\}, \quad 0 \in \mathcal{W}. \quad (3.31)$$

The deterministic case corresponds to the special case $\mathcal{W} = \{0\}$.

(A7) Input representation. The input set is assumed to be an intersection of possibly degenerate ellipsoids:

$$\mathcal{U} = \bigcap_{a=1}^{N_u} \mathcal{U}_a, \quad \mathcal{U}_a := \{u \in \mathbb{R}^m : \|U_a u\|_2 \leq 1\}, \quad (3.32)$$

for some matrices U_a of appropriate dimensions.

(A8) Sound local remainder. For each transition, the hyperrectangle $\mathcal{H}_{r,k}$ used in the LMI certificate is a sound over-approximation of the nonlinear remainder on $\mathcal{B}_k$. Equivalently, if $\tilde{f}_k$ denotes the affine linearization of f around $(c_k, \bar{u}_k, 0)$, then the bound satisfies

$$f(x, u, w) \in \{\tilde{f}_k(x, u, w)\} \oplus \mathcal{H}_{r,k}, \quad \forall (x, u, w) \in \mathcal{B}_k. \quad (3.33)$$

(A9) Discrete-time model. If f is obtained by discretizing a continuous-time system, then all certificates are stated for the resulting sampled-data map. In the experiments, this map is the fourth-order Runge–Kutta recursion with the chosen sampling time. The RK4 recursion is assumed to be well-defined on the local certification regions used by the planner and the certifier.

3.4 Nominal planning and certification problems

The trajectory-guided pipeline separates the computation into two subproblems. The first one is a nominal planning problem. The second one is a robust certification problem.

Problem 3.1 (Nominal reach-avoid planning). Given

$$\mathcal{P} := (f, \mathcal{X}, \mathcal{U}, \mathcal{W}, \mathcal{X}_I, \mathcal{X}_T, \mathcal{X}_O),$$

find a horizon N and a sequence

$$\tau_{\text{nom}} := (\bar{x}_{0:N}, \bar{u}_{0:N-1}) \in \mathcal{X}^{N+1} \times \mathcal{U}^N \quad (3.34)$$

such that

$$\bar{x}_0 \in \mathcal{X}_I, \quad (3.35)$$

$$\bar{x}_N \in \mathcal{X}_T, \quad (3.36)$$

$$\bar{x}_k \in \mathcal{X}_{\text{safe}}, \quad k = 0, \dots, N-1, \quad (3.37)$$

$$\bar{x}_{k+1} = f(\bar{x}_k, \bar{u}_k, 0), \quad k = 0, \dots, N-1. \quad (3.38)$$

Remark 3.7 (Optimization viewpoint). Problem 3.1 is a feasibility problem. In practice, a nominal guide can be selected by minimizing a finite-horizon cost, as in Section 2.4.1. Ideally, this cost should not only produce a feasible nominal motion, but also favour trajectories that admit low-cost or large robust certificates.

Problem 3.2 (Certification of a nominal trajectory). Given the reach-avoid problem data

$$\mathcal{P} := (f, \mathcal{X}, \mathcal{U}, \mathcal{W}, \mathcal{X}_I, \mathcal{X}_T, \mathcal{X}_O),$$

and a nominal trajectory

$$\tau_{\text{nom}} := (\bar{x}_{0:N}, \bar{u}_{0:N-1}) \in \mathcal{X}^{N+1} \times \mathcal{U}^N,$$

find a certified local reach-avoid chain

$$\mathfrak{C}_N := ((\Omega_k)_{k=0}^N, (\kappa_k)_{k=0}^{N-1})$$

such that

$$\bar{x}_k \in \Omega_k, \quad k = 0, \dots, N. \quad (3.39)$$

$$\Omega_0 \cap \mathcal{X}_I \neq \emptyset, \quad (3.40)$$

$$\Omega_k \subseteq \mathcal{X}_{\text{safe}}, \quad k = 0, \dots, N-1, \quad (3.41)$$

$$\Omega_N \subseteq \mathcal{X}_T, \quad (3.42)$$

$$\kappa_k(\Omega_k) \subseteq \mathcal{U}, \quad k = 0, \dots, N-1, \quad (3.43)$$

$$\text{Reach}_f(\Omega_k, \kappa_k) \subseteq \Omega_{k+1}, \quad k = 0, \dots, N-1. \quad (3.44)$$

Remark 3.8 (Ellipsoidal instantiation). In this thesis, Problem 3.2 is instantiated with ellipsoidal local sets and local affine feedback laws, namely $\Omega_k = \mathcal{E}_k$ and $\kappa_k(x) = \ell_k + K_k(x - c_k)$.

Problem 3.3 (Trajectory-guided certified synthesis). Given

$$(f, \mathcal{X}, \mathcal{U}, \mathcal{W}, \mathcal{X}_I, \mathcal{X}_T, \mathcal{X}_O),$$

compute either

$$(\bar{x}_{0:N}, \bar{u}_{0:N-1}, \mathfrak{C}_N) \quad (3.45)$$

such that

1. $\tau_{\text{nom}} := (\bar{x}_{0:N}, \bar{u}_{0:N-1})$ solves Problem 3.1;
2. $\mathfrak{C}_N$ solves Problem 3.2;

or return *failure*.

Remark 3.9 (Certified chains as controllers and partial abstractions). Only the certified chain $\mathfrak{C}_N$ carries a formal guarantee. Through the feedback sequence $(\kappa_0, \dots, \kappa_{N-1})$, it induces a sound finite-horizon controller on the certified initial set $\mathcal{X}_I^{\text{cert}}(\mathfrak{C}_N) := \Omega_0 \cap \mathcal{X}_I$, as stated in Proposition 3.3.

The same object also admits an abstraction-based interpretation. One may view Problem 3.3 either as constructing a robust certificate around a nominal trajectory, or as constructing a sound partial state-feedback abstraction guided by that trajectory.

A Modular Trajectory-Guided Certification Pipeline

This chapter presents the main contribution of the thesis: a modular pipeline for trajectory-guided certified synthesis. The mathematical problem was formulated in Chapter 3; the purpose of the chapter is to describe how this problem is solved in practice.

We denote by

$$\mathcal{P} := (f, \mathcal{X}, \mathcal{U}, \mathcal{W}, \mathcal{X}_I, \mathcal{X}_T, \mathcal{X}_O) \quad (4.1)$$

the reach-avoid problem data introduced in Chapter 3. The goal is to build a solver for Problem 3.3: given $\mathcal{P}$, the solver should either return a nominal trajectory together with a certified reach-avoid chain, or report failure.

The proposed solver follows the same decomposition as the problem formulation. First, a planning module computes a nominal trajectory solving Problem 3.1. This stage is heuristic: it is used to identify a promising region of the state space, but it does not provide a robust guarantee. Second, a certification module attempts to solve Problem 3.2 along the nominal trajectory. This second stage is the source of the formal guarantee.

In the spirit of DIONYSOS.JL [9], the objective is not only to implement a particular certification method, but to organize the solver as a flexible framework in which trajectory generation and certification are independent components. Different nominal planners can therefore be combined with different local certifiers without changing the global problem statement.

4.1 Overview of the proposed pipeline

Algorithm 1 summarizes the high-level structure of the solver. Besides the problem data $\mathcal{P}$, the algorithm takes two implementation-level configurations. The planner configuration Π_{plan} contains the parameters of the nominal trajectory generator. The certifier configuration Π_{cert} contains the parameters of the local certification routine.

The subroutine PLAN is required to produce a nominal trajectory satisfying Problem 3.1. The subroutine CERTIFY is required to return a certified chain satisfying Problem 3.2. Consequently, whenever Algorithm 1 returns a certified solution, the output solves Problem 3.3, and the reach-avoid guarantee follows from Proposition 3.3.

Algorithm 1 Trajectory-guided certified synthesis**Require:** problem data $\mathcal{P}$, planner configuration Π_{plan} , certifier configuration Π_{cert} **Ensure:** either a certified solution $(\bar{x}_{0:N}, \bar{u}_{0:N-1}, \mathfrak{C}_N)$ or FAILURE**Nominal planning**1: $(\bar{x}_{0:N}, \bar{u}_{0:N-1}) \leftarrow \text{PLAN}(\mathcal{P}, \Pi_{\text{plan}})$ 2: **if** PLAN fails **then**3: **return** FAILURE4: **end if****Local certification**5: $(\mathfrak{C}_N) \leftarrow \text{CERTIFY}(\mathcal{P}, \bar{x}_{0:N}, \bar{u}_{0:N-1}, \Pi_{\text{cert}})$ 6: **if** CERTIFY fails **then**7: **return** FAILURE8: **end if**9: **return** $(\bar{x}_{0:N}, \bar{u}_{0:N-1}, \mathfrak{C}_N)$

4.2 Chosen instantiation of the pipeline

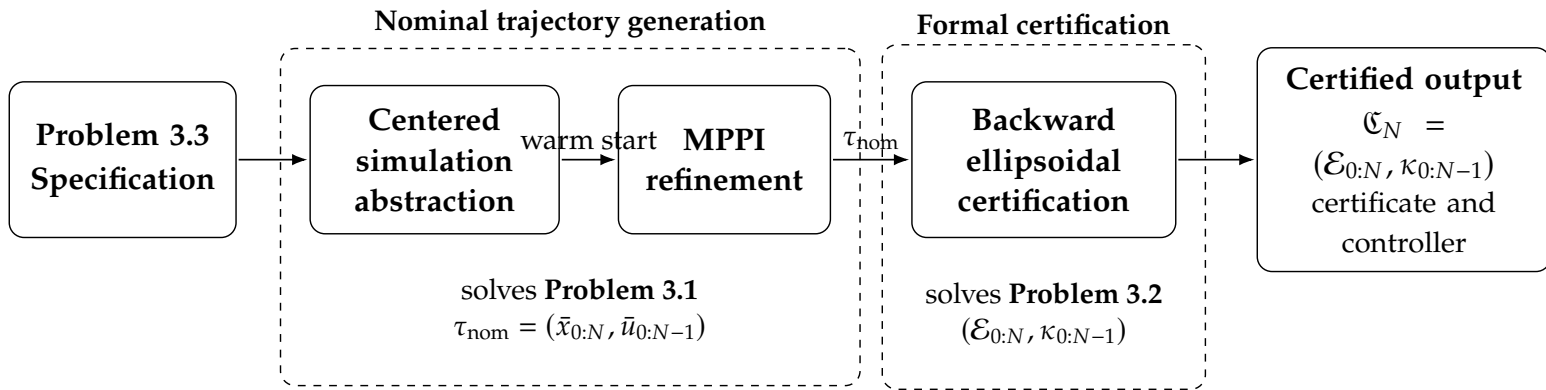
**Figure 4.1:** High-level structure of the instantiated trajectory-guided certification pipeline.

Figure 4.1 summarizes our concrete instantiation of Algorithm 1. In this thesis, the local sets are ellipsoids and the transition labels are affine feedback laws. The nominal planning problem of Problem 3.1 is handled by a two-stage procedure. The second part of the pipeline is the certification step. Given the nominal trajectory, the certifier works backward from the target set and solves a sequence of local LMI problems. When all local problems are feasible, the resulting ellipsoids and feedback laws satisfy the certified transition conditions of Definition 3.1, and therefore solve the synthesis problem formulated in Chapter 3. The following sections describe the warm-start construction, the MPPI refinement, the backward ellipsoidal certifier, and the complete algorithmic wrapper.

4.3 Nominal trajectory generation

The first module of the pipeline computes a nominal trajectory for Problem 3.1. As discussed in Section 2.4, this role could be played by many different planning methods. In the implementation studied in this thesis, we use a two-stage procedure. A coarse centered abstraction first computes a coarse symbolic path, obtained by simulating representative cell centers, which is then used to warm-start MPPI. MPPI refines this candidate at simulation level and returns a dynamically consistent nominal trajectory.

4.3.1 Centered simulation path warm start

Let $h \in \mathbb{R}_{>0}^n$ be a coarse grid size, and let Q_h be the corresponding finite set of abstract states. Each $q \in Q_h$ represents a cell $R_h^{-1}(q) \subseteq \mathcal{X}$, with center $c(q) \in \mathcal{X}$. For a finite input grid $U_h \subseteq \mathcal{U}$, we define centered transitions by simulating only from cell centers:

$$q^+ = \text{quant}_h(f(c(q), u, 0)), \quad q \in Q_h, \quad u \in U_h, \quad (4.2)$$

where $\text{quant}_h : \mathcal{X} \rightarrow Q_h$ maps a continuous state to its abstract cell.

This construction is deliberately not a conservative abstraction. It does not over-approximate all successors from $R_h^{-1}(q)$, and therefore carries no formal guarantee. Its only purpose is to cheaply compute a coarse symbolic path whose simulated cell centers avoid obstacles.

Algorithm 2 Centered abstract path warm start

Require: reach-avoid data $\mathcal{P}$, grid size h , input grid U_h

Ensure: warm start $\tau_{\text{abs}} := (\bar{x}_{0:N}^{\text{abs}}, \bar{u}_{0:N-1}^{\text{abs}})$ OR FAILURE

- 1: Build the abstract state set Q_h
- 2: Define

$$Q_I := \{q \in Q_h : R_h^{-1}(q) \cap \mathcal{X}_I \neq \emptyset\}, \quad Q_{\mathcal{T}} := \{q \in Q_h : c(q) \in \mathcal{X}_{\mathcal{T}}\},$$

$$Q_{\text{safe}} := \{q \in Q_h : c(q) \in \mathcal{X} \setminus \mathcal{X}_O\}.$$

- 3: **for** $q \in Q_h, u \in U_h$ **do**

- 4: $q^+ \leftarrow \text{quant}_h(f(c(q), u, 0))$

- 5: **if** $q^+ \in Q_{\text{safe}}$ **then**

- 6: Add the transition (q, u, q^+)

- 7: **end if**

- 8: **end for**

- 9: Search a path : $q_0 \xrightarrow{\bar{u}_0^{\text{abs}}} q_1 \xrightarrow{\bar{u}_1^{\text{abs}}} \dots \xrightarrow{\bar{u}_{N-1}^{\text{abs}}} q_N, \quad q_0 \in Q_I, \quad q_N \in Q_{\mathcal{T}}.$

- 10: **if** no such path is found **then**

- 11: **return** FAILURE

- 12: **end if**

- 13: Extract the center sequence : $\bar{x}_k^{\text{abs}} := c(q_k), \quad k = 0, \dots, N.$

- 14: **return** $\tau_{\text{abs}} := (\bar{x}_{0:N}^{\text{abs}}, \bar{u}_{0:N-1}^{\text{abs}}).$

The returned sequence τ_{abs} is a warm start, not a nominal trajectory. In general,

$$\bar{x}_{k+1}^{\text{abs}} = c(q_{k+1}) \neq f(\bar{x}_k^{\text{abs}}, \bar{u}_k^{\text{abs}}, 0), \quad (4.3)$$

because each successor is quantized back to a cell center. Thus τ_{abs} should be read as a sequence of coarse waypoints and controls.

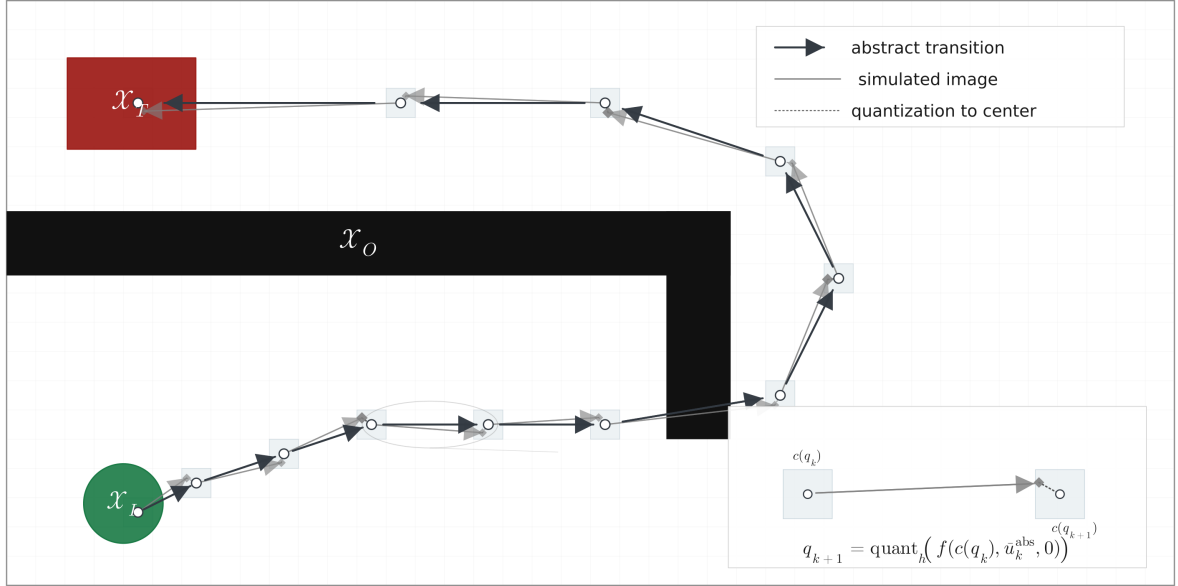


Figure 4.2: Coarse centered simulation path used to initialize MPPI. The arrows show center-to-center transitions induced by simulated successors and quantization.

The next stage uses this information to initialize MPPI. The waypoints provide a global indication of the desired route through the state space, while the control sequence gives an initial guess for the sampling distribution. MPPI then refines this coarse guide at simulation level and returns the nominal trajectory used by the certification module.

4.3.2 MPPI refinement

The second stage refines the coarse warm start τ_{abs} using Model Predictive Path Integral control (MPPI), the general update was recalled in Section 2.4; here we describe its role inside the proposed pipeline.

Starting from the control sequence extracted from the centered abstract path, MPPI performs a stochastic search in control-sequence space. At each iteration, it samples perturbations around the current nominal sequence, simulates the corresponding rollouts under the nominal dynamics $w = 0$, evaluates their costs, and updates the nominal controls by a cost-weighted average of the perturbations. In the notation of Algorithm 1, this stage returns

$$\text{Plan}(\mathcal{P}, \Pi_{\text{plan}}) = \tau_{\text{nom}} = (\bar{x}_{0:N}, \bar{u}_{0:N-1}), \quad (4.4)$$

provided that the resulting simulated trajectory satisfies the nominal reach-avoid conditions of Problem 3.1.

Algorithm 3 MPPI refinement

Require: problem data $\mathcal{P}$, warm start τ_{abs} , MPPI parameters Π_{MPPI}

Ensure: nominal trajectory τ_{nom} OR FAILURE

```

1: Initialize the nominal control sequence:  $U \leftarrow \bar{u}_{0:H-1}^{\text{abs}}$ .
2: for  $r = 1, \dots, N_{\text{it}}$  do
3:   for  $i = 1, \dots, M$  do
4:     Sample perturbations:  $\varepsilon_k^{(i)} \sim \mathcal{N}(0, \Sigma)$ ,  $k = 0, \dots, H-1$ .
5:     Simulate the rollout:  $x_0^{(i)} = x_{\text{init}}$ ,  $x_{k+1}^{(i)} = f(x_k^{(i)}, \Pi_{\mathcal{U}}(\bar{u}_k + \varepsilon_k^{(i)}), 0)$ .
6:     Compute the rollout cost  $S_i$ 
7:   end for
8:   Compute the MPPI weights:  $\omega_i = \frac{\exp(-\frac{1}{\lambda}(S_i - S_{\min}))}{\sum_{j=1}^M \exp(-\frac{1}{\lambda}(S_j - S_{\min}))}$ .
9:   for  $k = 0, \dots, H-1$  do
10:    Update the nominal control:  $\bar{u}_k \leftarrow \Pi_{\mathcal{U}}\left(\bar{u}_k + \sum_{i=1}^M \omega_i \varepsilon_k^{(i)}\right)$ .
11:   end for
12:   Simulate the updated nominal sequence from  $x_{\text{init}}$ 
13:   if the trajectory satisfies the nominal reach-avoid conditions then
14:     return  $\tau_{\text{nom}} := (\bar{x}_{0:N}, \bar{u}_{0:N-1})$ 
15:   end if
16: end for
17: return FAILURE

```

The cost used by MPPI is an implementation choice. In this thesis, it is chosen to favour trajectories that are not only nominally feasible, but also easier to certify locally. A typical rollout cost has the form $S_i = \Phi(x_H^{(i)}) + \sum_{k=0}^{H-1} \ell(x_k^{(i)}, u_k^{(i)})$, where Φ encourages reaching the target and ℓ may penalize obstacle proximity, control effort, input saturation, or rapid input variations.

MPPI is convenient in this architecture for three reasons. First, it only requires forward simulations of the dynamics and is therefore simple to apply to nonlinear benchmarks. Second, the rollout evaluations are independent and can be parallelized. Third, it naturally accepts warm starts: the centered abstract path provides an initial route through the state space, while MPPI improves this route at simulation level.

The planning configuration used in Algorithm 1 is therefore $\Pi_{\text{plan}} := (h, U_h, \Pi_{\text{MPPI}})$, where h is the coarse grid size, U_h is the representative input grid, and

$$\Pi_{\text{MPPI}} := (H, M, \Sigma, \lambda, N_{\text{it}}, \ell, \Phi).$$

Here H is the planning horizon, M the number of rollouts, Σ the sampling covariance, λ the temperature, N_{it} the number of MPPI iterations, and (ℓ, Φ) the running and terminal costs.

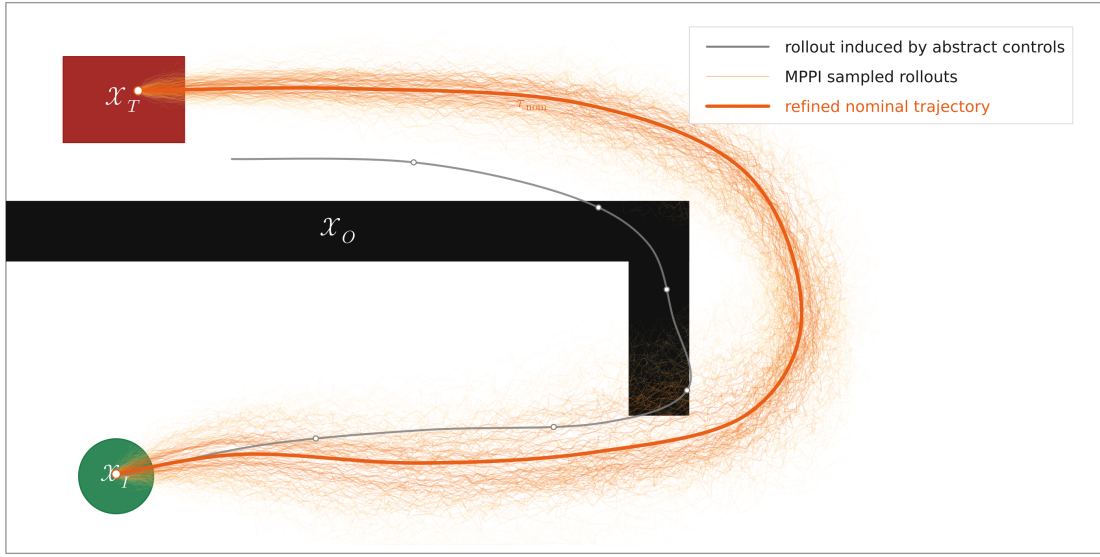


Figure 4.3: MPPI refinement of the centered simulation path. The warm start provides a coarse route and an initial control sequence; MPPI then refines it into a dynamically consistent nominal trajectory.

