Single Degree of Freedom Model Predictive Control with Variable Horizon

Artemi Makarow, Christoph Rösmann and Torsten Bertram

The 2020 American Control Conference, Denver, CO, USA

Single Degree of Freedom Model Predictive Control with Variable Horizon

Artemi Makarow, Christoph Rösmann and Torsten Bertram

***Abstract*— This paper proposes a sampling-based model predictive control scheme with a single degree of freedom in control. A variable horizon and stabilizing terminal conditions ensure recursive feasibility, asymptotic stability, and improvement of the closed-loop performance. The initial derivation leads to a computationally demanding mixed-integer nonlinear programming problem. To address this issue, the paper presents an algorithm that seeks the optimal horizon while rolling out a finite number of control input candidates. The proposed derivative-free optimization even finds the control invariant terminal set online at reasonable computational overhead. The approach is straightforward to implement and aims for fast, albeit small-scale systems. A comparative analysis with a fixed-horizon scheme and standard model predictive control on a nonlinear benchmark system shows high closed-loop performance with low computational effort.**

I. INTRODUCTION

Model predictive control (MPC) constitutes a high-performance approach which considers nonlinear systems and explicitly adheres to input and state constraints. MPC repeatedly solves an optimal control problem (OCP) in every sampling interval [1], [2]. For its realization, the control engineer has to tackle three challenges, which are first the profound familiarization with numerical discretization methods and constrained nonlinear optimization algorithms, second, its efficient implementation under real-time constraints, and third ensuring stability and recursive feasibility.

The majority of approaches rely on direct transcription and Newton-type solvers. From a numerical point of view, multiple shooting splits the prediction horizon into several smaller initial value problems and hence reaches faster convergence [3]. Furthermore, the real-time iteration scheme enables high controller frequencies by applying a single warm-started sequential-quadratic programming step in every sampling interval [4]. An augmented Lagrangian formulation in combination with projected gradients facilitates fast applications of MPC in [5]. In [6], the structure of the underlying nonlinear program (NLP) is exploited by applying sparse and early terminated interior-point methods. Another way to speed up MPC is to utilize condensing techniques which transform the large albeit sparse NLP into a smaller dense NLP [7]. Generally, the underlying solver computes the time-consuming first (and second) order derivatives either using automatic differentiation or finite differences. Established MPC frameworks are, for example, the ACADO(S) toolkit [8], [9] and GRAMPC [5]. Sparse finite differences also exploit sparsity by evaluating only structured nonzero partial derivatives. Instead of seeking the sparsity pattern online by graph coloring, hypergraphs capture the sparse problem structure inherently and support varying NLP dimensions during runtime [10].

Move-blocking MPC reduces the computational burden by limiting the degrees of freedom in control [11]. Hereby, multiple consecutive controls are kept constant and are hence represented by a single optimization parameter for each group. The resulting NLP exhibits fewer optimization parameters. However, established stability and recursive feasibility results from MPC with terminal conditions do not apply as the optimal solution of the previous OCP is not necessarily a feasible solution of the subsequent OCP.

Terminal conditions provide a remedy in MPC to enforce stability and recursive feasibility. Dual-mode MPC switches to a locally stabilizing linear state controller within an invariant terminal set [12]. Quasi-infinite horizon MPC avoids explicit switching and estimates the infinite horizon costs as long as the horizon enters a suitable and predefined terminal set [13]. Detailed discussions on stabilizing terminal conditions can be found in [14], [15]. Variable horizons usually solve the problem of recursive feasibility [12], even in the case of move-blocking [16], [17]. [18] investigates recursive feasibility of move-blocking MPC in detail. Time-varying adaptive blocking ensures recursive feasibility as well as stability for receding and variable horizon MPC [11], [17]. Other related work [19] proposes a modified OCP formulation, which explicitly includes the homotopy to the previously planned admissible control sequence.

The numerous advances in theoretical results and approaches during the last two decades already pushed the application of MPC to nonlinear systems that require sampling times of just a few hundred of microseconds even though the required computational resources and the implementation effort are immense. *Fast mechatronic systems*, like proportional valves and servo motors, *which are present in industry* can often be described by small state space dimensions, *small input dimensions*, e.g., *a single input*, and *simple constraint sets*. These observations motivate to *search for dedicated MPC formulations that are effective in terms of performance and constraint adherence while simultaneously requiring minimal implementation effort and simple controller design*. Our previous work demonstrated experimentally that even move-blocking MPC with a single degree of freedom meets the demands on closed-loop performance for several benchmark systems under box-constraints [20]. By choosing only a single control input parameter, sampling and testing

The authors are with the Institute of Control Theory and Systems Engineering, TU Dortmund University, D-44227, Germany, artemi.makarow@tu-dortmund.de.

multiple control candidates points out to be much more efficient than applying derivative-based algorithms. Note, for larger problem definitions the curse of dimensionality limits this strategy, but its particular potential for the previously mentioned system classes is reasonable. Moreover, the possible parallelization of the sampling-based approach counteracts the curse of dimensionality. The approach exhibits the following properties: 1. Comparably low computational effort, 2. Tendency to find the global minimum (subject to blocking), 3. Deterministic number of computing operations (real-time capable), 4. Straightforward implementation and intuitive controller design, which is attractive for industrial applications. However, in its basic formulation with a receding horizon, it does not guarantee recursive feasibility and asymptotic stability.

In this contribution, we present a novel formulation which explicitly considers stability and feasibility while inheriting the previous properties. The actual contribution is twofold: First, theoretical results on variable horizon MPC with a single degree of freedom (SDOF) in control are provided. The resulting NLP constitutes a mixed-integer problem which is, in case of conventional MPC, computationally demanding. To this end, the second contribution reformulates the problem in terms of sampling multiple candidates online without computing any derivatives. This is, for the previously mentioned system classes, not only considerably more efficient but also it does not need any offline-computed terminal set. Under reasonable conditions, the approach is able to find the terminal set online while rolling out the candidates.

Section II recapitulates MPC preliminaries and introduces the nomenclature. Conditions for recursive feasibility and asymptotic stability in case of SDOF MPC are discussed in section III. Section IV describes the proposed derivative-free MPC scheme, and section V provides a proof of concept as well as demonstrates its effectiveness on a common benchmark system. The paper closes with a conclusion and an outlook on future work in section VI.

