Received: 28 March 2022 | Revised: 8 April 2023 | Accepted: 18 May 2023

DOI: 10.1002/rnc.6808

RESEARCH ARTICLE

WILEY

Safe hierarchical model predictive control and planning for autonomous systems

Markus Kögel[1] | Mohamed Ibrahim[1] | Christian Kallies[2] | Rolf Findeisen[3]

[1]Laboratory for Systems Theory and Automatic Control, Otto von Guericke University, Magdeburg, Germany

[2]Institute of Flight Guidance, German Aerospace Center (DLR), Braunschweig, Germany

[3]Control and Cyber-Physical Systems Laboratory, Technical University of Darmstadt, Darmstadt, Germany

Correspondence
Rolf Findeisen, TU Darmstadt, Germany
Email: rolf.findeisen@tu-darmstadt.de

Abstract

Planning and control for autonomous vehicles usually are hierarchically separated. However, increasing performance demands and operating in highly dynamic environments requires a frequent re-evaluation of the planning and tight integration of control and planning to guarantee safety, performance, and reliability. We propose an integrated hierarchical predictive control and planning approach to tackle this challenge. The planner and controller are based on repeated solutions of moving horizon optimal control problems. To increase flexibility and feasibility, the planner can choose different low-layer controller modes for increased flexibility and performance instead of using a single controller with a large safety margin for collision avoidance under uncertainty. Planning is based on simplified system dynamics and safety, yet flexible operation is ensured by constraint tightening based on a mixed-integer linear programming formulation. A cyclic horizon tube-based model predictive controller guarantees constraint satisfaction for different control modes and disturbances. Examples of different modes are slow-speed movement with high precision and fast-speed movements with large uncertainty bounds. Allowing for different control modes reduces conservatism, while the hierarchical decomposition of the problem reduces the computational cost and enables real-time implementation. We derive conditions for recursive feasibility to ensure constraint satisfaction and obstacle avoidance to guarantee safety and compatibility between the layers and modes. Simulation results illustrate the efficiency and applicability of the proposed hierarchical strategy.

KEYWORDS

autonomous systems, hierarchies, model predictive control, obstacle avoidance, planning, safety

1 | INTRODUCTION

Autonomous vehicles, including drones, mobile robots, and autonomous transportation systems, are becoming more prevalent in a wide array of applications such as geo-surveillance, agricultural tasks, logistics, and search and rescue operations.[1,2] These vehicles are frequently tasked with navigating from a starting point to a destination while avoiding

obstacles in dynamically changing environments, as illustrated in Figure 1. Ensuring safe operation, such as collision avoidance, under all circumstances for autonomous vehicles in dynamic environments is a challenging task.[3-5] Commonly, this problem is addressed using a hierarchical approach, where planning is performed once or repeatedly on a slower time scale, providing a reference or path for a lower-layer control system that operates on a faster time scale. This control system counteracts disturbances, model uncertainties, and reacts to rapid environmental changes, as depicted in Figure 2A.[6-13]

Numerous planning strategies for autonomous vehicles have been proposed, as outlined in References 1,14-16 and their referenced works. These strategies are based on search methods or reformulations of the problem as a mathematical optimization problem. However, most of them do not explicitly account for detailed vehicle dynamics, environmental conditions, or disturbances.

To counteract significant uncertainties, such as changing environmental conditions or dynamic environments, frequent replanning and tight integration of control and planning are necessary. Recent research has explored various

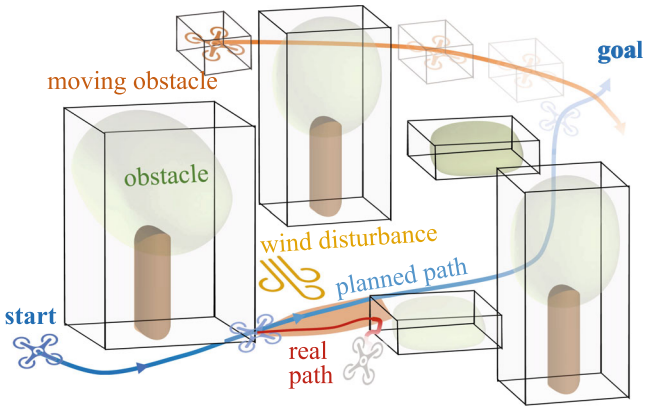
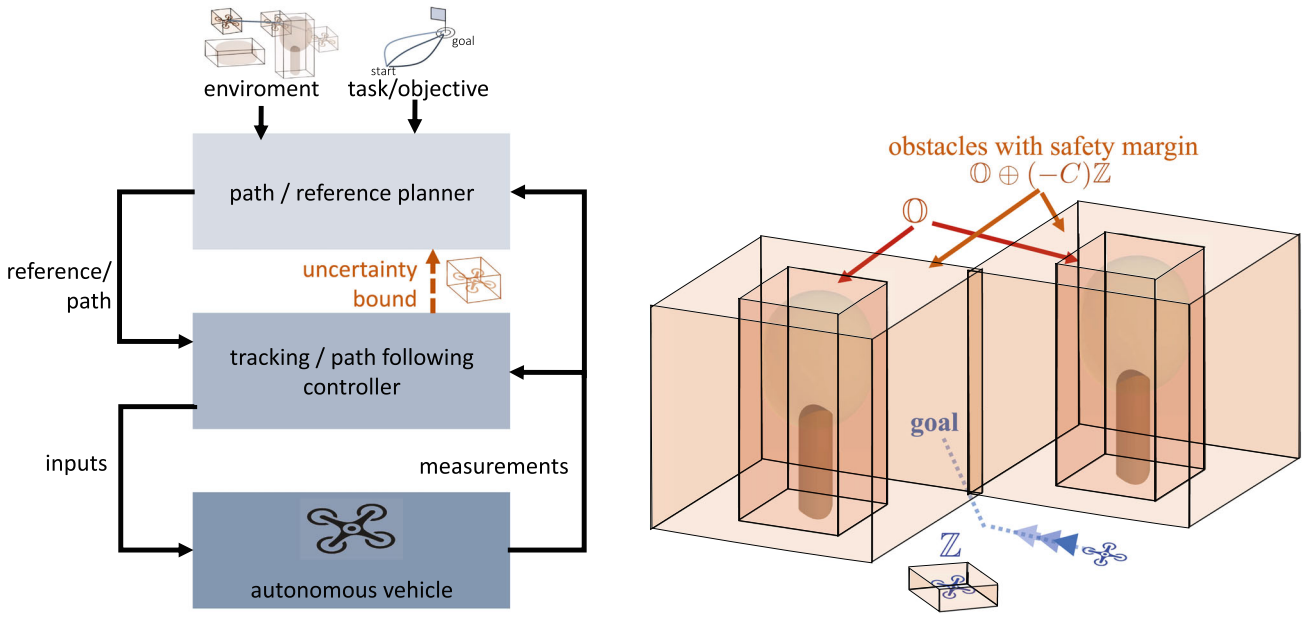


FIGURE 1 Planning and control for autonomous vehicles in environments with uncertainty and dynamic changes, such as moving obstacles or wind, present unique challenges.



(A) Typical decomposition of planning and control in a high-level planning and a low level tracking/path following task.

(B) Safe planning by constraint tightening/constraint back-off considering a single, fixed tracking control mode.

FIGURE 2 (A) Hierarchical separation of the planning and control layers; (B) safe planning with a single tracking control mode.

approaches to address these challenges. For example, a multi-rate hierarchical method[6] features three control layers operating at different sampling times and uses moving horizon formulation in the planning layer, while barrier functions in the tracking layer ensure obstacle avoidance and constraint satisfaction. Co-designing the planner and control[7] allows consideration of vehicle dynamics and kinematic constraints. A multi-layer framework[10] employs a low-fidelity optimization-based reference planner and a low-level controller that tracks the planned trajectory with an error bound. Other approaches include a two-layer moving horizon framework[12] and a stable hierarchical control scheme[13] that guarantees overall stability using an inner loop reference model and contracted constraint sets.

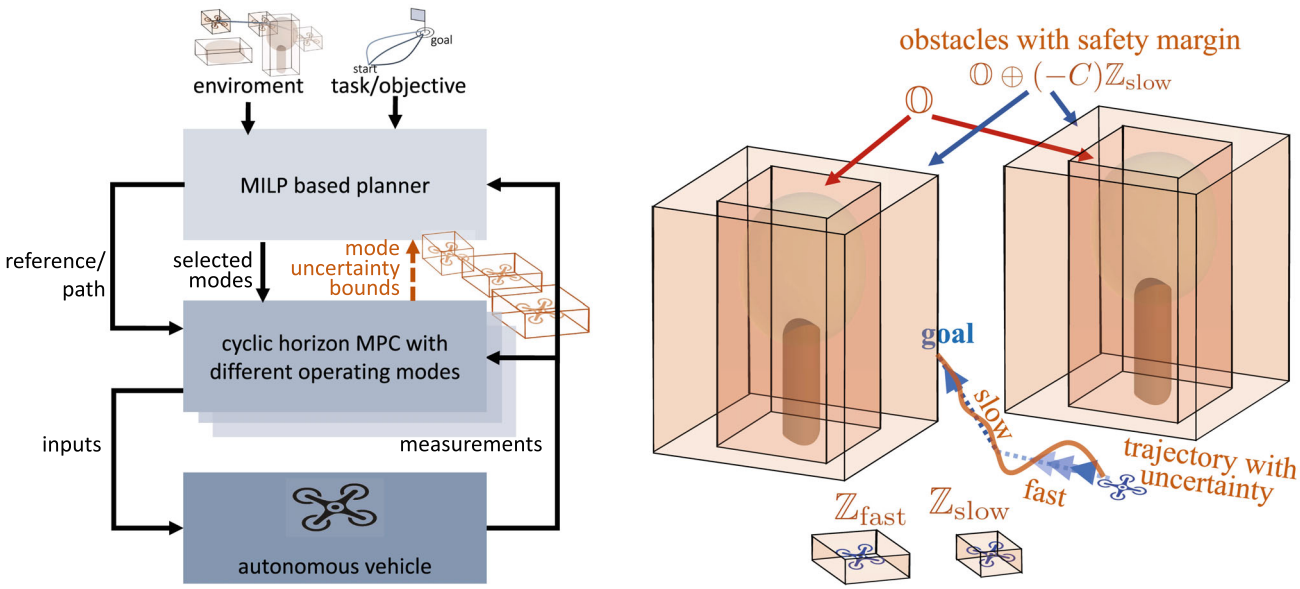
We propose an integrated hierarchical planning and control approach that reduces conservatism while maintaining computational feasibility. Both planner and controller rely on the repeated solution of moving horizon optimal control problems. Specifically, both layers utilize robust model predictive control (MPC)[17-21] formulations. The key concept is that the planning and lower-layer tracking controller agree on "contracts" (safety corridors), ensuring consistency and compatibility between layers. This agreement guarantees constraint satisfaction, such as collision avoidance, even in the presence of disturbances. To enhance flexibility and decrease conservatism, the planner can select different operating modes corresponding to various accuracies achievable by the lower-layer controller in closed-loop. In contrast, existing hierarchical formulations typically employ a single, conservative tracking controller mode, compare Figure 2.