4.4 Backward ellipsoidal certification

The second module of the pipeline certifies the nominal trajectory returned by the planner. In the notation of Algorithm 1, it implements

$$(\mathcal{E}_{0:N}, \kappa_{0:N-1}) \leftarrow \text{Certify}(\mathcal{P}, \bar{x}_{0:N}, \bar{u}_{0:N-1}, \Pi_{\text{cert}}). \quad (4.5)$$

When successful, the output is a certified local reach-avoid chain in the sense of Definition 3.1.

The construction is performed backward. Starting from a terminal ellipsoid $\mathcal{E}_N \subseteq \mathcal{X}_T$, the certifier iteratively fixes $\mathcal{E}_{k+1}$ and searches for a predecessor ellipsoid $\mathcal{E}_k = \mathcal{E}(c_k, P_k)$, with $c_k = \bar{x}_k$, together with a local affine feedback law such that

$$\text{Reach}_f(\mathcal{E}_k, \kappa_k) \subseteq \mathcal{E}_{k+1}. \quad (4.6)$$

Thus each backward step solves a one-step local certification problem: the successor set is fixed, and the predecessor geometry and feedback gain are optimized jointly.

The semidefinite certificate used for this step is the local ellipsoidal transition certificate introduced in Section 2.5 and detailed in Appendix A. It is based on a local affine model of the nonlinear dynamics and on a certified bound for the nonlinear remainder. In practice, for each transition we choose a local certification box

$$\mathcal{B}_k = \mathcal{H}(c_k, r_k^x) \times \mathcal{H}(\bar{u}_k, r_k^u) \times \mathcal{W}, \quad (4.7)$$

compute the local linearization and remainder bounds on $\mathcal{B}_k$, and solve the corresponding SDP. Feasibility gives $P_k > 0$, K_k and ℓ_k , hence the certified transition above,

provided the consistency conditions

$$\mathcal{E}_k \subseteq \mathcal{H}(c_k, r_k^x), \quad \kappa_k(\mathcal{E}_k) \subseteq \mathcal{H}(\bar{u}_k, r_k^u), \quad \kappa_k(\mathcal{E}_k) \subseteq \mathcal{U} \quad (4.8)$$

are satisfied.

Two implementation details are useful in practice. First, the local SDP may be solved in scaled coordinates, using diagonal state and input scalings. This does not change the mathematical certificate after mapping the solution back to the original coordinates, but it can significantly improve numerical conditioning, see Appendix A.6. Second, if a computed ellipsoid intersects the obstacle set, the ellipsoid is shrunk around its center until it lies in the safe set. Equivalently, one replaces

$$\mathcal{E}(c_k, P_k) \text{ by } \mathcal{E}(c_k, \alpha^2 P_k), \quad \alpha \in (0, 1). \quad (4.9)$$

Since the source set is made smaller, the already-certified inclusion into $\mathcal{E}_{k+1}$ remains valid. If no safe shrinkage is possible, the certifier returns failure.

Algorithm 4 Backward ellipsoidal certification

Require: problem data $\mathcal{P}$, nominal trajectory $\tau_{\text{nom}} = (\bar{x}_{0:N}, \bar{u}_{0:N-1})$, certifier configuration Π_{cert}

Ensure: certified chain $\mathfrak{C}_N$ or FAILURE

- 1: Choose $P_N > 0$ such that $\mathcal{E}_N := \mathcal{E}(\bar{x}_N, P_N) \subseteq \mathcal{X}_{\mathcal{T}}$
- 2: **for** $k = N - 1, \dots, 0$ **do**
- 3: $c_k \leftarrow \bar{x}_k, u_k^{\text{ff}} \leftarrow \bar{u}_k$
- 4: Define : $\mathcal{B}_k := \mathcal{H}(c_k, r_k^x) \times \mathcal{H}(u_k^{\text{ff}}, r_k^u) \times \mathcal{W}$.
- 5: Linearize f at $(c_k, u_k^{\text{ff}}, 0)$ and compute remainder bounds on $\mathcal{B}_k$
- 6: Solve the local LMI certificate of Appendix A:

$$(P_k, K_k, \ell_k) \leftarrow \text{LocalLMI}(\mathcal{P}, \mathcal{B}_k, \mathcal{E}_{k+1}, c_k, u_k^{\text{ff}}).$$

- 7: **if** no feasible solution is found **then**
 - 8: **return** FAILURE
 - 9: **end if**
 - 10: $\mathcal{E}_k \leftarrow \mathcal{E}(c_k, P_k)$
 - 11: $\kappa_k(x) \leftarrow \ell_k + K_k(x - c_k)$
 - 12: Shrink $\mathcal{E}_k$, if needed, so that $\mathcal{E}_k \cap \mathcal{X}_O = \emptyset$
 - 13: **if** the consistency or input constraints fail **then**
 - 14: **return** FAILURE
 - 15: **end if**
 - 16: **end for**
 - 17: **return** $\mathfrak{C}_N := ((\mathcal{E}_k)_{k=0}^N, (\kappa_k)_{k=0}^{N-1})$.
-

The backward ordering is useful because each step certifies a predecessor of an already-fixed successor ellipsoid. Repeating this construction from the target to the initial set produces a chain of robust inclusions. Therefore, if Algorithm 4 returns a

chain $\mathfrak{C}_N$, Proposition 3.3 applies and the finite-horizon feedback policy satisfies the reach-avoid specification on $\mathcal{E}_0 \cap \mathcal{X}_I$.

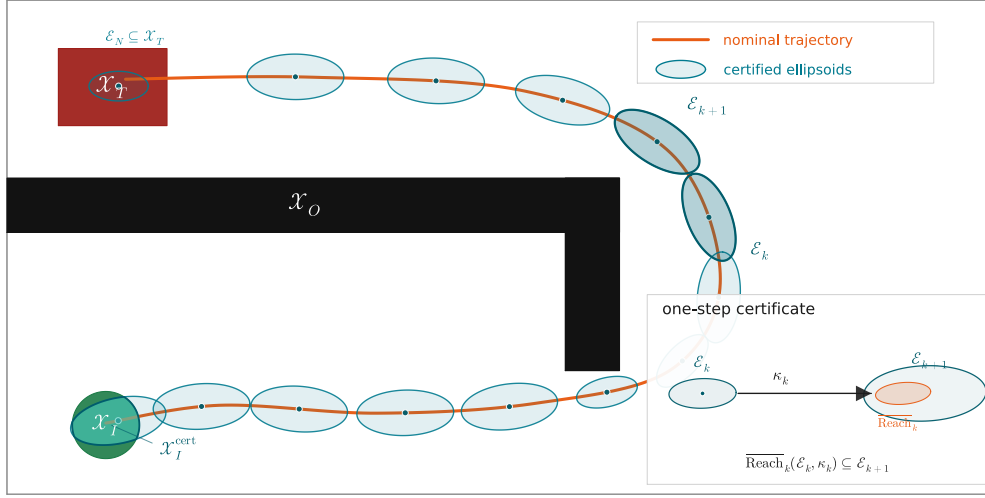


Figure 4.4: Backward construction of local ellipsoidal certificates along the nominal trajectory.

4.5 Complete algorithm

Algorithm 5 is the concrete instantiation of the generic wrapper in Algorithm 1: the generic planning routine is realized by a centered warm start followed by MPPI, and the generic certification routine is realized by backward ellipsoidal certification.

Algorithm 5 Complete trajectory-guided certification pipeline

Require: problem data $\mathcal{P}$, solver configuration $\Pi_{\text{plan}} = (\Pi_{\text{abs}}, \Pi_{\text{MPPI}})$, Π_{cert}

Ensure: $(\tau_{\text{nom}}, \mathfrak{C}_N)$ or FAILURE

- 1: $\tau_{\text{abs}} \leftarrow \text{CENTEREDWARMSTART}(\mathcal{P}, \Pi_{\text{abs}})$
 - 2: **if** $\tau_{\text{abs}} = \text{FAILURE}$ **then**
 - 3: **return** FAILURE
 - 4: **end if**
 - 5: $\tau_{\text{nom}} \leftarrow \text{MPPIREFINE}(\mathcal{P}, \tau_{\text{abs}}, \Pi_{\text{MPPI}})$
 - 6: **if** $\tau_{\text{nom}} = \text{FAILURE}$ **then**
 - 7: **return** FAILURE
 - 8: **end if**
 - 9: $\mathfrak{C}_N \leftarrow \text{BACKWARDCERTIFY}(\mathcal{P}, \tau_{\text{nom}}, \Pi_{\text{cert}})$
 - 10: **if** $\mathfrak{C}_N = \text{FAILURE}$ **then**
 - 11: **return** FAILURE
 - 12: **end if**
 - 13: **return** $(\tau_{\text{nom}}, \mathfrak{C}_N)$
-

4.6 Soundness, scope, and limitations

We finally summarize the guarantee provided by the complete pipeline.

Proposition 4.1 (Soundness of the complete pipeline). *Assume that Algorithm 5 returns*

$$(\tau_{\text{nom}}, \mathfrak{C}_N), \quad \mathfrak{C}_N = ((\mathcal{E}_k)_{k=0}^N, (\kappa_k)_{k=0}^{N-1}).$$

Assume moreover that each call to the local LMI certifier is sound, in the sense of Appendix A. Then $\mathfrak{C}_N$ is a certified local reach-avoid chain. Consequently, the finite-horizon feedback policy

$$\pi = (\kappa_0, \dots, \kappa_{N-1})$$

solves the robust reach-avoid specification on $\mathcal{X}_I^{\text{cert}}(\mathfrak{C}_N) = \mathcal{E}_0 \cap \mathcal{X}_I$.

Proof. If Algorithm 5 returns successfully, then the MPPI stage has produced a nominal trajectory $\tau_{\text{nom}} = (\bar{x}_{0:N}, \bar{u}_{0:N-1})$ satisfying the nominal reach-avoid conditions. The backward certifier then returns ellipsoids $(\mathcal{E}_k)_{k=0}^N$ and feedback laws $(\kappa_k)_{k=0}^{N-1}$.

By construction, the returned objects satisfy the conditions of Definition 3.1, with the ellipsoidal structure of Definition 3.2:

- since $\bar{x}_0 \in \mathcal{X}_I$, $c_0 = \bar{x}_0$, and $c_0 \in \mathcal{E}_0$, we have $\mathcal{E}_0 \cap \mathcal{X}_I \neq \emptyset$;
- $\mathcal{E}_k \subseteq \mathcal{X}$ and $\mathcal{E}_k \cap \mathcal{X}_O = \emptyset$, for $k = 0, \dots, N-1$;
- $\mathcal{E}_N \subseteq \mathcal{X}_{\mathcal{T}}$;
- $\kappa_k(\mathcal{E}_k) \subseteq \mathcal{U}$, for $k = 0, \dots, N-1$;
- by soundness of the local LMI certificate,

$$\text{Reach}_f(\mathcal{E}_k, \kappa_k) \subseteq \mathcal{E}_{k+1}, \quad k = 0, \dots, N-1.$$

Thus $\mathfrak{C}_N$ satisfies the definition of a certified local reach-avoid chain. The robust reach-avoid guarantee then follows directly from Proposition 3.3. $\square$

Remark 4.2 (Meaning of failure). A failure of Algorithm 5 should not be interpreted as a proof that the reach-avoid specification is unrealizable. It only means that the chosen planner, nominal trajectory, local model, certification boxes, and SDP parameters did not produce a certified chain. Different warm starts, horizons, costs, local boxes, or numerical scalings may lead to a successful certificate.

Benchmarks and Experimental Evaluation

This chapter evaluates the trajectory-guided certification pipeline introduced in Chapter 4 on nonlinear reach-avoid problems. The objective is to test whether the pipeline can solve the synthesis problem formulated in Chapter 3.

When a chain is returned, the formal guarantee follows from the certified one-step inclusions of Definition 3.1 and the soundness result of Proposition 3.3. The sampled simulations reported below are therefore only diagnostic. In each successful benchmark, we sample 12000 initial states in $\mathcal{X}_I^{\text{cert}}(\mathcal{C}_N) = \mathcal{E}_0 \cap \mathcal{X}_I$, apply the feedback sequence $(\kappa_0, \dots, \kappa_{N-1})$, and record whether the resulting closed-loop trajectories satisfy the reach-avoid constraints. All continuous-time models are evaluated through their RK4 discretization. In the experiments, we set $\mathcal{W} = \{0\}$. The chapter ends with supplementary animations and a discussion of the main limitations, the role of the abstract warm start for MPPI, the comparison with uniform-grid abstractions, and the possible composition of certified chains.

5.1 Benchmark design and evaluation protocol

The benchmarks are selected to stress complementary certification mechanisms:

Table 5.1: Role of each benchmark in the experimental campaign.

Benchmark	Role in the evaluation
Simple pendulum	Low-dimensional sanity check; direct visualization of the certified sets.
Double pendulum	More nonlinear four-dimensional dynamics; tests the effect of dimension and coupling on local certification.
Forward articulated vehicle	Nonholonomic reach-avoid problem with obstacles, steering limits, and articulation constraints.
Reverse manoeuvre	Tests certification near direction changes and trailer realignment phases, where local nonlinear effects become more conservative.
Parallel parking	Tests the role of the nominal planner in finding the topology of a constrained parking motion with limited clearance.

The discussion then focuses on what makes a nominal trajectory certifiable: local model accuracy, feedback authority, input saturation, numerical conditioning, and obstacle clearance.

5.1.1 Evaluation metrics

Table 5.2: Metrics used to evaluate the experimental results.

Category	Metric	Interpretation
Nominal planning	Planning success	Whether a nominal trajectory satisfying the reach-avoid task is found.
Nominal planning	Minimum obstacle clearance	Indicates how much geometric margin is left for robust certification.
Nominal planning	Input saturation	Measures whether the nominal trajectory leaves room for local feedback corrections.
Certification	Certification success	Whether all local LMI problems are feasible and return a complete chain.
Certification	Minimum ellipsoid size along the chain	Identifies bottleneck transitions where certification becomes tight.
Computation	Planning and certification runtime	Measures the practical cost of the pipeline.
Stress test	Sampled transition success rate	Empirical check that sampled closed-loop successors remain inside the certified chain.

5.2 Main certification results

5.2.1 Simple pendulum

Simple pendulum benchmark

Purpose. The simple pendulum swing-up is used as a low-dimensional diagnostic benchmark. The dynamics are nonlinear, but the two-dimensional state space makes the nominal trajectory, the certified ellipsoids, and the input profile directly interpretable. The purpose of this example is therefore twofold: to validate the pipeline on a nonlinear swing-up task, and to isolate the mechanisms that limit the size of the certified sets.

State and input. The state is $x = (\theta, \dot{\theta})$, where θ is the pendulum angle and $\dot{\theta}$ its angular velocity. The control input $u \in \mathbb{R}$ acts as an external torque.

Continuous-time dynamics.

$$\dot{x} = f_c(x, u) = \begin{bmatrix} \dot{\theta} \\ -\frac{g}{\ell} \sin(\theta) + u \end{bmatrix}, \quad \ell = 1, \quad g = 9.81. \quad (5.1)$$

Simple pendulum: problem specification

The state and input constraints are

$$\mathcal{X} = [-\pi, \pi] \times [-7, 7], \quad \mathcal{U} = [-4.5, 4.5]. \quad (5.2)$$

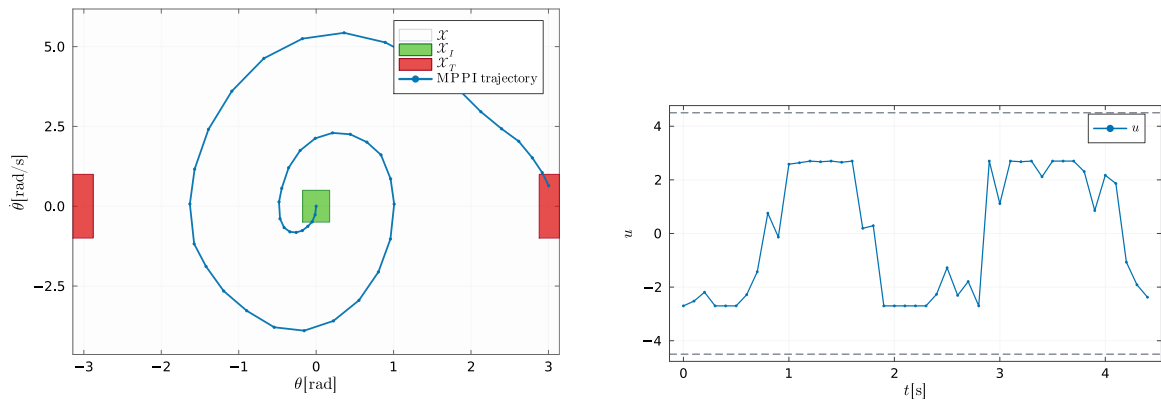
The initial and target sets are

$$\mathcal{X}_I = \left[-\frac{10\pi}{180}, \frac{10\pi}{180} \right] \times [-0.5, 0.5], \quad (5.3)$$

$$\mathcal{X}_T = \left[\pi - \frac{15\pi}{180}, \pi + \frac{15\pi}{180} \right] \times [-1, 1]. \quad (5.4)$$

No obstacle constraint is imposed, i.e., $\mathcal{X}_O = \emptyset$.

Angles are interpreted modulo 2π , so that the target set represents a neighbourhood of the upright equilibrium.



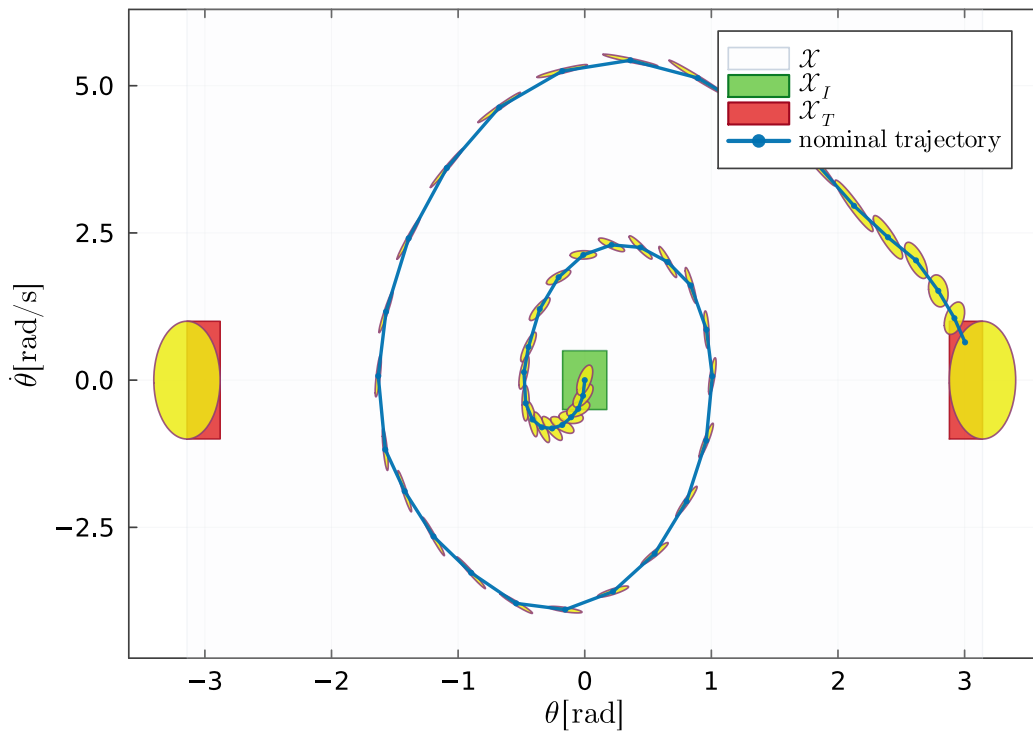
(a) Refined MPPI trajectory in the phase space.

(b) Nominal input sequence.

Figure 5.1: Nominal swing-up trajectory produced by the planning stage. The left panel shows the refined state trajectory, while the right panel shows the corresponding nominal control sequence $\bar{u}_k$.

Table 5.3: Summary for the simple pendulum benchmark.

Quantity	Value
Time step	$\Delta t = 0.1$
Abstract trajectory horizon	$N_{\text{abs}} = 42$
MPPI horizon	$H_{\text{MPPI}} = 45$
MPPI iterations	20
MPPI samples per iteration	2000
Coarse grid resolution	$h = (5^\circ, 0.25)$
Certified chain	yes
Sampled initial states	12000
Reached target	12000/12000
State constraint violations	0
Input constraint violations	0
Abstract warm-start runtime	0.74 s
MPPI refinement runtime	2.05 s
LMI certification runtime	8.41 s

**Figure 5.2:** Certified swing-up chain for the simple pendulum. The refined MPPI trajectory is surrounded by the ellipsoidal certificates produced by the backward LMI step.

Discussion. This benchmark highlights a key distinction between nominal feasibility and certifiability. The LMI certificate must enforce both the transition inclusion and the admissibility of the whole feedback image. For an ellipsoid written with the shape-matrix convention $\mathcal{E}_k = \{x : (x - c_k)^\top P_k^{-1}(x - c_k) \leq 1\}$, the certified affine controller has the form $\kappa_k(x) = K_k(x - c_k) + \ell_k$. Hence, for the scalar input of the simple pendulum, the feedback image is an interval centered at ℓ_k :

$$\kappa_k(\mathcal{E}_k) = [\ell_k - r_k^u, \ell_k + r_k^u], \quad r_k^u = \sqrt{K_k P_k K_k^\top} \cdot 1. \quad (5.5)$$

Thus, if $\mathcal{U} = [-u_{\max}, u_{\max}]$, input admissibility is equivalent to

$$|\ell_k| + r_k^u \leq u_{\max}. \quad (5.6)$$

Nominal input saturation therefore does not, by itself, force the feedback radius r_k^u to vanish. However, since the successor ellipsoid is centered around the nominal successor state, and since the local linearization is built around the nominal input $\bar{u}_k$, a transition generated by a nearly saturated nominal input often requires the certified offset ℓ_k and the whole feedback image to remain close to the boundary of $\mathcal{U}$. In that case, only a small admissible margin remains for feedback variation. The certifier must then reduce the feedback gain, shrink the state ellipsoid, or declare the local transition infeasible.