II. PRELIMINARIES

The nomenclature is partially adopted from [15], [21]. Consider the nonlinear discrete-time system given by

$$x(k+1;x_0) = f\big(x(k;x_0),u(k)\big),\ x(0;x_0) = x_0. \tag{1}$$

The state space $X = \mathbb{R}^p$ with $x(k) \in X$ and the input space $U = \mathbb{R}^m$ with $u(k) \in U$ for $k \in \mathbb{N}_0$ and $p, m \in \mathbb{N}$ form the continuous transition map $f : X \times U \to X$. The system is subject to input and state constraints, $\mathbb{U} \subset U$ and $\mathbb{X} \subseteq X$, respectively. The sets $\mathbb{X}$ and $\mathbb{U}$ are chosen to be compact. For initial state $x_0 \in \mathbb{X}$ and horizon length $N \in \mathcal{N} = \{N \in \mathbb{N} | 1 \leq N \leq N_{\max}\}$ an input sequence $u(\cdot) = \big(u(0), u(1), ..., u(N-1)\big) \in \mathcal{U}^N$ is admissible if the corresponding state trajectory satisfies $x(\cdot) = \big(x(0;x_0), x(1;x_0), ..., x(N-1;x_0)\big) \in \mathbb{X}^N$ and $x(N;x_0) \in \mathbb{X}_\mathrm{f} \subseteq \mathbb{X}$. Corresponding means that the relation of an input sequence and a state trajectory is always subject to the system dynamics (1). The space of all admissible input sequences is denoted by $\mathscr{U}_{|\mathbb{U}}^N(x_0)$. Hereby, subscript $|\mathbb{U}$ emphasizes that the underlying input set is $\mathcal{U} = \mathbb{U}$. With these definitions, the feasible set for a fixed horizon is

$$\mathcal{X}_{N|\mathbb{U}} := \{x_0 \in \mathbb{X} | \exists u(\cdot) \in \mathscr{U}_{|\mathbb{U}}^N(x_0)\}. \tag{2}$$

We consider the following finite horizon cost function

$$J_N(x_0, u(\cdot)) = \sum_{k=0}^{N-1} \ell\big(x(k;x_0), u(k)\big) + V_\mathrm{f}\big(x(N;x_0)\big) \tag{3}$$

with the continuous running cost $\ell : X \times U \to \mathbb{R}_0^+$ and the continuous terminal cost $V_\mathrm{f} : X \to \mathbb{R}_0^+$ (local Lyapunov function). For brevity, $x(k)$ replaces $x(k;x_0)$ in the following whenever the context to x_0 is either obvious or irrelevant. Without the loss of generality, we set the reference pair $(x_\mathrm{f}, u_\mathrm{f})$ to $\big(0,0\big)$ such that $f(0,0) = 0$ holds. Arbitrary reference pairs are taken into account by applying the coordinate transformations $\bar{x}(k) = x(k) - x_\mathrm{f}$ and $\bar{u}(k) = u(k) - u_\mathrm{f}$. The cost terms are designed for the stabilizing task such that they satisfy $\ell(0,0) = 0$ and $V_\mathrm{f}(0) = 0$. Set $\mathbb{X}_\mathrm{f}$ denotes the control invariant terminal set. It is compact and contains the reference state x_f in its interior. For MPC with terminal conditions, $\mathbb{X}_\mathrm{f}$ is constructed such that

$$\begin{aligned} x_{\mu_\mathrm{f}}^+ := x_{\mu_\mathrm{f}}(n+1) = f\Big(x_{\mu_\mathrm{f}}(n), \mu_\mathrm{f}\big(x_{\mu_\mathrm{f}}(n)\big)\Big) \in \mathbb{X}_\mathrm{f} \text{ and} \\ V_\mathrm{f}\big(x_{\mu_\mathrm{f}}^+\big) - V_\mathrm{f}\big(x_{\mu_\mathrm{f}}(n)\big) \leq -\gamma \ell\Big(x_{\mu_\mathrm{f}}(n), \mu_\mathrm{f}\big(x_{\mu_\mathrm{f}}(n)\big)\Big) \end{aligned} \tag{4}$$

hold with the local control law $\mu_\mathrm{f} : \mathbb{X}_\mathrm{f} \to \mathbb{U}$ for all $x_{\mu_\mathrm{f}}(n) \in \mathbb{X}_\mathrm{f}$ and all $n \in \mathbb{N}_0$. The parameter $\gamma \in (0,1]$ indicates that this work focuses mainly on the asymptotic stability rather than the optimality aspect. Ultimately, conventional MPC with horizon N rests upon the solution to the OCP

$$V_N(x_0) = \min_{u(\cdot) \in \mathscr{U}_{|\mathbb{U}}^N(x_0)} J_N\big(x_0, u(\cdot)\big) \tag{5}$$

with optimal costs $V_N(x_0)$, optimal input sequence $u^*(\cdot) := \big(u^*(0), u^*(1), \ldots, u^*(N-1)\big)$, and the corresponding optimal trajectory $x^*(\cdot) := \big(x^*(0;x_0), x^*(1;x_0), \ldots, x^*(N;x_0)\big)$. The closed-loop system results from applying the implicit control law $\mu\big(x_\mu(n)\big) := u^*(0)$ with the static mapping $\mu : \mathcal{X}_{N|\mathbb{U}} \to \mathbb{U}$ for all $x_\mu(n) \in \mathcal{X}_{N|\mathbb{U}}$. We denote the closed-loop successor state by

$$x_\mu^+ := x_\mu(n+1) = f\Big(x_\mu(n), \mu\big(x_\mu(n)\big)\Big). \tag{6}$$

The optimal costs $V_N(\cdot)$ must constitute a Lyapunov function for closed-loop system (6) to ensure asymptotic stability. In addition to (4), the following summarized common assumptions of MPC with stabilizing conditions must hold, e.g. refer to [15], [21]: There exist a function $\alpha : \mathbb{R}_0^+ \to \mathbb{R}_0^+$ that is further continuous, strictly increasing and zero at zero. This function is said to be of class $\mathcal{K}$. If it is additionally unbounded, then it is of $\mathcal{K}_\infty$. It is assumed that there exist $\alpha_1, \alpha_2 \in \mathcal{K}_\infty$ such that $\alpha_1\big(\|x(n)\|\big) \leq V_N\big(x(n)\big) \leq \alpha_2\big(\|x(n)\|\big)$ holds for all $x(n) \in \mathcal{X}_{N|\mathbb{U}}$, and $n \in \mathbb{N}_0$. $\|\cdot\|$ denotes the Euclidean norm. Also $V_\mathrm{f}\big(x(n)\big) \leq \alpha_3(\|x(n)\|)$ with $\alpha_3 \in \mathcal{K}_\infty$ must hold for all $x(n) \in \mathbb{X}_\mathrm{f}$ and $n \in \mathbb{N}_0$. Since for $x_\mu(n) \in \mathcal{X}_{N|\mathbb{U}}$ the property $x_\mu^+ \in \mathcal{X}_{N|\mathbb{U}}$ holds, recursive feasibility is ensured as well. These assumptions

usually hold under weak controllability conditions and positive definite running cost.

