A stability governor for constrained linear-quadratic MPC without terminal constraints [⋆]

Jordan Leung [a], Frank Permenter [b], Ilya V. Kolmanovsky [a],

[a] *University of Michigan, Ann Arbor, MI 48109 USA*

[b] *Toyota Research Institute, Cambridge, MA 02139 USA*

Abstract

This paper introduces a supervisory unit, called the stability governor (SG), that provides improved guarantees of stability for constrained linear systems under Model Predictive Control (MPC) without terminal constraints. At each time step, the SG alters the setpoint command supplied to the MPC problem so that the current state is guaranteed to be inside of the region of attraction for an auxiliary equilibrium point. The proposed strategy is shown to be recursively feasible and asymptotically stabilizing for all initial states sufficiently close to any equilibrium of the system. Thus, asymptotic stability of the target equilibrium can be guaranteed for a large set of initial states even when a short prediction horizon is used. A numerical example demonstrates that the stability governed MPC strategy can recover closed-loop stability in a scenario where a standard MPC implementation without terminal constraints leads to divergent trajectories.

Key words: Control of constrained systems, model predictive control, asymptotic stabilization, tracking.

1 Introduction

Model Predictive Control (MPC) is a feedback strategy defined by the solution of a receding horizon optimal control problem (OCP). MPC is an attractive option for many control tasks due to its ability to provide high-performance control in the presence of system constraints. Moreover, theoretical properties (e.g., stability and robustness) of MPC have been extensively studied [13,16] and there is a wealth of practical evidence of the effectiveness of MPC.

Asymptotic stability of a system under MPC feedback is often guaranteed through the choice of the terminal penalty and the terminal set constraint of the receding horizon OCP [13]. However, there are compelling reasons to avoid the use of a terminal constraint when implementing MPC. For example, OCPs with only control constraints are much easier to solve and it is possible to establish continuity of the value function for a range of OCPs with no state constraints [16, Chapter 2.6]. The total number of constraints in the OCP may also be significantly increased by the inclusion of a stabilizing terminal constraint. Moreover, offline computation of a stabilizing terminal set may be intractable for high dimensional systems (e.g., [15, Example 2]).

Additionally, the performance of many algorithms used to solve linear-quadratic OCPs can significantly improve when there are no terminal constraints and the pointwise-in-time state and control constraints are of a particular form. For example, if projections onto the state and input constraint sets are easy to compute (e.g., box constraints), then expensive sub-problems reduce to inexpensive projection operations for a variety of first-order optimization methods. Examples of such methods are the dual projected gradient method (e.g., [14]), the alternating directional method of multipliers (e.g., [1]), the Chambolle & Pock method [6], and the proportional-integral projected gradient method [20]. In addition, second-order methods (e.g., [3, 12, 19]) can take advantage of the problem structure that arises when all

[⋆] Toyota Research Institute (TRI) provided funds to assist the authors with their research but this article solely reflects the opinions and conclusions of its authors and not TRI or any other Toyota entity. The third author acknowledges support by the National Science Foundation award number CMMI-1904394 and the Air Force Office of Scientific Research Grant number FA9550-20-1-0385. This paper was not presented at any IFAC meeting. Corresponding author Jordan Leung.

Email addresses: `jmleung@umich.edu` (Jordan Leung), `frank.permenter@tri.global` (Frank Permenter), `ilya@umich.edu` (Ilya V. Kolmanovsky).

constraint sets are boxes.

Omitting the terminal constraint in MPC clearly provides certain theoretical and practical benefits. However, doing this also has the undesirable consequence of weakening the associated theoretical guarantees of closed-loop stability and recursive feasibility. Limón et al. [10] showed that, under reasonable assumptions, a system under MPC without terminal constraints has a region of attraction (ROA) characterized by a sublevel set of the optimal cost. However, the size of this ROA is often much smaller than the ROA resulting from an MPC law with terminal constraints.

This paper develops a stability governor (SG) for MPC without terminal constraints that expands the ROA of [10] by altering the reference command provided to the MPC problem. At each time step, the SG constructs an auxiliary reference command that ensures the current state is inside of the ROA for the corresponding equilibrium. Under this paradigm, asymptotic stability of the target equilibrium can be guaranteed for all initial states contained in the union of ROAs corresponding to each equilibria of the system. As a result, closed-loop stability and recursive feasibility can be guaranteed using shorter prediction horizons and a smaller terminal penalty parameter than the existing method in [10].

The SG is similar in philosophy to the feasibility governors (FGs) in [17, 18], which expand the feasible region of MPC *with* terminal constraints by altering the reference command so that the predicted final state is maintained in the terminal region of the modified reference. In contrast, the SG maintains the current state in a sequence of ROA estimates for MPC *without* terminal constraints. Thus, the challenges associated with enforcing terminal constraints are avoided and efficient solvers for OCPs without terminal constraints can be used.

The recent work in [15] also warrants discussion in the context of our work. Asymptotic stability and recursive feasibility of the approach in [15] is enforced through a set of implicit terminal conditions. These terminal conditions are explicitly enforced in the OCP, but are implicit in the sense that they constitute sufficient conditions for satisfaction of a terminal constraint based on the maximum output admissible set (MOAS) [4] of a terminal control law. The work in [15] is motivated by scenarios where computation of the MOAS is intractable (e.g., high dimensional systems). In contrast, our work is primarily motivated by the computational simplicity of MPC without terminal constraints and we impose no additional constraints on the predicted terminal state.

The paper is organized as follows. Section 2 describes the problem setting. Section 3 establishes properties of MPC without terminal constraints. Section 4 derives properties of the OCP that are used to develop the SG in Section 5. Section 6 describes modifications that can be made to the SG. A numerical example is reported in Section 7. Finally, Section 8 contains concluding remarks.

Notation: Let $\mathbb{N}$ represent the natural numbers including 0. Given $a, b \geq 0$, let $\mathbb{N}_{[a,b]} = \mathbb{N} \cap [a, b]$. Given $x \in \mathbb{R}^n$ and $W \in \mathbb{R}^{n\times n}$ with $W \succ 0$, the W-norm of x is $\|x\|_W = \sqrt{x^TWx}$. Let $\|\cdot\|$ represent the 2-norm when no subscript is specified. Given $\rho \in \mathbb{R}$ and $X \subset \mathbb{R}^n$, let $\rho X = \{\rho x \mid x \in X\}$. Let $(x, y) = [x^T\ y^T]^T$. The normal cone mapping for a convex set $\mathcal{Z}$ is defined as

$$\mathcal{N}_{\mathcal{Z}}(z) = \begin{cases} \{y \mid y^T(w - z) \leq 0, \forall w \in \mathcal{Z}\}, & \text{if } z \in \mathcal{Z}, \\ \emptyset & \text{otherwise.} \end{cases}$$

The classes of $\mathcal{K}$ and $\mathcal{KL}$ functions follow the usual definitions (e.g., see [5]). For a sequence $\{v_k\}_{k=0}^{\infty}$, the notation $v_k \equiv v$ is used as shorthand for $v_k = v$ for all $k \in \mathbb{N}$.

2 Problem Setting

We consider a class of linear time-invariant systems given by

$$x_{k+1} = Ax_k + Bu_k, \tag{1a}$$
$$z_k = Cx_k + Du_k, \tag{1b}$$

where $x_k \in \mathbb{R}^n$ is the state, $u_k \in \mathbb{R}^{n_u}$ is the control, and $z_k \in \mathbb{R}^{n_z}$ is the tracking output. The control objective is to drive the tracking output z_k to a desired setpoint $r \in \mathbb{R}^{n_z}$ subject to pointwise-in-time constraints

$$x_k \in \mathcal{X}, u_k \in \mathcal{U},\ \forall k \in \mathbb{N}, \tag{2}$$

where $\mathcal{X} \subseteq \mathbb{R}^n$ and $\mathcal{U} \subseteq \mathbb{R}^{n_u}$ are constraint sets.

Assumption 1 *The pair (A, B) is stabilizable and the sets $\mathcal{X}$ and $\mathcal{U}$ are closed, convex, and contain the origin in their interior.*

Equilibria of (1) satisfy $Z[\bar{x}^T\ \bar{u}^T\ \bar{z}^T]^T = 0$, where

$$Z = \begin{bmatrix} A - I & B & 0 \\ C & D & -I \end{bmatrix}.$$

Moreover, these equilibria can be parameterized by a reference $v \in \mathbb{R}^{n_v}$ according to

$$\begin{bmatrix} \bar{x}_v \\ \bar{u}_v \\ \bar{z}_v \end{bmatrix} = \begin{bmatrix} G_x \\ G_u \\ G_z \end{bmatrix} v, \tag{3}$$

where $G^T = [G_x^T\ G_u^T\ G_z^T]^T$ is a basis for the nullspace of Z and Assumption 1 ensures that $\text{Null}(Z) \neq \{0\}$ [11]. The following assumption excludes ill-posed reference tracking problems.

Assumption 2 *The matrix G_z is full rank and $\exists v_r^* \in \mathcal{V}$ such that $G_z v_r^* = r$, where*

$$\mathcal{V} = \{v \in \mathbb{R}^{n_v} \mid G_x v \in \text{Int } \mathcal{X},\ G_u v \in \text{Int } \mathcal{U}\}.$$

is the set of strictly steady-state feasible references.

