A Computational Governor for Maintaining Feasibility and Low Computational Cost in Model Predictive Control

Jordan Leung, Frank Permenter, and Ilya V. Kolmanovsky, *Fellow, IEEE*

***Abstract*— This paper introduces an approach for reducing the computational cost of implementing Linear Quadratic Model Predictive Control (MPC) for set-point tracking subject to pointwise-in-time state and control constraints. The approach consists of three key components: First, a log-domain interior-point method used to solve the receding horizon optimal control problems; second, a method of warm-starting this optimizer by using the MPC solution from the previous timestep; and third, a computational governor that maintains feasibility and bounds the suboptimality of the warm-start by altering the reference command provided to the MPC problem. Theoretical guarantees regarding the recursive feasibility of the MPC problem, asymptotic stability of the target equilibrium, and finite-time convergence of the reference signal are provided for the resulting closed-loop system. In a numerical experiment on a lateral vehicle dynamics model, the worst-case execution time of a standard MPC implementation is reduced by over a factor of 10 when the computational governor is added to the closed-loop system.**

***Index Terms*— Interior-point methods, model predictive control (MPC), quadratic programming, stability analysis.**

I. Introduction

Model Predictive Control (MPC) is a broadly used feedback strategy capable of providing high-performance control in the presence of system constraints [1]. MPC has been successfully applied in settings such as industrial process control [2], hybrid electric vehicle energy management [3], vehicle stabilization at the limits of handling [4], and spacecraft rendezvous maneuvers [5]. Moreover, theoretical properties of MPC have been extensively studied [1], [6]–[9].

Stability guarantees for MPC are often established through the choice of the terminal penalty, the terminal set constraint, and the prediction horizon length in the receding horizon Optimal Control Problem (OCP) [6]. In such formulations, the region of attraction (ROA) of the closed-loop system corresponds to the set of states for which the OCP is feasible [6]. It is well known that the size of this ROA is directly related to the length of the prediction horizon. However, lengthening the prediction horizon has the undesirable consequence of increasing the computational cost of solving the OCPs. Thus, it is often challenging to implement MPC in a way that is both computationally efficient and supported by strong guarantees of closed-loop stability and recursive feasibility.

The development of efficient optimization algorithms designed to solve these OCPs has helped address this issue [10]–[15]; nevertheless implementing MPC in applications with fast sampling rates and/or limited computing power still remains challenging. A common approach to reduce the computational burden of MPC is to approximate the OCP solutions by performing a limited number of optimizer iterations per timestep — a procedure referred to as suboptimal MPC. However, careful consideration must be given to the stability of the closed-loop system when suboptimal MPC is used, since the guarantees provided by optimal MPC do not necessarily hold. The stability and robustness of such methods has been studied in generalized theoretical frameworks (e.g., [16]–[18]). However, computable certification bounds are often somewhat conservative and limited to simple cases (e.g., input constrained linear systems [19]–[21]), and more complex cases often require significant modification of the OCP (e.g., constraint tightening [22], stabilizing constraints [23], and the use of relaxed barrier functions [24]).

Alternatively, the computational cost of MPC can be reduced by shortening the prediction horizon and employing other methods for enlarging the ROA. One such method involves including a parameterization of the terminal set constraint as an optimization variable in the OCP [25]. However, this results in increased dimensionality of the OCP. An alternative method is to replace the terminal set constraint with a contractive sequence of sets that are computed offline [26]. The feasibility governors proposed in [27] and [28] are add-on units that expand the ROA of MPC by altering the reference (set-point) command supplied to the OCP. Under this paradigm, the structure and complexity of the OCP remains unchanged and only the supplied set-point command is altered.

The strategy proposed in this paper for reducing the computational cost of MPC consists of three key components: First, a log-domain interior-point method used to solve the receding horizon OCPs; second, a method of warm-starting this optimizer by using the MPC solution from the previous timestep; and third, a *computational governor* (CG) that alters the MPC reference command so that the suboptimality of the warm-start is bounded and the resulting OCP is feasible. The warm-

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.

J. Leung and I. Kolmanovsky are with the University of Michigan, Ann Arbor, MI 48109 USA (e-mail: {jmleung, ilya}@umich.edu).

F. Permenter is with the Toyota Research Institute, Cambridge, MA 02139 USA (email:frank.permenter@tri.global).

start suboptimality bound enforced by the CG ensures that the ensuing OCP is well-initialized and can be solved with low computational cost. Meanwhile, the OCP feasibility constraint ensures recursive feasibility of the MPC problem and thus results in an expansion of the closed-loop ROA. Thus, the CG can reduce the computational cost of implementing MPC by simultaneously ensuring that the OCP is well-initialized and by allowing for the use of a shorter prediction horizon. To the authors' best knowledge, this scheme is unique in its ability to simultaneously ensure good initialization of the OCP while also expanding the closed-loop ROA.

This paper is an extension of [29], which only explored the use of the CG to ensure good initialization of the OCP. The utility of the CG is expanded in this paper by also considering the effect of expanding the closed-loop ROA. Specifically, the new contributions introduced in this paper are theoretical guarantees regarding the recursive feasibility of the MPC problem, asymptotic stability of the target equilibrium, expansion of the closed-loop ROA, and finite-time convergence of the computationally governed reference signal. Numerical experiments are reported which demonstrate the effectiveness of the CG when it is used as a means of both expanding the ROA and ensuring good initialization of the OCP.

The paper is organized as follows. Section II introduces the MPC formulation used to generate control inputs. Section III discusses how the log-domain interior-point method (LDIPM) is used to solve the receding horizon OCP. Section IV introduces the computational governor and Section V establishes the theoretical properties of the resulting closed-loop system. Section VI discusses implementation aspects of the computational governor and Section VII demonstrates the efficacy of the computational governor through numerical experiments.

Notation: Let $\mathbb{R}^n_{>0}$ denote the set of real $n \times 1$ vectors with strictly positive elements (define $\mathbb{R}^n_{\geq 0}$ accordingly). 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 $\|x\|_W = \sqrt{x^T W x}$. Let $\|\cdot\|$ represent the 2-norm when no subscript is specified. Given $A \in \mathbb{R}^{m\times n}$ and $p \in [1, \infty]$, let $\|A\|_p$ represent the induced p-norm. Let $\lambda_{\min}(A)$ and $\lambda_{\max}(A)$ denote the minimum and maximum eigenvalues of $A \in \mathbb{R}^{n\times n}$. Let $\mathcal{B}(x, \delta) \subset \mathbb{R}^n$ represent a closed 2-norm ball of radius $\delta > 0$ centered at $x \in \mathbb{R}^n$. Given $\rho \in \mathbb{R}$ and $X \subset \mathbb{R}^n$, let $\rho X = \{\rho x \mid x \in X\}$. Given $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^n$, let $(x \odot y) \in \mathbb{R}^n$ denote the elementwise multiplication and $x^{-1} \in \mathbb{R}^n$ denote the elementwise inverse. Let $(x, y) = [x^T \; y^T]^T$. If $f : \mathbb{R} \to \mathbb{R}$ and $x \in \mathbb{R}^n$, then $f(x) \in \mathbb{R}^n$ is understood to be an elementwise operation. Given $x \in \mathbb{R}^n$, let $\mathrm{diag}(x) = \mathrm{diag}(x_1, ..., x_n)$ denote the $n \times n$ diagonal matrix containing x_i in the i^{th} diagonal element for all $i \in \mathbb{N}_{[1,n]}$.

II. Problem Setting

We consider a linear time-invariant discrete-time system given by

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

where $k \in \mathbb{N}$ is the discrete-time index, $x_k \in \mathbb{R}^n$ is the state, $u_k \in \mathbb{R}^{n_u}$ is the control, $y_k \in \mathbb{R}^{n_y}$ is the constrained output, and $z_k \in \mathbb{R}^{n_z}$ is the tracking output[1]. The control objective is to drive the tracking output z_k to a desired reference $r \in \mathbb{R}^{n_z}$ subject to pointwise-in-time constraints

$$y_k \in \mathcal{Y}, \; \forall k \in \mathbb{N}, \tag{2}$$

where $\mathcal{Y} \subseteq \mathbb{R}^{n_y}$ is a specified constraint set.

Assumption 1: The pair (A, B) is stabilizable and $\mathcal{Y}$ is a compact polyhedron that contains the origin in the interior.

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

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

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 $\mathrm{Null}(Z) \neq \{0\}$ [25], [27]. The following assumption excludes ill-posed reference tracking problems, e.g., G_z = 0, and ensures that the target reference r uniquely determines the target equilibrium.

Assumption 2: The matrix G_z is full rank and $n_z = n_v$.

Remark 1: It is also possible to formulate reference tracking problems with G_z full rank and $n_z < n_v$ [27]. The assumption $n_z = n_v$ is made to simplify the computational governor introduced in Section IV.

A reference tracking MPC strategy similar to [25] is employed to solve the specified control problem. The following OCP is used to generate the MPC feedback law:

$$\min_{(\hat{\mathrm{x}}, \hat{\mathrm{u}})} \; ||\hat{x}_N - \bar{x}_v||_P^2 + \sum_{i=0}^{N-1} ||\hat{x}_i - \bar{x}_v||_Q^2 + ||\hat{u}_i - \bar{u}_v||_R^2 \tag{4a}$$
$$\text{s.t.} \quad \hat{x}_0 = x, \tag{4b}$$
$$\hat{x}_{i+1} = A\hat{x}_i + B\hat{u}_i, \quad i \in \mathbb{N}_{[0,N-1]}, \tag{4c}$$
$$C\hat{x}_i + D\hat{u}_i \in \mathcal{Y}, \quad i \in \mathbb{N}_{[0,N-1]}, \tag{4d}$$
$$(\hat{x}_N, v) \in \mathcal{T}, \tag{4e}$$

where $N \in \mathbb{N}$ is the prediction horizon, $\hat{\mathrm{x}} = (\hat{x}_0, \ldots, \hat{x}_N)$ and $\hat{\mathrm{u}} = (\hat{u}_0, \ldots, \hat{u}_{N-1})$ are the predicted state and control sequences, $P \in \mathbb{R}^{n\times n}, Q \in \mathbb{R}^{n\times n}, R \in \mathbb{R}^{n_u \times n_u}$ are weighting matrices, and $\mathcal{T} \subseteq \mathbb{R}^{n+n_v}$ is a terminal set. The initial state x and reference command v are parameters in this OCP. We use the notation $\mathcal{P}_N(x, v)$ to refer to problem (4) specified with parameters (x, v). The following assumption ensures that $\mathcal{P}_N(x, v)$ can be used to generate a stabilizing feedback law.

Assumption 3: The cost matrices satisfy $Q \succ 0$, $R \succ 0$, and P is the positive-definite solution to the Discrete Algebraic Riccati Equation: $P = Q + A^T PA - A^T PBK$, where $K = (R + B^T PB)^{-1}(B^T PA)$ is the Linear Quadratic Regulator (LQR) gain.

[1]The constrained output y_k and the tracking output z_k may share common elements. In this case, the matrices C and D will share common rows with E and F respectively.

We define $\mathcal{T} = \mathcal{O}_\infty$, where $\mathcal{O}_\infty \subseteq \mathbb{R}^{n+n_v}$ is the maximum constraint admissible set for the closed-loop system under LQR feedback [30]. That is, $\mathcal{O}_\infty$ is the maximal set of pairs (x, v) such that if LQR is applied to (1) with $x_0 = x$ and a constant reference v, then (2) is satisfied. Since $\mathcal{O}_\infty$ is polyhedral under Assumption 1 [30], then $\mathcal{P}_N(x, v)$ can be written in the following condensed form

$$\min_{\hat{\mathrm{u}}} \quad \frac{1}{2}\hat{\mathrm{u}}^T H \hat{\mathrm{u}} + \hat{\mathrm{u}}^T W\theta \tag{5a}$$

$$\text{s.t.} \quad M\hat{\mathrm{u}} + L\theta + \ell \geq 0, \tag{5b}$$

where $\theta = (x, v)$ and $H \succ 0, W, M, L, \ell$ are defined in [27] as a function of the problem data in (4) and the parameterization in (3). We define the cost function J, feasible set map $\mathcal{F}$, and value function V of $\mathcal{P}_N(x, v)$ as

$$J(x, v, \hat{\mathrm{u}}) = \frac{1}{2}\hat{\mathrm{u}}^T H \hat{\mathrm{u}} + \hat{\mathrm{u}}^T W\theta, \tag{6a}$$

$$\mathcal{F}(x, v) = \{\hat{\mathrm{u}} \mid M\hat{\mathrm{u}} + L\theta + \ell \geq 0\}, \tag{6b}$$

$$V(x, v) = \min_{\hat{\mathrm{u}} \in \mathcal{F}(x,v)} J(x, v, \hat{\mathrm{u}}). \tag{6c}$$

The set of feasible parameters for (5) is

$$\Gamma_N = \{(x, v) \in \mathbb{R}^{n+n_v} \mid \mathcal{F}(x, v) \neq \emptyset\},$$

which is equivalent to the N-step backwards reachable set to $\mathcal{O}_\infty$. Note that both $\mathcal{O}_\infty$ and Γ_N are polyhedral sets with representations that can be computed offline [27], [30]. The following set-valued maps are defined for convenience:

$$\mathcal{O}_\infty(v) = \{x \in \mathbb{R}^n \mid (x, v) \in \mathcal{O}_\infty\},$$
$$\Gamma_N(v) = \{x \in \mathbb{R}^n \mid (x, v) \in \Gamma_N\}.$$

The closed-loop system under MPC feedback is defined by

$$x_{k+1} = Ax_k + Bu^*(x_k, v_k), \tag{7a}$$

$$u^*(x_k, v_k) = \Xi\hat{\mathrm{u}}^*(x_k, v_k), \tag{7b}$$

where $\{v_k\}$ is a sequence of reference commands, $\hat{\mathrm{u}}^*(x, v)$ is the optimal solution to $\mathcal{P}_N(x, v)$, and $\Xi = [I_{n_u}\ 0\ \ldots\ 0]$ is a matrix that selects the first control input. The following theorem details the stability and convergence properties of the closed-loop MPC system for a constant reference $v_k \equiv v$.

Theorem 1: ([9, Theorem 4.4.2], [27, Theorem 1]) Let Assumptions 1-3 hold and consider the closed-loop system (7) starting from an initial condition x_0 with a constant reference $v_k \equiv v$. Then $\forall (x_0, v) \in \Gamma_N$:

- $(x_k, v) \in \Gamma_N$ for all $k \in \mathbb{N}$;
- $y_k \in \mathcal{Y}$ for all $k \in \mathbb{N}$;
- $\lim_{k\to\infty} x_k = \bar{x}_v$.

If, in addition, $v \in \operatorname{Int} \mathcal{V}$ then $\bar{x}_v$ is asymptotically stable, where $\mathcal{V} = \{v \mid G_y v \in \mathcal{Y}\}$ is the set of constraint admissible references and $G_y = CG_x + DG_u$.

Assumption 4: The target reference r is constant and satisfies $r \in \mathcal{V}_\epsilon$, where $\mathcal{V}_\epsilon = \{v \in \mathbb{R}^z \mid G_y v \in (1-\epsilon)\mathcal{Y}\}$, and $0 < \epsilon \ll 1$ is a parameter chosen to ensure $r \in \operatorname{Int} \mathcal{V}$.

Remark 2: The target reference r is assumed to be constant to simplify statements of closed-loop stability. However, the results herein can naturally be applied to time-varying reference sequences $\{r_k\}$ where for every $i \in \mathbb{N}$ there exists an interval $\mathcal{I}_i \subset \mathbb{N}$ of non-zero length and a reference r_i such that $i \in \mathcal{I}_i$ and $r_k = r_i$ for all $k \in \mathcal{I}_i$. The reference changes can then be interpreted as new instances of a constant reference as long as these intervals $\mathcal{I}_i$ are sufficiently long.

In theory, the stated control problem can be solved by simply specifying $v_k \equiv r$ if $(x_0, r) \in \Gamma_N$. In this case, $\bar{x}_r = G_x r$ is an asymptotically stable equilibrium of (7). However, if $\bar{x}_r$ is far from x_0, it may be necessary for N to be very large to enforce $(x_0, r) \in \Gamma_N$. As previously discussed, the computational cost of solving (5) increases with increased N. Therefore, choosing N to satisfy $(x_0, r) \in \Gamma_N$ may make solving (5) computationally infeasible.