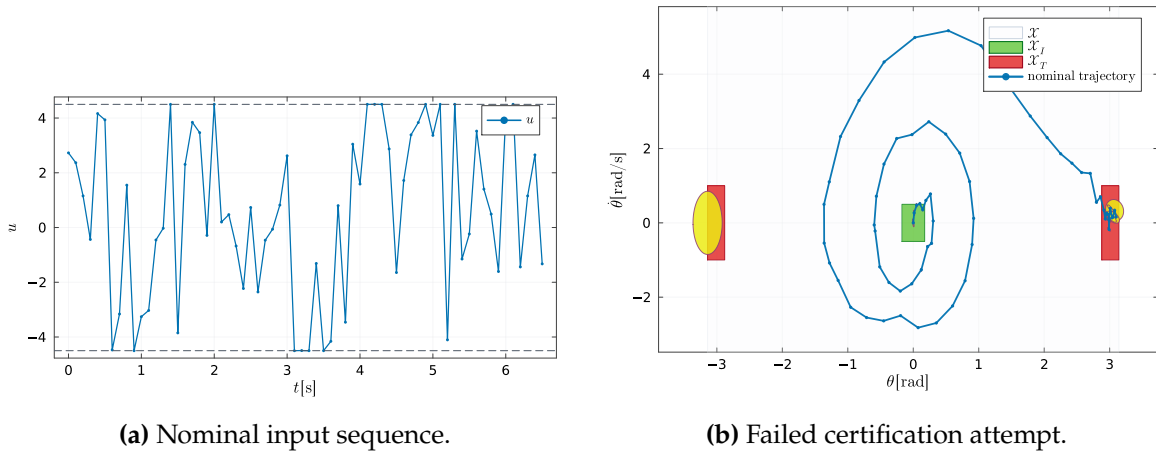


Figure 5.3: Input-saturation diagnostic for the simple pendulum. Left: nominal input sequence. Right: failed certification attempt along the corresponding nominal trajectory.

In Figure 5.3, the nominal trajectory reaches the target but uses nearly saturated inputs. The resulting lack of input margin forces the certified ellipsoids to become very small, and the backward certification eventually becomes infeasible.

¹Indeed, with $y = x - c_k$, one has $r_k^u = \max_{y^\top P_k^{-1} y \leq 1} |K_k y|$. By symmetry it is enough to maximize $K_k y$, and since the objective is linear the optimum is attained on the boundary $y^\top P_k^{-1} y = 1$. The Lagrangian $\mathcal{L}(y, \lambda) = K_k y - \lambda(y^\top P_k^{-1} y - 1)$ gives $K_k^\top - 2\lambda P_k^{-1} y = 0$, hence $y^\star = (2\lambda)^{-1} P_k K_k^\top$. Imposing $(y^\star)^\top P_k^{-1} y^\star = 1$ yields $4\lambda^2 = K_k P_k K_k^\top$, and therefore $K_k y^\star = \sqrt{K_k P_k K_k^\top}$. The opposite point gives the minimum.

5.2.2 Double pendulum

Double pendulum benchmark

Purpose. The double pendulum swing-up is a more challenging benchmark than the simple pendulum. The state dimension increases to four and the dynamics contain nonlinear coupling terms between the two links. This makes the benchmark representative of situations where uniform grid abstractions become rapidly conservative or computationally expensive.

State and input. The state is $x = (\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2)$, where θ_1, θ_2 are the angular positions of the two links and $\dot{\theta}_1, \dot{\theta}_2$ are their angular velocities. The scalar input $u \in \mathbb{R}$ acts on the first joint.

Continuous-time dynamics.

$$\dot{x} = f_c(x, u) = \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ f_3(x, u) \\ f_4(x) \end{bmatrix}. \quad (5.7)$$

With $\Delta\theta = \theta_1 - \theta_2$, $M = m_1 + m_2$, $\alpha = m_1 + m_2 \sin^2(\Delta\theta)$, the acceleration terms are

$$f_3(x, u) = -\sin(\Delta\theta) (m_2 \ell_1 \dot{\theta}_1^2 \cos(\Delta\theta) + m_2 \ell_2 \dot{\theta}_2^2) - \frac{g (M \sin(\theta_1) - m_2 \sin(\theta_2) \cos(\Delta\theta))}{\ell_1 \alpha} + u, \quad (5.8)$$

$$f_4(x) = \sin(\Delta\theta) (M \ell_1 \dot{\theta}_1^2 + m_2 \ell_2 \dot{\theta}_2^2 \cos(\Delta\theta)) + \frac{g (M \sin(\theta_1) \cos(\Delta\theta) - M \sin(\theta_2))}{\ell_2 \alpha}. \quad (5.9)$$

The parameters are fixed to $\ell_1 = \ell_2 = 1$, $m_1 = m_2 = 1$, $g = 9.81$.

Double pendulum: problem specification

The state and input constraints are

$$\mathcal{X} = [-\pi, \pi]^2 \times [-6, 6]^2, \quad \mathcal{U} = [-8.5, 8.5]. \quad (5.10)$$

The initial set is a small neighbourhood of the downward configuration with small angular velocities,

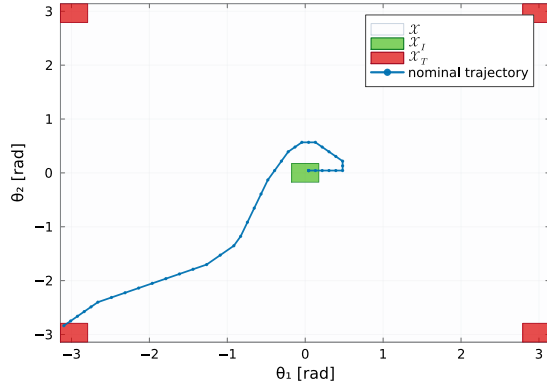
$$\mathcal{X}_I = \left[-\frac{10\pi}{180}, \frac{10\pi}{180} \right]^2 \times [-0.5, 0.5]^2. \quad (5.11)$$

The target set is a neighbourhood of the upright configuration for both links,

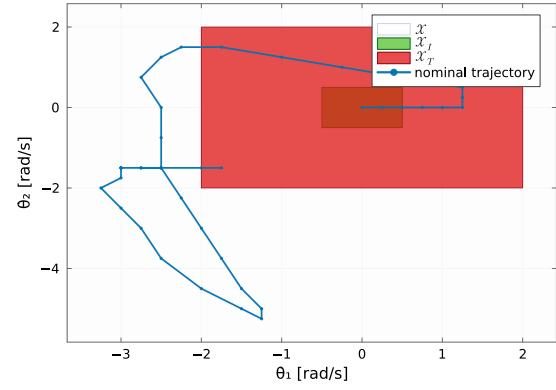
$$\mathcal{X}_T = \left[\pi - \frac{25\pi}{180}, \pi + \frac{25\pi}{180} \right]^2 \times [-5, 5]^2. \quad (5.12)$$

No obstacle constraint is imposed in this benchmark: $\mathcal{X}_O = \emptyset$.

Angles are interpreted modulo 2π , so that the target set represents a neighbourhood of the upright equilibrium for both links.



(a) (θ_1, θ_2) -projection.



(b) $(\dot{\theta}_1, \dot{\theta}_2)$ -projection.

Figure 5.4: Abstract trajectory used as warm start for the double pendulum benchmark.

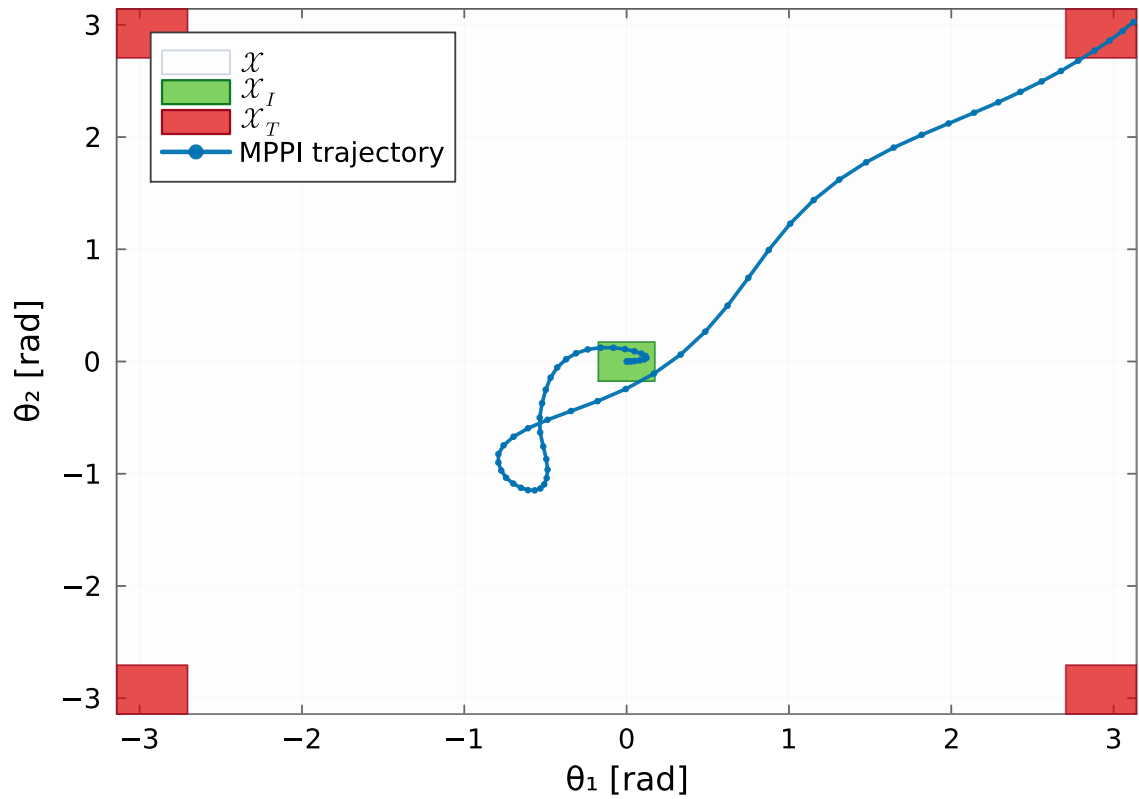


Figure 5.5: Refined MPPI trajectory, (θ_1, θ_2) -projection, for the double pendulum benchmark.

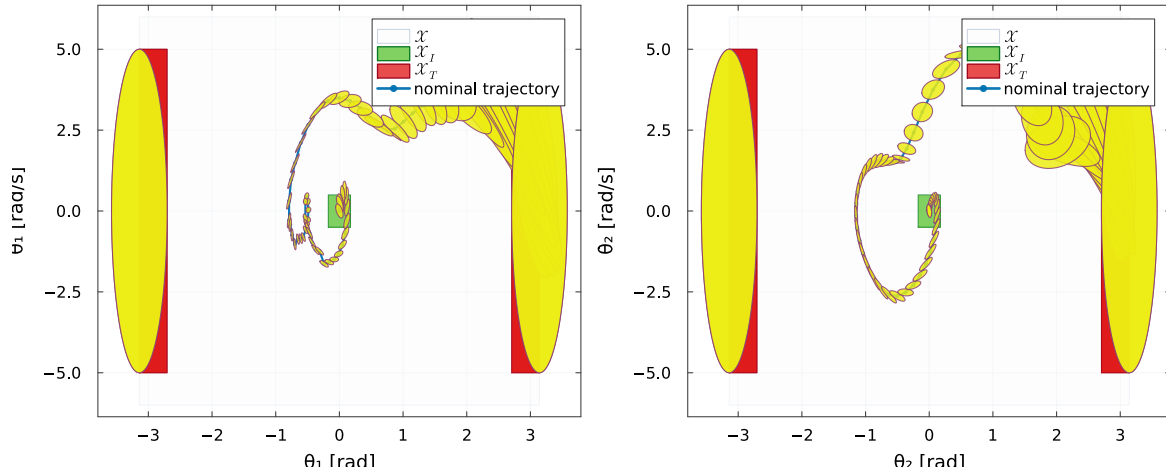


Figure 5.6: Certified ellipsoidal chain for the double pendulum benchmark.

Table 5.4: Summary for the double pendulum benchmark.

Quantity	Value
Time step	$\Delta t = 0.05$
Abstract trajectory horizon	$N_{\text{abs}} = 41$
MPPI horizon	$H_{\text{MPPI}} = 66$
MPPI iterations	20
MPPI samples per iteration	1800
Coarse grid resolution	$h = (5^\circ, 5^\circ, 0.25, 0.25)$
Certified chain	yes
Sampled initial states	12000
Reached target	12000/12000
State constraint violations	0
Input constraint violations	0
Abstract warm-start runtime	174.2 s
MPPI refinement runtime	41.9 s
LMI certification runtime	221.3 s

Discussion. This benchmark is challenging because the double pendulum is underactuated and strongly nonlinear. The coupling between the two links can make local linearization errors grow quickly along the swing-up trajectory, which increases the conservatism of the LMI certificate. We therefore use a small sampling time, so that consecutive nominal states remain close and the local models vary less from one step to the next. The SDP is also solved in scaled coordinates to improve numerical conditioning.

5.2.3 Articulated vehicle benchmarks

Articulated vehicle benchmark

Purpose. This benchmark is based on the kinematic articulated-vehicle model studied by Bolzern and Locatelli [40], which provides the continuous-time dynamics used below. We use this model to build a parking benchmark inspired by trailer-driving manoeuvres associated with Belgian category B/BE driving licences, such as forward parking, reverse parking, and parallel parking with a car-like tractor connected to a trailer.

This benchmark is interesting because it is both simple enough to be visualized and close to a practical control problem. The state dimension is four, the dynamics are nonlinear and nonholonomic, and the vehicle must satisfy steering, articulation, and obstacle-avoidance constraints.

State and input. The state is $x = (p_x, p_y, \theta, \varphi)$, where (p_x, p_y) is the planar position of the tractor, θ is its heading angle, and φ is the articulation angle between the tractor and the trailer. The input is $u = (v, \tilde{\sigma})$, where v is the longitudinal velocity and $\tilde{\sigma} = \tan(\delta)$ is the steering variable associated with the steering angle δ .

Continuous-time dynamics.

$$\dot{x} = f_c(x, u) = \begin{bmatrix} v \cos(\theta) \\ v \sin(\theta) \\ \frac{v}{L_1} \tilde{\sigma} \\ -\frac{v}{L_1 L_2} (L_1 \sin(\varphi) + (L_c \cos(\varphi) + L_2) \tilde{\sigma}) \end{bmatrix}. \quad (5.13)$$

The first two equations describe the planar motion of the tractor. The third equation gives the evolution of the heading angle, while the last equation describes the evolution of the articulation angle between the tractor and the trailer.

Geometric parameters. The parameters are fixed to

$$L_1 = 2.2, \quad L_2 = 2.8, \quad L_c = 0.9.$$

Here, L_1 denotes the tractor wheelbase, L_2 the trailer length, and L_c the hitch offset parameter. As for the other benchmarks, the continuous-time dynamics are discretized with a fourth-order Runge–Kutta scheme before applying the certification pipeline.

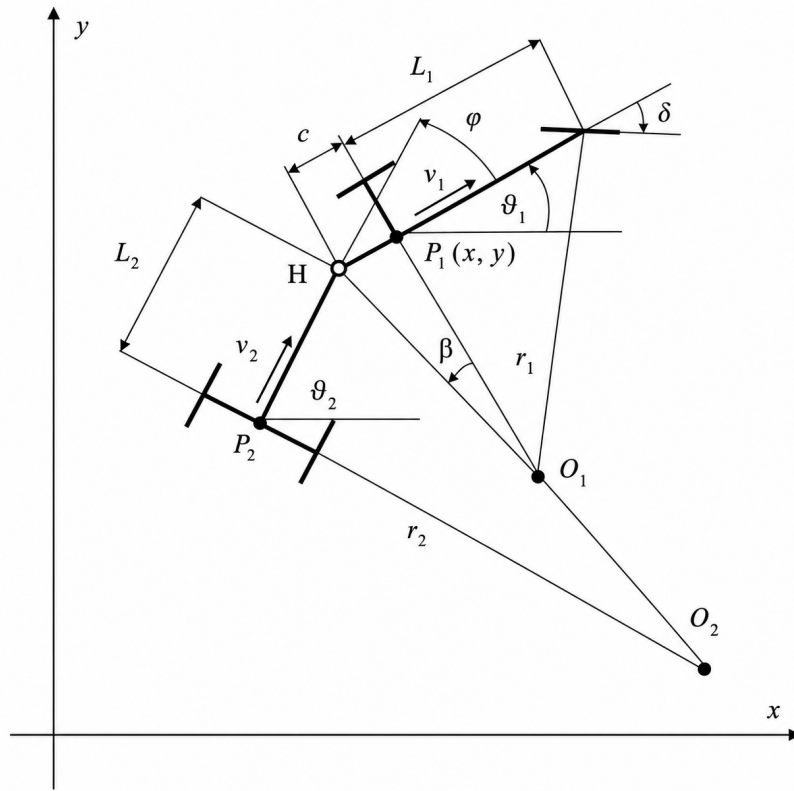


Figure 5.7: Kinematic geometry of the articulated vehicle, adapted from Bolzern and Locatelli [40]. The original notation c corresponds here to the hitch offset L_c .

Forward manoeuvre

Forward manoeuvre: problem specification

State and input constraints. The state is $x = (p_x, p_y, \theta, \varphi)$ and the input is $u = (v, \tilde{\sigma})$, with $\tilde{\sigma} = \tan(\delta)$. The admissible state domain is

$$\mathcal{X} = [-1, 10] \times [-1, 10] \times [-\pi, \pi] \times \left[-\frac{75\pi}{180}, \frac{75\pi}{180}\right].$$

The input constraints are

$$\mathcal{U} = [-5, 5] \times [-\sigma_{\max}, \sigma_{\max}], \quad \sigma_{\max} = \tan\left(\frac{55\pi}{180}\right) \simeq 1.428.$$

Obstacles. The obstacle set is defined by two planar rectangles, lifted to the full state space:

$$\mathcal{O}_{xy}^{(1)} = [4, 10] \times [-1, 4.7], \quad \mathcal{O}_{xy}^{(2)} = [4, 10] \times [7, 10],$$

and

$$\mathcal{X}_O = \left\{ x \in \mathcal{X} : (p_x, p_y) \in \mathcal{O}_{xy}^{(1)} \cup \mathcal{O}_{xy}^{(2)} \right\}.$$

Initial and target sets. The nominal initial state is

$$x_0 = (0, 0, 0, 0).$$

The initial set is a neighbourhood of the straight configuration at the origin:

$$\mathcal{X}_I = [-0.4, 0.4]^2 \times \left[-\frac{20\pi}{180}, \frac{20\pi}{180} \right]^2.$$

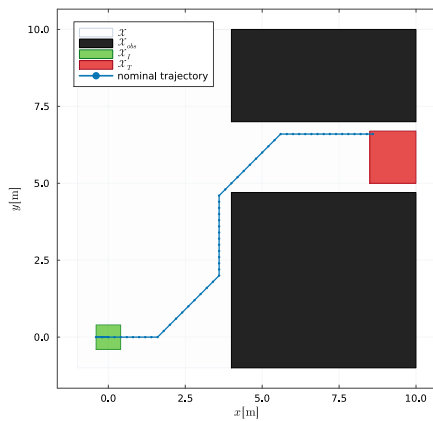
The target set is the upper parking region:

$$\mathcal{X}_T = [8.5, 10] \times [5.0, 6.7] \times \left[-\frac{35\pi}{180}, \frac{35\pi}{180} \right]^2.$$

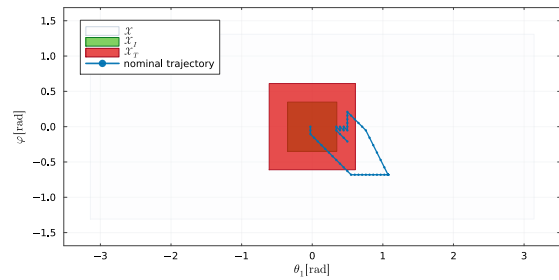
The objective is to drive the vehicle from $\mathcal{X}_I$ to $\mathcal{X}_T$, while satisfying the steering and articulation constraints and avoiding $\mathcal{X}_O$.

The vehicle starts from the lower-left initial region and must reach the upper parking region by passing between the two obstacles. This benchmark tests whether the proposed pipeline can lift a nominal parking manoeuvre into a certified feedback-controlled motion while respecting the steering, articulation, state, and obstacle constraints.

We report the experiment in four parts: the coarse abstract trajectory used as warm start, the refined MPPI trajectory together with the certified ellipsoids, a compact summary table of the main numerical results, and a statistical closed-loop validation obtained by sampling initial states in the certified initial ellipsoid.

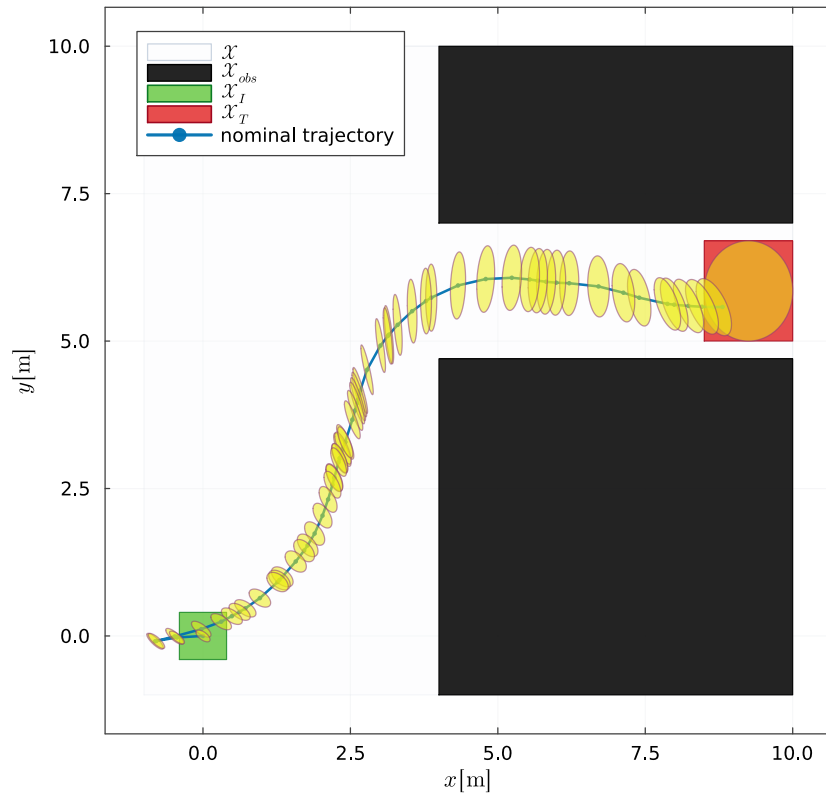
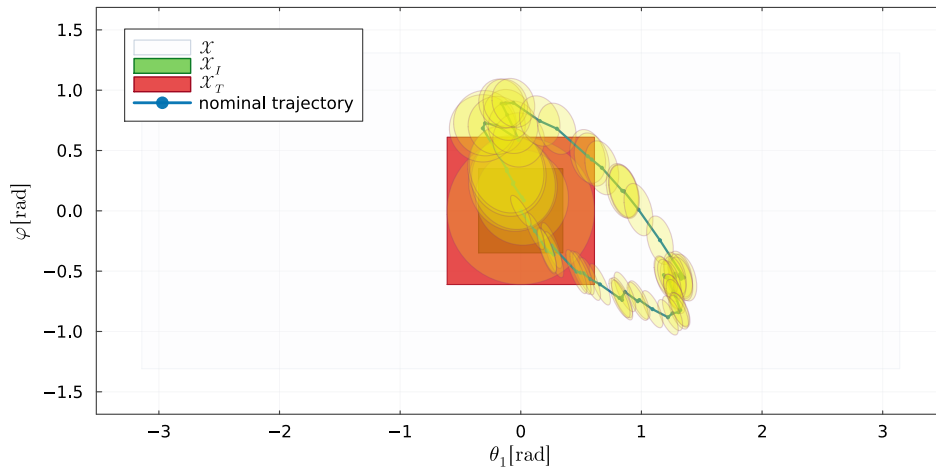


(a) (p_x, p_y) -projection.



