Reference Governor for Input-Constrained MPC to Enforce State Constraints at Lower Computational Cost

Miguel Castroviejo-Fernandez, Jordan Leung and Ilya Kolmanovsky

Abstract—In this paper, a control scheme is developed based on an input constrained Model Predictive Control (MPC) law and the idea, usual of Reference Governors (RG), of modifying the reference command to enforce constraints. The proposed scheme, referred to as the RG-MPC, can handle (possibly nonlinear) state and input constraints and only requires optimization for MPC with polytopic input constraints for which fast algorithms exist. Conditions are given that ensure recursive feasibility of the RG-MPC scheme and finite-time convergence of the modified reference command to the desired reference command. Simulation results for a spacecraft rendezvous maneuver with linear and nonlinear constraints demonstrate that the RG-MPC scheme has lower average computational time than state and input constrained MPC with similar performance.

I. INTRODUCTION

Model Predictive Control (MPC) is informed by optimization of a state and input dependent cost function. At each time step, the input sequence that minimizes this cost subject to constraints on the inputs and/or the states [1] is computed and the input is set to the first element of the sequence. While MPC has emerged as an effective control strategy for constrained systems and is used in many applications, one of its primary drawbacks is the high computational cost associated with solving the optimization problem at each time step. This computational cost can be significantly lowered in the case of short horizon Linear Quadratic MPC with only input constraints (uMPC) by exploiting the underlying structure of the cost to speed up gradient computations as in the Fast MPC algorithm [2] or by employing accelerated primal projected gradient methods [3]. In addition, it is easier to enforce anytime feasibility properties [4] for input constrained MPC (e.g., by saturating the computed input in the case of box constraints), analyze the impact of inexact optimization [5], [6], certify an inexact solution [7] and exploit the regularity properties as compared to the state constrained case. For example, the analysis of inexact implementation of state and input constrained MPC in [8] requires the addition of regularization terms in the primal-dual hessian and requires to allow for bounded constraint violation. Finally, to handle nonlinear constraints the use of more computationally expensive nonlinear MPC is required.

To capitalize on advantages of uMPC with polytopic constraints yet be able to handle state constraints and additional nonlinear input constraints, in this paper, we consider the augmentation of uMPC with a reference governor (RG). RGs [9] are add-on schemes that ensure, at each time step, appropriate selection of the reference command so that subsequent trajectories remain feasible with respect to constraints. However, the direct application of existing RGs to MPC-based closed-loop systems is difficult. For instance, if RG is based on online prediction [10], [11], a uMPC optimization problem will need to be solved at each time step over the reference governor prediction horizon; this will likely exceed the computational cost of a state and input constrained MPC (cMPC).

In this paper we propose a new scheme which enables a computationally efficient application of RGs to complement uMPC in controlling linear systems with (possibly nonlinear) state constraints and nonlinear input constraints. This scheme, that we refer to as RG-MPC, only requires that a single uMPC optimization problem be solved per time step. This is especially relevant for systems with limited computing power or fast sampling rates.

For the proposed RG-MPC scheme we show, under suitable assumptions, the recursive feasibility as well as finite-time convergence of the modified reference command to the desired constant reference command, i.e. properties expected of conventional RGs. Simulation results for a spacecraft rendezvous (RdV) problem demonstrate low computational requirements and good closed-loop performance being achieved with the proposed approach.

The paper is organized as follows. In Section II the class of systems being addressed is discussed and the two main ingredients: uMPC and the Incremental Reference Governor (IRG) of [11], needed for subsequent developments are reviewed. Section III introduces the proposed RG-MPC scheme and presents theoretical results. Finally, numerical simulations of the proposed scheme applied to a spacecraft RdV maneuver are reported in Section IV.

Notations: $S^n_{++}$, $S^n_+$ denote the set of symmetric $n \times n$ positive definite and positive semi-definite, matrices respectively. $I_n$ denotes the $n \times n$ identity matrix and $0_{n \times m}$ an $n \times m$ matrix with zero at every entry. Given $x \in \mathbb{R}^n$ and $W \in S^n_+$, the $W$-norm of $x$ is $\|x\|_W = \sqrt{x^T W x}$. Given $P \in S^n_{++}$, $y \in \mathbb{R}^n$, $B_P(y, r) = \{ x \in \mathbb{R}^n \mid \|y - x\|_{P} \leq r \}$ and $\lambda_+(P)$ is the maximum eigenvalue of $P$. Given $a \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, $(a, b) = [a^T, b^T]^T$. The sequence made of the $\alpha_j \in \mathbb{R}^n$, $j = n, \ldots, m$ elements is denoted by $\{a_j\}_{j=n}^m$. For a given set $W \subseteq \mathbb{R}^n$, $int W$ denotes the interior of $W$. The set $\mathbb{N}$ is the set of positive integers and $\mathbb{N}_0$ the set of non negative ones.

