A Terminal State Feasibility Governor for Real-Time Nonlinear Model Predictive Control over Arbitrary Horizons

Bryan Convens[1,2], Dominic Liao-McPherson[3], Kelly Merckaert[1,2], Bram Vanderborght[1,2], and Marco M. Nicotra[4]

***Abstract*—This paper introduces a novel Feasibility Governor (FG), which enlarges the region of attraction of a Nonlinear Model Predictive Control (NMPC) setpoint regulation law with an arbitrarily short prediction horizon. The efficient online FG is developed for nonlinear systems subject to pointwise-in-time state and input constraints and relies on a discrete-time solution of a trajectory-based Explicit Reference Governor with Lyapunov-based terminal energy constraint. It ensures the FG-NMPC scheme's recursive feasibility to any target and asymptotic stability to constant targets by adaptively integrating the derivative of an auxiliary reference applied to the closed-loop NMPC. Compared to recently published FG schemes, this scheme scales better to higher-dimensional nonlinear systems with a priori unknown constraints, as it does not require expensive offline computations to construct the feasible set or the maximal output admissible set associated with the NMPC's terminal control law. The scheme is implemented as a C++ algorithm and validated through simulations on a quadrotor that aggressively but safely flies through a priori unknown environments cluttered with obstacles. It is shown to satisfy all constraints for any piecewise-continuous reference, achieve asymptotic stability and zero-offset tracking to constant constraint-admissible targets, and require low computational effort. Supplementary video material can be found at https://youtu.be/2LSYNwuYpzI.**

***Index Terms*—Nonlinear Model Predictive Control, Explicit Reference Governor, Collision Avoidance, Aerial Vehicle**

I. Introduction

MODEL Predictive Control [1], [2] (MPC) is a high-performance control strategy that defines a feedback policy by solving a receding horizon Optimal Control Problem (OCP) at every sampling instant. MPC is widely applied since it can handle dynamical systems with input and state constraints and is supported by robust theoretical literature [3]. Due to the numerous methods for designing MPC laws, stability guarantees are useful to ensure that the closed-loop system behaves as desired. Typically, stability and constraints are ensured by incorporating terminal ingredients into the OCP, e.g., a terminal cost and an invariant terminal set [4].

Often, MPC laws require the ability to track piecewise-constant references [5] and to transition between them. Although this is a simple modification to the OCP, if the reference command being tracked has a large change in value, the OCP may become infeasible since the terminal set is no longer reachable within the specified prediction horizon. While increasing the prediction horizon is an intuitive method for avoiding this problem, it is not always applicable since it drastically increases the computational complexity of the underlying OCP. This can result in significantly longer computation times that may exceed the real-time requirements.

Another technique for avoiding OCP infeasibility is to treat aspects of the terminal set as optimization variables inside the OCP and use these additional degrees of freedom to enlarge the terminal set. This method has been applied to various tracking problems [5], [6] as well as the economic operation of nonlinear systems with terminal state constraints [7]. A different approach is employed in [8], where a contractive sequence of terminal sets is computed offline. This sequence is then incorporated into the OCP to increase the Region Of Attraction (ROA) of the MPC controller. A drawback of all these methods is that they require OCP modifications, making them difficult to implement on standardized MPC toolboxes.

An alternative approach is to modify the reference to ensure that the OCP remains feasible. The dual-mode controller in [9] incorporates a recovery mode that computes a modified reference and control input. Although this method converges in finite-time, it may reduce performance. Ad-hoc methods such as rate limiting the reference are commonly used but are suboptimal, and do not come with theoretical guarantees. A governor-like algorithm in [10] uses ellipsoidal terminal sets and a specific reference parameterization to avoid infeasibility. A different governor-like algorithm is proposed in [11] and is used as an intermediate design stage for an explicit MPC. Unfortunately, all these methods require OCP modifications, very often significantly increasing the computational cost, and expensive offline computations require a priori known constraints. Therefore, they are mostly limited to linear systems.

A. Prior works on Feasibility Governors for linear systems

The Feasibility Governor (FG), first introduced in [12], is an add-on unit to a closed-loop MPC which is inspired by Reference & Command Governors (RG & CG) [13], [14]. An FG manipulates the reference to guarantee safe transitions between setpoints while avoiding OCP infeasibility. Moreover, it expands the MPC's ROA to the set of initial conditions that can reach the terminal set of any steady-state admissible reference. This is accomplished by solving an online optimization problem using the MPC's feasible set, i.e., its N-step backwards reachable set. While this method is modular and avoids OCP infeasibility, computing feasible sets is challenging and only realistic offline. Using explicit feasible sets is intuitive for FG construction but comes with implementation downsides.

This work was supported by the Flemish Government under the program Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen and EU project SPEAR 101119774.

[1] Authors are with the Mechanical Engineering Dep., Vrije Universiteit Brussel, Brussels, Belgium.

[2] Bryan Convens (e-mail: bryan.convens@vub.be), Kelly Merckaert (e-mail: kelly.merckaert@vub.be), and Bram Vanderborght (e-mail: bram.vanderborght@vub.be) are affiliated to the Interuniversity Microelectronics Institute (IMEC), Leuven, Belgium.

[3] Dominic Liao-McPherson (e-mail: dliaomcp@mech.ubc.ca) is with the Dep. of Mechanical Eng., University of British Columbia, Vancouver, BC V6T 1Z4, Canada.

[4] Marco M. Nicotra (e-mail: marco.nicotra@colorado.edu) is with the Electrical, Computer, and Energy Engineering Dep., University of Colorado Boulder, Boulder, CO, USA.

Typically, feasible sets are ellipsoidal or polyhedral. While ellipsoidal sets can be easier to compute, they lead to quadratic constraints and are typically conservative. Polyhedral feasible sets are relatively easy to compute offline by orthogonal projections [15] for which several toolboxes are available [16], [17]. The downside is that projection methods suffer from the curse of dimensionality, and inner-approximations [18], [19] are usually required even for moderately sized systems (e.g., see [20, Fig. 3]). Moreover, computing these feasible sets becomes even more intractable for nonlinear systems.

Recently, a terminal set FG [20] was proposed that does not require an explicit estimate of the feasible set nor OCP modifications. As opposed to [12], the formulation avoids the issue of offline computational intractability that arises when performing high-dimensional set-valued polyhedral projections to compute the feasible set. Instead, it uses information from the terminal set to modify the reference ensuring both safety and stability. However, the main issue is that this FG needs to compute offline an explicit representation of a convex polyhedral terminal set, chosen as an inner approximation of the Maximal Output Admissible Set (MOAS) of the LQR stabilized system subject to a priori known constraints. This set is the largest constraint admissible positively invariant set [13], [21], [22], which is then used in the FG to solve a Quadratic Program (QP) online. Algorithms for computing the MOAS are described for linear dynamics in [21, Section III.], but require the solution of a sequence of expensive mathematical programs. To partially overcome the high computational cost, polyhedral inner approximations of the constraint set and the possibly nonconvex MOAS can be computed by solving a sequence of linear programs. Once again, this is only deemed realistic offline. Reference [22] generalized the MOAS theory to the case of a certain class of nonlinear systems, i.e., polynomial systems subject to polyhedral state constraints. However, solutions of expensive nonlinear programs are again required which rapidly becomes intractable as the dimension and complexity of the system and constraints increase.

In conclusion, computational intractability of the feasible and terminal sets makes the application domain of existing FGs [12], [20] limited to linear dynamics, polyhedral constraints, and where all constraints are known before run-time, as online re-computation of these sets would exceed the real-time requirements. Hence, these methods cannot be applied to agile quadrotor flight through a priori unknown environments populated with obstacles that can only be detected on-the-fly.

B. Contributions of the proposed Feasibility Governor for nonlinear systems subject to nonconvex constraints

The Explicit Reference Governor (ERG) [23] is a continuous-time closed-form RG that has been successfully applied to nonlinear systems subject to input and (possibly nonconvex) state constraints. In recent years, several ERG formulations have been proposed using Lyapunov and invariant sets in [24], [25] and trajectory predictions in [26]–[28], of which the latter achieves superior performance. Taking inspiration from the continuous-time version of the trajectory-based ERG [23], this article proposes a discrete-time predictive terminal state FG for nonlinear constrained systems controlled by an NMPC law with an arbitrary (short) prediction horizon. This FG ensures feasibility for any reference and asymptotic stability to a steady-state admissible approximation of a constant reference with no offline and little additional online computational cost compared to significantly more computationally expensive NMPC. This is achieved because the proposed FG does not require an explicit formulation of the feasible nor the terminal set, and instead directly operates on the terminal state and the actual (nonlinear) constraints activated in the NMPC's OCP. Moreover, it can seamlessly consider new constraints that were not known before run-time (e.g., obstacles). The efficacy of the real-time FG-NMPC algorithm is demonstrated via hardware-in-the-loop simulations on an actuation-constrained planar quadrotor that safely and aggressively flies through an obstacle-cluttered environment.

II. Problem Formulation

Consider a nonlinear discrete-time dynamical system

$$\boldsymbol{x}(k+1) = \boldsymbol{f}(\boldsymbol{x}(k), \boldsymbol{u}(k)) \tag{1a}$$
$$\boldsymbol{y}(k) = \boldsymbol{g}(\boldsymbol{x}(k), \boldsymbol{u}(k)) \tag{1b}$$
$$\boldsymbol{z}(k) = \boldsymbol{h}(\boldsymbol{x}(k), \boldsymbol{u}(k)), \tag{1c}$$

where $k \in \mathbb{N}$ is the discrete-time index and $\boldsymbol{x}(k) \in \mathbb{R}^{n_x}$, $\boldsymbol{u}(k) \in \mathbb{R}^{n_u}$, $\boldsymbol{y}(k) \in \mathbb{R}^{n_y}$, and $\boldsymbol{z}(k) \in \mathbb{R}^{n_z}$ are the states, control inputs, constrained outputs, and tracking outputs.

System (1) is subject to pointwise-in-time output constraints

$$\boldsymbol{y}(k) \in \mathcal{Y}, \quad \forall k \in \mathbb{N}, \tag{2}$$

where $\mathcal{Y} \subseteq \mathbb{R}^{n_y}$ is a closed set of static constraints.

Assumption 1: There exists an equilibrium manifold of connected[1] and constraint admissible equilibria for system (1). We introduce an auxiliary reference $\boldsymbol{v}(k) \in \mathbb{R}^{n_v}$ that parameterizes this equilibrium manifold, i.e., every solution to (1a)-(1c) for which $\boldsymbol{x}(k+1) = \boldsymbol{x}(k) = \boldsymbol{f}(\bar{\boldsymbol{x}}_{v(k)}, \bar{\boldsymbol{u}}_{v(k)})$, as

$$\bar{\boldsymbol{x}}_{v(k)} = \boldsymbol{m}_x(\boldsymbol{v}(k)), \tag{3a}$$
$$\bar{\boldsymbol{u}}_{v(k)} = \boldsymbol{m}_u(\boldsymbol{v}(k)), \tag{3b}$$
$$\bar{\boldsymbol{z}}_{v(k)} = \boldsymbol{m}_z(\boldsymbol{v}(k)) = \boldsymbol{h}(\bar{\boldsymbol{x}}_{v(k)}, \bar{\boldsymbol{u}}_{v(k)}), \tag{3c}$$

where the $\boldsymbol{m}_{(\cdot)}$ functions are (locally) unique and invertible.

MPC laws cannot stabilize points on the boundary of the feasible set, so we introduce a design parameter $\delta \in (0, 1)$ and a set of strictly steady-state admissible auxiliary references

$$\mathcal{V}_\delta \triangleq \boldsymbol{m}_y^{-1}\left((1-\delta)\,\mathcal{Y}\right) = \{\boldsymbol{v} \mid \boldsymbol{m}_y(\boldsymbol{v}) \in (1-\delta)\mathcal{Y}\}, \tag{4}$$

where $\boldsymbol{m}_y(\boldsymbol{v}) \triangleq \boldsymbol{g}(\boldsymbol{m}_x(\boldsymbol{v}), \boldsymbol{m}_u(\boldsymbol{v})) = \boldsymbol{g}(\bar{\boldsymbol{x}}_v, \bar{\boldsymbol{u}}_v)$. The set of strictly steady-state admissible target references is defined as

$$\mathcal{R}_\delta \triangleq \boldsymbol{m}_z\left(\mathcal{V}_\delta\right) = \{\boldsymbol{m}_z(\boldsymbol{v}) \mid \boldsymbol{v} \in \mathcal{V}_\delta\}, \tag{5}$$

and the set of strictly steady-state admissible equilibria as

$$\mathcal{E} \triangleq \{(\bar{\boldsymbol{x}}_v, \boldsymbol{v}) \mid \bar{\boldsymbol{x}}_v = \boldsymbol{m}_x(\boldsymbol{v}),\ \boldsymbol{v} \in \mathcal{V}_\delta\}. \tag{6}$$

[1] Although there exist trajectory optimization schemes for moving a system between unconnected equilibrium sets, the proposed FG and the existing FGs require a connected path of steady-state admissible equilibria to exists.

Control Objectives: Given the nonlinear system (1) with constraint set $\mathcal{Y} \subseteq \mathbb{R}^{n_y}$, and let $\boldsymbol{r}(k) \in \mathbb{R}^{n_z}$ be a piecewise-continuous target reference. The goal of this paper is to design a state feedback law that achieves the following objectives:

1) Safety: ensure that $\boldsymbol{y}(k) \in \mathcal{Y}$, $\forall k \geq 0$;
2) Convergence[2]: $\lim_{k\to\infty} \boldsymbol{z}(k) = \boldsymbol{r}^*(k)$, where $\boldsymbol{r}^*(k) = \arg\min_{\boldsymbol{s}(k)\in\mathcal{R}_\delta} ||\boldsymbol{s}(k) - \boldsymbol{r}(k)||$;
3) Asymptotic stability: $\forall \epsilon > 0, \exists \delta > 0 : ||\boldsymbol{x}(0) - \boldsymbol{m}_x(\boldsymbol{v}_{r^*(0)})|| < \delta \Rightarrow ||\boldsymbol{x}(k) - \boldsymbol{m}_x(\boldsymbol{v}_{r^*(k)})|| < \epsilon, \forall k \geq 0$ (stable) and $\exists \gamma > 0 : ||\boldsymbol{x}(0) - \boldsymbol{m}_x(\boldsymbol{v}_{r^*(0)})|| < \gamma \Rightarrow \lim_{k\to\infty}(\boldsymbol{x}(k), \boldsymbol{v}(k)) = (\bar{\boldsymbol{x}}_{r^*(k)}, \boldsymbol{v}_{r^*(k)})$ (attractive), with an asymptotically stable equilibrium point $(\bar{\boldsymbol{x}}_{r^*(k)}, \boldsymbol{v}_{r^*(k)}) = (\boldsymbol{m}_x(\boldsymbol{v}_{r^*(k)}), \boldsymbol{v}_{r^*(k)})$ satisfying $\boldsymbol{r}^*(k) = \boldsymbol{m}_z(\boldsymbol{v}_{r^*(k)})$;
4) Performance: when it is feasible to do so, the control law should be optimal for a suitably weighted cost function.
5) Computational complexity: ensure 1)−4) in real-time without prior task knowledge on constraints $\mathcal{Y}$ or targets $\boldsymbol{r}(k)$ nor rely on offline computations before run-time.

III. Proposed Control Framework

The main challenge is to ensure that the nonlinear dynamics satisfy output constraints in real-time for any piecewise-continuous target reference and asymptotically reach a constant target, or an admissible approximation thereof, without any prior task knowledge on setpoints or constraints.

We approach the control objectives using an NMPC formulation. However, a major problem of NMPC laws is that the underlying OCP may become infeasible depending on the initial conditions, the prediction horizon, and the imposed constraints. This problem is conceptually explained in Fig. 1a. Formal definitions of the used symbols are given in the next sections. When the initial condition is inside the NMPC's feasible (or N-step backwards reachable) set, i.e., $(\boldsymbol{x}(0), \boldsymbol{r}) \in \mathcal{F}_N^{\mathcal{T}}$ depicted in blue, for which the terminal state constraint is imposed in the OCP, then the terminal state reaches the terminal set after at most N prediction steps, i.e., $(\boldsymbol{x}(N), \boldsymbol{r}) \in \mathcal{T}$, as is depicted by the blue trajectory. However, without imposing the terminal set constraint in the OCP for the same initial condition, the terminal state might not reach the terminal set, i.e., $(\boldsymbol{x}(N), \boldsymbol{r}) \notin \mathcal{T}$, as is depicted by the red trajectory. Though, as visualized in Fig. 1b, there exists a subset of initial conditions, i.e., $\mathcal{F}_N \subset \mathcal{F}_N^{\mathcal{T}}$ depicted in green, for which the terminal set constraint does not need to be imposed in the OCP such that the terminal state reaches the terminal set, i.e., if $(\boldsymbol{x}(0), \boldsymbol{r}) \in \mathcal{F}_N \Rightarrow (\boldsymbol{x}(N), \boldsymbol{r}) \in \mathcal{T}$, which is depicted by the green trajectory.

As explained in Section I, it is computationally intractable to construct (or recompute online when new obstacles appear) $\mathcal{F}_N^{\mathcal{T}}$, $\mathcal{F}_N$ or $\mathcal{T}$, even for low-dimensional nonlinear systems and especially when subject to non-convex obstacle avoidance constraints. An effective solution would not require any of these sets. Interestingly, the existence of $\mathcal{F}_N$ indicates that there exist initial conditions for which an explicit construction of $\mathcal{T}$ is not necessary to find a constraint-admissible trajectory towards the target reference $\boldsymbol{r}$.