(b) (θ, φ) -projection.

Figure 5.8: Coarse abstract trajectory used as warm start for the forward articulated-vehicle manoeuvre. The left projection shows the geometric route in the plane, while the right projection shows the corresponding heading and articulation angles.

(a) (p_x, p_y) -projection.(b) (θ, φ) -projection.**Figure 5.9:** Refined MPPI trajectory and certified ellipsoidal chain for the forward manoeuvre.

The left plot shows the geometric motion in the plane, while the right plot shows the evolution of the heading and articulation angles. The yellow ellipsoids are the certified local sets produced by the backward certification step.

Table 5.5: Summary for the forward articulated-vehicle benchmark.

Quantity	Value
Time step	$\Delta t = 0.1$
Abstract trajectory horizon	$N_{\text{abs}} = 51$
MPPI horizon	$H_{\text{MPPI}} = 60$
Coarse grid resolution	$h = (0.4, 0.4, 5^\circ, 3^\circ)$
MPPI samples per iteration	1800
MPPI iterations	15
MPPI temperature	$\lambda_{\text{MPPI}} = 0.3$
MPPI noise scale	$(\sigma_v, \sigma_{\bar{\sigma}}) = (0.45, 0.15)$
Certified chain	yes
Sampled initial states	12000
Reached target	12000/12000
Obstacle violations	0
State constraint violations	0
Input constraint violations	0
Abstract warm-start runtime	110.74 s
MPPI refinement runtime	114.35 s
LMI certification runtime	139.4 s

We then sample 12000 concrete initial states in the certified initial ellipsoid (for this benchmark : $\mathcal{X}_I^{\text{cert}}(\mathfrak{C}_N) = \mathcal{E}_0 \subseteq \mathcal{X}_I$) and replay the RK4 dynamics under the certified feedback sequence. All sampled trajectories reach the target, with no obstacle, state, or input constraint violation.

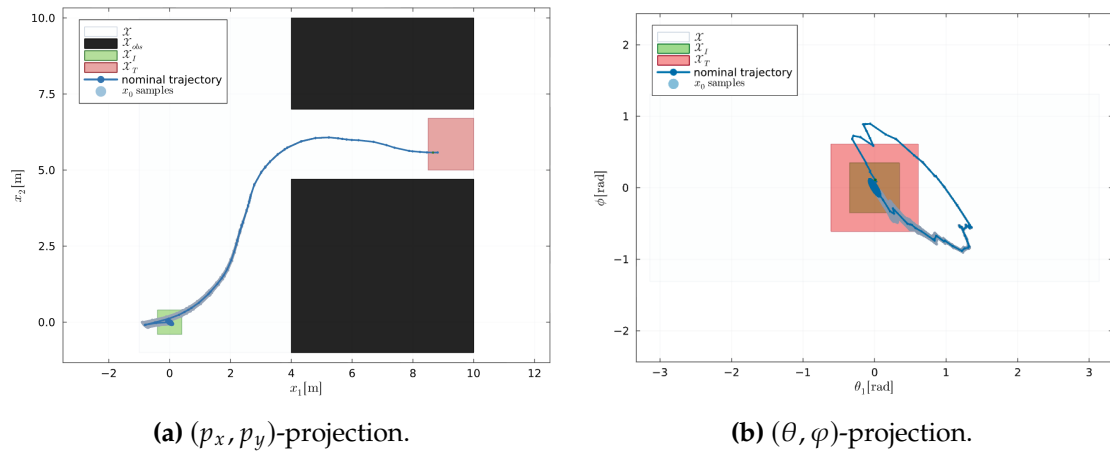


Figure 5.10: Closed-loop rollouts from 12000 initial states sampled in the certified initial set. All sampled trajectories reach the target while satisfying the obstacle, state, and input constraints.

Discussion. The forward manoeuvre is successfully certified. Beyond this success result, the rollouts highlight two qualitative features.

- **The closed-loop trajectories concentrate along the chain.** Samples are initially spread inside the certified initial state, but progressively merge toward the nominal motion as the vehicle approaches the target.
- **The certified ellipsoids shrink along the trajectory.** This reflects the conservatism of the certificate: each local set must satisfy the transition inclusion, input constraints, obstacle avoidance, and local-model consistency checks. The certified set can therefore be smaller than the empirical region where the feedback controller succeeds.

We run an empirical inflation stress test around the certified initial ellipsoid $\mathcal{E}_0$. Inflated ellipsoids $\mathcal{E}_0(\alpha)$ are obtained by scaling the radii of $\mathcal{E}_0$ by a factor $\alpha \geq 1$. For $\alpha = 1$, 1000 initial states are sampled on the certified boundary $\partial\mathcal{E}_0$, while for $\alpha > 1$ they are sampled in the shell between $\partial\mathcal{E}_0$ and the inflated boundary $\partial\mathcal{E}_0(\alpha)$. From each sampled state, we replay the same certified feedback sequence $(\kappa_0, \dots, \kappa_{N-1})$ on the concrete RK4 dynamics.

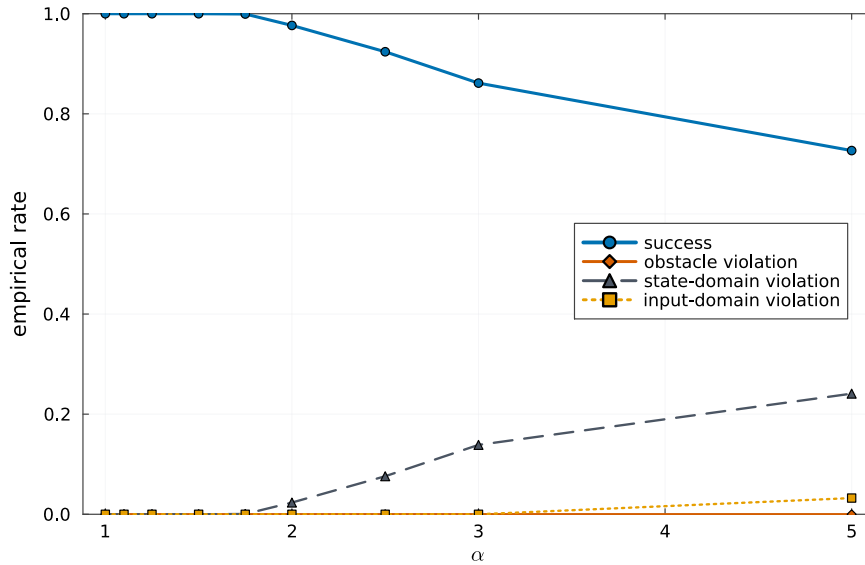


Figure 5.11: Empirical inflation stress test around $\mathcal{E}_0$. The plot reports empirical success and violation rates; only $\alpha = 1$ is certified.

Figure 5.11 supports the interpretation suggested by the rollouts. The feedback sequence remains effective for a region larger than the certified initial ellipsoid. This shows that the certified ellipsoids are conservative approximations of the region where the controller works empirically. In particular, the local certificate must absorb the linearization error and prove a robust one-step inclusion. This tends to force the closed-loop image of each ellipsoid under κ_k to lie well inside the next ellipsoid, rather than merely reaching its boundary. This compression explains both observations above: the certified sets shrink along the chain, while the simulated closed-loop trajectories progressively concentrate around the nominal motion.

*Reverse manoeuvre***Reverse manoeuvre: problem specification**

State and input constraints. The state is $x = (p_x, p_y, \theta, \varphi)$ and the input is $u = (v, \tilde{\sigma})$, with $\tilde{\sigma} = \tan(\delta)$. The admissible state domain is

$$\mathcal{X} = [-1, 16] \times [-1, 10] \times [-\pi, \pi] \times \left[-\frac{75\pi}{180}, \frac{75\pi}{180}\right],$$

and

$$\mathcal{U} = [-5, 5] \times [-\sigma_{\max}, \sigma_{\max}], \quad \sigma_{\max} = \tan\left(\frac{55\pi}{180}\right) \simeq 1.428.$$

Obstacles. The obstacle set is defined by two planar rectangles:

$$\mathcal{O}_{xy}^{(1)} = [9, 16] \times [-1, 4.0], \quad \mathcal{O}_{xy}^{(2)} = [9, 16] \times [7.4, 10],$$

lifted to the full state space as

$$\mathcal{X}_O = \left\{x \in \mathcal{X} : (p_x, p_y) \in \mathcal{O}_{xy}^{(1)} \cup \mathcal{O}_{xy}^{(2)}\right\}.$$

Initial and target sets. The nominal initial state is

$$x_0 = (0, 0, 0, 0).$$

The initial and target sets are

$$\mathcal{X}_I = [-0.4, 0.4]^2 \times \left[-\frac{20\pi}{180}, \frac{20\pi}{180}\right]^2,$$

and

$$\mathcal{X}_T = [14, 16] \times [4.5, 6.9] \times \left[\pi - \frac{20\pi}{180}, \pi + \frac{20\pi}{180}\right] \times \left[-\frac{20\pi}{180}, \frac{20\pi}{180}\right].$$

The objective is to drive the vehicle in reverse from $\mathcal{X}_I$ to $\mathcal{X}_T$, while satisfying the steering and articulation constraints and avoiding $\mathcal{X}_O$.

The reverse manoeuvre considers a more delicate parking task. The vehicle starts from a forward-facing configuration outside the alley and must enter the parking region in reverse. In practice, the motion first creates a curved alignment, close to a half-turn in the plane, and then backs the trailer into the target region. This benchmark is therefore more demanding than the forward case: the controller must stabilize the articulation angle while driving backward and while respecting the same steering, articulation, state, and obstacle constraints.

We report the experiment in five parts: the coarse abstract trajectory used as warm start, the refined MPPI trajectory, the certified ellipsoidal chain, a compact summary table of the main numerical results, and a short discussion of the obtained certificate.

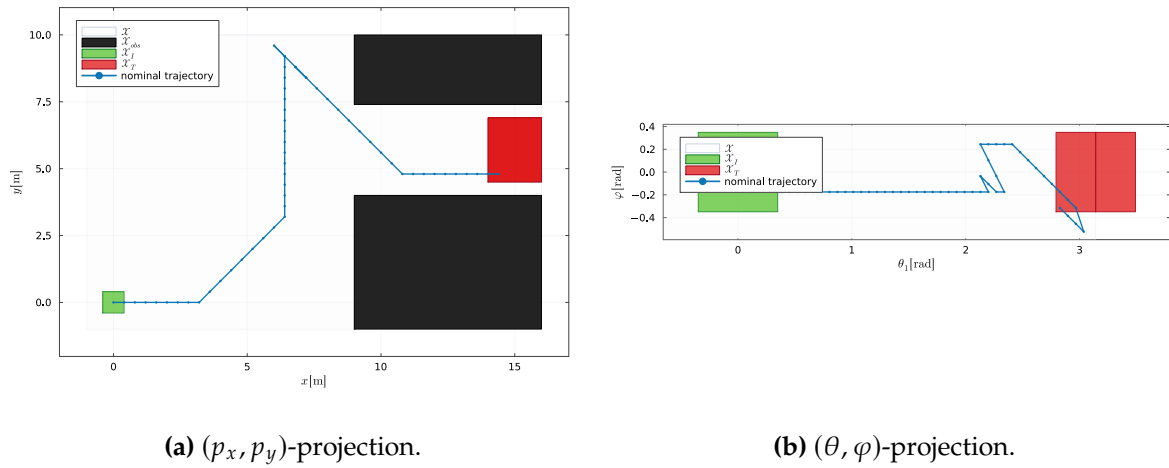


Figure 5.12: Coarse abstract trajectory used as warm start for the reverse articulated-vehicle manoeuvre. The left projection shows the geometric route used to align the vehicle with the alley, while the right projection shows the corresponding heading and articulation angles.

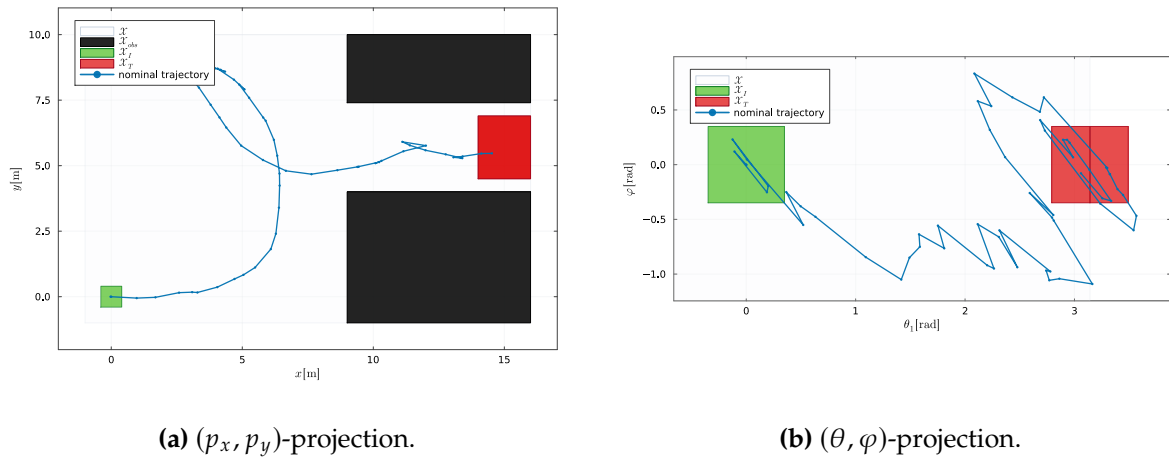


Figure 5.13: Refined MPPI trajectory for the reverse manoeuvre. The trajectory smooths the coarse warm start and produces a dynamically consistent nominal motion satisfying the geometric structure of the parking task.

Remark 5.1 (Angular plots). Since angular variables are interpreted modulo 2π , the angular projections are displayed using lifted angles to avoid artificial jumps at the $\pi/-\pi$ cut. This is only a plotting convention: the dynamics, constraints, and certification remain unchanged.

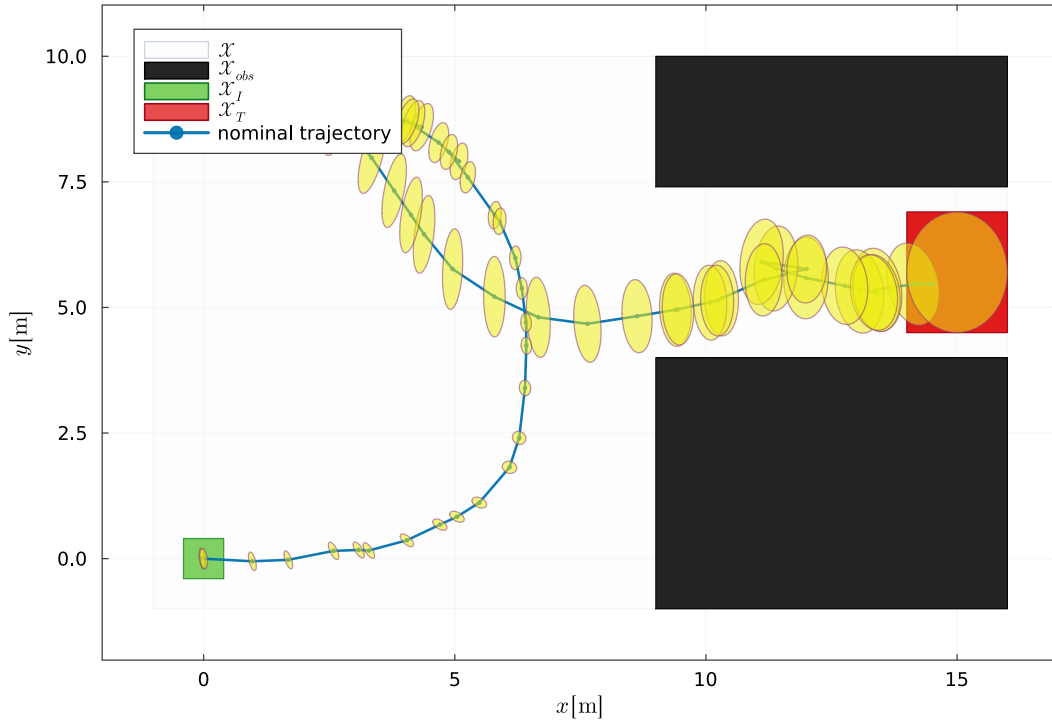
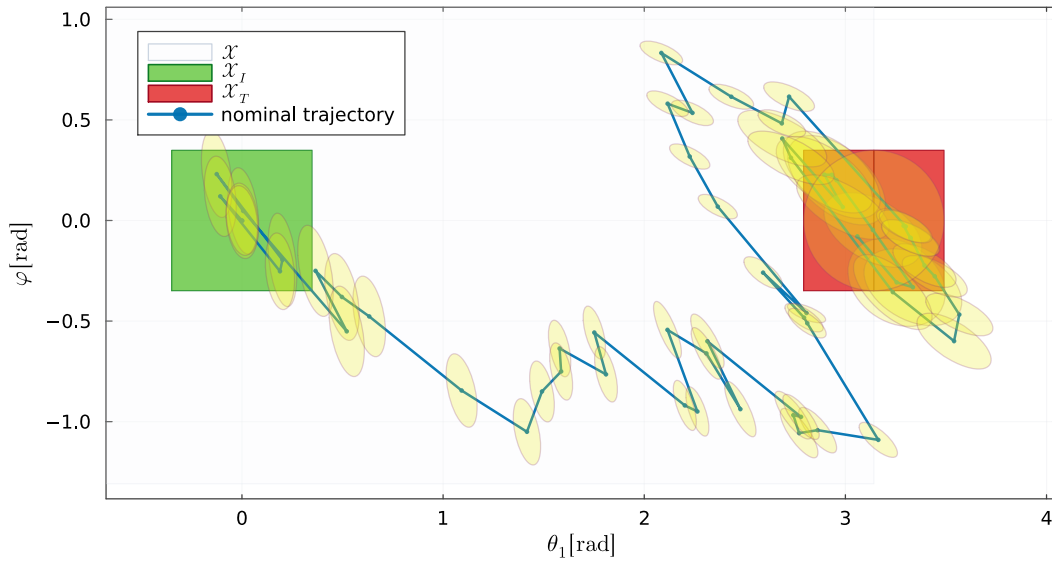
(a) (p_x, p_y) -projection.(b) (θ, φ) -projection.

Figure 5.14: Certified ellipsoidal chain for the reverse manoeuvre. The planar projection shows the certified local sets along the reverse parking motion, while the angular projection shows the evolution of the heading and articulation coordinates.

Table 5.6: Summary for the reverse articulated-vehicle benchmark.

Quantity	Value
Time step	$\Delta t = 0.2$
Abstract trajectory horizon	$N_{\text{abs}} = 48$
MPPI horizon	$H_{\text{MPPI}} = 60$
Coarse grid resolution	$h = (0.4, 0.4, 4^\circ, 4^\circ)$
MPPI samples per iteration	3800
MPPI iterations	60
MPPI temperature	$\lambda_{\text{MPPI}} = 1.3$
MPPI noise scale	$(\sigma_v, \sigma_{\bar{\sigma}}) = (0.55, 0.25)$
Certified chain	yes
Sampled initial states	12000
Reached target	12000/12000
Obstacle violations	0
State constraint violations	0
Input constraint violations	0
Abstract warm-start runtime	33.78 s
MPPI refinement runtime	54.57 s
LMI certification runtime	172.41 s

Discussion. The reverse manoeuvre confirms, and amplifies, the observations made on the forward benchmark. In the reverse case, these effects are even more visible because the motion contains two qualitatively different phases.

- **Planning difficulty.** The vehicle starts from a forward-facing configuration, must first create a curved alignment with the alley, and then enter the target region in reverse. The nominal motion is not a simple smooth arc: it requires several small corrections of heading and articulation, including a short forward adjustment near the end to realign the trailer before the final reverse insertion. This is a positive point for the planning stage of the pipeline. The coarse warm start provides the global topology of the manoeuvre, while MPPI refines it into a dynamically consistent trajectory without requiring an exhaustive search over all possible parking strategies.
- **Certification difficulty.** The transition between the forward and reverse phases is the main source of conservatism. Around this part of the motion, small deviations in heading or articulation can induce larger deviations in the trailer position. The local nonlinear effects are therefore stronger, and the interval bounds used to control the linearization error become more conservative. Consequently, the backward LMI step tends to certify smaller ellipsoids than in the forward manoeuvre.