1University of Michigan, Ann Arbor, MI 48109 USA mcastrov, jmlleung, ilya@umich.edu. This research is supported by Air Force Office of Scientific Research Grant number FA9550-20-1-0385.
II. PRELIMINARIES

A. Class of systems

We consider a class of systems represented by the following linear discrete-time models,
\[
\begin{align*}
    x_{k+1} &= Ax_k + Bu_k, \\
    y_k &= Cx_k,
\end{align*}
\]
where \( k \in \mathbb{N}_0 \) is the discrete time index, \( x_k \in \mathbb{R}^n \) is the state vector, \( u_k \in \mathbb{R}^m \) is the input vector and \( y_k \in \mathbb{R}^p \) is the tracking output vector. The system is subject to pointwise-in-time constraints on both states and inputs:
\[
z_k = (x_k, u_k) \in \mathcal{Z}, \quad \forall k \in \mathbb{N}_0, \quad \mathcal{Z} = \mathcal{X} \times \mathcal{U},
\]
where \( \mathcal{X} \subset \mathbb{R}^n \), \( \mathcal{U} \subset \mathbb{R}^m \) are compact, convex sets with the origin in their interiors. Note that \( \mathcal{U} \) is defined considering both polytopic and non-polytopic input constraints.

We define \( \mathcal{U}_p \supseteq \mathcal{U} \) as the set obtained when considering only polytopic input constraints. Furthermore, we make the following assumption:

Assumption 1: The pair \((A, B)\) is stabilizable.

B. Characterization of the steady states and inputs

We consider the reference command (set-point) tracking problem of bringing the output, state, and input of the system to a specific set-point \( r \in \mathbb{R}^p \) and to the associated steady states and inputs \( x_{ss}, u_{ss} \), respectively. Using the usual definition of a steady state of (1), the set-points must satisfy the following:
\[
\begin{bmatrix}
    A - I_n & -B \\
    C & -I_p
\end{bmatrix}
\begin{bmatrix}
    x_{ss} \\
    u_{ss}
\end{bmatrix}
= 0.
\]

Assumption 1 ensures that (3) has a non-trivial solution set of steady states \([12]\). In the following, we only consider the case where, for a given \( r \), (3) has a unique solution and define \( z_{ss}(r) = (x_{ss}(r), u_{ss}(r)) \), where \( (x_{ss}(r), u_{ss}(r)) \) are solutions to (3). Given the existence of constraints, the following equation describes an inner approximation of the set of admissible reference commands.
\[
\mathcal{R} = \left\{ r \in \mathbb{R}^p \mid \exists z \in \bar{\mathcal{Z}}, \quad M \begin{bmatrix} z \\ r \end{bmatrix} = 0 \right\}
\]
where \( \mathcal{Z} \subset \text{int} \mathcal{Z} \) is a compact and convex set. This, under Assumption 1, implies that \( \mathcal{R} \) is compact and convex.

C. Input constrained MPC

As explained in the introduction, uMPC offers several advantages as compared to state and input constrained MPC (cMPC). In the following, we consider short-horizon uMPC with a quadratic cost function,
\[
J(\xi, \mu, v) = \sum_{i=0}^{N_{\text{MPC}}-1} ||\xi_i - x_{ss}(v)||_Q^2 + ||\mu_i - u_{ss}(v)||_P^2,
\]
where \( \xi = \{\xi_i\}_{i=0}^{N_{\text{MPC}}-1} \), \( \mu = \{\mu_i\}_{i=0}^{N_{\text{MPC}}-1} \), \( Q \in \mathbb{R}^{n \times n} \), \( R \in \mathbb{R}^{m \times m} \), \( P \in \mathbb{R}^{n \times n} \) and \( N_{\text{MPC}} \in \mathbb{N} \). The MPC law is defined using the solution to the following Optimal Control Problem (OCP) \( Pr(x, v, N_{\text{MPC}}) \):
\[
\begin{align}
    \min_{\xi, \mu} J(\xi, \mu, v) \\
    \text{such that} \quad & \xi_0 = x \\
    & \xi_{i+1} = A\xi_i + B\mu_i, \quad i = 0, \ldots, N_{\text{MPC}} - 1, \\
    & \mu_i \in \mathcal{U}_p, \quad i = 0, \ldots, N_{\text{MPC}} - 1.
\end{align}
\]
We assume that