To this end, we propose a computational governor for selecting $\{v_k\}$ that removes the requirement that $(x_0, r) \in \Gamma_N$ and also ensures good initialization of the optimization algorithm used to solve $\mathcal{P}_N(x_k, v_k)$. The computational governor can then reduce the computational cost of implementing MPC by allowing for the use of shorter prediction horizons and by expediting the convergence of the optimization algorithm used to solve $\mathcal{P}_N(x_k, v_k)$.

The computational governor proposed in this paper is derived by exploiting properties of the log-domain interior-point method (LDIPM) used to solve (5). In Section IV, we will show that the Newton step of the LDIPM can be directly parameterized by the reference command v_k. This observation facilitates the development of an optimization-based strategy that selects the reference command v_k so that a primal-dual feasible solution to (5) with bounded suboptimality is obtained upon warm-starting the LDIPM. As a preamble to this development, the following section describes some important properties of the LDIPM that are used to derive the computational governor.

III. Application of the Log-Domain Interior-point Method to MPC

In this section, we describe how the log-domain interior-point method (LDIPM) from [31] can be used to solve (5).

A. Properties of the LDIPM

Consider the QP in (5) for a fixed parameter $\theta = (x, v) \in \Gamma_N$. For ease of notation, we define $c = W\theta$, $b = L\theta + \ell$, $p = Nn_u$, and $m \in \mathbb{N}$ as the number of rows of $M \in \mathbb{R}^{m\times p}$. Consider the following *central-path* equations for (5)

$$M^T\lambda = H\hat{\mathrm{u}} + c, \quad s = M\hat{\mathrm{u}} + b, \tag{8a}$$

$$\lambda \geq 0,\ s \geq 0, \quad s_i\lambda_i = \eta,\ i \in \mathbb{N}_{[1,m]}, \tag{8b}$$

where $\eta > 0$ is a fixed homotopy parameter, $s \in \mathbb{R}^m$ is the constraint slack, and $\lambda \in \mathbb{R}^m$ is the vector of dual variables. Note that when $\eta = 0$, these equations reduce to the Karush-Kuhn-Tucker (KKT) optimality conditions for (5). Next, consider the following logarithmic change-of-variables. Let $\gamma \in \mathbb{R}^m$ and define $\lambda = \sqrt{\eta}e^{\gamma}$ and $s = \sqrt{\eta}e^{-\gamma}$, such that the *log-domain central-path* equations are

$$\sqrt{\eta}M^T e^{\gamma} = H\hat{\mathrm{u}} + c, \quad \sqrt{\eta}e^{-\gamma} = M\hat{\mathrm{u}} + b. \tag{9}$$

Note that the conditions in (8b) are satisfied for all $\gamma \in \mathbb{R}^m$ by definition. We define the *central-path* as the graph of the map $\eta \mapsto (\hat{\mathrm{u}}, \gamma)$ such that $(\hat{\mathrm{u}}, \gamma, \eta)$ satisfy (9).

The LDIPM solves (5) by applying Newton's method to (9) with a decreasing sequence of η. The Newton direction $d = d(\gamma, \eta) \in \mathbb{R}^m$ and the primal variable update $\hat{\mathrm{u}} = \hat{\mathrm{u}}(\gamma, \eta)$ of the LDIPM satisfy

$$\sqrt{\eta} M^T (e^{\gamma} + e^{\gamma} \odot d) = H\hat{\mathrm{u}} + c, \tag{10a}$$
$$\sqrt{\eta} (e^{-\gamma} - e^{-\gamma} \odot d) = M\hat{\mathrm{u}} + b. \tag{10b}$$

The following theorem establishes uniqueness of $d(\gamma, \eta)$ and $\hat{\mathrm{u}}(\gamma, \eta)$, and provides a method of computing both.

Theorem 2: ([31, Theorem 2.1]) For all $\gamma \in \mathbb{R}^m$ and $\eta > 0$, the Newton direction $d = d(\gamma, \eta)$ and the decision variable update $\hat{\mathrm{u}} = \hat{\mathrm{u}}(\gamma, \eta)$ satisfy

$$d = \mathbf{1} - \frac{1}{\sqrt{\eta}} e^{\gamma} \odot (M\hat{\mathrm{u}} + b), \tag{11a}$$
$$(M^T \mathcal{D}(\gamma) M + H)\hat{\mathrm{u}} = 2\sqrt{\eta} M^T e^{\gamma} - (c + M^T \mathcal{D}(\gamma) b), \tag{11b}$$

where $\mathbf{1} \in \mathbb{R}^m$ is a vector of ones and $\mathcal{D}(\gamma) = \mathrm{diag}(e^{2\gamma})$. Moreover, $M^T \mathcal{D}(\gamma) M + H \succ 0$.

Iterates generated by the update rule

$$\gamma_{i+1} = \gamma_i + \frac{1}{\alpha_i} d(\gamma_i, \eta), \tag{12a}$$
$$\alpha_i = \max\left\{1, \|d(\gamma_i, \eta)\|_\infty^2\right\}, \tag{12b}$$

are globally convergent to the central-path point $(\hat{\mathrm{u}}, \gamma, \eta)$ [31, Theorem 2.3]. The following lemma describes conditions under which iterates are primal-dual feasible with bounded suboptimality.

Lemma 1: ([31, Lemma 3.3]) For $\eta > 0$ and $\gamma \in \mathbb{R}^m$, let $d = d(\gamma, \eta)$ and $\hat{\mathrm{u}} = \hat{\mathrm{u}}(\gamma, \eta)$. Let $\lambda = \sqrt{\eta}(e^{\gamma} + e^{\gamma} \odot d)$ and $s = \sqrt{\eta}(e^{-\gamma} - e^{-\gamma} \odot d)$. If $\|d(\gamma, \eta)\|_\infty \leq 1$, then $(\hat{\mathrm{u}}, s, \lambda)$ satisfy the primal-dual feasibility conditions

$$M\hat{\mathrm{u}} + b = s, \;\; M^T\lambda = H\hat{\mathrm{u}} + c, \;\; \lambda \geq 0, \;\; s \geq 0.$$

Further, $\|s \odot \lambda\|_1 = \eta(m - \|d\|^2)$.

The condition $\|d(\gamma, \eta)\|_\infty \leq 1$ also implies that the update in (12) decreases the *divergence*[2] by a factor of $\frac{1}{2}$ [31, Theorem 3.2]. In other words, if the Newton step satisfies $\|d(\gamma, \eta)\|_\infty \leq 1$, then the iterate resulting from (12) is primal-dual feasible, has a suboptimality bounded by η, and reduces a measure of convergence by (at least) a factor of $\frac{1}{2}$. These observations motivate the *longstep* implementation of the LDIPM [31, Section 3.2] described by Algorithm 1.

The longstep procedure reduces η by computing

$$\eta^*(\gamma) := \inf\{\eta > 0 \mid \|d(\gamma, \eta)\|_\infty \leq 1\}, \tag{13}$$

where $\eta^*(\gamma) := \infty$ when (13) is infeasible. In other words, Algorithm 1 performs the largest possible reduction of η that ensures the condition $\|d(\gamma, \eta)\|_\infty \leq 1$ holds. The procedure used to compute $\eta^*(\gamma)$ is given in [31, Section 3.2].

[2]A metric-like function that is used to establish convergence of the LDIPM in [31]. Specifically, the divergence of $\gamma \in \mathbb{R}^m$ and a centered-point $\hat{\gamma} \in \mathbb{R}^m$ is defined by $h(\gamma, \hat{\gamma}) = \langle e^{\gamma}, e^{-\hat{\gamma}} \rangle + \langle e^{-\gamma}, e^{\hat{\gamma}} \rangle - 2m$, where $\langle \cdot, \cdot \rangle$ is the Euclidean inner-product [31, Definition 2.3].

Algorithm 1 LDIPM$(x, v, \gamma_{\text{init}}, \eta_{\text{init}}, \eta_{\text{final}})$

$\gamma \leftarrow \gamma_{\text{init}}$, $\eta \leftarrow \eta_{\text{init}}$, $\eta_{\text{tol}} \leq \eta_{\text{final}}$
$\theta = (x, v)$, $c = W\theta$, $b = L\theta + \ell$
while $\eta > \eta_{\text{final}}$ **or** $\|d(\gamma, \eta)\|_\infty > 1$ **do**
  $\eta \leftarrow \min\{\eta, \inf\{\eta > 0 \mid \|d(\gamma, \eta)\|_\infty \leq 1\}\}$
  $\eta \leftarrow \max\{\eta, \eta_{\text{tol}}\}$
  $\alpha \leftarrow \max\{1, \|d(\gamma, \eta)\|_\infty^2\}$
  $\gamma \leftarrow \gamma + \frac{1}{\alpha} d(\gamma, \eta)$
end while
$\hat{\mathrm{u}} \leftarrow (M^T \mathcal{D}(\gamma) M + H)^{-1} [2\sqrt{\eta} M^T e^{\gamma} - (c + M^T \mathcal{D}(\gamma) b)]$,
return $(\hat{\mathrm{u}}, \gamma, \eta)$

The following theorem states that Algorithm 1 terminates globally.

Theorem 3: ([31, Theorem 3.2]) For any inputs $(x, v) \in \Gamma_N$ and $(\gamma_{\text{init}}, \eta_{\text{init}}, \eta_{\text{final}}) \in \mathbb{R}^m \times \mathbb{R}_{>0} \times \mathbb{R}_{>0}$, Algorithm 1 terminates and returns $(\hat{\mathrm{u}}, \gamma, \eta)$ such that $\eta \leq \eta_{\text{final}}$, $\|d(\gamma, \eta)\|_\infty \leq 1$, and

$$\hat{\mathrm{u}} \in \mathcal{F}(x, v), \qquad J(x, v, \hat{\mathrm{u}}) \leq V(x, v) + m\eta,$$

where $\mathcal{F}$, J, and V are defined in (6).

B. Warm-starting the LDIPM

Next, we discuss how the LDIPM can be warm-started at timestep k given the solution to the OCP $\mathcal{P}_N(x_{k-1}, v_{k-1})$ at the previous timestep. Let $(\hat{\mathrm{u}}_k, \gamma_k, \eta_k)$ denote the output of Algorithm 1 applied to $\mathcal{P}_N(x_k, v_k)$ at timestep $k \in \mathbb{N}$. We label the block elements of $\hat{\mathrm{u}}_k$ using the notation $\hat{\mathrm{u}}_k = (\hat{u}_{0|k}, \ldots, \hat{u}_{N-1|k})$ and define $\hat{\mathrm{x}}_k = (\hat{x}_{0|k}, \ldots, \hat{x}_{N|k})$ as the corresponding state trajectory, where $\hat{x}_{0|k} = x_k$. The warm-started primal decision variable at timestep k is generated by

$$\tilde{\mathrm{u}}_k = (\hat{u}_{1|k-1}, \ldots, \hat{u}_{N-1|k-1}, \bar{u}_v - K(\hat{x}_{N|k-1} - \bar{x}_v)), \tag{14}$$

where K is the LQR gain. Note that $\tilde{\mathrm{u}}_k \in \mathcal{F}(x_k, v_{k-1})$ since $\hat{\mathrm{u}}_{k-1} \in \mathcal{F}(x_{k-1}, v_{k-1})$ by definition of the terminal set $\mathcal{T} = \mathcal{O}_\infty$ [6]. Thus, if $v_k = v_{k-1}$, then the corresponding slack variable satisfies $\tilde{s}_k := M\tilde{\mathrm{u}}_k + L\theta_k + \ell \geq 0$, where $\theta_k = (x_k, v_k)$. The log-domain variable used to initialize Algorithm 1 is then generated by

$$\tilde{\gamma}_k = -\log\left(\max\left\{\frac{\tilde{s}_k}{\sqrt{\eta_{k-1}}}, \epsilon_s \mathbf{1}\right\}\right), \tag{15}$$

where $\epsilon_s > 0$ is a small tolerance and $\max\{x, y\} := (\max(x_1, y_1), ..., \max(x_n, y_n)) \in \mathbb{R}^n$ for $x, y \in \mathbb{R}^n$. The first argument in the $\max$ operator is a rearrangement of the parameterization in (9), whereas the second argument is included to ensure that (15) is defined when $\tilde{s}_k$ has elements that are equal to zero.

Once initialized with $\tilde{\gamma}_k$, Algorithm 1 begins by computing $\eta^*(\tilde{\gamma}_k)$ in (13). That is, it finds the smallest η that ensures that the warm-start $\tilde{\gamma}_k$ is within a neighborhood of the corresponding central-path point. In this sense, the warm-started solution only needs to be sufficiently close to *some* point on the central-path.

IV. A Computational Governor for the Log-domain Interior-point Method

Consider the closed-loop system (7) with an auxiliary reference sequence $\{v_k\}$ and a target reference command r. As discussed in Section II, one could attempt to complete the control task by setting $v_k \equiv r$, but this poses two potential problems. First, if $(x_k, r) \notin \Gamma_N$, then the OCP will be infeasible. Second, if the commanded reference r is changed (see Remark 2), then warm-starting may be ineffective since the solutions to $\mathcal{P}_N(x_k, v_k)$ and $\mathcal{P}_N(x_{k-1}, v_{k-1})$ may be substantially different. The computational governor (CG) discussed in this section addresses both of these issues. Specifically, the CG selects the auxiliary reference v_k such that the (parameterized) distance from r is minimized subject to constraints on the feasibility and suboptimality of the warm-start in (15).

Before describing the CG, we first establish closed-loop stability for a constant reference signal $v_k \equiv v$ in the case where MPC solutions are computed using Algorithm 1 with a non-zero final homotopy parameter η.

Theorem 4: Let Assumptions 1-3 hold. Let $\hat{\mathrm{u}} : \Gamma_N \to \mathcal{F}(x, v)$ be a function that generates a feasible solution to $\mathcal{P}_N(x, v)$ satisfying

$$J(x, v, \hat{\mathrm{u}}(x, v)) \leq V(x, v) + \beta \|x - \bar{x}_v\|_Q^2, \tag{16}$$

where J and V are defined in (6) and $\beta \in (0, 1)$ is a constant. Consider the closed-loop dynamics

$$x_{k+1} = Ax_k + B\Xi\hat{\mathrm{u}}(x_k, v),$$

starting from an initial condition x_0 with a constant reference v. Then for all $(x_0, v) \in \Gamma_N$, all solutions satisfy:

- $(x_k, v) \in \Gamma_N$ for all $k \in \mathbb{N}$;
- $y_k \in \mathcal{Y}$ for all $k \in \mathbb{N}$;
- $\lim_{k\to\infty} x_k = \bar{x}_v$.

If, in addition, $v \in \operatorname{Int} \mathcal{V}$ then $\bar{x}_v$ is asymptotically stable.

Proof: The result is a direct consequence of [25, Theorem 1] and [7, Theorem 13.1]. ∎

Corollary 1: Consider the output $(\hat{\mathrm{u}}_k, \gamma_k, \eta_k)$ of Algorithm 1 applied to $\mathcal{P}_N(x_k, v)$ and suppose that η_k satisfies

$$m\eta_k \leq \beta \|x_k - \bar{x}_v\|_Q^2, \tag{17}$$

where $m \in \mathbb{N}$ is the number of constraints in (5). Then, $\hat{\mathrm{u}}_k$ satisfies (16), i.e. $J(x_k, v, \hat{\mathrm{u}}_k) \leq V(x_k, v) + \beta \|x_k - \bar{x}_v\|_Q^2$.

Proof: The solution $\hat{\mathrm{u}}_k$ generated by Algorithm 1 is feasible and satisfies $J(x_k, v, \hat{\mathrm{u}}_k) - V(x_k, v) \leq m\eta_k$ by Theorem 3. Then, (16) is satisfied by the bound in (17). ∎

Assumption 5: At all timesteps k, the tolerance η_{final} in Algorithm 1 satisfies $m\eta_{\text{final}} \leq \beta \|x_k - \bar{x}_v\|_Q^2$.

