Abstract—In this paper, we present a quasi infinite horizon nonlinear model predictive control (MPC) scheme for tracking of generic reference trajectories. This scheme is applicable to nonlinear systems, which are locally incrementally stabilizable. For such systems, we provide a reference generic offline procedure to compute an incrementally stabilizing feedback with a continuously parameterized quadratic quasi infinite horizon terminal cost. This generic terminal cost can be used to extend (various) existing MPC approaches that deal with online changing operating conditions to nonlinear systems.

Index Terms—Nonlinear model predictive control, Constrained control, Reference tracking, Incremental Stability

I. INTRODUCTION

Model Predictive Control (MPC) [1] is a well established control method, that computes the control input by repeatedly solving an optimization problem online. The main advantages of MPC are the ability to cope with general nonlinear dynamics, hard state and input constraints, and the inclusion of performance criteria. In MPC (theory), recursive feasibility and closed-loop stability of a desirable setpoint are usually ensured by including suitable terminal ingredients (terminal set and terminal cost) in the optimization problem [2].

In many applications, the control goal goes beyond the stabilization of a pre-determined setpoint. These practical challenges include tracking of changing reference setpoints, stabilization of dynamic trajectories, output regulation and general economic optimal operation. There exist many promising ideas to tackle these issues in MPC, for example by simultaneously optimizing an artificial reference [3], [4], [5], [6], [7]. However, most of these approaches are limited in some form to linear systems. The computation of suitable terminal ingredients seems to be a bottleneck for the practical extension of these methods to nonlinear systems. We bridge this gap, by providing a reference generic offline computation for the terminal ingredients. Thus, we can provide practical schemes for nonlinear systems subject to changing operating conditions.

Related work

For linear stabilizable systems, a terminal set and terminal cost can be computed based on the linear quadratic regulator (LQR) and the maximal output admissible set [8]. For the purposes of stabilizing a given setpoint, a suitable design procedure for nonlinear systems with a stabilizable linearization has been provided in [9], [1].

In practice, the setpoint to be stabilized can change and thus procedures independent of the setpoint are necessary. In [10], the issue of finding a setpoint independent terminal cost has been investigated based on the concept of pseudo linearizations. While in principle very appealing, the computation of such a pseudo linearization for general nonlinear systems seems impractical. In [11], a locally stabilizing controller is assumed and the terminal cost and constraints are defined implicitly based on the infinite horizon tail cost. The main drawback of this method is the implicit description of the terminal cost, which can significantly increase the online computational demand. In [12] the feasible setpoints are partitioned into disjoint sets and for each such set a fixed stabilizing controller and terminal cost are designed using the methods in [13], [14] based on a local linear time-varying (LTV) system description. This method is mainly limited to systems with a one dimensional steady-state manifold, due to the otherwise complex and difficult partitioning. The piece-wise definition can also lead to numerical difficulties since the terminal cost is not differentiable with respect to the setpoint.

Furthermore, there are many applications in which we want to stabilize some dynamic trajectory or periodic orbit. The nonlinear system along this trajectory can be locally approximated with a LTV system. In [15], this is used to compute a (time-varying) terminal cost for asymptotically constant trajectories. In [16] periodic trajectories are considered and a (periodic) terminal cost is computed based on linear matrix inequalities (LMIs). Both of these methods have short comings, including a computationally expensive design procedure and the restriction to certain classes of trajectories. But more than this, a significant practical restriction is that the offline computation is accomplished for a specific trajectory.

In general, the existing procedures to compute terminal ingredients for MPC are focused on computing a terminal cost for a specific reference point or reference trajectory. Thus, online changes in the setpoint or trajectory cannot be handled directly and necessitate repeated offline computations.

Contribution

In this work, we provide a reference generic offline procedure to compute a parameterized terminal cost. This procedure is applicable to both setpoint or trajectory stabilization. The feasibility of this approach requires local incremental
stabilizability of the nonlinear dynamics. The existing design procedures [9], [15], [16] use the linearization around the considered setpoint or trajectory to locally establish properties of the nonlinear systems. In a similar spirit, we consider the linearization of the nonlinear system dynamics around all possible points in the constraint set and describe the dynamics analogous to quasi-linear parameter-varying (LPV) systems. With this description, we formulate the desired properties on the linearized dynamics and provide suitable LMI{s} to compute the parameter dependent terminal cost and controller. In closed-loop operation we have a quadratic terminal cost with an ellipsoidal terminal constraint directly available. This provides a generalization of the offline computations in [9], [15], [16] to generic references.

Given these terminal ingredients, it is possible to extend existing tracking MPC schemes [3], [4], [5], [7] to nonlinear system dynamics, which is a fundamental step towards practical nonlinear MPC schemes.

Outline

The remainder of this paper is structured as follows: Section II presents the reference tracking MPC scheme based on the proposed parameterized terminal ingredients. Section III provides a constructive procedure to design parametric terminal ingredients independent of the considered reference. Section IV concludes the paper.

We close this section by noting related previous work of the authors in the field of nonlinear reference tracking MPC without terminal constraints based on local incremental stabilizability [17].