The objective of the proposed FG is to select a sequence of intermediate targets, i.e., auxiliary references $\boldsymbol{v}$, and to keep $(\boldsymbol{x}, \boldsymbol{v}) \in \mathcal{F}_N$ without explicitly constructing the full set $\mathcal{F}_N$, such that the intermediate targets are pair-wise reachable. This is depicted in Fig. 1b for the intermediate pairs $(\boldsymbol{x}_{(0|k)}, \boldsymbol{v}_{(k)}) \in \mathcal{F}_{N,(k)}$ and $(\boldsymbol{x}_{(0|k+1)}, \boldsymbol{v}_{(k+1)}) \in \mathcal{F}_{N,(k+1)}$ in a space with target reference $\boldsymbol{r}$ and two red spherical obstacles. For each pair $(\boldsymbol{x}_{(0|k)}, \boldsymbol{v}_{(k)})$, our solution computes a (sub)optimal N-step (padded one-step shifted) trajectory till a terminal state $\boldsymbol{x}_{(N|k)}$ by solving the NMPC's OCP without terminal set constraint, as is depicted in purple. Then, the $M - N$-step closed-loop trajectory prediction of a terminal stabilizing control law from $\boldsymbol{x}_{(N|k)}$ to $\boldsymbol{x}_{(M|k)}$ is computed by explicit integration, as is depicted in blue. The FG uses this terminal trajectory and the small grey terminal energy sets to safely update the reference $\boldsymbol{v}_{(k)}$ to $\boldsymbol{v}_{(k+1)}$ in real-time by integration of $\dot{\boldsymbol{v}}_{(k)}$ with an adaptive step-size, as will be explained in Section V. Note that in practice, we can choose the FG horizon much larger than the NMPC horizon, i.e., $M - N \gg N$, while keeping the computational cost low, as the FG does not involve any online optimization.

IV. Nonlinear Model Predictive Control Law

Due to the constrained nonlinear dynamics and high-performance objectives, we approach the control problem with NMPC. The feedback law at sample index k is defined using the solution to the following discrete-time parameterized OCP,

$$\min_{\boldsymbol{\mu}} \; p(\boldsymbol{\xi}_N, \boldsymbol{v}(k)) + \sum_{i=0}^{N-1} l(\boldsymbol{\xi}_i, \boldsymbol{\mu}_i, \boldsymbol{v}(k)) \tag{7a}$$

$$\text{s.t.} \quad \boldsymbol{\xi}_0 = \boldsymbol{x}(k), \tag{7b}$$

$$\boldsymbol{\xi}_{i+1} = \boldsymbol{f}(\boldsymbol{\xi}_i, \boldsymbol{\mu}_i), \quad i \in \mathbb{N}_{[0,N-1]}, \tag{7c}$$

$$\boldsymbol{g}(\boldsymbol{\xi}_i, \boldsymbol{\mu}_i) \in \mathcal{Y}, \qquad i \in \mathbb{N}_{[0,N-1]}, \tag{7d}$$

$$(\boldsymbol{\xi}_N, \boldsymbol{v}(k)) \in \mathcal{T} \;\; \text{[excluded in our OCP]}, \tag{8}$$

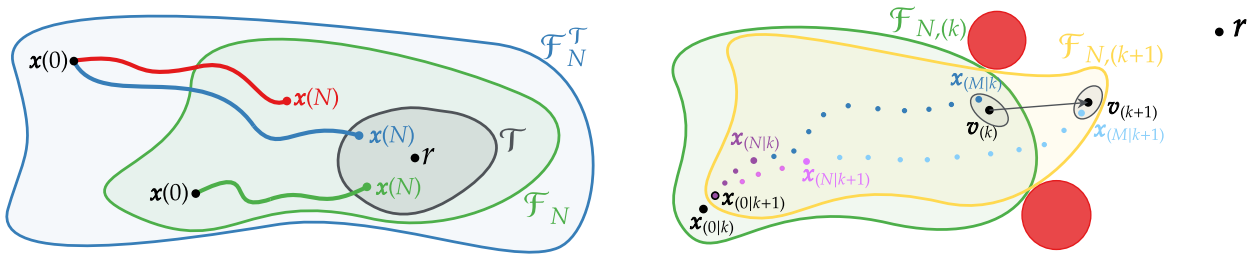
where $N \in \mathbb{N}^+$ is the number of predicted samples over the time horizon $H \in \mathbb{R}^+$ with $N+1$ nodes and a prediction time step of $\Delta t_p = H/N$ in between each pair of subsequent nodes, the parameters $\boldsymbol{x}(k)$ and $\boldsymbol{v}(k)$ denote the measured state and applied auxiliary reference signal (considered constant over the prediction horizon), $\boldsymbol{\mu} = (\boldsymbol{\mu}_0, \ldots, \boldsymbol{\mu}_{N-1})$ are the decision variables[3] containing the predicted inputs and the corresponding predicted states $\boldsymbol{\xi} = (\boldsymbol{\xi}_0, \ldots, \boldsymbol{\xi}_N)$ directly follow from the system dynamics (7b)-(7c), the continuous functions p and l in the objective function are the terminal state weighting and the per-stage weighting, and $\mathcal{T} \in \mathbb{R}^{n_x \times n_v}$ is a constraint admissible invariant terminal set[4].

The following assumptions ensure that (7)-(8) is well-posed and that it can be used to construct a stabilizing feedback law.

[2]Our formulation handles steady-state inadmissible targets $\boldsymbol{r}(k) \notin \mathcal{R}_\delta$ by steering the system to the closest strictly steady-state admissible constant target reference $\boldsymbol{r}^*(k)$. When the tracking problem is well posed, i.e., when $\boldsymbol{r}(k) \in \mathcal{R}_\delta$, we recover the more intuitive objective $\lim_{k\to\infty} \boldsymbol{z}(k) = \boldsymbol{r}(k)$.

[3]We use the subscript notation $\boldsymbol{\mu}_i$ as shorthand for $\boldsymbol{\mu}(i)$.

[4]Standard MPC formulations use a difficult-to-obtain terminal set constraint (8) to guarantee ∞-horizon stability and safety, but we solve the OCP in the absence of (8), which makes the MPC less computationally expensive.



(a) The infeasibility problem

(b) The proposed control framework

Fig. 1: The left figure visualizes the NMPC's feasible or N-step backwards reachable set $\mathcal{F}_N^{\mathcal{T}}$, its feasible subset $\mathcal{F}_N \subset \mathcal{F}_N^{\mathcal{T}}$, and the terminal set $\mathcal{T}$ for the target reference $\boldsymbol{r}$, the initial states $\boldsymbol{x}(0)$, and whether the terminal states $\boldsymbol{x}(N)$ of the three trajectories reach $\mathcal{T}$. The right figure depicts the proposed FG & NMPC approach consisting of: the arbitrary short-horizon N-step NMPC prediction from $\boldsymbol{x}_{(0|k)}$ to $\boldsymbol{x}_{(N|k)}$ without accounting for a terminal set constraint in the NMPC's OCP, the long-horizon $M-N$-step explicit trajectory prediction of a closed-form terminal stabilizing control law from the NMPC's terminal state $\boldsymbol{x}_{(N|k)}$ to the FG's terminal state $\boldsymbol{x}_{(M|k)}$, the small grey terminal closed-loop energy sets for ensuring ∞-horizon stability and safety from the FG's terminal state $\boldsymbol{x}_{(M|k)}$ onward, the current and updated auxiliary references $\boldsymbol{v}_{(k)}$ and $\boldsymbol{v}_{(k+1)}$ ensuring the system dynamics are attracted towards the desired reference $\boldsymbol{r}$ while satisfying output constraints including the avoidance constraints for the two red spherical obstacles.

Assumption 2: There exists a non-expensive (e.g., closed-form) terminal Pre-Stabilizing Control (PSC) law for (1a), i.e.,

$$\boldsymbol{u}(k) \triangleq \boldsymbol{\kappa}_{\text{PSC}}(\boldsymbol{x}(k), \boldsymbol{v}(k)) \,. \tag{9}$$

The terminal set $\mathcal{T}$ is positively invariant under (9) and $\forall (\boldsymbol{x}(k), \boldsymbol{v}(k)) \in \mathcal{T}$, the solution trajectory to the pre-stabilized (PS) terminal dynamics

$$\begin{aligned} \boldsymbol{x}(k+1) &\triangleq \boldsymbol{f}_{\text{PS}}(\boldsymbol{x}(k), \boldsymbol{v}(k)) \\ &= \boldsymbol{f}(\boldsymbol{x}(k), \boldsymbol{\kappa}_{\text{PSC}}(\boldsymbol{x}(k), \boldsymbol{v}(k)) \,, \end{aligned} \tag{10}$$

satisfies all constraints, i.e.,

$$\boldsymbol{y}(k) = \boldsymbol{g}(\boldsymbol{x}(k), \boldsymbol{\kappa}_{\text{PSC}}\,(\boldsymbol{x}(k), \boldsymbol{v}(k)) \in \mathcal{Y},\, \forall k \geq 0 \,. \tag{11}$$

Hence, we can choose the terminal set inside the MOAS $\mathcal{O}_\infty$

$$\begin{aligned} \mathcal{T} \subseteq \mathcal{O}_\infty \triangleq \{(\boldsymbol{x}, \boldsymbol{v}) \,|\, \boldsymbol{x}(k+1) = \boldsymbol{f}_{\text{PS}}(\boldsymbol{x}(k), \boldsymbol{v}), \\ \boldsymbol{x}(0) = \boldsymbol{x},\, \boldsymbol{y}(k) \in \mathcal{Y}, \forall k \geq 0\} \,. \end{aligned} \tag{12}$$

Assumption 3: For the terminal state weighting in (7a): $p(\bar{\boldsymbol{x}}_v, \boldsymbol{v}) = 0$ and $p(\boldsymbol{x}, \boldsymbol{v}) \geq 0,\ \forall (\boldsymbol{x}, \boldsymbol{v}) \in \mathcal{T}$. Also, $\exists\, \boldsymbol{\kappa}_{\text{PSC}}$: $\forall (\boldsymbol{x}, \boldsymbol{v}) \in \mathcal{T}$, the following relation between the terminal state weighting and the per-stage weighting holds

$$p(\boldsymbol{f}_{\text{PS}}(\boldsymbol{x}, \boldsymbol{v}), \boldsymbol{v}) - p(\boldsymbol{x}, \boldsymbol{v}) \leq -l(\boldsymbol{x}, \boldsymbol{\kappa}_{\text{PSC}}(\boldsymbol{x}, \boldsymbol{v}), \boldsymbol{v}). \tag{13}$$

Assumption 4: For the per-stage weighting in (7a): $l(\bar{\boldsymbol{x}}_v, \bar{\boldsymbol{u}}_v, \boldsymbol{v}) = 0$ and $l(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{v}) \geq \gamma(||\boldsymbol{x} - \bar{\boldsymbol{x}}_v||),\ \forall (\boldsymbol{x}, \boldsymbol{v}) \in \mathcal{F}_N^{\mathcal{T}}$ and admissible $\boldsymbol{u}$, where $\gamma : [0, \infty) \mapsto [0, \infty)$ is continuous, $\gamma(t) > 0\ \forall t > 0$, and $\lim_{t \to \infty} \gamma(t) = \infty$.

Assumption 5: Sets $\mathcal{Y}$ and $\mathcal{T}$ contain the equilibrium $\bar{\boldsymbol{x}}_v$.

Assumption 6: If $\mathcal{Y}$ contains obstacle avoidance constraints, we assume the obstacles are static, can be represented with or encapsulated in convex primitive shapes such as ellipses, can be added/removed from $\mathcal{Y}$ only at the boundary of an invariant sensing region (i.e., the dynamics (1a) should never leave it), and are sufficiently separated such that the robot's convex bounded geometry always fits between pairs of obstacles.

The MPC feedback policies $\boldsymbol{\kappa}_{\text{MPC}}^{\mathcal{T}} : \mathcal{F}_N^{\mathcal{T}} \mapsto \mathbb{R}^{n_u}$ and $\boldsymbol{\kappa}_{\text{MPC}} : \mathcal{F}_N \mapsto \mathbb{R}^{n_u}$ are defined as

$$\boldsymbol{u}(k) \triangleq \boldsymbol{\kappa}_{\text{MPC}}^{\mathcal{T}}(\boldsymbol{x}(k), \boldsymbol{v}(k)) \triangleq \boldsymbol{\mu}_0^{\mathcal{T}*}(\boldsymbol{x}(k), \boldsymbol{v}(k)), \tag{14a}$$

$$\boldsymbol{u}(k) \triangleq \boldsymbol{\kappa}_{\text{MPC}}(\boldsymbol{x}(k), \boldsymbol{v}(k)) \triangleq \boldsymbol{\mu}_0^{*}(\boldsymbol{x}(k), \boldsymbol{v}(k)), \tag{14b}$$

where the minimizer of (7), including the terminal set constraint (8), is denoted as $\boldsymbol{\mu}^{\mathcal{T}*}(\boldsymbol{x}(k), \boldsymbol{v}(k)) = [\boldsymbol{\mu}_0^{\mathcal{T}*}(\boldsymbol{x}(k), \boldsymbol{v}(k))^\mathsf{T}, \ldots, \boldsymbol{\mu}_{N-1}^{\mathcal{T}*}(\boldsymbol{x}(k), \boldsymbol{v}(k))^\mathsf{T}]^\mathsf{T}$, and where the corresponding optimal state sequence is $\boldsymbol{\xi}^{\mathcal{T}*}(\boldsymbol{x}(k), \boldsymbol{v}(k)) = [\boldsymbol{\xi}_0^{\mathcal{T}*}(\boldsymbol{x}(k), \boldsymbol{v}(k))^\mathsf{T}, \ldots, \boldsymbol{\xi}_N^{\mathcal{T}*}(\boldsymbol{x}(k), \boldsymbol{v}(k))^\mathsf{T}]^\mathsf{T}$. We use the same notation but exclude the superscript $^{\mathcal{T}}$ in $\boldsymbol{\mu}^*(\boldsymbol{x}(k), \boldsymbol{v}(k))$ and $\boldsymbol{\xi}^*(\boldsymbol{x}(k), \boldsymbol{v}(k))$ to define the minimizer of (7) excluding the terminal set constraint (8) and the optimal state sequence.

It is only possible to compute a control action from (7)-(8) if it admits a non-empty solution $(\boldsymbol{\mu}^*, \boldsymbol{\xi}^*)$. The set of all parameters for which this is possible is

$$\mathcal{F}_N^{\mathcal{T}} = \{(\boldsymbol{x}, \boldsymbol{v}) | \ \exists\, (\boldsymbol{\mu}, \boldsymbol{\xi}) : (7\text{b}) - (8)\}, \tag{15a}$$

which is the MPC's feasible set or N-step backward reachable set of $\mathcal{T}$. Similarly, we define $\mathcal{F}_N$ as a $\mathcal{T}$-dependent subset of $\mathcal{F}_N^{\mathcal{T}}$ for OCP (7) in the absence of terminal constraint (8) as

$$\mathcal{F}_N = \{(\boldsymbol{x}, \boldsymbol{v}) \mid (\boldsymbol{\xi}_N^*(\boldsymbol{x}, \boldsymbol{v}), \boldsymbol{v}) \in \mathcal{T}\} \subset \mathcal{F}_N^{\mathcal{T}}. \tag{15b}$$

The following theorem summarizes the properties of the closed-loop system by NMPC for a constant reference $\boldsymbol{v}$.

Theorem 1: Consider system (1) controlled by receding-horizon NMPC law (14a). Let above Assumptions 1, 2, 3, 4, 5, and 6, (i.e., sufficient conditions for stability) hold and let $\boldsymbol{\phi}(k, \boldsymbol{x}(0), \boldsymbol{v}) \in \mathbb{R}^{n_x}$ denote the solution at timestep $k \geq 0$ of the closed-loop dynamics $\boldsymbol{x}(k+1) = \boldsymbol{f}(\boldsymbol{x}(k), \boldsymbol{\kappa}_{\text{MPC}}^{\mathcal{T}}(\boldsymbol{x}(k), \boldsymbol{v}))$ with initial condition $\boldsymbol{x}(0) = \boldsymbol{x}_0$ and for a *constant* desired reference $\boldsymbol{v} \in \mathcal{V}_\delta$, $\delta \in (0, 1)$. Then the set of feasible states is positively invariant for the closed-loop system, i.e., $\forall (\boldsymbol{x}(0), \boldsymbol{v}) \in \mathcal{F}_N^{\mathcal{T}}$: $(\boldsymbol{\phi}(k, \boldsymbol{x}(0), \boldsymbol{v}), \boldsymbol{v}) \in \mathcal{F}_N^{\mathcal{T}},\ \forall k \geq 0,\ \boldsymbol{y}(k) \in \mathcal{Y},\ \forall k \geq 0$. The equilibrium point $\bar{\boldsymbol{x}}_v$ is globally attractive in $\mathcal{F}_N^{\mathcal{T}}$. If additionally, $(\bar{\boldsymbol{x}}_v, \boldsymbol{v}) \in \text{Int}\mathcal{F}_N^{\mathcal{T}}$, and the optimal objective function (7a) is continuous in some neighborhood of $\bar{\boldsymbol{x}}_v$, then $\bar{\boldsymbol{x}}_v$ is an asymptotically stable equilibrium point in $\mathcal{F}_N^{\mathcal{T}}$, i.e., $\lim_{k \to \infty} \boldsymbol{\phi}(k, \boldsymbol{x}(0), \boldsymbol{v}) = \bar{\boldsymbol{x}}_v$.