Remark 1 *If $n_v > n_z$, then there may exist multiple v satisfying $G_z v = r$. In this case, let v_r^* denote any particular choice (e.g., the minimum norm solution).*

Consider the following OCP, which defines a reference tracking MPC policy:

$$\begin{aligned}
\min_{(\hat{\text{x}},\hat{\text{u}})}\ & J_N(x, v, \hat{\text{u}}) = \lambda\omega(\hat{x}_N, v) + \sum_{i=0}^{N-1} \ell(\hat{x}_i, \hat{u}_i, v) && \text{(4a)}\\
\text{s.t.}\ & \hat{x}_0 = x, && \text{(4b)}\\
& \hat{x}_{i+1} = A\hat{x}_i + B\hat{u}_i, \qquad i \in \mathbb{N}_{[0,N-1]}, && \text{(4c)}\\
& \hat{x}_{i+1} \in \mathcal{X},\ \hat{u}_i \in \mathcal{U}, \qquad i \in \mathbb{N}_{[0,N-1]}, && \text{(4d)}
\end{aligned}$$

where $N \in \mathbb{N}$ is the prediction horizon, $\lambda \geq 1$ is a weighting parameter, and

$$\begin{aligned}
\ell(x, u, v) =& ||x - G_x v||_Q^2 + ||u - G_u v||_R^2,\\
\omega(x, v) =& ||x - G_x v||_P^2,
\end{aligned}$$

where Q, R, and P are weighting matrices. The predicted state and control sequences are denoted as $\hat{\text{x}} = (\hat{x}_0, \ldots, \hat{x}_N)$ and $\hat{\text{u}} = (\hat{u}_0, \ldots, \hat{u}_{N-1})$. The state x and reference v are parameters in this OCP. We use the notation $\mathcal{P}_N^\lambda(x, v)$ to refer to problem (4) specified with parameters (x, v) and fixed constants $N \in \mathbb{N}$ and $\lambda \geq 1$.

Assumption 3 *The cost matrices in (4) satisfy $Q = Q^T \succ 0$ and $R = R^T \succ 0$. The terminal cost matrix $P = P^T \succ 0$ is chosen as*

$$P = (A - BK)^T P(A - BK) + \eta(Q + K^T RK), \quad (5)$$

where $\eta \geq 1$ is a constant and K is a gain matrix chosen so that $(A - BK)$ is Schur stable.

Consider the closed-loop system under the following MPC feedback strategy:

$$\begin{aligned}
x_{k+1} &= Ax_k + Bu^*(x_k, v_k), && \text{(6a)}\\
u^*(x_k, v_k) &= \Xi\hat{\text{u}}^*(x_k, v_k), && \text{(6b)}
\end{aligned}$$

where $\{v_k\}_{k=0}^\infty$ is a sequence of reference commands, $\hat{\text{u}}^*(x, v)$ is the optimal solution to $\mathcal{P}_N^\lambda(x, v)$, and $\Xi = [I_{n_u}\ 0\ \ldots\ 0]$ is a matrix that selects the first control input. The SG developed in this paper chooses v_k at each time step so that the target equilibrium $G_x v_r^*$ is asymptotically stable with a region of attraction (ROA) [1] that is significantly larger than the ROA that results from selecting $v_k \equiv v_r^*$.

3 Properties of Terminal Unconstrained MPC

In this section, we establish nominal stability of the closed-loop system (6) for a static reference $v_k \equiv v$. To begin, we define the following set-valued map:

$$\Omega(v, \alpha) := \{x \in \mathbb{R}^n \mid \omega(x, v) \leq \alpha\}, \quad (7)$$

which is used to define an implicit terminal set.

Definition 2 *(Implicit terminal set) A set $\mathcal{S}$ is called an implicit terminal set for $v \in \mathcal{V}$ if: (1) $\mathcal{S} = \Omega(v, \alpha)$ for some $\alpha > 0$, (2) $\mathcal{S} \subset \mathcal{X}$, and (3) $\inf_{u\in\mathcal{U}}\{\omega(Ax+Bu, v) - \omega(x, v) + \ell(x, u, v)\} \leq 0$ for all $x \in \mathcal{S}$.*

The following proposition provides a sufficient condition for $\Omega(v, \alpha)$ to be an implicit terminal set. The result follows directly from Assumption 3.

Proposition 1 *Let the stated assumptions hold, $v \in \mathcal{V}$, and $\alpha > 0$. Then, $\Omega(v, \alpha)$ is an implicit terminal set if*

$$\Omega(v, \alpha) \subseteq \{x \in \mathcal{X} \mid [G_u v - K(x - G_x v)] \in \mathcal{U}\}, \quad (8)$$

where K is defined in Assumption 3.

The results in [10] can then be used to establishes a ROA for the closed-loop system in (6) for $v_k \equiv v$.

Theorem 3 *Let the stated assumptions hold and let $v \in \mathcal{V}$ and $\alpha > 0$ be such that $\Omega(v, \alpha)$ is an implicit terminal set. Consider the closed-loop system in (6) for a static reference sequence $v_k \equiv v$. Define $d = \alpha/(\|Q^{-\frac{1}{2}}P^{\frac{1}{2}}Q^{-\frac{1}{2}}\|^2)$ and let $N \geq 1$ and $\lambda \geq 1$. Then, $\bar{x}_v = G_x v$ is an asymptotically stable equilibrium with an ROA given by*

$$\begin{aligned}
\Gamma_N^\lambda(v) := \{x \in \mathbb{R}^n \mid V_N^\lambda(x, v) \leq\ & \ell(x, u^*(x, v), v)\\
& + (N - 1)d + \lambda\alpha\},
\end{aligned}$$

where $V_N^\lambda(x, v)$ is the optimal cost of $\mathcal{P}_N^\lambda(x, v)$. Moreover, if $x \in \Gamma_N^\lambda(v)$ and $u^* = u^*(x, v)$, then $\hat{x}_N^* \in \Omega(v, \alpha)$ and

$$V_N^\lambda(Ax + Bu^*, v) \leq V_N^\lambda(x, v) - \ell(x, u^*, v). \quad (9)$$

PROOF. The proof follows identically to that of [10, Theorem 1] since $d < \ell(x, u, v)$ for all $x \notin \Omega(\alpha, v)$ and $u \in \mathbb{R}^{n_u}$. Hence, Assumptions 1-2 of [10] are satisfied.

One can easily show that $\Gamma_N^\lambda(v) \supset \Omega(v, \alpha)$ and that

$$\Upsilon_N^\lambda(v) := \{x \in \mathbb{R}^n \mid V_N^\lambda(x, v) \leq Nd + \lambda\alpha\} \subset \Gamma_N^\lambda(v),$$

is also an ROA for the closed-loop system [10].

[1] Throughout the paper, the terminology "region of attraction" will be used to refer to any invariant subset of the largest region of attraction of (6).

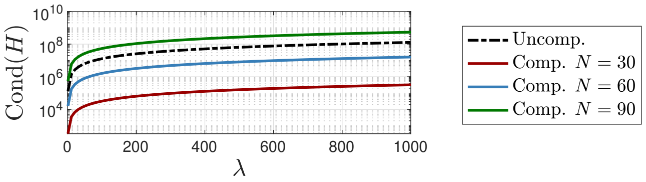


Fig. 1. The dependence of the condition number of the uncompressed Hessian $\tilde{H} = \text{blkdiag}(\hat{R}, \hat{H})$ and the compressed Hessian $H = \hat{R} + \hat{B}^T \hat{H} \hat{B}$ (see Appendix A) on the terminal weighting λ for the inverted pendulum example in Section 7.

Remark 4 *For all initial states x_0 in the N-step backward reachable set to Int $\Omega(v, \alpha)$, the terminal weighting $\lambda \geq 1$ can be selected to be sufficiently large so that $x_0 \in \Upsilon_N^\lambda(v)$ [10, Theorem 3].*

Remark 5 *The volume of $\Upsilon_N^\lambda(v)$ is also influenced by the choice of K and P in Assumption 3. Hence, it is possible to increase the volume of $\Upsilon_N^\lambda(v)$ by tuning K.*

In premise, the closed-loop system (6) can track any target auxiliary reference $v_r^* \in \mathcal{V}$ by fixing $v_k \equiv v_r^*$, specifying α such that $\Omega(v_r^*, \alpha)$ is an implicit terminal set, and selecting (N, λ) such that $x_0 \in \Upsilon_N^\lambda(v_r^*)$. However, if x_0 is far from $G_x v_r^*$, then $x_0 \in \Upsilon_N^\lambda(v_r^*)$ is only satisfied for large values of N and λ. Hence, computing $\hat{\mathrm{u}}^*(x_k, v_r^*)$ would require solving a high-dimensional and poorly conditioned quadratic program (see Figure 1). This may be computationally prohibitive — especially when using first-order optimization methods.

To address these issues, the SG developed in Section 5 selects the reference v_k at each time step so that the closed-loop system (6) can be asymptotically stabilized around $G_x v_r^*$ without requiring that $x_0 \in \Upsilon_N^\lambda(v_r^*)$.