II. REFERENCE TRACKING MODEL PREDICTIVE CONTROL

A. Notation

The quadratic norm with respect to a positive definite matrix $Q = Q^T$ is denoted by $\|x\|^2_Q = x^T Q x$. The minimal and maximal eigenvalue of a symmetric matrix $Q = Q^T$ is denoted by $\lambda_{\min}(Q)$ and $\lambda_{\max}(Q)$, respectively. The identity matrix is $I_n \in \mathbb{R}^{n \times n}$. The interior of a set $\mathcal{X}$ is denoted by $\text{int}(\mathcal{X})$. The vertices of a polytopic set $\Theta$ are denoted by $\theta_i \in \text{Vert}(\Theta)$.

B. Setup

We consider the following nonlinear discrete-time system

$$x(t + 1) = f(x(t), u(t))$$

with the state $x \in \mathbb{R}^n$, control input $u \in \mathbb{R}^m$, and time step $t \in \mathbb{N}$. We impose point-wise in time constraints on the state and input

$$(x(t), u(t)) \in \mathcal{Z},$$

with some compact set $\mathcal{Z}$. We consider the following assumption regarding the reference signal $r = (x_r, u_r) \in \mathbb{R}^{n+m}$.

Assumption 1. The reference signal $r$ satisfies $r(t) \in \mathcal{Z}_r$, $\forall t \geq 0$, with some set $\mathcal{Z}_r \subseteq \text{int}(\mathcal{Z})$. Furthermore, the evolution of the reference signal is restricted by $r(t + 1) \in \mathcal{R}(r(t))$, with $\mathcal{R}(r) = \{ (x^r_r, u^r_r) \in \mathcal{Z}_r \mid x^r_r = f(x_r, u_r) \}$.

Remark 1. Additional incremental input constraints $\|u_r(t + 1) - u_r(t)\|_\infty \leq \epsilon$ can be easily incorporated in $\mathcal{R}(r)$.

Setpoints can be modeled with $\mathcal{R}(r) = r$ and a corresponding set $\mathcal{Z}_r$.

C. Terminal cost and terminal set

Denote the tracking error by $e_r(t) = x(t) - x_r(t)$. The control goal is to stabilize the tracking error $e_r(t) = 0$ and achieve constraint satisfaction $(x(t), u(t)) \in \mathcal{Z}$, $\forall t \geq 0$. To this end we define the quadratic reference tracking stage cost

$$J^r(x, u, r) = \|x - x_r\|^2 + \|u - u_r\|^2,\tag{3}$$

with positive definite weighting matrices $Q, R$. As discussed in the introduction, we need suitable terminal ingredients to ensure stability and recursive feasibility for the closed-loop.

Assumption 2. There exist matrices $K_f(r) \in \mathbb{R}^{m \times n}, P_f(r) \in \mathbb{R}^{n \times n}$ with $c_l I_n \leq P_f(r) \leq c_u I_n$, a terminal set $\mathcal{X}_f(r) = \{ x \in \mathbb{R}^n \mid V_f(x, r) \leq \alpha \}$ with the terminal cost $V_f(x, r) = \|x - x_r\|^2 + \|P_f(r)\|^2$, such that the following properties hold for any $r \in \mathcal{Z}_r$, any $x \in \mathcal{X}_f(r)$ and any $r^+ \in \mathcal{R}(r)$

$$V_f(x^+, r^+) \leq V_f(x, r) - l(x, k_f(x, r), r),\tag{4}$$

$$(x, k_f(x, r)) \in \mathcal{Z},\tag{5}$$

with $x^+ = f(x, k_f(x, r))$, $k_f(x, r) = u_r + K_f(r) \cdot (x - x_r)$ and positive constants $c_l, c_u, \alpha$.

For $r = r^+ = 0$ this reduces to the standard conditions in [9]. For a given trajectory $r$, this implies time-varying terminal ingredients, compare [15], [16]. Designing suitable terminal ingredients that satisfy this assumption is the main contribution of this paper and is discussed in more detail in the Section III.

D. Preliminary results

Denote the reference $r$ over the prediction horizon $N$ by $r(|t|) \in \mathbb{R}^{n+m \times N + 1}$ with $r(k|t|) = (r(t + k), k = 0, \ldots, N$. Given a predicted state and input sequence $x(|t|) \in \mathbb{R}^{n \times N + 1}$, $u(|t|) \in \mathbb{R}^{m \times N}$ the tracking cost with respect to the reference $r(|t|)$ is given by

$$J_N(x(|t|), u(|t|), r(|t|)) := \sum_{k=0}^{N-1} l(x(k|t|), u(k|t|), r(k|t|)) + V_f(x(N|t|), r(N|t|)).$$

The MPC scheme is based on the following (standard) MPC optimization problem

$$V(x(t), r(|t|)) = \min_{u(|t|)} J_N(x(|t|), u(|t|), r(|t|))\tag{6a}$$

s.t. $x(k + 1|t|) = f(x(k|t|), u(k|t|))$, $x(0|t|) = x(t)$, $x(k|t|), u(k|t|) \in \mathcal{Z}$, $x(N|t|) \in \mathcal{X}_f(r(N|t|))$. $\tag{6b} \tag{6c} \tag{6d} \tag{6e}$

In principle, this assumption can always be satisfied with a zero-terminal constraint $\mathcal{X}_f(r) = x_r$. This would/could however lead to numerical problems. In addition, the non-empty terminal set is relevant to ensure robustness properties of the MPC scheme. Furthermore, tracking schemes such as [3], [12], also require a non-vanishing terminal set size $\alpha$. 