Proof: see [1, Theorem 4.4.2]. ■

Note that Theorem 1 also holds when replacing $\mathcal{F}_N^{\mathcal{T}}$ by $\mathcal{F}_N$ and (14a) by (14b). Though, Theorem 1 achieves the control objectives assuming there exists a $\boldsymbol{v}$ and a $\boldsymbol{x}(0)$ satisfying $(\boldsymbol{x}(0), \boldsymbol{v}) \in \mathcal{F}_N^{\mathcal{T}}$. This condition can be checked in case $\mathcal{F}_N^{\mathcal{T}}$ is fully determined before computing the MPC law.

Its main limitation, however, lies in the fact that the OCP (7)-(8) is infeasible if $\boldsymbol{x}(0)$ cannot be steered in at most N steps into the set $\mathcal{X}(\boldsymbol{v}) = \{\boldsymbol{x} \,|\, (\boldsymbol{x}, \boldsymbol{v}) \in \mathcal{T}\}$. Moreover, Theorem 1 assumes a constant reference $\boldsymbol{v}$. However, as explained in Section III, our approach relies on safely manipulating the auxiliary reference $\boldsymbol{v}(k)$ towards the desired reference $\boldsymbol{r}(k)$. In general, both references are piecewise-continuous time-varying signals. Hence, the infeasibility problem can arise $\forall k \geq 0$, typically when the reference is changed too abruptly. Although increasing the MPC prediction horizon N may seem like a suitable workaround to increase the size of $\mathcal{F}_N^{\mathcal{T}}$, this solution may be inapplicable in practice since the computational time required to solve OCP (7)-(8) scales unfavorably with N and may exceed the desired sampling period, especially for small-scale nonlinear constrained systems with fast dynamics.

Another limitation of Theorem 1 is that for application it explicitly assumes the knowledge of the feasible set $\mathcal{F}_N^{\mathcal{T}}$ associated to the MPC's OCP (7)-(8) to ensure a feasible solution is computed for the couple $(\boldsymbol{x}(0), \boldsymbol{v})$. As discussed in Section I-A there exist methods to compute this set offline, but only for linear systems with a priori known constraints as it suffers from the curse of dimensionality. Hence it complicates the consideration of nonlinear systems and non-convex obstacle constraints, which are added/removed on-the-fly when new obstacles are detected with limited-range sensors.

Finally, the last limitation of Theorem 1 is related to the requirement of an invariant terminal set $\mathcal{T}$ in the formulation of OCP (7). Although this very general formulation ensures recursive feasibility under the conditions of Theorem 1, this formulation is not very practical from a design perspective as it requires the control designer to construct a constraint admissible invariant terminal set $\mathcal{T}$ of the terminal law $\boldsymbol{\kappa}_{\text{PSC}}$. As mentioned in Section I-A, it is possible to compute $\mathcal{T} \subseteq \mathcal{O}_\infty$ for some classes of nonlinear systems, but is even more expensive than for linear systems. Hence, the close to the exact determination of $\mathcal{O}_\infty$ is complicated by the nonlinearity of the dynamics and the nonconvexity of constraints and requires all the constraints to be known before run-time.

Next, we describe an add-on FG unit that ensures feasibility (constraint satisfaction of the FG-NMPC scheme) for *any* piecewise-continuous desired reference $\boldsymbol{r}(k)$, that may be constraint inadmissible (infeasible) for the system dynamics or equilibrium manifold (statics) under ungoverned (direct) tracking of that infeasible reference. We accomplish this by using NMPC law (14b) instead of (14a), without extending the OCP's prediction horizon or decision variables, nor by modifying the OCP. Moreover, we show that the FG does not require expensive offline computations to construct an explicit representation of the feasible set $\mathcal{F}_N^{\mathcal{T}}$ (or its subset $\mathcal{F}_N$) as in [12], nor online recomputations of a terminal set $\mathcal{T} \subseteq \mathcal{O}_\infty$, when new detected constraints (e.g., for collision avoidance) need to be considered and/or removed on-the-fly. Hence, the FG exhibits no offline and very low online computational cost.

V. Feasibility Governor Law

The NMPC feedback policy (14b) is well-defined if the terminal set $\mathcal{T}$ associated with the target equilibrium $\bar{\boldsymbol{x}}_{r(k)}$ is N-step reachable from the current state $\boldsymbol{x}(k)$. Intuitively, this limitation can be overcome by selecting a sequence of intermediate targets that are pairwise reachable. This paper formalizes this idea by defining the auxiliary reference $\boldsymbol{v}(k)$ as a time-varying system to ensure that $(\boldsymbol{x}(k), \boldsymbol{v}(k)) \in \mathcal{F}_N$, $\forall k \in \mathbb{N}$, and that $\boldsymbol{m}_z(\boldsymbol{v}(k)) = \boldsymbol{r}(k)$ for sufficiently large $k \in \mathbb{N}$ if the target is kept constant and steady-state admissible, i.e., $\boldsymbol{r}(k) = \boldsymbol{r} \in \mathcal{R}_\delta$. Moreover, we will show the objectives can be achieved in a computationally efficient way without requiring explicit knowledge of the feasible set $\mathcal{F}_N$, nor of the terminal set $\mathcal{T}$, but by exploiting the terminal closed-loop trajectory emerging from the NMPC's terminal state $\boldsymbol{\xi}_{(N|k)}$ and a terminal energy ingredient at the end of this trajectory.

We will make use of the Padded One-Step Shifted (POSS) operation, which is defined as follows. Using a feasible (not necessarily optimal) MPC state and input trajectory from the previous time step, i.e., $\boldsymbol{\xi}(i|k-1)\,\forall i \in \mathbb{N}_{[0,N]}$ and $\boldsymbol{\mu}(i|k-1)$, $\forall i \in \mathbb{N}_{[0,N-1]}$, the POSS operation for the states, i.e., $\boldsymbol{\xi}(i|k) = \text{POSS}(\boldsymbol{\xi}(i|k-1), \boldsymbol{v}(k-1))$, is defined as

$$\begin{aligned} \text{discard} \quad & \boldsymbol{\xi}(0|k-1) \\ \boldsymbol{\xi}(0|k) = \; & \boldsymbol{\xi}(1|k-1) \\ & \vdots \\ \boldsymbol{\xi}(N-1|k) = \; & \boldsymbol{\xi}(N|k-1) \\ \text{pad } \boldsymbol{\xi}(N|k) = \; & \boldsymbol{f}_{\text{PS}}(\boldsymbol{\xi}(N|k-1), \boldsymbol{v}(k-1)), \end{aligned} \tag{16}$$

and the POSS operation for the inputs, i.e., $\boldsymbol{\mu}(i|k) = \text{POSS}(\boldsymbol{\mu}(i|k-1), \boldsymbol{\xi}(N|k-1), \boldsymbol{v}(k-1))$, is defined as

$$\begin{aligned} \text{discard} \quad & \boldsymbol{\mu}(0|k-1) \\ \boldsymbol{\mu}(0|k) = \; & \boldsymbol{\mu}(1|k-1) \\ & \vdots \\ \boldsymbol{\mu}(N-2|k) = \; & \boldsymbol{\mu}(N-1|k-1) \\ \text{pad } \boldsymbol{\mu}(N-1|k) = \; & \boldsymbol{\kappa}_{\text{PSC}}(\boldsymbol{\xi}(N|k-1), \boldsymbol{v}(k-1)). \end{aligned} \tag{17}$$

When the auxiliary reference is kept constant in between two decision cycles, i.e., $\boldsymbol{v}(k) = \boldsymbol{v}(k-1)$, the POSS operation shifts the MPC trajectory to one that is feasible at the next time step. Note that if the original trajectory was optimal (denoted with *), the shifted trajectory is not necessarily optimal anymore.

A. Feasibility Governor Design

The idea behind the proposed FG is to continuously modify the auxiliary reference $\boldsymbol{v}(k)$ to ensure that the terminal set $\mathcal{T}$ remains reachable for the solution trajectory of OCP (7) without imposing (8).

We draw inspiration from the RG & CG [13], [14] and more specifically the ERG [23] literature, as the ERG scheme is mainly tailored to handle computationally-limited systems with nonlinear dynamics and constraints. As formally stated in [23], the ERG is a continuous-time closed-form (i.e., explicit) feedback policy that ensures in the general case of nonlinear pre-stabilized system dynamics with constraints on states and inputs that: 1) for any target reference $\boldsymbol{r}(t)$ the constraints (2) are always satisfied, and 2) when $\boldsymbol{r}(t) = \boldsymbol{r}$ is kept constant and outside a small $\delta > 0$ region around the boundary of the set of steady-state admissible target references (5), the equilibrium point $\bar{\boldsymbol{x}}_r$ is asymptotically stable.

This paper proposes a novel discrete-time trajectory-based FG policy with Lyapunov-based terminal energy constraint that computes a candidate auxiliary reference, i.e.,

$$\boldsymbol{v}(k) = \boldsymbol{\kappa}_{\text{FG}}\left(\boldsymbol{\xi}\left(N|k\right), \boldsymbol{v}\left(k-1\right), \boldsymbol{r}\left(k\right), \kappa(k)\right), \tag{18}$$

at index k based on the POSS terminal state $\boldsymbol{\xi}(N|k)$, the last feasible auxiliary reference $\boldsymbol{v}(k-1)$, the current target reference $\boldsymbol{r}(k)$, and an auxiliary reference scaling factor $\kappa(k) \in \mathbb{R}^+$. The original continuous-time ERG manipulates the auxiliary reference velocity $\dot{\boldsymbol{v}}(k)$. Hence, in the context of this paper, $\dot{\boldsymbol{v}}(k)$ is the solution of the differential equation

$$\dot{\boldsymbol{v}}(k) = \Delta^+\left(\boldsymbol{\xi}(N|k), \boldsymbol{v}(k-1)\right)\, \boldsymbol{\rho}\left(\boldsymbol{v}(k-1), \boldsymbol{r}(k)\right), \tag{19}$$

where $\Delta^+ = \max(\Delta, 0)$ with $\Delta : \mathbb{R}^{n_x} \times \mathbb{R}^{n_v} \mapsto \mathbb{R}$ denotes the ERG's Dynamic Safety Margin (DSM), and $\boldsymbol{\rho} : \mathbb{R}^{n_v} \times \mathbb{R}^{n_r} \mapsto \mathbb{R}^{n_v}$ denotes the ERG's Navigation Field (NF). The DSM represents a scaled worst-case (i.e., minimal) distance metric to constraint violation of the closed-loop terminal dynamics $\boldsymbol{f}_{\text{PS}}(\boldsymbol{\xi}(N|k), \boldsymbol{v}(k-1))$. The NF defines the direction of the auxiliary reference velocity $\dot{\boldsymbol{v}}(k)$ such that it moves along a path of strictly steady-state admissible equilibria that attracts the auxiliary reference $\boldsymbol{v}(k)$ to the target reference $\boldsymbol{r}(k)$ and pushes $\boldsymbol{v}(k)$ away from steady-state inadmissible regions such as obstacles. Closed-loop safety and stability of the ERG law (19)-(20) was shown in [23], in the continuous-time case, i.e., with constant $\kappa(k) = 1$, $T_s \to 0$, and for the current state $\boldsymbol{x}(k)$ instead of the terminal state $\boldsymbol{\xi}(N|k)$ as it was never combined with an MPC law. In discrete-time and for arbitrarily large values of $\kappa(k)$ and T_s these guarantees do not hold anymore. To integrate the principles of the continuous-time ERG with a discrete-time NMPC law, we extend the standard forward Euler integration of $\dot{\boldsymbol{v}}(k)$ to an integration with adaptive step-size $\kappa(k)T_s$ where $\kappa(k)$ is a time-varying scaling factor (reduced over consecutive samples to ensure constraint satisfaction) and T_s is the FG's sampling time

$$\boldsymbol{v}(k) = \boldsymbol{v}(k-1) + \kappa(k) T_s \dot{\boldsymbol{v}}(k). \tag{20}$$

Next, we describe how the DSM and NF can be computed.

1) Dynamic Safety Margin: We denote the $N_{\text{FG}} = M - N$ (see caption of Fig. 1b) number of predicted outputs at sample time k as $\boldsymbol{y}(i|k) = \boldsymbol{g}(\boldsymbol{x}(i|k), \boldsymbol{u}(i|k))$ and $\boldsymbol{u}(i|k) = \boldsymbol{\kappa}_{\text{PSC}}(\boldsymbol{x}(i|k), \boldsymbol{v}(k-1))$ with $i \in \mathbb{N}_{[1,N_{\text{FG}}]}$. The DSM Δ is a positively scaled minimal distance metric to predicted constraint violation over an ∞-horizon and is computed as

$$\Delta(\boldsymbol{\xi}_{N|k}, \boldsymbol{v}(k-1)) = \min\{\Delta_{\text{traj}}, \Delta_{\text{TE}}\} \text{ with:} \tag{21a}$$
$$\boldsymbol{x}(1|k) = \boldsymbol{\xi}_{N|k} \tag{21b}$$
$$\boldsymbol{x}(i+1|k) = \boldsymbol{f}_{\text{PS}}(\boldsymbol{x}(i|k), \boldsymbol{v}(k-1)) \tag{21c}$$
$$\Delta_{\text{traj}} = \min_{i \in \mathbb{N}_{[1,N_{\text{FG}}-1]}} \{\text{sdist}\left(\boldsymbol{y}(i|k), \partial\mathcal{Y}\right)\} \tag{21d}$$
$$\Delta_{\text{TE}} = \text{sdist}\left(E(\boldsymbol{x}_{(N_{\text{FG}}|k)}, \boldsymbol{v}_{(k-1)}), E_{\max}(\boldsymbol{v}_{(k-1)})\right), \tag{21e}$$

with Δ_{traj} and Δ_{TE} the $N_{\text{FG}} - 1$-truncated trajectory-based DSM and terminal energy DSM, sdist a free-to-select positively scaled distance metric[5] between the explicitly predicted terminal output trajectory $\boldsymbol{y}(i|k)$ and the boundary of the constraint set $\partial\mathcal{Y}$ (i.e., the worst-case over all the constraints), that if negative implies constraint violation. In (21e), $E : (\mathbb{R}^n, \mathbb{R}^v) \mapsto \mathbb{R}$ denotes a general Lyapunov function, and $E_{\max}(\boldsymbol{v}(k-1))$ the Lyapunov threshold value that limits the maximal closed-loop energy of the terminal dynamics (10) at the FG's predicted terminal state $\boldsymbol{x}_{(N_{\text{FG}}|k)} = \boldsymbol{\xi}_{(M|k)}$ to ensure ∞-horizon stability and safety of the FG's trajectory originating from $\boldsymbol{x}_{(N_{\text{FG}}|k)}$.

Note that the very computationally efficient DSM formulations based on Lyapunov, invariant, and returnable sets (see [23, Figure 3]), which by the use of such sets can provide infinite horizon guarantees, are less generically applicable than the trajectory-based DSM. The former can still be applied to a broad set of systems but requires more stringent assumptions on the type of dynamics and/or constraints. For example, optimal closed-form solutions to Lyapunov-level set DSM threshold values can be designed for nonlinear systems characterized by: i) ellipsoidal Lyapunov functions subject to linear constraints, ii) spherical Lyapunov functions subject to distance constraints. Suboptimal solutions exist for the following cases: i) (non-)ellipsoidal Lyapunov functions and (non)linear constraint, and ii) generic Lyapunov functions and distance constraints. For linear systems, the Lyapunov matrix can be selected in a way that ensures good performance. For a nonlinear system with a Lyapunov function/level-set lower-bounded by a class-$\mathcal{K}$ function $\beta : [0, \mathbb{R}^+) \mapsto [0, \infty)$ (e.g., simple quadratic form/ellipsoidal set as depicted in Fig. 1b),

$$\beta(||\boldsymbol{x} - \boldsymbol{m}_x(\boldsymbol{v})||_{P(v)}) \leq E(\boldsymbol{x}, \boldsymbol{v}),\, \forall(\boldsymbol{x}, \boldsymbol{v}), \tag{22}$$

and subject to (locally) linear constraints $\boldsymbol{c}(\boldsymbol{x}, \boldsymbol{v}) = \boldsymbol{a}^T(\boldsymbol{v})\boldsymbol{x} + \boldsymbol{b}(\boldsymbol{v}) \geq 0$, the DSM threshold value can be explicitly computed via [23, (S10) or (S12)]. Due to the reference dependence, one can apply this even to nonconvex obstacle constraints where the obstacles are (enclosed in) ellipsoids.

Due to the predictive nature of the trajectory-based DSM (21d), it is more computationally expensive but therefore exhibits superior performance with respect to the aforementioned set-based DSMs that can be used to limit the terminal closed-loop energy (21e) without requiring explicit trajectory predictions. The set-based DSM ensures safety from the end of the FG's prediction horizon till ∞. Although this makes a more conservative terminal set condition than what is typically used in NMPC, the performance degradation is mitigated by the fact that its use is delayed to the end of the truncated FG horizon. As the NMPC is much more computationally expensive than the trajectory-based ERG, we can choose almost for free $N_{\text{FG}} \gg N$. In other words, it is as if the FG's terminal state (not to be confused with that of the NMPC) always ends up already very close to the auxiliary reference.