One example of such modes includes slow-speed movement with high precision and fast-speed movement with large uncertainty bounds, compare Figure 3B. As illustrated in Figure 3B, the two control modes enable the planner to provide a collision-free path by switching online between different velocity ranges and corresponding controller parametrization.

We propose a moving horizon formulation based on a simplified system model for the planning layer, resulting in an efficiently solvable mixed-integer linear programming (MILP). The tracking control for the vehicle is achieved by a high sampling frequency operated cyclic horizon MPC.[20,22] Despite disturbances, the tracking controller provides safety bounds for each operation mode. We derive conditions for recursive feasibility to ensure constraint satisfaction and obstacle avoidance. Simulation results demonstrate the efficiency and applicability of the proposed hierarchical strategy. Different modes reduce conservatism, while the hierarchical decomposition of the control problem decreases computational burden and enables real-time implementation with provided guarantees.

The results expand on the work presented in Reference 23, addressing less conservative conditions and formulations. The overall approach is evaluated considering a quadcopter operating in a 3D environment.

The main contributions of this work are threefold:



(A) Proposed moving horizon planning and control approach allowing for multiple control modes.

(B) Reduced conservatism due to different tracking controller operation regions/modes.

FIGURE 3 Proposed hierarchical planning and control scheme with multiple tracking controller operation modes.

1. First, we present a planning approach operating on a moving horizon, which accommodates different tracking controller modes. We demonstrate how to formulate the resulting optimization problem as an efficiently solvable MILP.
2. Second, we develop a lower-layer cyclic horizon tube-based MPC controller, which provides safety bounds. In conjunction with the planning layer, the tube-based controller ensures constraint satisfaction and collision avoidance.
3. We furthermore present conditions that guarantee repeated feasibility and solvability of the presented approach under uncertainties.
4. Finally, we showcase our approach, considering a quadcopter operating in a 3D environment.

The remainder of the paper is organized as follows. Section 2 introduces the problem setup. Section 3 outlines the hierarchical planning and control scheme, where Section 3.1 details the MILP moving horizon planning problem. Section 3.2 elaborates on the lower-layer cyclic horizon tube MPC, and Section 3.3 discusses the switching between operating modes. Section 4 presents the simulation example and results for a quadcopter, while Section 5 summarizes our findings, conclusions, and future directions.

Notation For two given sets $\mathbb{X}, \mathbb{Y} \subset \mathbb{R}^n$, the Minkowski set sum $\oplus$ and the Minkowski set difference $\ominus$ are defined by: $\mathbb{X} \oplus \mathbb{Y} \triangleq \{x + y | x \in \mathbb{X}, y \in \mathbb{Y}\}$, $\mathbb{X} \ominus \mathbb{Y} \triangleq \{x | x \oplus \mathbb{Y} \subseteq \mathbb{X}\}$. We use *rem* to denote the remainder function of the Euclidean division, $\mathbf{1}$ denotes a column vector with 1 in each entry, and $\|x\|_Q^2 = x^T Q x$.

2 | PROBLEM FORMULATION

We consider the control of an autonomous vehicle, which should move (drive, fly, etc.) from a starting point, $x(0)$, to a goal, x_{goal}, while avoiding obstacles and satisfying constraints despite uncertainties and disturbances, see Figure 1. We assume that the vehicle dynamics are subject to unknown but bounded additive disturbances and that they are governed by

$$x(k+1) = Ax(k) + Bu(k) + w(k), \tag{1a}$$

$$y(k) = Cx(k), \tag{1b}$$

$$x(k) \in \mathbb{X}, \quad u(k) \in \mathbb{U}. \tag{1c}$$

Here $x(k) \in \mathbb{R}^n$ is the vehicle's state, $u(k) \in \mathbb{R}^m$ is the applied control input, and $y(k) \in \mathbb{R}^p$ is the output, while $w(k)$ is an unknown, but bounded disturbance. The state $x(k)$ and the input $u(k)$ need to satisfy constraints: they are restricted to the sets $\mathbb{X}$ and $\mathbb{U}$, which are both closed and convex. We assume that the goal state x_{goal} is a steady state of the dynamics (1a) with zero input under no disturbances, for simplicity, that is, $x_{\text{goal}} = Ax_{\text{goal}}$.

Obstacle avoidance: Beside the constraints (1c) on the state $x(k)$ and input $u(k)$ of the autonomous vehicle, we want to achieve obstacle avoidance, which we formulate in terms of the output $y(k)$ (e.g., the position). We assume that there are H obstacles and that each obstacle is modeled as a bounded set of the form $\mathbb{O}_\ell = \{y | E_\ell y < f_\ell\}$, where $E_\ell \in \mathbb{R}^{q_\ell \times p}$ and $f_\ell \in \mathbb{R}^{q_\ell \times 1}$. So, $\mathbb{O}_\ell$ is the interior of a convex, compact polytope. In the most simple case (box obstacles) $E_\ell = [I - I]$. Consequently, to avoid that the vehicle "collides" with the obstacles we require that:

$$y(k) \notin \mathbb{O}, \quad \mathbb{O} = \{E_\ell y(k) < f_\ell, \quad \ell = 1, \dots, H\}, \tag{2}$$

where $\mathbb{O}$ is the collection of all H obstacles. Clearly, the set of admissible output/position $y(k) \notin \mathbb{O}$, defined in (2), is nonconvex, but contains its boundaries. One can also formulate it as

$$\forall \ell \in \{1, \dots, H\}, \quad \exists a \in \{1, \dots, q_\ell\} : \; E_{\ell,a} y(k) \geq f_{\ell,a}, \tag{3}$$

where $E_{\ell,a}$ and $f_{\ell,a}$ denote the ath row of E_ℓ and f_ℓ, respectively. This formulation allows to use an MILP framework with the so-called big-M approach, which enables handling the obstacles systematically, see Section 3.3.

Disturbance bounds and operating regions/modes: The disturbance $w(k)$, and its bounds, may depend in parts also on the vehicle's state $x(k)$ and/or the applied control input $u(k)$. This might, for example, be due to model uncertainties, which

are captured as disturbances. For example, if the vehicle is operated at a lower speed, then the (worst case) disturbance might be smaller than at high speed.

To enhance the operational capabilities of the system, we aim to explore different uncertainty bounds for various operating modes, such as flying slowly or quickly. Specifically, we consider N_w distinct operating modes (operating regions), each with its own disturbance bound. In detail, we assume that:

Assumption 1 (operating modes dependent disturbance bounds). There are N_w operating modes defined by the state set $\mathbb{X}_i \subseteq \mathbb{X}$ and input sets $\mathbb{U}_i \subseteq \mathbb{U}$ and an uncertainty bound $\mathbb{W}_i$, where $i = 1, \dots, N_w$, such that

$$x(k) \in \mathbb{X}_i \text{ and } u(k) \in \mathbb{U}_i \Rightarrow w(k) \in \mathbb{W}_i.$$

We assume that the sets $\mathbb{X}_i$, $\mathbb{U}_i$ are convex and closed polytopes and $\mathbb{W}_i$ is a convex and compact polytope, which contains the origin.

For example if we consider a vehicle with two modes: a quick movement of the vehicle (mode 1) and a slower, but more accurate movement (mode 2), this results in $\mathbb{X}_1 \supset \mathbb{X}_2$ (larger admissible state space/faster movements possible), and $\mathbb{W}_1 \supset \mathbb{W}_2$ (larger uncertainty)[*].

3 | ROBUST HIERARCHICAL MOVING HORIZON PLANNING AND CONTROL

We focus on a hierarchical two-layer planning and control decomposition as depicted in Figure 3A, to achieve safe and collision-free motion of the autonomous vehicle to the goal. Both layers utilize (robust) MPC formulations for constraint satisfaction under uncertainties.

To provide guarantees despite the hierarchical decomposition of the problem in a planning and control layer, we utilize the concept of "contracts", inspired by References 24-26. Loosely speaking, a contract specifies the achievable precision in terms of a bound on the disturbance the lower-layer tracking controller can achieve for a specific operating region/mode i—a particular set of states and inputs. In the proposed approach, the moving horizon planner provides a reference path and selects the operating region the controller should operate in. To calculate a safe passage—satisfy the constraints (1c) and avoid all obstacles (3)—the planner takes the uncertainty bounds corresponding to the different operating modes directly into account.

In detail, the upper-layer planner calculates the reference based on a model with simplified dynamics of the form

$$x_p(k_p + 1) = A_p x_p(k_p) + B_p u_p(k_p), \tag{4}$$

where k_p is the planning time index. The generated reference path and measurements of the vehicle's state are used by the lower-layer controller to calculate the control input $u(k)$, see Figure 2. The lower-layer control loop aims to efficiently counteract the disturbances, to ensure that the vehicle follows the planned reference with a specified accuracy, to satisfy the contract, and to guarantee constraint satisfaction. As the obstacles are handled by the planner, the tracking controller does not need to consider them, which enables a fast and efficient implementation as non-convex constraints are avoided.

We assume that the planning operates on a slower time scale than the controls, that is, only every M (> 1) time steps. So we have

$$k = k_p \cdot M, \quad A_p = A^M, \quad B_p = \sum_{i=0}^{M-1} A^i B. \tag{5}$$

We assume that the real dynamics (1a) and the "planning dynamics" (4) satisfy:

Assumption 2 (Controllability of the planning and control dynamics). The pairs (A, B) and (A_p, B_p) are controllable.