The solution to this optimization problem are the value function $V$ and the optimal input trajectory $u^*(\cdot|t)$. In closed-loop operation we apply the first part of the optimized input trajectory to the system, leading to the following closed-loop operation

$$x(t + 1) = f(x(t), u^*(0|t)) = x^*(1|t), \quad t \geq 0.$$  

**Theorem 1.** Let Assumptions 1 and 2 hold. Assume that Problem (6) is feasible at $t = 0$. Then Problem (6) is recursively feasible and the tracking error $e_r = 0$ is (uniformly) exponentially stable for the resulting closed-loop system (7).

**Proof.** This theorem is a straight forward extension of standard MPC results in [1], compare also [15]. Given the optimal solution $u^*(\cdot|t)$, the candidate sequence

$$u(k|t + 1) = \left\{ \begin{array}{ll} u^*(k + 1|t) & k \leq N - 2 \\ k_f(x^*(N|t), r(N|t)) & k = N - 1 \end{array} \right.,$$

is a feasible solution to (6a) and implies

$$V(x(t), r(\cdot|t + 1)) \leq V(x(t), r(\cdot|t)) - l(x(t), u(t), r(t)).$$

Compact constraints in combination with the quadratic terminal cost imply

$$l(x(t), u(t), r(t)) \leq V(x(t), r(\cdot|t)) \leq c_u l(x(t), u(t), r(t)).$$

(Uniform) Exponential stability follows from standard Lyapunov arguments based on the value function $V$.

This theorem shows that if we can design suitable terminal ingredients (Ass. 2), the closed-loop tracking MPC has all the (standard) desirable properties. This scheme can be easily modified to ensure robust reference tracking, compare [18].

**Remark 2.** A powerful alternative to the proposed quasi-infinite horizon reference tracking MPC scheme would be a reference tracking MPC scheme without terminal constraints [17] ($V_f(x, r) = 0, \lambda_f(r) = X$). If it is possible to design terminal ingredients (Ass. 2), the value function of such an MPC scheme without terminal constraints is locally bounded by $V(x(t), r(\cdot|t)) \leq \gamma l(x,u,r)$, with a suitable constant $\gamma$, compare [17, Prop. 2]. Thus, an MPC scheme without terminal constraints enjoys similar closed-loop properties to Theorem 1, provided a sufficiently large prediction horizon $N$ is used, compare [17, Thm. 2]. One of the core advantages of including suitably designed terminal ingredients is that we can implement the MPC scheme with a short prediction horizon $N$. On the other hand, if the reference is not reachable (Ass. 1), MPC schemes without terminal constraints can still be successfully applied [17, Thm. 4], which is in general not the case for terminal constraint schemes.

## III. Reference Generic Offline Computations

This section provides a reference generic offline computation to design terminal ingredients for nonlinear reference tracking MPC. In Lemma 1 we provide sufficient conditions for the terminal ingredients based on properties of the linearization. Subsequently, the connection between these conditions and incremental stabilizability is clarified in Propositions 1 and 2. Then, two approaches based on LMI computations are described to compute the terminal ingredients, based on Lemma 2 and Proposition 3. Then, a procedure to obtain a non conservative terminal set size $\alpha$ is discussed. Finally, the overall offline procedure is summarized in Algorithm 2. For the special case of setpoint tracking, existing methods are discussed in relation to the proposed procedure.

### A. Sufficient conditions based on the linearization

We define the linearization around an arbitrary point $r \in \mathcal{Z}_r$ by

$$A(r) = \left[ \frac{\partial f}{\partial x} \right]_{(x,u) = r}, \quad B(r) = \left[ \frac{\partial f}{\partial u} \right]_{(x,u) = r}.$$  

The following lemma establishes local incremental properties of the nonlinear system dynamics based on the linearization.

**Lemma 1.** Suppose that $f$ is twice continuously differentiable. Assume that there exists a matrix $K_f(r) \in \mathbb{R}^{m \times n}$ and a positive definite matrix $P_f(r) \in \mathbb{R}^{n \times n}$ continuous in $r$, such that for any $r \in \mathcal{Z}_r$, $r^+ \in \mathcal{R}(r)$, the following matrix inequality is satisfied

$$A(r) + (B(r)K_f(r))^T P_f(r^+) (A(r) + B(r)K_f(r)) \leq P_f(r) - (Q + K_f(r)^T R K_f(r)) \leq c_I n$$

with some positive constant $c$. Then there exists a sufficiently small constant $\alpha$, such that $P_f, K_f$ satisfy Assumption 2.

**Proof.** The proof is very much in line with the result for setpoints in [9], [1]. First we show satisfaction of the decrease condition (4) and then constraint satisfaction (5).