4 Properties of the Value Function

This section derives properties of the value function that are used to develop the stability governor. To begin, note that $\mathcal{P}_N^\lambda(x, v)$ can be written in condensed form as

$$\begin{aligned} \min_{\hat{\mathrm{u}}} \quad & J_N(x, v, \hat{\mathrm{u}}) = \hat{\mathrm{u}}^T H \hat{\mathrm{u}} + 2\hat{\mathrm{u}}^T F \theta + \theta^T W \theta \\ \text{s.t.} \quad & \hat{\mathrm{u}} \in \mathcal{Z}(x) = \{\hat{\mathrm{u}} \in \mathcal{U}^N \mid (\hat{A}x + \hat{B}\hat{\mathrm{u}}) \in \mathcal{X}^N\}. \end{aligned} \tag{10}$$

where $\theta = (x, v)$ and the matrices in (10) are defined in Appendix A. The set of feasible states $\mathcal{F} \subseteq \mathcal{X}$ and the value function $V_N^\lambda : \mathcal{F} \times \mathcal{V} \to [0, \infty)$ of (10) are defined as

$$\mathcal{F} = \{x \mid \mathcal{Z}(x) \neq \emptyset\}, \quad V_N^\lambda(x, v) = \min_{\hat{\mathrm{u}} \in \mathcal{Z}(x)} J_N(x, v, \hat{\mathrm{u}}).$$

The square-root of the value function, $\psi(x, v) := \sqrt{V_N^\lambda(x, v)}$, is vital for the analysis and development of the stability governor. In particular, we will demonstrate that ψ is Lipschitz continuous with respect to v for all $x \in \mathcal{F}$, i.e.

$$|\psi(x, v') - \psi(x, v)| \leq \|v' - v\|_{W_v}, \forall x \in \mathcal{F}, \ \forall v', v \in \mathcal{V},$$

where $W_v \succ 0$ is defined in Appendix A. Note that we are only interested in perturbations of ψ with respect to perturbations in v for a fixed x. Moreover, the constraint set mapping $\mathcal{Z}$ depends only on x and not v. Hence, the desired bound can be derived by analyzing the solutions to (10) for a fixed $x \in \mathcal{F}$.

Note that $\mathcal{Z}(x)$ is convex and non-empty for all $x \in \mathcal{F}$ by Assumptions 1 and 3. Thus, the following variational inequality is necessary and sufficient for optimality of (10) [2, Chapter 2.7]:

$$H\hat{\mathrm{u}} + F_v v + F_x x + \mathcal{N}_{\mathcal{Z}(x)}(\hat{\mathrm{u}}) \ni 0.$$

Hence, the solution mapping of this variational inequality at a fixed $x \in \mathcal{F}$ can be written as

$$S_x(v) = \mathcal{A}_x^{-1}(-F_v v - F_x x), \quad \mathcal{A}_x := H + \mathcal{N}_{\mathcal{Z}(x)}.$$

Note that S_x is a function for all $x \in \mathcal{F}$ by strong convexity of (10) (Assumptions 1 and 3) and that $V_N^\lambda(x, v) = J_N(x, v, S_x(v))$. Moreover, the operator $\mathcal{A}_x : \mathbb{R}^{Nn_u} \rightrightarrows \mathbb{R}^{Nn_u}$ has the following properties.

Proposition 2 *Fix $x \in \mathcal{F}$ and let $y, y' \in \mathcal{Z}(x)$, $u \in \mathcal{A}_x(y), u' \in \mathcal{A}_x(y')$. The map $\mathcal{A}_x$ satisfies:*

(1) $\langle u' - u, y' - y \rangle \geq \|y' - y\|_H^2$;
(2) $\langle \mathcal{A}_x^{-1} u' - \mathcal{A}_x^{-1} u, u' - u \rangle \geq ||\mathcal{A}_x^{-1} u' - \mathcal{A}_x^{-1} u||_H^2$;
(3) $\|\mathcal{A}_x^{-1} u' - \mathcal{A}_x^{-1} u\|_H \leq ||u' - u||_{H^{-1}}$.

The proof of Proposition 2 is omitted for brevity as it follows analogously to that of [8, Proposition 1]. The following corollary arises from the second statement in Proposition 2 by selecting $y = S_x(v)$ and $y' = S_x(v')$.

Corollary 6 *Fix $x \in \mathcal{F}$ and let $v', v \in \mathcal{V}$, then*

$$\langle S_x(v') - S_x(v), F_v(v' - v) \rangle \leq -\|S_x(v') - S_x(v)\|_H^2,$$

and $\|S_x(v') - S_x(v)\|_H \leq \|F_v(v' - v)\|_{H^{-1}}$, where $F_v \in \mathbb{R}^{Nn_u \times n}$ is defined in Appendix A.

Corollary 6 can then be used to derive the desired Lipschitz condition on ψ.

Lemma 7 *Fix $x \in \mathcal{F}$ and let $v', v \in \mathcal{V}$, then*

$$|\psi(x, v') - \psi(x, v)| \leq \|v' - v\|_{W_v},$$

where $W_v \succ 0$ is defined in Appendix A.

PROOF. Note that $J(x,v,\hat{\mathrm{u}}) = \|(x,v,\hat{\mathrm{u}})\|_\Phi^2$, where

$$\Phi = \begin{bmatrix} W & F^T \\ F & H \end{bmatrix},\ F = [F_x \ \ F_v],\ W = \begin{bmatrix} W_x & W_{xv} \\ W_{xv}^T & W_v \end{bmatrix}.$$

Moreover, $V_N^\lambda(x,v) = \|(x,v,S_x(v))\|_\Phi^2$ and $\psi(x,v) = \|(x,v,S_x(v))\|_\Phi$. Thus,

$$\begin{aligned} \|\psi(x,v') - &\psi(x,v)\|^2 \\ &= |\|(x,v',S_x(v'))\|_\Phi - \|(x,v,S_x(v))\|_\Phi|^2 \\ &\le \|(0, v'-v, S_x(v') - S_x(v))\|_\Phi^2, \end{aligned}$$

where the last line follows from the reverse triangle inequality. By definition of Φ,

$$\begin{aligned} \|\psi(x,v') - \psi(x,v)\|^2 &\le \|S_x(v') - S_x(v)\|_H^2 \\ + 2\langle S_x(v') - S_x(v), &F_v(v'-v)\rangle + \|v'-v\|_{W_v}^2, \end{aligned}$$

and Corollary 6 can be used to write

$$|\psi(x,v') - \psi(x,v)|^2 \le \|v'-v\|_{W_v}^2 - \|S_x(v') - S_x(v)\|_H^2,$$

where the desired result follows by dropping the negative term and taking the square-root.

5 Stability Governor

In this section, we describe the reference selection policy that defines the SG. To begin, the following lemma establishes a bound for $V_N^\lambda(x_{k+1}, v_k)$.

Lemma 8 *Consider the closed-loop system in* (6)*. Let $v_k \in \mathcal{V}$ and $\alpha > 0$ be such that $\Omega(v_k,\alpha)$ is an implicit terminal set, define $d = \alpha/(\|Q^{-\frac{1}{2}}P^{\frac{1}{2}}Q^{-\frac{1}{2}}\|^2)$, and let $N \ge 1$, $\lambda \ge 1$, and $x_k \in \Gamma_N^\lambda(v_k)$. Define $\hat{\mathrm{u}}^*(x_k,v_k) =: (\hat{u}_{k|0}^*, ..., \hat{u}_{k|N-1}^*)$ and*

$$\tilde{\mathrm{u}}_{k+1} := (\hat{u}_{k|1}^*, ..., \hat{u}_{k|N-1}^*, G_u v_k - K(\hat{x}_{N|k}^* - G_x v_k)).$$

Then, $V_N^\lambda(x_{k+1}, v_k) \le J_N(x_{k+1}, v_k, \tilde{\mathrm{u}}_{k+1}) \le V_N^\lambda(x_k,v_k) - \ell(x_k,u_k,v_k)$. Moreover, $\tilde{\mathrm{u}}_{k+1} \in \mathcal{Z}(x_{k+1})$.

PROOF. The result follows from the fact that $\hat{x}_{N|k}^* \in \Omega(v_k,\alpha)$ [10] and by using standard arguments in MPC stability analysis (e.g., see [16, Chapter 2.4]).

The following lemma forms the basis of the SG.

Lemma 9 *Let v_k, α, d, N, and λ be defined as in Lemma 8. If $x_k \in \Gamma_N^\lambda(v_k)$, x_{k+1} is given by* (6)*, and*

$$\begin{aligned} \|v_{k+1} - v_k\|_{W_v} \le \sqrt{Nd + \lambda\alpha}& \\ - \sqrt{J_N(x_{k+1}, v_k, \tilde{\mathrm{u}}_{k+1})},& \quad (11) \end{aligned}$$

then $V_N^\lambda(x_{k+1}, v_{k+1}) \le Nd + \lambda\alpha$.

PROOF. By Lemma 7:

$$\psi(x_{k+1}, v_{k+1}) \le \sqrt{V_N^\lambda(x_{k+1}, v_k)} + \|v_{k+1} - v_k\|_{W_v}.$$

Hence, $V_N^\lambda(x_{k+1}, v_{k+1}) \le Nd + \lambda\alpha$ is implied by the upper-bound of $\|v_{k+1} - v_k\|_{W_v}$ in (11) and the upper-bound of $V_N^\lambda(x_{k+1}, v_k)$ in Lemma 8.