We will now describe the strategy that the CG uses to select v_k at each timestep. First, we define the following parameterization of the reference command v_k:

$$v_k = v_{k-1} + \kappa_k(r - v_{k-1}), \tag{18}$$

where r is the target reference and $\kappa_k \in [0, 1]$ is a time-varying parameter that dictates the rate at which v_k converges to r. Note that $\kappa_k = 1$ implies $v_k = r$ and that $\kappa_k = 0$ implies $v_k = v_{k-1}$. The CG developed in this section chooses κ_k at each timestep subject to restrictions on the suboptimality and feasibility of the warm-start. This is accomplished by selecting κ_k using the following optimization problem:

$$\max_{\eta, \kappa_k} \quad \kappa_k \tag{19a}$$
$$\text{s.t.} \quad \|d(\tilde{\gamma}_k, \eta, \kappa_k)\|_\infty \leq 1, \tag{19b}$$
$$\eta \in [\eta_{\min}, \eta_{\max}], \tag{19c}$$
$$\kappa_k \in [0, 1], \tag{19d}$$

where $d(\tilde{\gamma}_k, \eta, \kappa_k)$ denotes the Newton step of the QP defined by $\mathcal{P}_N(x_k, v_k)$, where $v_k = v_{k-1} + \kappa_k(r - v_{k-1})$, at the log-domain variable $\tilde{\gamma}_k \in \mathbb{R}^m$ in (15) and a homotopy parameter $\eta > 0$. We denote the solution of (19) given $\tilde{\gamma}_k$ as (η_k^*, κ_k^*).

Remark 3: Throughout the rest of this paper, the notation $d(\gamma, \eta, \kappa)$ is used to refer to the Newton step of the LDIPM applied to $\mathcal{P}_N(x, v + \kappa(r - v))$ for some state-reference pair (x, v), log-domain variable γ, and homotopy parameter η. The dependence on (x, v) is suppressed, but made clear through context throughout the paper. Note that the notation $d(\gamma, \eta)$ used in Section III is equivalent to $d(\gamma, \eta, 0)$.

Recall from Lemma 1 that a point $(\hat{\mathrm{u}}, s, \lambda)$ is primal-dual feasible when generated by a Newton step satisfying $\|d(\gamma, \eta, \kappa)\|_\infty \leq 1$. Moreover, the resulting iterate has a suboptimality bounded by η and will reduce a measure of convergence by a factor of $\frac{1}{2}$. In other words, the CG looks for the largest parameterized reference step κ_k^* towards the target that ensures the first iterate of the warm-started LDIPM is primal-dual feasible, has suboptimality bounded by $\eta_k^* \leq \eta_{\max}$, and is contained in a region of fast convergence. Thus, if $\eta_{\max}$ is chosen to be relatively small, then feasible solutions to (19) correspond to reference steps κ_k that ensure $\mathcal{P}_N(x_k, v_k)$ is feasible and can be solved using relatively few iterations of the LDIPM.

So, we propose the following computationally governed MPC procedure:

1) *Warm-start*: Compute $\tilde{\gamma}_k$ according to (15),
2) *Computational governor*: Compute (η_k^*, κ_k^*) using (19) when feasible, otherwise define $(\eta_k^*, \kappa_k^*) = (\eta_{\text{const}}, 0)$ where $\eta_{\text{const}} > 0$ is a constant,
3) *Update Reference:* Set v_k using (18) with $\kappa_k = \kappa_k^*$,
4) *LDIPM*: Solve $\mathcal{P}_N(x_k, v_k)$ using Algorithm 1 with inputs $x = x_k$, $v = v_k$, $\gamma_{\text{init}} = \tilde{\gamma}_k$, $\eta_{\text{init}} = \eta_k^*$, and some η_{final} satisfying Assumptions 5 and 6.

Remark 4: This procedure can also be interpreted as a time-distributed homotopy method. The CG increments v_k towards r under the restriction that $\mathcal{P}_N(x_k, v_k)$ can be solved efficiently using the solution to $\mathcal{P}_N(x_{k-1}, v_{k-1})$. In this sense, v_k is a time-distributed homotopy parameter since it is incremented once per timestep. In contrast, the LDIPM homotopy parameter η is incremented several times per timestep to solve $\mathcal{P}_N(x_k, v_k)$.

The procedure used to formulate and solve the optimization problem in (19) will be discussed in Section VI. The following section analyzes the closed-loop properties of (7) when the reference v_k is chosen using the CG.

V. Analysis of the Computationally Governed Closed-loop System

Consider the closed-loop system

$$v_0 = \bar{v}_0 + \kappa_0(r - \bar{v}_0), \tag{20a}$$
$$v_k = v_{k-1} + \kappa_k(r - v_{k-1}), \quad \forall k \in \mathbb{N}_{[1,\infty)}, \tag{20b}$$
$$x_{k+1} = Ax_k + B\Xi\hat{\mathrm{u}}_k(x_k, v_k), \quad \forall k \in \mathbb{N}, \tag{20c}$$

where $\hat{\mathrm{u}}_k(x_k, v_k)$ is the output of the LDIPM applied to $\mathcal{P}_N(x_k, v_k)$ and $\bar{v}_0$ is a reference initialization. This section is devoted to analyzing the dynamics of (20) when κ_k is chosen using the CG. In particular, we show that under reasonable assumptions (20) has the following characteristics when κ_k is chosen using the CG:

- recursive feasibility of $\mathcal{P}_N(x_k, v_k)$, i.e. $(x_0, \bar{v}_0) \in \Gamma_N$ implies that $(x_k, v_k) \in \Gamma_N$ for all $k \in \mathbb{N}$,
- asymptotic stability, i.e. $(\bar{x}_r, r)$ is a Lyapunov stable equilibrium and $\exists \mathcal{R} \subseteq \Gamma_N$ such that $\forall (x_0, v_0) \in \mathcal{R}$ $\lim_{k\to\infty}(x_k, v_k) = (\bar{x}_r, r)$,
- finite-time convergence of the reference, i.e. $\exists k^* \in \mathbb{N}$ such that $v_k = r$ for all $k \geq k^*$.

To begin, we establish conditions under which the CG advances the reference v_k towards r. To this end, the following lemma provides conditions under which (19) is strictly feasible and the optimal solution is bounded away from zero. The proof of Lemma 2 is found in Appendix I. The following assumption is stated first to simply the statement of Lemma 2.

Assumption 6: Let $\bar{\eta} > 0$, $\bar{\kappa} \in (0, 1]$, and $\bar{\epsilon}_s > 0$ be the constants defined in Lemma 2. The constants η_{final} in Algorithm 1, $\eta_{\min}$ and $\eta_{\max}$ in (19), and ϵ_s in (15) are chosen so that $\eta_{\text{final}} \leq \bar{\eta}$, $(\eta_{\min}, \eta_{\max}) \ni \bar{\eta}$, and $\epsilon_s \leq \bar{\epsilon}_s$.

Lemma 2: Let Assumptions 1-5 hold and consider the closed-loop system in (20). There exist constants $\bar{\eta} > 0$, $\bar{\kappa} \in (0, 1]$, $\bar{\epsilon}_s > 0$, and $\delta_V > 0$ such that if Assumption 6 holds, $v_{k-1} \in \mathcal{V}_\epsilon$, and $V(x_{k-1}, v_{k-1}) \leq \delta_V$, then the CG optimization problem (19) is strictly feasible and the optimal solution satisfies $\kappa_k^* \geq \bar{\kappa}$.

In other words, if the algorithmic parameters η_{final}, $\eta_{\min}$, and ϵ_s are chosen to be sufficiently small, then a non-zero reference step $\kappa_k^* \geq \bar{\kappa}$ is selected by (19) whenever x_{k-1} is sufficiently close to $G_x v_{k-1}$. The result in Lemma 2 constitutes most of what is needed to analyze the closed-loop stability of (20). The following lemma provides the last result required to prove Theorem 5.

Lemma 3: Let Assumptions 1-5 hold and consider the closed-loop system

$$x_{k+1} = Ax_k + B\Xi\hat{\mathrm{u}}_k(x_k, v_k),$$

where $\hat{\mathrm{u}}_k(x_k, v_k)$ is the output of the LDIPM applied to $\mathcal{P}_N(x_k, v_k)$ and $\{v_k\}$ is any reference sequence satisfying $v_k \in \mathcal{V}_\epsilon$ and $(x_k, v_k) \in \Gamma_N$ for all $k \in \mathbb{N}$. Then, the error signal $x_k - G_x v_k$ is input-to-state stable (ISS) [32] with respect to the reference change $\Delta v_k = v_{k+1} - v_k$, i.e. there exists $\rho \in \mathcal{KL}$ and $\zeta \in \mathcal{K}$ such that

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

where the classes of $\mathcal{K}$ and $\mathcal{KL}$ functions follow the usual definition in [32].

Proof: ISS of the closed-loop system under the optimal MPC feedback policy was proven in [27, Lemma 5]). In contrast, the control sequence $\hat{\mathrm{u}}(x_k, v_k)$ in Lemma 3 corresponds to an LDIPM output with non-zero suboptimality proportional to η_k. However, Lemma 3 can easily be proven using the same steps as [27, Lemma 5]) by noting that

$$V(x_{k+1}, v_k) - V(x_k, v_k) \leq -(1 - \beta)\|x_k - G_x v_k\|_Q^2,$$

for all $(x_k, v_k) \in \Gamma_N$ according to Theorem 4 and Assumption 5. Then, since $f(\cdot) = (1 - \beta)\|\cdot\|_Q^2$ is a class $\mathcal{K}$ function, the rest of the proof follows directly from [27, Lemma 5]). ■

The results of Lemma 2 and Lemma 3 can then be combined to analyze the closed-loop system (20) when κ_k is selected using the CG. The following theorem, which is the main result of this paper, demonstrates that the MPC problem $\mathcal{P}_N(x_k, v_k)$ is recursively feasible, $(\bar{x}_r, r)$ is an asymptotically stable equilibrium point, and the size of the closed-loop ROA of (20) is expanded relative to a standard MPC implementation.

Theorem 5: Let Assumptions 1-6 hold and consider the closed-loop system (20) with $\bar{v}_0 \in \mathcal{V}_\epsilon$ and $(x_0, \bar{v}_0) \in \Gamma_N$. Let κ_k be the optimal value of (19) when feasible, otherwise let $\kappa_k = 0$. Then, $\lim_{k\to\infty}(x_k, v_k) = (\bar{x}_r, r)$ and $\forall k \in \mathbb{N}$, $(x_k, v_k) \in \Gamma_N$ and $y_k \in \mathcal{Y}$. Moreover, $(\bar{x}_r, r)$ is an asymptotically stable equilibrium point of (20).

Proof: To begin, we prove that $(x_k, v_k) \in \Gamma_N$ for all $k \in \mathbb{N}$. First, consider timestep $k = 0$ as its treatment differs slightly from the later timesteps. If $\kappa_0 = 0$, then $(x_0, v_0) = (x_0, \bar{v}_0) \in \Gamma_N$ by assumption. Whereas if $\kappa_0 > 0$, then for some $\gamma \in \mathbb{R}^m$ and $\eta > 0$ the Newton step satisfies $\|d(\gamma, \eta, \kappa_0)\|_\infty \leq 1$ since κ_0 is the optimal value of (19). Thus, there exists a feasible solution to $\mathcal{P}_N(x_0, v_0)$ by Lemma 1, and so $(x_0, v_0) \in \Gamma_N$. Next, we prove by induction that $(x_k, v_k) \in \Gamma_N$ for all $k \in \mathbb{N}$. Let $k \in \mathbb{N}$ and assume $(x_k, v_k) \in \Gamma_N$. If $\kappa_{k+1} = 0$, then $(x_{k+1}, v_{k+1}) = (x_{k+1}, v_k) \in \Gamma_N$ by Theorem 4. Whereas if $\kappa_{k+1} > 0$, then $(x_{k+1}, v_{k+1}) \in \Gamma_N$ by the same argument as the case where $\kappa_0 > 0$. So, $(x_k, v_k) \in \Gamma_N$ for all $k \in \mathbb{N}$. Additionally, note that $y_k \in \mathcal{Y}$ is implied by $(x_k, v_k) \in \Gamma_N$.

Next, we prove that $\lim_{k\to\infty}(x_k, v_k) = (\bar{x}_r, r)$. Consider the non-trivial case where $\bar{v}_0 \neq r$ and let $e_k = \|r - v_k\|$. Note that by definition $e_k = (1 - \kappa_k)e_{k-1}$ and so e_k converges since it is monotone nonincreasing and lower bounded. So, define $e \geq 0$ such that $\lim_{k\to\infty} e_k = e$. For every $k \in \mathbb{N}$, a parameter[3] $\omega_k \in [0, 1]$ can be defined such that $v_k = \bar{v}_0 + \omega_k(r - \bar{v}_0)$. Define $\omega := 1 - \frac{e}{\|r - \bar{v}_0\|} \geq 0$, then

$$\lim_{k\to\infty} v_k = \bar{v}_0 + \omega(r - \bar{v}_0) =: v.$$

In other words, since v_k is constrained to follow a line between $\bar{v}_0$ and r, then it must converge to the unique point on this line where $\|v_k - r\| = e$.

To prove that $v = r$, we first use Lemma 3 to state that

$$\lim_{k\to\infty} \|x_k - G_x v_k\| = 0, \tag{21}$$

[3]The parameter ω_k is different (but related) to the parameter κ_k in (18). More specifically, ω_k is a function of $(\kappa_0, ..., \kappa_k)$.

since v_k converges and $\rho \in \mathcal{KL}$. Thus,

$$\lim_{k\to\infty} x_k = \lim_{k\to\infty} G_x v_k = G_x v = \bar{x}_v. \tag{22}$$

Next, let $\delta_V > 0$ and $\bar{\kappa} > 0$ be defined as per Lemma 2. Since $\exists \sigma \in \mathcal{K}$ such that $V(x,v) \leq \sigma(\|x - G_x v\|)$ [25], then $\exists k' \in \mathbb{N}$ such that $\forall k \geq k'$, $V(x_k, v) \leq \delta_V$ by (22). Thus, $\kappa_k \geq \bar{\kappa}$ for all $k \geq k'$ according to Lemma 2.

Assume, for the sake of contradiction, that $v \neq r$ (and equivalently $e \neq 0$). Note that $\lim_{k\to\infty} \|v_k - v_{k-1}\| = 0$ since v_k converges. However, by definition of v_k:

$$\lim_{k\to\infty} \|v_k - v_{k-1}\| \geq \liminf_{k\to\infty} \|\kappa_k(r - v_{k-1})\| \geq \bar{\kappa} e > 0,$$

where the second inequality follows from monotonicity of e_k and the fact that $\kappa_k \geq \bar{\kappa}$ for all $k > k'$. Therefore, we have arrived at a contradiction. Thus, we must have that $\lim_{k\to\infty} v_k = r$ since we know that v_k converges. Moreover, $\lim_{k\to\infty} x_k = \bar{x}_r$ by (22).

Last, we show that $(\bar{x}_r, r)$ is Lyapunov stable. Let $\|(x_0, v_0) - (\bar{x}_r, r)\| \leq \delta_\theta$. Then $\forall k \in \mathbb{N}$, $\|v_k - r\| \leq \|v_0 - r\| \leq \delta_\theta$ by definition of v_k. Next, we note that

$$\begin{aligned}\|x_0 - G_x v_0\| &\leq \|x_0 - \bar{x}_r\| + \|\bar{x}_r - G_x v_0\| \\ &\leq \delta_\theta + \|G_x\|\|r - v_0\| \leq (1 + \|G_x\|)\delta_\theta,\end{aligned}$$

and $\|\Delta v_k\| = \kappa_k \|r - v_{k-1}\| \leq \delta_\theta$ for all $k \in \mathbb{N}$. Thus, by the triangle inequality and Lemma 3 we have that

$$\begin{aligned}\|x_k - \bar{x}_r\|_Q &\leq \|x_k - G_x v_k\|_Q + \|G_x(v_k - r)\|_Q \\ &\leq \rho(p_1\delta_\theta, k) + \zeta(\delta_\theta) + p_2\delta_\theta,\end{aligned}$$

for some constants $p_1, p_2 > 0$, where the second inequality follows by $\rho \in \mathcal{KL}$ and $\zeta \in \mathcal{K}$. Thus, it is possible to bound $\|(x_k, v_k) - (\bar{x}_r, r)\|$ arbitrarily small given δ_θ sufficiently small; hence $(\bar{x}_r, r)$ is a Lyapunov stable equilibrium. Moreover, $(\bar{x}_r, r)$ is an asymptotically stable equilibrium since $\lim_{k\to\infty}(x_k, v_k) = (\bar{x}_r, r)$. ∎

Remark 5: Theorem 5 does not require that $(x_0, r) \in \Gamma_N$. In fact, there are no requirements on the target reference aside from Assumption 4. Moreover, the ROA of the closed-loop system (20) is expanded to

$$\Upsilon_N = \{x \in \mathbb{R}^n \mid \exists v \in \mathcal{V}_\epsilon \text{ s.t. } (x,v) \in \Gamma_N\} = \bigcup_{v\in\mathcal{V}_\epsilon} \Gamma_N(v).$$

Thus, the size of the ROA of (20) is expanded relative to $\Gamma_N(r)$ — the ROA of the closed-loop system when the feedback law in (7) is implemented with $v_k \equiv r$. As discussed in the introduction, the expansion of the closed-loop ROA allows for the use of a shorter prediction horizon N, which reduces the computational cost of solving $\mathcal{P}_N(x_k, v_k)$.