**Part 1:** Denote $\Delta x := x - x_r$ and $\Delta u := K_f(r) \Delta x$. Using a first order Taylor approximation at $r = (x_r, u_r)$, we get

$$f(x, u) = f(x_r, u_r) + A(r) \Delta x + B(r) \Delta u + \Phi_r(\Delta x),$$

with the remainder term $\Phi_r$. The terminal cost satisfies

$$V_f(x^+, r^+) = \|f(x, u) - f(x_r, u_r)\|_{P_f(r^+)}^2$$

$$\leq \|(A(r) + B(r)K_f(r))\Delta x + \Phi_r(\Delta x)\|_{P_f(r^+)}^2$$

$$\leq \|(A(r) + B(r)K_f(r))\Delta x\|_{P_f(r^+)}^2 + \Phi_r(\Delta x)\|_{P_f(r^+)}^2$$

$$\leq V_f(x, r) - c_1 \Delta x^2 - l(x, K_f(x, r), r) + \Phi_r(\Delta x)\|_{P_f(r^+)}^2$$

$$+ 2\Phi_r(\Delta x)\|_{P_f(r^+)}^2 (A(r) + B(r)K_f(r))\Delta x\|_{P_f(r^+)}.$$  

Using the continuity of $P_f(r)$, $K_f(r)$ and the compactness of the constraint set $\mathcal{Z}_r$, there exist finite constants

$$c_u = \max_{r \in \mathcal{Z}_r} \lambda_{\max}(P_f(r)), \quad c_l = \min_{r \in \mathcal{Z}_r} \lambda_{\min}(P_f(r)),$$  

$$k_u = \max_{r \in \mathcal{Z}_r} \|K_f(r)\|,$$  

$$c_{u,2} = \max_{r \in \mathcal{Z}_r} \lambda_{\max}(P_f(r)) - (c_I + Q + K_f(r)^T R K_f(r)).$$  


Suppose that the remainder term $\Phi_r$ is locally Lipschitz continuous in the terminal set with a constant $L_{\Phi,\alpha}$ satisfying
\[
\|\Phi_r(\Delta x)\| \leq L_{\Phi,\alpha} \|\Delta x\|,
\]
\[
L_{\Phi,\alpha} \leq L_{\Phi} := \sqrt{\frac{c_u,2 + \epsilon}{c_u}} - \sqrt{\frac{c_u,2}{c_u}} \tag{15}
\]
Then we have
\[
\|\Phi_r(\Delta x)\|^2 \leq T(\|\Delta x\|^2 + \|\Delta u\|^2) \leq T(1 + k^2_\alpha)\|\Delta x\|^2.
\]
Using $\|\Delta x\| \leq \sqrt{\frac{\rho}{c_l}}$ from the terminal constraint, we get (15) for all $\alpha \leq \alpha_1$ with
\[
\alpha_1 := \alpha l \left(\frac{L^2_{\Phi}}{T(1 + k^2_\alpha)}\right)^2. \tag{16}
\]

**Part II:** Constraint satisfaction: The terminal constraint $\|\Delta x\|^2 \geq \alpha$ in combination with (13), (14) implies
\[
(\Delta x, \Delta u) \in B(\alpha) = \left\{ z \in \mathbb{R}^{n+m} \mid \|z\|^2 \leq \frac{\alpha}{\alpha l} (1 + k^2_\alpha) \right\}.
\]
Given that $Z_r \subseteq \text{Int}(Z)$, there exists a small enough $\alpha_2$ such that
\[
(x, u) = r + (\Delta x, \Delta u) \subseteq Z_r \oplus B(\alpha) \subseteq Z, \quad \forall \alpha \leq \alpha_2, \tag{17}
\]

As a summary, given matrices $P_f$, $K_f$ satisfying (11), we can compute a local Lipschitz bound (15), which in turn implies a maximal size $\alpha_1$. Similarly, the constraint sets $\mathcal{Z}$ and $\mathcal{Z}_r$ in combination with $K_f$, $P_f$ imply an upper bound $\alpha_2$ to ensure constraint satisfaction. Then Assumption 2 is satisfied for any $\alpha \leq \min\{\alpha_1, \alpha_2\}$. This result is an extension of [9], [1] to arbitrary dynamic references.

**B. (Local) Incremental exponential stabilizability**

In the following we clarify the connection between incremental stabilizability properties and the terminal ingredients.

**Definition 1.** A set of reference trajectories $r$ specified by some dynamic inclusion $r(t + 1) \in \mathcal{R}(r(t))$ is locally exponentially incrementally stabilizable by the system (1), if there exists constants $\rho \in (0, 1), c, M > 0$ and a control law $\kappa(x, r)$, such that for any initial condition satisfying $\|x(0) - x_r(0)\| \leq c$ the trajectory $x(t)$ with $x(t + 1) = f(x(t), \kappa(x(t), r(t)))$ satisfies $\|x(t) - x_r(t)\| \leq M\rho^t\|x(0) - x_r(0)\|$, $\forall t \geq 0$.

This definition is closely related to the concept of universal exponential stabilizability [19], which characterizes the stabilizability of arbitrary trajectories in continuous-time. One of the core differences in the definitions is the treatment of constraints, i.e. we study stabilizability of classes of trajectories $r$ that satisfy certain constraints, compare Assumption 1 and Remark 1. This difference is crucial when discussing local versus global stabilizability.

The following proposition shows that the conditions in Lemma 1 directly imply local incremental exponential stabilizability of the reference trajectory.

**Proposition 1.** Suppose that there exist matrices $P_f(r)$, $K_f(r)$ that satisfy the conditions in Lemma 1. Then the control law $k_f(x, r) = u_r + K_f(r)(x - x_r)$ locally exponentially incrementally stabilizes any reference $r$ satisfying Assumption 1.