The remainder of this paper considers quadratic costs:

$$\ell\big(x(k;x_0),u(k)\big) := \gamma\left[\|x(k)\|_Q^2 + \|u(k)\|_R^2\right] \tag{7}$$

and the terminal costs:

$$V_{\mathrm{f}}(x(N)) := \|x(N)\|_P^2 \approx \sum_{k=N}^{\infty}\left[\|x(k)\|_Q^2 + \|u(k)\|_R^2\right] \tag{8}$$

with $\|x\|_P^2 = x^\intercal P x$. Matrices $Q \in \mathbb{R}^{p\times p}$ and $R \in \mathbb{R}^{m\times m}$ are positive definite weighting matrices. Matrix $P \in \mathbb{R}^{p\times p}$ is the solution of the discrete-time algebraic Riccati equation

$$P = A^\intercal PA - (A^\intercal PB)(R + B^\intercal PB)^{-1}(B^\intercal PA) + Q. \tag{9}$$

The matrices $A \in \mathbb{R}^{p\times p}$ and $B \in \mathbb{R}^{p\times m}$ form the linear state space model $x(k+1) = Ax(k) + Bu(k)$, which is sufficient to approximate the nonlinear system in (1) at some neighborhood of the steady-state $(x_{\mathrm{f}}, u_{\mathrm{f}})$. We assume that the pair (A, B) is stabilizable such that there exists a state feedback $\mu_{\mathrm{f}}\big(x(n)\big) := Kx(n)$ for all $x(n) \in \mathbb{X}_{\mathrm{f}}$. When applying the local control law to the nonlinear system (1), the integration of the Riccati solution P in (8) only serves as an approximation of the infinite costs. However, a terminal region can be estimated with $0 < \gamma \leq 1$ which renders the nonlinear system asymptotically stable by applying the local control law $Kx_\mu(n)$ [22]. Another possibility is to seek a terminal region which constitutes an upper bound for the infinite costs with $\|x(N)\|_P^2 \geq J_\infty\big(x(N), Kx(\cdot)\big)$. For more information on this quasi-infinite horizon approach refer to [13].

III. SDOF MODEL PREDICTIVE CONTROL

This section introduces single degree of freedom (SDOF) move-blocking subject to a variable horizon.

Definition 1: (*SDOF move-blocking*). The SDOF move-blocking policy reduces the degrees of freedom in control to one such that $u_\triangle(\cdot) = \big(u(0) = u_\triangle, u(1) = u_\triangle, \ldots, u(N-1) = u_\triangle\big)$ holds with $u_\triangle \in U$. We denote the corresponding admissible input sequence set by $\mathscr{U}_{\triangle|\mathbb{U}}^N(x_0)$.

To illustrate the lack of recursive feasibility in the case of move-blocking MPC, let us first define the blocking-free and admissible input sequences $u_0(\cdot) = \big(u_0(0), u_0(1), \ldots, u_0(N-1)\big) \in \mathscr{U}_{|\mathbb{U}}^N\big(x_\mu(0)\big)$ at time instance $n = 0$ and $u_1(\cdot) = \big(u_1(0), u_1(1), \ldots, u_1(N-1)\big) \in \mathscr{U}_{|\mathbb{U}}^N\big(x_\mu(1)\big)$ at time instance $n = 1$. At $n = 1$, conventional MPC chooses $u_1(\cdot) = \big(u_0(1), u_0(2), \ldots, u_0(N-1), \zeta\big) \in \mathscr{U}_{|\mathbb{U}}^N\big(x_\mu(1)\big)$ with an admissible $\zeta \in \mathbb{U}$. For move-blocking, the tail ζ cannot be chosen independently and hence linking consecutive OCP solutions to guarantee feasibility is not ensured. A variable horizon resolves this issue. By considering N as an optimization parameter, OCP (5) becomes:

$$V_{N^*}(x_0) = \min_{u(\cdot)\in\mathscr{U}_{|\mathbb{U}}^N(x_0),N} J_N\big(x_0, u(\cdot)\big) \tag{10}$$

and provides, besides the optimal input sequence $u^*(\cdot)$ and state trajectory $x^*(\cdot)$ also the optimal horizon length N^*. For SDOF move-blocking the OCP is as follows:

$$V_{N_\triangle^*}(x_0) = \min_{u_\triangle(\cdot)\in\mathscr{U}_{\triangle|\mathbb{U}}^N(x_0),N} J_N\big(x_0, u_\triangle(\cdot)\big). \tag{11}$$

However, the following common assumption ensures that the very first solution of (11) is feasible:

Assumption 1 (Feasible solution): It exists an $N \in \mathcal{N}$ for which an admissible input sequence $u_\triangle(\cdot) = \big(u(0) = u_\triangle, u(1) = u_\triangle, ..., u(N-1) = u_\triangle\big) \in \mathscr{U}_{\triangle|\mathbb{U}}^N(x_0)$ at time instance n generates the trajectory $x_\triangle(\cdot) = \big(x_\triangle\big(0; x_\mu(n)\big), x_\triangle\big(1; x_\mu(n)\big), \ldots, x_\triangle\big(N-1; x_\mu(n)\big)\big) \in \mathbb{X}^N$ and $x_\triangle\big(N; x_\mu(n)\big) \in \mathbb{X}_{\mathrm{f}}$. We further assume zero model-mismatch and no disturbances.

In other words, the system must be controllable to $\mathbb{X}_{\mathrm{f}}$ with a constant control value in N steps. Even though this seems to be rather limiting, recapitulate that we address certain types of systems as mentioned in section I.

Theorem 1 (Recursive feasibility): Let *Assumption 1* hold. The closed-loop system (6) based on OCP (11) ensures recursive feasibility.

Proof: The theorem and proof follow immediately from the dual-mode [12] and the variable horizon in [16], [17]. However, the derivation here is tailored for SDOF move-blocking. *Assumption 1* ensures that at time $n = 0$ a feasible solution for horizon length N exists. We now show that subsequent solutions are feasible as well. The control action $\mu\big(x_\mu(n)\big) := u(0) = u_\triangle$ steers the system to $x_\mu^+ = f\big(x_\mu(n), \mu\big(x_\mu(n)\big)\big) = x_\triangle\big(1; x_\mu(n)\big)$. Since the horizon length is variable, the solution at x_μ^+ with horizon $N-1$ is identical to the shortened input sequence $u_{\triangle-1}(\cdot) = \big(u(1) = u_\triangle, u(2) = u_\triangle, \ldots, u(N-1) = u_\triangle\big)$ for which admissibility and hence feasibility immediately hold from the previous solution. □

The feasible set for a fixed horizon is denoted by

$$\mathcal{X}_{N_\triangle^*|\mathbb{U}} := \{x_0 \in \mathbb{X} | \exists u_\triangle(\cdot) \in \mathscr{U}_{\triangle|\mathbb{U}}^N(x_0)\}. \tag{12}$$

Further, let us define the variable horizon feasible set by

$$\mathscr{X}_{\triangle|\mathbb{U}} := \{x_0 \in \mathbb{X} | \exists N \in \mathcal{N} \text{such that } \mathscr{U}_{\triangle|\mathbb{U}}^N(x_0) \neq \emptyset\}. \tag{13}$$

Theorem 2 (Asymptotic stability): Let *Assumption 1* hold. Furthermore, let the common assumptions on MPC with terminal conditions hold (see section I). Then, equilibrium $x_{\mathrm{f}} = 0$ is asymptotically stable for closed-loop system (6) if the implicit control law results from OCP (11).