The following parameterization of the reference command is adopted to reduce the computational cost of implementing the SG:

$$v_{k+1} = v_k + \kappa_{k+1}(v_r^* - v_k), \tag{12}$$

where $v_0 \in \mathcal{V}$ is a given initialization, $v_r^* \in \mathcal{V}$ is the target auxiliary reference, and $\kappa_k \in [0,1]$ is a time-varying parameter that dictates the rate at which v_k converges to v_r^*. Note that $\kappa_k = 1$ implies $v_k = v_r^*$ and that $\kappa_k = 0$ implies $v_k = v_{k-1}$.

Lemma 9 states that the reference command can be updated so that $x_{k+1} \in \Upsilon_N^\lambda(v_{k+1})$ if $x_k \in \Gamma_N^\lambda(v_k)$ and $\Omega(v_k,\alpha)$ is an implicit terminal set. However, $\Omega(v_{k+1},\alpha)$ must also be an implicit terminal set to apply this result recursively. The following proposition, the proof of which is in Appendix B, provides a constructive method for computing $\alpha > 0$ such that $\Omega(v_k,\alpha)$ is an implicit terminal set for all v_k parameterized by (12).

Proposition 3 *Let $v_r^* \in \mathcal{V}$, $v_0 \in \mathcal{V}$, and $\alpha > 0$ satisfy*

$$\begin{aligned} \Omega(v_0,\alpha) &\subset \{x \in \mathcal{X} \mid [G_u v_0 - K(x - G_x v_0)] \in \mathcal{U}\}, \\ \Omega(v_r^*,\alpha) &\subset \{x \in \mathcal{X} \mid [G_u v_r^* - K(x - G_x v_r^*)] \in \mathcal{U}\}, \end{aligned}$$

where Ω is defined in (7)*. Then, $\Omega(v,\alpha)$ is an implicit terminal set for all $v \in \{v_0 + \kappa(v_r^* - v_0) \mid \kappa \in [0,1]\} \subset \mathcal{V}$.*

The SG is defined by the following reference selection policy, which enforces (11) by design:

$$\begin{aligned} &\kappa_{k+1} = \min\{\tilde{\kappa}_{k+1}, 1\}, \qquad (13) \\ &\tilde{\kappa}_{k+1} = \begin{cases} \frac{\sqrt{Nd+\lambda\alpha} - \sqrt{J_N(x_{k+1}, v_k, \tilde{\mathrm{u}}_{k+1})}}{\|v_r^* - v_k\|_{W_v}}, & \text{if } v_k \ne v_r^*, \\ \infty & \text{if } v_k = v_r^*. \end{cases} \end{aligned}$$

So, consider the system

$$\begin{aligned} x_{k+1} &= Ax_k + Bu^*(x_k, v_k), &(14a) \\ v_{k+1} &= v_k + \kappa_{k+1}(v_r^* - v_k), &(14b) \end{aligned}$$

where κ_{k+1} is defined by (13). The following theorem, which is the primary result of this paper, details the closed-loop properties of (14).

Theorem 10 *Let $v_r^* \in \mathcal{V}$, $v_0 \in \mathcal{V}$, and $\alpha > 0$ satisfy the conditions in Proposition 3. Define $d =$*

$\alpha/(\|Q^{-\frac{1}{2}}P^{\frac{1}{2}}Q^{-\frac{1}{2}})\|^2)$, $N \geq 1$, and $\lambda \geq 1$. If $x_0 \in \Gamma_N^\lambda(v_0)$, then the closed-loop system in (14) has the following properties:

- *Recursive feasibility, i.e., the optimization problem $\mathcal{P}_N^\lambda(x_k, v_k)$ is feasible and $x_k \in \Gamma_N^\lambda(v_k)$ for all $k \in \mathbb{N}$.*
- *Asymptotic stability, i.e., $\lim_{k\to\infty}(x_k, v_k) = (G_x v_r^*, v_r^*)$ and $(G_x v_r^*, v_r^*)$ is a Lyapunov stable equilibrium.*
- *Finite-time convergence of the reference, i.e., $\exists k' \in \mathbb{N}$ such that $v_k = v_r^*$ for all $k \geq k'$.*

PROOF. To begin, we prove recursive feasibility. Note that $V_N^\lambda(x_k, v_k) < \infty$ directly implies feasibility of $\mathcal{P}_N^\lambda(x_k, v_k)$. So, we prove the recursive feasibility property by proving that $x_k \in \Gamma_N^\lambda(v_k)$ for all $k \in \mathbb{N}$. Note that $\Omega(v_{k+1}, \alpha)$ is an implicit terminal set for all $k \geq 0$ by Proposition 3 and that $x_0 \in \Gamma_N^\lambda(v_0)$ by assumption. Fix $k \in \mathbb{N}$ and assume the non-trivial case of $v_k \neq v_r^*$. If $\kappa_{k+1} = 0$, then $v_{k+1} = v_k$ and $x_{k+1} \in \Upsilon_N^\lambda(v_{k+1})$ by (9). If $\kappa_{k+1} > 0$, then $x_{k+1} \in \Upsilon_N^\lambda(v_{k+1})$ by Lemma 9. If instead $v_k = v_r^*$, then $v_{k+1} = v_k = v_r^*$ and $x_{k+1} \in \Upsilon_N^\lambda(v_{k+1})$ by (9). Thus, $x_k \in \Upsilon_N^\lambda(v_k) \subset \Gamma_N^\lambda(v_k)$ for all $k \geq 1$ by induction.

Next, we prove that $\forall k \in \mathbb{N}$, $\kappa_k \geq \bar{\kappa}$ for a constant $\bar{\kappa} > 0$. Recall that $J_N(x_{k+1}, v_k, \tilde{\mathrm{u}}_{k+1}) \leq V_N^\lambda(x_k, v_k) - \ell(x_k, u_k, v_k)$ by Lemma 8. Hence, $J_N(x_{k+1}, v_k, \tilde{\mathrm{u}}_{k+1}) \leq (N-1)d + \lambda\alpha$ since $x_k \in \Gamma_N^\lambda(v_k)$, and so

$$\tilde{\kappa}_{k+1} \geq \bar{\kappa} := \frac{\sqrt{Nd + \lambda\alpha} - \sqrt{(N-1)d + \lambda\alpha}}{\|v_r^* - v_0\|_{W_v}},$$

where $\tilde{\kappa}_{k+1}$ is defined in (13) and we have used the fact that $\|v_r^* - v_k\|_{W_v}$ is monotonically decreasing by definition of v_k. Thus, asymptotic convergence of v_k to v_r^* follows from $\|v_r^* - v_k\| \leq (1 - \bar{\kappa})\|v_r^* - v_{k-1}\|$.

To prove convergence of x_k, we use [17, Lemma 5] to state that there exists $\rho \in \mathcal{KL}$ and $\zeta \in \mathcal{K}$ such that

$$\begin{aligned} \|x_k - G_x v_k\|_Q \leq\ & \rho(\|x_0 - G_x v_0\|, k) \\ & + \zeta\left(\sup_{j\geq 0}\|\Delta v_j\|\right), \end{aligned} \tag{15}$$

and $\lim\sup_k\|x_k - G_x v_k\|_Q \leq \zeta\,(\lim\sup_k\|\Delta v_k\|)$. Thus, $\lim_{k\to\infty}\ \|x_k - G_x v_k\| = 0$ since v_k converges, and so $\lim_{k\to\infty} x_k = G_x v_r^*$.

Next, we show that $(G_x v_r^*, v_r^*)$ is Lyapunov stable. Let $\delta > 0$ and $\|(x_0, v_0) - (G_x v_r^*, v_r^*)\| \leq \delta$. Then $\|v_k - v_r^*\| \leq \|v_0 - v_r^*\| \leq \delta$ for all $k \in \mathbb{N}$ by definition of v_k. Note that

$$\begin{aligned} \|x_0 - G_x v_0\| &\leq \|x_0 - G_x v_r^*\| + \|G_x v_r^* - G_x v_0\| \\ &\leq \delta + \|G_x\|\|v_r^* - v_0\| \leq (1 + \|G_x\|)\delta, \end{aligned}$$

and $\|\Delta v_k\| = \|v_k - v_{k-1}\| = \kappa_k\|v_r^* - v_{k-1}\| \leq \delta$ for all $k \in \mathbb{N}$. Thus, by the triangle inequality:

$$\begin{aligned} \|x_k - G_x v_r^*\|_Q &\leq \|x_k - G_x v_k\|_Q + \|G_x(v_k - v_r^*)\|_Q \\ &\leq \rho(p_1\delta, k) + \zeta(\delta) + p_2\delta, \end{aligned}$$

for some constants $p_1, p_2 > 0$, where the second inequality follows by (15). Thus, it is possible to bound $\|x_k - G_x v_r^*\|$ and $\|v_k - v_r^*\|$ arbitrarily small given δ sufficiently small; hence $(G_x v_r^*, v_r^*)$ is Lyapunov stable.