**Proof.** For any $\|x(0) - x_r(0)\| \leq c$ with $c = \sqrt{\alpha/l_c}$, we have $x(0) \in \mathcal{X}_f(r)$, for $\alpha$, $c_l$ according to Lemma 1. Thus, the terminal cost $V_f(x, r)$ is a local incremental Lyapunov function that satisfies
\[
V_f(x(t), r(t + 1)) \leq \rho^2 V_f(x(t), r(t)), \quad \rho^2 = 1 - \frac{\lambda_{\min}(Q)}{c_l},
\]
and thus
\[
\|x(t) - x_r(t)\| \leq \rho^t M\|x_r(0) - x(0)\|, \quad M = \sqrt{\frac{c_l}{c_l}},
\]
The proof follows the arguments of [17, Prop. 1.2]. □

**Remark 3.** This result establishes local incremental stabilizability with the incremental Lyapunov function $V_f(x, r)$ based on properties of the linearization, compare [17, Prop. 1]. This system property is a natural extension of previous works on incremental stability and corresponding incremental Lyapunov functions, see [20], [17, Ass. 1]. This property implies stabilizability of $(A(r), B(r))$ around any setpoint $r$, but it does not necessarily imply stabilizability of $(A(r), B(r))$, as $P_f(r)$ might decrease along the trajectory.

For continuous-time systems, an analogous result exists based on contraction metrics and universal stabilizability [19].

The following proposition shows that in the absence of constraints we recover non-local results similar to [19].

**Proposition 2.** Consider $Z_r = Z = \mathbb{R}^{n+m}$. Suppose that there exist matrices $P_f(r)$, $K_f(r)$ that satisfy the conditions in Lemma 1. Assume further that $c_l I \leq P_f(r) \leq c_l I$ for all $r \in \mathbb{R}^{n+m}$ with some constants $c_l$, $c_u$ and $K_f(r) = K_f(x_r)$. Then any reference $r$ satisfying Assumption 1 is exponentially incrementally stabilizable with the control law
\[
\kappa(x, r) = K_f(x)x - K_f(x_r)x_r + u_r,
\]
i.e., for any initial condition $x(0) \in \mathbb{R}^n$ the state trajectory $x(t + 1) = f(x(t), \kappa(x(t), r(t)))$ satisfies $\|x(t) - x_r(t)\| \leq M\rho^t\|x(0) - x_r(0)\|$. 

---

2We first deriving a sufficient local Lipschitz bound $L_{\Phi}$ and then obtaining a local region $\alpha_1$ (16). This procedure is in line with existing procedures [9]. Alternatively, it is possible to directly use the quadratic bound $\|\Phi_r(\Delta x)\| \leq c\|\Delta x\|^2$ and work with higher order terms to obtain $\alpha_1$, compare [17, Prop. 1].
Proof. Consider an auxiliary (pre-stabilized) system defined by \( f(x,v) = f(x,K(x)x + v) \). Consider a reference \( r \) generated by some input trajectory \( u_r \) with the system dynamics (1) (Ass. 1) and some initial condition \( x_r(0) \) resulting in the state reference \( x_r \). Now, consider a reference \( \tilde{r} \) generated by the input \( v_r(t) = u_r(t) - K(x_r(t))x_r(t) \) with the system dynamics according to \( f \) and the same initial condition. Due to the definition of the auxiliary system we have \( \tilde{x}_r(t) = x_r(t), \forall t \geq 0 \). For an arbitrary, but fixed input \( v \), stability of the reference trajectory is equivalent to contractivity of the nonlinear time-varying system \( \tilde{f}(x,t) \). This can be established with the contractivity metric \( P_f(x(t),t) = P_f(x_r(t),t) \), compare [21].

In the absence of constraints, it is crucial that \( P_f \) has a constant lower and upper bound. If the matrix \( K_f \) depends on the full reference \( r \) (not just \( x_r \)), the corresponding controller \( \kappa \) in Proposition 2 is not necessarily well defined.

Remark 4. The relation between the controller \( \kappa_f \) (Prop. 1) and \( \kappa \) (Prop. 2), is that of reference tracking versus pre-stabilization. The first is more natural in the context of tracking MPC and contains existing results for the design of terminal ingredients as special cases [9], [15], [16]. The second controller \( \kappa \) allows for non-local stability results and is more suited for unconstrained control problems [19]. For constant matrices \( K \) the two controllers are equivalent, but the incremental Lyapunov functions (and thus terminal costs) are differently parameterized (\( P_f(r), P_f(x,u) \)).

Remark 5. The problem of computing reference generic terminal ingredients is equivalent to computing an incrementally stabilizing controller and is thus strongly related to the computation of robust positive invariant (RPI) tubes in nonlinear robust MPC schemes, compare [22], [18]. For comparison, in [23], [24] constant matrices \( P_f, K_f \) are computed that certify incremental stability for continuous-time systems (by considering small Lipschitz nonlinearities or by describing the linearization as a convex combination of different linear systems). This approach can be directly extended to more general nonlinear systems using the proposed terminal ingredients. In particular, by changing the stage cost to

\[
I(x,u,r) = \| u - u_r + K_f(x_r)x_r - K_f(x)x \|^2_R
\]

one can design a nonlinear version of [25], compare also [18].