2) Navigation Field: The ERG's NF $\boldsymbol{\rho}$ represents a motion direction for $\dot{\boldsymbol{v}}(k)$, addressing the kinematic problem of finding a steady-state admissible path that connects the current auxiliary reference $\boldsymbol{v}(k)$ to the target $\boldsymbol{r}(k)$. Since the auxiliary reference dynamics is a first-order system (meaning it has no inertia), the NF enforces constraints for $\boldsymbol{v}(k)$. The DSM is then responsible for ensuring that the higher-order

[5]The most common choice of sdist$(\boldsymbol{a}, \boldsymbol{b})$ is $k||\boldsymbol{a} - \boldsymbol{b}||, k > 0$, $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^n$ and $n > 0$, which is also used in this paper.

transient dynamics of the system stay sufficiently close to the reference path. We use standard attractive and repulsive vector fields for first-order dynamics that asymptotically steer $\boldsymbol{v}$ towards $\boldsymbol{r}$ without exiting the constraints $(1-\delta)\mathcal{Y}$, such that $\boldsymbol{v}(k) \in \mathcal{V}_\delta$, $\forall k \geq 0$. For more details on designing the NF, please see [23, p. 102]. For the case of a spherical agent with radius $R_\mathrm{a} > 0$ in a 2D environment with $N_\mathrm{o} \in \mathbb{N}^{>0}$ spherical obstacles of radius $R_{\mathrm{o}_j} > 0$ centered in $\boldsymbol{o}_j \in \mathbb{R}^2$, $\boldsymbol{\rho}$ is the sum of an attractive term $\boldsymbol{\rho}^{\text{att}}$ pointing from $\boldsymbol{v}$ to $\boldsymbol{r}$, conservative terms in $\boldsymbol{\rho}^{\text{o, co}}$ that repel $\boldsymbol{v}$ with bounded linearly-varying amplitudes from nearby obstacles, and non-conservative terms in $\boldsymbol{\rho}^{\text{o, n-co}}$ that circulate $\boldsymbol{v}$ tangentially around nearby obstacles,

$$\boldsymbol{\rho}(\boldsymbol{v}, \boldsymbol{r}) = \boldsymbol{\rho}^{\text{att}} + \boldsymbol{\rho}^{\text{o, co}} + \boldsymbol{\rho}^{\text{o, n-co}}, \tag{23}$$

$$\boldsymbol{\rho}^{\text{att}} = \frac{\boldsymbol{r} - \boldsymbol{v}}{\max\{\|\boldsymbol{r} - \boldsymbol{v}\|, \eta\}}, \tag{24a}$$

$$\boldsymbol{\rho}^{\text{o, co}} = -\sum_{j=1}^{N_\mathrm{o}} \max\left(\frac{\zeta_{\mathrm{o}_j} - C_j(\boldsymbol{v})}{\zeta_{\mathrm{o}_j} - \delta_{\mathrm{o}_j}}, 0\right) \widehat{(\boldsymbol{o}_j - \boldsymbol{v})}, \tag{24b}$$

$$\boldsymbol{\rho}^{\text{o, n-co}} = \sum_{j=1}^{N_\mathrm{o}} \alpha_{\mathrm{o}_j} H\left(\zeta_{\mathrm{o}_j} - C_j(\boldsymbol{v})\right) \begin{bmatrix} \widehat{\boldsymbol{o}_j[2] - \boldsymbol{v}[2]} \\ -\boldsymbol{o}_j[1] + \boldsymbol{v}[1] \end{bmatrix}, \tag{24c}$$

with $\hat{\boldsymbol{a}}$ the unit vector of $\boldsymbol{a}$, $\boldsymbol{a}[i]$ the i-th entry of $\boldsymbol{a}$, smoothing radius $\eta > 0$ for numerical division when $\boldsymbol{v}$ is close to $\boldsymbol{r}$, influence margins $\zeta_{\mathrm{o}_j} > 0$ and static safety margins $\delta_{\mathrm{o}_j} \in (0, \zeta_{\mathrm{o}_j})$ around the boundaries of the obstacles, $C_j(\boldsymbol{v}) = \|\boldsymbol{v} - \boldsymbol{o}_j\| - (R_{\mathrm{o}_j} + R_\mathrm{a})$, $H(\cdot)$ denoting the Heavyside step function, and circulation gains $\alpha_{\mathrm{o}_j} \in \mathbb{R}$. The sign of α_{o_j} is selected such that the circulation term works in the same minimal distance direction, either clockwise or counterclockwise around obstacles, as the direction of the net attraction plus conservative repulsive field. Note that for improved navigation through an obstacle-cluttered environment in terms of reducing the possibility to stabilize $\boldsymbol{v}$ on saddle points (i.e., local minima of $\boldsymbol{\rho}$ where $\boldsymbol{v}$ gets stuck in between obstacles), the law (24) can be slightly adapted by not summing the repulsion and circulation terms over all (nearby) obstacles, but by only considering the term of the worst-case obstacle j^*:

$$\frac{\zeta_{\mathrm{o}_{j^*}} - C_{j^*}(\boldsymbol{v})}{\zeta_{\mathrm{o}_{j^*}} - \delta_{\mathrm{o}_{j^*}}} > \frac{\zeta_{\mathrm{o}_j} - C_j(\boldsymbol{v})}{\zeta_{\mathrm{o}_j} - \delta_{\mathrm{o}_j}}, \forall j \neq j^* \in \mathbb{N}_{[1, N_\mathrm{o}]}. \tag{25}$$

B. Combined FG and NMPC Algorithm

The pseudo-code of the combined FG and NMPC laws at sample index k is stated in Algorithm 1 and a block diagram is displayed in Fig. 2. In what follows, a summary is given first and then each phase of the algorithm is detailed.

At each sampling instant, the algorithm requires a state measurement $\boldsymbol{x}(k)$, a possibly time-varying target reference $\boldsymbol{r}(k)$, the previously computed feasible auxiliary reference $\boldsymbol{v}(k-1)$, the POSS NMPC's feasible (not necessarily optimal) state $\boldsymbol{\xi}(i|k)$, $\forall i \in \mathbb{N}_{[0,N]}$, and input $\boldsymbol{\mu}(i|k)$, $\forall i \in \mathbb{N}_{[0,N-1]}$ trajectories, and the auxiliary reference scaling factor $\kappa(k)$.

No maximization of $\kappa(k)$ will be computed in a single control cycle, but a guess of $\kappa(k)$ will be used to compute a candidate auxiliary reference which will be checked for static and dynamic feasibility as depicted in Fig. 2. In case the guessed $\kappa(k)$ leads to infeasibility, $\kappa(k)$ is reduced while

Algorithm 1: Combined FG & NMPC laws:

Input : measured state $\boldsymbol{x}(k)$
desired target reference $\boldsymbol{r}(k)$
feasible auxiliary reference $\boldsymbol{v}(k-1)$
POSS NMPC feasible states $\boldsymbol{\xi}_{(i|k)}$
POSS NMPC feasible inputs $\boldsymbol{\mu}_{(i|k)}$
auxiliary reference scaling factor $\kappa(k)$

Output: guess $\kappa(k+1)$, feasible $\boldsymbol{\xi}_{(i|k+1)}$, $\boldsymbol{\mu}_{(i|k+1)}$, $\boldsymbol{u}(k)$ for $\boldsymbol{v}(k) \in \mathcal{V}_\delta \cap \text{Proj}(\mathcal{F}_N, \boldsymbol{x}(k)))$

if $k = 0$ **then**
  $\boldsymbol{v}(k-1) \leftarrow \boldsymbol{m}_x^{-1}(\boldsymbol{x}(0))$;
  $\boldsymbol{v}(k) \leftarrow \boldsymbol{v}(k-1)$;
  $\left(\boldsymbol{\mu}^*_{(i|k)}, \boldsymbol{\xi}^*_{(i|k)}\right) \leftarrow$ solve (7) under (7b)-(7d) ;
  apply $\boldsymbol{u}(k) \leftarrow \boldsymbol{\mu}^*_{(0|k)}$;
  $\boldsymbol{\xi}_{(i|k+1)} \leftarrow \text{POSS}(\boldsymbol{\xi}^*_{(i|k)}, \boldsymbol{v}(k))$;
  $\boldsymbol{\mu}_{(i|k+1)} \leftarrow \text{POSS}(\boldsymbol{\mu}^*_{(i|k)}, \boldsymbol{\xi}^*_{(N|k)}, \boldsymbol{v}(k))$;
  $\kappa(k+1) \leftarrow \kappa_0$;
else
  $\Delta^+ \leftarrow \Delta^+(\boldsymbol{\xi}_{(N|k)}, \boldsymbol{v}(k-1))$;
  $\boldsymbol{\rho} \leftarrow \boldsymbol{\rho}\left(\boldsymbol{v}\left(k-1\right), \boldsymbol{r}\left(k\right)\right)$;
  for $0 < l < \bar{l}$ **do**
    $\dot{\boldsymbol{v}}(k) \leftarrow \Delta^+ \boldsymbol{\rho}$;
    $\boldsymbol{v}(k) \leftarrow \boldsymbol{v}(k-1) + \min\{||\kappa(k)\, T_s \dot{\boldsymbol{v}}(k)||, ||\boldsymbol{r}(k) - \boldsymbol{v}(k-1)||\} \frac{\dot{\boldsymbol{v}}(k)}{\max(||\dot{\boldsymbol{v}}(k)||, \eta)}$;
    if $\boldsymbol{v}(k) \in \mathcal{V}_\delta$ **then**
      break ;
    else
      $\kappa(k) \leftarrow \max\{\lambda \kappa(k), \underline{\kappa}\}$;
  if $\boldsymbol{v}(k) \in \mathcal{V}_\delta$ **then**
    $\left(\boldsymbol{\mu}^*_{(i|k)}, \boldsymbol{\xi}^*_{(i|k)}\right) \leftarrow$ solve (7) under (7b)-(7d) ;
    $\boldsymbol{\xi}^\mathrm{c}_{(i|k+1)} \leftarrow \text{POSS}(\boldsymbol{\xi}^*_{(i|k)}, \boldsymbol{v}(k))$;
    $\boldsymbol{\mu}^\mathrm{c}_{(i|k+1)} \leftarrow \text{POSS}(\boldsymbol{\mu}^*_{(i|k)}, \boldsymbol{\xi}^*_{(N|k)}, \boldsymbol{v}(k))$;
    if $\Delta(\boldsymbol{\xi}^c_{(N|k+1)}, \boldsymbol{v}(k)) \geq 0$ *& line 20 solved* **then**
      apply $\boldsymbol{v}(k)$;
      apply $\boldsymbol{u}(k) \leftarrow \boldsymbol{\mu}^*_{(0|k)}$;
      $\boldsymbol{\xi}_{(i|k+1)} = \boldsymbol{\xi}^\mathrm{c}_{(i|k+1)}$;
      $\boldsymbol{\mu}_{(i|k+1)} = \boldsymbol{\mu}^\mathrm{c}_{(i|k+1)}$;
      $\kappa(k+1) \leftarrow \min\{\frac{1}{\lambda}\kappa(k), \overline{\kappa}\}$;
    else
      apply $\boldsymbol{v}(k) \leftarrow \boldsymbol{v}(k-1)$;
      apply $\boldsymbol{u}(k) \leftarrow \boldsymbol{\mu}_{(0|k)} + \boldsymbol{K}(\boldsymbol{\xi}_{(0|k)} - \boldsymbol{x}(k))$;
      $\boldsymbol{\xi}_{(i|k+1)} \leftarrow \text{POSS}(\boldsymbol{\xi}_{(i|k)}, \boldsymbol{v}(k))$;
      $\boldsymbol{\mu}_{(i|k+1)} \leftarrow \text{POSS}(\boldsymbol{\mu}_{(i|k)}, \boldsymbol{\xi}_{(N|k)}, \boldsymbol{v}(k))$;
      $\kappa(k+1) \leftarrow \max\{\lambda \kappa(k), \underline{\kappa}\}$;
  else
    apply $\boldsymbol{v}(k) \leftarrow \boldsymbol{v}(k-1)$;
    apply $\boldsymbol{u}(k) \leftarrow \boldsymbol{\mu}_{(0|k)} + \boldsymbol{K}(\boldsymbol{\xi}_{(0|k)} - \boldsymbol{x}(k))$;
    $\boldsymbol{\xi}_{(i|k+1)} \leftarrow \text{POSS}(\boldsymbol{\xi}_{(i|k)}, \boldsymbol{v}(k))$;
    $\boldsymbol{\mu}_{(i|k+1)} \leftarrow \text{POSS}(\boldsymbol{\mu}_{(i|k)}, \boldsymbol{\xi}_{(N|k)}, \boldsymbol{v}(k))$;

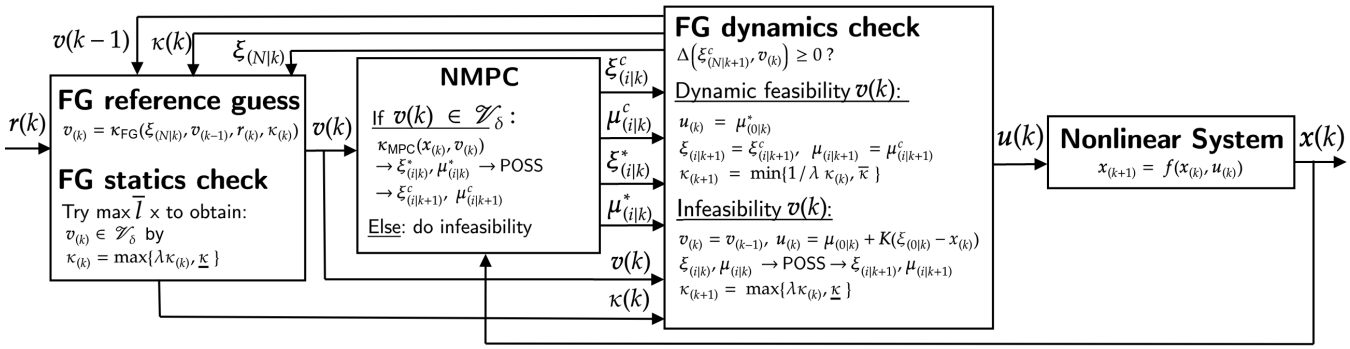


Fig. 2: A block diagram overview of the closed-loop FG & NMPC architecture and the main symbols defined throughout this paper. The scheme is composed of the following blocks: the FG's auxiliary reference guesser and steady-state (i.e., static) admissibility checker, the NMPC law without terminal set constraint, the FG's dynamic feasibility checker, and the nonlinear constrained dynamical system.

the last feasible auxiliary reference and input remain applied. In between the static and dynamic feasibility checks, the algorithm solves OCP (7) without terminal set constraint (8) for arbitrary short N and for pairs $(\boldsymbol{x}(k), \boldsymbol{v}(k)) \in \mathcal{F}_N$ as depicted in Fig. 1b, without the need to explicitly construct $\mathcal{F}_N$. It returns a new guess for $\kappa(k+1)$ and feasible NMPC sequences $\boldsymbol{\xi}_{(i|k+1)}$, $\boldsymbol{\mu}_{(i|k+1)}$, and a feasible control input $\boldsymbol{u}(k)$ for a feasible auxiliary reference $\boldsymbol{v}(k)$ that ensures asymptotic stability and infinite-horizon constraint satisfaction for the combined FG & NMPC state and output trajectories going from the current state $\boldsymbol{x}(k)$ and output $\boldsymbol{y}(k)$ towards the closed-loop equilibrium point $\bar{\boldsymbol{x}}_v(k)$ and $\bar{\boldsymbol{y}}_v(k)$.

In the initialization phase of Algorithm 1 at sampling index $k = 0$ (line 1-8), the first auxiliary reference $\boldsymbol{v}(0)$ is selected as the steady-state projection of the initial state, i.e., $\boldsymbol{m}_x^{-1}(\boldsymbol{x}(0))$ (line 2-3). Though, any feasible $\boldsymbol{v}(0)$ can be selected. Subsequently, the NMPC law is computed for a state-auxiliary reference pair $(\boldsymbol{x}(0), \boldsymbol{v}(0))$ and the optimal and feasible control input is applied (line 4-5). Under the assumption[6] that the velocity sub-vector of $\boldsymbol{x}(0)$ is close to zero, it is ensured that an optimal (and feasible) solution trajectory is found with a terminal state $\boldsymbol{\xi}(N)$ near the closed-loop equilibrium $\bar{\boldsymbol{x}}_{v(0)}$. Then, the optimal input and state trajectories undergo a POSS operation (line 6-7) and the auxiliary reference scaling factor is initialized (line 8) as a preparation for using them at the next sampling instant.

In the main routine of Algorithm 1, when $k > 0$ (line 9-39), the main purpose is to compute a feasible and preferably optimal control input $\boldsymbol{u}(k)$ and select a suitable $\boldsymbol{v}(k) \in \mathcal{V}_\delta$ for which the pair $(\boldsymbol{x}(k), \boldsymbol{v}(k)) \in \mathcal{F}_N$. This process is composed of four parts: the auxiliary reference guess, the steady-state/statics admissibility check, the NMPC computation, and the dynamic feasibility check.