Assumption 2: \( Q \in \mathbb{S}^{n}_{++}, R \in \mathbb{S}^{m}_{++}, P \in \mathbb{S}^{n}_{++} \) and \( P \) is the solution to the Discrete Algebraic Riccati Equation (DARE): 

\[
K = (B^TPB + R)^{-1}(B^TPA),
\]
is the associated LQR gain. Finally, let
\[
\{u_j^*(x, v, N_{\text{MPC}})\}_{j=0}^{N_{\text{MPC}}-1}
\]
denote the solution to \( Pr(x, v, N_{\text{MPC}}) \). Then, at time instant \( k \) the MPC input is given by \( u_k = u_0^*(x_k, v_k, N_{\text{MPC}}) \).

Assumption 1 and \( Q \in \mathbb{S}^{n}_{++} \) ensure the existence of a stabilizing solution to the DARE in Assumption 2, and since \( 0 \in \text{int} \mathcal{U}_p \), the MPC law is locally stabilizing at strictly constraint admissible equilibria \([13]\). Note that the MPC law described in this section only handles polytopic input constraints. State constraints and non-polytopic input constraints will be addressed by the IRG developed in the next section.

D. Incremental Reference Governor (IRG)

For the time being, suppose that a control law for system (1),
\[
u = g(x, r),
\]
which depends on the state \( x \) and reference command \( r \), is available. For \( j \in \mathbb{N}_0 \), we define \( u_j^0(x, r) = g(x_j^0(x, r), r) \) and \( x_j^0(x, r) = A^jx + \sum_{i=0}^{j-1} A^{j-1-i}Bu_i^0 \) for \( j \geq 1 \) and \( x_0^0(x, r) = x \). The corresponding state-input vector is \( z_j^0(x, r) = (x_j^0, u_j^0) \).

Now, considering (1) in closed-loop with controller (7), the aim of the IRG is to adjust the reference command that the system follows so that constraints are enforced. The IRG accomplishes this by testing whether an increment of the current reference command leads to constraint admissible trajectories.

More specifically, at each time step, the reference increment is parameterized as \( v^+ = v_{k-1} + \kappa v_{\text{dir}} \), where \( \kappa \in [0, 1] \) is a parameter that dictates the rate at which \( v_k \) converges to \( r \),
\[
v_{\text{dir}} = r - v_0,
\]
where \( v_0 \in \mathcal{R} \) is such that \( \{z_j^0(x_0, v_0)\}_{j=0}^{\infty} \) does not violate constraints and \( x_0 \) is the initial state. If the constraints hold for \( \{z_j^0(x_k, v^+)\}_{j=0}^{\infty} \) then \( v_k = v^+ \), otherwise, \( v_k = v_{k-1} \).

For certain problems, e.g., if the control law (7) is an LQR and there are only polytopic constraints, it is possible to compute the Maximum Output Admissible Set (MOAS),
\( O_{\infty} (v) \), associated with \( Z \), (7) and \( v \in \mathbb{R} \). The constraint evaluation step is then reduced to verifying
\[
x_0^g (x_k, v^+) \in O_{\infty} (v^+).
\]
However, if \( O_{\infty} (v^+) \) (or a good inner approximation of it) cannot be computed, an alternative approach [10] is to predict state and control trajectories and verify if
\[
z_j^g (x_k, v^+) \in Z, \ j = 0, \ldots, N_{\text{RG}} - 1,
\]
\[
x_j^R (x_k, v^+) \in T^R (v^+),
\]
where \( T^R (v^+) \subset O_{\infty} (v^+) \) is a forward invariant set that contains \( x_{ss} (v^+) \) in its interior and \( N_{\text{RG}} \in \mathbb{N} \) is a fixed horizon length. Note that \( T^R (v^+) \), is potentially small as compared to \( O_{\infty} (v^+) \). In this case, using the prediction instead of verifying \( x_0^g (x_k, v^+) \in T^R (v^+) \) extends the feasible region as entering \( T^R (v^+) \) is only required after \( N_{\text{RG}} \) steps.