Remark 6: The core principle of the CG differs from the feasibility governor (FG) of [27] and [28] due to the additional restriction imposed on the warm-start suboptimality in the CG. To understand the difference between the two approaches, suppose a (potentially infeasible) set-point command r is supplied to FG/CG. The FG will return the closest possible reference command that maintains feasibility of the OCP, whereas the CG will return a reference command that maintains feasibility of the OCP *and* ensures that the OCP is well-initialized (i.e. $\|d(\gamma, \eta, \kappa)\|_\infty \leq 1$).

Finally, the following theorem demonstrates that the reference signal v_k converges to the target reference r in *finite-time*. The proof of this theorem is given in Appendix II.

Theorem 6: Let Assumptions 1-6 hold and consider the closed-loop system (20) described in Theorem 5. There exist constants $k^* \in \mathbb{N}$, $\bar{\eta}' > 0$, and $\bar{\epsilon}'_s > 0$ such that if $k > k^*$, $\eta_k \leq \bar{\eta}'$, and $\epsilon_s \leq \bar{\epsilon}'_s$, then $v_k = r$.

Remark 7: The strategy described in this paper could also be applied to problems with an arbitrarily time-varying target reference sequence $\{r_k\}$ with no changes to the approach. In this scenario, the CG will return auxiliary references v_k that ensure $\mathcal{P}_N(x_k, v_k)$ remains feasible by the same arguments as in the proof of Theorem 5. Of course, there is no guarantee that v_k will converge to r_k for an arbitrary sequence $\{r_k\}$. However if, roughly speaking, the changes in r_k are slower than the speed at which v_k converges, then it may be possible for the closed-loop system to track the target reference sequence. Alternatively, if the reference sequence $\{r_k\}$ is such that $r_k = r$ for all $k > k'$ for some r and $k' \in \mathbb{N}$, then it can easily be shown that $\lim(x_k, v_k) = (\bar{x}_r, r)$ using Theorem 5 and that v_k will converge to r in finite-time using Theorem 6. In other words, the CG possesses all of the conventional convergence properties of a standard reference governor [33].

VI. Implementation of the Computational Governor

In this section, the computational procedure used to implement the CG will be addressed. To facilitate this development, an affine parameterization of the Newton step $d(\gamma, \eta, \kappa)$ is defined in the following proposition.

Proposition 1: Consider the condensed OCP (5) for $\mathcal{P}_N(x, v')$, where $v' = v + \kappa(r - v)$ for some $\kappa \in [0, 1]$ and $v \in \mathcal{V}_\epsilon$. Suppose $(x, v') \in \Gamma_N$ and let $d = d(\gamma, \eta, \kappa)$ represent the LDIPM Newton step for $\mathcal{P}_N(x, v')$ at a given $\gamma \in \mathbb{R}^m$ and $\eta > 0$. Then, there exist $d_0(\gamma), d_1(\gamma), d_2(\gamma) \in \mathbb{R}^m$ such that

$$d(\gamma, \eta, \kappa) = d_0(\gamma) + d_1(\gamma)\frac{1}{\sqrt{\eta}} + d_2(\gamma)\frac{1}{\sqrt{\eta}}\kappa. \tag{23}$$

Proof: Equation (11) can be rewritten as

$$d = \mathbf{1} - \frac{1}{\sqrt{\eta}} e^\gamma \odot (M\hat{\mathrm{u}} + L_x x + L_v v' + \ell),$$

$$\begin{aligned}(M^T\mathcal{D}(\gamma)M + H)\frac{1}{\sqrt{\eta}}\hat{\mathrm{u}} &= 2M^T e^\gamma - \frac{1}{\sqrt{\eta}}(W_x x + W_v v') \\ &\quad - \frac{1}{\sqrt{\eta}} M^T\mathcal{D}(\gamma)(L_x x + L_v v' + \ell),\end{aligned}$$

where $L = [L_x\ L_v]$ and $W = [W_x\ W_v]$. Then by considering the parameterization $v' = v + \kappa(r - v)$ and observing that $\frac{1}{\sqrt{\eta}}\hat{\mathrm{u}}$ is affine with respect to $(\frac{1}{\sqrt{\eta}}, \frac{1}{\sqrt{\eta}}\kappa)$ and d is affine with respect to $(\frac{1}{\sqrt{\eta}}\hat{\mathrm{u}}, \frac{1}{\sqrt{\eta}}, \frac{1}{\sqrt{\eta}}\kappa)$, it follows that d is affine with respect to $(\frac{1}{\sqrt{\eta}}, \frac{1}{\sqrt{\eta}}\kappa)$. ∎

An efficient procedure for computing $d_0(\gamma)$, $d_1(\gamma)$, and $d_2(\gamma)$ is described in Appendix III.

The observation in Proposition 1 allows for the optimization problem in (19) to be written as a 2-dimensional linear

program (LP) in the variable $\xi = [\kappa \ \sqrt{\eta}]^T$. Thus, the CG is implemented by solving the following 2-dimensional LP:

$$\max_{\xi \in \mathbb{R}^2} \ c_\xi^T \xi \tag{24a}$$
$$\text{s.t.} \ \ \Pi(\tilde{\gamma}_k)\xi \leq \pi(\tilde{\gamma}_k) \tag{24b}$$
$$\xi_{\min} \leq \xi \leq \xi_{\max}, \tag{24c}$$

where $c_\xi = [1 \ c_\eta]^T$, $\xi_{\min} = [0 \ \sqrt{\eta_{\min}}]^T$, $\xi_{\max} = [1 \ \sqrt{\eta_{\max}}]^T$, and

$$\Pi(\gamma) = \begin{bmatrix} d_0(\gamma) - (1-\epsilon_d)\mathbf{1} & d_2(\gamma) \\ -d_0(\gamma) - (1-\epsilon_d)\mathbf{1} & -d_2(\gamma) \end{bmatrix},$$
$$\pi(\gamma) = \begin{bmatrix} -d_1(\gamma) \\ d_1(\gamma) \end{bmatrix}.$$

Note that (24) is equivalent to (19) with the exception of two minor practical modifications. First, the constraint (24b) is equivalent to $\|d(\tilde{\gamma}_k, \eta, \kappa)\|_\infty \leq (1-\epsilon_d)$, where $0 < \epsilon_d \ll 1$ is a small tolerance. Second, the cost in (24a) contains a weighting parameter $c_\eta \geq 0$ on $\sqrt{\eta}$. This parameter allows one to trade-off the size of the reference step (as measured by κ) with the suboptimality of the warm-start (as measured by η). If $\epsilon_d = c_\eta = 0$, then (24) is equivalent to (19) and the largest reference step κ_k satisfying the constraints (19b)-(19d) is selected. The use of a non-zero ϵ_d and c_η is not necessary to solve (24), but these additions have been found to improve the efficacy of the CG in experiments.

Seidel's algorithm [34] is employed to solve the resulting LP. The application of this algorithm to the 2-variable LP in (24) is shown in Algorithm 2. Seidel's algorithm is adopted here since it is an efficient method designed for solving low-dimensional LPs [34]. The expected execution time of Algorithm 2 is $O(m)$, where m is the number of constraints in (5). This expected execution time is derived by noting that line 6 only executes when the current solution estimate violates the randomly sampled constraint in line 4 [34]. Moreover, line 6 can be solved by explicitly eliminating one of the decision variables using the equality constraint $\Pi_i^T \xi = \pi_i$ and by iterating over the m' constraints defined by (Π', π'), where $\pi' \in \mathbb{R}^{m'}$ and $m' < m$ for all $j < m$.

Algorithm 2 Seidel($c_\xi, \Pi, \pi, \xi_{\min}, \xi_{\max}$)

1: $\xi \leftarrow$ Project c_ξ onto the box defined by $\xi_{\min}$ and $\xi_{\max}$
2: Define (Π', π') such that $\Pi'\xi \leq \pi' \iff \xi_{\min} \leq \xi \leq \xi_{\max}$
3: **for** $i = 1 : \text{length}(\pi)$ **do**
4: $\quad (\Pi_i, \pi_i) \leftarrow$ Random unchosen constraint in (Π, π)
5: $\quad$ **if** $\Pi_i^T \xi > \pi_i$ **then**
6: $\quad\quad \xi \leftarrow \arg\max c_\xi^T \xi$ s.t. $\Pi'\xi \leq \pi'$ and $\Pi_i^T \xi = \pi_i$
7: $\quad$ **end if**
8: $\quad (\Pi', \pi') \leftarrow$ Update to include (Π_i, π_i)
9: **end for**
10: **return** ξ

Remark 8: The computational governor proposed in this paper is derived by exploiting properties of the LDIPM. It may be possible to develop computational governors for other optimization algorithms by replacing the condition $\|d(\tilde{\gamma}_k, \eta, \kappa_k)\|_\infty \leq 1$ with a similar condition. For example, the path-following algorithms described in [35, Chapter 5] ensure that primal-dual iterates are contained in a neighborhood denoted as $\mathcal{N}_q(\theta)$ where $\theta \in [0, 1)$ is a fixed parameter related to the size of the neighborhood. One could attempt to derive a computational governor that ensures a primal-dual warm-start is contained in $\mathcal{N}_q(\theta)$ with a bounded duality gap. However, it is not immediately clear whether this would lead to an optimization problem that is as tractable as (24). Further investigation of primal-dual computational governors and more general computational governors is left to future work.

VII. Examples

This section demonstrates the application of the computationally governed MPC strategy to the lateral dynamics of a passenger vehicle moving forward at a constant speed of $V_x = 30$ m/s. The model is taken directly from the example in [27], which is based on the model in [36] and roughly represents a 2017 BMW 740i sedan.

The system states are $x = (s, \psi, \beta, \omega)$, where s is the lateral position of the vehicle, ψ is the yaw angle, β is the sideslip angle, and ω is the yaw rate. The control input is the steering angle $u = \delta$ and the tracking output is the lateral position $z = s$ (i.e. $E = [1 \ 0 \ 0 \ 0]$ and $F = 0$). The continuous-time state-space matrices (A_c, B_c) are

$$\left(\begin{bmatrix} 0 & V_x & V_x & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\frac{2C_\alpha}{mV_x} & \frac{C_\alpha(\ell_r - \ell_f)}{mV_x^2} - 1 \\ 0 & 0 & \frac{C_\alpha(\ell_r - \ell_f)}{I_{zz}} & -\frac{C_\alpha(\ell_r^2 + \ell_f^2)}{I_{zz}V_x} \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ \frac{C_\alpha}{mV_x} \\ \frac{C_\alpha \ell_f}{I_{zz}} \end{bmatrix} \right),$$

where $m = 2041$ kg is the vehicle mass, $I_{zz} = 4964$ kg$\cdot$m^2 is the yaw moment of inertia, $\ell_f = 1.56$ m and $\ell_r = 1.64$ m are the moment arms of the front and rear wheels relative to the center of mass, and $C_\alpha = 246994$ N/rad is the tire stiffness. The system is discretized using a sampling time of $T = 0.01$.

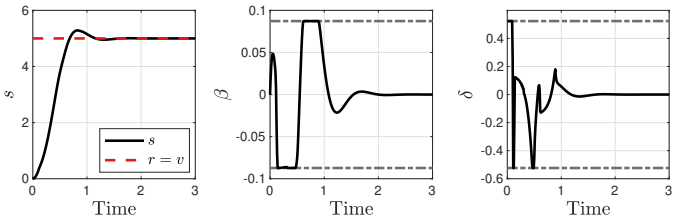
The following case studies compare the performance of a standard MPC (S-MPC) implementation, i.e. executing the feedback law in (7) with $v_k \equiv r$, with the computationally governed MPC (CG-MPC) implementation described in the previous sections. In each case, the MPC weight matrices are specified as $Q = \text{diag}(1, 0.1, 0.1, 0.1)$ and $R = 0.1$ and the resulting OCPs are solved using the LDIPM in Algorithm 1 with a final tolerance of $\eta_{\text{final}} = 10^{-8}$. The CG is implemented by solving (24) with parameters $c_\eta = 1$, $\eta_{\min} = 10^{-10}$, $\eta_{\max} = 10^{-2}$, and $\epsilon_d = 10^{-2}$.

Case 1: Sideslip angle and steering angle constraints

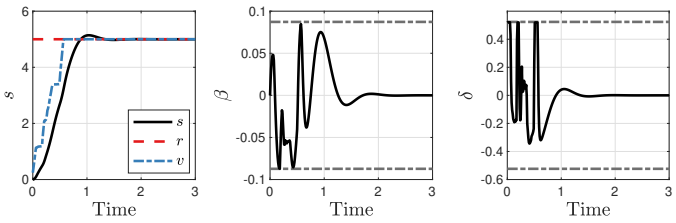
The first case corresponds to a simple maneuver where constraints are imposed on the sideslip and the steering angles. The constrained output is $y = (\beta, \delta)$, such that

$$C = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad D = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

The constraint set is $\mathcal{Y} = \frac{\pi}{180^\circ}[-5^\circ, 5^\circ] \times \frac{\pi}{180^\circ}[-30^\circ, 30^\circ]$. The first constraint represents a conservative limit on the sideslip angle to prevent tire slip, while the second constraint represents the mechanical limits of the steering angle. The



(a) Standard MPC (S-MPC)



(b) Computationally governed MPC (CG-MPC)

Fig. 1. Case 1: Closed-loop trajectories of each MPC implementation. The grey dotted lines represent the constraint limits of each output.

system is commanded to track a constant set-point command of $r = 5$ starting from an equilibrium of $x_0 = 0$. The S-MPC implementation uses a horizon length of $N = 48$, which corresponds to the minimum $N \in \mathbb{N}$ satisfying $x_0 \in \Gamma_N(r)$. The CG-MPC implementation uses a smaller horizon length of $N = 15$ to demonstrate the expansion of the ROA provided by the CG. In both cases, the solution of $\mathcal{P}_N(x_0, 0)$ is used to warm-start[4] the LDIPM at $k = 0$ (i.e. we assume that the system was being stabilized at the origin for $k < 0$).

Figure 1 shows the closed-loop trajectories that arise when each MPC implementation is used to accomplish the control task. In the CG-MPC implementation, the auxillary reference v_k is incremented towards $r = 5$ over a window of 0.57 seconds. As a result, the system operates slightly more conservatively and the outputs β and δ do not dwell on their constraint boundaries as much as in the S-MPC implementation.

Figure 2 shows a comparison of the performance of each MPC implementation. In the S-MPC case, the warm-start procedure described in Section III-B is mostly ineffective in the transient phase of the maneuver and many iterations are required for the LDIPM to terminate. In contrast, the CG-MPC implementation only requires a maximum of 6 LDIPM iterations. The second plot in Figure 2 shows the initial homotopy parameter η_k^* chosen by the CG is less than 10^{-5} at each timestep. Thus, the LDIPM is initialized so that only a few iterations are needed to obtain a primal-dual feasible solution with $\eta \leq \eta_{\text{final}} = 10^{-8}$. Moreover, individual iterations of the LDIPM are also much cheaper in the CG-MPC implementation due to the use of the smaller prediction horizon of $N = 15$. As previously discussed, this combination of ensuring good warm-starting of the OCP and reducing the horizon length N is the key contribution of the CG.

Last, we note that the CG does not introduce a dramatic degradation in closed-loop performance and the settling time

[4]The provided warm-start is only used in the S-MPC implementation if $\eta^*(\tilde{\gamma}_k) < \infty$ is found. Otherwise, the LDIPM is initialized with $\eta_{\text{init}} = 10^8$ and $\gamma_{\text{init}} = 0$ to avoid providing the LDIPM with a very poor warm-start.