The interaction between the planner and the lower-layer controller is based on the concept of contracts. Basically, if the planner determines the reference by taking certain additional restrictions into account, then the lower-layer controller can bound the tracking error, that is, the difference between the real state $x\left((k_p + 1) \cdot M\right)$ and the corresponding planning

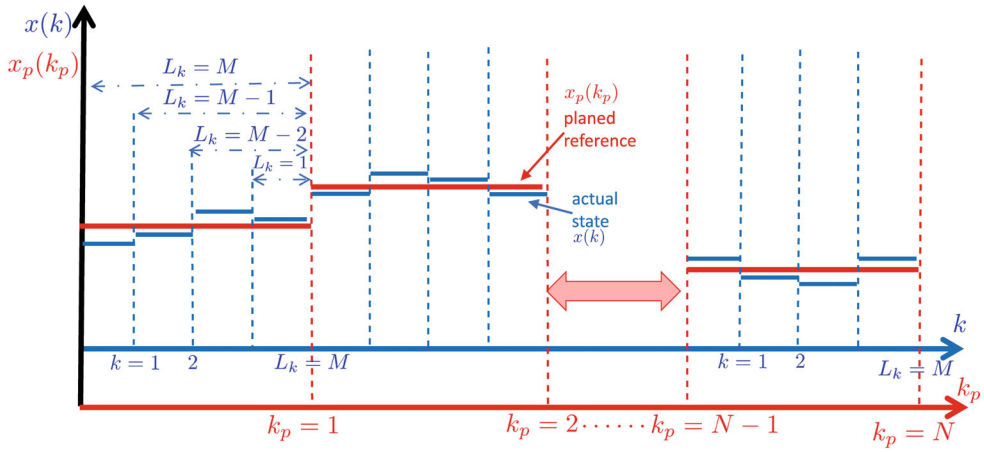


FIGURE 4 Different horizons for each planning and control problems. For planning problem: N is prediction horizon, k_p is time index, and x_p is planned state. For the low-level control problem: M is the maximum prediction horizon, while L_k denotes a shrinking horizon to guarantee repeated feasibility. Furthermore, k is the time index, and x is actual state.

state $x_p(k_p + 1)$ at the next time instant by

$$x\left((k_p + 1) \cdot M\right) - x_p(k_p + 1) \in \mathbb{Z}_i, \tag{6}$$

for $i = 1, \ldots, N_w$. Here the sets $\mathbb{Z}_i$ are convex, compact polytopes and N_w is the number of the operating modes. We refer for the exact definition of the sets $\mathbb{Z}_i$ to Section 3.2 and (15).

The lower-layer controller can guarantee the constraints, if $x \in \mathbb{X}_i$ and $u \in \mathbb{U}_i$, that is, that the state and the input are inside for the operation mode i, see Assumption 1. Additionally, constraints due to the obstacle avoidance requirements and the different sampling times need to be satisfied, which are introduced in a second step.

We assume that the contracts—the operating modes—are designed offline and known by both control layers. They depend on the design of the low-layer controller and the operation mode i, that is, the (partly) selectable uncertainty bound $\mathbb{W}_i$ on the disturbance w.

Summarizing: Utilizing the idea of contracts—different operating modes—enables the planner to utilize and take the capability of the low-layer control loop into account for the computation of the reference. Consequently, the planner calculates and sends to the low-layer controller a reference and selects via the choice of the operation mode i (and thus the set $\mathbb{Z}_i$) the maximum allowed tracking discrepancy. In other words, the reference planner can switch between different operation modes in order to improve the performance, as illustrated in Figure 3B.

Moreover, we use different prediction horizons and sampling times for each planning and control problem, see Figure 4. In detail, we use the planning horizon N, the time index k_p, a shrinking prediction horizon L_k, and the fast time index k for the lower-level control problem.

3.1 | Upper-layer: Moving horizon reference planning

The reference planner should guarantee constraint satisfaction, including obstacle avoidance, despite the presence of disturbances and uncertainties in combination with a suitably designed lower-layer control loop, compare to Figure 3A.

The key idea is to incorporate the concept of contracts—different operating modes—into mathematical programming based on moving horizon planning schemes.[9,27-30]

To do so, the reference planning problem $\mathcal{P}\left(x(k_pM)\right)$ is formulated on a moving horizon as an optimization problem:

$$\min_{\{x_p\},\{u_p\},i} J_p\left(\{x_p\},\{u_p\}\right), \tag{7a}$$

s.t.

$$i \in \{1, \ldots, N_w\} \tag{7b}$$

$$x(k_pM) - x_p(k_p|k_p) \in \mathbb{Z}_i, \tag{7c}$$

$$\forall j \in \{0, \dots, N-1\}: \quad x_p(k_p+j+1|k_p) = A_p x_p(k_p+j|k_p) + B_p u_p(k_p+j|k_p), \tag{7d}$$

$$\forall j \in \{0, \dots, N-1\}: \quad x_p(k_p+j|k_p) \in \mathbb{X}_i \ominus \mathbb{Z}_i, \tag{7e}$$

$$\forall j \in \{0, \dots, N-1\}: \quad u_p(k_p+j|k_p) \in \mathbb{U}_i \ominus K\mathbb{Z}_i, \tag{7f}$$

$$\forall j \in \{0, \dots, N-1\}: \quad Cx_p(k_p+j|k_p) \notin \mathbb{O} \oplus (-C)\mathbb{Z}_i, \tag{7g}$$

$$\forall j \in \{0, \dots, N-1\}: \quad \big(x_p(k_p+j|k_p), u_p(k_p+j|k_p)\big) \in \mathbb{I}_i, \tag{7h}$$

$$x_p(k_p+N|k_p) \in \mathbb{X}_i^f. \tag{7i}$$

Here N denotes the planning horizon, i the operation mode, and $(k_p+j|k_p)$ corresponds to the prediction of a value at time k_p+j made at time k_p. The sets $\mathbb{I}_i$ and $\mathbb{X}_i^f$ are introduced to impose constraints on the inter-sample behavior and the terminal state and are discussed below. The constraint (7c) represent an initial constraint at the beginning of the planning time k_p. Equation (7d) represents the vehicle dynamics used by the planning layer. While the constraints (7e,7f) are the tightened state and input constraints, where K is a control gain, which is discussed later in detail. For the output constraints (7g) the constraint tightening corresponds to an enlargement of the obstacles, see Figure 3B.

To allow for an efficient reformulation of (7) as an MILP, compare Section 3.3, we consider the following cost function for the planning problem, assuming that the goal state is fixed and a steady state.

$$J_{\mathrm{p}}\left(\{x_p\},\{u_p\}\right) = \|x_{\mathrm{goal}} - x_p(k_p+N)\|_\infty + \sum_{j=k_p}^{k_p+N-1} \alpha_x \|x_{\mathrm{goal}} - x_p(j)\|_\infty + \alpha_u \|u_p(j)\|_\infty.$$

The stage cost penalizing the state x_p and control input u_p with different weights ($\alpha_x \geq 0, \alpha_u \geq 0$), respectively. The terminal cost penalizes the vehicle's distance at the end of the planning horizon to the goal x_{goal}.

The inter-sample constraints (7h) and the terminal constraint (7i) depend on the operation mode i and are non-convex. We make the following assumption with respect to the inter-sample constraints (7h).

Assumption 3 (inter-sample constraints). The inter-sample constraints (7h) determined by the sets $\mathbb{I}_i$ are such that, if $(x_p, u_p) \in \mathbb{I}_i$, then for $\ell = 1, \dots, M-1$ it holds

$$A^\ell x_p + \sum_{m=0}^{\ell-1} A^m B u_p \in \mathbb{X}_i \ominus \mathbb{Z}_i, \tag{8a}$$

$$C\left(A^\ell x_p + \sum_{m=0}^{\ell-1} A^m B u_p\right) \notin \mathbb{O} \oplus (-C)\mathbb{Z}_i. \tag{8b}$$

This assumption guarantees that the lower-layer/tracking control loop, operating at the faster time scale, can always satisfy the constraints. Note that the constraints (7e) and (7g) on the state x_p and the output Cx_p of the planning dynamics (4) alone usually do not guarantee that the constraints between two consecutive planning time indices are satisfied as illustrated in Figure 5.

A straightforward choice is to choose $\mathbb{I}_i$ directly as (8) with $\ell = 1, \dots, M-1$, which can lead to a large number of constraints and thus might increase the computational effort. Note that depending on the actual dynamics (4) certain constraints in (7h) might be redundant and thus also the overall optimization (7) and can be removed without changing the solution of the optimization problem, for example, using physical insight into the system dynamics or with the procedure presented in Reference 31.

For the terminal sets $\mathbb{X}_i^f$ we assume that they are positive invariant sets of (4) satisfying all constraints:

Assumption 4 (terminal sets and terminal control laws). The terminal control laws $\kappa_i^f(x_p)$ and the terminal sets $\mathbb{X}_i^f$ are such that $x_p \in \mathbb{X}_i^f$ implies:

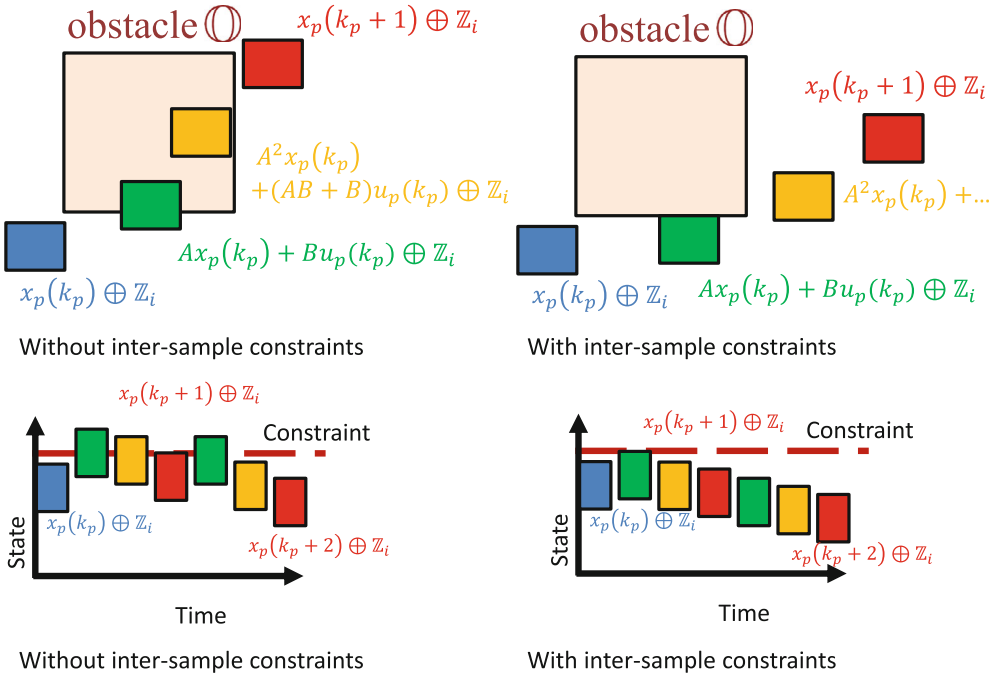


FIGURE 5 Illustration of inter-sample behavior. Planning without inter-sample constraints (left) in planning problem (7) leads to possible collision with obstacle (top, left) and/or constraint violation (bottom, left). With inter-sample constraints obstacle is successfully passed (top, right) and constraints are satisfied (bottom, right).

$$A_p x_p + B_p \kappa_i^f(x_p) \in \mathbb{X}_i^f, \tag{9a}$$

$$x_p \in \mathbb{X}_i \ominus \mathbb{Z}_i, \tag{9b}$$

$$\kappa_i^f(x_p) \in \mathbb{U}_i \ominus K\mathbb{Z}_i, \tag{9c}$$

$$\left(x_p, \kappa_i^f(x_p)\right) \in \mathbb{I}_i, \tag{9d}$$

$$Cx_p \notin \mathbb{O} \oplus (-C)\mathbb{Z}_i. \tag{9e}$$

Clearly, the terminal sets $\mathbb{X}_i^f$ are non-convex due to the presence of the inter-sample constraints (9d) and the obstacle avoidance condition (9e). A possible choice are admissible, nominal steady states $x_p = A_p x_p + B_p \kappa_i^f(x_p)$ for the terminal control sets and the corresponding inputs as terminal control laws. For autonomous vehicles, these are basically all points where the vehicle can stop its motion safely. These points can also be determined for systems with complex dynamics, such as unmanned aerial vehicles. Note that the terminal control laws $\kappa_i^f(x_p)$ are fictitious and never implemented. In Appendix B we discuss how such a set can be calculated.