Last, we demonstrate finite-time convergence of the reference. Note that $\exists \alpha_2 \in \mathcal{K}$ such that $V_N^\lambda(x_k, v_k) \leq \alpha_2(\|x_k - G_v v_k\|)$. Thus, $\lim_{k\to\infty} V_N^\lambda(x_k, v_k) = 0$. So, $\forall c > 0$, $\exists k' \in \mathbb{N}$ such that $V_N^\lambda(x_k, v_k) < c$ and $\|v_r^* - v_k\|_{W_v} < c$ for all $k > k'$. Consequently,

$$\tilde{\kappa}_{k+1} \geq \frac{\sqrt{Nd + \lambda\alpha} - \sqrt{c}}{c},\ \forall k > k'.$$

Thus, one can define $c > 0$ sufficiently small and define $k' \in \mathbb{N}$ accordingly such that $\tilde{\kappa}_{k+1} > 1$ for all $k \geq k'$. Therefore, $\kappa_{k+1} = 1$ and $v_k = v_r^*$ for all $k \geq k'$.

Remark 11 *The requirement $x_0 \in \Upsilon_N^\lambda(v_0)$ is much less restrictive than the condition $x_0 \in \Upsilon_N^\lambda(v_r^*)$ since $v_0 \in \mathcal{V}$ is a free parameter. In other words, the set of initial states that can be stabilized is*

$$\mathcal{R}_N^\lambda = \{x \in \mathcal{X} \mid \exists v \in \mathcal{V} \text{ s.t. } x \in \Gamma_N^\lambda(v)\} = \bigcup_{v\in\mathcal{V}} \Gamma_N^\lambda(v).$$

The SG-based MPC procedure is summarized as follows.

Offline:

(1) Choose $Q \succ 0$ and $R \succ 0$.
(2) Choose K such that $(A - BK)$ is Schur stable and compute P using (5) for some $\eta \geq 1$.
(3) Choose v_0 so that $G_x v_0$ is close to x_0 (e.g., $v_0 = G_x^\dagger x_0$) and define v_r^* such that $r = G_z v_r^*$.
(4) Compute the parameter $\alpha > 0$ using Proposition 3 and define $d = \alpha/(\|Q^{-\frac{1}{2}}P^{\frac{1}{2}}Q^{-\frac{1}{2}})\|^2)$.
(5) *Optional*: Repeat steps 2-4 by tuning K so that the volumes of $\Omega(\alpha, v_0)$ and $\Omega(\alpha, v_r^*)$ are maximized.
(6) Define N and λ such that $x_0 \in \Gamma_N^\lambda(v_0)$.

Online (at time step $k+1$):

(1) Compute κ_{k+1} using (13).
(2) Update the reference v_{k+1} according to (12).
(3) Solve $\mathcal{P}_N^\lambda(x_{k+1}, u_{k+1})$ and let $u_{k+1} = u^*(x_{k+1}, v_{k+1})$.

If the target setpoint r is changed during online operation, then $\alpha > 0$ must be recomputed using Proposition 3. Alternatively, this can be avoided by using a value of $\alpha > 0$ that ensures $\Omega(v, \alpha)$ is an implicit terminal set for all v in a subset of $\mathcal{V}$. The following proposition, the proof of which is in Appendix B, provides a method for computing such an α.

Proposition 4 *Define*

$$\mathcal{V}_\epsilon = \{v \in \mathbb{R}^{n_v} \mid G_x v \in (1-\epsilon_x)\mathcal{X},\ G_u v \in (1-\epsilon_u)\mathcal{U}\},$$

where $\epsilon_x \in (0,1)$ and $\epsilon_u \in (0,1)$ are user-defined parameters. Let $\alpha > 0$ satisfy

$$\{x \in \mathbb{R}^n \mid \|x\|_P^2 \leq \alpha\} \subset \epsilon_x \mathcal{X}. \qquad (16)$$

In addition, if $G_u = 0$, let α satisfy

$$\{x \in \mathbb{R}^n \mid \|x\|_P^2 \leq \alpha\} \subset \{x \in \mathbb{R}^n \mid -Kx \in \mathcal{U}\}, \qquad (17)$$

whereas if $G_u \neq 0$, let α satisfy

$$\{x \in \mathbb{R}^n \mid \|x\|_P^2 \leq \alpha\} \subset \{x \in \mathbb{R}^n \mid -Kx \in \epsilon_u \mathcal{U}\}. \qquad (18)$$

Then, $\Omega(v, \alpha)$ is an implicit terminal set for all $v \in \mathcal{V}_\epsilon$.

Note that the value of α determined using Proposition 4 will generally be smaller than that of Proposition 3.

6 Modifications to the SG

6.1 Performance-based Modifications

In this subsection, we discuss modifications that can be made to the SG to improve closed-loop performance at the expense of increasing the computational cost.

(1) Improving the bound on $V_N^\lambda(x_{k+1}, v_k)$: The SG relies on the shifted-and-padded cost $J_N(x_{k+1}, v_k, \tilde{\mathrm{u}}_{k+1})$ to upper bound $V_N^\lambda(x_{k+1}, v_k)$. Hence, it is possible to improve the performance of the SG by using an improved upper-bound $\hat{V}_{k+1}$ satisfying $V_N^\lambda(x_{k+1}, v_k) \leq \hat{V}_{k+1} \leq J_N(x_{k+1}, v_k, \tilde{\mathrm{u}}_{k+1})$. That is, one could instead compute

$$\tilde{\kappa}_{k+1} = \frac{\sqrt{Nd + \lambda\alpha} - \hat{V}_{k+1}^{\frac{1}{2}}}{\|v_r^* - v_k\|_{W_v}}, \text{ if } v_k \neq v_r^*,$$

in (13). To obtain $\hat{V}_{k+1}$, one could (for example) suboptimally solve $\mathcal{P}_N^\lambda(x_{k+1}, v_k)$ using $\tilde{\mathrm{u}}_{k+1}$ as a warm-start.

(2) Online recomputation of α: The SG uses Proposition 3 to compute a constant $\alpha > 0$ that ensures $\Omega(v_k, \alpha)$ is an implicit terminal set for all $k \in \mathbb{N}$. However, one could instead recompute $\alpha > 0$ during operation. For example, at each time step k, one could use Proposition 3 to compute a value α_k that ensures $\Omega(v, \alpha_k)$ is an implicit terminal set for all $v \in \{v_k + \kappa(v_k' - v_k) \mid \kappa \in [0,1]\}$, where $v_k' = v_k + \kappa'(v_r^* - v_k)$ for some $\kappa' \in (0,1]$. The reference update under such a scheme would then be limited to $\kappa_{k+1} \in [0, \kappa']$. This modification could be advantageous if α must be very small to enforce that v_0 and/or v_r^* satisfy the conditions in Proposition 3.

(3) Direct optimization over $\mathcal{V}_\epsilon$: The SG relies on the scalar reference parameterization in (12). One could instead choose v_{k+1} by solving

$$\begin{aligned} \min_{v_{k+1} \in \mathcal{V}_\epsilon} \quad & \|v_r^* - v_{k+1}\|_L^2 \\ \text{s.t.} \quad & \|v_{k+1} - v_k\|_{W_v} \leq \sqrt{Nd + \lambda\alpha} - \hat{V}_{k+1} \end{aligned} \qquad (19)$$

for some $L \succ 0$ and $\hat{V}_{k+1} \geq V_N^\lambda(x_{k+1}, v_k)$. This formulation, referred to as the quadratically constrained SG (QC-SG), removes the restriction that $v_k \in \{v_0 + \kappa(v_r^* - v_0) \mid \kappa \in [0,1]\}$. The drawback being that (19) is a quadratically constrained quadratic program. Hence, the QC-SG is more computationally expensive to implement than the SG if $n_v > 1$. Moreover, the QC-SG optimizes v_{k+1} over the closed set $\mathcal{V}_\epsilon$ defined in Proposition 4. Therefore, it is necessary use a value of $\alpha > 0$ computed using Proposition 4 if $n_v > 1$. Note that the SG and QC-SG are equivalent in the case of $n_v = 1$.

(4) Offline sampling of the MPC policy: At each time step, the SG ensures that the Lyapunov descent condition in (9) holds by enforcing the sufficient condition $V_N^\lambda(x_k, v_k) \leq Nd + \lambda\alpha$. Hence, the performance of the SG can be improved if one can determine (e.g., using offline sampling) a value $V^* > Nd + \lambda\alpha$ such that $V_N^\lambda(x, v) \leq V^*$ implies that the descent condition in (9) holds. The SG scheme can then be altered to enforce that $V_N^\lambda(x_k, v_k) \leq V^*$ for all $k \geq 0$ by selecting

$$\tilde{\kappa}_{k+1} = \frac{\sqrt{V^*} - \sqrt{J_N(x_{k+1}, v_k, \tilde{\mathrm{u}}_{k+1})}}{\|v_r^* - v_k\|_{W_v}}, \text{if } v_k \neq v_r^*.$$

Remark 12 *The SG can be interpreted as a method of obtaining feasible (suboptimal) solutions to*

$$\begin{aligned} \min_{v \in \mathcal{V}} \quad & \|v_r^* - v\|_L^2, \\ s.t. \quad & x \in \mathcal{R}(v), \end{aligned} \qquad (20)$$

where $\mathcal{R}(v)$ represents the largest ROA of (6) when $v_k \equiv v$. In general, $\mathcal{R}(v)$ is non-convex and unknown, so solving (20) is not practically feasible. Instead, the SG enforces that $x_k \in \Upsilon_N^\lambda(v_k) \subset \mathcal{R}(v_k)$ through the use of tractable analytical bounds, while the aforementioned modifications act to improve how well the SG approximates the solution of (20).