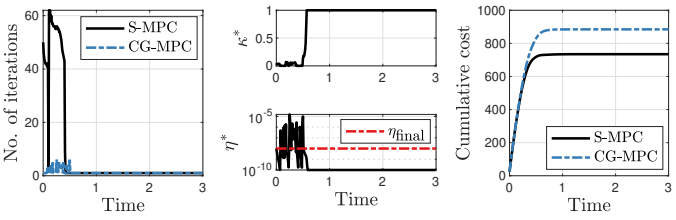


Fig. 2. Case 1: The CG-MPC implementation only requires a maximum of 6 iterations for the LDIPM to terminate since the CG initializes the LDIPM so that the initial homotopy parameter $\boldsymbol{\eta_k^*}$ is close to $\boldsymbol{\eta}_{\text{final}}$. However, the cumulative cost of the maneuver is increased by approximately 20% as a consequence of altering the auxiliary reference $\boldsymbol{v_k}$.

of s. To quantify the closed-loop performance of each implementation, the cumulative cost defined by

$$\text{Cumulative cost at } k := \sum_{i=0}^{i=k} \|x_i - G_x r\|_Q^2 + \|u_i - G_x r\|_R^2,$$

is shown in the third plot of Figure 2 for both MPC implementations. The CG-MPC implementation results in a total cumulative cost that is 20% greater than the S-MPC implementation due to the alteration of the auxiliary reference v_k.

Case 2: Front and rear slip angle constraints

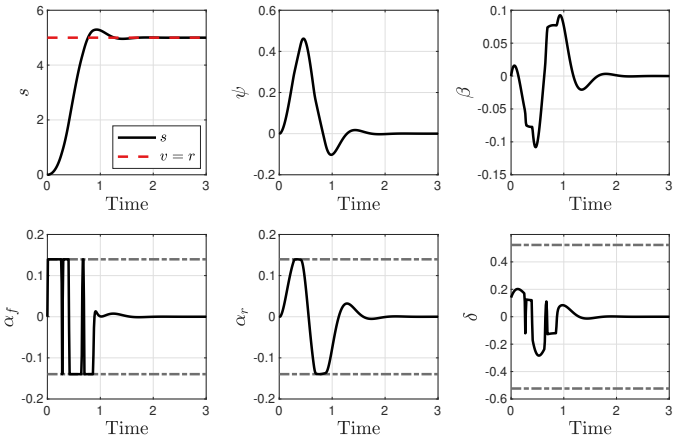
In the second case, the constrained output is $y = (\alpha_f, \alpha_r, \delta)$, where α_f and α_r are the front and rear slip angles of the vehicle, and

$$C = \begin{bmatrix} 0 & 0 & -1 & -\frac{\ell_f}{V_x} \\ 0 & 0 & -1 & \frac{\ell_r}{V_x} \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad D = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}.$$

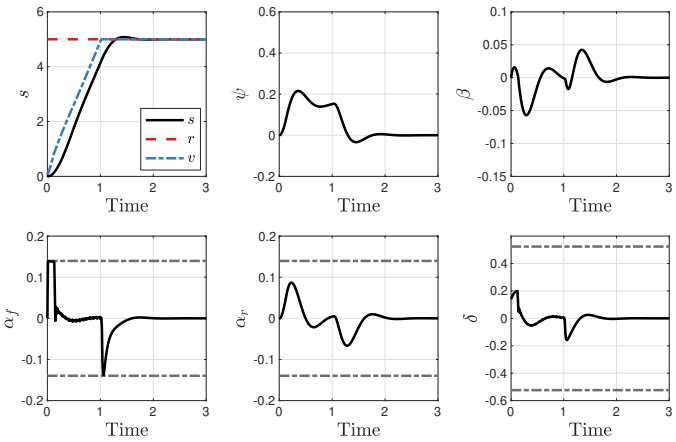
The constraint set is $\mathcal{Y} = \frac{\pi}{180^\circ}[-8^\circ, 8^\circ] \times \frac{\pi}{180^\circ}[-8^\circ, 8^\circ] \times \frac{\pi}{180^\circ}[-30^\circ, 30^\circ]$. The first two constraints are angular limits that prevent tire slip, while the third constraint represents the mechanical limits of δ. These constraints are slightly more complex than in Case 1 and constitute a more principled approach of preventing tire slip. The control task is the same as in Case 1. The S-MPC implementation uses a horizon length of $N = 66$, which corresponds to the minimum $N \in \mathbb{N}$ to enforce $x_0 \in \Gamma_N(r)$ for the new set of constraints. The CG-MPC implementation uses $N = 15$ as in Case 1.

Figures 3 and 4 show comparisons of the closed-loop trajectories and performance when the new constraints are imposed on the maneuver. The end result is similar to Case 1, but the CG-MPC implementation is forced to operate slightly more conservatively in Case 2. For example, the CG-MPC implementation does not operate near the constraint boundary of α_r and it operates near the constraint boundary of α_f for less time than in the S-MPC implementation. Moreover, the auxiliary reference v_k takes longer to converge (1.02 seconds as opposed to 0.57 seconds in Case 1) and the increase in the cumulative cost of the CG-MPC implementation relative to the S-MPC implementation is higher in Case 2 (a 30% increase as opposed to a 20% increase in Case 1).

Nevertheless, the CG-MPC implementation still achieves its intended purpose and a maximum of 5 iterations are required for the LDIPM to terminate. However, these computational



(a) Standard MPC (S-MPC)



(b) Computationally governed MPC (CG-MPC)

Fig. 3. Case 2: Closed-loop trajectories of each MPC implementation. The grey dotted lines represent the constraint limits of each output.

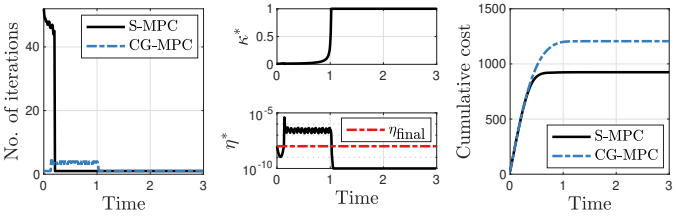


Fig. 4. Case 2: The computational comparison of both MPC implementations remains similar to Case 1, however there is a larger (30%) increase in the cumulative cost of the CG-MPC implementation.

savings come at a slightly higher cost in closed-loop performance in this case. The decrease in closed-loop performance of the CG-MPC implementation in Case 2 relative to Case 1 is likely a consequence of the increased complexity of the constraints.

Case 3: Varying initial conditions

The final case study compares the performance of both MPC implementations when applied to solve the control task in Case 2 for several initial conditions. The simulations shown in Case 2 are repeated for 10 initial conditions of the form $x_0 = (s_0, 0, 0, 0)$, where $s_0 \in \{-5, -4, ..., 4\}$. For each initial condition, the S-MPC implementation uses the minimum

TABLE I

THE S-MPC HORIZON LENGTH N FOR EACH INITIAL CONDITION s_0 IN CASE 3.

s_0	-5	-4	-3	-2	-1	0	1	2	3	4
N	101	95	89	82	74	66	55	44	30	16

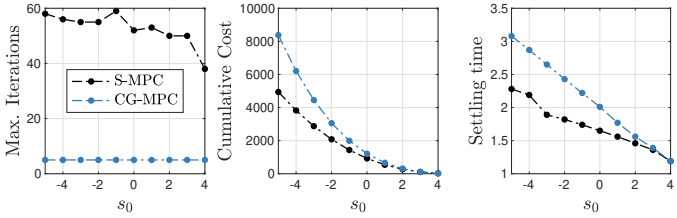


Fig. 5. Case 3: Performance metrics for both MPC implementation when applied to simulations with varying initial conditions $\boldsymbol{x_0} = (\boldsymbol{s_0}, \boldsymbol{0}, \boldsymbol{0}, \boldsymbol{0})$. The settling time is defined as the time $\boldsymbol{t_k} = \boldsymbol{Tk}$ at which $\|\boldsymbol{G_x x_{k-1}} - \boldsymbol{r}\| > \boldsymbol{0.01}$ and $\forall \boldsymbol{k'} \geq \boldsymbol{k}\ \|\boldsymbol{G_x x_{k'}} - \boldsymbol{r}\| \leq \boldsymbol{0.01}$.

horizon length $N \in \mathbb{N}$ necessary to enforce $x_0 \in \Gamma_N(r)$ (see Table I). Meanwhile, the CG-MPC implementation uses $N = 15$ for all simulations. The solution to $\mathcal{P}_N(x_0, s_0)$ is used for warm-starting at $k = 0$.

Figure 5 compares the performance of each MPC implementation over the varied initial conditions. As one might expect, the S-MPC strategy requires more iterations when the initial position s_0 is varied away from the reference of $r = 5$. In contrast, the CG-MPC implementation never requires more than 5 iterations for termination of the LDIPM in any of the simulations. As seen in the previous cases, the CG-MPC implementation reduces the worst-case number of LDIPM iterations at the expense of increasing in the total cumulative cost and settling time of the maneuver. However, the difference in settling time between the two methods is less than 1 second even in the most extreme case of $s_0 = -5$.

Figure 6 compares the worst-case execution times of both MPC implementations. The S-MPC procedure was implemented using a variety of quadratic programming algorithms (including the LDIPM in Algorithm 1) to give additional points of comparison. The following algorithms were used:

- QP-AS and QP-IP: The active set and interior-point methods implemented in MATLAB's `quadprog` function.
- ADMM: The alternating direction method of multipliers in [37, Algorithm 8].
- PDIP: The primal-dual IPM in [38, Algorithm 16.4].
- QPKWIK: The dual active set method in [39].
- GPAD: The dual gradient-projection algorithm in [11].
- FBRS: The Fischer–Burmeister method in [40].
- FBStab: The semismooth algorithm in [12].

Remark 9: Each algorithm (except for QP-IP) was implemented using native MATLAB functions and compiled into mex functions using MATLAB's `codegen` toolbox. The QP-IP algorithm was implemented using a standard MATLAB function since `quadprog`'s interior-point option does not support code generation. The experiments were executed using MATLAB 2022b on a 2018 MacBook Pro with a 2.2 GHz 6-Core Intel Core i7 processor and 16 GB RAM. The tolerance of each algorithm's respective truncation criteria was set to $m \cdot \eta_{\text{final}}$, where m is the number of constraints and $\eta_{\text{final}} = 10^{-8}$,

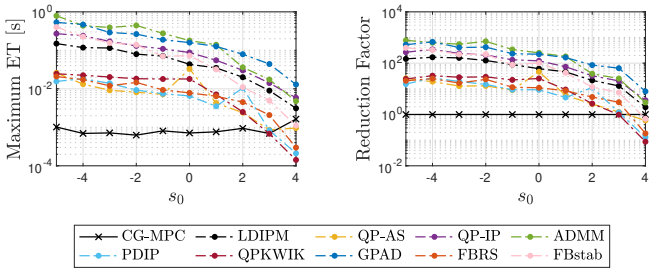


Fig. 6. Case 3: The maximum execution times (ET) for the simulations in Figure 5. Each legend entry corresponding to a QP algorithm (i.e. LDIPM through FBstab) corresponds to an S-MPC implementation where $\mathcal{P}_N(\boldsymbol{x_k}, \boldsymbol{r})$ is solved using the respective QP algorithm. The right-side plot shows the ETs in the left-side plot normalized by the maximum ET of the CG-MPC implementation for each $\boldsymbol{s_0}$ (i.e. a y-axis value of 10 means that the maximum ET of the S-MPC implementation was 10 times longer than the CG-MPC implementation).

which corresponds to the cost suboptimality upon truncation of the LDIPM (Theorem 3). Each simulation was averaged as necessary to obtain consistent timing.

Remark 10: The reported execution times of the S-MPC implementations correspond to the maximum amount of time required to solve $\mathcal{P}_N(x_k, r)$ for any $k \geq 0$. The reported execution times of the CG-MPC implementation correspond to the maximum time required to solve both the CG optimization problem in (24) and the resulting MPC problem.

Remark 11: We emphasize that Figure 6 is not meant to constitute a conclusive comparison of these algorithms. The relative performance of each algorithm will depend strongly on the given problem. For instance, the QP-IP algorithm exhibits comparable performance to the PDIP, QPKWIK, and FBRS algorithms in the "Asteroid" example shown in [40, Table 2]. In contrast, the QP-IP algorithm is much slower in the example shown in Figure 6. A variety of algorithms are shown in Figure 6 to give a rough upper and lower bound on the performance of a reasonable S-MPC implementation.

The CG-MPC implementation results in a significantly lower worst-case computational cost than all of the S-MPC implementations for all of the simulations with $s_0 \leq 2$. In the most extreme example of $s_0 = -5$, the CG-MPC implementation is 15 to 777 times faster than the S-MPC implementations. In a less extreme example of $s_0 = 0$, the CG-MPC implementation is 9 to 251 times faster. Note that the LDIPM-based S-MPC implementation exhibits mediocre performance relative to the rest of the S-MPC implementations. Thus, the CG-MPC implementation may exhibit even higher performance when applied to an example where the LDIPM performs particularly well.

VIII. Conclusions

This paper introduced a supervisory scheme for computationally efficient implementation of Linear Quadratic MPC for reference tracking with state and control constraints. The key component of this scheme is a computational governor that maintains feasibility and bounds the suboptimality of a warm-starting strategy by adjusting the reference command supplied to the MPC problem. Theoretical guarantees regarding the recursive feasibility of the MPC problem, asymptotic stability of the target equilibrium, and finite-time convergence of the supervised reference signal were provided. In a numerical experiment, the computational governor was shown to reduce the worst-case execution time of a standard MPC implementation for control of lateral vehicle dynamics by over a factor of 10.

Appendix I
Proof of Lemma 2

Appendix I-A derives properties of the LDIPM and the closed-loop system (20) that are required to prove Lemma 2. Appendix I-B uses these results to complete the proof.

A. Preliminary results

This subsection proceeds as follows: First, Lemma 4 and Corollary 2 establish properties of the Newton step d when constraints are inactive at optimality. Second, Propositions 2 and 3 describe characteristics of the set $\mathcal{O}_\infty$ — the relevance being that the constraints of $\mathcal{P}_N(x, v)$ are inactive at optimality when $x \in \text{Int}\, \mathcal{O}_\infty(v)$. Third, Lemma 5 and Corollary 3 derive bounds on the suboptimality of the warm-start $\tilde{\gamma}_k$ in (15) when $x_{k-1} \in \text{Int}\, \mathcal{O}_\infty(v_{k-1})$ and $v_{k-1} = v_k$.

Lemma 4: Consider the QP in (5) for a fixed parameter $\theta = (x, v) \in \Gamma_N$ and define $c = W\theta$, $b = L\theta + \ell$, $p = Nn_u$, and $m \in \mathbb{N}$ as the number of rows of $M \in \mathbb{R}^{m \times p}$. Let $\hat{\mathrm{u}}^* \in \mathbb{R}^p$ be the optimal solution of (5). Suppose that the optimal slack $s^* = M\hat{\mathrm{u}}^* + b$ satisfies $s^* > 0$ and let $\eta > 0$, $\gamma \in \mathbb{R}^m$, $s = \sqrt{\eta}e^{-\gamma}$, and $q = s^* \odot s^{-1}$. Then, the LDIPM Newton step is

$$d(\gamma, \eta) = \mathbf{1} - q - 2\Psi(\gamma)^T \Phi(\gamma)^{-1} \Psi(\gamma)(\mathbf{1} - \frac{1}{2}q),$$

where $\Phi(\gamma) = H + M^T \mathcal{D}(\gamma) M$, $\Psi(\gamma) = M^T \text{diag}(e^\gamma)$, and $\mathcal{D}$ is defined in Theorem 2.

Proof: Let $\hat{\mathrm{u}} = \hat{\mathrm{u}}(\gamma, \eta)$ represent the primal variable update and define $\Delta\hat{\mathrm{u}} = \hat{\mathrm{u}} - \hat{\mathrm{u}}^*$. Rewrite (11a) as

$$\begin{aligned} d &= \mathbf{1} - \frac{1}{\sqrt{\eta}} e^\gamma \odot (M\hat{\mathrm{u}}^* + M\Delta\hat{\mathrm{u}} + b) \\ &= \mathbf{1} - q - \frac{1}{\sqrt{\eta}} e^\gamma \odot (M\Delta\hat{\mathrm{u}}). \end{aligned} \tag{25}$$

Observing that

$$H\hat{\mathrm{u}}^* = -c, \qquad \mathcal{D}(\gamma)s^* = e^\gamma \odot (e^\gamma \odot s^*) = \sqrt{\eta} e^\gamma \odot q,$$

and writing (11b) in terms of $\Delta\hat{\mathrm{u}}$ and $\Phi(\gamma)$ yields

$$\begin{aligned} \Phi(\gamma)\Delta\hat{\mathrm{u}} &= 2\sqrt{\eta} M^T e^\gamma - (c + M^T \mathcal{D}(\gamma) b) \\ &\qquad - (M^T \mathcal{D}(\gamma) M + H)\hat{\mathrm{u}}^* \\ &= 2\sqrt{\eta} M^T e^\gamma - M^T \mathcal{D}(\gamma) s^* \\ &= 2\sqrt{\eta}\Psi(\gamma)(\mathbf{1} - \frac{1}{2}q). \end{aligned}$$

where $\Psi(\gamma) = M^T \text{diag}(e^\gamma)$. Substituting $\Delta\hat{\mathrm{u}}$ into (25) proves the claim. ∎

For a given $s > 0$, there are infinitely many pairs (γ, η) satisfying $s = \sqrt{\eta}e^{-\gamma}$. The next corollary shows that a pair that makes $\|d(\gamma, \eta)\|_\infty$ arbitrary close to $\|\mathbf{1} - q\|_\infty$ can be chosen if $s^* > 0$. Hence, if we know an s that is close to s^*,

we can pick (γ, η) satisfying $\|d(\gamma,\eta)\|_\infty < 1$, ensuring good initialization of Algorithm 1.