If the control law (7) is the uMPC from section II-C, computing the MOAS is difficult as the closed-loop system is nonlinear. A prediction-based approach, nevertheless, can be used to implement the IRG. Note, however, that at each time instant, to compute the predicted input sequence over \( N_{\text{RG}} \) steps, one must solve \( N_{\text{RG}} \) optimization problems of the form (4). This has the potential to be computationally prohibitive, negating the advantages of using efficient uMPC solvers to alleviate computational burden. In the next section, we introduce the RGMPC scheme that has lower computational requirements.

### III. PROPOSED RGMPC SCHEME

Based on the ingredients introduced in the last two sections we now introduce our RGMPC scheme which augments uMPC to handle (potentially non-polytopic) state constraints and non-polytopic input constraints, whilst having a low computational effort.

Consider an input sequence, \( \{ u_j^{\text{ext}} (x, v) \} \), where
\[
u_j^{\text{ext}} (x, v) = \begin{cases} u_j^* (x, v, N_{\text{GPC}}) & \text{if } j < N_{\text{GPC}} \\ \Pi_{U_0} [K (x_j^{\text{ext}} - x_{ss} (v)) + u_{ss} (v)] & \text{if } j \geq N_{\text{GPC}} \end{cases}
\]
where \( K \) is defined in (5). \( \Pi_{U_0} (...) \) denotes the projection operator onto the set \( U_0, x_j^{\text{ext}} = A_j x + \sum_{i=0}^{j-1} A_{j-i} B_j u_i^{\text{ext}} \) and \( v \in \mathbb{R} \). Sequence (9) is the optimal input sequence of (4) padded with a saturated LQR law for \( j \geq N_{\text{GPC}} \).

Suppose that the sequence (9) has been computed at a time instant \( k \) for the reference command \( v^+ \). A sufficient condition to ensure that this sequence and its associated state trajectory satisfy the constraints is that
\[
z_j^{\text{ext}} (x_k, v^+) \in Z, \ j \leq N_{\text{RG}} - 1
\]
\[
x_j^{R} (x_k, v^+) \in T^{LQR} (v^+),
\]
where \( N_{\text{GPC}} \) is typically much smaller than \( N_{\text{RG}} \) and \( T^{LQR} (v^+) \subset \mathbb{R}^n \) is a constraint admissible forward invariant set for system (1) under the LQR law associated with \( Q \) and \( R \). Algorithm 1 describes the proposed RGMPC scheme and the interaction between the different elements is portrayed in Figure 1. At any given time, Algorithm 1 requires solving one short horizon LQ MPC optimization problem with polytopic input constraints and, at most, verifying constraints satisfaction \( N_{\text{RG}} + 1 \) times.

**Algorithm 1** Input generation at time instant \( k \).

**Require:** \( x_k \): the current state, \( v_{k-1} \): the reference used at time \( k-1 \), \( k' \): the last time instant at which \( v_{k'} \neq v_{k'-1} \) (default \( k' = 0 \)), \( \{ u_j^{\text{ext}} (x_k, v_k) \} \) \( j = 0 \) to \( N_{\text{GPC}} - 1 \), and \( v_{dir} \).

1. select \( k_k \in [0, 1] \)
2. compute \( v^+ = v_{k-1} + \kappa_k v_{dir} \)
3. compute \( \{ u_j^{\text{ext}} (x_k, v_k) \} \) \( j = 0 \) to \( N_{\text{GPC}} - 1 \), and \( z_j^{\text{ext}} \) \( j = 0 \)
4. if \( \{ z_j^{\text{ext}} \} \) violates (10) then
5. \( v_k = v_{k-1} \), \( k_k = 0 \)
6. if \( k - k' < N_{\text{GPC}} \) then
7. \( u_k = u_k^{\text{ext}} (x_k, v_k) \)
8. else
9. \( u_k = \Pi_{U_0} [K (x_k - x_{ss} (v_k)) + u_{ss} (v_k)] \)
10. end
11. \( v_k = v^+ \)
12. \( u_k = u_0^{\text{ext}} (x_k, v^+) \)
13. \( k' = k \)
14. return \( u_k, v_k, k' \), \( \{ u_j^{\text{ext}} (x_k, v_k) \} \) \( j = 0 \) to \( N_{\text{GPC}} - 1 \), \( \kappa_k \)

**Remark 1:** Algorithm 1 checks constraints for sequence (9) corresponding to the incremented reference command \( v^+ \). If constraints are satisfied, the incremented reference is accepted, \( v_k = v^+ \). If not, the reference is held constant and the corresponding element of the MPC sequence computed at the time instant \( k' \) (the last instant the reference command was updated) is applied. Note that, if RGMPC is not able to update \( v_k \) for more than \( N_{\text{GPC}} - 1 \) steps it switches to saturated LQR feedback.