The upper-layer planning algorithm solves the optimization problem $\mathcal{P}\left(x(k_p M)\right)$ (7). Based on the optimal solution it sends to the lower-layer controller the chosen operation mode $i^\star$ and an inter-sampled reference

$$x_{\text{ref}}(k_p M) = x_p^\star(k_p|k_p), \tag{10a}$$

$$x_{\text{ref}}(k_p M + j) = A^j x_p^\star(k_p|k_p) + \sum_{m=0}^{j-1} A^m B u_p^\star(k_p|k_p), \quad j = 1, \ldots, M. \tag{10b}$$

Clearly, $\left(x_p^\star(k_p|k_p), u_p^\star(k_p|k_p)\right) \in \mathbb{I}_i$ together with the obstacle avoidance constraint (7g) implies that the reference satisfies

$$Cx_{\text{ref}}(k_pM+j) \notin \mathbb{O} \oplus (-C)\mathbb{Z}_i, \quad j = 0, \dots, M, \tag{11}$$

which means that the planned reference x_{ref} robustly avoids obstacles, that is, it satisfies the condition (2).

Using the idea of contracts between both layers, we can guarantee the following.

Proposition 1. *(Recursive feasibility of the upper layer planning) Let Assumptions 1-4 hold and assume that the lower-layer controller guarantees the contract (6) for the reference (10). If the planning optimization problem $\mathcal{P}\left(x(k_pM)\right)$ (7) is feasible, then the planning problem $\mathcal{P}\left(x((k_p+1)M)\right)$ is also feasible at the next replanning instant.*

Proof. The key idea for the proof is that the upper-layer planning MPC (in combination with the contracts) corresponds to a tube-based MPC using robust invariant sets, compare References 17, 19, 21. We denote the optimal solution of $\mathcal{P}\left(x(k_pM)\right)$ by $x_p^\star(k_p|k_p), \dots, u_p^\star(k_p|k_p), \dots, i^\star$. Let us consider the following guess as solution for the optimization problem $\mathcal{P}\left(x((k_p+1)M)\right)$

$$\begin{aligned}
i &= i^\star, \\
\forall j \in \{1, \dots, N\} : x_p(k_p+j|k_p+1) &= x_p^\star(k_p+j|k_p), \\
x_p(k_p+N+1|k_p+1) &= A_p x_p^\star(k_p+N|k_p) + B_p \kappa_i^f\left(x_p^\star(k_p+N|k_p)\right), \\
\forall m \in \{1, \dots, N-1\} : u_p(k_p+m|k_p+1) &= u_p^\star(k_p+m|k_p), \\
u_p(k_p+N|k_p+1) &= \kappa_i^f\left(x_p^\star(k_p+N|k_p)\right).
\end{aligned}$$

Note that this guess is based on the previous solution, the selected operation mode $i^\star$ and the terminal control law κ_i^f. We need to verify that this guess is feasible (but it might be possibly suboptimal) for the optimization problem $\mathcal{P}\left(x((k_p+1)M)\right)$. Using the contract (6), that is, the guarantee on the lower-layer control loop, we have that $x\left((k_p+1)M\right) - x_p(k_p+1|k_p+1) \in \mathbb{Z}_i$, that is, (7c) holds for $\mathcal{P}\left(x((k_p+1)M)\right)$. The above initial guess satisfies the constraints (7d), (7e)-(7h) for $1 \le j < N-1$ for the optimization problem $\mathcal{P}\left(x(k_pM)\right)$, thus also the similar constraints for $\mathcal{P}\left(x((k_p+1)M)\right)$. Finally, using the conditions on the terminal sets and terminal control laws in Assumption 4 imply that also the remaining constraints of $\mathcal{P}\left(x((k_p+1)M)\right)$, that is, (7d)-(7h) with $j = N-1$ and (7i) are satisfied. ∎

Remark 1 (Planning without feedback). In (7), the initial constraint (7c) provides a feedback between the planning state x_p and the real state x for all k_p. One can only enforce the constraint (7c) at the initial time ($k_p = 0$) and replace it by the simpler equality constraint $x_p^\star(k_p+1|k_p) = x_p(k_p+1|k_p+1)$ for $k_p > 0$. This would remove the feedback from the plant to the upper-layer planner. This has the advantage that it avoids the need to wait for plant feedback for the planning and thus could enable computationally less restrictive planning, but it would also decrease the control performance.

The optimization problem $\mathcal{P}$ (7) is non-convex, but it can be reformulated as an MILP, which can be efficiently solved, see Section 3.3.

3.2 | Lower-layer: Cyclic horizon robust model predictive tracking control

The lower-layer controller tracks the generated reference based on the (faster) dynamics of the real system and needs to guarantee the contracts (6). Also at the lower-layer we use the concept of robust tube-based MPC,[17,19-21] but we rely on growing tubes[20] instead of tubes based on robust positive invariance[21] as in the upper-layer.

The proposed tube-based MPC of the lower-layer is based on a nominal prediction dynamics (nominal state z, nominal input v), which starts from the real state $x(k)$ at the current time k:

$$z(k+j+1|k) = Az(k+j|k) + Bv(k+j|k), \tag{12a}$$

$$z(k|k) = x(k). \tag{12b}$$

The effect of disturbances $w(k+j)$ onto the closed loop is taken into account using a fictitious, auxiliary control law of the form

$$u(k+j|k) = v(k+j|k) + K\left(x(k+j) - z(k+j|k)\right). \tag{13}$$

The control gain K in this affine feedback is chosen such that $A+BK$ is Schur stable. The auxiliary control law is utilized to determine sets to bound the difference $e(k+j|k) = x(k+j) - z(k+j|k)$ between the real system state x and the predictions z made using (12). In detail, for the ith operation mode the error bounds satisfy $e(k+j|k) \in \mathbb{E}_i(j)$ where

$$\mathbb{E}_i(j+1) = (A+BK)\mathbb{E}_i(j) \oplus \mathbb{W}_i, \quad \mathbb{E}_i(0) = \{0\}. \tag{14}$$

Note that the size of the sets $\mathbb{E}_i$ monotonically increases with j, that is, $\mathbb{E}_i(j) \subseteq \mathbb{E}_i(j+1)$. However, for any $j \geq 0$ we have that $\mathbb{E}_i(j) \subseteq \mathbb{Z}_i$, where $\mathbb{Z}_i$ is the (minimum) robust positive invariant set, compare Reference 19, satisfying:

$$\mathbb{Z}_i \supseteq (A+BK)\mathbb{Z}_i \oplus \mathbb{W}_i. \tag{15}$$

In the lower-layer MPC, we predict until the next planning instant utilizing a cyclic horizon $L_k = M - rem(k, M)$, see Reference 22. For the case that k is a multiple of M, we have $L_k = M$. Otherwise, L_k is smaller than M, but $k + L_k$ is a multiple of M. Consequently, the horizon shrinks between two planning instants and is increased at the next planning instant again to length M.

The lower-layer MPC predicts and optimizes nominal state and input sequences

$$\mathbf{z}(k) = \{z(k|k), \ldots, z(k+L_k|k)\}, \quad \mathbf{v}(k) = \{v(k|k), \ldots, v(k+L_k-1|k)\}, \tag{16}$$

based on the nominal dynamics (12) and subject to satisfaction of the constraints

$$\forall j \in \{0, \ldots, L_k - 1\}: \quad z(k+j|k) \in \mathbb{X}_i \ominus \mathbb{E}_i(j), \tag{17a}$$

$$\forall j \in \{0, \ldots, L_k - 1\}: \quad v(k+j|k) \in \mathbb{U}_i \ominus K\mathbb{E}_i(j), \tag{17b}$$

$$\forall j \in \{0, \ldots, L_k - 1\}: \quad C\left(z(k+j|k) - x_{\text{ref}}(k+j)\right) \in C\left(\mathbb{Z}_i \ominus \mathbb{E}_i(j)\right), \tag{17c}$$

$$z(k+L_k|k) - x_{\text{ref}}(k+L_k) \in \mathbb{Z}_i \ominus \mathbb{E}_i(L_k). \tag{17d}$$

Note that these constraints include the convex state and input constraints (1c). In contrast, the non-convex obstacle avoidance constraints are taken into account using the concept of contracts. Basically, the lower-layer controller needs to enforce the guaranteed accuracy with respect to the output (condition (17c)) or even the full state at the end of the prediction (condition (17d)). Note that, the constraints (12), (17) are convex.

The lower-layer MPC penalizes the deviation error from the reference x_{ref} and utilizes the convex cost function

$$J_{\text{t}}\left(\mathbf{z}(k), \mathbf{v}(k)\right) = \sum_{j=k}^{k+L_k-1} \|x_{\text{ref}}(j) - z(j|k)\|_Q^2 + \|v(j|k)\|_R^2 + \|x_{\text{ref}}(k+L_k) - z(k+L_k|k)\|_P^2, \tag{18}$$

where the matrices $Q \in \mathbb{R}^{n\times n}$, $P \in \mathbb{R}^{n\times n}$, and $R \in \mathbb{R}^{m\times m}$ are positive definite and represent the weighting for the inter-sample states, the final state and the inputs, respectively. Note that the objective of the lower-layer controller is to guarantee the satisfaction of the constraints (17), so the weightings Q, R, and P do not need to satisfy additional conditions.

The applied control input $u(k) = v^\star(k|k)$ is given by solving the optimization problem $\mathcal{L}\left(x(k), \{x_{\text{ref}}\}, i, k\right)$

$$\min_{\mathbf{z}(k),\mathbf{v}(k)} J_{\text{t}}\left(\mathbf{z}(k), \mathbf{v}(k)\right) \quad \text{s.t. (12), (17).} \tag{19}$$

This optimization problem depends on the current state available at the lower-layer as well as the reference and the operation mode determined by the upper-layer planner. The resulting optimization problem is a convex quadratic program (QP)

and has, in addition, a special structure, which allows its efficient solution, even on computationally limited hardware, see for example, References 32 and 33.