Corollary 2: Let $\hat{\mathrm{u}}^* \in \mathbb{R}^p$ be the optimal solution to (5) for some $\theta \in \Gamma_N$ and define c, b, p, m as in Lemma 4. Suppose that $s^* = M\hat{\mathrm{u}}^* + b$ satisfies $s^* > 0$. Fix $s \in \mathbb{R}^m_{>0}$ and let $q = s^* \odot s^{-1}$. Finally, define $\gamma(\alpha) := -(\log s + \alpha \mathbf{1})$ and $\eta(\alpha) = e^{-2\alpha}$ for a parameter $\alpha \in \mathbb{R}$. Then $s = \sqrt{\eta(\alpha)} e^{-\gamma(\alpha)}$ for all α, and

$$\lim_{\alpha\to\infty} \|d(\gamma(\alpha), \eta(\alpha))\|_\infty \leq \|\mathbf{1} - q\|_\infty.$$

Proof: That $s = \sqrt{\eta(\alpha)} e^{-\gamma(\alpha)}$ is trivial. Letting $g(\gamma) = \|2e^\gamma \odot M\Phi(\gamma)^{-1} M^T e^\gamma (\mathbf{1} - \frac{1}{2}q)\|_\infty$ we have by the triangle inequality and Lemma 4 that

$$\|d(\gamma,\eta)\|_\infty \leq \|\mathbf{1} - q\|_\infty + g(\gamma).$$

We will show that $\lim_{\alpha\to\infty} g(\gamma) = 0$. First, note that

$$\begin{aligned} g(\gamma) &\leq 2\|e^\gamma\|^2 \|M\|^2 \|(H + M^T\mathcal{D}(\gamma)M)^{-1}\| \|\mathbf{1} - \frac{1}{2}q\| \\ &\leq 2k_1 e^{-2\alpha} \|(H + M^T\mathcal{D}(\gamma)M)^{-1}\|, \end{aligned}$$

where k_1 is a constant depending on $\log s$, M, and q. Further,

$$\lim_{\alpha\to\infty} \|(H + M^T\mathcal{D}(\gamma)M)^{-1}\| = \|H^{-1}\|.$$

Hence, for some k_2, we have $\lim_{\alpha\to\infty} g(\gamma) \leq \lim_{\alpha\to\infty} k_2 e^{-2\alpha} = 0$, proving the claim. ∎

The previous lemma and corollary give additional insight into the behavior of the LDIPM when constraints of (5) are inactive at optimality. Next, we derive some properties of the set $\mathcal{O}_\infty$, since the constraints of $\mathcal{P}_N(x, v)$ are inactive when $x \in \text{Int } \mathcal{O}_\infty(v)$.

Proposition 2: Let Assumptions 1-3 hold. Then, $\exists \bar{\delta} > 0$ such that $\mathcal{B}(\bar{x}_v, \bar{\delta}) \subset \mathcal{O}_\infty(v)$ for all $v \in \mathcal{V}_\epsilon$, where $\bar{x}_v = G_x v$.

Proof: Consider system (1) under LQR feedback with a constant reference $v \in \mathcal{V}_\epsilon$:

$$\tilde{x}_{k+1} = \tilde{A}\tilde{x}_k, \quad \tilde{y}_k = \tilde{C}\tilde{x}_k,$$

where $\tilde{x}_k = x_k - \bar{x}_v$, $\tilde{y}_k = y_k - \bar{y}_v$, $\tilde{A} = A - BK$, and $\tilde{C} = C - DK$. Further, consider the tightened constraint set in the modified coordinate system $(1-\epsilon)\tilde{\mathcal{Y}} = \{y - \bar{y}_v \mid y \in (1-\epsilon)\mathcal{Y}\}$. Note that $v \in \mathcal{V}_\epsilon \iff \bar{y}_v \in (1-\epsilon)\mathcal{Y} \iff 0 \in (1-\epsilon)\tilde{\mathcal{Y}}$. As noted in [30, Theorem 2.1], there exists $\delta_1 > 0$ such that $\forall \tilde{x} \in \mathbb{R}^n$ $\|\tilde{C}\tilde{A}^k\tilde{x}\| \leq \delta_1\|\tilde{x}\|$ for all $k \in \mathbb{N}$ since $\tilde{A}$ is stable. Next we define $\delta_2 = \text{dist}(\mathcal{Y}, (1-\epsilon)\mathcal{Y}) > 0$, where $\text{dist}(A, B)$ is the minimum 2-norm distance between sets A and B defined in the usual manner. Then, $\mathcal{B}(\bar{y}_v, \delta_2) \subset \mathcal{Y}$. Thus, if $\delta_1\|\tilde{x}\| \leq \delta_2$ this implies that $\tilde{C}\tilde{A}^k\tilde{x} \in \tilde{Y}$ for all $k \in \mathbb{N}$. Hence, $\mathcal{B}(\bar{x}_v, \delta_2/\delta_1) \subset \mathcal{O}_\infty(v)$ and the statement is proven for $\bar{\delta} = \delta_2/\delta_1$. ∎

Proposition 3: Let Assumptions 1-3 hold and define constants δ and δ' such that $0 < \delta < \delta' < \bar{\delta}$, where $\bar{\delta}$ is defined in Proposition 2. Let $v \in \mathcal{V}_\epsilon$, $\kappa \in [0, 1]$, and $v' = v + \kappa(r - v)$. Then,

1) $\mathcal{B}(\bar{x}_v, \delta) \subset \text{Int } \mathcal{O}_\infty(v)$ and $\mathcal{B}(\bar{x}_{v'}, \delta') \subset \text{Int } \mathcal{O}_\infty(v')$,
2) $\exists \bar{\kappa} \in (0, 1]$ such that $\mathcal{B}(\bar{x}_v, \delta) \subset \mathcal{B}(\bar{x}_{v'}, \delta')$ if $\kappa \leq \bar{\kappa}$,

where $\bar{x}_v = G_x v$ and $\bar{x}_{v'} = G_x v' = G_x[v + \kappa(r - v)]$.

Proof: Let δ and δ' be such that $0 < \delta < \delta' < \bar{\delta}$. Thus, $\mathcal{B}(\bar{x}_v, \delta) \subset \text{Int } \mathcal{O}_\infty(v)$ and $\mathcal{B}(\bar{x}_{v'}, \delta') \subset \text{Int } \mathcal{O}_\infty(v')$ by Proposition 2. To prove the second statement in Proposition 3, let $x \in \mathcal{B}(\bar{x}_v, \delta)$ and $\kappa \leq \bar{\kappa} = \frac{\delta' - \delta}{c_r\|G_x\|}$, where $c_r = \sup\{\|r - v\| \mid v \in \mathcal{V}_\epsilon\}$. Next, we use the triangle inequality and the definition of v' to write

$$\begin{aligned} \|x - \bar{x}_{v'}\| &\leq \|x - \bar{x}_v\| + \|G_x\kappa(r - v)\|, \\ &\leq \delta + \kappa c_r\|G_x\| \leq \delta'. \end{aligned}$$

So, $x \in \mathcal{B}(\bar{x}_{v'}, \delta')$ and thus $\mathcal{B}(\bar{x}_v, \delta) \subset \mathcal{B}(\bar{x}_{v'}, \delta')$. ∎

Proposition 2 states that the volume of the maximal constraint admissible set $\mathcal{O}_\infty(v)$ has a uniform lower-bound for every $v \in \mathcal{V}_\epsilon$. Proposition 3 states that the interiors of $\mathcal{O}_\infty(v)$ and $\mathcal{O}_\infty(v + \kappa(v - r))$ have an intersection with non-zero volume for every $v \in \mathcal{V}_\epsilon$ if κ is smaller than a constant $\bar{\kappa}$.

Next, we use these results to characterize the suboptimality of the warm-start $\tilde{\mathrm{u}}_k$ when the state is in the interior of $\mathcal{O}_\infty(v_k)$. Note that the value function $V(\cdot, v)$ in (6) is a Lyapunov function for the closed-loop system (20c) for a static reference v by Theorem 4 and Corollary 1. Moreover, sublevel sets of $V(\cdot, v)$ are forward invariant. Define the set-valued map $\Omega : \mathcal{V}_\epsilon \times \mathbb{R}_{>0} \rightrightarrows \mathbb{R}^n$ such that $\Omega(v, a)$ is the largest sublevel set of $V(\cdot, v)$ contained in $\mathcal{B}(\bar{x}_v, a)$. Note that $\forall (v, a) \in \mathcal{V}_\epsilon \times (0, \bar{\delta})$ $\Omega(v, a)$ has non-zero volume and $\Omega(v, a) \subset \text{Int } \mathcal{O}_\infty(v)$ by Proposition 2.

The following lemma and corollary demonstrate that if the state x_k is sufficiently close to the equilibrium $\bar{x}_v$, then the primal suboptimality of the warm-start in (14) is bounded by the homotopy parameter η_{k-1} of Algorithm 1 when the reference is held constant.

Lemma 5: Let Assumptions 1-5 hold. Suppose $v \in \mathcal{V}_\epsilon$ and $x_{k-1} \in \Omega(v, \bar{\delta})$ where $\bar{\delta}$ is defined in Proposition 2, and let $\hat{\mathrm{u}}_{k-1}$ be the output of the LDIPM applied to $\mathcal{P}_N(x_{k-1}, v)$. Then, the warm-start $\tilde{\mathrm{u}}_k$ in (14) satisfies

$$\|\hat{\mathrm{u}}^*(x_k, v) - \tilde{\mathrm{u}}_k\|^2 \leq c_1\eta_{k-1} + c_2\sqrt{\eta_{k-1}},$$

for $x_k = Ax_{k-1} + B\Xi\hat{\mathrm{u}}_{k-1}$, where $\hat{\mathrm{u}}^*(x, v)$ is the optimal solution of $\mathcal{P}_N(x, v)$ and $c_1, c_2 > 0$ are constants only dependent on the problem data of the QP in (5).

Proof: By Theorem 3, $\hat{\mathrm{u}}_{k-1}$ satisfies

$$J(x_{k-1}, v, \hat{\mathrm{u}}_{k-1}) - V(x_{k-1}, v) \leq m\eta_{k-1}.$$

Note that for all $\hat{\mathrm{u}} \in \mathcal{F}(x, v)$ and $(x, v) \in \Gamma_N$

$$\frac{\lambda_{\min}(H)}{2}\|\hat{\mathrm{u}}^*(x, v) - \hat{\mathrm{u}}\|^2 \leq J(x, v, \hat{\mathrm{u}}) - V(x, v), \tag{26}$$

by strong convexity of J and optimality of $\hat{\mathrm{u}}^*$. Combining these two inequalities yields

$$\|\hat{\mathrm{u}}^*(x_{k-1}, v) - \hat{\mathrm{u}}_{k-1}\|^2 \leq 2\lambda^{-1}_{\min}(H)m\eta_{k-1}.$$

The restriction $x_{k-1} \in \Omega(v, \bar{\delta}) \subset \mathcal{O}_\infty(v)$ implies that $\hat{\mathrm{u}}^*(x_{k-1}, v)$ is a sequence of LQR control inputs due to the specification of the terminal penalty in (4) [1, Section 2.5.3]. Moreover, the following state also satisfies $x_k \in \Omega(v, \bar{\delta})$ since sublevel sets of $V(\cdot, v)$ are invariant under Assumption 5. So, the optimal cost at x_{k-1} and x_k can be evaluated using the cost-to-go of the LQR feedback law [1, Section 2.5.3]. Then, by using standard procedures in MPC stability analysis (e.g., [7, Equation 12.19]), one can show that

$$J(x_k, v, \tilde{\mathrm{u}}_k) = J(x_{k-1}, v, \hat{\mathrm{u}}_{k-1}) - l(x_{k-1}, u_{k-1}, v),$$

where $l(x,u,v) = \|x-\bar{x}_v\|_Q^2 + \|u-\bar{u}_v\|_R^2$ is the stage cost. Define $e_{k-1} = J(x_{k-1}, v, \hat{\mathfrak{u}}_{k-1}) - V(x_{k-1}, v) \leq m\eta_{k-1}$ and use fact that $V(x_{k-1}, v) = \|x_{k-1}-\bar{x}_v\|_P^2$ to write

$$J(x_k, v, \tilde{\mathfrak{u}}_k) = \|x_{k-1}-\bar{x}_v\|_P^2 + e_{k-1} - l(x_{k-1}, u_{k-1}, v).$$

Then, let $\sigma_{k-1} = u_{k-1} - (\bar{u}_v - K(x_{k-1}-\bar{x}_v))$ be the error in the control input u_{k-1}. Substituting this into the previous equation gives

$$J(x_k, v, \tilde{\mathfrak{u}}_k) \leq \|x_{k-1}-\bar{x}_v\|_{P-Q-K^TRK} + \|\sigma_{k-1}\|_R^2 + e_{k-1}.$$

Next, we again make use of the LQR cost-to-go to write

$$V(x_k, v) = \|Ax_{k-1} + B(\bar{u}_v - K(x_{k-1}-\bar{x}_v) + \sigma_{k-1}) - \bar{x}_v\|_P^2.$$

We now combine these two equations and use the identity

$$\|(A-BK)x\|_P^2 = \|x\|_{P-Q-K^TRK}^2,$$

to write

$$\begin{aligned} &J(x_k, v, \tilde{\mathfrak{u}}_k) - V(x_k, v) \\ &\quad \leq \|\sigma_{k-1}\|_{R+B^TPB}^2 + e_{k-1} - 2(x_{k-1}-\bar{x}_v)^T PB\sigma_{k-1} \\ &\quad \leq \|\sigma_{k-1}\|_{R+B^TPB}^2 + e_{k-1} + a\|x_{k-1}-\bar{x}_v\|\|\sigma_{k-1}\|, \end{aligned}$$

where $a > 0$ is a constant. Note that $\|x_{k-1}-\bar{x}_v\| \leq \bar{\delta}$ since $x_{k-1} \in \Omega(v_{k-1}, \bar{\delta})$, $e_{k-1} \leq m\eta_{k-1}$, and $\|\sigma_{k-1}\|^2 \leq 2\lambda_{\min}(H)^{-1} m\eta_{k-1}$. Then, $\exists c_1', c_2' > 0$ such that

$$J(x_{k-1}, v, \tilde{\mathfrak{u}}_k) - V(x_k, v) \leq c_1'\eta_{k-1} + c_2'\sqrt{\eta_{k-1}}.$$

Then, the proof is completed by using (26). ∎

Corollary 3: Under the same conditions as Lemma 5, the warm-started slack variable $\tilde{s}_k$ in (15) satisfies

$$\|s^*(x_k, v) - \tilde{s}_k\|^2 \leq c_3\eta_{k-1} + c_4\sqrt{\eta_{k-1}},$$

where $x_k = Ax_{k-1} + B\Xi\hat{\mathfrak{u}}_{k-1}$, $c_3, c_4 > 0$ are constants only dependent on the problem data of (5), $s^*(x_k, v)$ is the optimal slack variable at $\theta_k = (x_k, v)$, and $\tilde{s}_k = M\tilde{\mathfrak{u}}_k + L\theta_k + \ell$.