6.2 Robustness-based Modifications

In this subsection, we briefly address modifications that can be made to the SG to ensure that the closed-loop system is inherently robust. Suppose that the plant in (1a) is instead $x_{k+1} = Ax_k + Bu_k + w_k$. Under the stated assumptions, V_N^λ is uniformly continuous [9, Proposition 1] and hence $\exists \xi \in \mathcal{K}$ such that $\forall x, x + \Delta x \in \mathcal{F}$:

$$|V_N^\lambda(x + \Delta x, v) - V_N^\lambda(x, v)| \leq \xi(\|\Delta x\|).$$

Let $\Omega(v_k, \alpha)$ be an implicit terminal set, $x_k \in \Upsilon_N^\lambda(v_k)$, and $u_k = u^*(x_k, v_k)$. Then,

$$\begin{aligned} V_N^\lambda(x_{k+1}, v_k) &\le V_N^\lambda(Ax_k + Bu_k, v_k) + \xi(\|w_k\|) \\ &\le J_N(Ax_k + Bu_k, v_k, \tilde{\mathrm{u}}_{k+1}) + \xi(\|w_k\|). \end{aligned}$$

Suppose that $\|w_k\| \le \xi^{-1}(\bar{w})$ for some $\bar{w} \in (0, d)$. Then, $V_N^\lambda(x_{k+1}, v_k) \le (N-1)d + \lambda\alpha + \bar{w} < Nd + \lambda\alpha$ by Lemma 8 and $\ell(x_k, u_k, v_k) \ge d > \xi(\|w_k\|)$.

Thus, if $\xi(\|w_k\|) \le \bar{w} < d$ for all $k \in \mathbb{N}$, then the reference v_{k+1} can be selected to enforce $x_{k+1} \in \Upsilon_N^\lambda(v_{k+1})$ by satisfying

$$\begin{aligned} \|v_{k+1} - v_k\|_{W_v} \le \sqrt{Nd + \lambda\alpha} \\ - \sqrt{J_N(x_{k+1}, v_k, \tilde{\mathrm{u}}_{k+1}) + \bar{w}}, \end{aligned}$$

instead of the bound in (11).

In summary, inherent robustness can be obtained by reducing the size of the reference steps. Hence, a simple heuristic method for promoting inherent robustness is to increment the reference command according to

$$v_{k+1} = v_k + \beta\kappa_{k+1}(v_r^* - v_k),$$

where $\beta \in (0, 1)$ is a fixed scale factor.

We also remark that the undisturbed closed-loop trajectories produced by SG-MPC will often satisfy $x_k \in \operatorname{Int} \Upsilon_N^\lambda(v_k)$ for all $k \ge 0$ (e.g., see Figures 3-4). This is caused by the use of analytical bounds in the development of the SG. Hence, there is some inherent robustness that is unintentionally introduced due to conservatism in the bounds used to compute the reference step.

7 Example

A linear model of an inverted pendulum on a cart is used for demonstration. The system states are $x = (s, \dot{s}, \phi, \dot{\phi})$ where s is the position of the cart and ϕ is the angle of the pendulum relative to the upright position. The continuous-time equations of motion are

$$\begin{aligned} \frac{4}{3}ml^2\ddot{\phi} - ml\ddot{s} = mgl\phi \\ (M+m)\ddot{s} - ml\ddot{\phi} = -b\dot{s} + F, \end{aligned}$$

where $m = 0.1$ kg is the mass of the pendulum, $l = 1$ m is the length of the pendulum, $g = 9.81$ m/s^2 is the gravitational constant, $M = 1$ kg is the mass of the cart, $b = 0.1$ Ns/m is the damping coefficient of the cart, and $u = F$ is the input force. The system is discretized using a zero-order hold with a sampling period of $T = 0.02$ s.

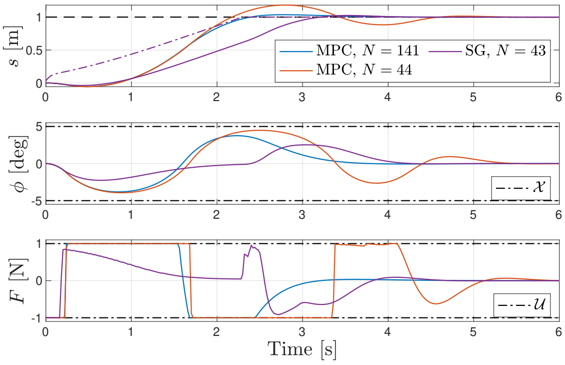
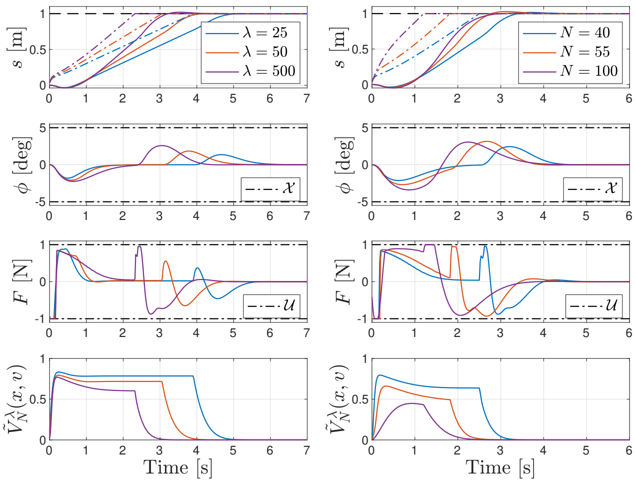


Fig. 2. Two ungoverned MPC implementations (labelled "MPC") compared to an SG-MPC implementation (labelled "SG"). Each implementation uses a terminal weight of $\lambda = 1000$. The dotted line in the top plot shows the reference command trajectory produced by the SG.

The tracking output is the cart position (i.e. $C = [1\,0\,0\,0]$ and $D = 0$) and the control task is to steer the system from $x_0 = (0, 0, 0, 0)$ to a setpoint of $r = 1$. The system constraints are $F \in [-1, 1]$ and $\phi \in [-5\pi/180, 5\pi/180]$. Weight matrices of $Q = \operatorname{diag}(10, 1, 10, 1)$ and $R = 0.1$ are used in all cases. In each SG-MPC implementation, the initial reference is chosen as $v_0 = 0$ unless otherwise specified. This choice automatically ensures that $x_0 \in \Upsilon_N^\lambda(v_0)$ for all $N \ge 1$ and $\lambda \ge 1$ since $x_0 = G_x v_0$, hence $V_N^\lambda(x_0, v_0) = 0$. Thus, selection of N and λ is solely a trade-off between closed-loop performance and computational cost when the SG is adopted in this example.

The terminal matrix P is obtained by solving (5) with $\eta = 1$ and K given by the LQR gain corresponding to $Q_K = \operatorname{diag}(5, 0.01, 120, 0.01)$ and $R_K = 0.5$. Proposition 3 yields a value of $\alpha = 0.875$ for this choice of v_0, P, and K. The matrices Q_K and R_K were tuned to increase the volume of $\Omega(v_r^*, \alpha)$ (Remark 5).

Figure 2 compares two ungoverned MPC implementations (i.e., executing the feedback law $u_k = u^*(x_k, 1)$) with $N \in \{44, 141\}$, and an SG implementation with $N = 43$. All three implementations use a terminal weight of $\lambda = 1000$. Note that $N = 141$ is the minimum horizon length necessary to enforce $x_0 \in \Upsilon_N^{1000}(1)$, whereas $N = 44$ is the minimum horizon length for which convergence and recursive feasibility are observed in simulation. Hence, Figure 2 demonstrates that the SG-MPC strategy results in recursively feasible and asymptotically stable trajectories in a case where the corresponding ungoverned MPC controller would lead to divergent trajectories. Moreover, the closed-loop performance of the SG-MPC strategy does not dramatically differ from the ungoverned MPC policies. For reference, the cumulative stage cost $\sum_k \ell(x_k, u_k, 1)$ for each implementation is 773.9 (MPC, $N = 141$), 814.5 (MPC, $N = 44$), and 871.7 (SG-MPC). However, the SG-MPC implementation guarantees stability with much smaller N.



(a) $N = 43$, λ varying (b) $\lambda = 500$, N varying

Fig. 3. Closed-loop trajectories resulting from SG-MPC implementations with different values of N and λ. The dotted lines in the top plots correspond to the reference command trajectories of each SG implementation.