*Parallel-parking manoeuvre***Parallel-parking manoeuvre: problem specification**

State and input constraints. The state is $x = (p_x, p_y, \theta, \varphi)$ and the input is $u = (v, \tilde{\sigma})$, with $\tilde{\sigma} = \tan(\delta)$. The admissible state domain is

$$\mathcal{X} = [0, 21] \times [0, 5.6] \times [-\pi, \pi] \times \left[-\frac{65\pi}{180}, \frac{65\pi}{180}\right],$$

and

$$\mathcal{U} = [-5, 5] \times [-\sigma_{\max}, \sigma_{\max}], \quad \sigma_{\max} = \tan\left(\frac{55\pi}{180}\right) \approx 1.428.$$

Obstacles. The obstacle set is defined by two planar rectangles representing the parked vehicles:

$$\mathcal{O}_{xy}^{(1)} = [0, 6.3] \times [0, 2.1], \quad \mathcal{O}_{xy}^{(2)} = [14.3, 21] \times [0, 2.1],$$

lifted to the full state space as

$$\mathcal{X}_O = \left\{x \in \mathcal{X} : (p_x, p_y) \in \mathcal{O}_{xy}^{(1)} \cup \mathcal{O}_{xy}^{(2)}\right\}.$$

Initial and target sets. The nominal initial state is

$$x_0 = (19.0, 3.8, 0, 0).$$

The initial and target sets are

$$\mathcal{X}_I = [18.0, 20.6] \times [3.5, 4.5] \times \left[-\frac{3\pi}{180}, \frac{3\pi}{180}\right] \times \left[-\frac{2\pi}{180}, \frac{2\pi}{180}\right],$$

and

$$\mathcal{X}_T = [10.0, 12.8] \times [0.5, 1.6] \times \left[-\frac{6\pi}{180}, \frac{6\pi}{180}\right] \times \left[-\frac{3\pi}{180}, \frac{3\pi}{180}\right].$$

The objective is to drive the vehicle from $\mathcal{X}_I$ into the parallel-parking slot $\mathcal{X}_T$, while satisfying the steering and articulation constraints and avoiding $\mathcal{X}_O$.

The parallel-parking manoeuvre considers a more geometrically constrained parking task. The vehicle starts in the road lane, in front of the available slot, and must enter the free space between two parked vehicles. In practice, the motion must first move away from the initial straight configuration, create the correct alignment, and then place the tractor-trailer system inside the target region. This benchmark is therefore demanding because the planner must identify the topology of the parking motion, while the certifier must handle tight obstacle clearance, steering limits, and articulation constraints.

We report the experiment in five parts: the coarse abstract trajectory used as warm

start, the refined MPPI trajectory, the certified ellipsoidal chain, a compact summary table of the main numerical results, and a short discussion of the obtained certificate.

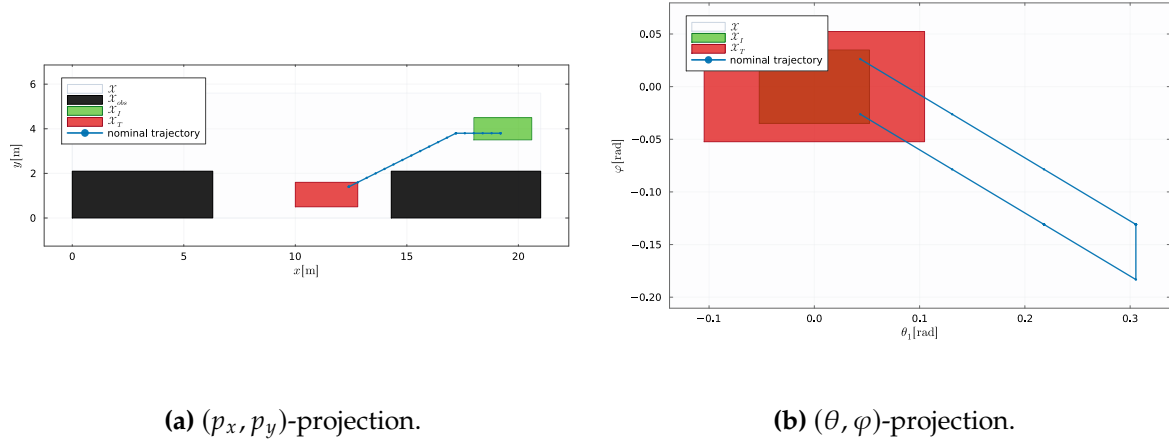


Figure 5.15: Coarse abstract trajectory used as warm start for the parallel-parking manoeuvre.

The left projection shows the geometric route used to enter the free slot between the two parked vehicles, while the right projection shows the corresponding heading and articulation angles.

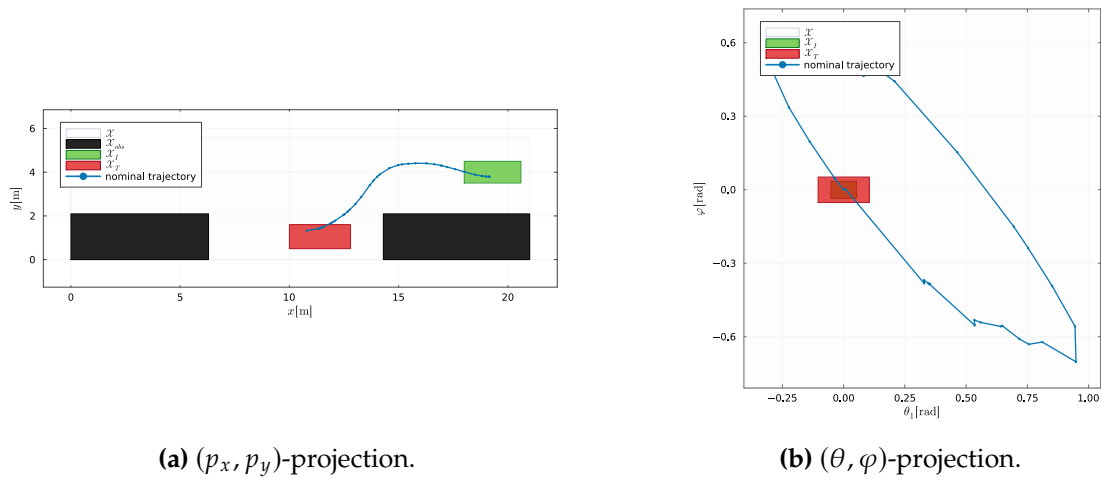


Figure 5.16: Refined MPPI trajectory for the parallel-parking manoeuvre. The trajectory smooths the coarse warm start and produces a dynamically consistent nominal motion satisfying the geometric structure of the parking task.

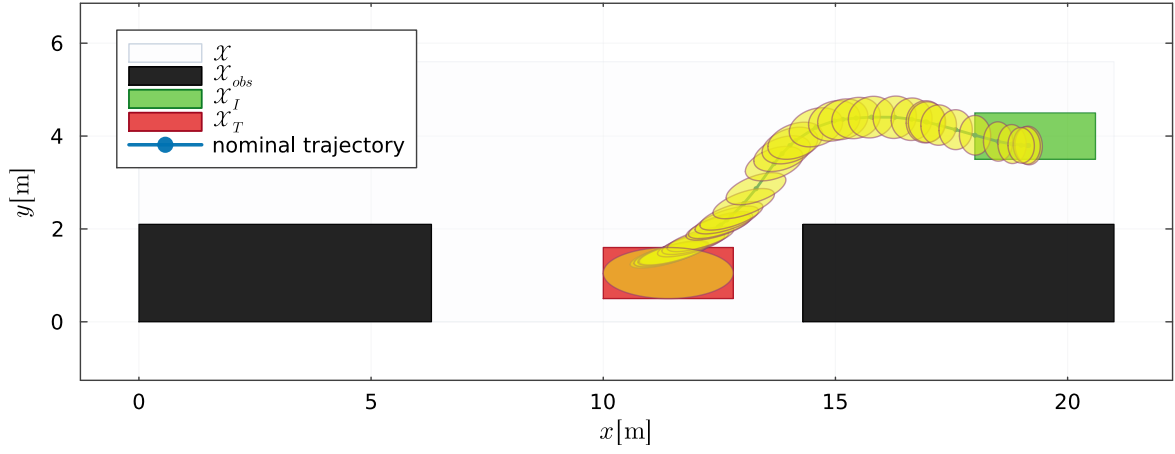
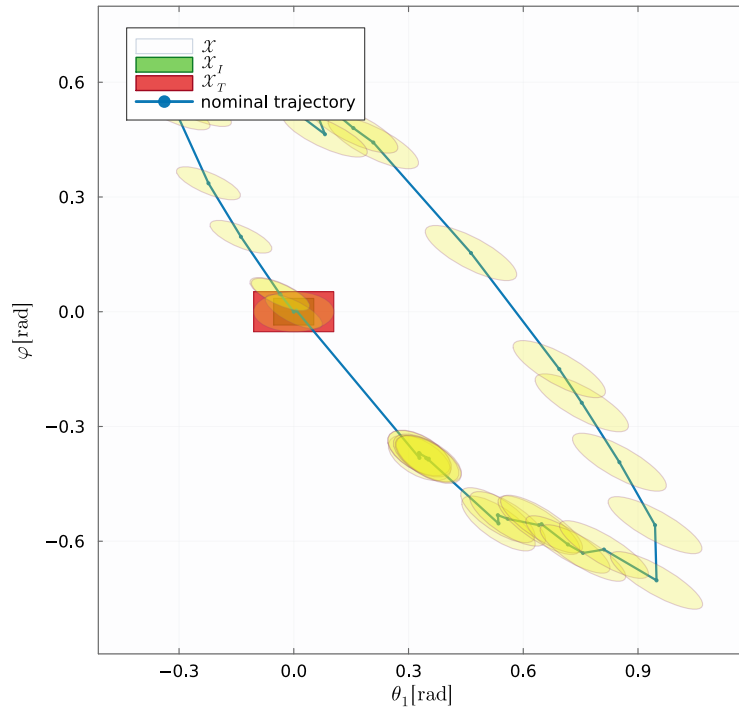
(a) (p_x, p_y) -projection.(b) (θ, φ) -projection.

Figure 5.17: Certified ellipsoidal chain for the parallel-parking manoeuvre. The planar projection shows the certified local sets along the parking motion, while the angular projection shows the evolution of the heading and articulation coordinates.

Table 5.7: Summary for the parallel-parking articulated-vehicle benchmark.

Quantity	Value
Time step	$\Delta t = 0.2$
Abstract trajectory horizon	$N_{\text{abs}} = 30$
MPPI horizon	$H_{\text{MPPI}} = 40$
Coarse grid resolution	$h = (0.4, 0.2, 5^\circ, 3^\circ)$
MPPI samples per iteration	1900
MPPI iterations	75
MPPI temperature	$\lambda_{\text{MPPI}} = 0.3$
MPPI noise scale	$(\sigma_v, \sigma_{\tilde{\sigma}}) = (0.15, 0.08)$
Certified chain	yes
Sampled initial states	12000
Reached target	12000/12000
Obstacle violations	0
State constraint violations	0
Input constraint violations	0
Abstract warm-start runtime	46.33 s
MPPI refinement runtime	49.65 s
LMI certification runtime	113.67 s

Discussion. The parallel-parking manoeuvre is successfully certified. This benchmark is mainly informative about the nominal-planning stage.

- **Smooth single-phase motion.** The certified trajectory is essentially one backward phase, with no forward–reverse switch. The nominal motion is therefore smooth, and the local dynamics vary progressively along the chain.
- **Favourable certification.** Because the trajectory is smooth and avoids sharp transitions, the backward LMI problems are comparatively benign. The certified ellipsoids do not exhibit a severe volume contraction along the manoeuvre.
- **Importance of the warm start.** The hard part is to obtain a good nominal parking trajectory. Reversing with a trailer is strongly nonlinear: without guidance, MPPI may produce trajectories close to obstacles or with excessive articulation. The abstract trajectory gives the correct parking topology and allows MPPI to use small perturbations,

$$\sigma_v = 0.15, \quad \sigma_{\tilde{\sigma}} = 0.08,$$

while still returning a smooth feasible motion.

Overall, this benchmark illustrates a favourable case for the pipeline: the abstract planner identifies the parking topology, MPPI refines it into a smooth trajectory, and the LMI certifier turns it into a local feedback-controlled reach-avoid certificate.

5.3 Supplementary animations

Table 5.8 contains QR codes linking to supplementary videos for the main benchmarks. In the digital version of the manuscript, the QR codes can be clicked directly to open the corresponding animations. These videos are intended as visual illustrations of the closed-loop behaviour; the formal guarantees are provided by the certified reach-avoid chains.²

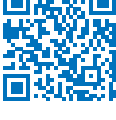
Benchmark	Content of the animation	Video
Simple pendulum	Closed-loop simulation of the certified swing-up behaviour on the single pendulum.	
Double pendulum	Closed-loop simulation of the certified swing-up behaviour on the double pendulum.	
Articulated vehicle: forward manoeuvre	Closed-loop simulation of the forward manoeuvre.	
Articulated vehicle: reverse manoeuvre	Closed-loop simulation of the reverse manoeuvre.	
Articulated vehicle: parallel parking	Closed-loop simulation of the parallel-parking manoeuvre.	

Table 5.8: Supplementary animations associated with the benchmarks.

Remark on the animations. The animations are only visual illustrations. The certified reach-avoid problem is defined on the mathematical state-space model, not on the full physical geometry of the rendered object. Hence, the guarantees apply to the certified state variables and specification sets. For the articulated vehicle, guaranteeing the full vehicle footprint would require additional collision constraints or inflated obstacle sets.

²For each animation, the initial condition is chosen inside the certified initial set. The closed-loop trajectory is then generated by applying, step by step, the local feedback controllers associated with the certified ellipsoidal chain.

5.4 Ablation, comparison, and qualitative studies

5.4.1 Planning-stage ablation: effect of the warm start

This subsection isolates the role of the nominal planning stage. The local certifier can only be applied after a dynamically consistent nominal trajectory has been found. The trajectory must first select a feasible geometric route, and in obstacle-rich problems this often amounts to choosing the right passage, or homotopy class, around the obstacles. A cold-started MPPI solver may therefore spend a large part of its sampling budget discovering this global route, instead of refining an already plausible motion.

This motivates the use of a coarse guide before MPPI refinement. The idea is close in spirit to RRT-guided MPPI [33], where a global planner biases the MPPI sampling distribution toward promising regions of trajectory space. In our pipeline, this role is played by the non-certified centered-abstraction path computed by Dionysos.jl. The abstraction provides the topology of the motion and an initial control sequence; MPPI then refines this coarse guide into a dynamically consistent nominal trajectory.

Maze warm-start ablation: dynamics and motivation

Dynamics. The state is $x = (p_x, p_y, \theta)$, the input is $u = (v, \tilde{\sigma})$, and the continuous dynamics are

$$\dot{p}_x = v \frac{\cos(\theta + \alpha)}{\cos(\alpha)}, \quad \dot{p}_y = v \frac{\sin(\theta + \alpha)}{\cos(\alpha)}, \quad \dot{\theta} = v \tan(\tilde{\sigma}), \quad \alpha = \arctan\left(\frac{\tan(\tilde{\sigma})}{2}\right). \quad (5.14)$$

The system is discretized with RK4 and sampling time $T_s = 0.3$.

Purpose. The maze is used as a planning-stage ablation: we compare cold-started MPPI with MPPI initialized from the centered-abstraction path.

Maze problem and MPPI settings

Sets. The domain, input set, initial state, and target set are

$$\mathcal{X} = [0, 4] \times [0, 10] \times [-\pi - 0.4, \pi + 0.4], \quad \mathcal{U} = [-1, 1]^2, \quad x_0 = (0.4, 0.4, 0), \quad (5.15)$$

$$\mathcal{X}_{\mathcal{T}} = [3.0, 3.6] \times [0.3, 0.8] \times \mathbb{R}. \quad (5.16)$$

The obstacle set is the union of three rectangular walls in the (p_x, p_y) -plane:

$$\mathcal{X}_O = ([1.0, 1.2] \times [0.0, 9.0]) \cup ([2.2, 2.4] \times [0.0, 5.0]) \cup ([2.2, 2.4] \times [6.0, 10.0]). \quad (5.17)$$

Abstraction and MPPI. The Dionysos abstraction uses resolutions $\eta_x = (0.2, 0.2, 0.2)$ and $\eta_u = (0.3, 0.3)$. Both MPPI variants use the same budget:

$$H = 120, \quad N_{\text{it}} = 5, \quad N_{\text{samples}} = 500, \quad \lambda = 1, \quad \sigma = (0.3, 0.4). \quad (5.18)$$

The cold start uses a zero control sequence, while the warm start uses the control sequence induced by the abstract Dionysos trajectory.

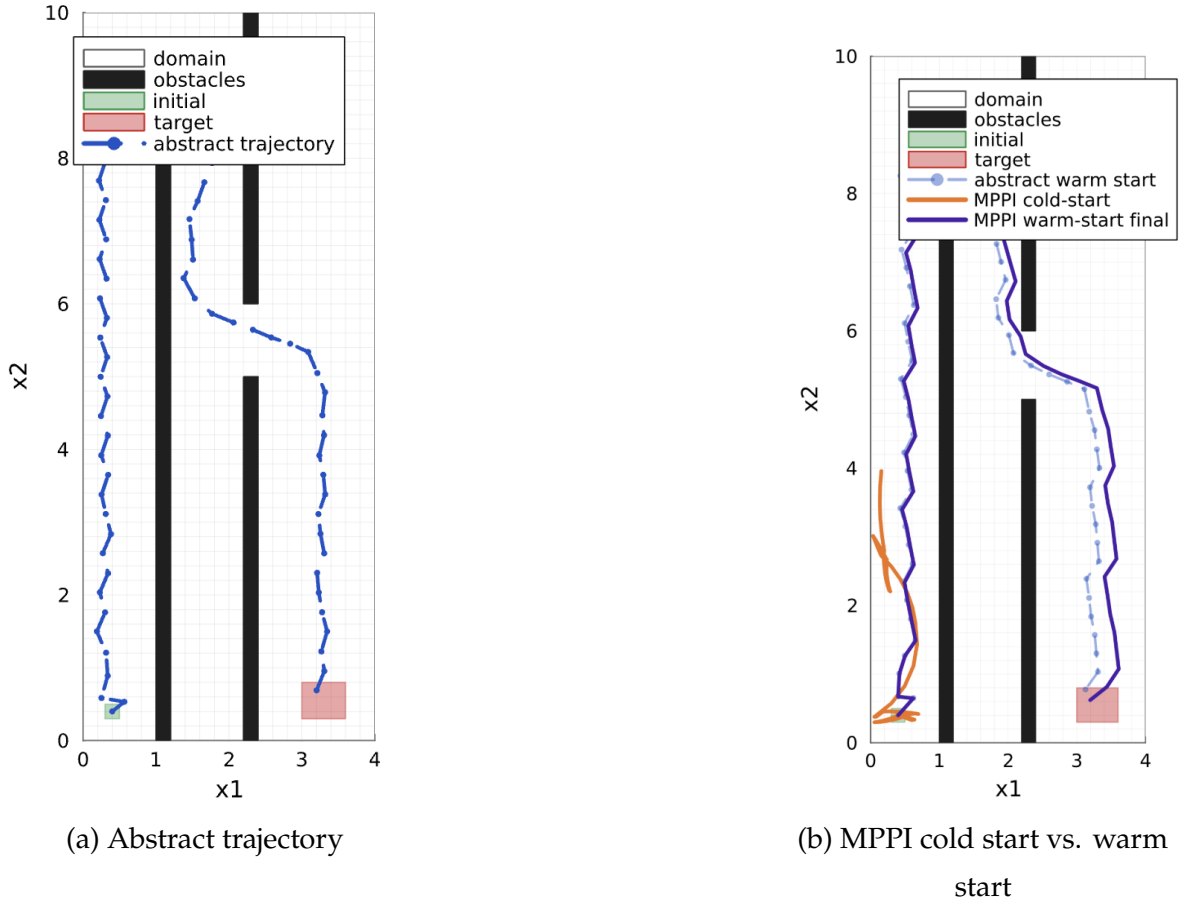


Figure 5.18: Left: the centered-abstraction path identifies the passage through the maze. Right: under the same MPPI budget, the cold start fails to discover the global route, whereas the warm start refines the abstract path into a feasible nominal trajectory.

Discussion. This experiment should not be interpreted as a proof that MPPI always needs a warm start. Rather, it illustrates a practical issue that becomes important in obstacle-rich tasks: with a fixed sampling budget, MPPI performance depends strongly on the initial sampling distribution, the horizon, the exploration variance, and the cost function. These choices are known to be delicate in practice, and planner-guided variants of MPPI have been proposed precisely to reduce this tuning burden [32, 33]. In this benchmark, the abstract trajectory gives MPPI useful global information before optimization starts. In particular:

- it identifies the relevant homotopy class through the maze;
- it provides a natural estimate of the horizon needed to reach the target;
- it allows the exploration noise to be chosen more tightly, since MPPI no longer has to discover the passage from scratch.

This is conceptually similar to RRT-guided MPPI, but the guide is obtained from a symbolic abstraction rather than from a sampling-based tree. Unlike RRT, this does not provide a probabilistic-completeness statement. However, in the benchmarks considered here, a coarse abstract trajectory is typically obtained at moderate cost and is sufficient to initialize MPPI in the right region of the trajectory space.

5.4.2 Abstraction baseline: comparison with a uniform abstraction

This subsection compares the proposed trajectory-guided ellipsoidal abstraction with classical uniform-grid abstractions. The purpose is not to benchmark all symbolic abstraction techniques, but to isolate a more specific question: *Is it enough to restrict a uniform abstraction to a region of the state space where the desired behaviour is expected to occur?*

We study this question on the articulated-vehicle dynamics (5.13) and on the forward manoeuvre specification (5.2.3).

The experiment should not be read as a cherry-picked failure case for uniform abstractions. The same tests were also run on the other articulated-vehicle manoeuvres and on the double-pendulum benchmark, with the same qualitative conclusion: restricting the abstraction to a local region is helpful, but it does not by itself remove the conservatism induced by growth-bound transitions.

Conservative transitions and growth bounds. As recalled in Section 2.2.1, a uniform abstraction constructs transitions by enclosing the one-step successors of each cell. If $R^{-1}(q) = H(c_q, r_q)$ is a hyperrectangular cell in the sense of Definition 2.10, and u is a constant representative input, an enclosure $\widehat{F}(R^{-1}(q), u)$ is computed, and every grid cell intersecting this enclosure is added as an abstract successor.

We use the growth-bound construction of Reissig et al. [8], as implemented in `Dionysos.jl`. The system given to `Dionysos` is the continuous-time vector field f_c . For a constant input u , let $\varphi_\tau(c_q, u)$ denote the sampled successor of the cell center over one sampling period τ . A growth bound gives a vector $\beta_\tau(r_q, u) \in \mathbb{R}_+^n$ such that

$$|\varphi_\tau(x, u) - \varphi_\tau(c_q, u)| \leq \beta_\tau(r_q, u), \quad \forall x \in H(c_q, r_q),$$

componentwise. The successor enclosure is then $\widehat{F}(R^{-1}(q), u) := H(\varphi_\tau(c_q, u), \beta_\tau(r_q, u))$. In the implementation, $\varphi_\tau(c_q, u)$ is evaluated with RK4, while the growth bound is computed from the continuous-time dynamics.