Proof: For brevity, the proof assumes that the reader is already familiar with standard MPC stability results and hence omits the detailed mathematical exposition. *Assumption 1* ensures that the feasible set $\mathscr{X}_{\triangle|\mathbb{U}}$ is non-empty and that the current state $x_\mu(n)$ is contained. Two conditions ensure asymptotic stability: $\alpha_1(\|x_\mu(n)\|) \leq V_{N_\triangle^*}\big(x_\mu(n)\big) \leq \alpha_2(\|x_\mu(n)\|)$ must hold for all $x_\mu(n) \in \mathscr{X}_{\triangle|\mathbb{U}}$ and $\alpha_1, \alpha_2 \in \mathcal{K}_\infty$. This is already fulfilled by assumption. The second condition requires $V_{N_\triangle^*}(x_\mu^+) - V_{N_\triangle^*}\big(x_\mu(n)\big) \leq -\gamma\ell\big(x_\mu(n), \mu\big(x_\mu(n)\big)\big)$. Consider the following cases:

$N^* > 1$: As the horizon is variable, the solution for $N^* - 1$ is embedded in the solution space and hence the second condition can be ensured trivially (as it follows from

the dynamic programming principle). Note, the trajectory is indeed optimal with respect to the SDOF blocking pattern. However, since the optimizer is allowed to choose any feasible $N \in \mathcal{N}$, the previous solution might be further improved by new least-cost solutions explored online or the $N^* - 1$ solution applies as fallback.

$N^* = 1$: By design, the closed-loop system is able to reach the terminal set $\mathbb{X}_\mathrm{f}$ within a single step. Herein, the stability theory of standard MPC with stabilizing conditions applies. For $x_\mu(n) \in \mathbb{X}_\mathrm{f}$, the optimizer either provides the solution $\mu(x_\mu(n)) := u^*_\triangle(0) = Kx_\mu$ or a solution which leads to greater reduction in the final costs. In both cases equation (4) and $x_\mu(n+1) \in \mathbb{X}_\mathrm{f}$ hold. □

Remark 1 (Decreasing influence of move-blocking): Notice, the proposed strategy starts with a sub-optimal performance (due to blocking) compared to classic MPC as the system is controlled subject to constraints with a single degree of freedom (for $N^* > 1$). Close to the terminal set $\mathbb{X}_\mathrm{f}$ ($N^* \to 1$), the closed-performance increases since the number of blocked inputs decreases. For $N^* = 1$ blocking does not apply anymore. From a practical point of view, it is advantageous if $\mathbb{X}_\mathrm{f}$ is large to limit performance losses. Ensuring *Assumption 1* is facilitated by defining rather simple constraint sets (e.g., box-constraints).

The definition of admissibility includes the property that the predicted final state ensures $x(N; x_0) \in \mathbb{X}_\mathrm{f}$. According to the formulation in [19], an approximation by a convex polytope is given by:

$$\mathscr{P}_\mathrm{T} := \{x \in X | Zx \leq \tau\} \tag{14}$$

with $Z \in \mathbb{R}^{s \times p}$, $\tau \in \mathbb{R}^s$ for some $s \in \mathbb{N}$ and $\mathscr{P}_\mathrm{T} \subseteq \mathbb{X}_\mathrm{f}$. Here, the distance between the final state and s-hyperplanes is determined for every component.

Remark 2 (Pre-Calculation of the terminal region): Solving (5) with derivative-based optimization algorithms requires a continuous distance metric which measures progress towards $\mathbb{X}_\mathrm{f}$. Hence, for any practical application it is crucial to determine $\mathbb{X}_\mathrm{f}$ or its approximations (Z, τ) in advance. This is difficult as discussed in [15]. The steady state $(x_\mathrm{f}, u_\mathrm{f})$ could be subject to change during operation.

Remark 3 (Mixed-Integer formulation): Solving (11) with standard solvers is computationally demanding since N is an integer. Mixed-integer programming deals with such problems. The most straightforward approach is to iteratively refine N with an outer optimization loop.

IV. DERIVATIVE-FREE MPC

This section proposes an approach with derivative-free optimization which deals with the drawbacks discussed in *Remark 2* and *Remark 3*. For illustration reasons, only one input is considered. If the dynamic system at hand has an input dimension $m > 1$, the following procedure can be repeated. The key idea is to maintain a finite set approximation of the input set $\mathbb{U}$ during runtime [20]. Since $\mathbb{U}$ is compact and a strict subset of U, lower and upper bounds, $u_\mathrm{min} := \min \mathbb{U}$ and $u_\mathrm{max} := \max \mathbb{U}$ respectively, exist. We derive the finite set approximation in two steps. First, linear sampling with step width $\Delta u \in \mathbb{R}$ leads to the finite subset $\mathbb{D} \subset \mathbb{U}$:

$$\mathbb{D} := \{u_\mathrm{min}, u_\mathrm{min} + \Delta u, u_\mathrm{min} + 2\Delta u, \ldots, u_\mathrm{max}\}. \tag{15}$$

Notice, by choosing $\mathbb{D}$, the control performance would suffer as long as Δu is not extremely small. However, decreasing Δu results in a very large cardinality. The assumption (observation) is that the difference between two consecutive control signals $\mu\big(x_\mu(n)\big) - \mu\big(x_\mu(n-1)\big)$ is small (at least for a constant reference). Hence, to overcome the large cardinality problem, we secondly impose this prior knowledge by reshaping $\mathbb{D}$ online at time instance n to enforce a dense distribution in the neighborhood of value $\mu\big(x_\mu(n-1)\big)$. We propose the mapping $\eta(\mu_{-1})$ with $\eta : \mathbb{U} \to \mathbb{A}_n$ and $\mu_{-1} := \mu\big(x_\mu(n-1)\big)$. Hence, this function $\eta(\cdot)$, which is based on two polynomials, maps every element $u_\mathrm{l} \in \mathbb{D}$ to a new (quasi-continuous) set $\mathbb{A}_n \subset \mathbb{U}$ with same cardinality:

$$\eta(\mu_{-1}) = \begin{cases} p_1(\mu_{-1})u_\mathrm{l}^r + p_2(\mu_{-1})u_\mathrm{l} + p_3(\mu_{-1}) & \text{if } u_\mathrm{l} \geq u_\mathrm{h} \\ p_4(\mu_{-1})u_\mathrm{l}^r + p_5(\mu_{-1})u_\mathrm{l} + p_6(\mu_{-1}) & \text{otherwise} \end{cases} \tag{16}$$