For the lower-layer, we can derive the following properties assuming that the upper-layer reference is chosen suitably.

Proposition 2. *(Constraint satisfaction and obstacle avoidance) Let Assumptions 1 and 3 hold and assume that the reference x_{ref} is given by (10) and satisfies (11). If the lower-layer MPC problem $\mathcal{L}\left(x(k), \{x_{ref}\}, i, k\right)$ (19) is feasible, then the constraints (1c) are satisfied: $x(k) \in \mathbb{X}_i \subseteq \mathbb{X}$ and $u(k) \in \mathbb{U}_i \subseteq \mathbb{U}$ and the obstacles are avoided, that is, (3) holds.*

Proof. Combining (12), (13) and (17) we have that $z^\star(k|k) = x(k)$, $v^\star(k|k) = u(k)$ and $Cz^\star(k|k) = y(k)$. Together with $\mathbb{E}_i(0) = \{0\}$ and Assumption 1 this implies that the state and input constraints (1c) are satisfied. (17c) together with (14) yield that $y(k) - Cx_{\text{ref}}(k) \in C\mathbb{Z}_i$. Combined with (7g) (11) this implies that the avoidance constraint $y_k \notin \mathbb{O}$ (3) holds. ∎

Note that if the upper-layer planner provides the reference x_{ref} (10), then also (11) holds.

Proposition 3. *(Recursive feasibility of the overall control scheme) Let Assumptions 1-4 hold. If the planning problem (7) is feasible for $x(0)$, then the optimization problems (7), (19) remain feasible for the closed loop system consisting of the upper-layer moving horizon planner (7), (10), the lower-layer controller (19) and the uncertain plant dynamics (1).*

Proof. The proof consists of three parts: first it is shown that feasibility of the planning problem $\mathcal{P}\left(x(k_pM)\right)$ (7) implies feasibility of the low-layer optimization problem (19) $\mathcal{L}\left(x(k_pM), \{x_{\text{ref}}\}, i^\star, k_pM\right)$. Afterwards we verify that feasibility of $\mathcal{L}\left(x(k), \{x_{\text{ref}}\}, i, k\right)$ (19) implies: if $k+1$ is not a multiple of M, feasibility of $\mathcal{L}\left(x(k+1), \{x_{\text{ref}}\}, i, k+1\right)$ (19) (part 2) and otherwise feasibility of the planning problem $\mathcal{P}\left(x(k+1)\right)$ (7) (part 3).

(1) If $\mathcal{P}(x(k_pM)$ (7) is feasible, then the following (suboptimal) input trajectory and state sequence for the lower-layer problem $\mathcal{L}\left(x(k_pM), \{x_{\text{ref}}\}, i^\star, k_pM\right)$

$$\begin{aligned} z(k|k) =& x(k) \\ v(k+j|k) =& u_p^\star(k_p|k_p) + K(z(k+j|k) - x_{\text{ref}}(k+j|k)), \quad j = 0, \dots, M-1, \\ z(k+j+1|k) =& Az(k+j|k) + Bv(k+j|k), \quad j = 0, \dots, M-1, \end{aligned}$$

satisfies all constraints of (19) due to the inter-sample constraints $\mathbb{I}_i$ and the consistent constraint tightening utilized at both layers, that is, the definition of $\mathbb{E}_i$ and $\mathbb{Z}_i$, see (14) and (15) and that $\mathbb{E}_i \subseteq \mathbb{Z}_i$.

(2) In the case that $k+1$ is not a multiple of M, that is, no planning takes place, the horizon L_k shrinks. Due to the design of the set $\mathbb{E}_i(j)$ (14) a feasible nominal state trajectory $\mathbf{z}(k+1)$ and a nominal input trajectory $\mathbf{v}(k+1)$ for $\mathcal{L}\left(x(k+1), \{x_{\text{ref}}\}, i, k+1\right)$, satisfying (12) and (17), can be obtained from the solution of $\mathcal{L}\left(x(k), \{x_{\text{ref}}\}, i, k\right)$:

$$\begin{aligned} z(k+j|k+1) =& z^\star(k+j|k) + (A+BK)^{j-1}w(k), \\ v(k+j|k+1) =& v^\star(k+j|k) + K(A+BK)^{j-1}w(k). \end{aligned}$$

(3) Finally, if $k+1$ is a multiple of M, that is, the planning problem is solved at $k+1$, then feasibility of $\mathcal{L}\left(x(k), \{x_{\text{ref}}\}, i, k\right)$, in particular (17d) implies that $x(k+1) - x_{\text{ref}}(k+1) \in \mathbb{Z}_{i^\star}$. This together with Proposition 1 yields that the planning problem (7) is feasible at $k+1$. ∎

Remark 2 (adaption of sets). We assume that state and input constraints $\{\mathbb{X}_i, \mathbb{U}_i\}$ and the tubes/contracts $\{\mathbb{E}_i, \mathbb{Z}_i\}$ are determined offline. The proposed approach can in principle be extended to allow an adaption of these sets. This could be useful for example to consider the influence of varying weather conditions. We do not consider such an extension in this work.

Remark 3 (relaxing condition (17c)). In the optimization problem (19) the difference between the real/predicted output y_k/z_k and its reference $Cx_{\text{ref}}(k)$ is restricted to the set $C(\mathbb{Z}_i \ominus \mathbb{E}_i)$, compare to (17c). This

restriction is used to enable guarantees on the obstacle avoidance constraints (2). However, if the vehicle position/output (or /certain directions of it) is at time instant k far away from an obstacle, then this restriction can be conservative. In principle, it is possible to relax these constraints by generating online based on the solution of the upper-layer less conservative output constraints of the form: $Cz(k+j|k) \in \mathbb{Y}_{k+j} \ominus C\mathbb{E}_i$, which have to be satisfied.

Remark 4 (More general tube scheme). We use a basic tube scheme with a single gain K and focused on linear dynamics. One could use a more general scheme with multiple gains, see Reference 34, or a more complex tube control law, see for example, References 35 and 36. Also, an extension to nonlinear lower-layer dynamics is in principle possible using for example, References 37 and 38.

3.3 | MILP solution of the planning problem

In the following, we discuss how the non-convex optimization problem $\mathcal{P}$ (7) can be reformulated using the big-M method[9,28] into an MILP. Note that $\mathcal{P}$ (7) is non-convex due to two reasons: firstly, due to the operation mode i, and secondly the non-convex obstacles avoidance constraints (3) result in the non-convex constraints (7f)–(7i).

Scheduling of operation modes: In the proposed hierarchical scheme the operating modes of the vehicle given in the form of different constraint sets $\mathbb{X}_i$ and $\mathbb{U}_i$ directly influences the uncertainty bounds $w \in \mathbb{W}_i$, see Assumption 1. The lower-layer controller guarantees constraint satisfaction and guarantees bounds on the tracking error in form of a set $\mathbb{Z}_i$, which depends on the operation mode. So, the sets appearing in the initial constraint (7c) and the tightened state/input constraints (7e), (7f) are of the form

$$\begin{aligned}
\mathbb{Z}_i &\equiv \{x | F_i^z x \leq G_i^z\}, \quad \forall i \in \{1, \dots, N_w\},\\
\mathbb{X}_i \ominus \mathbb{Z}_i &\equiv \{x | F_i^x x \leq G_i^x\}, \quad \forall i \in \{1, \dots, N_w\},\\
\mathbb{U}_i \ominus K\mathbb{Z}_i &\equiv \{u | F_i^u u \leq G_i^u\}, \quad \forall i \in \{1, \dots, N_w\}.
\end{aligned}$$

As result, this contract provides the planner an extra degree of freedom to reduce the planning conservatism by switching between different operating modes of the lower-layer controller.

We use the so called big-M method to formulate the mode scheduling as:

$$\begin{aligned}
F_i^z(x - x_p) &\leq G_i^z + M_{\text{big}}(1 - d_i)\mathbf{1}, \quad \forall i \in \{1, \dots, N_w\},\\
F_i^x x &\leq G_i^x + M_{\text{big}}(1 - d_i)\mathbf{1}, \quad \forall i \in \{1, \dots, N_w\},\\
F_i^u u &\leq G_i^u + M_{\text{big}}(1 - d_i)\mathbf{1}, \quad \forall i \in \{1, \dots, N_w\},\\
\sum_{i=1}^{N_w} d_i &= 1.
\end{aligned}$$

Here we use a large positive number M_{big} to deactivate the constraints of the ith mode by relaxing its constraints using the binary decision variable d_i. The last constraint guarantees that exactly one mode is active in the planning.

Obstacle avoidance constraints: For each mode, the tracking error set $\mathbb{Z}_i$ is used to enlarge boundaries of the obstacles $\mathbb{O}$, compare (7). The enlarged, non-convex avoidance constraints (3) can be rewritten/over-approximated by

$$\forall j = 0, \dots, N-1, \ \forall, \ell = 1 \dots, H : \exists a \in 1, \dots, q_j \ \text{ s.t. } \ E_{\ell,a} C x_p(k_p + j|k_p) \geq f_{\ell,a}^i.$$

In this case one can enforce this constraint for the active mode i by using additional binary variables $b_{\ell,a}^i(j)$ by requiring for $j = 0, \dots, N-1$ and $\ell = 1 \dots, H$ that

$$E_{\ell,a} C x_p(k_p + j|k_p) \geq f_{\ell,a}^i - M_{\text{big}} b_{\ell,a}^i(j), \qquad \sum_{a=1}^{q_j} b_{\ell,a}^i(j) \leq q^\ell - d_i. \tag{20}$$

Here we impose an extra constraint to ensure that at least one active constraint for mode i, where q^ℓ is number of faces of each obstacle. Loosely speaking, the planner can choose different low-layer controller modes i to adjust the obstacle

boundary $f^i_{\ell,a}$ by changing the operation modes. This is illustrated in Figure 8B considering that the operation modes corresponds to different velocities.

In a similar fashion, this can be done for the obstacle avoidance conditions arising in the terminal sets. Also the inter-sample constraint (8) feature obstacle avoidance conditions. In principle they can also be handled using binary variables, which increases the computational complexity.