First, the auxiliary reference guess and the steady-state admissibility check are performed together. The DSM (21) and the NF (23) are computed (line 10-11) and multiplied to obtain an auxiliary reference velocity (19), which is subsequently integrated with an adaptive step-size $\kappa(k)\, T_s$ as in (20) (though different as explained later) to obtain an auxiliary reference guess (line 13-14). This process is iterated at most $\bar{l}$ times and terminated in case a steady-state admissible auxiliary reference is found (line 15-16). As long as it is steady-state inadmissible during this controller time step, i.e., $\boldsymbol{v}(k) \notin \mathcal{V}_\delta$, the integration step is decreased by exponentially reducing $\kappa(k)$ with a reduction rate $(1-\lambda)$ with $\lambda \in (0,1)$ (line 17-18). This means that the new guess $\boldsymbol{v}(k)$ is moved back towards the last feasible $\boldsymbol{v}(k-1)$ along the direction of the NF $\boldsymbol{\rho}(k)$.

In case a steady-state admissible guess $\boldsymbol{v}(k) \in \mathcal{V}_\delta$ is obtained after at most $\bar{l}$ iterations, the goal is to find out if it is also dynamically feasible (ensuring both stability and safety) over an infinite horizon (line 19-34). Therefore, the algorithm first computes the NMPC law under the auxiliary reference $\boldsymbol{v}(k)$ without considering the terminal constraints (line 20). Subsequently, if optimal trajectories are returned, they are shifted into candidate trajectories (line 21-22) for the dynamics feasibility check. The dynamics check (line 23) involves computing the $N_{\text{FG}} - 1$-horizon state and output trajectory predictions of the terminal closed-loop dynamics emerging from the POSS terminal state $\boldsymbol{\xi}^c(N|k+1)$ and subsequently checking if these outputs satisfy constraints over an infinite horizon towards $\bar{\boldsymbol{x}}_v$. This is equivalent to checking if the DSM (21) associated with the auxiliary reference guess $\boldsymbol{v}(k)$, which is the worst-case of the trajectory-based DSM (21d) and the terminal energy DSM (21e), is non-negative.

If the dynamics check succeeds, the algorithm applies the feasible auxiliary reference guess $\boldsymbol{v}(k)$ (line 24) and the optimal (and feasible) control input $\boldsymbol{u}(k)$ (line 25). The candidate POSS trajectories from the NMPC are stored as backup solutions for the next time step (line 26-27) and the auxiliary reference scaling factor $\kappa(k)$ is increased with a growth rate of $(1-\lambda)$ (line 28). When the new candidate $\boldsymbol{v}(k)$ is checked to be feasible at the next sampling time, this can speed up convergence.

The dynamics check fails (line 29) if one of the following conditions holds: i) the DSM is negative, i.e., implying that the auxiliary reference guess $\boldsymbol{v}(k)$ results in constraint violation(s) of the terminal control law and/or that the terminal state of the FG did not reach sufficiently close to the equilibrium $\bar{\boldsymbol{x}}_v$ as to obtain infinite horizon guarantees, or ii) the OCP (line 20) could not be solved as $(\boldsymbol{x}(k), \boldsymbol{v}(k))$ led to solver difficulties (e.g., reached solver iteration limit). Then, the algorithm falls

[6]This assumption is not stringent as it certainly holds for all systems initialized at equilibrium and for most applications it is not stringent to assure that $||\boldsymbol{x}(0) - \boldsymbol{m}_x(\boldsymbol{v}(0))||$ is small.

back on the last feasible solution by applying the previous $\boldsymbol{v}(k-1)$, discarding the reference guess $\boldsymbol{v}(k)$ (line 30), and by applying the first input of the previously computed POSS input sequence (line 31), locally stabilized with a proportional tracking control law, sacrificing optimality for feasibility. The candidate POSS trajectories from the previous time step undergo a shift and are stored as backup solutions for the next time step (line 32-33). The auxiliary reference scaling factor $\kappa(k)$ is reduced with a reduction rate of $(1-\lambda)$ (line 34).

The same strategy (excluding the $\kappa(k)$ decay that is already done) is employed if no steady-state admissible auxiliary reference is found after at most $\bar{l}$ iterations (line 35-39).

Finally, it is important to note that on line 10 the DSM Δ^+ is a positive distance metric to constraint violation where negative values are saturated to zero due to the presence of the $(\cdot)^+$ operator. A value of zero ensures that the auxiliary reference remains stationary. In contrast, in order to be able to check for dynamic feasibility on line 23, the DSM Δ may not be lower bounded to zero as negative DSM values imply the detection of future constraint violations of the terminal control law and may not be ignored.

C. Theoretical Guarantees

There are some technical details worth mentioning that enable the discrete-time implementation of the standard continuous-time ERG law [23] in combination with NMPC while ensuring recursive feasibility/safety, and asymptotic stability. The main theoretical results of this paper are summarized in Theorem 2 and subsequently proven.

Theorem 2: Consider system (1) that is controlled by the combined FG & NMPC law of Algorithm 1, i.e., $\boldsymbol{u}_{(k)} = \boldsymbol{\kappa}_{\text{FG \& NMPC}}(\boldsymbol{x}_{(k)}, \boldsymbol{r}_{(k)}, \boldsymbol{v}_{(k-1)}, \boldsymbol{\xi}_{(i|k)}, \boldsymbol{\mu}_{(i|k)}, \kappa_{(k)})$. Let the above (often slightly modified) Assumptions 1, 2 (redefining $\mathcal{T} \triangleq \{(\boldsymbol{x},\boldsymbol{v}) \,|\, \Delta(\boldsymbol{x},\boldsymbol{v}) \geq 0$ as per line 23$\}$ which is efficiently computable online), 3 (with $p(\boldsymbol{x},\boldsymbol{v}) \geq 0, \forall (\boldsymbol{x},\boldsymbol{v}) \in \mathbb{R}^{n_x} \times \mathcal{V}_\delta$ and (13) checked in the DSM as per line 23), 4 (with $l(\boldsymbol{x},\boldsymbol{u},\boldsymbol{v}) \geq \gamma(||\boldsymbol{x} - \bar{\boldsymbol{x}}_v||), \forall (\boldsymbol{x},\boldsymbol{v}) \in \mathbb{R}^{n_x} \times \mathcal{V}_\delta$), 5 (only assumed for the initial equilibrium $\bar{\boldsymbol{x}}_{v(0)}$ but not for the desired $\bar{\boldsymbol{x}}_{r(k)}$), and 6 hold. Let $\boldsymbol{\phi}(k, \boldsymbol{x}_{(0)}, \boldsymbol{r}_{(k)}) \in \mathbb{R}^{n_x}$ denote the solution at time step $k \geq 0$ of the closed-loop dynamics $\boldsymbol{x}_{(k+1)} = \boldsymbol{f}(\boldsymbol{x}_{(k)}, \boldsymbol{\kappa}_{\text{FG \& NMPC}}(\boldsymbol{x}_{(k)}, \boldsymbol{r}_{(k)}, \boldsymbol{v}_{(k-1)}, \boldsymbol{\xi}_{(i|k)}, \boldsymbol{\mu}_{(i|k)}, \kappa_{(k)}))$ with initial condition $\boldsymbol{x}_{(0)} = \boldsymbol{x}_0$ and piecewise-continuous desired reference sequence $\boldsymbol{r}_{(k)}$. By adding the FG to the NMPC and obtaining a floating terminal set $\mathcal{T}$ around the auxiliary reference $\boldsymbol{v}_{(k)}$, the original NMPC's feasibility region $\mathcal{F}_N$ is significantly extended to the N-step backward reachable set of *any* target $\boldsymbol{r}(k)$, instead of just the a priori known to be reachable constant target $\boldsymbol{v}$ in Theorem 1). Then the set of feasible states is positively invariant for the closed-loop system, i.e., $\forall (\boldsymbol{x}_{(0)}, \boldsymbol{r}_{(k)}) \in \mathcal{F}_N$: $(\boldsymbol{\phi}(k, \boldsymbol{x}_{(0)}, \boldsymbol{r}_{(k)}), \boldsymbol{r}_{(k)}) \in \mathcal{F}_N, \forall k \geq 0$, $\boldsymbol{y}(k) \in \mathcal{Y}, \forall k \geq 0$. For *constant* $\boldsymbol{r} \in \mathcal{R}_\delta$, $\delta = 0$, the equilibrium point $\bar{\boldsymbol{x}}_r$ is globally attractive in $\mathcal{F}_N \cap (\mathbb{R}^{n_x} \times \mathcal{R}_0)$. If additionally, $(\bar{\boldsymbol{x}}_r, \boldsymbol{r}) \in \text{Int}\mathcal{F}_N \cap (\mathbb{R}^{n_x} \times \mathcal{R}_\delta)$, with $\delta \in (0,1)$, and the optimal objective function (7a) is continuous in some neighborhood of the $\bar{\boldsymbol{x}}_v$ sequence to $\bar{\boldsymbol{x}}_r$, then $\bar{\boldsymbol{x}}_r$ is an asymptotically stable equilibrium point in $\mathcal{F}_N \cap (\mathbb{R}^{n_x} \times \mathcal{R}_\delta)$, i.e., $\lim_{k\to\infty} \boldsymbol{\phi}(k, \boldsymbol{x}_{(0)}, \boldsymbol{r}) = \bar{\boldsymbol{x}}_r$. Instead, if the constant $\boldsymbol{r} \notin \mathcal{R}_\delta$, then $\lim_{k\to\infty} \boldsymbol{\phi}(k, \boldsymbol{x}_{(0)}, \boldsymbol{r}) = \bar{\boldsymbol{x}}_{r^*}$ with $\boldsymbol{r}^* = \arg\min_{\boldsymbol{s}\in\mathcal{R}_\delta} ||\boldsymbol{s} - \boldsymbol{r}||$ the closest steady-state admissible approximation of $\boldsymbol{r}$.

Proof: In general, the proof follows the same reasoning as for the standard NMPC of Theorem 1 and the continuous-time ERG law [23]. However, for Algorithm 1, which combines the discrete-time FG feedback policy (18) with NMPC law (14b), some additional points have to be considered.

1) Recursive feasibility and ∞-horizon safety $\forall \boldsymbol{r}(k)$: At $k = 0$, a feasible solution $\boldsymbol{u}(k)$ and feasible POSS NMPC sequence is guaranteed by construction of $\boldsymbol{v}(k-1)$ as the steady-state map of $\boldsymbol{x}(0)$ in line 2. For $k > 0$, we first consider the case $\boldsymbol{v}(k) \in \mathcal{V}_\delta$ and the NMPC on line 20 is solved. Then the terminal dynamics sequence emanating from the POSS N-step NMPC trajectory is explicitly checked for constraint satisfaction via the FG's DSM, as per line 23, where $\Delta \geq 0$ implies ∞-horizon safety/feasibility for the current $\boldsymbol{v}(k)$, such that the associated feasible input $\boldsymbol{u}(k)$ can be applied. This terminal dynamics feasibility check is initiated from the POSS NMPC terminal state using the NMPC sampling period (i.e., the expected terminal state at the next control cycle under nominal conditions) instead of the current NMPC terminal state. Note that we do not make any assumptions on the size of the NMPC sampling period (used in lines 21 and 22), the NMPC prediction time step (used in line 20), and the FG's DSM prediction time step (used in line 23), which can be different in general. Else, if $\boldsymbol{v}(k) \notin \mathcal{V}_\delta$, or if on line 20 the NMPC OCP solver does not converge (e.g., reached upperbound on the number of solver iterations), or if terminal constraint violations are predicted ($\Delta < 0$) we fall back on the feasible POSS sequence for the frozen auxiliary reference $\boldsymbol{v}(k) = \boldsymbol{v}(k-1)$, sacrificing optimality for feasibility/safety. Algorithm 1 always results in a feasible sequence for the next system's state. Moreover, each of the above cases can be obtained independently on the piecewise-continuous target $\boldsymbol{r}(k)$, as it only affects the auxiliary reference guess $\boldsymbol{v}(k)$ on line 14. Hence recursive feasibility and ∞-horizon safety hold $\forall \boldsymbol{r} \in \mathbb{R}^{n_z}$ (even if $\boldsymbol{r} \notin \mathcal{R}_\delta$, far away, or rapidly changing). Hereby, the feasible set $\mathcal{F}_N$ is enlarged, however, it is not required for computing the policy.

2) Asymptotic stability of $\boldsymbol{x}$ to a constant steady-state-admissible equilibrium $\bar{\boldsymbol{x}}_{r^*}$: We break this down in proving asymptotic stability from $\boldsymbol{x}$ to $\bar{\boldsymbol{x}}_v$ (part 1) and from $\bar{\boldsymbol{x}}_v$ (and hence $\boldsymbol{x}$) to $\bar{\boldsymbol{x}}_{r^*}$ (part 2). Part 1 follows directly from the above feasibility proof, as the FG selects $\boldsymbol{v}(k)$ such that pairs $(\boldsymbol{x}(k), \boldsymbol{v}(k))$ ensure feasibility. Then, by applying Theorem 1, it follows that $\bar{\boldsymbol{x}}_v$ is an asymptotically stable equilibrium point. Part 2 has been proven in [23] for the continuous time ERG law. From the design of the DSM and NF, it follows that the Lyapunov function $W(\boldsymbol{v}, \boldsymbol{r}^*) = \|\boldsymbol{v} - \boldsymbol{r}^*\|$ is monotonically non-increasing, proving asymptotic stability of $\boldsymbol{v} = \boldsymbol{r}^*$. Hence, the auxiliary reference will not stop moving indefinitely before reaching the constant desired reference. For this discrete-time FG, asymptotic convergence still holds as explained below. $\boldsymbol{v}$ could only remain constant indefinitely while $\boldsymbol{r} \neq \boldsymbol{v}$ if there

exists an instant after which the algorithm would result in either at least one of the following cases being satisfied overall consecutive control cycles:

1) $\dot{\boldsymbol{v}}(k) = 0$ on lines 13-14 by A) $\Delta^+ = 0$ or B) $\boldsymbol{\rho} = 0$;
2) $\kappa(k) = 0$ on line 14;
3) $\boldsymbol{v}(k) \notin \mathcal{V}_\delta$ on line 19;
4) line 23 is false by A) $\Delta < 0$ or B) NMPC not solved.

Case 1-A occurs if the DSM predicts terminal constraint violation for the auxiliary reference guess. The first auxiliary reference is initialized to be feasible, hence this condition cannot hold indefinitely due to the stability and constraint enforcement of the NMPC to the current feasible reference. If one waits sufficiently long, the state has safely converged sufficiently close to the frozen auxiliary reference, making the predicted NMPC terminal state and FG trajectories strictly safe such that $\Delta > 0$. This strictly positive DSM implies that $\boldsymbol{v}$ can be perturbed again to move towards $\boldsymbol{r}^*$. Moreover, for improved convergence, the standard continuous-time ERG law of [23] is redesigned (see line 14 that ensures the magnitude of change of the auxiliary reference cannot be larger than the difference between the desired and auxiliary reference) such that under non-negligible sampling periods and large scaling factors (i.e., large values of $\kappa(k)\, T_s$) it prevents the auxiliary reference from overshooting the desired one. This avoids indefinite chattering (that would occur for the standard naively scaled ERG of (20)) around the desired reference when both references are close. **Case 1-B** occurs if the auxiliary reference guess gets stuck in local minima (e.g., overlapping obstacles) or saddle points (e.g., segment $\boldsymbol{v} - r$ goes through the obstacle's center). This is prevented as we assume there is enough space for the robot to pass between obstacle pairs and due to the addition of the non-conservative NF term (24c)-(25). The auxiliary reference circulates the closest obstacle even if the conservative term exactly cancels the attraction term. Note that the used NF is designed for (quasi-)holonomic systems.
Case 2 is never attainable as we assume $\lambda \in (0, 1)$ and $\overline{\kappa} \geq \kappa_0 \geq \underline{\kappa} > 0$, which implies $\overline{\kappa} \geq \kappa(k) \geq \underline{\kappa} > 0$.
Case 3 occurs if the auxiliary reference guess is non-strictly steady-state admissible. This cannot hold indefinitely as the $\kappa(k)$ reductions over a single control cycle at lines 12-18 (with at most $\bar{l}$ iterations per control cycle to speed up convergence) and the $\kappa(k)$ reductions on line 34 over consecutive control cycles make the auxiliary reference guess converge back towards the latest feasible (and strictly steady-state admissible) auxiliary reference. In other words, the candidate auxiliary reference can never remain stuck forever inside obstacles.
Case 4-A cannot hold indefinitely for the same reasons as in Case 1-A. **Case 4-B** neither, as we assume as in the majority of MPC literature, that a sufficiently large number of maximum solver iterations of the SQP routine is available (but used only if needed) to solve the NMPC up to numerical tolerance. ■

Remark 1: The proposed FG-NMPC directly operates on the non-convex static constraint set $\mathcal{Y}$, in which online obstacles are added or removed if they enter or leave the sensing region. The FG explicitly predicts a trajectory of the terminal closed-loop system and checks feasibility for the candidate auxiliary reference. As such, it does not require an explicit representation of the NMPC's terminal set $\mathcal{T} \subseteq \mathcal{O}_\infty$ nor the feasible set $\mathcal{F}_N$. used in QP-based FG schemes [12], [20]. These work under stringent assumptions whereby these sets must be computed before run-time and are limited to low-dimensional linear systems with linear or convex constraints and hence cannot consider obstacle constraints. This makes the proposed FG applicable for safe real-time NMPC of a broad class of constrained nonlinear systems with fast dynamics operating in a priori unknown obstacle-cluttered environments.