Remark 6. In case a system is not exponentially/quadratically stabilizable [26], it might be possible to make a nonlinear transformation resulting in a quadratically stabilizable system.

C. Quasi-LPV based procedure

Lemma 1 states that matrices satisfying inequality (11) also satisfy Assumption 2 with a suitable terminal set size \( \alpha \). In the following, we illustrate formulate computationally tractable problems to compute matrices that satisfy the conditions in Lemma 1.

The following Lemma transforms the conditions in (11) to be linear in the arguments.

**Lemma 2.** Suppose that there exists matrices \( X(r) \), \( Y(r) \) continuous in \( r \), that satisfy the constraints in (18) for all \( r \in Z_r \), \( r^+ \in R(r) \). Then \( P_f = X^{-1} \), \( K_f = YP_f \) satisfy (11).

**Proof.** The proof is standard, compare for example [27].

The resulting optimization problem (18) is convex, linear in \( X(r), Y(r) \) and minimizes the worst-case terminal cost \( P_f(r) \leq X_{\text{min}} \).

So far, the result is only conceptual, since (18) is an infinite optimization problem (infinite dimensional optimization variables with infinite dimensional constraints). In particular, we need a finite parameterization of \( X, Y \) and the infinite constraints need to be converted into a finite set of sufficient constraints.

One solution to this problem would be sum-of-squares (SOS) optimization [28]. Assuming \( A, B \) are polynomial, consider matrices \( X, Y \) polynomial in \( r \) (with a specified order \( d \)) and ensure that the matrix in (18) is SOS. A similar approach is suggested in [19] to find a control contraction metric for continuous-time systems (which is a strongly related problem). This approach is not pursued here since most systems require a high order polynomial to approximate the nonlinear dynamics and the computational complexity grows exponentially in \( n^d \), thus prohibiting the application to many practical systems.

We approach this problem from the perspective of quasi-LPV systems and gain-scheduling [29]. First, write the linearization (10) as

\[
A(r) = A_0 + \sum_{j=1}^{p} \theta_j(r)A_j, \quad B(r) = B_0 + \sum_{j=1}^{p} \theta_j(r)B_j,
\]

with some nonlinear (continuously differentiable) parameters \( \theta \in \mathbb{R}^p \). This can always be achieved with \( p \leq n(n+m) \). We impose the same structure on the optimization variables with

\[
X(r) = X_0 + \sum_{j=1}^{p} \theta_j(r)X_j, \quad Y(r) = Y_0 + \sum_{j=1}^{p} \theta_j(r)Y_j.
\]

Remark 7. For input affine systems of the form \( f(x,u) = f_x(x) + Bu \), we have \( P_f(r) = P_f(x_r), K_f(r) = K_f(x_r) \), i.e., the linearization (19) and correspondingly the parameters \( \theta_i \) only depend on \( x_r \).

Now (18) contains only finite optimization variables, but still needs to be verified for all \( r \in Z_r \), \( r^+ \in R(r) \). There are two options to deal with this: griding the constraint set or (conservatively) convexifying the problem.

1) Griding: A heuristic to ensure that the constraints in (18) hold for all \( (r,r^+) \) is to consider the constraints on sufficiently many sample points in the constraint set. Due to continuity, the constraint is satisfied on the full constraint set if it holds on a sufficiently fine grid. The theoretically sufficient grid size is typically too small to be computationally tractable and a corser\(^3\) grid needs to be considered. Thus, for this method it is crucial that satisfaction of (4) is verified by using a fine grid (compare Algorithm 1).

\(^3\)Using sparse grids, instead of a uniform grid can be advantages.
This gridding consists of a grid over all possible state and input combinations \( r \in \mathcal{Z}_r \) and \( r^+ \in \mathcal{R}(r) \). Note, that only combinations \((r, r^+)\) that satisfy Assumption 1 are included in the optimization problem, i.e. only combinations that satisfy

\[
r, r^+ \in \mathcal{Z}_r, \quad r^+ \in \mathcal{R}(r), \quad \mathcal{R}(r^+) \neq \emptyset. \tag{20}
\]

For the simple structure \( \mathcal{R}(r) \) in Assumption 1 this can be achieved by gridding \( r \), computing \( x^+_r = f(x_r, u_r) \), and considering all \( u^+_r \), such that \((x^+_r, u^+_r) \in \mathcal{Z}_r \) and \((f(x^+_r, u^+_r), \tilde{u}_r) \in \mathcal{Z}_r \) with some \( \tilde{u}_r \). This approach does not introduce additional conservatism, but is computationally challenging for high dimensional systems. As discussed in Remark 1 we can include additional constraints on the reference, which makes the offline computation easier.