Here, the exponent is chosen to be $r \in \mathbb{N}$ and $u_\mathrm{h} \in \mathbb{D}$ is the point which is mapped to $\mu_{-1} \in \mathbb{A}_n$. Hence, the previous applied input value μ_{-1} is always included in the online determined set $\mathbb{A}_n$. To obtain the parameters $p_1(\mu_{-1}), p_2(\mu_{-1}), ..., p_6(\mu_{-1})$ at time instance n, the following system of linear equations must be solved twice online:

$$\begin{bmatrix} \bar{u}^r & \bar{u} & 1 \\ u_\mathrm{h}^r & u_\mathrm{h} & 1 \\ r u_\mathrm{h}^{(r-1)} & 1 & 0 \end{bmatrix} \begin{bmatrix} p_\mathrm{a} \\ p_\mathrm{b} \\ p_\mathrm{c} \end{bmatrix} = \begin{bmatrix} \bar{u} \\ \mu_{-1} \\ \varphi \end{bmatrix} \tag{17}$$

with $\varphi \in \mathbb{R}_0^+$. For the left polynomial the setup is $\bar{u} = u_\mathrm{min}, p_\mathrm{a} = p_4, p_\mathrm{b} = p_5, p_\mathrm{c} = p_6$ and for the right polynomial it is $\bar{u} = u_\mathrm{max}, p_\mathrm{a} = p_1, p_\mathrm{b} = p_2, p_\mathrm{c} = p_3$. The value u_h is determined by $u_\mathrm{h} = (u_\mathrm{min} + u_\mathrm{max})/2$. The composite function $\eta(\cdot)$ is designed to be monotonously rising between u_min and u_max to generate a set $\mathbb{A}_n$ with unique elements. Only in the case of $\mu_{-1} = u_\mathrm{min}$ or $\mu_{-1} = u_\mathrm{max}$ the set $\mathbb{A}_n$ has redundant elements which can be skipped while testing. The point u_h constitutes an inflection point (saddle point if $\varphi = 0$) and enables a local finer discretization. At the very first time instance $n = 0$ the set $\mathbb{A}_0$ can be set to $\mathbb{A}_0 = \mathbb{D}$ or determined with $\mu_{-1} = 0$. Figure 1 provides a graphical example for different previous applied control inputs μ_{-1}.

The OCP with respect to time-variant input sets $\mathbb{A}_n$ is defined as (*Derivative-free SDOF MPC formulation*):

$$\tilde{V}_{N^*_\triangle}\big(x_\mu(n)\big) = \min_{u_\triangle(\cdot) \in \mathscr{U}^N_{\triangle|\mathbb{A}_n}\left(x_\mu(n)\right), N} J_N\big(x_\mu(n), u_\triangle(\cdot)\big). \tag{18}$$

Since $\mathbb{A}_n \subset \mathbb{U}$ for all $n = 0, 1, ..., N$, the question arises whether this property is sufficient to guarantee recursive feasibility and asymptotic stability for the closed-loop system.

Proposition 1: Modify *Assumption 1* such that $\mathscr{U}^N_{\triangle|\mathbb{A}_0}(x_0) \neq \emptyset$ replaces $\mathscr{U}^N_{\triangle|\mathbb{U}}(x_0)$ at time instance $n = 0$. Further, a local state space controller (dual-mode) with $\mu\big(x_\mu(n)\big) = Kx_\mu(n)$ takes over as soon as the horizon reduces to $N^* = 1$ and $x_\mu(n) \in \mathbb{X}_\mathrm{f}$. The nonlinear

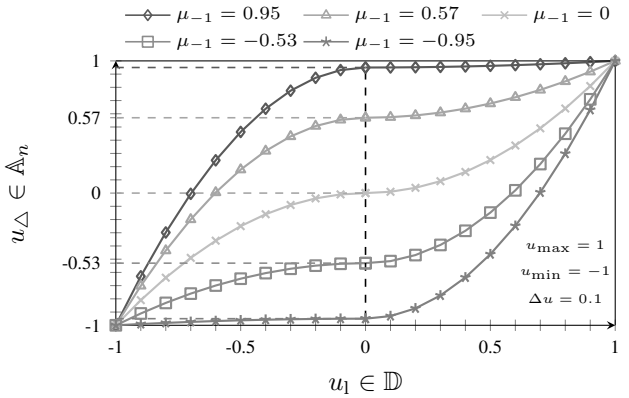

Fig. 1. Adaptive discretization with $r = 2$, $\varphi = 0$, $u_\mathrm{h} = 0$. The time-varying mapping $\eta(\cdot)$ enables a quasi-continuous input set.

system (1) which is driven by the solutions to OCP (18) ensures recursive feasibility and asymptotic stability.

Proof: Assumption 1 ensures that a single input sample at $n = 0$ steers the system into the terminal set $\mathbb{X}_\mathrm{f}$. Recursive feasibility and asymptotic stability for $N > 1$ are maintained as the adaptive discretization ensures that the previous input μ_{-1} is within the current set $\mathbb{A}_n$. In case of $N^* = 1$ the dual-mode strategy applies $\mu\big(x_\mu(n)\big) = Kx_\mu(n)$ which leads (by design) to recursive feasibility and asymptotic stability. □

Remark 4 (Disturbance in control for $N^* = 1$): The use of the dual-mode strategy in *Proof 3* is only necessary to provide a theoretical framework. Due to the sampling-based optimization with $\Delta u > 0$ a deviation between $u^*_\triangle(\cdot) \in \mathscr{U}^N_{|\mathbb{U}}(x_0)$ and $u^*_\triangle(\cdot) \in \mathscr{U}^N_{|\mathbb{A}_n}(x_0)$ occurs for $N^* = 1$ and $x_\mu(n) \in \mathbb{X}_\mathrm{f}$. For a sufficiently small step width $\Delta u \to 0$ in combination with the adaptive discretization, the assumption is that the deviation is negligible for practical applications. However, to guarantee asymptotic stability for the case $x_\mu(n) \in \mathbb{X}_\mathrm{f}$ it is possible to include the sample $Kx_\mu(n)$ into the input set with $\tilde{\mathbb{A}}_n = \big\{\mathbb{A}_n, Kx_\mu(n)\big\}$.

Algorithm 1 illustrates how to solve the mixed-integer OCP (18) in a derivative-free way subject to *Remark 4*. Notably, Algorithm 1 includes the optimization of N into a single roll-out starting at line 4, which is intractable with standard solvers. In lines 12-13 the current roll-out immediately terminates if state constraints are violated. Line 14 checks if the terminal set is already reached. Practical implementations favor the approximations (14) for $\mathbb{X}_\mathrm{f}$. If the terminal set is reached in line 14-17, the corresponding control value, the full costs and required horizon length are appended to the set of feasible candidates. Finally, the least-cost solution is selected in line 19 and the algorithm returns the input sequence $u^*_\triangle(\cdot)$ which immediately follows from $u^*_\triangle$ and horizon N^*.

Remark 5: For performance reasons, Algorithm 1 provides a sub-optimal solution to OCP (18) since it fixes $N^* = 1$ as soon as the terminal region is reached. Theoretical results still hold. However, modifying the algorithm to find the optimal solution in line 14 is straightforward but does not lead to a large increase in performance.