Remark 2: The computational complexity of the NMPC is dominated by the constrained nonlinear optimization of the input trajectory over a horizon N (lines 4 and 20) and that of the FG is mostly dominated by the two DSM computations on line 10 and 23. The trajectory-based FG boils down to closed-form PSC trajectory predictions over horizon N_{FG}, distance to constraints computations, and a scalar minimization of this worst-case distance as in (21). However, the cumulative computational cost of two such DSMs, even for large N_{FG}, is significantly smaller than the cost required to solve the OCP for small NMPC horizon N. Moreover, the algorithm does not restrict the designer on the choice of the NMPC horizon length N, which enables scaling the computational cost accordingly. Obtaining infinite-horizon safety guarantees is nontrivial as most NMPC applications on real-world systems ignore terminal set constraints and have no provably feasible backup policy in case the OCP suddenly fails to find a feasible solution trajectory, often with disastrous consequences. Therefore, in order to prevent infeasibility in practice, they rely on heuristically tuning the NMPC cost and selecting a "sufficiently" long NMPC prediction horizon (at the cost of increased computational burden) for the system at hand and the specific reference signal that needs to be tracked.

VI. Hardware-in-the-loop-simulations

We numerically validate the FG & NMPC Algorithm 1 by applying it to the constrained nonlinear dynamics of a planar quadrotor Unmanned Aerial Vehicle (UAV).

The algorithm is implemented in `C++` and all validations are executed in a single thread of a small Intel Core i7-10710U CPU @ 1.10GHz in performance mode, which is commonly used as an onboard UAV companion computer.

The system dynamics can be written in the following form

$$\frac{d}{dt}\begin{bmatrix} y \\ z \\ \phi \\ \dot{y} \\ \dot{z} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \dot{y} \\ \dot{z} \\ \dot{\phi} \\ 0 \\ -g \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ -\frac{1}{m}\sin\phi & 0 \\ \frac{1}{m}\cos\phi & 0 \\ 0 & \frac{1}{J} \end{bmatrix} \begin{bmatrix} T \\ \tau \end{bmatrix}. \tag{26}$$

The UAV has a mass $m = 2.0\,\text{kg}$ and a moment of inertia $J = 0.0417\,\text{kg}\,\text{m}^2$ with respect to its body frame $\mathcal{B}$. It is subject to a gravitational force $-mg\hat{\boldsymbol{z}}_{\mathcal{W}}$ with gravitational acceleration $g \approx 9.81\,\text{m/s}^2$. Its state is $\boldsymbol{x} = [\boldsymbol{p}^\mathsf{T}, \phi, \dot{\boldsymbol{p}}^\mathsf{T}, \dot{\phi}]^\mathsf{T} \in \mathbb{R}^6$ where $\boldsymbol{p} = [y, z]^\mathsf{T} \in \mathbb{R}^2$ and $\dot{\boldsymbol{p}} = [\dot{y}, \dot{z}]^\mathsf{T} \in \mathbb{R}^2$ denote the position and the velocity of $\mathcal{B}$ with respect to the world frame $\mathcal{W}$. The attitude and attitude rate is represented by the roll angle $\phi \in \mathbb{R}$ and roll rate $\dot{\phi} \in \mathbb{R}$. The control input is represented

by $\boldsymbol{u} = [T, \tau]^\mathsf{T} \in \mathbb{R}^2$ with a unidirectional total thrust force $T \in \mathbb{R}^+$ in the $\hat{\boldsymbol{z}}_\mathcal{B}$ direction, and a torque $\tau \in \mathbb{R}$ about $\hat{\boldsymbol{x}}_\mathcal{B}$.

The UAV is subject to actuator saturation constraints $0 \leq T \leq T_{\max}$ and $\tau_{\min} \leq \tau \leq \tau_{\max}$, to velocity constraints $|\dot{y}| \leq \dot{y}_{\max}$, and to collision avoidance constraints with $N_\mathrm{o} = 15$ circular obstacles (assuming a large sensing radius such that all are simultaneously considered) with radii R_{o_j} centered in $\boldsymbol{o}_j = [y_{\mathrm{o}_j}, z_{\mathrm{o}_j}]^\mathsf{T} \in \mathbb{R}^2$, $\forall j \in \{1, \ldots, N_\mathrm{o}\}$ as depicted in Fig. 3 and Fig. 4. The UAV has a collision radius of $R_{\mathrm{UAV}} = 0.375\mathrm{m}$. Hence, all constraints can be expressed on the output, i.e.,

$$\boldsymbol{y} = \begin{bmatrix} T \\ \tau \\ \dot{y} \\ \|\boldsymbol{p} - \boldsymbol{o}_1\| - R_{\mathrm{o}_1} - R_{\mathrm{UAV}} \\ \cdots \\ \|\boldsymbol{p} - \boldsymbol{o}_{N_\mathrm{o}}\| - R_{\mathrm{o}_{N_\mathrm{o}}} - R_{\mathrm{UAV}} \end{bmatrix}. \tag{27}$$

As such, the constraint set $\mathcal{Y} = [0\mathrm{N}, 40\mathrm{N}] \times [-5\mathrm{Nm}, 5\mathrm{Nm}] \times [-3\mathrm{m/s}, 3\mathrm{m/s}] \times [0\mathrm{m}, \infty\mathrm{m}) \times \ldots \times [0\mathrm{m}, \infty\mathrm{m})$.

The time-varying target and auxiliary reference position signals are denoted by $\boldsymbol{r} = [r_y, r_z]^T \in \mathbb{R}^2$ and $\boldsymbol{v} = [v_y, v_z]^T \in \mathbb{R}^2$. The target $\boldsymbol{r}$ consists of a sequence of three distant waypoints depicted by the green squares in Fig. 3 and Fig. 4. The UAV's initial condition is always at $\boldsymbol{x}_0 = \boldsymbol{0}_{6\times 1}$ and the first auxiliary reference $\boldsymbol{v}(0)$ is selected as the steady-state projection of the initial state, i.e., $\boldsymbol{m}_x^{-1}(\boldsymbol{x}(0))$. This ensures in the initialization phase of Algorithm 1 (line 1-8) that the NMPC law computed for the state-auxiliary reference pair $(\boldsymbol{x}(0), \boldsymbol{v}(0))$ certainly returns a feasible $(\boldsymbol{\mu}^*_{(i|0)}, \boldsymbol{\xi}^*_{(i|0)})$ by solving (7) under (7b)-(7d).

For what concerns the NMPC law, we solve OCP (7) with standard quadratic terminal and stage costs, i.e., $||\boldsymbol{\xi}_N - \bar{\boldsymbol{x}}_{v(k)}||^2_P + \sum_{i=0}^{N-1} ||\boldsymbol{\xi}_i - \bar{\boldsymbol{x}}_{v(k)}||^2_Q + ||\boldsymbol{\mu}_i - \bar{\boldsymbol{u}}_{v(k)}||^2_R$, with $\boldsymbol{P} \geq 0 \in \mathbb{R}^{n_x \times n_x}$, $\boldsymbol{Q} > 0 \in \mathbb{R}^{n_x \times n_x}$, and $\boldsymbol{R} > 0 \in \mathbb{R}^{n_u \times n_u}$ being the cost matrices for the terminal state, stage states, and stage inputs, respectively. This standard formulation is slightly adapted to increase the numerical robustness of the solver (see e.g., [29]) by adding a small robustness margin of $0.02\mathrm{m}$ around obstacles and adding additional slacks variables $s_j \in \mathbb{R}$ to each of the state constraints (i.e., $\dot{y}_{\max}$ and $\dot{y}_{\min}$ velocities and N_o obstacles). Defining the slack vector as $\boldsymbol{s} = [s_1, \ldots, s_{n_s}]^T \in \mathbb{R}^{n_s}$ with $n_s = 2 + N_\mathrm{o}$, the control decision variables at stage index i are augmented from $\boldsymbol{\mu}_i$ to $\boldsymbol{\mu}'_i = [\boldsymbol{\mu}_i^T, \boldsymbol{s}_i^T]^T$, and the cost is accumulated with $\sum_{i=0}^{N} ||\boldsymbol{s}_i||^2_{R_s}$ and the constraints with $\boldsymbol{R}_s = 10^6 \boldsymbol{I}_{n_s}$.

The OCP (7) without (8) is a special case of the following general discrete-time OCP

$$\min_{\boldsymbol{\mu}'} \quad \frac{1}{2} \sum_{i=0}^{N} \ell(\boldsymbol{\xi}_i, \boldsymbol{\mu}'_i, i) \tag{28a}$$

$$\text{s.t.} \quad \boldsymbol{\xi}_0 = \boldsymbol{x}(k), \tag{28b}$$

$$\boldsymbol{\xi}_{i+1} = \boldsymbol{f}(\boldsymbol{\xi}_i, \boldsymbol{\mu}_i), \quad i \in \mathbb{N}_{[0,N-1]}, \tag{28c}$$

$$\boldsymbol{c}(\boldsymbol{\xi}_i, \boldsymbol{\mu}'_i) \leq 0, \qquad i \in \mathbb{N}_{[0,N-1]}, \tag{28d}$$

with cost $\ell(\boldsymbol{\xi}_i, \boldsymbol{\mu}'_i, i)$ defined differently for the cases $i = N$ (terminal) and $0 \leq i < N$ (stage). The OCP (28) is then solved with Sequential Quadratic Programming (SQP) specified in Appendix A, but any suitable algorithm could be used. The SQP routine is implemented with a maximum of 15 iterations and at most 25 iterations can be used to solve each QP to the specified tolerance. In each control cycle, the initial guess for the SQP is chosen as the POSS solution trajectory. The stage cost matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$, and the terminal cost $\boldsymbol{P}$ are typically designed starting from a terminal control law $\boldsymbol{K} \in \mathbb{R}^{n_u \times n_x}$,

$$\boldsymbol{\kappa}_{\mathrm{PSC}}(\boldsymbol{x}(k), \boldsymbol{v}(k)) = \bar{\boldsymbol{u}}_{v(k)} + \boldsymbol{K}(\bar{\boldsymbol{x}}_{v(k)} - \boldsymbol{x}(k)). \tag{29}$$

They are tuned for aggressive performance and are set to $\boldsymbol{Q} = \mathrm{diag}(200, 200, 20, 0, 0, 0)$, $\boldsymbol{R} = \mathrm{diag}(2, 20)$, and $\boldsymbol{P} \in \mathbb{R}^{6\times 6}$ equal to the discrete-time Linear Quadratic Regulator (LQR) cost of the linearized version of the UAV dynamics (26), i.e.,

$$\frac{d}{dt}\begin{bmatrix} y \\ z \\ \phi \\ \dot{y} \\ \dot{z} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & -g & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0] \end{bmatrix} \begin{bmatrix} y \\ z \\ \phi \\ \dot{y} \\ \dot{z} \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{1}{m} & 0 \\ 0 & \frac{1}{J} \end{bmatrix} \begin{bmatrix} T' \\ \tau \end{bmatrix}, \tag{30}$$

with $T' = T - mg$. The sampling time and the prediction time step of the NMPC are both set to $T_{\mathrm{s,\ MPC}} = \Delta t_{\mathrm{p,\ MPC}} = 50\mathrm{ms}$ and its prediction horizon consists of N prediction steps that will be specified per validation case.

The FG's DSM Δ defined in (21) predicts the smallest distance to constraint violation of the nonlinear UAV's (26) terminal dynamics pre-stabilized with an input-saturated version of $\boldsymbol{\kappa}_{\mathrm{PSC}}$ in (29). This corresponds to the LQR control law $\boldsymbol{K}$ computed with the $\boldsymbol{Q}, \boldsymbol{R}$ costs of the NMPC. The sdist function of (21) is scaled for each constraint in $\mathcal{Y}$ with a large factor of 50 to push the performance of the FG. Similar to the NMPC, we increase the robustness of the FG by adding a small robustness margin of $0.02\mathrm{m}$ around obstacles. The adaptive scaling factor $\kappa(k)$ is initialized to $\kappa_0 = 1$ and is exponentially reduced with reduction rate $\lambda = 0.8$ or increased with rate $\frac{1}{0.8}$ in the interval $[\underline{\kappa}, \overline{\kappa}] = (0, 1]$. The FG's sampling time and its prediction time step are both set to $T_{\mathrm{s,\ FG}} = \Delta t_{\mathrm{p,\ MPC}} = 50\mathrm{ms}$. The long prediction horizon of the FG contains $M - N = 100$ samples, which ensures in the subsequent simulations that the final predicted FG state $\boldsymbol{\xi}_M$ always reaches very close to the time-varying steady-state $\boldsymbol{m}_x(\boldsymbol{v}(k))$ and hence that the DSM of the terminal energy constraint (21e) is not the dominating worst-case. During one sample time, at most $\bar{l} = 5$ iterations are used to try computing a steady-state admissible auxiliary reference, i.e, $\boldsymbol{v}(k) \in \mathcal{V}_\delta$. However, this can be easily set to a larger number as it is very computationally efficient to check for steady-state admissibility. The NF $\boldsymbol{\rho}$ of the FG defined in (23) is implemented with $\eta = 0.01\mathrm{m}$, $\zeta_{\mathrm{o}_j} = 0.75\mathrm{m}$, $\delta_{\mathrm{o}_j} \in (0, \zeta_{\mathrm{o}_j}) = 0.07\mathrm{m}$, and $\alpha_{\mathrm{o}_j} = \pm 0.2$.

A. FG-NMPC with a short NMPC prediction horizon

The resulting closed-loop performance, constraint satisfaction, and operation diagnostics of the real-time FG-NMPC scheme with a short NMPC prediction horizon of $N = 5$ are depicted in Fig. 3. Due to the complexity of the overall problem (i.e., system and constraints nonlinearity, obstacle non-convexity, and the short prediction horizon), an NMPC