Proof: The proof is completed by simply noting that

$$\|s^*(x_k, v) - \tilde{s}_k\|^2 \leq \|M\|^2\|\hat{\mathfrak{u}}^*(x_k, v) - \tilde{\mathfrak{u}}_k\|^2,$$

and using Lemma 5 to bound $\|\hat{\mathfrak{u}}^*(x_k, v) - \tilde{\mathfrak{u}}_k\|^2$. ∎

B. Proof of the Lemma

The sketch of the proof is as follows. We define the constants δ_V and $\bar{\kappa}$ so that $x_{k-1} \in \text{Int}\, \mathcal{O}_\infty(v_{k-1})$ is enforced by the assumption $V(x_{k-1}, v_{k-1})$, and $x_k \in \text{Int}\, \mathcal{O}_\infty(v_k)$ for all $\kappa_k \leq \bar{\kappa}$. Lemma 4 is then used to analyze the Newton step $d(\tilde{\gamma}_k, \eta_{k-1}, \kappa_k)$ since the optimal slack of $\mathcal{P}_N(x_k, v_k)$ is bounded away from zero since $x_k \in \text{Int}\, \mathcal{O}_\infty(v_k)$. The proof is completed by using the results in Appendix I-A to show that the constants $\bar{\eta}$ and $\bar{\epsilon}_s$ can be selected sufficiently small so that $\|d(\tilde{\gamma}_k, \eta_{k-1}, \kappa_k)\|_\infty < 1$ for all $\kappa_k \leq \bar{\kappa}$ and $\eta_{k-1} \leq \bar{\eta}$. Thus, (19) is strictly feasible and the solution must satisfy $\kappa_k^* \geq \bar{\kappa}$.

To begin, we specify the constants $\delta_V > 0$ and $\bar{\kappa} > 0$ that appear in the statement of the lemma. First, define a constant $\delta \in (0, \bar{\delta})$, where $\bar{\delta}$ is defined in Proposition 2. Then, let δ_V be defined so that $x_{k-1} \in \Omega(v_{k-1}, \delta)$ is satisfied by the assumption that $V(x_{k-1}, v_{k-1}) \leq \delta_V$, where the set-valued map Ω is defined in Appendix I-A. Moreover, note that $\Omega(v_{k-1}, \delta) \subset \text{Int}\, \mathcal{O}_\infty(v_{k-1})$ by Proposition 2, and that the following state satisfies $x_k \in \Omega(v_{k-1}, \delta)$ by invariance of sublevel sets of $V(\cdot, v_{k-1})$. Next, define a constant $\delta' \in (\delta, \bar{\delta})$. Then by Proposition 3, $\exists \bar{\kappa}' > 0$ (proportional to $\delta' - \delta$) such that $\kappa_k \leq \bar{\kappa}'$ implies that $x_k \in \mathcal{B}(\bar{x}_{v_k}, \delta') \subset \text{Int}\, \mathcal{O}_\infty(v_k)$. Thus, let $\bar{\kappa} \leq \bar{\kappa}'$ and note that $\kappa_k \leq \bar{\kappa}$ by assumption. Moreover, note that δ_V, δ, δ', and $\bar{\kappa}$ are all independent of the state, reference, and timestep.

Next, we show that the norm of the warm-started constraint slack $\tilde{s}_k$ is lower bounded by a constant. Note that the norm of the optimal slack $s^*(x_k, v_{k-1})$ is lower bounded since $x_k \in \Omega(v_{k-1}, \delta) \subset \text{Int}\, \mathcal{O}_\infty(v_{k-1})$. So, define $c_s > 0$ such that $\|s^*(x, v)\|_\infty \geq c_s$ for all $(x, v) \in \Omega(v, \delta) \times \mathcal{V}_\epsilon$. Then,

$$\|\tilde{s}_k\|_\infty \geq c_s - \left(c_3\eta_{k-1} - c_4\sqrt{\eta_{k-1}}\right)^{\frac{1}{2}},$$

by Corollary 3. Thus for any $\bar{c}_s \in (0, c_s)$, $\exists \bar{\eta}_1 > 0$ such that $\eta_{k-1} \leq \bar{\eta}_1$ implies that $\|\tilde{s}_k\|_\infty \geq \bar{c}_s$. So henceforth, fix $\bar{c}_s \in (0, c_s)$, define $\bar{\eta}_1$ accordingly, and let $\bar{\eta} \leq \bar{\eta}_1$ such that $\eta_{k-1} \leq \bar{\eta}_1$.

Lemma 4 can then be used to analyze the Newton step $d(\tilde{\gamma}_k, \eta_{k-1}, \kappa_k)$ of $\mathcal{P}_N(x_k, v_k)$, since $x_k \in \text{Int}\, \mathcal{O}_\infty(v_k)$ implies that $s^*(x_k, v_k) > 0$. So, we use Lemma 4 with $s^* = s^*(x_k, v_k) > 0$ and $s = \tilde{s}_k > 0$ to write

$$\begin{aligned} d(\tilde{\gamma}_k, \eta_{k-1}, \kappa_k) &= \mathbf{1} - q(\kappa_k) \\ &\quad - 2\Psi(\tilde{\gamma}_k)^T\Phi(\tilde{\gamma}_k)^{-1}\Psi(\tilde{\gamma}_k)(\mathbf{1} - \frac{1}{2}q(\kappa_k)), \end{aligned}$$

where $\Phi(\gamma) = H + M^T\mathcal{D}(\gamma)M$, $\Psi(\gamma) = M^T\text{diag}(e^\gamma)$, and $q(\kappa_k) = \tilde{s}_k^{-1} \odot s^*(x_k, v_k)$. Thus,

$$\|d(\tilde{\gamma}_k, \eta_{k-1}, \kappa_k)\|_\infty \leq \|\mathbf{1} - q(\kappa_k)\|_\infty + g(\tilde{\gamma}_k), \tag{27}$$

where

$$g(\tilde{\gamma}_k) = 2\|e^{\tilde{\gamma}_k}\|^2\|M\|^2\|\Phi(\tilde{\gamma}_k)^{-1}\|\|1 - \frac{1}{2}q(\kappa_k)\|. \tag{28}$$

We proceed by bounding both terms in (27). Consider the following bound for the first term:

$$\begin{aligned} \|\mathbf{1} - q(\kappa_k)\|_\infty &= \|\tilde{s}_k^{-1} \odot (\tilde{s}_k - s^*(x_k, v_k))\|_\infty \\ &\leq \|\tilde{s}_k^{-1}\|_\infty(\|\tilde{s}_k - s^*(x_k, v_{k-1})\|_\infty \\ &\quad + \|s^*(x_k, v_k) - s^*(x_k, v_{k-1})\|_\infty). \end{aligned}$$

The factor $\|\tilde{s}_k^{-1}\|_\infty$ can be upper bounded using the previously established lower bound on $\|\tilde{s}_k\|_\infty$. Meanwhile, the first term in the bracket can be bounded directly by Corollary 3. To bound the second term, note that for any $v \in \mathcal{V}_\epsilon$ and $x \in \mathcal{O}_\infty(v)$, the optimal control sequence is a sequence of LQR inputs and can thus be written as

$$\hat{\mathfrak{u}}^*(x, v) = Sx + Tv,$$

where S and T are matrices dependant on A, B, K, N, G_x, and G_u. Then,

$$\begin{aligned} &\|s^*(x_k, v_k) - s^*(x_k, v_{k-1})\|_\infty \\ &\quad = \|M(\hat{\mathfrak{u}}^*(x_k, v_k) - \hat{\mathfrak{u}}^*(x_k, v_{k-1})) + L_v(v_k - v_{k-1})\|_\infty \\ &\quad \leq \|M\|\|T(v_k - v_{k-1})\| + \|L_v\|\|v_k - v_{k-1}\| \\ &\quad \leq (\|M\|\|T\| + \|L_v\|)\|r - v_{k-1}\|\kappa \\ &\quad \leq c_m c_r \kappa \end{aligned}$$

where $c_m = \|M\|\|T\| + \|L_v\|$ and c_r is defined in Proposition 3. So,

$$\|\mathbf{1} - q(\kappa_k)\|_\infty \leq \frac{1}{\bar{c}_s}\left[(c_3\eta_{k-1} + c_4\sqrt{\eta_{k-1}})^{\frac{1}{2}} + c_m c_r \kappa\right],$$

and thus for any constant $c_q' > 0$, $\bar{\eta}_2' \in (0, \bar{\eta}_1)$ such that $\eta_{k-1} \leq \bar{\eta}_2'$ implies

$$\|\mathbf{1} - q(\kappa_k)\|_\infty \leq \frac{c_m c_r}{\bar{c}_s}\kappa_k + c_q'. \tag{29}$$

Next, we will bound $g(\tilde{\gamma}_k)$ in (27). Define the parameter $\bar{\epsilon}_s$ that appears in the statement of the lemma as $\bar{\epsilon}_s = c_s\bar{\eta}^{-1/2}$. Then, $\tilde{s}_k\eta_{k-1}^{-1/2} \geq \epsilon_s\mathbf{1}$ since $\epsilon_s \leq \bar{\epsilon}_s$ by assumption. The expression for $\tilde{\gamma}_k$ in (15) can then be reduced to $e^{\tilde{\gamma}_k} = \tilde{s}_k^{-1}\sqrt{\eta_{k-1}}$. One can show that $\lim_{\eta_{k-1}\to 0} g(\tilde{\gamma}_k) = 0$ using the same steps as in the proof of Corollary 2, but we will go a step further here and show that $g(\gamma_k)$ can be bounded by a constant if η_{k-1} satisfies a time-independent bound. We proceed by bounding each factor in (28) that depends on k. Consider first

$$\|e^{\tilde{\gamma}_k}\| \leq \|e^{\tilde{s}_k^{-1}}\|\|e^{\sqrt{\eta_{k-1}}}\| \leq e^{\bar{c}_s^{-1}}\sqrt{m}\|e^{\sqrt{\eta_{k-1}}}\|.$$

Next,

$$\|\mathbf{1} - \frac{1}{2}q(\kappa_k)\|_\infty \leq \|\mathbf{1} - q(\kappa_k)\|_\infty + \frac{1}{2}\|q(\kappa_k)\|_\infty,$$

where

$$\|q(\kappa_k)\|_\infty \leq \|\tilde{s}_k^{-1}\|_\infty\|s^*(x_k, v_k)\|_\infty \leq c_y/\bar{c}_s,$$

where we have used the fact that $\exists c_y > 0$ such that $\|s^*(x,v)\|_\infty \leq c_y$ for all $(x,v) \in \Gamma_N$ since the constraint set $\mathcal{Y}$ is compact. Then, by using the bound in (29), we can state that for any constant $c_q > 0$, $\exists\bar{\eta}_2 > 0$ such that if $\eta_{k-1} \leq \bar{\eta}_2$, then

$$\|\mathbf{1} - \frac{1}{2}q(\kappa_k)\| \leq \sqrt{m}(c_m c_r/\bar{c}_s + c_q + c_y/\bar{c}_s).$$

where we have also used the fact that $\kappa_k \leq 1$ here.

Last, consider the term

$$\|\Phi(\tilde{\gamma}_k)^{-1}\| = \|(H + M^T\mathcal{D}(\tilde{\gamma}_k)M)^{-1}\|.$$

To bound this term, we use the Sherman–Morrison–Woodbury formula and the triangle inequality to write that

$$\|\Phi(\tilde{\gamma}_k)^{-1}\| \leq \|H^{-1}\| + b_1\|(\mathcal{D}(\tilde{\gamma}_k)^{-1} + MH^{-1}M^T)^{-1}\|,$$

for a constant b_1 dependant on H and M. Note that

$$\mathcal{D}(\tilde{\gamma}_k)^{-1} = \text{diag}(e^{2\tilde{\gamma}_k})^{-1} = \text{diag}(\tilde{s}_k^2/\eta_{k-1}).$$

Moreover, note that the matrix $\tilde{\mathcal{D}} = \mathcal{D}(\tilde{\gamma}_k)^{-1} + MH^{-1}M^T$ is diagonally dominant if η_{k-1} is sufficiently small. More specifically, for any constant $c_\phi > 0$ there exists $\bar{\eta}_3 > 0$ such that $\eta_{k-1} \leq \bar{\eta}_3$ implies that

$$\Delta_i(\tilde{\mathcal{D}}) := |\tilde{\mathcal{D}}_{ii}| - \sum_{j\neq i}|\tilde{\mathcal{D}}_{ij}| \geq c_\phi,\ \forall i \in \mathbb{N}_{[1,m]},$$

where $\tilde{\mathcal{D}}_{ij}$ denotes the (i,j) element of the matrix $\tilde{\mathcal{D}}$. To see this, let $\tilde{H} = MH^{-1}M^T$ and note that

$$\begin{aligned}\Delta_i(\tilde{\mathcal{D}}) &= |(\tilde{s}_k)_i^2/\eta_{k-1} + \tilde{H}_{ii}| - \sum_{j\neq i}\|\tilde{H}_{ij}\| \\ &\geq |\bar{c}_s^2/\eta_{k-1} + \tilde{H}_{ii}| - \sum_{j\neq i}\|\tilde{H}_{ij}\|,\end{aligned}$$

since $\|\tilde{s}_k\|_\infty \geq \bar{c}_s$. So, fix a constant $c_\phi > 0$ and let $\bar{\eta}_3$ be such that $\eta_{k-1} \leq \bar{\eta}_3$ implies that $\min_i \Delta_i(\tilde{\mathcal{D}}) \geq c_\phi$. Next, we use [41, Corollary 2] to state that if $\eta_{k-1} \leq \bar{\eta}_3$, then

$$\|(\mathcal{D}(\tilde{\gamma}_k)^{-1} + MH^{-1}M^T)^{-1}\| \leq c_\phi^{-1},$$

and thus $\|\Phi(\tilde{\gamma}_k)^{-1}\| \leq \|H^{-1}\| + b_1 c_\phi^{-1}$.

These bounds can be combined with (28) to state that if $\eta_{k-1} \leq \min\{\bar{\eta}_1, \bar{\eta}_2, \bar{\eta}_3\}$, then

$$g(\tilde{\gamma}_k) \leq b_0\|e^{\sqrt{\eta_{k-1}}}\|, \tag{30}$$

for some constant $b_0 > 0$ independent of k. Thus, for any $c_g > 0$ there exists $\bar{\eta}_4 > 0$ such that $g(\tilde{\gamma}_k) \leq c_g$ if $\eta_{k-1} \leq \bar{\eta}_4$.

Then, by considering the bound for the Newton step in (27) and the bounds in (29) and (30), one can conclude that for any constants $\bar{c}_s \in (0, c_s)$ and $c_0 > 0$, $\exists\bar{\eta} > 0$ such that if $\eta_{k-1} \leq \bar{\eta}$, then

$$\|d(\tilde{\gamma}_k, \eta_{k-1}, \kappa_k)\|_\infty \leq \frac{c_m c_r}{\bar{c}_s}\kappa_k + c_0. \tag{31}$$

Thus, one can define constants, $\bar{\kappa} \in (0, \bar{\kappa}']$, $\bar{c}_s \in (0, c_s)$, and $c_0 > 0$ and such that

$$\frac{c_m c_r}{\bar{c}_s}\bar{\kappa} + c_0 < 1,$$

and define $\bar{\eta}$ accordingly. Thus, $\|d(\tilde{\gamma}_k, \eta_{k-1}, \kappa_k)\|_\infty < 1$ for any $\kappa \leq \bar{\kappa}$ and $\eta_{k-1} \leq \bar{\eta}$. Thus, $\kappa_k = \bar{\kappa}$ and $\eta = \eta_{k-1}$ strictly satisfies the constraints (19b)-(19d). So, (19) is strictly feasible and the optimal solution must satisfy $\kappa_k^* \geq \bar{\kappa}$. ■