2) Convexity: The systematic alternative to gridding the constraint set, is a (conservative) convexification of the optimization problem. In order to convexify \eqref{eq:18} we match the constraints \( \mathcal{Z}_r, \mathcal{R}(r) \) on the reference \( r \) to polytopic constraint sets \( \Theta, \Omega \) on the parameters \( \theta \). The polytopic sets \( \Theta, \Omega(\theta) \) need to satisfy

\[
\theta(r) \in \Theta, \quad \forall r \in \mathcal{Z}_r, \\
\theta(r^+) \in \Omega(\theta(r)), \quad \forall r^+ \in \mathcal{R}(r). \tag{22}
\]

Computing a set \( \Theta \), such that \( \theta(r) \in \Theta \) for all \( r \in \mathcal{Z}_r \) can be achieved by considering a conservative hyperbox \( \Theta = \{ \theta \in \mathbb{R}^p | \theta_i \in [\underline{\theta}_i, \bar{\theta}_i] \} \). For \( \Omega \), a simple approach is \( \Omega(\theta) = \{ \theta \} \oplus \Omega \), where \( \Omega \) is a simple hyperbox that encompasses the maximal change in the parameters \( \theta \) in one time step, i.e. \( \Omega = \{ \Delta \theta \in \mathbb{R}^p | \Delta \theta_i \in [\underline{\theta}_i, \bar{\theta}_i] \} \). We denote the joint polytopic constraint set by

\[
(\theta, \theta^+) \in \overline{\Theta} = \{ (\theta, \theta^+) \in \Theta \times \Theta | \theta^+ \in \{ \theta \} \oplus \Omega \}, \tag{23}
\]

which consists of \( 6p \) vertices\(^4\). The following proposition provides a simple convex procedure to compute a terminal cost, by solving a finite collection of LMIs.

**Proposition 3.** Suppose that there exists matrices \( X_\min, \ Y_i, \ L_i, \ X_{\min} \) that satisfy the constraints in \eqref{eq:21}.

\[
\min_{X(r), Y(r), X_{\min}} \log \det X_{\min}
\]

\[
\begin{align*}
\text{s.t.} \quad \begin{pmatrix} X(r) & Y(r) \end{pmatrix} & = \begin{pmatrix} A(r) & B(r) \end{pmatrix} \begin{pmatrix} X(r) & Y(r) \end{pmatrix} \\
0 & 0 \\
0 & I \\
I & 0 \\
\end{pmatrix} \geq 0, \\
X_{\min} & \preceq X(r), \\
\forall r & \in \mathcal{Z}_r, \quad r^+ \in \mathcal{R}(r).
\end{align*}
\]

Then the matrices:

\[
P_f(r) = \left( X_0 + \sum_{j=1}^{p} \theta_i(r) X_i \right)^{-1},
\]

\[
K_f(r) = \left( Y_0 + \sum_{j=1}^{p} Y_i \theta_i(r) \right) P_f(r),
\]

satisfy \eqref{eq:11}.

**Proof.** Due to Lemma 2, it suffices if \( X(r), \ Y(r) \) satisfy the constraints in \eqref{eq:18}. Due to the definition of the set \( \overline{\Theta} \) \eqref{eq:23} and \( \Lambda_i \geq 0 \), any solution that satisfies the constraints \eqref{eq:18} over all \( (\theta, \theta^+) \in \overline{\Theta} \), also satisfies the constraints for all \( r \in \mathcal{Z}_r \), \( r^+ \in \mathcal{R}(r) \). It remains to show that it suffices to check the inequality on the vertices of the constraint set \( \overline{\Theta} \). This last result is a consequence of multi-convexity \( \cite[Corollary 3.2]{Shaifub} \). In particular, if a function \( f \) is multi-convex along the edges of the constraint set \( \overline{\Theta} \), then it attains its maximum at a vertex of \( \overline{\Theta} \). Similar to \( \cite[Corollary 3.5]{Shaifub} \), the additional constraint \eqref{eq:21d} ensures multi-convexity. Thus, it suffices to verify the conditions on the vertices of the constraint set.

The advantage of this procedure compared to the gridding is that it typically scales better with the system dimension. This comes at the cost of additional conservatism due to the construction of the sets \( \Theta \). A more refined characterization of the set \( \overline{\Theta} \) can be the key to reducing the conservatism of this method. The computational demand can be reduced by considering (block-)diagonal multipliers \( \Lambda_i = \lambda_i I \). It can often be beneficial to consider a combination of the two approaches, i.e grid in some dimensions and conservatively convexify in others.

The main result is that we can formulate the offline design procedure similar to the gain scheduling synthesis of (Quasi-)LPV systems and thus can draw on a well established field to formulate offline LMI procedures, compare \cite{LMI}.

D. Non-conservative terminal set size \( \alpha \)

The terminal set size \( \alpha \) derived in Lemma 1 can be quite conservative. In the following we illustrate how a non conservative value \( \alpha \) can be computed.

\(^4\)Clearly, a conservative over approximation of this set is given by \( (\Theta) \times (\Theta \oplus \Omega) \) which has only \( 4p \) vertices. The conservatism of this approximation is negligible if \( \Omega \) is very small compared to \( \Theta \).
\[
\min_{X_r, Y_r, \Delta x, X_{\min}} \log \det X_{\min} \quad \text{s.t.} \quad \begin{pmatrix} X(\theta) & Y(\theta)^T \end{pmatrix} = 0, \quad \begin{pmatrix} X(\theta) & Y(\theta)^T \end{pmatrix} = 0, \quad (Q + \varepsilon)^{1/2} X(\theta) = (R^{1/2} Y(\theta))^T = 0, \quad (Q + \varepsilon)^{1/2} X(\theta) = (R^{1/2} Y(\theta))^T = 0, \quad \varepsilon > 0.
\]

\[\alpha_2 := \max_{\alpha} \alpha \quad \text{s.t.} \quad \| P_f(r)^{-1/2} K_f(\theta)(r) L_{r,j}^T \| \leq \sqrt{\frac{l_j - L_{r,j} r}{2}}, \quad \forall r \in \mathcal{Z}_r, \quad j = 1, \ldots, n_z. \]

This problem can be efficiently solved by gridding the constraint set \( \mathcal{Z}_r \) and solving the resulting scalar linear program (LP) for each point \( r \). In the special case that \( P_f \), \( K_f \) are constant this reduces to one small scale LP.