For the articulated vehicle, with $x = (p_x, p_y, \theta, \varphi)$ and $u = (v, \tilde{\sigma})$, the continuous-time state Jacobian satisfies

$$\left| \frac{\partial f_c}{\partial x}(x, u) \right| \leq J_b(u) := \begin{pmatrix} 0 & 0 & |v| & 0 \\ 0 & 0 & |v| & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{|v|}{L_1 L_2} (L_1 + |L_c| |\tilde{\sigma}|) \end{pmatrix}.$$

Since no additive disturbance is used in this baseline, we take

$$\beta_\tau(r_q, u) = \exp(J_b(u)\tau) r_q.$$

This enclosure is sound but conservative: it may intersect cells that are not reached by the actual sampled dynamics. These cells become spurious successors and increase the nondeterminism of the uniform abstraction.

Experimental protocol. We compare two uniform-grid abstractions generated with the symbolic abstraction and reach-avoid synthesis tools of `Dionysos.jl`. In both cases, the input data given to `Dionysos.jl` are the continuous-time articulated-vehicle dynamics, the state and input constraints, the reach-avoid specification, and the state and input grid resolutions. The resulting abstraction is uniform in the following sense: abstract states are axis-aligned grid cells, inputs are constant representatives from a finite input alphabet, and transitions are computed using the growth-bound enclosure described above.³

We consider two abstraction domains:

- **Global uniform abstraction.** The uniform grid is built on the full state domain $\mathcal{X}$ of the forward articulated-vehicle benchmark.
- **Local-tube uniform abstraction.** The same uniform-grid construction is restricted to a tube around the nominal MPPI trajectory $(\bar{x}_0, \dots, \bar{x}_N)$ used by the ellipsoidal certificate. The tube is defined as

$$\mathcal{X}_{\text{tube}} := \mathcal{X} \cap \bigcup_{k=0}^N H(\bar{x}_k, r_{\text{tube}}), \quad r_{\text{tube}} = (1.8, 1.6, 45^\circ, 45^\circ).$$

Thus, the local-tube baseline reduces the region where cells are created, but it still uses constant representative inputs and growth-bound transitions.

Both baselines use the same sampling time $\Delta t = 0.1$, input bounds, obstacles, initial set, and target set as the forward manoeuvre. The center successors used in the transition construction are evaluated with RK4, as in the other experiments. The input alphabet is obtained by discretizing the admissible input set with mesh $h_u = (0.5, 0.2)$. We test three state resolutions:

$$G_1 : h_x = (0.7, 0.7, 2\pi/8, 2\pi/8), \quad G_2 : h_x = (0.4, 0.4, 2\pi/12, 2\pi/12),$$

$$G_3 : h_x = (0.2, 0.2, 2\pi/16, 2\pi/16).$$

For each resolution, we construct the abstraction, solve the finite reach-avoid problem, and report the number of symbolic states, the number of transitions, the runtime, and the synthesis status. The reported times exclude the computation of the nominal MPPI trajectory; they only measure abstraction construction and controller synthesis. The purpose of this comparison is not to exhaustively tune the uniform baselines, but to show the orders of magnitude obtained with standard uniform-grid abstractions under the same reach-avoid specification.

³The uniform-abstraction baselines were generated using the existing symbolic abstraction and reach-avoid synthesis tools available in `Dionysos.jl`; see [9].

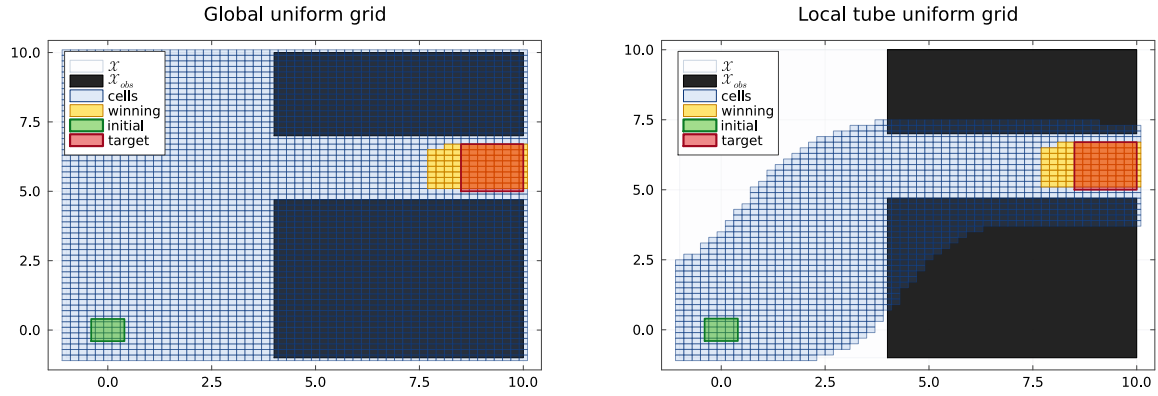


Figure 5.19: The left plot shows a standard uniform abstraction on the full state domain. The right plot restricts the abstraction to a tube around the nominal MPPI trajectory. In both cases, transitions are constructed using conservative growth bounds. The projection is shown in the (p_x, p_y) -plane; the cells are four-dimensional in the actual abstraction. Yellow cells denote the reach-avoid winning set Y^* ; see Definition 2.13. The baseline is successful only if this set intersects the abstract initial set, Definition 2.6.

Table 5.9: Uniform-grid abstraction baselines for the forward articulated-vehicle benchmark.

Method	Region	h_x	Cells	Transitions	Time [s]	Status
Uniform grid	global	G_1	10×10^3	24×10^6	9.6	fail
Uniform grid	global	G_2	60×10^3	178×10^6	48.0	fail
Uniform grid	global	G_3	282×10^3	837×10^6	348.5	fail
Uniform grid	local tube	G_1	1.7×10^3	2×10^6	2.7	fail
Uniform grid	local tube	G_2	10×10^3	21×10^6	8.2	fail
Uniform grid	local tube	G_3	49×10^3	122×10^6	32.9	fail
Ellipsoidal chain	trajectory-guided	–	60	59	139.4	success

Discussion. The experiment highlights two limitations of uniform abstractions.

- **Computational growth.** The global uniform abstraction does not solve the benchmark at any of the tested resolutions, and its size grows very quickly with grid refinement. At level G_1 , it already contains 10×10^3 cells and 24×10^6 transitions. At level G_3 , it reaches 282×10^3 cells and 837×10^6 transitions, with a construction and synthesis time of 348.5 seconds. These numbers are obtained on a four-dimensional benchmark only. They therefore suggest that global uniform abstractions rapidly become demanding in memory and computation, and would be difficult to generate for higher-dimensional systems.
- **Effect of localization.** Restricting the abstraction to a tube around the nominal trajectory strongly reduces the number of cells and transitions. For example, at level G_3 , the number of cells decreases from 282×10^3 to 49×10^3 , and the number of transitions from 837×10^6 to 122×10^6 . However, the local-tube abstraction

still does not solve the reach-avoid problem. Localization reduces the size of the finite model, but the abstraction remains large and still inherits the same conservative transition mechanism.

- **Growth-bound conservatism.** In both the global and local-tube cases, transitions are built by enclosing the successors of an entire cell under a constant representative input. The growth-bound enclosure may intersect many cells that are not reached by the true dynamics. These cells become spurious successors in the symbolic model. During reach-avoid synthesis, they must be treated as genuine nondeterministic outcomes. As a result, many inputs are rejected because the controller cannot guarantee progress and safety for all conservative successors, even when the nominal behaviour itself is feasible.

Thus, the failure of the uniform baselines is not only a matter of abstracting too large a region of the state space. The global abstraction is rapidly expensive to build and manipulate, while the tube-restricted abstraction remains affected by the nondeterminism introduced by growth-bound transitions. This motivates the abstraction proposed in this thesis.

5.4.3 Pipeline bottlenecks

The previous experiments show that the main difficulty is not only to find a nominal reach-avoid trajectory, but to find a trajectory that can be lifted into a certified feedback tube. As stated in Definition 3.1, certification requires more than nominal feasibility: each local set must be mapped inside the next one, while satisfying the input, state, obstacle, and local-model consistency constraints. Thus, a trajectory may solve the nominal planning problem 3.1 and still be difficult, or impossible, to certify through Problem 3.2.

Across the benchmarks, the observed bottlenecks can be summarized as three missing margins:

- *model margin*: the nonlinear dynamics must be locally well represented by the affine model used in the certificate;
- *control margin*: the nominal input must leave enough room for feedback corrections;
- *geometric margin*: the nominal motion must remain sufficiently far from obstacles and state constraints to contain a nontrivial tube.

These limitations do not contradict the soundness statement of Proposition 4.1. They only characterize when the sufficient local certificates used in the current pipeline become too conservative.

Conservatism of the local LMI certificate

The local LMI certificate is based on a sufficient replacement of the exact nonlinear inclusion. Instead of checking directly

$$f(\mathcal{E}_k, \kappa_k(\mathcal{E}_k), \mathcal{W}) \subseteq \mathcal{E}_{k+1}, \quad (5.19)$$

the certifier linearizes the discrete-time map around $(c_k, \bar{u}_k, 0)$, bounds the nonlinear remainder on a local box $\mathcal{B}_k$, and enforces a stronger inclusion of the form

$$\tilde{f}_k(\mathcal{E}_k, \kappa_k(\mathcal{E}_k), \mathcal{W}) \oplus \mathcal{H}_r \subseteq \mathcal{E}_{k+1}. \quad (5.20)$$

This is the geometric core of the certificate introduced in [Section 2.5](#) and detailed in [Appendix A](#).

The conservatism mainly enters through the remainder set $\mathcal{H}_k$. In the implementation, the nonlinear error is bounded by interval arithmetic on a local region. This gives sound bounds, but it may lose algebraic dependencies between repeated occurrences of the same variable [\[41\]](#). For example, evaluating $g(x) = x - x$ on $[-1, 1]$ by interval arithmetic gives $(-2, 2)$ as an enclosure, although g is identically zero. The same dependency effect appears, in a less trivial form, in RK4-expanded dynamics containing trigonometric terms, products, and trailer-angle couplings.

This explains two observations from the articulated-vehicle benchmarks:

- In the forward manoeuvre, the empirical inflation test of [Figure 5.11](#) suggests that the feedback sequence works on a region larger than the certified state. The gap comes from the fact that the LMI must prove the robust one-step inclusion [\(5.20\)](#), not merely succeed in sampled simulations.
- In the reverse manoeuvre, the direction changes and trailer realignment phases amplify local nonlinear effects. Small errors in heading or articulation can produce larger deviations of the trailer position, so the interval remainder becomes larger and the backward certificate tends to return smaller ellipsoids.

A second, more numerical, source of difficulty is scaling. The articulated-vehicle state mixes positions, angles, and articulation variables, while the input mixes velocity and steering. Without appropriate scaling, the SDP matrices may be poorly conditioned. Scaling does not change the mathematical certificate after mapping the solution back to the original coordinates, but it can make the feasible set easier for the solver to explore.

Possible improvements are therefore:

- use sharper nonlinear remainder representations, such as Taylor models, affine arithmetic, zonotopes [\[42\]](#), or local branch-and-bound refinements;
- adapt the linearization boxes $\mathcal{B}_k$ jointly with the ellipsoid size: smaller boxes reduce $\mathcal{H}_k$, but must still contain $\mathcal{E}_k$ and $\kappa_k(\mathcal{E}_k)$;
- replace the current ellipsoid-S-procedure certificate by less conservative local set representations or higher-order certificates, as in funnel and sum-of-squares approaches [\[37\]](#).

Input saturation and feedback authority

The simple pendulum experiment isolates a second bottleneck: nominal feasibility does not imply feedback authority. The certificate must enforce both the transition

inclusion and the admissibility of the whole feedback image,

$$f(\mathcal{E}_k, \kappa_k(\mathcal{E}_k), \mathcal{W}) \subseteq \mathcal{E}_{k+1}, \quad \kappa_k(\mathcal{E}_k) \subseteq \mathcal{U}. \quad (5.21)$$

Thus the relevant object is not only the nominal input $\bar{u}_k$, but the full range of inputs generated by the certified affine feedback law $\kappa_k(x) = K_k(x - c_k) + \ell_k$.

Here, $\bar{u}_k$ defines the nominal motion and the local linearization point, whereas ℓ_k is the optimized offset of the certified controller.

For a scalar input constraint $\mathcal{U} = [-u_{\max}, u_{\max}]$, the feedback image is the interval

$$\kappa_k(\mathcal{E}_k) = [\ell_k - r_k^u, \ell_k + r_k^u], \quad r_k^u = \sqrt{K_k P_k K_k^\top}. \quad (5.22)$$

Input admissibility is therefore equivalent to

$$|\ell_k| + r_k^u \leq u_{\max}. \quad (5.23)$$

Consequently, saturation of the nominal input does not by itself force r_k^u to vanish. However, when the nominal transition from c_k to c_{k+1} requires an input close to the boundary of $\mathcal{U}$, the certified offset ℓ_k is often also pushed close to that boundary. The remaining admissible radius r_k^u is then small, so the certifier must either reduce the feedback gain, shrink the source ellipsoid, or fail to certify the local transition. It's illustrated in the saturated simple-pendulum experiment of [Figure 5.3](#).

This observation suggests that the nominal planner should not only search for a feasible trajectory, but for a trajectory that leaves room for certification. In practice, this can be encouraged in several ways:

- *Input tightening.* Generate the nominal trajectory in a reduced input set $\mathcal{U}_{\text{plan}} \subset \mathcal{U}$, for instance by imposing a margin from the boundary of $\mathcal{U}$. The remaining input range is then reserved for feedback corrections during certification.
- *Saturation-aware MPPI cost.* Add a running cost that penalizes inputs close to the boundary of $\mathcal{U}$. This keeps the role of MPPI as a proposal mechanism, while shaping its solutions toward trajectories that are more likely to admit a local closed-loop certificate.

The key point is that the pipeline certifies a local closed-loop transition around the nominal motion, not the nominal input sequence itself. Preserving input margin during planning therefore directly improves the amount of feedback authority available to the LMI certificate.

Obstacle clearance and geometric margins

The third bottleneck is geometric. In the reach-avoid specification of [Section 3.1](#), obstacle avoidance is encoded as a constraint on the state: $\mathcal{E}_k \cap \mathcal{X}_O = \emptyset$. In the articulated-vehicle benchmarks, however, the obstacle set is defined mainly through the planar reference position (p_x, p_y) . This means that the mathematical specification controls the position of a reference point, whereas the physical vehicle has an extended body: tractor, trailer, orientation, and hitch geometry.

This distinction is visible in the Articulated vehicle forward manoeuvre video. A trajectory may satisfy the mathematical reach-avoid constraint while still passing close to obstacles because the obstacle set does not explicitly encode the full footprint of the vehicle. A more faithful specification would replace the point-obstacle condition by a footprint-aware one:

$$\mathcal{X}_O^{\text{body}} := \{x \in \mathcal{X} : \text{Body}(x) \cap O_{xy} \neq \emptyset\}. \quad (5.24)$$

Equivalently, one may impose a clearance condition $\text{dist}(\text{Body}(x), O_{xy}) \geq d_{\min}$.

This issue affects both planning and certification:

- During planning, insufficient clearance can produce nominal trajectories that are feasible as curves but leave no geometric room for a certified tube. This is especially relevant in the parallel-parking benchmark, where the hard part is to identify a parking topology with enough free space.
- During certification, even if the nominal state is safe, the whole ellipsoid must remain safe. Therefore, small clearance forces small ellipsoids, and may break the backward chain.

A direct improvement is to make MPPI clearance-aware. For example, the running cost can include a term of the form $\ell_{\text{clear}}(x) = \psi(d(x))$, where $d(x) = \text{dist}(\text{Body}(x), O_{xy})$ and ψ increases rapidly when $d(x)$ approaches zero. Safety-aware variants of MPPI also point in this direction. For instance, MPPI-DBaS incorporates discrete barrier states into MPPI to bias sampling and improve behaviour in tightly constrained environments [43]. More generally, clearance and barrier information should be treated as planning quantities, not only as post-hoc certification constraints.

Overall, these bottlenecks suggest that future improvements should target both sides of the pipeline. Stronger local certificates would reduce conservatism, but the planner should also become certification-aware: it should favor trajectories with input margin, obstacle clearance, smooth local dynamics, and enough geometric room for feedback corrections.

5.4.4 From certified chains to task libraries

Contract interpretation. A certified chain can be interpreted as a local contract for the closed-loop system [44]. Its entry set is the certified initial set, its exit set is the terminal ellipsoid, and its guarantee is that every admissible closed-loop trajectory starting in the entry set reaches the exit set while satisfying the state, input, and safety constraints. Thus, a chain is not only a tube around a nominal trajectory; it is a certified behaviour available to a higher-level planner.

Consider two certified chains $\mathfrak{C}_A$ and $\mathfrak{C}_B$, with terminal and initial sets $\mathcal{R}_A$ and $\mathcal{D}_B$. If $\mathcal{R}_A \subseteq \mathcal{D}_B$, then the two contracts can be concatenated. Indeed, the guarantee of $\mathfrak{C}_A$ implies that every trajectory starting in $\mathcal{D}_A$ reaches $\mathcal{R}_A$, and therefore enters the certified domain of $\mathfrak{C}_B$. Applying the controller of $\mathfrak{C}_B$ then gives the second guarantee. The concatenated feedback sequence is therefore a certified controller from $\mathcal{D}_A$ to $\mathcal{R}_B$.

Illustrative example. We illustrate this viewpoint on the articulated-vehicle dynamics (5.13). The first chain, shown in blue in Figure 5.20, represents a navigation contract: it takes the vehicle from an initial region, drives it around a roundabout, and brings it to an intermediate handoff region after the exit. The second chain, shown in red, represents a parking contract: starting from this handoff region, it performs a parallel-parking manoeuvre and ends in the parking target.

This illustrative example highlights the composition principle. If the terminal set of the first chain is contained in the entry set of the second one, namely $\mathcal{E}_{N_A}^A \subseteq \mathcal{E}_0^B$, then any trajectory covered by the first contract reaches a state from which the second contract can be applied. Thus, certified chains have the right structure to become reusable behavioural primitives. A future high-level planner could select and concatenate such contracts, instead of constructing one monolithic abstraction for the whole task.

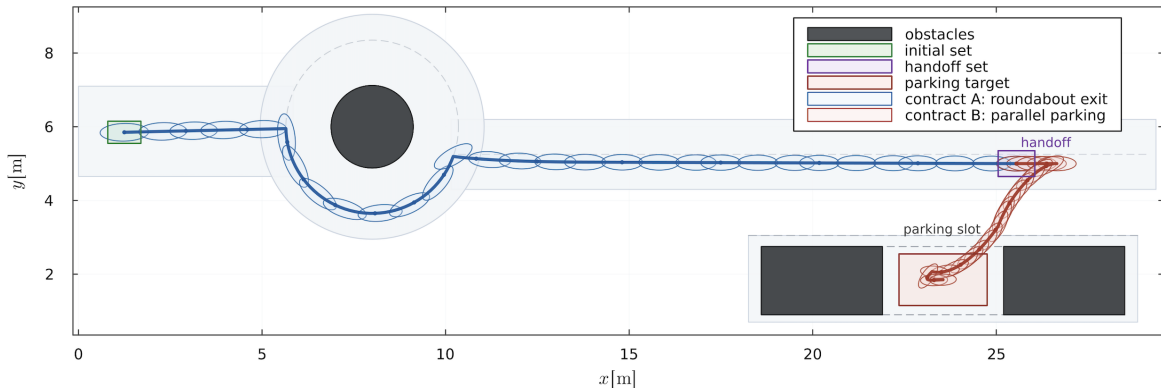


Figure 5.20: Illustrative composition of two chain contracts for the articulated-vehicle benchmark. The blue chain represents a navigation behaviour: the vehicle goes around a roundabout and reaches a handoff region after the exit. The red chain represents a parallel-parking behaviour starting from this handoff region. The inclusion $\mathcal{E}_{N_A}^A \subseteq \mathcal{E}_0^B$ illustrates the interface condition required to concatenate the two behaviours.

Conclusion and Future Work

This thesis started from a tension between two families of methods. On the one hand, abstraction-based control provides a systematic way to synthesize controllers with formal guarantees for continuous-state systems. On the other hand, constructing a global symbolic model may become prohibitively expensive or overly conservative for nonlinear systems. Trajectory optimization and sampling-based model predictive methods offer a complementary strength: they often find useful motions in complex systems, but the resulting trajectories are nominal and do not, by themselves, prove robust reach-avoid correctness.

The approach developed in this thesis is to use nominal trajectories not as certificates, but as guides for certification. Instead of abstracting the whole state space, the proposed method constructs a finite sequence of local feedback-labelled sets around a candidate behaviour. When the corresponding transition inclusions are certified, this sequence becomes both a robust controller on a subset of initial states and a partial abstraction of the concrete system.

6.1 Summary of contributions

The contributions of the thesis can be summarized as follows.

- **Certified local reach-avoid chains.** Chapter 3 introduced the notion of a certified local reach-avoid chain. Such a chain consists of local sets and feedback laws,

$$(\Omega_0, \dots, \Omega_N), \quad (\kappa_0, \dots, \kappa_{N-1}),$$

satisfying the reach-avoid constraints, the input constraints, and the robust transition inclusions

$$\text{Reach}_f(\Omega_k, \kappa_k) \subseteq \Omega_{k+1}.$$

By [Proposition 3.3](#), such a chain induces a finite-horizon feedback controller solving the robust reach-avoid problem on the certified initial set $\Omega_0 \cap \mathcal{X}_I$.

- **A trajectory-guided synthesis problem.** The computation was formulated as a two-stage problem in [Problem 3.3](#): a nominal planner first proposes a candidate reach-avoid trajectory, and a local certifier then attempts to lift it into a certified chain. This separation is central: the planner decides where certification should be attempted, whereas the certifier is the source of the formal guarantee.
- **A modular implementation in Dionysos.jl.** Chapter 4 instantiated this principle

through a modular pipeline combining:

- a coarse centered simulation abstraction, used as a non-certified global guide;
- an MPPI refinement step, used to obtain a dynamically consistent nominal trajectory;
- a backward ellipsoidal certifier, used to construct local ellipsoids $\mathcal{E}_k$ and affine feedback laws $\kappa_k(x) = \ell_k + K_k(x - c_k)$.

When the pipeline returns successfully and the local certificates are sound, the resulting feedback policy solves the reach-avoid specification on the certified initial set, as stated in Proposition 4.1.

- **An experimental evaluation on nonlinear benchmarks.** Chapter 5 evaluated the method on pendulum and articulated-vehicle reach-avoid tasks. The experiments show that trajectory-guided certification can produce nontrivial certified behaviours, while also identifying the main mechanisms that limit certification in practice.