**Remark 2:** The choice of the terminal set \( T^{LQR} (v) \) is application specific. A common choice is the MOAS of the LQR controlled closed-loop system. In the case of polytopic constraints the MOAS is a polytope and can be computed in closed form [14]. For non polytopic constraints, if a polytopic approximation is possible, the problem is reduced to the previous case. Another choice for \( T^{LQR} (v) \) are constraint admissible sublevel sets of Lyapunov functions of the LQR controlled system. If \( \hat{P} \) is the solution to the Lyapunov equation: \( (A + BK)^T \hat{P} (A + BK) - \hat{P} = I \) then sets of the form \( T^e (v) = \{ x \mid \| x - x_{ss} (v) \|_2 \leq c \} \) are forward
invariant. We can then choose \( c \) such that \( I^c(v) \subseteq Z \). This is a specific case of sets used in the RGs introduced in [15].

The values of \( \kappa_k \) in line 1 of Algorithm 1 must be carefully selected. For example, if some constraints are active in specific regions of the state space, entering that region may require a smaller reference increment. Conversely, to accelerate the response, we usually look for the largest \( \kappa_k \) that is admissible. In reference governors, the choice of \( \kappa_k \) is often resolved by solving an optimization problem: maximize \( \kappa_k \) such that the corresponding reference increment leads to a closed-loop state and input sequence that is constraint admissible [9]. In Algorithm 2, we propose a simple \( \kappa_k \) selection logic to ensure that a reference increment is feasible in finite-time without the need for the RG optimization problem to be solved.

**Algorithm 2** Selection of reference increment, \( \kappa_k \), for
Algorithm 1 at time instant \( k \)

Require: \( \kappa^0 \in (0,1] \): a default value of the increment. \( N_a \in \mathbb{N} \) a tuning parameter, \( k' \): the last time step such that \( v_{k'} \neq v_{k-1} \) (default \( k' = 0 \), \( v_{k-1}, v_0 \), \( r \) and \( \{\kappa_j\}_{j=0}^{k} = 0 \).

1: if \( k - k' \leq N_a \) then
2: \( \kappa_k = \kappa^0 \)
3: else
4: \( \kappa_k = \frac{\kappa^0}{k-k'-N_a} \)
5: if \( \sum_{j=0}^{k} \kappa_j > 1 \) then
6: \( \kappa_k = \kappa_r \), where \( \kappa_r = 1 - \sum_{j=0}^{k-1} \kappa_j \), so that \( r = v_{k-1} + \kappa_r (r - v_0) \)
7: return \( \kappa_k \): to be used in Algorithm 1 at time step \( k \)

For \( r \in \mathcal{R} \) we define the set
\[
\mathcal{P}(r) = \left\{ x \in \mathbb{R}^n \mid x_{j}^{ext}(x,v) \in \mathcal{Z} \forall j \geq 0 \right\} \cap \left\{ x \in \mathbb{R}^n \mid x_{N_a}^{ext}(x,v) \in \mathcal{LQR}(r) \right\}
\]
as the set of states for which the sequence generated by control (9) satisfies (10). We assume that:

**Assumption 3:** There exists \( \epsilon > 0 \) such that \( \forall v \in \mathcal{R}, B(x_{ss}(v), \epsilon) \subseteq \mathcal{P}(v) \).

Asymptotic convergence of a reference governor literature. Moreover, in this paper, we consider trajectories of a linear system with inputs from sequence (9): an initial MPC sequence followed by a stabilizing LQR controller to a steady state that lies in the interior of \( \mathcal{Z} \). Given this, it is reasonable to assume that there will exist a neighborhood of the steady state such that the trajectory obtained from any initial state in that neighborhood will not lead to subsequent constraint violations. We also introduce the set
\[
\Gamma = \{(x,v) \in \mathbb{R}^n \times \mathcal{R} \mid x \in \mathcal{P}(v)\},
\]
as of state and reference command pairs for which (10) is verified. We now study theoretical properties of RG-MPC as defined by Algorithm 1 and Algorithm 2. To facilitate this analysis, we first establish some preliminary results.