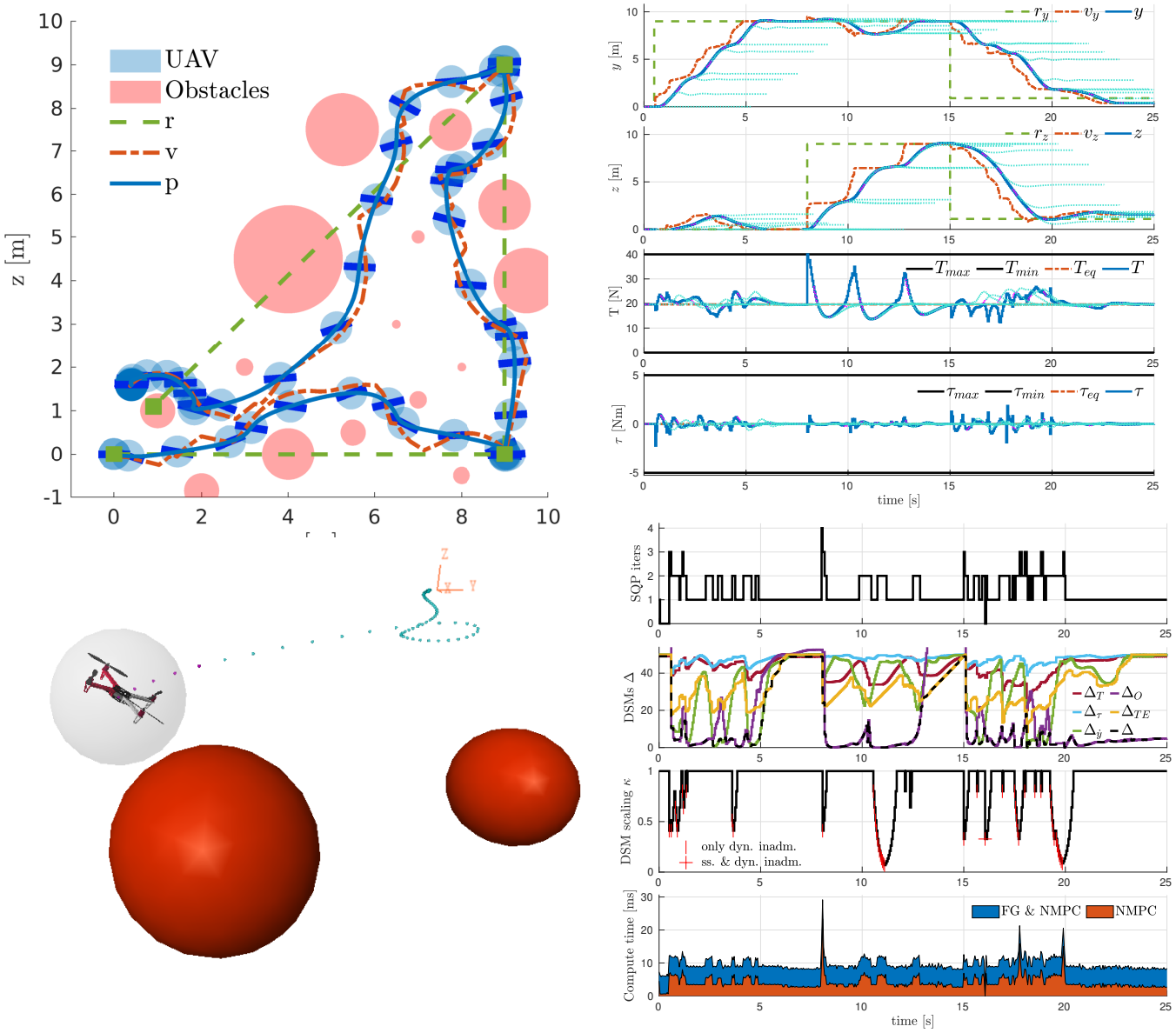


Fig. 3: The FG-NMPC law with $N = 5$ ensures real-time stable and collision-free UAV point-to-point transitioning in an a priori unknown environment cluttered with 15 circular obstacles in red. Snapshots of the UAV's pose and collision radius R_{UAV} are depicted each 0.5s along the UAV's path in blue. The dashed orange lines in the top sub-figures and the orange reference frame in the bottom left sub-figure represent the path of feasible auxiliary references $\boldsymbol{v}(k)$ computed online by the FG and applied to the NMPC. The UAV passes very close to the red obstacles as the NMPC's objective strongly penalizes the position errors between the predicted optimal trajectory, depicted by the magenta dots in the bottom left and top right sub-figures, for the equilibrium $\bar{\boldsymbol{x}}_{v(k)}$. The FG predicts the terminal dynamics over a long horizon of $M - N = 100$ samples, between the POSS NMPC's terminal state $\boldsymbol{\xi}_{N|k+1}$ and the FG's terminal state $\boldsymbol{\xi}_{M|k+1}$ near $\bar{\boldsymbol{x}}_{v(k)}$, depicted by the cyan dots in the bottom left and top right sub-figures. The bottom right sub-figure shows that the DSM $\Delta(k)$ (line 10) is mainly dominated by state constraints (i.e, velocities $\Delta_{\dot{y}}$, obstacles Δ_{O}, and the terminal energy Δ_{TE}) as the DSMs of the input constraints (i.e., the total thrust Δ_T and the body torque Δ_τ) are always significantly larger. The DSM together with the NF and the time-varying scaling factor $\kappa(k)$ allows computing a sequence of statically and dynamically feasible auxiliary references $\boldsymbol{v}(k)$ on the fly. This $\boldsymbol{v}(k)$ sequence asymptotically converges to the time-varying target reference trajectory $\boldsymbol{r}(k)$ when held constant, which are four consecutive hard step setpoints depicted by the green squares in $[0, 0]^T$, $[9, 0]^T$, $[9, 9]^T$, and $[0.9, 1.1]^T$ requested at 0.5s, 8s, and 15s, respectively. As the last target setpoint lies inside the obstacle centered at $[1, 1]^T$ and is therefore steady-state inadmissible (i.e., $\boldsymbol{r}(k) \notin \mathcal{R}_\delta$), the UAV converges to the closest steady-state $\boldsymbol{r}^*$ that satisfies the obstacle constraints and lies on the bisector of the second quadrant. In the bottom right sub-figure it is shown that the computational time of Algorithm 1 remains well below the sampling time of 50ms, which allows real-time execution. Moreover, the NMPC dominates the total computational time as the peaks at 8s and between $17 - 20$s require most SQP iterations. The average computational time of this short-horizon NMPC is similar to the very long-horizon FG. Supplementary video material can be found on https://youtu.be/2LSYNwuYpzI.

without an FG fails to find feasible trajectories when solving (7) directly for the waypoints $\boldsymbol{r}$, even under very large terminal set $\mathcal{T}$ sizes and after increasing the max number of SQP iterations to 150.

Note that if the terminal set is reachable within N steps, the FG's adaptation of the auxiliary reference could be slow such that the standard approach (assuming suitable terminal conditions in an NMPC that can be solved fast enough) outperforms the proposed one. However, it approximates the standard approach arbitrarily well for sufficiently large gains κ (at least, in continuous time). If the terminal set is not reachable within N steps, then this FG recovers feasibility. The convergence speed mainly depends on the system dynamics and the time-varying scaling factor $\kappa(k)$ that is affected by its initial value κ_0, its growth/reduction rate $(1-\lambda)$, on the foreseen maximum number of iterations $\bar{l}$ (to reduce $\kappa(k)$ over a single control period if the auxiliary reference guess is steady-state inadmissible), and on the scaling factors inside the DSM Δ, i.e., a scaled distance metric to constraint violation.

B. FG-NMPC over arbitrary NMPC prediction horizons

The results of the FG-NMPC scheme over varying NMPC prediction horizons N and a fixed FG predicting horizon $M-N$ are depicted in Fig. 4. One can observe that independently of the horizon length N, a feasible trajectory is always found. Moreover, as N increases, the closed-loop performance improves in terms of rise/settling time and results in a reduction of the cumulative closed-loop cost, defined as

$$J_{\text{CUM-CL}} = \sum_{k=0}^{k_f} ||\boldsymbol{x}_k - \bar{\boldsymbol{x}}_{v(k)}||_Q^2 + ||\boldsymbol{u}_k - \bar{\boldsymbol{u}}_{v(k)}||_R^2, \qquad (31)$$

which is the NMPC's stage cost function (ignoring the terminal cost) evaluated over the closed-loop state/input trajectory $\boldsymbol{x}(k)$/$\boldsymbol{u}(k)$ and integrated over the entire task duration expressed as the number of control samples $k_f > 0$. The trade-off between the performance benefit and the increased computational cost is summarized in Table I. One can see that both the maximum and average execution time of the FG is significantly lower than that of the NMPC as N is increased. Note that even for the longest NMPC horizon of $N = 60$, a pure NMPC law without FG cannot find feasible trajectories for the far reference $\boldsymbol{r}$. For longer NMPC horizons, i.e., $N > 60$, no significant performance gain could be obtained.

C. FG-NMPC: terminal control law tuning effects

Fig. 5 shows the effects of different terminal LQR control law gains, while leaving the NMPC gains untouched. The terminal law is computed using a scaled input penalty $\boldsymbol{R}$. We observe that a significant performance improvement can be obtained for $\boldsymbol{R}/5$, $\boldsymbol{R}/2$, $\boldsymbol{R}/10$, and $\boldsymbol{R}/15$ in the obstacle-cluttered environment. However, when there are no obstacles, no performance benefit is obtained. This difference can be explained since scaling factors smaller than 1 result in smaller overshoots and larger obstacle DSM's Δ_O. Also, the performance loss by largely de-tuning the LQR (e.g., $50\,\boldsymbol{R}$) can be explained as the FG's prediction horizon is too short for the more oscillatory dynamics to settle, hence the terminal energy DSM is the worst-case, which means that the FG's prediction horizon should be even chosen longer to prevent this.

D. Application to (non)holonomic systems

Our method is applied to a quadrotor, which is in principle a nonholonomic system. However, with a fast-controlled attitude loop, they can produce velocities quickly in any direction. In contrast, true nonholonomic systems such as wheeled mobile robots have directions in which they cannot move [30], often requiring complex compositions of primitive maneuvers to steer them to a desired pose. The current FG & NMPC is not yet tailored to the application of nonholonomic systems, but this is an interesting direction for future work. Foundational works, such as NMPC schemes with terminal constraints and cost [31] (no obstacles) or without terminal constraints and costs but with nonquadratic stage cost and minimal prediction horizon lengths [32], [33] (no obstacles), and the Lyapunov-based ERG of [34] (no input and obstacle constraints) can serve as a good starting point. Also, [32] reported that the minimum required MPC prediction horizon for nonholonomic robots rapidly grows if the initial condition is located (very) close to the origin. Therefore, it is important to consider this in the design of the FG as its filtering action on the auxiliary reference might cause an increased computational load.

VII. Conclusion

We can conclude that the proposed discrete-time trajectory-based FG with Lyapunov-based terminal energy constraint is an effective and computationally efficient tool to prevent the common problem of an NMPC's OCP that becomes infeasible. This problem typically occurs when the target reference moves too far too quickly, whereby the terminal set constraint cannot be reached within the limited NMPC's horizon. This FG-NMPC scheme ensures feasibility for any setpoint and enlarges the region of attraction of the NMPC with arbitrarily short prediction horizons to any strictly steady-state admissible setpoint. The FG is developed for constrained nonlinear systems subject to pointwise-in-time state and input constraints. It relies on a minimal set of non-stringent assumptions and excludes the terminal set constraints from the OCP. This significantly improves and simplifies prior work on FGs that are limited to linear systems, that require expensive (and quickly intractable) offline computations to construct MPC's feasible and/or terminal sets, and that require a priori known constraints. The efficacy of the combined FG & NMPC algorithm was demonstrated through hardware-in-the-loop simulation benchmarks on a planar quadrotor that aggressively and safely flies through an a priori unknown obstacle environment and is shown to satisfy all constraints, achieve asymptotic stability, exhibit zero-offset tracking for piecewise-continuous and possibly steady-state inadmissible target references, and require low computational effort. In these validations, we have employed an SQP with non-convex constraints. The proposed framework is independent of the solver type, making it flexible to use different out-of-the-box solvers where the NMPC does not need to be touched. Finally, we also gave some heuristics on how improved closed-loop

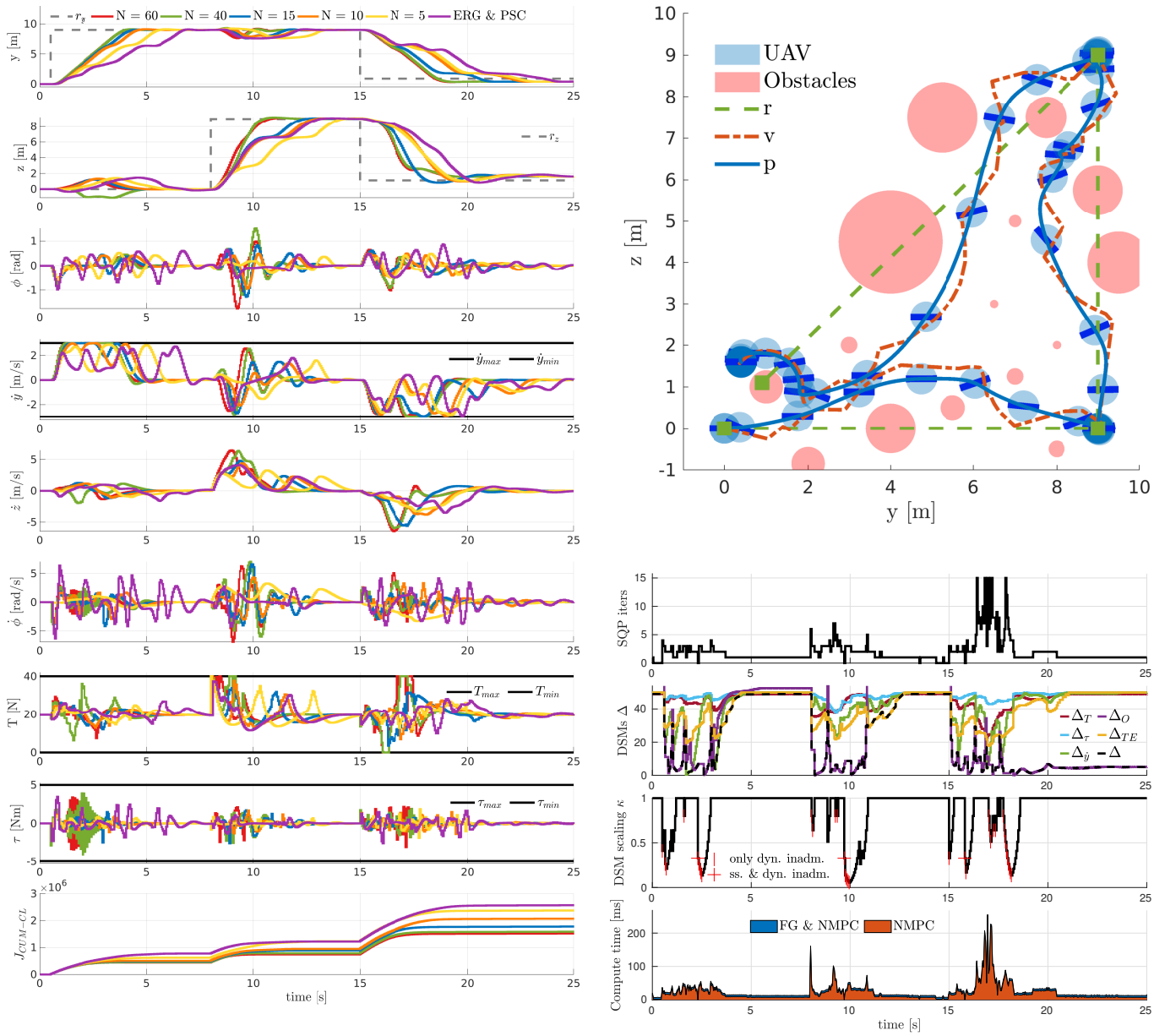


Fig. 4: The left sub-figure depicts the bounded state and input dynamics of the FG-NMPC law and demonstrates the asymptotic stability, constraint satisfaction, and tuneable closed-loop performance of the proposed scheme for arbitrary NMPC prediction horizons N, where a performance benefit (i.e., the reduced cumulative closed-loop cost $J_{\text{CUM-CL}}$ computed from (31) and rise/settling time) is obtained as N is increased at the cost of significantly more NMPC computations. The bottom right sub-figure depicts the NMPC and FG diagnostics for the case of a medium NMPC horizon with $N = 15$ and although safety with better performance compared to $N = 5$ in Fig. 3 is obtained, the computational time peaks required by the NMPC law already largely exceed the real-time requirements of 50ms. Supplementary video material can be found on https://youtu.be/2LSYNwuYpzI.

TABLE I: Average (TAVE) and maximum (TMAX) computational times and the performance gain margin (i.e., the margin for cumulative cost reduction) of the ERG & PSC and the FG & NMPC schemes over arbitrary NMPC prediction horizons N and a fixed long ERG/FG prediction horizon $M - N = 100$ when applied to the safe navigation through obstacles of a nonlinear quadrotor model which is subject to several state and input constraints.

		ERG & PSC (M =100)	FG & NMPC (N = 5, M = 105)	FG & NMPC (N = 10, M = 110)	FG & NMPC (N = 15, M = 115)	FG & NMPC (N = 40, M = 140)	FG & NMPC (N = 60, M = 160)
TOTAL	TAVE [ms]	5.487	9.203	13.616	22.273	58.118	79.005
	TMAX [ms]	16.675	29.142	81.342	255.46	810.82	1739.5
CONTROLLER (PSC or NMPC)	TAVE [ms]	0.005	3.715	8.149	16.799	52.599	73.49
	TMAX [ms]	0.066	23.35	75.678	249.78	805.48	1734.2
GOVERNOR (ERG or FG)	TAVE [ms]	5.482	5.488	5.466	5.473	5.519	5.514
	TMAX [ms]	16.652	6.182	6.111	6.613	6.24	6.826
PERFORMANCE GAIN MARGIN TO FG & NMPC (N = 60, M = 160) [%]		68	55	36	17	4	0

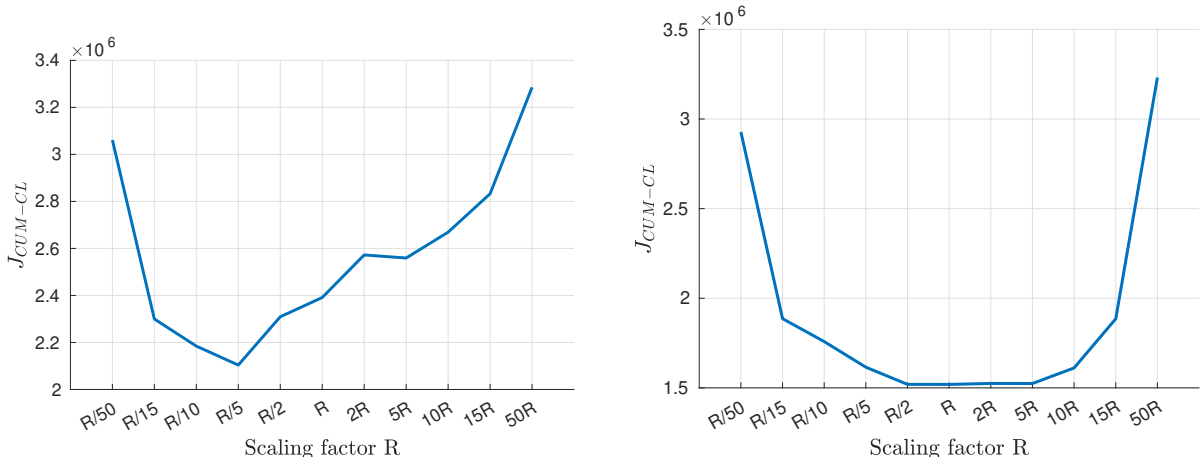


Fig. 5: The effect of re-tuning the terminal control law input penalty $\boldsymbol{R}$ of an FG-NMPC with $N = 5$ on the cumulative closed-loop cost for an obstacle-cluttered environment (left) and obstacle-free environment (right).

performance can be obtained by re-tuning the terminal LQR law without changing the NMPC. Future work will be focused on extending this method for nonholonomic systems, dynamic obstacles [35], and escaping more local minima [36], [37].