($\star$) *Optional Extension* (*Online determination of* $\mathbb{X}_\mathrm{f}$). In line 14 it can either be tested whether the next predicted state vector is within the terminal polytope $x(k+1) \in \mathscr{P}_\mathrm{T}$. This test is also suitable for derivative-based algorithms but requires the pre-calculation of these regions offline. An interesting property of Algorithm 1 is the fact that the terminal set $\mathbb{X}_\mathrm{f}$ can be determined online. An extended version of the algorithm 1 applies the local control law to the nonlinear system dynamics and verifies the Lyapunov conditions (4). The local control law $\mu_\mathrm{f}\big(x(n)\big) = Kx(n)$ is applied for all $n > k$ and every input candidate until either a specified convergence criterion is reached or constraints are violated (forward invariance). To limit the additional computational effort, we only simulate the system with $\mu_\mathrm{f}(x(n))$ towards a ball $\mathscr{B}_\mathrm{T} := \{x \in \mathbb{X}_\mathrm{f} | \|x\| \leq \alpha_{\mathscr{B}_\mathrm{T}}\}$ with some small $\alpha_{\mathscr{B}_\mathrm{T}} \in \mathbb{R}^+_0$ for which we assume that asymptotic stability holds (at least practical stability applies). For time-varying reference pairs $\big(x_\mathrm{f}(n), u_\mathrm{f}(n)\big)$, linearization and the solution to the Riccati equation must be performed online in line 2.

Algorithm 1 Procedure to solve OCP (18)

1: **procedure** SOLVEOCP($x_\mu(n), \mu\big(x_\mu(n-1)\big)$)
2: $\{K, P\} \leftarrow$ (Optional: Solve Riccati equation online) ▷ ($\star$)
3: $F \leftarrow \emptyset$ ▷ Initialize set of feasible candidates
4: **for all** candidates $u_\mathrm{l} \in \mathbb{D}$ **do** ▷ Parallelizable
5: $u_\triangle \leftarrow \mu\big(x_\mu(n-1)\big)$ ▷ see (16)
6: $k \leftarrow 0$
7: $x(0) \leftarrow x_\mu(n)$
8: $J \leftarrow 0$
9: **while** $k < N_\mathrm{max}$ **do**
10: $J \leftarrow J + \ell\big(x(k), u_\triangle\big)$ ▷ see (7)
11: $x(k+1) \leftarrow f\big(x(k), u_\triangle\big)$ ▷ see (1)
12: **if** $x(k+1) \notin \mathbb{X}$ **then**
13: break
14: **else if** $x(k+1) \in \mathbb{X}_\mathrm{f}$ **then** ▷ ($\star$)
15: $J \leftarrow J + V_\mathrm{f}\big(x(k+1)\big)$ ▷ V_f depends on P, see (8)
16: F.append($(u_\triangle, J, k+1)$) ▷ Add feasible tuple
17: break
18: $k \leftarrow k + 1$
19: $(u^*_\triangle, \tilde{V}_{N^*_\triangle}\big(x_\mu(n)\big), N^*) \leftarrow$ Select least-cost tuple from F
20: **return** $u^*_\triangle(\cdot)$

V. EVALUATION AND ANALYSIS

This section evaluates the performance and the computational effort of the proposed SDOF MPC with several configurations. The discrete-time system formulation results from sampling a continuous-time ordinary differential equation

$$\dot{\varphi}(t) = g\big(\varphi(t), v(t)\big) \tag{19}$$

with the continuous and in its first argument Lipschitz vector field $g : \mathbb{R}^p \times \mathbb{R}^m \to \mathbb{R}^p$. We assume a sufficiently small sampling time $T \in \mathbb{R}^+$ and choose an input parameterization with piece-wise constant inputs, i.e., $\tilde{u}(t) := u(k)$ for all $t \in [t_k, t_{k+1})$ with $t_k \in \mathbb{R}^+_0$ and $t_{k+1} = t_k + T$ for $k = 0, 1, ..., N-1$. The exact and unique solution of (19) with $\tilde{u}(t)$ as input is denoted by $\phi\big(t; x_0, \tilde{u}(t)\big)$ with $\varphi(t_0) = x_0$ for $k = 0, 1, ..., N-1$. Thus, (1) follows from (19) by

$$x^+ = f\big(x(k), u(k)\big) := \phi\big(T; \varphi(t_k), u(k)\big). \tag{20}$$

For more details on sampled-data systems refer to [15], [23]. The following analysis considers the Van-der-Pol-Oscillator (VDP) which is a common benchmark system in

the literature and its dimensions match to the system classes discussed in section I. The system's nonlinear second-order differential equation is given by

$$\ddot{\varphi}_1(t) + (\varphi_1^2(t) - 1)\dot{\varphi}_1(t) + \varphi_1(t) = v(t). \tag{21}$$

A possible state-space representation with $\varphi(t) = [\varphi_1(t), \varphi_2(t)]^\intercal = [\varphi_1(t), \dot{\varphi}_1(t)]^\intercal$ is

$$\dot{\varphi}(t) = \begin{bmatrix} \dot{\varphi}_1(t) \\ -(\varphi_1^2(t) - 1)\dot{\varphi}_1(t) - \varphi_1(t) + v(t) \end{bmatrix}. \tag{22}$$

The nonlinear discrete-time state space for prediction (20) is generated online by solving numerically the underlying initial value problem with 4^{th}-order Runge-Kutta (RK4) and $T = 2^{-7}$s. For the design of the terminal costs and terminal set, the discrete-time Riccati equation requires a linear model $x(k+1) = Ax(k) + Bu(k)$ which follows from linearizing the sampled-data formulation (20) with $A = (\partial f(\cdot)/\partial x)(0,0), B = (\partial f(\cdot)/\partial u)(0,0)$. The following list summarizes all related parameters and sets:

$$A \approx \begin{bmatrix} 1.0000 & 0.0078 \\ -0.0078 & 1.0078 \end{bmatrix}, B \approx \begin{bmatrix} 0.0000 \\ 0.0078 \end{bmatrix}, Q = \begin{bmatrix} 1.0 & 0.0 \\ 0.0 & 0.1 \end{bmatrix}$$
$$R = 0.1, P \approx \begin{bmatrix} 122.98 & 29.80 \\ 29.80 & 46.00 \end{bmatrix}, K \approx \begin{bmatrix} 2.27 \\ 3.55 \end{bmatrix}^\intercal, \begin{matrix} u_{\min} = -1 \\ u_{\max} = 1 \end{matrix},$$
$$r = 2, \mathbb{X} = \{x \in \mathbb{R}^2 | [-1,-1]^\intercal \leq x \leq [1,1]^\intercal\},$$
$$\mathbb{U} = \{u \in \mathbb{R} | u_{\min} \leq u \leq u_{\max}\}, \alpha_{\mathscr{B}_\mathrm{T}} = 0.1, \gamma = 0.9.$$