APPENDIX II
PROOF OF THEOREM 6

Begin by defining a constant δ satisfying $0 < \delta < \bar{\delta}$, where $\bar{\delta}$ is defined in Proposition 2. By Theorem 5, $\exists k_1 \in \mathbb{N}$ such that $x_k \in \Omega(r, \delta)$ for all $k \geq k_1$. So, consider the warm-started Newton step at a timestep $k > k_1$. Note that $s^*(x_k, r) > 0$ since $x_k \in \Omega(r, \delta)$, and so the warm-started Newton step given a reference $v_k = r$ can be analyzed using Lemma 4 and Corollary 2. So, we write

$$\begin{aligned}d(\tilde{\gamma}_k, \eta_{k-1}, 1) = \mathbf{1} - q(1) \\ - 2\Psi(\tilde{\gamma}_k)^T\Phi(\tilde{\gamma}_k)^{-1}\Psi(\tilde{\gamma}_k)(\mathbf{1} - \frac{1}{2}q(1)),\end{aligned}$$

where and $q(1) = \tilde{s}_k^{-1} \odot s^*(x_k, r)$, with Φ and Ψ following the definitions in Lemma 4. Thus,

$$\|d(\tilde{\gamma}_k, \eta_{k-1}, 1)\|_\infty \leq \|\mathbf{1} - q(1)\|_\infty + g(\tilde{\gamma}_k),$$

where $g(\tilde{\gamma}_k)$ is defined in Appendix I-B. Then, by retracing the steps in Appendix I-B, one can derive that for any $\bar{c}_s' \in (0, c_s)$

and $c_0' > 0$, there exists $\bar{\eta}' > 0$ and $\bar{\epsilon}_s' > 0$ such if $\eta_{k-1} \leq \bar{\eta}'$ and $\epsilon_s \leq \bar{\epsilon}_s'$, then

$$\|d(\tilde{\gamma}_k, \eta_{k-1}, 1)\|_\infty \leq \frac{c_m}{\bar{c}_s'}\|r - v_{k-1}\| + c_0',$$

where c_m is the constant in (31). In addition, since v_k converges to r, then for any $\epsilon_r > 0$, $\exists k_2 \geq k_1$ such that $k \geq k_2$ implies $\|r - v_{k-1}\| \leq \epsilon_r$. Thus, define $\bar{c}_s'$, c_0', and ϵ_r such that

$$\frac{c_m \epsilon_r}{\bar{c}_s'} + c_0' \leq 1,$$

and define $\bar{\eta}'$ and k_2 accordingly. Thus if $k > k_2$ and $\eta_{k-1} \leq \bar{\eta}'$, then $\|d(\tilde{\gamma}_k, \eta_{k-1}, 1)\|_\infty \leq 1$ and so $\kappa = 1$ and $\eta = \eta_{k-1}$ will be a feasible solution to the CG optimization problem (19). ■

APPENDIX III
COMPUTATION OF (d_0, d_1, d_2)

Let $\gamma \in \mathbb{R}^m$ be a fixed log-domain variable and define constants $\eta_1, \eta_2 > 0$, $\kappa_1, \kappa_2 \in [0, 1]$, $\eta_1 \neq \eta_2$, $\kappa_1 \neq \kappa_2$. Let $(x, v) \in \Gamma_N$ and consider the Newton directions $d(\gamma, \eta_i, \kappa_i)$ corresponding to $\mathcal{P}_N(x, v + \kappa_i(r - v))$ for each of the four permutations of the parameters η_i and κ_i for $i \in \{1, 2\}$. These Newton directions satisfy

$$\begin{aligned}
\sqrt{\eta_1} M^T e^\gamma \odot (\mathbf{1} + \hat{d}_1) &= H\hat{\mathrm{u}}_1 + W\theta_1, \\
\sqrt{\eta_1} M^T e^\gamma \odot (\mathbf{1} + \hat{d}_2) &= H\hat{\mathrm{u}}_2 + W\theta_2, \\
\sqrt{\eta_2} M^T e^\gamma \odot (\mathbf{1} + \hat{d}_3) &= H\hat{\mathrm{u}}_3 + W\theta_1, \\
\sqrt{\eta_2} M^T e^\gamma \odot (\mathbf{1} + \hat{d}_4) &= H\hat{\mathrm{u}}_4 + W\theta_2,
\end{aligned}$$

where $\theta_i = (x, v + \kappa_i(r - v))$ for $i \in \{1, 2\}$. By combining these four equations and performing some algebraic manipulation, one can arrive at the equation,

$$\sqrt{\eta} M^T e^\gamma \odot (\mathbf{1} + d) = H\hat{\mathrm{u}} + W_x x + W_v[v + \kappa(r - v)],$$

where

$$\begin{aligned}
\kappa &= p\kappa_1 + (1-p)\kappa_2, & \hat{\mathrm{u}} &= q\bar{U}_1 + (1-q)\bar{U}_2, \\
d &= q\bar{d}_1 + (1-q)\bar{d}_2, & \frac{1}{\sqrt{\eta}} &= q\frac{1}{\sqrt{\eta_1}} + (1-q)\frac{1}{\sqrt{\eta_2}}, \\
\bar{d}_1 &= p\hat{d}_1 + (1-p)\hat{d}_2, & \bar{U}_1 &= p\hat{\mathrm{u}}_1 + (1-p)\hat{\mathrm{u}}_2, \\
\bar{d}_2 &= p\hat{d}_3 + (1-p)\hat{d}_4, & \bar{U}_2 &= p\hat{\mathrm{u}}_3 + (1-p)\hat{\mathrm{u}}_4,
\end{aligned} \tag{32}$$

and $p, q \in \mathbb{R}$ are free constants that we will specify later. Repeating the exact same procedure, but instead starting with the four equations for (10b) gives

$$\sqrt{\eta} e^{-\gamma} \odot (\mathbf{1} - d) = M\hat{\mathrm{u}} + \ell + L_x x + L_v[v + \kappa(r - v)].$$

One can observe that $\hat{\mathrm{u}}$ and d defined in (32) solve the Newton step equations (10a) and (10b) for the parameters κ and η defined in (32).

Thus, one can compute $d(\gamma, \eta, \kappa)$ for a desired (η, κ) by defining $p = a_0 + a_1\kappa$ and $q = b_0 + b_1\eta^{-1/2}$, where $a_0 = -\kappa_2(\kappa_1 - \kappa_2)^{-1}$, $a_1 = (\kappa_1 - \kappa_2)^{-1}$, $b_0 = -\eta_2^{-1/2}(\eta_1^{-1/2} - \eta_2^{-1/2})^{-1}$, $b_1 = (\eta_1^{-1/2} - \eta_2^{-1/2})^{-1}$. Substituting these expressions for p and q into the equation for d in (32) yields, after some algebraic manipulation is performed,

$$d = d_0 + d_1\frac{1}{\sqrt{\eta}} + d_2\frac{\kappa}{\sqrt{\eta}} + d_3\kappa,$$

where

$$\begin{aligned}
d_0 &= b_0c_1 + (1-b_0)c_3, & d_1 &= b_1(c_1 - c_3), \\
d_2 &= b_1(c_2 - c_4), & d_3 &= b_0c_2 + (1-b_0)c_4, \\
c_1 &= a_0\hat{d}_1 + (1-a_0)\hat{d}_2, & c_2 &= a_1(\hat{d}_1 - \hat{d}_2), \\
c_3 &= a_0\hat{d}_3 + (1-a_0)\hat{d}_4, & c_2 &= a_1(\hat{d}_3 - \hat{d}_4).
\end{aligned} \tag{33}$$

Further, we must have that $d_3 = 0$ according to (23), so we can eliminate the need to directly compute one Newton direction (e.g., $\hat{d}_4$) by using the equality $d_3 = 0$ to obtain

$$\hat{d}_4 = b_0(1-b_0)^{-1}(\hat{d}_1 - \hat{d}_2) + \hat{d}_3. \tag{34}$$

Thus, $d_0(\gamma)$, $d_1(\gamma)$, and $d_2(\gamma)$ in Proposition 1 can be computed using the following procedure:

1) Define $\eta_1 > 0, \eta_2 > 0, \kappa_1 \in [0, 1]$, and $\kappa_2 \in [0, 1]$ such that $\eta_1 \neq \eta_2$ and $\kappa_1 \neq \kappa_2$,
2) Compute a factorization of the matrix $(M^T\mathcal{D}(\gamma)M + H)$,
3) For three permutations of η_1, η_2, κ_1, and κ_2, compute the corresponding Newton directions $\hat{d}_1$, $\hat{d}_2$, and $\hat{d}_3$ by solving (11) using the factorization of $(M^T\mathcal{D}(\gamma)M + H)$,
4) Compute $\hat{d}_4$ using (34),
5) Compute d_0, d_1, d_2 using (33).

ACKNOWLEDGMENT

The authors thank Dominic Liao-McPherson for providing implementations of the optimization algorithms used to produce Figure 6.

REFERENCES

[1] J. Rawlings and D. Q. Mayne, *Model Predictive Control: Theory and Design*. Madison, WI: Nob Hill Publishing, 2009.
[2] J. Richalet, A. Rault, J. Testud, and J. Papon, "Model predictive heuristic control: Applications to industrial processes," *Automatica*, vol. 14, no. 5, pp. 413–428, 1978.
[3] H. Borhan, A. Vahidi, A. M. Phillips, M. L. Kuang, I. V. Kolmanovsky, and S. Di Cairano, "MPC-based energy management of a power-split hybrid electric vehicle," *IEEE Transactions on Control Systems Technology*, vol. 20, no. 3, pp. 593–603, 2012.
[4] C. E. Beal and J. C. Gerdes, "Model predictive control for vehicle stabilization at the limits of handling," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 4, pp. 1258–1269, 2013.
[5] S. DiCairano, H. Park, and I. Kolmanovsky, "Model predictive control approach for guidance of spacecraft rendezvous and proximity maneuvering," *International Journal of Robust and Nonlinear Control*, vol. 22, no. 12, pp. 1398–1427, 2012.
[6] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.
[7] F. Borrelli, A. Bemporad, and M. Morari, *Predictive Control for Linear and Hybrid Systems*. Cambridge University Press, 2017.
[8] B. Kouvaritakis and M. Cannon, "Model predictive control: Classical, robust and stochastic," *Switzerland: Springer International Publishing*, 2016.
[9] G. C. Goodwin, M. M. Seron, and J. A. De Doná, *Constrained Control and Estimation: An Optimisation Approach*, 1st ed. Springer Publishing Company, Incorporated, 2010.
[10] A. Domahidi, E. Chu, and S. P. Boyd, "ECOS: An SOCP solver for embedded systems," *2013 European Control Conference (ECC)*, pp. 3071–3076, 2013.

[11] P. Patrinos and A. Bemporad, "An accelerated dual gradient-projection algorithm for embedded linear model predictive control," *IEEE Transactions on Automatic Control*, vol. 59, no. 1, pp. 18–33, 2014.
[12] D. Liao-McPherson and I. Kolmanovsky, "FBstab: A proximally stabilized semismooth algorithm for convex quadratic programming," *Automatica*, vol. 113, p. 108801, 2020.
[13] G. Frison and M. Diehl, "HPIPM: a high-performance quadratic programming framework for model predictive control," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 6563–6569, 2020, 21st IFAC World Congress.
[14] Y. Wang and S. Boyd, "Fast model predictive control using online optimization," *IEEE Transactions on Control Systems Technology*, vol. 18, no. 2, pp. 267–278, 2010.
[15] C. Kirches, A. Potschka, H. G. Bock, and M. Diehl, "qpOASES: a parametric active-set algorithm for quadratic programming," *Mathematical Programming Computation*, vol. 6, pp. 327–363, 2014.
[16] M. Diehl, R. Findeisen, F. Allgower, H. G. Bock, and J. P. Schloder, "Nominal stability of real-time iteration scheme for nonlinear model predictive control," *IEE Proceedings - Control Theory and Applications*, vol. 152, no. 3, pp. 296–308, May 2005.
[17] D. Liao-McPherson, M. Nicotra, and I. Kolmanovsky, "Time-distributed optimization for real-time model predictive control: Stability, robustness, and constraint satisfaction," *Automatica*, vol. 117, p. 108973, 2020.
[18] A. Zanelli, Q. Tran-Dinh, and M. Diehl, "A Lyapunov function for the combined system-optimizer dynamics in inexact model predictive control," *Automatica*, vol. 134, p. 109901, 2021.
[19] D. Liao-McPherson, T. Skibik, J. Leung, I. V. Kolmanovsky, and M. M. Nicotra, "An analysis of closed-loop stability for linear model predictive control based on time-distributed optimization," *IEEE Transactions on Automatic Control*, pp. 1–1, 2021.
[20] J. Leung, D. Liao-McPherson, and I. V. Kolmanovsky, "A computable plant-optimizer region of attraction estimate for time-distributed linear model predictive control," in *2021 American Control Conference (ACC)*, 2021, pp. 3384–3391.
[21] S. Richter, C. N. Jones, and M. Morari, "Computational complexity certification for real-time MPC with input constraints based on the fast gradient method," *IEEE Transactions on Automatic Control*, vol. 57, no. 6, pp. 1391–1403, 2011.
[22] M. Rubagotti, P. Patrinos, and A. Bemporad, "Stabilizing linear model predictive control under inexact numerical optimization," *IEEE Transactions on Automatic Control*, vol. 59, no. 6, pp. 1660–1666, June 2014.
[23] M. N. Zeilinger, D. M. Raimondo, A. Domahidi, M. Morari, and C. N. Jones, "On real-time robust model predictive control," *Automatica*, vol. 50, no. 3, pp. 683–694, 2014.
[24] C. Feller and C. Ebenbauer, "A stabilizing iteration scheme for model predictive control based on relaxed barrier functions," *Automatica*, vol. 80, pp. 328–339, 2017.
[25] D. Limon, I. Alvarado, T. Alamo, and E. Camacho, "MPC for tracking piecewise constant references for constrained linear systems," *Automatica*, vol. 44, no. 9, pp. 2382–2387, 2008.
[26] D. Limon, T. Alamo, and E. Camacho, "Enlarging the domain of attraction of MPC controllers," *Automatica*, vol. 41, no. 4, pp. 629–635, 2005.
[27] T. Skibik, D. Liao-McPherson, T. Cunis, I. Kolmanovsky, and M. M. Nicotra, "A feasibility governor for enlarging the region of attraction of linear model predictive controllers," *IEEE Transactions on Automatic Control*, vol. 67, no. 10, pp. 5501–5508, 2022.
[28] T. Skibik, D. Liao-McPherson, and M. M. Nicotra, "A terminal set feasibility governor for linear model predictive control," *IEEE Transactions on Automatic Control*, pp. 1–7, 2022.
[29] J. Leung, F. Permenter, and I. V. Kolmanovsky, "A computationally governed log-domain interior-point method for model predictive control," in *2022 American Control Conference (ACC)*, 2022, pp. 900–905.
[30] E. G. Gilbert and K. T. Tan, "Linear systems with state and control constraints: the theory and application of maximal output admissible sets," *IEEE Transactions on Automatic Control*, vol. 36, no. 9, pp. 1008–1020, 1991.
[31] F. Permenter, "Log-domain interior-point methods for convex quadratic programming," *Optimization Letters*, 2023.
[32] Z.-P. Jiang and Y. Wang, "Input-to-state stability for discrete-time nonlinear systems," *Automatica*, vol. 37, no. 6, pp. 857–869, 2001.
[33] E. Garone, S. Di Cairano, and I. Kolmanovsky, "Reference and command governors for systems with constraints: A survey on theory and applications," *Automatica*, vol. 75, pp. 306–328, 2017.
[34] R. Seidel, "Small-dimensional linear programming and convex hulls made easy," *Discrete & Computational Geometry*, vol. 6, no. 3, pp. 423–434, 1991.
[35] S. J. Wright, *Primal-dual interior-point methods*. Society for Industrial and Applied Mathematics, 1997.
[36] J. Wurts, J. L. Stein, and T. Ersal, "Collision imminent steering using nonlinear model predictive control,," in *2018 American Control Conference (ACC)*, 2018, pp. 4772–4777.
[37] T. Goldstein, B. O'Donoghue, S. Setzer, and R. Baraniuk, "Fast alternating direction optimization methods," *SIAM Journal on Imaging Sciences*, vol. 7, no. 3, pp. 1588–1623, 2014.
[38] J. Nocedal and S. J. Wright, *Numerical optimization*, 2nd ed. New York, NY: Springer, 2006.
[39] C. Schmid and L. Biegler, "Quadratic programming methods for reduced hessian sqp," *Computers Chemical Engineering*, vol. 18, no. 9, pp. 817–832, 1994, an International Journal of Computer Applications in Chemical Engineering.
[40] D. Liao-McPherson, M. Huang, and I. Kolmanovsky, "A regularized and smoothed fischer–burmeister method for quadratic programming with applications to model predictive control," *IEEE Transactions on Automatic Control*, vol. 64, no. 7, pp. 2937–2944, 2019.
[41] J. Varah, "A lower bound for the smallest singular value of a matrix," *Linear Algebra and its Applications*, vol. 11, no. 1, pp. 3–5, 1975.

Jordan Leung is a Ph.D. candidate in the department of aerospace engineering at the University of Michigan, Ann Arbor, MI, USA. He received his MASc. in aerospace engineering from the University of Toronto, ON, Canada in 2019, and his BASc. in engineering physics from Queen's University, Kingston, ON, Canada 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), Cambridge MA, USA, 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, Ann Arbor, MI, USA, 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, Kolmanovsky was with Ford Research and Advanced Engineering in Dearborn, Michigan for close to 15 years. He is a Fellow of IEEE and a Senior Editor of IEEE Transactions on Control Systems Technology.