6.2 Main findings

Certified chains are the central abstraction object. The main conceptual outcome of the thesis is the certified chain. It provides a compact representation of a verified behaviour of a nonlinear system:

- local sets encode where the behaviour is valid;
- feedback-labelled transitions encode how all admissible successors are propagated to the next set;
- the resulting object can be read either as a robust tube around a nominal trajectory or as a finite partial abstraction whose transitions are certified on concrete sets.

This interpretation separates abstraction from the requirement of covering the whole state space. Classical symbolic abstractions aim to represent all behaviours in a region. Here, the objective is more local and task-oriented: construct only the part of the abstraction needed to certify one relevant behaviour.

The planner proposes, the certifier proves. A recurring lesson of the thesis is that the nominal trajectory is only a scaffold. It may identify a useful route, suggest a horizon, and provide an initial control sequence, but it does not carry the reach-avoid guarantee. The guarantee appears only after the local transition inclusions, input constraints, and safety constraints have been certified.

This makes the architecture modular. In this thesis, the nominal trajectory is obtained from a centered abstraction warm start followed by MPPI, but the same certification problem could be attempted with trajectories produced by RRT, nonlinear MPC, direct collocation, mixed-integer optimization, or hand-designed manoeuvres.

Similarly, the ellipsoidal LMI certifier could be replaced by another local certification backend without changing the certified-chain viewpoint.

Nominal feasibility is not certifiability. The experiments show that reaching the target in simulation is not enough. A nominal trajectory may satisfy the disturbance-free reach-avoid constraints and still fail to admit a certified feedback tube. Certification is a set-based property: at each step, the certifier must prove

$$\text{Reach}_f(\mathcal{E}_k, \kappa_k) \subseteq \mathcal{E}_{k+1}, \quad \kappa_k(\mathcal{E}_k) \subseteq \mathcal{U}.$$

Thus the relevant object is not only the nominal state-input pair $(\bar{x}_k, \bar{u}_k)$, but the whole closed-loop image of the local set.

The simple pendulum illustrates this point clearly. If the nominal input is close to saturation, the affine feedback law has little room left for corrective actions. The constraint $\kappa_k(\mathcal{E}_k) \subseteq \mathcal{U}$ applies to the full input set generated by the feedback law, not only to the nominal value $\bar{u}_k$. Hence a trajectory can be nominally successful but locally uncertifiable.

Certifiability is governed by margins. Across the benchmark suite, certifiability is governed by three types of margins:

- **model margin:** the local affine model must approximate the nonlinear dynamics accurately enough on the certification region;
- **control margin:** the nominal input must leave enough authority for feedback corrections;
- **geometric margin:** the nominal motion must remain far enough from obstacles and state constraints to contain a nontrivial certified set.

These margins provide a unified interpretation of the experiments. The pendulum benchmarks isolate input and nonlinear effects in low dimension. The double pendulum tests the effect of dimension and nonlinear coupling. The articulated-vehicle benchmarks add nonholonomic geometry, obstacle clearance, and constrained parking manoeuvres. In all cases, the relevant question is not only whether a nominal motion exists, but whether this motion has enough surrounding margin to support a certified feedback-labelled tube.

Certified chains can be viewed as local contracts. A certified chain also admits a contract interpretation. Its entry set is the certified initial set, its exit set is the terminal set, and its guarantee is that every admissible closed-loop trajectory starting in the entry set reaches the exit set while satisfying the constraints.

This viewpoint suggests a compositional use of certified chains. If two chains are available and the exit set of the first lies inside the entry set of the second, then their guarantees can be concatenated. This is the first step toward a library of certified behaviours, where a higher-level planner would select and compose local contracts instead of relying on one monolithic abstraction of the whole state space.

6.3 Future work

The thesis opens several directions for future research.

- **Certification-aware nominal planning.** The experiments suggest that nominal planners should optimize not only for feasibility, but also for certifiability. Future planners should favour trajectories with input margin, obstacle clearance, smooth local dynamics, and enough room for feedback corrections. A natural direction is to let failed certification attempts return diagnostic information to the planner, such as insufficient input margin, excessive nonlinear remainder, or lack of geometric clearance.
- **Improving the certification backend.** The current ellipsoidal LMI certificate is sound but conservative. Future work should address the bottlenecks identified in Chapter 5, in particular the conservatism of local remainder bounds, the interaction between ellipsoid size and linearization boxes, input saturation, numerical conditioning, and obstacle-clearance limitations. This could involve sharper remainder models, adaptive local boxes, alternative set representations, or higher-order certificates.
- **Comparing planner–certifier combinations.** The proposed architecture is modular, but only one main instantiation was studied in depth. A systematic comparison should test different nominal planners and different local certifiers within the same certified-chain framework. This would clarify which combinations are most effective for different classes of nonlinear reach-avoid problems.
- **Extending the benchmarks.** The experiments mainly isolate the deterministic case and use simplified obstacle models. Future benchmarks should include non-trivial disturbance sets $\mathcal{W} \neq \{0\}$, higher-dimensional systems, and more realistic safety specifications. For the articulated vehicle, this means replacing point-position obstacle constraints by body-aware collision constraints depending on the tractor, trailer, heading, hitch geometry, and articulation angle.
- **Building libraries of certified behaviours.** The most promising long-term direction is to use certified chains as reusable motion primitives. Each chain can be stored as a local contract with an entry set, an exit set, a feedback policy, and a reach-avoid guarantee. A higher-level planner could then search over such contracts and compose certified behaviours to solve more complex tasks, [16].

Closing perspective. The main outcome of this thesis is not a single planner or a single LMI certificate, but a modular viewpoint: nominal trajectories select behaviours, and local feedback-labelled certificates turn these behaviours into partial abstractions. This viewpoint preserves the soundness philosophy of abstraction-based control while avoiding the need to model every possible behaviour of the system. In the long term, it suggests replacing monolithic abstraction construction by libraries of certified behaviours, where each local chain is a reusable contract that can be selected, verified, and composed by a higher-level planner.

Acknowledgement of generative AI use

I hereby confirm that I used ChatGPT as an assistance tool during the preparation of this manuscript. It was used for language-related and editorial tasks, including sentence reformulation, clarification of explanations, structuring of text, and LaTeX drafting assistance.

ChatGPT was also used to assist with coding tasks related to the numerical experiments, such as drafting, debugging, and organizing code. The experimental design, methodological choices, results, interpretations, and final decisions remain my own.

No generative AI tool was used to replace the scientific work carried out in this thesis.

References

- [1] J. C. Knight. “Safety Critical Systems: Challenges and Directions”. In: *Proceedings of the 24th International Conference on Software Engineering*. ACM, 2002, pp. 547–550. doi: [10.1145/581339.581406](https://doi.org/10.1145/581339.581406).
- [2] M. Althoff and J. M. Dolan. “Online Verification of Automated Road Vehicles Using Reachability Analysis”. In: *IEEE Transactions on Robotics* 30.4 (2014), pp. 903–918. doi: [10.1109/TRO.2014.2312453](https://doi.org/10.1109/TRO.2014.2312453).
- [3] M. Althoff, G. Frehse, and A. Girard. “Set Propagation Techniques for Reachability Analysis”. In: *Annual Review of Control, Robotics, and Autonomous Systems* 4.1 (2021), pp. 369–395. doi: [10.1146/annurev-control-071420-081941](https://doi.org/10.1146/annurev-control-071420-081941).
- [4] S. Bansal, M. Chen, S. Herbert, and C. J. Tomlin. “Hamilton-Jacobi Reachability: A Brief Overview and Recent Advances”. In: *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*. 2017, pp. 2242–2253. doi: [10.1109/CDC.2017.8263977](https://doi.org/10.1109/CDC.2017.8263977).
- [5] C. Fan, U. Mathur, S. Mitra, and M. Viswanathan. “Controller Synthesis Made Real: Reach-Avoid Specifications and Linear Dynamics”. In: *Computer Aided Verification*. Ed. by H. Chockler and G. Weissenbacher. Vol. 10981. Lecture Notes in Computer Science. Cham: Springer, 2018, pp. 347–366. doi: [10.1007/978-3-319-96145-3_19](https://doi.org/10.1007/978-3-319-96145-3_19).
- [6] C. Fan, Z. Qin, U. Mathur, Q. Ning, S. Mitra, and M. Viswanathan. “Controller Synthesis for Linear System With Reach-Avoid Specifications”. In: *IEEE Transactions on Automatic Control* 67.4 (2022), pp. 1713–1727. doi: [10.1109/TAC.2021.3069723](https://doi.org/10.1109/TAC.2021.3069723).
- [7] P. Tabuada. *Verification and Control of Hybrid Systems: A Symbolic Approach*. Springer, 2009. doi: [10.1007/978-1-4419-0224-5](https://doi.org/10.1007/978-1-4419-0224-5).
- [8] G. Reissig, A. Weber, and M. Rungger. “Feedback Refinement Relations for the Synthesis of Symbolic Controllers”. In: *IEEE Transactions on Automatic Control* 62.4 (2017), pp. 1781–1796. doi: [10.1109/TAC.2016.2593947](https://doi.org/10.1109/TAC.2016.2593947).
- [9] J. Calbert, A. Banse, B. Legat, and R. M. Jungers. “Dionysos.jl: A Modular Platform for Smart Symbolic Control”. In: *Proceedings of the JuliaCon Conferences* 6.66 (2024), p. 160. doi: [10.21105/jcon.00160](https://doi.org/10.21105/jcon.00160).
- [10] J. T. Betts. “Survey of Numerical Methods for Trajectory Optimization”. In: *Journal of Guidance, Control, and Dynamics* 21.2 (1998), pp. 193–207. doi: [10.2514/2.4231](https://doi.org/10.2514/2.4231).
- [11] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert. “Constrained Model Predictive Control: Stability and Optimality”. In: *Automatica* 36.6 (2000), pp. 789–814. doi: [10.1016/S0005-1098\(99\)00214-9](https://doi.org/10.1016/S0005-1098(99)00214-9).

- [12] J. B. Rawlings, D. Q. Mayne, and M. M. Diehl. *Model Predictive Control: Theory, Computation, and Design*. 2nd ed. Nob Hill Publishing, 2017.
- [13] G. Williams, A. Aldrich, and E. A. Theodorou. “Model Predictive Path Integral Control: From Theory to Parallel Computation”. In: *Journal of Guidance, Control, and Dynamics* 40.2 (2017), pp. 344–357. doi: [10.2514/1.G001921](https://doi.org/10.2514/1.G001921).
- [14] G. Williams, P. Drews, B. Goldfain, J. M. Rehg, and E. A. Theodorou. “Information-Theoretic Model Predictive Control: Theory and Applications to Autonomous Driving”. In: *IEEE Transactions on Robotics* 34.6 (2018), pp. 1603–1622. doi: [10.1109/TRO.2018.2865891](https://doi.org/10.1109/TRO.2018.2865891).
- [15] A. Girard and G. J. Pappas. “Approximation Metrics for Discrete and Continuous Systems”. In: *IEEE Transactions on Automatic Control* 52.5 (2007), pp. 782–798. doi: [10.1109/TAC.2007.895849](https://doi.org/10.1109/TAC.2007.895849).
- [16] A. Majumdar and R. Tedrake. “Funnel Libraries for Real-Time Robust Feedback Motion Planning”. In: *The International Journal of Robotics Research* 36.8 (2017), pp. 947–982. doi: [10.1177/0278364917712421](https://doi.org/10.1177/0278364917712421).
- [17] G. Pola and P. Tabuada. “Symbolic Models for Nonlinear Control Systems: Alternating Approximate Bisimulations”. In: *SIAM Journal on Control and Optimization* 48.2 (2009), pp. 719–733. doi: [10.1137/070698580](https://doi.org/10.1137/070698580).
- [18] 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*. HSCC ’16. ACM, 2016, pp. 99–104. doi: [10.1145/2883817.2883834](https://doi.org/10.1145/2883817.2883834).
- [19] J. Calbert, A. Girard, and R. M. Jungers. “Characterizing simulation relations through control architectures in abstraction-based control”. In: *Automatica* 190 (2026), p. 113040. doi: [10.1016/j.automatica.2026.113040](https://doi.org/10.1016/j.automatica.2026.113040).
- [20] J. Calbert, S. Mattenet, A. Girard, and R. M. Jungers. “Memoryless Concretization Relation”. In: *Proceedings of the 27th ACM International Conference on Hybrid Systems: Computation and Control*. HSCC ’24. Hong Kong, Hong Kong: Association for Computing Machinery, 2024, pp. 147–155. doi: [10.1145/3641513.3650132](https://doi.org/10.1145/3641513.3650132).
- [21] J. Calbert, L. N. Egidio, and R. M. Jungers. “Smart Abstraction Based on Iterative Cover and Non-Uniform Cells”. In: *IEEE Control Systems Letters* 8 (2024), pp. 2301–2306. doi: [10.1109/LCSYS.2024.3409106](https://doi.org/10.1109/LCSYS.2024.3409106).
- [22] J. Calbert, B. Legat, L. N. Egidio, and R. M. Jungers. “Alternating Simulation on Hierarchical Abstractions”. In: *Proceedings of the 60th IEEE Conference on Decision and Control*. 2021, pp. 593–598. doi: [10.1109/CDC45484.2021.9683448](https://doi.org/10.1109/CDC45484.2021.9683448).
- [23] L. N. Egidio, T. A. Lima, and R. M. Jungers. “State-feedback Abstractions for Optimal Control of Piecewise-affine Systems”. In: *Proceedings of the 61st IEEE Conference on Decision and Control*. 2022, pp. 7455–7460. doi: [10.1109/CDC51059.2022.9992495](https://doi.org/10.1109/CDC51059.2022.9992495).
- [24] F. Blanchini. “Set Invariance in Control”. In: *Automatica* 35.11 (1999), pp. 1747–1767. doi: [10.1016/S0005-1098\(99\)00113-2](https://doi.org/10.1016/S0005-1098(99)00113-2).

- [25] F. Blanchini and S. Miani. *Set-Theoretic Methods in Control*. Systems & Control: Foundations & Applications. Boston, MA: Birkhäuser, 2008. doi: [10.1007/978-0-8176-4606-6](https://doi.org/10.1007/978-0-8176-4606-6).
- [26] D. Q. Mayne, M. M. Seron, and S. V. Raković. “Robust Model Predictive Control of Constrained Linear Systems with Bounded Disturbances”. In: *Automatica* 41.2 (2005), pp. 219–224. doi: [10.1016/j.automatica.2004.08.019](https://doi.org/10.1016/j.automatica.2004.08.019).
- [27] A. Bemporad and M. Morari. “Control of Systems Integrating Logic, Dynamics, and Constraints”. In: *Automatica* 35.3 (1999), pp. 407–427. doi: [10.1016/S0005-1098\(98\)00178-2](https://doi.org/10.1016/S0005-1098(98)00178-2).
- [28] S. M. LaValle and J. J. Kuffner. “Randomized Kinodynamic Planning”. In: *The International Journal of Robotics Research* 20.5 (2001), pp. 378–400. doi: [10.1177/02783640122067453](https://doi.org/10.1177/02783640122067453).
- [29] S. Karaman and E. Frazzoli. “Sampling-based Algorithms for Optimal Motion Planning”. In: *The International Journal of Robotics Research* 30.7 (2011), pp. 846–894. doi: [10.1177/0278364911406761](https://doi.org/10.1177/0278364911406761).
- [30] H. J. Kappen. “Path Integrals and Symmetry Breaking for Optimal Control Theory”. In: *Journal of Statistical Mechanics: Theory and Experiment* 2005.11 (2005), P11011. doi: [10.1088/1742-5468/2005/11/P11011](https://doi.org/10.1088/1742-5468/2005/11/P11011).
- [31] E. Theodorou, J. Buchli, and S. Schaal. “A Generalized Path Integral Control Approach to Reinforcement Learning”. In: *Journal of Machine Learning Research* 11 (2010), pp. 3137–3181. URL: <https://www.jmlr.org/papers/v11/theodorou10a.html>.
- [32] G. Williams, A. Aldrich, and E. A. Theodorou. *Model Predictive Path Integral Control using Covariance Variable Importance Sampling*. 2015. arXiv: [1509.01149](https://arxiv.org/abs/1509.01149) [cs.RO].
- [33] C. Tao, H. Kim, and N. Hovakimyan. “RRT Guided Model Predictive Path Integral Method”. In: *Proceedings of the American Control Conference*. 2023, pp. 776–781. doi: [10.23919/ACC55779.2023.10155837](https://doi.org/10.23919/ACC55779.2023.10155837).
- [34] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory*. Vol. 15. SIAM Studies in Applied Mathematics. Philadelphia, PA: Society for Industrial and Applied Mathematics, 1994. doi: [10.1137/1.9781611970777](https://doi.org/10.1137/1.9781611970777).
- [35] L. Schäfer and M. Althoff. “Robust Optimal Control Using Set-based Reachability Analysis”. In: *2025 European Control Conference*. 2025, pp. 372–379. doi: [10.23919/ECC65951.2025.11187216](https://doi.org/10.23919/ECC65951.2025.11187216).
- [36] M. Serry, L. Yang, N. Ozay, and J. Liu. “Safe Tracking Control of Discrete-Time Nonlinear Systems Using Backward Reachable Sets”. In: *Proceedings of the American Control Conference*. 2024, pp. 1172–1179.
- [37] A. Majumdar, A. A. Ahmadi, and R. Tedrake. “Control Design along Trajectories with Sums of Squares Programming”. In: *Proceedings of the IEEE International Conference on Robotics and Automation*. 2013, pp. 4054–4061. doi: [10.1109/ICRA.2013.6631149](https://doi.org/10.1109/ICRA.2013.6631149).

- [38] R. Tedrake, I. R. Manchester, M. Tobenkin, and J. W. Roberts. “LQR-Trees: Feedback Motion Planning via Sums-of-Squares Verification”. In: *The International Journal of Robotics Research* 29.8 (2010), pp. 1038–1052. doi: [10.1177/0278364910369189](https://doi.org/10.1177/0278364910369189).
- [39] W. Langson, I. Chryssochoos, S. V. Raković, and D. Q. Mayne. “Robust Model Predictive Control Using Tubes”. In: *Automatica* 40.1 (2004), pp. 125–133. doi: [10.1016/j.automatica.2003.08.009](https://doi.org/10.1016/j.automatica.2003.08.009).
- [40] P. Bolzern and A. Locatelli. “A Comparative Study of Different Solutions to the Path-Tracking Problem for an Articulated Vehicle”. In: *Proceedings of the 2002 IEEE International Conference on Control Applications*. Vol. 1. 2002, pp. 427–434. doi: [10.1109/CCA.2002.1040224](https://doi.org/10.1109/CCA.2002.1040224).
- [41] R. E. Moore, R. B. Kearfott, and M. J. Cloud. *Introduction to Interval Analysis*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2009. ISBN: 9780898716696. doi: [10.1137/1.9780898717716](https://doi.org/10.1137/1.9780898717716).
- [42] A. Girard. “Reachability of Uncertain Linear Systems Using Zonotopes”. In: *Hybrid Systems: Computation and Control*. Vol. 3414. Lecture Notes in Computer Science. Springer, 2005, pp. 291–305. doi: [10.1007/978-3-540-31954-2_19](https://doi.org/10.1007/978-3-540-31954-2_19).
- [43] F. Wang, Y. Cheng, and C. Tao. “MPPI-DBaS: Safe Trajectory Optimization with Adaptive Exploration”. In: *arXiv preprint arXiv:2502.14387* (2025). doi: [10.48550/arXiv.2502.14387](https://doi.org/10.48550/arXiv.2502.14387). arXiv: 2502.14387 [cs.RO].
- [44] R. R. Burridge, A. A. Rizzi, and D. E. Koditschek. “Sequential Composition of Dynamically Dexterous Robot Behaviors”. In: *The International Journal of Robotics Research* 18.6 (1999), pp. 534–555. doi: [10.1177/02783649922066385](https://doi.org/10.1177/02783649922066385).

Local LMI Certificate

This appendix recalls the one-step semidefinite certificate used in Section 4.4. The presentation follows Theorem 2 of [21], but is written with the notation of this manuscript and restricted to the single-transition problem needed by the backward certifier.

Role of the appendix. At one step of the backward construction, the successor ellipsoid is fixed. The certifier searches for a predecessor ellipsoid and a local affine feedback law such that every state in the predecessor ellipsoid is mapped into the successor ellipsoid, for every admissible disturbance, while satisfying the input constraints.

Ellipsoid convention. Throughout the manuscript, ellipsoids are written with a shape matrix:

$$\mathcal{E}(c, P) := \{x \in \mathbb{R}^{n_x} : (x - c)^\top P^{-1}(x - c) \leq 1\}, \quad P \in \mathbb{S}_{++}^{n_x}. \quad (\text{A.1})$$

Equivalently, if $L \in \mathbb{S}_{++}^{n_x}$ satisfies $P = L^2$, then

$$x \in \mathcal{E}(c, P) \iff x = c + Lz, \quad \|z\| \leq 1. \quad (\text{A.2})$$

Thus the precision matrix is $Q = P^{-1} = L^{-2}$. This is the matrix denoted by P in [21]; here we reserve P for the shape matrix used in the main text.

A.1 Local transition problem

Consider the nonlinear discrete-time transition system $S = (\mathcal{X}, \mathcal{U}, F)$, with bounded disturbances, where $F(x, u) := \{f(x, u, w) : w \in \mathcal{W}\}$. Here $\mathcal{X} \subseteq \mathbb{R}^{n_x}$, $\mathcal{U} \subseteq \mathbb{R}^{n_u}$, $\mathcal{W} \subseteq \mathbb{R}^{n_w}$, and $f : \mathcal{X} \times \mathcal{U} \times \mathcal{W} \rightarrow \mathcal{X}$ is a nonlinear one-step transition map. In the benchmarks of this thesis, f is the sampled-data map obtained by applying an RK4 discretization to the underlying continuous-time dynamics. We assume that f is continuous and that its Jacobian admits a Lipschitz bound on the local certification regions used below.

Equivalently, the uncertain one-step dynamics are written as

$$x^+ = f(x, u, w), \quad x \in \mathcal{X}, \quad u \in \mathcal{U}, \quad w \in \mathcal{W}. \quad (\text{A.3})$$

At a given backward step, the following objects are fixed:

- a successor ellipsoid

$$\xi^+ = \mathcal{E}(c^+, P^+), \quad P^+ \in \mathbb{S}_{++}^{n_x}, \quad (\text{A.4})$$

- a prescribed predecessor center $c \in \mathbb{R}^{n_x}$,
- a nominal input $\bar{u} \in \mathcal{U}$,
- a local linearization point $\bar{p} = (c, \bar{u}, 0)$.