2) Local Stability - \( \alpha_1 \): Determining a non-conservative constant \( \alpha_1 \), related to the local Lyapunov function \( V_f \) can be significantly more difficult. For comparison, in the setpoint stabilization case a non-convex optimization problem is formulated to check whether (4) holds for a specific value of \( \alpha \), compare [9, Rk. 3.1]. In a similar fashion, we consider the following algorithm\(^6\) to determine whether (4) holds for all \( \alpha \leq \alpha_1 \):

**Algorithm 1** Offline computation - Local stability \( \alpha_1 \)

Given a candidate constant \( \alpha_1 \):
1. for \( (r, r^+) \) satisfying (20) do
2. Compute \( P_f(r), P_f(r^+), K_f(r) \).
3. Generate random vectors \( \Delta x_i \); with \( \| \Delta x_i \|_{P_f(r)}^2 \leq \alpha_1 \).
4. Check if \( x_i = x_r + \Delta x_i \) satisfies (4).
5. end for

The value \( \alpha_1 \) is iteratively decreased until all considered combination \( (r, r^+, x_r) \) satisfy (4).

The overall offline procedure is summarized as follows:

\[\text{Algorithm 2 Offline computation} \]

1. Define \( \theta \) corresponding to the linearization (19).
2. LMI computation using gridding or convexification:
   - **Gridding**: Select \( (r_1, r^+_1) \) satisfying (20).
     Solve (18) for all \( (r_1, r^+_1) \).
   - **Convex**: Determine set \( \Theta \) that satisfies (22), solve (21).
3. Compute size of the terminal set \( \alpha = \min \{ \alpha_1, \alpha_2 \} \): a) compute \( \alpha_1 \) using Algorithm 1 (or (16)), b) compute \( \alpha_2 \) using (24) (or (17)).

The presented offline procedure is considerably more involved than for example the computation for one specific setpoint [9]. We emphasize that this procedure only has to be completed once and we need no repeated offline computations to account for changing operation conditions.

**E. Setpoint tracking**

Now we discuss setpoint tracking, which is included in the previous derivation as a special case with \( \mathcal{Z}_r \) such that \( \mathcal{Z}_r \) implies \( x_r = f(x_r, u_r) \) and \( R(r) = \rho \). Note, that both presented approaches significantly simplify in this case. For the gridding approach it suffices to grid along the steady-state manifold \( \mathcal{Z}_r \) which is typically low dimensional.

In the convex approach (Prop. 3) we have \( \theta^+ = \theta \) and thus we only consider the \( 2^p \) vertices of \( \Theta \).

Compared to the dynamic reference tracking problem, the problem of tracking a setpoint has received a lot of attention in the literature and many solutions have been suggested.

One of the first attempts to solve this issue is the usage of a pseudo linearization in [10]. There a nonlinear state and input transformation is sought, such that the linearization of the transformed around the setpoints is constant and thus constant terminal ingredients can be used. This approach seems unpractical, since there is no easy or simple method to compute the pseudo linearization.

In [14], [13], [12] the steady-state manifold \( \mathcal{Z}_r \) is partitioned into sets. In each set the nonlinear system is described as an LTV system and a constant terminal cost and controller are computed. Correspondingly, in closed-loop operation under changing setpoints [12] the terminal cost matrix \( P_f \) is piecewise constant. This might cause numerical problems in the optimization, since the cost is not differentiable with respect to the reference \( r \). Furthermore, the partitioning of the steady-state manifold seems difficult for general MIMO systems (if the dimension of the steady-state manifold is larger than one).

In comparison, Algorithm 2 yields continuously parameterized
cannot be directly translated into a simple optimization problem. However, this method cannot be directly translated into a simple optimization problem and might hence not be tractable.

In [6, Remark 8] it was proposed to compute a continuously parameterized controller \( K_f(\tau) \) by analytically using a pole-placement formula and solving the corresponding Lyapunov equation to obtain \( P_f(\tau) \). The resulting terminal ingredients are quite similar to the proposed one. However, this method cannot be directly translated into a simple optimization problem.

The main novelty in this approach, is that the offline computation only needs to be done once, irrespective of the to be stabilized setpoint or trajectory. This is possible by computing parameterized terminal ingredients and approximating the nonlinear system locally as a quasi-LPV system, with the to be stabilized reference trajectory as the parameters.

We believe that these generic offline computations are key to implementing nonlinear tracking schemes that allow for changing operation conditions.

**REFERENCES**


In [6], the terminal cost \( V_f(\tau) \) is computed for a general (differentiable) stage cost function that is not necessarily quadratic, compare also [5]. The computation of the terminal cost is decomposed into a linear and quadratic term, as in [36]. Computing the quadratic term of this economic terminal cost is equivalent to computing a quadratic terminal cost for a quadratic stage cost (Ass 2).