The first part is concerned with the qualitative analysis of the proposed SDOF MPC scheme with variable horizon (VH) and derivative-free (DF) optimization. Figure 2 shows several approximations of the feasible sets for different configurations, the terminal region approximation and the closed-loop phase portrait of the VDP with two example solutions. In case of the DF realizations, the finite input set is obtained with $\Delta u = 2/30$ (31 candidates). The feasible and terminal sets are determined by simulating and testing solutions on a predefined grid followed by fitting proper convex but inscribed hulls for the terminal set and concave hulls for the feasible sets. Recapitulate, the polygon $\mathscr{P}_\mathrm{T}$ describes almost all states from which the terminal set approximation $\mathscr{B}_\mathrm{T}$ can be reached without violating any constraints. Reference MPC schemes are realized with an own Matlab MPC framework in which the well-known iterative solver Interior Point OPTimizer (IPOPT) is embedded [24]. A simple outer loop which tests all horizons $N \in \mathcal{N}$ solves the mixed-integer problem taking into account *Remark 5*. First, we compare the feasible sets of conventional MPC with full ($\mathscr{P}_{\mathrm{f},1}$) and single degree ($\mathscr{P}_{\mathrm{f},2}$) of freedom with a fixed horizon ($N = 128$). Since the algorithm must reach the terminal region with a constant horizon and input, the feasible area of SDOF MPC is smaller ($\mathscr{P}_{\mathrm{f},2} \subset \mathscr{P}_{\mathrm{f},1}$) but still large. Now, we determine the feasible set for SDOF MPC and the derivative-free (DF) scheme ($\mathscr{P}_{\mathrm{f},3}$) with fixed horizon. It can be seen that the feasible set is not visibly affected by the discretization ($\mathscr{P}_{\mathrm{f},3} \subseteq \mathscr{P}_{\mathrm{f},2}$). By utilizing a variable horizon ($N_{\max} = 128$) and a single degree of freedom, the feasible region increases ($\mathscr{P}_{\mathrm{f},4} \subseteq \mathscr{X}_{\triangle|\mathbb{U}}$). The next

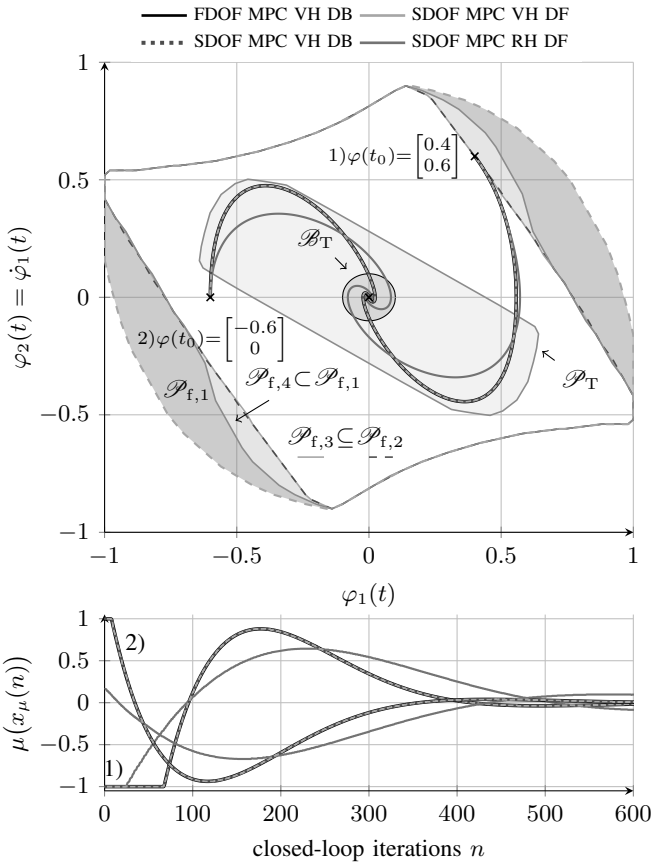

Fig. 2. Closed-loop performance. Top: Phase portrait of VDP with two closed-loop solutions and feasible respectively terminal sets. Bottom: Control input corresponding to the upper plot for two different initial values. Legend: F/SDOF: Full/Single Degree Of Freedom, RH: Receding Horizon, VH: Variable Horizon DB: Derivative-Based, DF: Derivative-Free.

section evaluates the closed-loop performances of different configurations. The conventional MPC scheme (derivative-based DB) with variable horizon and full input set $\mathbb{U}$ serves as reference controller. For full degree of freedom refer to FDOF MPC VH DB and for the single degree of freedom refer to SDOF MPC VH DB. The performance is compared to the derivative-free scheme with a single degree of freedom (SDOF MPC VH DF). The resulting trajectories of SDOF MPC VH DB and DF are identical which also applies to the input sequences. Notably, for this benchmark configuration, this fact experimentally confirms our discussion in *Remark 4* regarding the negligible negative impact of the proposed discretization. Considering the first and second trajectory and input sequence 1), 2), there is no significant difference in terms of the closed-loop performance between conventional FDOF MPC DB VH and SDOF MPC DB(F) VH. In fact, the control actions for this benchmark system and the chosen weighting matrices are fairly simple (minimum or maximum input) to reach the terminal set and as the horizon reduces to one during closed-loop evolution, SDOF MPC DB(F) VH devolves nearly to the local control law in $\mathscr{P}_\mathrm{T}$. Figure 2 also presents the closed-loop performance of the controller with a single degree of freedom and a receding horizon (RH) from our previous work [20] (SDOF MPC RH DF, $N = 70$).

TABLE I

VAN-DER-POL OSCILLATOR OCP COMPUTATION TIMES IN C. SDOF: SINGLE DEGREE OF FREEDOM, RH: RECEDING HORIZON, VH: VARIABLE HORIZON, DF: DERIVATIVE-FREE, RK4: 4^{th}-ORDER RUNGE-KUTTA, FE: FORWARD EULER, (MEDIAN; 0.95-QUANTILE)

Terminal region approach	SDOF RH DF RK4 $N = 85$, $\Delta u = \frac{2}{30}$	SDOF RH DF RK4 $N = 85$, $\Delta u = \frac{2}{100}$	SDOF VH DF RK4 $N_{\max} = 128$, $\Delta u = \frac{2}{30}$	SDOF VH DF RK4 $N_{\max} = 128$, $\Delta u = \frac{2}{100}$,	SDOF VH DF FE $N_{\max} = 128$, $\Delta u = \frac{2}{20}$
$x(N; x_0) \in \mathscr{P}_{\mathrm{T}}$	$(119\ \mu\mathrm{s}; 136\ \mu\mathrm{s})$	$(363\ \mu\mathrm{s}; 420\ \mu\mathrm{s})$	$(750\ \mu\mathrm{s}; 813\ \mu\mathrm{s})$	$(2457\ \mu\mathrm{s}; 2550\ \mu\mathrm{s})$	$(462\ \mu\mathrm{s}; 522\ \mu\mathrm{s})$
Online check	$(138\ \mu\mathrm{s}; 172\ \mu\mathrm{s})$	$(414\ \mu\mathrm{s}; 471\ \mu\mathrm{s})$	$(321\ \mu\mathrm{s}; 352\ \mu\mathrm{s})$	$(1061\ \mu\mathrm{s}; 1132\ \mu\mathrm{s})$	$(136\ \mu\mathrm{s}; 142\ \mu\mathrm{s})$