The objective is to find a predecessor ellipsoid

$$\xi = \mathcal{E}(c, P), \quad P = L^2, \quad L \in \mathbb{S}_{++}^{n_x}, \quad (\text{A.5})$$

and a local affine feedback law

$$\kappa(x) = K(x - c) + \ell, \quad (\text{A.6})$$

such that

$$f(x, \kappa(x), w) \in \xi^+, \quad \forall x \in \xi, \quad \forall w \in \mathcal{W}, \quad (\text{A.7})$$

and

$$\kappa(\xi) \subseteq \mathcal{U}. \quad (\text{A.8})$$

The first condition is the robust transition certificate. The second condition ensures that the local feedback law only applies admissible inputs.

Disturbance and input sets. As in [21], the exogenous input takes values in the convex hull of finitely many points:

$$\mathcal{W} = \text{co}\{w_1, \dots, w_{N_w}\}, \quad 0 \in \mathcal{W}. \quad (\text{A.9})$$

This disturbance set introduces nondeterminism in the transition map F . It can represent either non-modelled behaviours or adversarial disturbances.

The admissible input set is represented as an intersection of quadratic constraints:

$$\mathcal{U} = \bigcap_{a=1}^{N_u} \mathcal{U}_a, \quad \mathcal{U}_a := \{u \in \mathbb{R}^{n_u} : \|U_a u\| \leq 1\}. \quad (\text{A.10})$$

This includes, for instance, ellipsoidal and box-like constraints after a suitable choice of the matrices U_a .

A.2 Local affine model and nonlinear remainder

The nonlinear transition map is linearized around $\bar{p} = (c, \bar{u}, 0)$:

$$\tilde{f}(x, u, w) = Ax + Bu + Ew + g, \quad (\text{A.11})$$

where

$$A = \partial_x f(\bar{p}), \quad B = \partial_u f(\bar{p}), \quad E = \partial_w f(\bar{p}), \quad (\text{A.12})$$

and, since the nominal disturbance is 0,

$$g = f(c, \bar{u}, 0) - Ac - B\bar{u}. \quad (\text{A.13})$$

The affine model is not used as an exact replacement for the nonlinear dynamics. Instead, the nonlinear error is bounded by a set-valued remainder. We assume that, on the local certification region, the Jacobian of f admits a componentwise Lipschitz bound $L_f \in \mathbb{R}_+^{n_x}$. Here, $(L_f)_i$ is a Lipschitz constant for the gradient of the i -th component of f on this region. We denote by $\mathbb{B}(a, \rho) := \{y : \|y - a\|_2 \leq \rho\}$ the Euclidean ball centered at a with radius ρ . If

$$\xi \subseteq \mathbb{B}(c, \sqrt{\delta_X}), \quad \kappa(\xi) \subseteq \mathbb{B}(\bar{u}, \sqrt{\delta_U}), \quad (\text{A.14})$$

then, with

$$\delta_W := \max_{j=1, \dots, N_w} \|w_j\|^2, \quad r^2 := \delta_X + \delta_U + \delta_W, \quad (\text{A.15})$$

the nonlinear remainder is bounded by the hyperrectangle

$$\mathcal{H}_r := H\left(0, \frac{1}{2}L_f r^2\right). \quad (\text{A.16})$$

Consequently, it is sufficient to certify the stronger inclusion

$$\tilde{f}(\xi, \kappa(\xi), \mathcal{W}) \oplus \mathcal{H}_r \subseteq \xi^+. \quad (\text{A.17})$$

Indeed, this inclusion implies the nonlinear transition condition (A.7). This step is conservative: the same remainder set $\mathcal{H}_r$ is used for the whole predecessor ellipsoid.

A.3 Convex change of variables

The feedback gain K and the shape factor L appear bilinearly when the predecessor ellipsoid and the controller are optimized jointly. To obtain LMIs, we introduce the change of variables

$$K = FL^{-1}, \quad F \in \mathbb{R}^{n_u \times n_x}. \quad (\text{A.18})$$

For every $x = c + Lz$, with $\|z\| \leq 1$, the affine controller becomes

$$\kappa(x) = K(x - c) + \ell = Fz + \ell. \quad (\text{A.19})$$

Thus the closed-loop affine image depends linearly on the decision variables L, F, ℓ .

Let $v_i, i = 1, \dots, 2^{n_x}$, denote the vertices of the remainder box $\mathcal{H}_r$, and let $w_j \in \text{vert}(\mathcal{W})$. Define the closed-loop offset relative to the successor center by

$$\mu := g + Ac + B\ell - c^+. \quad (\text{A.20})$$

For fixed vertices w_j and v_i , the affine closed-loop image relative to c^+ is then

$$\mu + (AL + BF)z + Ew_j + v_i. \quad (\text{A.21})$$

Using the definition $g = f(c, \bar{u}, 0) - Ac - B\bar{u}$, the offset can be written as

$$\mu = f(c, \bar{u}, 0) + B(\ell - \bar{u}) - c^+. \quad (\text{A.22})$$

This expression separates the role of the nominal input from the role of the certified feedback offset. The nominal input $\bar{u}$ defines the input component of the linearization point $(c, \bar{u}, 0)$, whereas $\ell = \kappa(c)$ is the input applied by the certified affine controller at the ellipsoid center.

A.4 LMI transition certificate

The decision variables are

$$L \in \mathbb{S}_{++}^{n_x}, \quad F \in \mathbb{R}^{n_u \times n_x}, \quad \ell \in \mathbb{R}^{n_u}, \quad (\text{A.23})$$

together with scalars

$$\delta_X \geq 0, \quad \delta_U \geq 0, \quad \phi \geq 0, \quad \beta_{ij} \geq 0, \quad \tau_a \geq 0. \quad (\text{A.24})$$

The LMIs below are sufficient to certify one robust transition.

Proposition A.1 (One-step ellipsoidal transition certificate). *Assume that the local affine model (A.11) and the remainder bound (A.16) are valid on the certification region. If there exist decision variables satisfying the LMIs (A.28)–(A.32), then the ellipsoid $\xi = \mathcal{E}(c, L^2)$ and the feedback law*

$$\kappa(x) = FL^{-1}(x - c) + \ell \quad (\text{A.25})$$

satisfy

$$f(x, \kappa(x), w) \in \xi^+, \quad \forall x \in \xi, \quad \forall w \in \mathcal{W}, \quad (\text{A.26})$$

and

$$\kappa(\xi) \subseteq \mathcal{U}. \quad (\text{A.27})$$

State-size bound. The first LMI enforces $\xi \subseteq \mathbb{B}(c, \sqrt{\delta_X})$:

$$\begin{pmatrix} I & L \\ L & \delta_X I \end{pmatrix} \geq 0. \quad (\text{A.28})$$

Input-deviation bound. The second LMI enforces $\kappa(\xi) \subseteq \mathbb{B}(\bar{u}, \sqrt{\delta_U})$:

$$\begin{pmatrix} \phi I & 0 & F^\top \\ 0 & \delta_U - \phi & (\ell - \bar{u})^\top \\ F & \ell - \bar{u} & I \end{pmatrix} \geq 0. \quad (\text{A.29})$$

Robust transition inclusion. For every vertex v_i of $\mathcal{H}_r$ and every vertex w_j of $\mathcal{W}$,

$$\begin{pmatrix} \beta_{ij}I & 0 & (AL + BF)^\top \\ 0 & 1 - \beta_{ij} & (\mu + Ew_j + v_i)^\top \\ AL + BF & \mu + Ew_j + v_i & P^+ \end{pmatrix} \geq 0, \quad (\text{A.30})$$

for all $i = 1, \dots, 2^{n_x}$, $j = 1, \dots, N_w$.

This certifies

$$\tilde{f}(\xi, \kappa(\xi), \mathcal{W}) \oplus \mathcal{H}_r \subseteq \xi^+. \quad (\text{A.31})$$

Hard input constraints. For each quadratic input constraint $\mathcal{U}_a$, impose

$$\begin{pmatrix} \tau_a I & 0 & F^\top U_a^\top \\ 0 & 1 - \tau_a & \ell^\top U_a^\top \\ U_a F & U_a \ell & I \end{pmatrix} \geq 0, \quad a = 1, \dots, N_u. \quad (\text{A.32})$$

This enforces $\kappa(\xi) \subseteq \mathcal{U}_a$ for every a , and therefore $\kappa(\xi) \subseteq \mathcal{U}$.

A.5 Why the LMIs imply the certificate

The proof follows the S-procedure and the Schur complement, as in [21]. The point of the change of variables is that all inequalities remain convex in L, F, ℓ .

First, since $x = c + Lz$ and $\|z\| \leq 1$, the condition $\xi \subseteq \mathbb{B}(c, \sqrt{\delta_X})$ is equivalent to

$$L^2 \leq \delta_X I. \quad (\text{A.33})$$

By the Schur complement, this is exactly (A.28).

Second, in the normalized coordinate z , the input-deviation condition is

$$\|z\| \leq 1 \implies \|Fz + \ell - \bar{u}\|^2 \leq \delta_U. \quad (\text{A.34})$$

Applying the S-procedure to this quadratic implication, followed by the Schur complement, gives (A.29).

Third, because both $\mathcal{W}$ and $\mathcal{H}_r$ are polytopic and ξ^+ is convex, it is sufficient to verify the transition inclusion at the vertices of these two polytopes. For fixed i, j , the desired implication is

$$\|z\| \leq 1 \implies \mu + (AL + BF)z + Ew_j + v_i \in \mathcal{E}(c^+, P^+) - c^+. \quad (\text{A.35})$$

Equivalently,

$$\|z\| \leq 1 \implies (\mu + (AL + BF)z + Ew_j + v_i)^\top (P^+)^{-1} (\mu + (AL + BF)z + Ew_j + v_i) \leq 1. \quad (\text{A.36})$$

The S-procedure and the Schur complement give (A.30).

Finally, each hard input constraint is the implication

$$\|z\| \leq 1 \implies \|U_a(Fz + \ell)\| \leq 1. \quad (\text{A.37})$$

The same argument gives (A.32). Hence feasibility of (A.28)–(A.32) provides a valid local transition certificate.

Conservatism. The certificate is sufficient, not necessary. Infeasibility of the LMIs does not prove that no local controller exists. Conservatism enters through the common linearization remainder $\mathcal{H}_r$, the use of scalar bounds δ_X, δ_U , and the S-procedure relaxations.

The detailed LMI construction follows [21, Sec. IV-B, Thm. 2], up to the ellipsoid convention used in this thesis.

A.6 Scaling and numerical conditioning

Some local SDP problems are solved in scaled coordinates in order to improve the conditioning of the LMI matrices. This scaling is not an additional approximation: it is an invertible local change of variables.

Let D_x , D_x^+ , and D_u be invertible diagonal matrices. Around a transition from center c to center c^+ , define the scaled variables

$$\hat{x} := D_x^{-1}(x - c), \quad \hat{u} := D_u^{-1}(u - \bar{u}), \quad \hat{x}^+ := (D_x^+)^{-1}(x^+ - c^+).$$

The corresponding scaled dynamics are

$$\hat{f}(\hat{x}, \hat{u}, w) := (D_x^+)^{-1} (f(c + D_x \hat{x}, \bar{u} + D_u \hat{u}, w) - c^+).$$

$$x = c + D_x \hat{x}, \quad x^+ = c^+ + D_x^+ \hat{x}^+, \quad u = \bar{u} + D_u \hat{u}$$

same certificate, different coordinates

(a) scaled coordinates

(b) physical coordinates

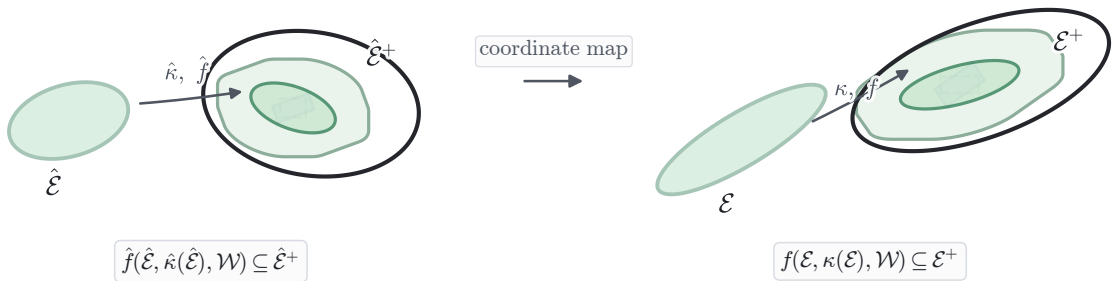


Figure A.1: Invariance of the local transition certificate under affine state and input scalings. The scaling changes only the numerical coordinates used by the SDP, not the physical transition being certified.

One-dimensional example. Consider the scalar system

$$x^+ = x + u + w, \quad w \in [-1, 1]. \quad (\text{A.38})$$

We want to certify the transition $[-10, 10] \rightarrow [-6, 6]$. Introduce the scaled coordinate $x = 10\hat{x}$ and $x^+ = 10\hat{x}^+$. The scaled dynamics are obtained by substituting this change of variables into the original system:

$$10\hat{x}^+ = 10\hat{x} + u + w, \quad \text{hence} \quad \hat{x}^+ = \hat{x} + \frac{1}{10}u + \frac{1}{10}w. \quad (\text{A.39})$$

Thus the scaling has not modified the physical dynamics; it has only changed the units in which the state is represented.

Choose the scaled feedback $u = -5\hat{x}$. In physical coordinates, this is $\kappa(x) = -x/2$. Then

$$\hat{x}^+ = \frac{1}{2}\hat{x} + \frac{1}{10}w. \quad (\text{A.40})$$

Therefore, if $\hat{x} \in [-1, 1]$ and $w \in [-1, 1]$, then $\hat{x}^+ \in [-0.6, 0.6]$. Mapping back to physical coordinates gives $x^+ \in [-6, 6]$. Equivalently, in the original coordinates,

$$x^+ = x - \frac{1}{2}x + w = \frac{1}{2}x + w, \quad (\text{A.41})$$

so $x \in [-10, 10]$ and $w \in [-1, 1]$ also imply $x^+ \in [-6, 6]$.

This illustrates that the scaled and unscaled certificates prove the same physical transition. The scaling only changes the numerical coordinates used by the solver.

Proposition A.2 (Invariance of the certificate under scaling). *Assume that, in scaled coordinates, the feedback law $\hat{\kappa}$ satisfies $\hat{f}(\hat{\mathcal{E}}, \hat{\kappa}(\hat{\mathcal{E}}), \mathcal{W}) \subseteq \hat{\mathcal{E}}^+$. Define the corresponding physical sets and feedback law by*

$$\mathcal{E} := c + D_x \hat{\mathcal{E}}, \quad \mathcal{E}^+ := c^+ + D_x^+ \hat{\mathcal{E}}^+,$$

and

$$\kappa(x) := \bar{u} + D_u \hat{\kappa}(D_x^{-1}(x - c)).$$

Then

$$f(\mathcal{E}, \kappa(\mathcal{E}), \mathcal{W}) \subseteq \mathcal{E}^+.$$

Proof. Let $x \in \mathcal{E}$. Then $\hat{x} := D_x^{-1}(x - c)$ belongs to $\hat{\mathcal{E}}$, and the definition of κ gives $\kappa(x) = \bar{u} + D_u \hat{\kappa}(\hat{x})$. For any $w \in \mathcal{W}$, the definition of $\hat{f}$ implies

$$f(x, \kappa(x), w) = c^+ + D_x^+ \hat{f}(\hat{x}, \hat{\kappa}(\hat{x}), w).$$

Since the scaled certificate gives $\hat{f}(\hat{x}, \hat{\kappa}(\hat{x}), w) \in \hat{\mathcal{E}}^+$, we obtain

$$f(x, \kappa(x), w) \in c^+ + D_x^+ \hat{\mathcal{E}}^+ = \mathcal{E}^+.$$

The inclusion follows for all $x \in \mathcal{E}$ and $w \in \mathcal{W}$. □

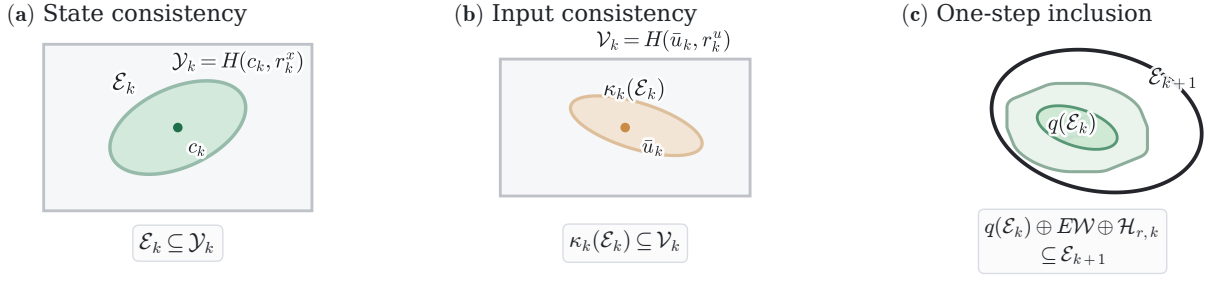


Figure A.2: Adaptive local certification boxes. With the notation of (3.28), the certificate enforces $\mathcal{E}_k \subseteq \mathcal{Y}_k$, $\kappa_k(\mathcal{E}_k) \subseteq \mathcal{V}_k$, and the one-step inclusion into $\mathcal{E}_{k+1}$.

Remark A.3 (Local boxes). The scaling does not remove the need for sound local certification boxes. The remainder bound must be computed on a region containing the physical source ellipsoid and the physical feedback image. In the implementation, these boxes are chosen adaptively and the soundness conditions are checked before accepting the transition. Thus, scaling improves numerical conditioning, while soundness is still obtained from the certificate audited in the original coordinates.

A.7 Optional numerical objective

The formal certificate only requires feasibility of the LMIs above. In practice, one may add an objective to select one feasible certificate among many. Suppose that the stage cost is quadratic:

$$J(x, u) = \begin{pmatrix} x \\ u \\ 1 \end{pmatrix}^\top Q_J \begin{pmatrix} x \\ u \\ 1 \end{pmatrix}, \quad Q_J = S^\top S. \quad (\text{A.42})$$

Introduce a scalar $\tilde{J}$ satisfying

$$\tilde{J} \geq \max_{x \in \xi} J(x, \kappa(x)). \quad (\text{A.43})$$

Since

$$\begin{pmatrix} x \\ \kappa(x) \\ 1 \end{pmatrix} = \begin{pmatrix} c \\ \ell \\ 1 \end{pmatrix} + \begin{pmatrix} L \\ F \\ 0 \end{pmatrix} z, \quad \|z\| \leq 1, \quad (\text{A.44})$$

the bound is enforced by the LMI

$$\begin{pmatrix} \gamma I & 0 & \begin{pmatrix} L \\ F \\ 0 \end{pmatrix}^\top S^\top \\ 0 & \tilde{J} - \gamma & \begin{pmatrix} c \\ \ell \\ 1 \end{pmatrix}^\top S^\top \\ S \begin{pmatrix} L \\ F \\ 0 \end{pmatrix} & S \begin{pmatrix} c \\ \ell \\ 1 \end{pmatrix} & I \end{pmatrix} \geq 0, \quad \gamma \geq 0. \quad (\text{A.45})$$

A typical local objective is then

$$\min \lambda \tilde{J} - (1 - \lambda) \log \det L, \quad \lambda \in [0, 1]. \quad (\text{A.46})$$

The first term penalizes the worst-case transition cost, while the second term maximizes the volume of the predecessor ellipsoid. This objective is only a numerical selection criterion; it is not part of the soundness argument.

Software implementation details

This appendix summarizes the software implementation used for the numerical experiments of this thesis. All experiments were implemented in `Dionysos.jl` [9], a Julia package for abstraction-based control and symbolic controller synthesis. The package provides the system, set, abstraction, and optimization tools on which the trajectory-guided certification pipeline was built.

Implementation in `Dionysos.jl`

The main parts of the implementation are located in the optimization layer of `Dionysos.jl`, with additional support from the system and utility layers. Table B.1 summarizes the components that were used or extended.

The planner and certifier communicate through a simple candidate-trajectory object, storing the nominal state sequence, the nominal input sequence, the sampling time, and metadata describing how the trajectory was generated. This interface is deliberately minimal: the certifier does not depend on whether the trajectory was obtained by MPPI, by a centered abstraction, or by another planner. This design makes it possible to replace the trajectory generator without changing the certification module.

The backward certifier was implemented as a separate component. Given a candidate trajectory, it starts from a terminal ellipsoid and proceeds backward along the trajectory. At each step, it builds a local affine approximation of the sampled dynamics, bounds the nonlinear remainder on a local box, and solves a semidefinite program to compute a predecessor ellipsoid and a local affine feedback law. If all local steps are feasible, the certifier returns a complete certified chain; otherwise, it returns the index at which certification failed.

Some implementation details are specific to the experiments. For angular systems, periodic coordinates may be unwrapped before certification to avoid artificial discontinuities at the $\pi/-\pi$ cut. For some benchmarks, local linearization boxes are adapted during certification. State scaling is also used in several experiments to improve the numerical conditioning of the semidefinite programs.

Component	Role
<code>Dionysos.System</code>	Defines the continuous and discrete dynamical systems used in the benchmarks, including RK4 discretizations and symbolic approximations.
<code>Dionysos.Utills</code>	Provides the set representations used by the pipeline, such as boxes, set differences, and ellipsoids, together with utilities for local ellipsoidal transitions.
<code>Optim.HeuristicGenerator</code>	Contains the trajectory-generation layer. This includes the centered abstraction warm start and the MPPI generator used to obtain nominal trajectories.
<code>Optim.SymbolicCertifier</code>	Contains the backward ellipsoidal certifier. This is the module that attempts to turn a candidate trajectory into a certified sequence of ellipsoids and feedback laws.
<code>Optim.CertifiedPipelineSolver</code>	Provides the wrapper connecting the trajectory generator and the certifier. It stores the candidate trajectory, the certification result, and the failure status if certification does not succeed.
<code>problems/</code>	Contains benchmark-specific system definitions, including the pendulum, double-pendulum, and articulated-vehicle models.
<code>utils/...</code>	Contains the scripts used to run the experiments, generate plots, save timing data, and perform sampled closed-loop rollout checks.

Table B.1: Main software components used in the thesis implementation.

Code status and numerical solver. At the time of writing, the code developed for this thesis is maintained on the experimental `memoire-updated` branch and has not yet been merged into the main branch of `Dionysos.jl`. The benchmark scripts are collected under `utils/benchmark_master_thesis_florentin/`. The local SDP/LMI problems used by the ellipsoidal backward certifier are formulated with `JuMP/MathOptInterface` and solved in the reported benchmarks with MOSEK.

UNIVERSITÉ CATHOLIQUE DE LOUVAIN
École polytechnique de Louvain

Rue Archimède, 1 bte L6.11.01, 1348 Louvain-la-Neuve, Belgique | www.uclouvain.be/epl