Figure 3 compares the closed-loop trajectories that arise from SG-MPC implementations with: (a) $N = 43$ and $\lambda \in \{25, 50, 500\}$, and (b) $N \in \{40, 55, 100\}$ and $\lambda = 500$. As expected, the closed-loop performance and settling time of the reference command improves as N or λ increase. We define a normalized value function $\tilde{V}_N^\lambda(x,v) = \frac{V_N^\lambda(x,v)}{Nd+\lambda\alpha}$, such that $\tilde{V}_N^\lambda(x,v) \leq 1 \iff V_N^\lambda(x,v) \leq Nd + \lambda\alpha$. One can see that $\tilde{V}_N^\lambda(x_k, v_k)$ peaks at values between 0.44 and 0.84 due to the fact that the SG relies on the upper-bounds in Lemma 7 and 8. Moreover, the peak of $\tilde{V}_N^\lambda(x_k, v_k)$ tends to decrease as N and λ are increased — presumably due to these bounds becoming less tight.

Thus, it is natural to wonder how the closed-loop performance of the proposed SG differs from an idealized implementation. Hence, we compare: (1) an idealized SG that selects v_k such that $V_N^\lambda(x_k, v_k) = Nd + \lambda\alpha$ if $v_k \neq v_r^*$, (2) the proposed SG with v_0 selected so that $V_N^\lambda(x_0, v_0) = Nd + \lambda\alpha$, and (3) the proposed SG with $v_0 = 0$. Figure 4 compares the closed-loop performance of these three strategies in the case of $N = 100$ and $\lambda = 500$. The first strategy cannot be implemented in practice, but is shown for comparison as it represents the hypothetical best-case performance of the SG. The second strategy is used to demonstrate the effect of the reference initialization v_0 on the closed-loop performance.

As expected, the reference sequence produced by the ideal SG converges the quickest since it maintains $\tilde{V}_N^\lambda(x_k, v_k) = 1$ whenever $v_k \neq v_r^*$. Similarly, the proposed SG exhibits its best performance when provided with the ideal initial reference, since this maintains the value of $\tilde{V}_N^\lambda(x_k, v_k)$ closer to 1 in the transient phase. Despite these differences, however, the closed-loop performance of each method is comparable. In particular, the cumulative stage cost for each method is 776.2 (ideal), 776.5 (ideal v_0), and 794.3 ($v_0 = 0$). Hence, the proposed SG obtains comparable performance to its best-case performance even with a conservative initial reference selection of $v_0 = 0$. Moreover, this gap in performance can be reduced through a better choice of the initial reference v_0. Last, we remark that the peak in $\tilde{V}_N^\lambda(x_k, v_k)$ is particularly low when $N = 100$, $\lambda = 500$, and $v_0 = 0$ (see Figure 3). Hence, the small difference in performance between the three SG implementations in Figure 4 can be expected to be even smaller for cases with smaller N and/or λ.

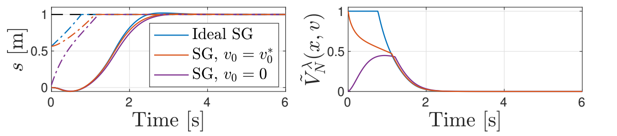


Fig. 4. A comparison of the ideal SG in (20), the SG with an ideal initial reference, and the SG with $v_0 = 0$. The dotted lines in the left plot correspond to the reference command trajectories of each SG implementation.

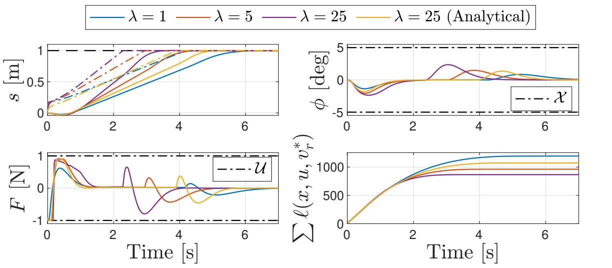


Fig. 5. Three SG-MPC implementations ($\lambda \in \{1, 5, 25\}$) based on numerically sampled ROAs compared to an SG-MPC implementation ($\lambda = 25$) based on an analytical ROA. A horizon length of $N = 43$ is used in each case. The accumulated stage cost is provided as a performance metric.

Finally, we note that the performance of the SG-MPC strategy is strongly tied to the value of the terminal weight λ. However, Figure 5 shows that the high performance can be obtained with small terminal weights λ if offline sampling [2] is used to obtain an expanded estimate of the ROA as discussed in Section 6.1. In particular, the SG-MPC based on a sampled ROA with $\lambda = 5$ and $\lambda = 25$ outperforms the analytical SG-MPC implementation with $\lambda = 25$. In fact, the sample-based implementation with $\lambda = 25$ slightly outperforms the analytical implementation with $\lambda = 1000$ in Figure 2 (accumulated stage costs of 866.5 and 871.7 respectively). Hence, large terminal weightings can be avoided if offline sampling of the MPC policy can be performed.

[2] The MPC policy was sampled in a 25^4 point uniform grid of state variables in $[0, 2] \times [-1.2, 1.2] \times [-5\pi/180, 5\pi/180] \times [-0.2, 0.2]$. The size of the sample-based ROA was defined to be 80% of the smallest value of $V_N^\lambda(x, 0)$ for which the descent condition in (9) did not hold.

8 Conclusions

This paper developed a supervisory scheme, called the stability governor, that expands the closed-loop region of attraction of a system controlled by MPC without terminal constraints. Theoretical guarantees were derived for the recursive feasibility of the MPC law, asymptotic stability of the target equilibrium, and finite-time convergence of the modified reference command. Moreover, it was shown that closed-loop stability under the supervised MPC strategy can be achieved using a much smaller prediction horizon length and terminal weighting parameter than an unsupervised MPC implementation; thereby reducing the computational cost of implementing the MPC law. Future work will investigate the extension of the stability governor to time-distributed MPC [7, 8] and nonlinear MPC.

Acknowledgements

The authors thank Dominic Liao-McPherson for his advice throughout the development of this work.

References

[1] M. S. Darup, G. Book, and P. Giselsson. Towards real-time ADMM for linear MPC. In *2019 18th European Control Conference (ECC)*, pages 4276–4282, 2019.

[2] Asen L Dontchev and R Tyrrell Rockafellar. Implicit functions and solution mappings. *Springer Monogr. Math.*, 2009.

[3] Gianluca Frison and Moritz Diehl. HPIPM: a high-performance quadratic programming framework for model predictive control. *IFAC-PapersOnLine*, 53(2):6563–6569, 2020. 21st IFAC World Congress.

[4] 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*, 36(9):1008–1020, 1991.

[5] Zhong-Ping Jiang and Yuan Wang. Input-to-state stability for discrete-time nonlinear systems. *Automatica*, 37(6):857–869, 2001.

[6] D. K. M. Kufoalor, S. Richter, L. Imsland, T. A. Johansen, M. Morari, and G. O. Eikrem. Embedded model predictive control on a plc using a primal-dual first-order method for a subsea separation process. In *22nd Mediterranean Conference on Control and Automation*, pages 368–373, 2014.

[7] Jordan Leung, Frank Permenter, and Ilya V. Kolmanovsky. A computationally governed log-domain interior-point method for model predictive control. In *2022 American Control Conference (ACC)*, pages 900–905, 2022.

[8] Dominic Liao-McPherson, Terrence Skibik, Jordan Leung, Ilya V. Kolmanovsky, and Marco M Nicotra. An analysis of closed-loop stability for linear model predictive control based on time-distributed optimization. *IEEE Transactions on Automatic Control*, pages 1–1, 2021.

[9] D. Limon, T. Alamo, D. M. Raimondo, D. Muñoz de la Peña, J. M. Bravo, A. Ferramosca, and E. F. Camacho. *Input-to-State Stability: A Unifying Framework for Robust Model Predictive Control*, pages 1–26. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[10] D. Limón, T. Alamo, F. Salas, and E. Camacho. On the stability of constrained MPC without terminal constraint. *IEEE Transactions on Automatic Control*, 51:832–836, 2006.

[11] D. Limon, I. Alvarado, T. Alamo, and E.F. Camacho. MPC for tracking piecewise constant references for constrained linear systems. *Automatica*, 44(9):2382–2387, 2008.

[12] Alexander Malyshev, Rien Quirynen, Andrew Knyazev, and Stefano Di Cairano. A regularized Newton solver for linear model predictive control. In *2018 European Control Conference (ECC)*, pages 1393–1398, 2018.

[13] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, 2000.

[14] Panagiotis Patrinos and Alberto Bemporad. An accelerated dual gradient-projection algorithm for embedded linear model predictive control. *IEEE Transactions on Automatic Control*, 59(1):18–33, 2014.

[15] Saša V. Raković and Sixing Zhang. Model predictive control with implicit terminal ingredients. *Automatica*, 151:110942, 2023.

[16] J.B. Rawlings, D. Q. Mayne, and M. M. Diehl. *Model Predictive Control: Theory, Computation, and Design.* Nob Hill Publishing, Madison, WI, 2nd edition, 2017.

[17] Terrence Skibik, Dominic Liao-McPherson, Torbjørn Cunis, Ilya Kolmanovsky, and Marco M. Nicotra. A feasibility governor for enlarging the region of attraction of linear model predictive controllers. *IEEE Transactions on Automatic Control*, 67(10):5501–5508, 2022.