Another way to handle the obstacle avoidance conditions in the inter-sample constraints (8b) is to use the same binary variables as in (20) also for the outputs at the inter-sample instants

$$E_{\ell,a}C(A^\lambda x_p(k_p+j|k_p)+\sum_{m=0}^{\lambda-1}A^m Bu_p(k_p+j|k_p)) \geq f^i_{\ell,a} - M_{\text{big}}b^i_{\ell,a}(j),\ \ \lambda = 1,\ \dots\ ,M-1.$$

This means in a nutshell that the planning output $Cx_p(k_p+j|k_p)$ and the inter-sample outputs have to lay on the same side of the enlarged obstacle.

The overall control algorithm is illustrated in Algorithm 1. Observe that, in the algorithm, solving the planning problem at planning time instants and determining the lower-level control (19) can be computationally challenging. However, this issue can be addressed using delay compensation techniques, as discussed in Reference 39.

Algorithm 1. Overall control algorithm

for $k = 0, 1, \dots$ **do**
  Measure state x_k
  if k is a multiple of M **then**
    Solve planning problem $\mathcal{P}\big(x(k)\big)$ (7) and calculate reference x_{ref} (10)
    Send reference $\{x_{\text{ref}}\}$ and selected mode $i^\star$ to lower level controller
  end if
  Determine optimal input sequence $\boldsymbol{v(k)}^\star$ from lower level problem $\mathcal{L}\big(x(k), \{x_{\text{ref}}\}, i^\star, k\big)$ (19)
  Apply input $u(k) = v(k|k)^\star$

end for

4 | UNMANNED AERIAL VEHICLE EXAMPLE

We consider a quadcopter that should fly from a starting point to a goal point without hitting obstacles, compare Figure 6.

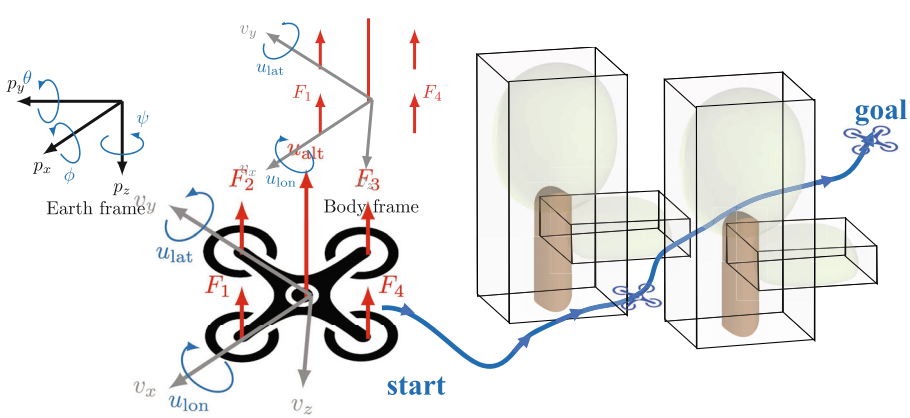


FIGURE 6 Illustration of the quadcopter state, forces, and moments described in the earth (p_x, p_y, p_z, θ, φ, ϕ) and body frame. The considered inputs are the lateral and longitudinal moments are provided by the four rotors resulting in the forces F_i.

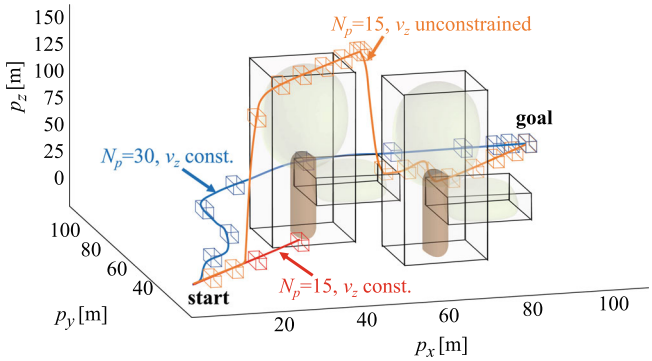
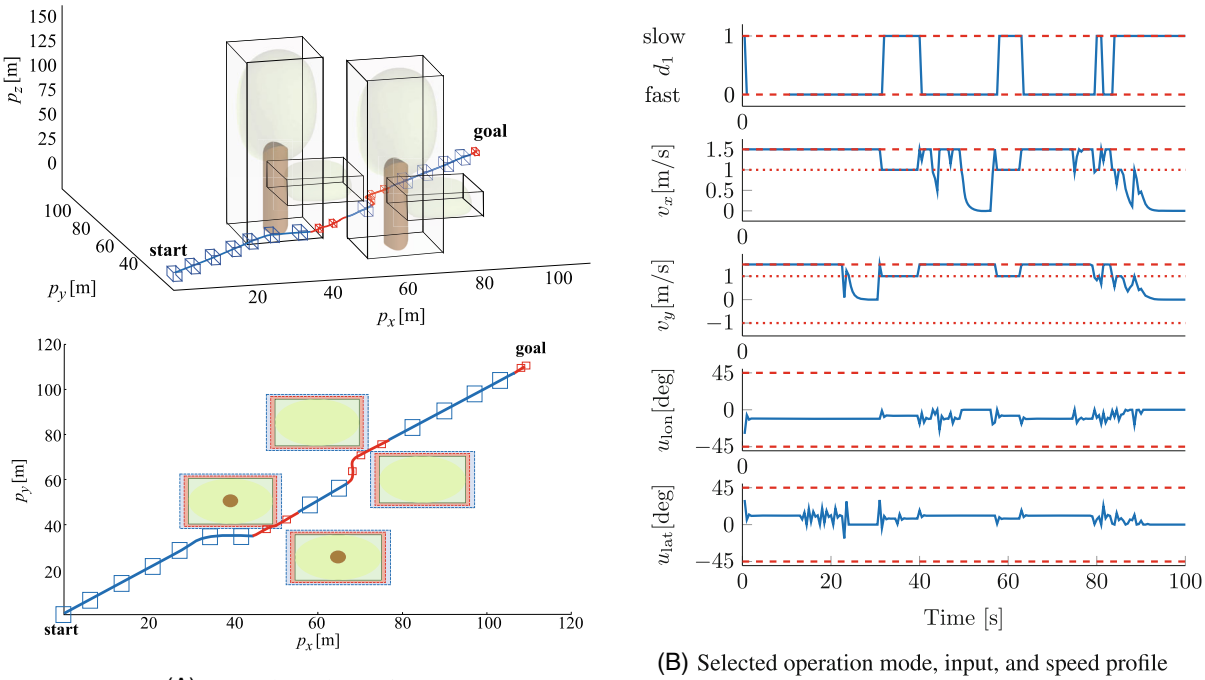


FIGURE 7 Disturbance free closed-loop simulation results for a single MPC controller mode.



(A) 3D and top down view.

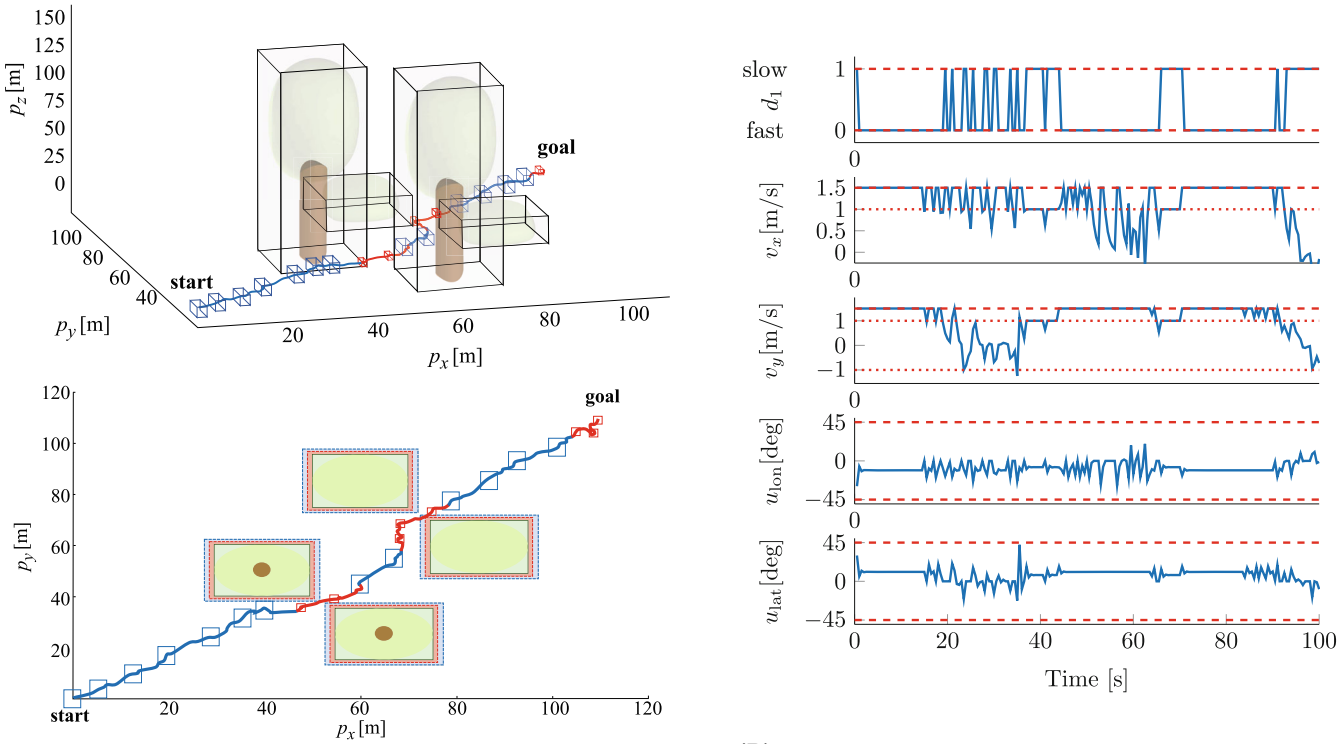
(B) Selected operation mode, input, and speed profile

FIGURE 8 (A) Using two operating modes allows the quadcopter to reach the goal on a direct way. The two operating modes are fast (blue sets) and slow (red sets). (B) The reference planner switches between the two operation modes slow/fast (variable d_1), resulting in different velocity constraints and uncertainty bounds, leading to different constraint back-offs (blue and red sets around the obstacles and the vehicle trajectories).

A linearized model, as presented in Reference 40 based on a nonlinear dynamic model is used, compare Appendix A. The states of the quadcopter are roll and pitch angles (ϕ, θ), roll and pitch rates (w_x, w_y), 3D position (p_x, p_y, p_z), and 3D velocities (v_x, v_y, v_z). The linearized model has three inputs: $u_{\text{lon}} = \theta_c$, $u_{\text{lat}} = \phi_c$ and $u_{\text{alt}} = T_c$. The input and states are constrained to:

$$\begin{bmatrix} -1.5 \text{ m/s} \\ -\pi/4 \text{ rad} \\ -\pi/4 \text{ rad} \end{bmatrix} \leq \begin{bmatrix} v_x,\ v_y,\ v_z \\ \phi,\ \theta, \\ \phi_c,\ \theta_c \end{bmatrix} \leq \begin{bmatrix} 1.5 \text{ m/s} \\ \pi/4 \text{ rad} \\ \pi/4 \text{ rad} \end{bmatrix}. \tag{21}$$

The model is discretized with a sampling time of $T_p = 0.5$ s for the planning problem and $T_t = 0.05$ s, that is, $M = 10$, for the cyclic horizon MPC controller (5). YALMIP[41] is used to implement and formulate the planning and control



(A) 3D and top down view.