In this particular example stability and feasibility are only ensured implicitly, but the control performance is inferior. The second part of this section evaluates computation times. As analyzed and reported in [10], computation times for a small-scale system (also VDP) and a horizon of $N = 85$ range from approximately $10\,\mathrm{ms}$ up to several hundreds of milliseconds in case of FDOF MPC DB. In comparison, Table I lists computation times for SDOF MPC RH/VH DF and two integration schemes. The initial state is set according to 1) in Figure 2. Measurements are obtained with 1000 repetitions to the solution of the initial OCP ($\mathbb{A}_0$ with $\mu_{-1} = 0$) based on a single-threaded C implementation (generated with Matlab Coder) of Algorithm 1 (PC Intel i5, 3.4 GHz, Visual Studio 2015). The optimal solution including the online determination of the terminal set is available after a few hundred of microseconds. For the variable horizon approach the online determination of the terminal region is even faster than checking whether the terminal state is inside the polygon $\mathscr{P}_{\mathrm{T}}$. Faster computation times can be expected when utilizing parallelization of line 4 in Algorithm 1. The parallelization addresses the curse of dimensionality when applying the proposed scheme for multi-input systems.

VI. CONCLUSION AND FUTURE WORK

The proposed formulation with a variable horizon ensures recursive feasibility and asymptotic stability under reasonable assumptions. The straight-forward to implement, derivative-free scheme facilitates its actual practical realization. Optionally, the proposed algorithm can determine the control invariant terminal set online, which is intractable with standard derivative-based solvers. Due to the limited degrees of freedom, the approach focuses primarily on small-scale up to middle-scale, albeit fast systems. The comparative analysis on a nonlinear benchmark system highlights the potential, fast computation times, high closed-loop performance, and a large feasible set. Future work focuses on the analysis of inherent robustness in cases of model mismatch and small disturbances. Further, the evaluation of the proposed control concept for time-varying reference states in combination with online-determined terminal sets is of particular interest.

REFERENCES

[1] M. Morari and J. H. Lee, "Model predictive control: past, present and future," *Computers & Chemical Engineering*, vol. 23, no. 4-5, pp. 667–682, 1999.

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

[3] M. Diehl, H. Bock, J. P. Schlöder, R. Findeisen, Z. Nagy, and F. Allgöwer, "Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations," *Journal of Process Control*, vol. 12, no. 4, pp. 577–585, 2002.

[4] M. Diehl, H. G. Bock, and J. P. Schlöder, "A real-time iteration scheme for nonlinear optimization in optimal feedback control," *SIAM Journal on Control and Optimization*, vol. 43, no. 5, pp. 1714–1736, 2005.

[5] T. Englert, A. Völz, F. Mesmer, S. Rhein, and K. Graichen, "A software framework for embedded nonlinear model predictive control using a gradient-based augmented lagrangian approach (GRAMPC)," *Optimization and Engineering*, vol. 20, no. 3, pp. 769–809, 2019.

[6] Y. Wang and S. Boyd, "Fast model predictive control using online optimization," *IEEE Transactions on Control Systems Technology*, vol. 18, no. 2, pp. 267–278, 2010.

[7] G. Frison, D. Kouzoupis, J. B. Jorgensen, and M. Diehl, "An efficient implementation of partial condensing for nonlinear model predictive control," in *IEEE Conference on Decision and Control (CDC)*, 2016.

[8] B. Houska, H. J. Ferreau, and M. Diehl, "ACADO toolkit-an open-source framework for automatic control and dynamic optimization," *Optimal Control Applications and Methods*, vol. 32, no. 3, pp. 298–312, 2010.

[9] R. Verschueren, G. Frison, D. Kouzoupis, N. van Duijkeren, A. Zanelli, R. Quirynen, and M. Diehl, "Towards a modular software package for embedded optimization," *IFAC-PapersOnLine*, vol. 51, no. 20, pp. 374–380, 2018.

[10] C. Rösmann, M. Krämer, A. Makarow, F. Hoffmann, and T. Bertram, "Exploiting sparse structures in nonlinear model predictive control with hypergraphs," in *IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, 2018.

[11] R. Cagienard, P. Grieder, E. Kerrigan, and M. Morari, "Move blocking strategies in receding horizon control," *Journal of Process Control*, vol. 17, no. 6, pp. 563–570, 2007.

[12] H. Michalska and D. Mayne, "Robust receding horizon control of constrained nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 38, no. 11, pp. 1623–1633, 1993.

[13] H. Chen and F. Allgöwer, "A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability," *Automatica*, vol. 34, no. 10, pp. 1205–1217, 1998.

[14] D. Mayne, "An apologia for stabilising terminal conditions in model predictive control," *International Journal of Control*, vol. 86, no. 11, pp. 2090–2095, 2013.

[15] L. Grüne and J. Pannek, *Nonlinear Model Predictive Control*. Springer London, 2011.

[16] A. Richards and J. P. How, "Robust variable horizon model predictive control for vehicle maneuvering," *International Journal of Robust and Nonlinear Control*, vol. 16, no. 7, pp. 333–351, 2006.

[17] R. C. Shekhar and J. M. Maciejowski, "Robust variable horizon MPC with move blocking," *Systems & Control Letters*, vol. 61, no. 4, pp. 587–594, 2012.

[18] R. Gondhalekar and J. Imura, "Least-restrictive move-blocking model predictive control," *Automatica*, vol. 46, no. 7, pp. 1234–1240, 2010.

[19] R. C. Shekhar and C. Manzie, "Optimal move blocking strategies for model predictive control," *Automatica*, vol. 61, pp. 27–34, 2015.

[20] A. Makarow, M. Keller, C. Rösmann, and T. Bertram, "Model predictive trajectory set control with adaptive input domain discretization," in *American Control Conference (ACC)*, 2018.

[21] J. B. Rawlings, D. Q. Mayne, and M. M. Diehl, *Model Predictive Control-Theory, Computation, and Design*, 2nd ed. Nob Hill Publishing, 2017.

[22] L. Magni, G. Nicolao, L. Magnani, and R. Scattolini, "A stabilizing model-based predictive control algorithm for nonlinear systems," *Automatica*, vol. 37, no. 9, pp. 1351–1362, 2001.

[23] D. Nesic and A. Teel, "A framework for stabilization of nonlinear sampled-data systems based on their approximate discrete-time models," *IEEE Transactions on Automatic Control*, vol. 49, no. 7, pp. 1103–1122, 2004.

[24] A. Wächter and L. T. Biegler, "On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, 2006.