Appendix A
Sequential Quadratic Program

Let $\bar{\boldsymbol{\xi}} = (\bar{\boldsymbol{\xi}}_0, \dots, \bar{\boldsymbol{\xi}}_N)$ and $\bar{\boldsymbol{\mu}}' = (\bar{\boldsymbol{\mu}}'_0, \dots, \bar{\boldsymbol{\mu}}'_N)$ be the trajectories corresponding to current SQP iteration. To reduce the computational cost and convergence properties of the optimization algorithm, we use the Gauss-Newton Hessian approximation to approximate the Hessian of the cost function. The QPs that need to be solved are

$$\min_{\delta\boldsymbol{\mu}'} \sum_{i=0}^{N} \frac{1}{2} \begin{bmatrix} \delta\boldsymbol{\xi}_i \\ \delta\boldsymbol{\mu}'_i \end{bmatrix}^T \begin{bmatrix} \boldsymbol{Q}_i & \boldsymbol{S}_i^T \\ \boldsymbol{S}_i & \boldsymbol{R}'_i \end{bmatrix} \begin{bmatrix} \delta\boldsymbol{\xi}_i \\ \delta\boldsymbol{\mu}'_i \end{bmatrix} + \begin{bmatrix} \boldsymbol{q}_i \\ \boldsymbol{r}_i \end{bmatrix}^T \begin{bmatrix} \delta\boldsymbol{\xi}_i \\ \delta\boldsymbol{\mu}'_i \end{bmatrix} \tag{32a}$$

$$\text{s.t.} \quad \delta\boldsymbol{\xi}_0 = \Delta\boldsymbol{x}_0, \tag{32b}$$

$$\delta\boldsymbol{\xi}_{i+1} = \boldsymbol{A}_i\delta\boldsymbol{\xi}_i + \boldsymbol{B}_i\delta\boldsymbol{\mu}'_i + \boldsymbol{d}_i, \quad i \in \mathbb{N}_{[0,N-1]}, \tag{32c}$$

$$\boldsymbol{E}_i\delta\boldsymbol{\xi}_i + \boldsymbol{L}_i\delta\boldsymbol{\mu}'_i + \boldsymbol{c}_i \leq 0, \qquad i \in \mathbb{N}_{[0,N-1]} \tag{32d}$$

with $\Delta\boldsymbol{x}_0 = \boldsymbol{x}(k) - \bar{\boldsymbol{\xi}}_0$,

$$\boldsymbol{A}_i = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{\xi}}(\bar{\boldsymbol{\xi}}_i, \bar{\boldsymbol{\mu}}'_i), \quad \boldsymbol{B}_i = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{\mu}'}(\bar{\boldsymbol{\xi}}_i, \bar{\boldsymbol{\mu}}'_i) \ , \boldsymbol{d}_i = \boldsymbol{f}(\bar{\boldsymbol{\xi}}_i, \bar{\boldsymbol{\mu}}'_i) - \bar{\boldsymbol{\xi}}_{i+1}, \tag{33}$$

$$\boldsymbol{E}_i = \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{\xi}}(\bar{\boldsymbol{\xi}}_i, \bar{\boldsymbol{\mu}}'_i, i), \quad \boldsymbol{L}_i = \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{\mu}'}(\bar{\boldsymbol{\xi}}_i, \bar{\boldsymbol{\mu}}'_i, i), \quad \boldsymbol{c}_i = \boldsymbol{c}(\bar{\boldsymbol{\xi}}_i, \bar{\boldsymbol{\mu}}'_i, i), \tag{34}$$

$$\begin{bmatrix} \boldsymbol{Q}_i & \boldsymbol{S}_i^T \\ \boldsymbol{S}_i & \boldsymbol{R}_i \end{bmatrix} = \begin{bmatrix} \frac{\partial^2 \ell}{\partial \boldsymbol{\xi}^2}(\bar{\boldsymbol{\xi}}_i, \bar{\boldsymbol{\mu}}'_i, i) & \frac{\partial^2 \ell}{\partial \boldsymbol{\xi} \partial \boldsymbol{\mu}}(\bar{\boldsymbol{\xi}}_i, \bar{\boldsymbol{\mu}}'_i, i) \\ \frac{\partial \ell}{\partial \boldsymbol{\mu}' \partial \boldsymbol{\xi}}(\bar{\boldsymbol{\xi}}_i, \bar{\boldsymbol{\mu}}'_i, i) & \frac{\partial^2 \ell}{\partial \boldsymbol{\mu}'^2}(\bar{\boldsymbol{\xi}}_i, \bar{\boldsymbol{\mu}}'_i, i), \end{bmatrix} \tag{35}$$

$$\boldsymbol{q}_i = \frac{\partial \ell}{\partial \boldsymbol{\xi}}(\bar{\boldsymbol{\xi}}_i, \bar{\boldsymbol{\mu}}'_i, i), \quad \boldsymbol{r}_i = \frac{\partial \ell}{\partial \boldsymbol{\mu}'}(\bar{\boldsymbol{\xi}}_i, \bar{\boldsymbol{\mu}}'_i, i) \tag{36}$$

with $\boldsymbol{Q}_i$, $\boldsymbol{R}_i$ the stage cost Hessians, $\boldsymbol{S}_i$ the Jacobian of the stage cost gradient, and $\boldsymbol{q}_i$, $\boldsymbol{r}_i$ the stage cost gradients.

We use a `C++` implementation of the QP solver FBstab [38] (https://github.com/dliaomcp/fbstab) to solve (32). After solving (32) for $\delta\boldsymbol{\mu}'$ and $\delta\boldsymbol{\xi}$ we can update the iterate as

$$(\bar{\boldsymbol{\xi}}, \bar{\boldsymbol{\mu}}') \leftarrow (\bar{\boldsymbol{\xi}}, \bar{\boldsymbol{\mu}}') + \alpha(\delta\boldsymbol{\xi}, \delta\boldsymbol{\mu}'), \tag{37}$$

where α is chosen using a backtracking line-search with a merit function as described in [39, Algorithm 6.7].

References

[1] G. Goodwin, M. M. Seron, and J. A. d. Doná, *Constrained Control and Estimation: An Optimisation Approach*. Communications and Control Engineering, London: Springer London, 1 ed., 2005.

[2] J. B. Rawlings, D. Q. Mayne, and M. M. Diehl, *Model Predictive Control: Theory, Computation, and Design*. Nob Hill Publishing, LLC, 2 ed., 2020.

[3] D. Limon, T. Alamo, D. M. Raimondo, D. M. De La Peña, J. M. Bravo, A. Ferramosca, and E. F. Camacho, "Input-to-State Stability: A Unifying Framework for Robust Model Predictive Control," *Lecture Notes in Control and Information Sciences*, vol. 384, pp. 1–26, 2009.

[4] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.

[5] D. Limon, I. Alvarado, T. Alamo, and E. F. Camacho, "MPC for tracking piecewise constant references for constrained linear systems," *Automatica*, vol. 44, pp. 2382–2387, 9 2008.

[6] D. Simon, J. Lofberg, and T. Glad, "Reference Tracking MPC Using Dynamic Terminal Set Transformation," *IEEE Transactions on Automatic Control*, vol. 59, pp. 2790–2795, 10 2014.

[7] L. Fagiano and A. R. Teel, "Generalized terminal state constraint for model predictive control," *Automatica*, vol. 49, pp. 2622–2631, 9 2013.

[8] D. Limon, T. Alamo, and E. F. Camacho, "Enlarging the domain of attraction of MPC controllers," *Automatica*, vol. 41, pp. 629–635, 4 2005.

[9] L. Chisci and G. Zappa, "Dual mode predictive tracking of piecewise constant references for constrained linear systems," *International Journal of Control*, vol. 76, pp. 61–72, 1 2010.

[10] D. Q. Mayne and P. Falugi, "Stabilizing conditions for model predictive control," *International Journal of Robust and Nonlinear Control*, vol. 29, pp. 894–903, 3 2019.

[11] S. Olaru and D. Dumur, "Compact explicit MPC with guarantee of feasibility for tracking," in *Proceedings of the 44th IEEE Conference on Decision and Control*, pp. 969–974, IEEE, 2005.

[12] T. Skibik, D. Liao-Mc Pherson, T. Cunis, I. V. Kolmanovsky, and M. M. Nicotra, "A Feasibility Governor for Enlarging the Region of Attraction of Linear Model Predictive Controllers," *IEEE Transactions on Automatic Control*, vol. 67, no. 10, pp. 5501 – 5508, 2022.

[13] E. Garone, S. Di Cairano, and I. Kolmanovsky, "Reference and command governors for systems with constraints: A survey on theory and applications," *Automatica*, vol. 75, pp. 306–328, 2017.

[14] A. Bemporad, A. Casavola, and E. Mosca, "Nonlinear control of constrained linear systems via predictive reference management," *IEEE Transactions on Automatic Control*, vol. 42, no. 3, pp. 340–349, 1997.

[15] C. N. Jones, E. C. Kerrigan, and J. M. MacIejowski, "On Polyhedral Projection and Parametric Programming," *Journal of Optimization Theory and Applications 2008 138:2*, vol. 138, pp. 207–220, 4 2008.

[16] M. Herceg, M. Kvasnica, C. N. Jones, and M. Morari, "Multi-parametric toolbox 3.0," in *European Control Conference*, pp. 502–510, IEEE, 2013.

[17] D. Ciripoi, A. Löhne, and B. Weißing, "Calculus of convex polyhedra and polyhedral convex functions by utilizing a multiple objective linear programming solver," *Optimization*, vol. 68, pp. 2039–2054, 10 2018.

[18] F. Scibilia, S. Olaru, and M. Hovd, "On feasible sets for MPC and their approximations," *Automatica*, vol. 47, pp. 133–139, 1 2011.

[19] E. M. Bronstein, "Approximation of convex sets by polytopes," *Journal of Mathematical Sciences 2008 153:6*, vol. 153, pp. 727–762, 9 2008.

[20] T. Skibik, D. Liao-McPherson, and M. M. Nicotra, "A Terminal Set Feasibility Governor for Linear Model Predictive Control," *IEEE Transactions on Automatic Control*, pp. 1–7, 10 2022.

[21] E. G. Gilbert and K. T. Tan, "Linear Systems with State and Control Constraints: The Theory and Application of Maximal Output Admissible Sets," *IEEE Transactions on Automatic Control*, vol. 36, no. 9, pp. 1008–1020, 1991.

[22] K. Hirata and Y. Ohta, "Exact determinations of the maximal output admissible set for a class of nonlinear systems," *Automatica*, vol. 44, pp. 526–533, 2 2008.

[23] M. M. Nicotra and E. Garone, "The Explicit Reference Governor: A General Framework for the Closed-Form Control of Constrained Nonlinear Systems," *IEEE Control Systems Magazine*, vol. 38, pp. 89–107, 8 2018.

[24] M. M. Nicotra and E. Garone, "An Explicit Reference Governor for the robust constrained control of nonlinear systems," in *IEEE Conference on Decision and Control*, no. 2, pp. 1502–1507, 2016.

[25] B. Convens, K. Merckaert, B. Vanderborght, and M. M. Nicotra, "Invariant Set Distributed Explicit Reference Governors for Provably Safe On-Board Control of Nano-Quadrotor Swarms," *Frontiers in Robotics and AI*, vol. 8, pp. 1–23, 6 2021.

[26] B. Convens, K. Merckaert, M. M. Nicotra, R. Naldi, and E. Garone, "Control of Fully Actuated Unmanned Aerial Vehicles with Actuator Saturation," in *IFAC-PapersOnLine*, vol. 50, pp. 12715–12720, Elsevier, 7 2017.

[27] K. Merckaert, B. Convens, C. j. Wu, A. Roncone, M. M. Nicotra, and B. Vanderborght, "Real-time motion control of robotic manipulators for safe human–robot coexistence," *Robotics and Computer-Integrated Manufacturing*, vol. 73, pp. 1–14, 2 2022.

[28] B. Convens, K. Merckaert, M. M. Nicotra, and B. Vanderborght, "Safe, Fast, and Efficient Distributed Receding Horizon Constrained Control of Aerial Robot Swarms," *IEEE Robotics and Automation Letters*, vol. 7, pp. 4173–4180, 4 2022.

[29] M. N. Zeilinger, M. Morari, and C. N. Jones, "Soft constrained model predictive control with robust stability guarantees," *IEEE Transactions on Automatic Control*, vol. 59, no. 5, pp. 1190–1202, 2014.

[30] R. W. Brockett, "Asymptotic stability and feedback stabilization," in *Differential Geometric Control Theory* (R. W. Brockett, R. S. Millman, and H. J. Sussmann, eds.), pp. 181–191, 1983.

[31] D. Gu and H. Hu, "A stabilizing receding horizon regulator for nonholonomic mobile robots," *IEEE Transactions on Robotics*, vol. 21, pp. 1022–1028, 10 2005.

[32] K. Worthmann, M. W. Mehrez, M. Zanon, G. K. Mann, R. G. Gosine, and M. Diehl, "Model Predictive Control of Nonholonomic Mobile Robots Without Stabilizing Constraints and Costs," *IEEE Transactions on Control Systems Technology*, vol. 24, pp. 1394–1406, 7 2016.

[33] M. A. Müller and K. Worthmann, "Quadratic costs do not always work in MPC," *Automatica*, vol. 82, pp. 269–277, 8 2017.

[34] A. B. Luiz, S. E. Ebrahim, and R. P. Edson, "Constrained Control of Non-Holonomic Mobile Manipulator by Explicit Reference Governor," in *LARS-SBR-WRE*, pp. 156–161, IEEE, 2021.

[35] I. Batkovic, M. Ali, P. Falcone, and M. Zanon, "Safe Trajectory Tracking in Uncertain Environments," *IEEE Transactions on Automatic Control*, vol. 68, pp. 4204–4217, 7 2023.

[36] R. Soloperto, A. Mesbah, and F. Allgower, "Safe Exploration and Escape Local Minima with Model Predictive Control under Partially Unknown Constraints," *IEEE Transactions on Automatic Control*, vol. 68, no. 12, pp. 7530 – 7545, 2023.

[37] K. Merckaert, B. Convens, M. M. Nicotra, and B. Vanderborght, "Real-time constraint-based planning and control of robotic manipulators for safe human-robot collaboration," *Robotics and Computer-Integrated Manufacturing*, vol. 87, p. 102711, 6 2024.

[38] D. Liao-McPherson and I. Kolmanovsky, "FBstab: A proximally stabilized semismooth algorithm for convex quadratic programming," *Automatica*, vol. 113, p. 108801, 3 2020.

[39] A. F. Izmailov and M. V. Solodov, *Newton-Type Methods for Optimization and Variational Problems*. Springer Series in Operations Research and Financial Engineering, Springer, 2014.

Bryan Convens is a postdoctoral researcher at the Vrije Universiteit Brussel (VUB). He received a joint M.S. degree in electromechanical engineering from the VUB and the Université Libre de Bruxelles, Brussels, Belgium, in 2016. He obtained his Ph.D. degree in robotics and control engineering at the VUB in 2023, which was largely funded by his strategic basic research scholarship from the Research Foundation Flanders and the Flemish AI Program. He was a visiting research scholar at the Center for Research on Complex Automated Systems, University of Bologna, Italy, in 2015, at the University of Colorado Boulder, USA, in 2019, and at the University of Michigan, USA, in 2023-2024. His current research interests include constrained, optimal, and distributed control theory for aerial robots, multi-robot systems, and human-robot interaction.

Dominic Liao-McPherson is an Assistant Professor of Mechanical Engineering at the University of British Columbia in Vancouver. He received his B.S. in Engineering Science from the University of Toronto in 2015, his Ph.D. in Aerospace Engineering and Scientific Computing from the University of Michigan, Ann Arbor, in 2020, and was a postdoctoral researcher at the ETH Zürich Automatic Control Laboratory before joining UBC. His research interests include real-time optimization, algorithmic game-theory, multi-agent control, and numerical algorithms with applications in robotics, energy, and manufacturing.

Kelly Merckaert is a postdoctoral researcher at the Vrije Universiteit Brussel (VUB). She received a joint M.S. degree in electromechanical engineering from the VUB and the Université Libre de Bruxelles, Belgium, in 2016. She received her Ph.D. degree in robotics and control engineering at the VUB, Belgium, in 2023. She has been a visiting research scholar at the Center for Research on Complex Automated Systems, University of Bologna, Italy, in 2015, at the University of Colorado Boulder, USA, in 2019, and at the University of Michigan, USA, in 2023-2024. Her research interests include control theory and learning for robotic manipulators, aerial robots, and human-robot interaction systems.

Bram Vanderborght obtained his Ph.D. from the Vrije Universiteit Brussel in 2007. He performed research at JRL lab in AIST, Tsukuba (Japan) and did his post-doc researcher at the Italian Institute of Technology. Since 2009 he is professor at the Vrije Universiteit Brussel. He had an ERC starting grant and is currently coordinating three EU projects on smart and self-healing materials for soft robots. His research interests are human-robot collaboration for applications for health and manufacturing like exoskeletons, prostheses, social robots, drones and cobots. He is affiliated to the Interuniversity Microelectronics Institute (IMEC), Belgium, as scientific collaborator.

Marco M. Nicotra received his double M.S. degree in mechanical and electromechanical engineering from Politecnico di Milano, Italy, and Université Libre de Bruxelles, Belgium, under the T.I.M.E. double degree program in 2012. He received his Ph.D. degree in control engineering under the joint supervision of the Université Libre de Bruxelles and the University of Bologna, Italy, in 2016. He is currently an Assistant Professor with the University of Colorado Boulder, USA. His current research interests include constrained control of nonlinear systems, unmanned aerial vehicles, collaborative robotics, and quantum systems.