(B) Selected operation mode, input, and speed profile

FIGURE 9 (A) Strong wind disturbance case: the reference planner allows to reach the goal and avoid obstacles despite strong wind disturbance. (B) The reference planner switches between the two operation modes, resulting in different velocity constraints and uncertainty bounds.

problems, exploiting Gurobi[42] for the solution of the optimization problems. The required sets are calculated via the MPT toolbox.[43]

Figure 7 shows simulations for the case that only a single low layer control mode is used.

As can be seen, for a small planning horizon of $N = 15$ no path around the obstacle can be found. Only an increase of the planning horizon to $N = 30$, or the removal of the vertical velocity constraint on v_z allows the controlled vehicle to reach the goal. Note that in all cases no collisions occur. The (maximum) computation times to solve the planning problem are $T_{com} = 0.4$ s for $N = 15$, and $T_{com} = 60$ s for $N = 30$, which is well above the desired re-planning time of 0.5 s[†].

Figure 8 shows simulation results using two operation modes for the low-layer MPC controller. They correspond to a fast operation mode, given by $-1.5\,\text{m/s} \leq (v_x, v_y) \leq 1.5\,\text{m/s}$ and a slow operation mode given by $-1.0\,\text{m/s} \leq (v_x, v_y) \leq 1.0\,\text{m/s}$. The fast operation mode corresponds to a large uncertainty set, while slow operation mode leads to a smaller uncertainty bound $\mathbb{W}_2$ and thus smaller sets $\mathbb{E}_2$ and $\mathbb{Z}_2$.

Figure 9 shows simulation results for large wind disturbances, which are not considered in the controller explicitly. As can be seen, the hierarchical control strategy is able to achieve the goal, while avoiding the obstacles and satisfying the input constraints.

Summarizing, introducing additional control modes allows avoiding conservative behavior while satisfying constraints and being computationally feasible.

5 | CONCLUSIONS AND OUTLOOK

We propose a tightly integrated hierarchical predictive control and planning approach. Both planner and controller repeatedly solve moving horizon optimal control problems. The upper- and lower-layer exploit different "contracts" (guaranteed

uncertainty bounds), that is, the planner can choose another low-layer controller modes instead of using a fixed safety corridor to capture the controller capabilities. For example, the planner can choose slow-speed movement with high precision or a fast speed with large uncertainty bounds.

The planning reference is determined by solving a moving horizon optimization problem considering a simplified model. It exploits constraints tightening, which represents the lower-layer tracking capabilities in the form of the precision contracts. The resulting planning algorithm can be reformulated as an MILP, allowing for efficient and reliable solutions. The low-layer tube-based MPC controller utilizes a cyclic horizon and results in a convex optimization problem. It guarantees constraint satisfaction and the desired tracking accuracy for the different modes. Moreover, it operates at a faster time scale. We derived conditions that ensure compatibility between the planning and control layers to guarantee recursive feasibility and ensure the satisfaction of constraints and obstacle avoidance.

Simulation results demonstrated the efficiency and applicability of the proposed hierarchical strategy. First, the contract option provides significant advantages, for example, it leads to a less conservative solution. Moreover, the hierarchical decomposition of the challenging vehicle control/planning problem leads to a decrease in the computational cost. It allows the implementation of robust control on-board while providing guarantees.

Possible extensions are the consideration of ellipsoidal tube MPC methods.[38,44] In this case, the lower controller online sends the tube parameterization to the upper layer. Therefore, the planner can predict a possible uncertainty evaluation over the planning horizon.

We also aim to experimentally evaluate the approach, implementing the upper-layer planner and the lower-layer MPC controller on computationally limited systems.

DATA AVAILABILITY STATEMENT

Data sharing not applicable to this article as no datasets were generated or analysed during the current study.

ENDNOTES

*In principle the sets $\mathbb{X}_i$ may overlap, that is, one can have that $\mathbb{X}_i \cap \mathbb{X}_j \neq \emptyset$ for $i \neq j$.

†The computation times are carried out on an Intel® Core™ i7-8550U CPU which operates on 1.99GHz.

ACKNOWLEDGMENT

Open Access funding enabled and organized by Projekt DEAL.

CONFLICT OF INTEREST STATEMENT

The authors declare no potential conflict of interests.

ORCID

Rolf Findeisen https://orcid.org/0000-0002-9112-5946

REFERENCES

1. Paden B, Čáp M, Yong SZ, Yershov D, Frazzoli E. A survey of motion planning and control techniques for self-driving urban vehicles. *IEEE Trans Int Veh*. 2016;1(1):33-55.
2. Schwarting W, Alonso-Mora J, Rus D. Planning and decision-making for autonomous vehicles. *Annu Revi Control Robotics Autonom Syst*. 2018;1:187-210.
3. Dadkhah N, Mettler B. Survey of motion planning literature in the presence of uncertainty: considerations for UAV guidance. *J Intell Robot Syst*. 2012;65(1-4):233-246.
4. Kingston Z, Moll M, Kavraki LE. Sampling-based methods for motion planning with constraints. *Annu Revi Control Robotics Autonom Syst*. 2018;1:159-185.
5. Ibrahim M. *Real-Time Moving Horizon Planning and Control of Aerial Systems under Uncertainties*. Phd thesis. Otto von Guericke University, Institute for Automation Engineering; 2020.
6. Rosolia U, Singletary A, Ames AD. Unified multi-rate control: from low level actuation to high level planning. arXiv preprint, arXiv:2012.06558 2020.
7. Pant YV, Yin H, Arcak M, Seshia SA. Co-design of control and planning for multi-rotor UAVs with signal temporal logic specifications. *Proc. American Control Conference (ACC)*. IEEE; 2021:4209-4216.
8. Broderick JA, Tilbury DM, Atkins EM. Optimal coverage trajectories for a UGV with tradeoffs for energy and time. *Auton Robot*. 2014;3:257-271.

9. Ibrahim M, Matschek J, Morabito B, Findeisen R. Hierarchical model predictive control for autonomous vehicle area coverage. *IFAC-PapersOnLine*. 2019;52(12):79-84.
10. Yin H, Bujarbaruah M, Arcak M, Packard A. Optimization based planner–tracker design for safety guarantees. *Proc. American Control Conference (ACC)*. IEEE; 2020:5194-5200.
11. Ibrahim M, Matschek J, Morabito B, Findeisen R. Improved area covering in dynamic environments by nonlinear model predictive path following control. *IFAC-PapersOnLine*. 2019;52(15):418-423.
12. Koeln J, Alleyne A. Two-level hierarchical Mission-based model predictive control. *Proc. American Control Conference (ACC)*. IEEE; 2018:2332-2337.
13. Vermillion C, Menezes A, Kolmanovsky I. Stable hierarchical model predictive control using an inner loop reference model and λ-contractive terminal constraint sets. *Automatica*. 2014;50(1):92-99.
14. Goerzen C, Kong Z, Mettler B. A survey of motion planning algorithms from the perspective of autonomous UAV guidance. *J Intell Robot Syst*. 2010;57(1-4):65.
15. Quan L, Han L, Zhou B, Shen S, Gao F. Survey of UAV motion planning. *IET Cyber-Syst Robotics*. 2020;2(1):14-21.
16. Mac TT, Copot C, Tran DT, De Keyser R. Heuristic approaches in robot path planning: a survey. *Robot Auton Syst*. 2016;86:13-28.
17. Grüne L, Pannek J. *Nonlinear Model Predictive Control*. Springer; 2017.
18. Findeisen R, Allgöwer F. An introduction to nonlinear model predictive control. *21st Benelux Meeting on Systems and Control*. Vol 11; 2002:119-141.
19. Rawlings JB, Mayne DQ, Diehl M. *Model Predictive Control: Theory, Computation, and Design*. Nob Hill Publishing; 2017.
20. Chisci L, Rossiter JA, Zappa G. Systems with persistent disturbances: predictive control with restricted constraints. *Automatica*. 2001;37(7):1019-1028.
21. Mayne DQ, Seron MM, Raković S. Robust model predictive control of constrained linear systems with bounded disturbances. *Automatica*. 2005;41(2):219-224.
22. Kögel M, Findeisen R. Stability of NMPC with cyclic horizons. *IFAC Proc Vol*. 2013;46(23):809-814.
23. Ibrahim M, Kögel M, Kallies C, Findeisen R. Contract-based hierarchical model predictive control and planning for autonomous vehicle. *IFAC-PapersOnLine*. 2020;53(2):15758-15764.
24. Bäthge T, Kögel M, Di Cairano S, Findeisen R. Contract-based predictive control for modularity in hierarchical systems. *IFAC-PapersOnLine*. 2018;51(20):499-504.
25. Lucia S, Kögel M, Findeisen R. Contract-based predictive control of distributed systems with plug and play capabilities. *IFAC-PapersOnLine*. 2015;48(23):205-211.
26. Blasi S, Kögel M, Findeisen R. Distributed model predictive control using cooperative contract options. *IFAC-PapersOnLine*. 2018;51(20):448-454.
27. Trodden P, Richards A. Multi-vehicle cooperative search using distributed model predictive control. *AIAA Guidance, Navigation and Control Conference and Exhibit*. AIAA; 2008:7138.
28. Ibrahim M, Kallies C, Findeisen R. Learning-supported approximated optimal control for autonomous vehicles in the presence of state dependent uncertainties. *Proc European Control Conf IEEE*. 2020;338-343. https://doi.org/10.23919/ECC51009.2020.9143737
29. Pinto SC, Afonso RJ. Risk constrained navigation using MILP-MPC formulation. *IFAC-PapersOnLine*. 2017;50(1):3586-3591.
30. Richards A, How JP. Robust variable horizon model predictive control for vehicle maneuvering. *Int J Robust Nonlinear Control*. 2006;16(7):333-351.
31. Kerrigan EC. *Robust Constraint Satisfaction: Invariant Sets and Predictive Control*. Phd thesis. University of Cambridge; 2001.
32. Wang Y, Boyd S. Fast model predictive control using online optimization. *IEEE Trans Control Syst Technol*. 2009;18(2):267-278.
33. Zometa P, Kögel M, Findeisen R. μAO-MPC: a free code generation tool for embedded real-time linear model predictive control. *Proceedings IEEE 2013 American Control Conference*. 2013:5320-5325. https://doi.org/10.1109/ACC.2013.6580668
34. Kögel M, Findeisen R. Robust MPC with reduced conservatism by blending multiples tubes. *Proceeding of the American Control Conference (ACC)*. IEEE; 2020.
35. Raković SV, Kouvaritakis B, Cannon M, Panos C, Findeisen R. Fully parameterized tube MPC. *IFAC Proc Vol*. 2011;44(1):197-202.
36. Raković SV. Invention of prediction structures and categorization of robust MPC syntheses. *IFAC Conf. Nonlinear Model Predictive Control*. IFAC; 2012:245-273.
37. Kögel M, Findeisen R. Discrete-time robust model predictive control for continuous-time nonlinear systems. *Proc. American Control Conference (ACC)*. IEEE; 2015:924-930.
38. Villanueva ME, Li JC, Feng X, Chachuat B, Houska B. Computing ellipsoidal robust forward invariant tubes for nonlinear MPC. *IFAC-PapersOnLine*. 2017;50(1):7175-7180.
39. Findeisen R, Grüne L, Pannek J, Varutti P. Robustness of prediction based delay compensation for nonlinear systems. *IFAC Proc Vol*. 2011;44(1):203-208.
40. Alexis K, Papachristos C, Siegwart R, Tzes A. Robust model predictive flight control of unmanned rotorcrafts. *J Intell Robot Syst*. 2016;81(3-4):443-469.
41. Löfberg J. YALMIP: a toolbox for modeling and optimization in MATLAB. *Proc. IEEE Int. Symp. On Comp. Aided Cont. Syst. Design*. IEEE; 2004:284-289.
42. Gurobi Optimization LLC. Gurobi optimizer reference manual. https://www.gurobi.com. 2022.
43. Kvasnica M, Grieder P, Baotić M. Multi-Parametric Toolbox (MPT). *Hybrid Systems: Computation and Control: 7th International Workshop, HSCC 2004*, Philadelphia, PA, USA, March 25-27, Springer; 2004:448-462.