[18] Terrence Skibik, Dominic Liao-McPherson, and Marco M. Nicotra. A terminal set feasibility governor for linear model predictive control. *IEEE Transactions on Automatic Control*, pages 1–7, 2022.

[19] Yang Wang and Stephen Boyd. Fast model predictive control using online optimization. *IEEE Transactions on Control Systems Technology*, 18(2):267–278, 2010.

[20] Yue Yu, Purnanand Elango, and Behçet Açıkmeşe. Proportional-integral projected gradient method for model predictive control. *IEEE Control Systems Letters*, 5(6):2174–2179, 2021.

A Condensed MPC Matrices

Let $\otimes$ denote the Kronecker product and define $\hat{G}_x = 1_N \otimes G_x$, $\hat{G}_u = 1_N \otimes G_u$, $\hat{R} = I_N \otimes R$, and $\hat{H} = \mathrm{blkdiag}(I_{N-1} \otimes Q, \lambda P)$, and

$$\hat{A} = \begin{bmatrix} I \\ A \\ \vdots \\ A^{N-1} \end{bmatrix}, \quad \hat{B} = \begin{bmatrix} 0 & \cdots & & 0 \\ B & 0 & \cdots & 0 \\ AB & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \\ A^{N-1}B & \ldots & AB & B \end{bmatrix}.$$

Then, the matrices in (10) are

$$H = \hat{R} + \hat{B}^T \hat{H} \hat{B},\ F_x = \hat{B}^T \hat{H} \hat{A},\ F_v = -\hat{B}^T \hat{H} \hat{G}_x - \hat{R} \hat{G}_u$$

$$F = [F_x\ F_v],\ W_x = \hat{A}^T \hat{H} \hat{A}, \quad W_{xv} = -\hat{A}^T \hat{H} \hat{G}_x,$$

$$W_v = G_x^T(\lambda P + NQ)G_x + NG_u^T R G_u,\; W = \begin{bmatrix} W_x & W_{xv} \\ W_{xv}^T & W_v \end{bmatrix}.$$

B Proof of Propositions

PROOF. (*Proposition 3*) Assume the conditions in the statement of the proposition hold. The proof follows by demonstrating that for all $v \in \{v_0 + \kappa(v_r^* - v_0) | \kappa \in [0,1]\}$,

$$\Omega(v, \alpha) \subset \{x \in \mathcal{X} \mid [G_u v - K(x - G_x v)] \in \mathcal{U}\}.$$

Let $\kappa \in [0,1]$ and $v = v_0 + \kappa(v_r^* - v_0)$. First, we prove that $\Omega(v,\alpha) \subset \mathcal{X}$. Let $x \in \Omega(v,\alpha)$ and define $\tilde{x}_1 = x - G_x(v - v_0)$, $\tilde{x}_2 = x - G_x(v - v_r^*)$. Then,

$$\|\tilde{x}_1 - G_x v_0\|_P^2 = \|\tilde{x}_2 - G_x v_r^*\|_P^2 = \|x - G_x v\|_P^2 \leq \alpha,$$

i.e., $\tilde{x}_1 \in \Omega(v_0, \alpha) \subset \mathcal{X}$ and $\tilde{x}_2 \in \Omega(v_r^*, \alpha) \subset \mathcal{X}$. Moreover,

$$\begin{aligned} x &= x + G_x(v_0 + \kappa(v_r^* - v_0) - v) \\ &= \kappa(x - G_x(v - v_r^*)) + (1-\kappa)(x - G_x(v - v_0)) \\ &= \kappa\tilde{x}_2 + (1-\kappa)\tilde{x}_1. \end{aligned}$$

Thus, x is a convex combination of points $\tilde{x}_1, \tilde{x}_2 \in \mathcal{X}$, so $x \in \mathcal{X}$ and $\Omega(v,\alpha) \subset \mathcal{X}$.

Next, we prove $\Omega(v,\alpha) \subset \{x \mid G_u v - K(x - G_x v) \in \mathcal{U}\}$. Since $\tilde{x}_1 \in \Omega(v_0, \alpha)$ and $\tilde{x}_2 \in \Omega(v_r^*, \alpha)$, then $\tilde{u}_1 = G_u v_0 - K(\tilde{x}_1 - G_x v_0) \in \mathcal{U}$ and $\tilde{u}_2 = G_u v_r^* - K(\tilde{x}_2 - G_x v_r^*) \in \mathcal{U}$. Define $u = G_u v - K(x - G_x v)$. It follows that

$$\begin{aligned} u &= G_u(v_0 + \kappa(v_r^* - v_0)) \\ &\quad - K\left[\kappa\tilde{x}_2 + (1-\kappa)\tilde{x}_1 - G_x(v_0 + \kappa(v_r^* - v_0))\right] \\ &= \kappa\tilde{u}_2 + (1-\kappa)\tilde{u}_1. \end{aligned}$$

and so $u \in \mathcal{U}$ by convexity of $\mathcal{U}$. Thus, $x \in \{x \mid G_u v - K(x - G_x v) \in \mathcal{U}\}$ and so $\Omega(v,\alpha) \subset \{x \mid G_u v - K(x - G_x v) \in \mathcal{U}\}$.

PROOF. (*Proposition 4*) Assume the conditions in the statement of the proposition hold. The proof follows by showing that for all $v \in \mathcal{V}_\epsilon$,

$$\Omega(v, \alpha) \subseteq \{x \in \mathcal{X} \mid [G_u v - K(x - G_x v)] \in \mathcal{U}\},$$

which implies the result by Proposition 1. To begin, we show that (16) implies that $\Omega(v,\alpha) \subset \mathcal{X}$ for all $v \in \mathcal{V}_\epsilon$. First, note that $\forall v \in \mathcal{V}_\epsilon$, $(x - G_x v) \in \epsilon_x \mathcal{X}$ implies that $x \in \mathcal{X}$. To see this, note that $(x - G_x v) \in \epsilon_x \mathcal{X} \implies \exists x_1 \in \mathcal{X}$ such that $x - G_x v = \epsilon_x x_1$. Moreover, $G_x v \in (1 - \epsilon_x)\mathcal{X} \implies \exists x_2 \in \mathcal{X}$ such that $G_x v = (1-\epsilon_x)x_2$. Therefore, $x = \epsilon_x x_1 + (1-\epsilon_x)x_2$, where $\epsilon_x \in (0,1), x_1, x_2 \in \mathcal{X}$, and so $x \in \mathcal{X}$ by convexity. Thus, $x \in \Omega(v,\alpha) \implies \|x - G_x v\|_P^2 \leq \alpha \implies (x - G_x v) \in \epsilon_x \mathcal{X} \implies x \in \mathcal{X}$, where the second implication follows from (16). Thus, $\Omega(v,\alpha) \subset \mathcal{X}$ for all $v \in \mathcal{V}_\epsilon$, and we note that $\text{Int}(\epsilon_x \mathcal{X}) \ni 0$ by Assumption 1.

Next, we show that (17) (if $G_u = 0$) and (18) (if $G_u \neq 0$) implies that $\Omega(v,\alpha) \subset \{x \mid [G_u v - K(x - G_x v)] \in \mathcal{U}\}$ for all $v \in \mathcal{V}_\epsilon$. First, consider the case of $G_u = 0$. The conclusion that $\Omega(v,\alpha) \subset \{x \mid [-K(x - G_x v)] \in \mathcal{U}\}$ follows directly from (17). Now consider the case of $G_u \neq 0$. Note that $\forall v \in \mathcal{V}_\epsilon$, $[-K(x - G_x v)] \in \epsilon_u \mathcal{U}$ implies that $[G_u v - K(x - G_x v)] \in \mathcal{U}$. This follows from identical convexity arguments used to prove the previous result. Thus, $x \in \Omega(v,\alpha) \implies \|x - G_x v\|_P^2 \leq \alpha \implies -K(x - G_x v) \in \epsilon_u \mathcal{U} \implies [G_u v - K(x - G_x v)] \in \mathcal{U}$, where the second implication follows from (18). Thus, $\Omega(v,\alpha) \subset \{x \mid [G_u v - K(x - G_x v)] \in \mathcal{U}\}$ for all $v \in \mathcal{V}_\epsilon$, and we note that $\text{Int}(\epsilon_u \mathcal{U}) \ni 0$ by Assumption 1.

Jordan Leung is a Ph.D. candidate in the department of aerospace engineering at the University of Michigan. He received his MASc. in aerospace engineering from the University of Toronto in 2019, and his BASc. in engineering physics from Queen's University in 2017. His research interests include constrained control and real-time optimization with applications in aerospace and autonomous systems.

Frank N. Permenter is a Staff Research Scientist at the Toyota Research Institute (TRI) with research interests in optimization and control. He received his Ph.D. degree in electrical engineering and computer science from MIT in 2017. Prior to joining TRI, he worked at NASA Johnson Space Center on the International Space Station and Robonaut programs.

Ilya V. Kolmanovsky is a professor in the department of aerospace engineering at the University of Michigan with research interests in control theory for systems with state and control constraints, and in control applications to aerospace and automotive systems. He received his Ph.D. degree in aerospace engineering from the University of Michigan in 1995. Prior to joining the University of Michigan as a faculty in 2010, he was with Ford Research and Advanced Engineering in Dearborn, Michigan for close to 15 years.