44. Hu H, Feng X, Quirynen R, Villanueva ME, Houska B. Real-time tube MPC applied to a 10-state quadrotor model. *Proc. American Control Conference (ACC)*. IEEE; 2018:3135-3140.
45. Limon D, Alvarado I, Alamo T, Camacho E. MPC for tracking piecewise constant references for constrained linear systems. *Automatica*. 2008;44(9):2382-2387.
46. Simon D, Löfberg J, Glad T. Reference tracking MPC using dynamic terminal set transformation. *IEEE Trans Autom Control*. 2014;59(10):2790-2795.

How to cite this article: Kögel M, Ibrahim M, Kallies C, Findeisen R. Safe hierarchical model predictive control and planning for autonomous systems. *Int J Robust Nonlinear Control*. 2023;1-19. doi: 10.1002/rnc.6808

APPENDIX A. QUADCOPTER MODEL

The quadcopter states and inputs are represented in two different coordinate systems, for example, earth and body fixed frame, see Figure 6. The resulting nonlinear dynamics are given by Reference 40:

$$\begin{bmatrix} m_t \mathbf{I}_{3\times 3} & 0 \\ 0 & \mathbf{I} \end{bmatrix} \begin{bmatrix} \dot{V} \\ W \end{bmatrix} + \begin{bmatrix} W \times m_t V \\ W \times \mathbf{I} W \end{bmatrix} = \begin{bmatrix} F \\ T \end{bmatrix}, \tag{A1}$$

where m_t and I are the mass and inertia matrix, V and W are the linear and angular velocities expressed in the body-fixed frame. F and T are the applied forces and moments. The model is linearized assuming decoupling of the translational and the attitude dynamics,[40] leading to

$$\dot{x}_{3\mathrm{D}} = A_{3\mathrm{D}} x_{3\mathrm{D}} + B_{3\mathrm{D}} u_{3\mathrm{D}}, \tag{A2a}$$

$$y_{3\mathrm{D}} = C_{3\mathrm{D}} x_{3\mathrm{D}} + D_{3\mathrm{D}} u_{3\mathrm{D}}, \tag{A2b}$$

with: $x_{3\mathrm{D}} = [x_{\mathrm{lon}}^\top \; x_{\mathrm{lat}}^\top \; x_{\mathrm{alt}}^\top]^\top$, $u_{3\mathrm{D}} = [u_{\mathrm{lon}}^\top \; u_{\mathrm{lat}}^\top \; u_{\mathrm{alt}}^\top]^\top$, $A_{3\mathrm{D}} = \begin{bmatrix} A_{\mathrm{lon}} & 0 & 0 \\ 0 & A_{\mathrm{lat}} & 0 \\ 0 & 0 & A_{\mathrm{alt}} \end{bmatrix}$, $B_{3\mathrm{D}} = \begin{bmatrix} B_{\mathrm{lon}} & 0 & 0 \\ 0 & B_{\mathrm{lat}} & 0 \\ 0 & 0 & B_{\mathrm{alt}} \end{bmatrix}$.The corresponding longitudinal, lateral and vertical sub-dynamics are given by:

$$\dot{x}_{\mathrm{lon}} = A_{\mathrm{lon}} x_{\mathrm{lon}} + B_{\mathrm{lon}} u_{\mathrm{lon}}, \tag{A3a}$$

$$\dot{x}_{\mathrm{lat}} = A_{\mathrm{lat}} x_{\mathrm{lat}} + B_{\mathrm{lat}} u_{\mathrm{lat}}, \tag{A3b}$$

$$\dot{x}_{\mathrm{alt}} = A_{\mathrm{alt}} x_{\mathrm{alt}} + B_{\mathrm{alt}} u_{\mathrm{alt}}, \tag{A3c}$$

with the matrices $A_{\mathrm{lon}} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\lambda_x & -g & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -a_{w_x,\theta} & -a_{w_x,w_x} \end{bmatrix}$, $B_{\mathrm{lon}} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ b_x \end{bmatrix}$,where the longitudinal state is $x_{\mathrm{lon}} = [p_x \; v_x \; \theta \; w_x]^\top$, with the input $u_{\mathrm{lon}} = \theta_c$. The lateral state is $x_{\mathrm{lat}} = [p_y \; v_y \; \phi \; w_y]^\top$, with the input $u_{\mathrm{lat}} = \phi_c$ and the matrices $A_{\mathrm{lat}} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\lambda_y & g & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -a_{w_y,\phi} & -a_{w_y,w_y} \end{bmatrix}$, $B_{\mathrm{lat}} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ b_y \end{bmatrix}$. Finally, the matrices for vertical (altitude) dynamics are given by $A_{\mathrm{alt}} = \begin{bmatrix} 0 & 1 \\ 0 & -\lambda_z \end{bmatrix}$, $B_{\mathrm{alt}} = \begin{bmatrix} 0 \\ b_z \end{bmatrix}$,with thevertical state $x_{\mathrm{alt}} = [p_z \, v_z]^\top$, and the input is $u_{\mathrm{alt}} = T_c$. The overall UAV states are the roll and pitch angles (ϕ, θ), roll and pitch rates (w_x, w_y), the positions (p_x, p_y, p_z), and the velocities (v_x, v_y, v_z). The parameters $\lambda_x, \lambda_y, \lambda_z, a_{w_x,\theta}, a_{w_x,w_x}, a_{w_y,\phi}, a_{w_y,w_y}, b_x, b_y, b_z$ can be found in Reference 40.

APPENDIX B. SUITABLE TERMINAL SET

In the following, we present one method to determine the terminal sets $\mathbb{X}_i^f$ satisfying Assumption 4. Inspired by the works,[45,46] the terminal state x_p are composed of two parts: a steady state $\tilde{x}_p$ for input $\tilde{u}_p$ and a state $\bar{x}_p$ inside

an invariant set $\overline{\mathbb{X}}_i^f$.

$$x_p = \tilde{x}_p + \bar{x}_p, \qquad \tilde{x}_p = A\tilde{x}_p + B\tilde{u}_p, \qquad \bar{x}_p \in \overline{\mathbb{X}}_i^f. \tag{B1a}$$

The artificial terminal control law takes the form $\kappa_i^f(x_p) = \tilde{u}_p + K_p^f \bar{x}_p$. Note that $\tilde{x}_p$ and $\tilde{u}_p$ are a steady state of a nominal version of the fast dynamics ((1a) with zero $w(k)$) and thus also the nominal planning dynamics (4)).

The set $\overline{\mathbb{X}}_i^f$ is assumed to be positive invariant: $(A_p + B_p K_p^f)\overline{\mathbb{X}}_i^f \subseteq \overline{\mathbb{X}}_i^f$. One can bound the inter-sample behavior for states $\bar{x}_p$ starting inside this set using a set $\mathbb{H}_i$

$$\forall \bar{x}_p \in \overline{\mathbb{X}}_i^f,\ \forall \ell = 1, \dots, M-1 : A^\ell \bar{x}_p + \sum_{m=0}^{\ell-1} A^m K_p^f \bar{x}_p \in \overline{\mathbb{X}}_i^f \oplus \mathbb{H}_i.$$

Both sets are assumed to be convex, compact polytopes. The conditions (B1a) together with

$$\tilde{x}_p \in \mathbb{X}_i \ominus \mathbb{Z}_i \ominus \overline{\mathbb{X}}_i^f \ominus \mathbb{H}_i, \qquad \tilde{u}_p \in \mathbb{U}_i \ominus K\mathbb{Z}_i \ominus K_p^f \overline{\mathbb{X}}_i^f, \tag{B1b}$$

$$C\tilde{x}_p \notin \mathbb{O} \oplus (-C)\mathbb{Z}_i \oplus (-C)\overline{\mathbb{X}}_i^f \oplus (-C)\mathbb{H}_i, \tag{B1c}$$

allow us to formulate the terminal sets $\mathbb{X}_i^f$ as

$$\mathbb{X}_i^f = \{\tilde{x}_p, \bar{x}_p \ \text{ s.t. } \ (B1) \ \text{ holds.}\} \tag{B2}$$

One can easily show that the calculated terminal sets $\mathbb{X}_i^f$ satisfy Assumption 4. The nonconvex obstacle avoidance constraint (B1c) appears and can be handled in a similar fashion as in (7). Here we assume that the set $\overline{\mathbb{X}}_i^f$ is fixed and its size is determined and tuned offline. In principle one can relax this conditions similarly as in Reference 46 using a scaling of this set or the ideas of Reference 45.

Note that one can determine the required sets $\overline{\mathbb{X}}_i^f$ and $\mathbb{H}_i$ using YALMIP[41] and the MPT toolbox.[43]