**Lemma 1:** Given an asymptotically stable (A.S.) linear system \( x_{k+1} = A_c x_k, x \in \mathbb{R}^n \), and a compact set \( \mathcal{S} \subseteq \mathbb{R}^n \) with the origin in its interior, it follows that \( \forall \delta > 0, \exists N \in \mathbb{N} \) such that \( \forall j \geq N, A_c^j x_0 \in B(0,\epsilon/2), \forall x_0 \in \mathcal{S} \).

**Proof:** Given the system is A.S., following classical Lyapunov stability results for discrete linear systems, \( \lambda_{\max} \in\{0,1\} \) such that \( V(A_c^j x_0) \leq \delta V(x_0) \), \( \forall j \geq 1 \). Now, let \( c_1 = \max \{V(x) \mid x \in \mathcal{S}\} \), which exists given continuity of \( V \) and compactness of \( \mathcal{S} \). Let \( c_2 \) be such that \( \{\delta \mid V(x) \leq c_2\} \subseteq B(0,\epsilon/2) \), then, choosing \( N = \min \{j \mid c_1 \geq c_2\} \) completes the proof.

**Lemma 2:** Given Assumptions 1 and 2, and \( v \in \mathcal{R} \) it follows that \( \forall \delta > 0, \exists \delta, \forall x_{ss}(v) \in \mathcal{S} \) such that \( \forall j \geq N_{\delta,v}, x_{j}^{ext}(x,v) \in B(x_{ss}(v),\delta), \forall x \in \mathcal{P}(v) \).

**Proof:** We define \( x_{j}^{ext} = x_{j}^{ext}(x,v) \). Given that (10) holds for \( \{x_{j}^{ext}\}_{j=0}^{N_a} \), then \( x_{N_a}^{ext} \in \mathcal{LQR}(v) \subseteq \mathcal{O}_{\mathcal{LQR}}(v) \). Hence, for all \( j \geq N_{\delta,v} \), \( x_{j}^{ext} \) is derived from an LQR with gain matrix \( K \). This, combined with Assumption 1, makes \( x_{j}^{ext} \) equivalent to a trajectory of (1a) controlled using LQR with A.S. equilibrium \( x_{ss}(v) \). Making the change of variable \( \tilde{x} = x - x_{ss}(v) \), the dynamics of the associated LQR controlled system are given by \( \dot{\tilde{x}}_{k+1} = (A-BK)\tilde{x}_k \). Also, note that, [14, Theorem 2.1 (ii)] states that \( \mathcal{O}_{\mathcal{LQR}}(v) \) is compact by compactness of \( X, U \). Therefore, for all \( v \in \mathcal{R} \), Lemma 1 states that for the system \( \dot{\tilde{x}}_{k+1} = (A-BK)\tilde{x}_k \) and associated \( \mathcal{O}_{\mathcal{LQR}}(v) \), \( \forall \delta > 0 \exists N \) such that \( \forall j \geq N, (A-BK)\tilde{x}_j \subseteq B(0,\epsilon/2), \forall x \in \mathcal{O}_{\mathcal{LQR}}(v) \). Introducing \( N_{\delta,v} = N_{\delta,v} + N \) directly implies \( \forall j \geq N_{\delta,v}, x_{j}^{ext}(x,v) \in B(x_{ss}(\delta),\delta), \forall x \in \mathcal{P}(v) \).

**Lemma 3:** Given Assumption 1 and 2, it follows that \( \forall \delta > 0, \exists \delta, \forall x_{ss}(v) \in \mathcal{S} \) such that \( \forall j \geq N_{\delta,v}, x_{j}^{ext}(x,v) \in B(x_{ss}(v),\delta), \forall x \in \mathcal{P}(v) \).

**Proof:** From Lemma 2 \( \forall v \in \mathcal{R} \exists \delta, \forall x_{ss}(v) \in \mathcal{S} \) such that \( \forall j \geq N_{\delta,v}, x_{j}^{ext}(x,v) \in B(x_{ss}(v),\delta), \forall x \in \mathcal{P}(v) \). As stated in the proof of Lemma 2, the rate of decay of the A.S. system associated with \( x_{j}^{ext} \) does not depend on the reference. Instead, \( N_{\delta,v} \) depends on \( v \) through the size of \( \mathcal{O}_{\mathcal{LQR}}(v) \). In other words, the set \( \mathcal{S} \) and associated \( c_1 \) in Lemma 1 change with \( v \). However, \( \forall v \in \mathcal{R}, \mathcal{O}_{\mathcal{LQR}}(v) \subseteq \mathcal{X} \). Thus, given compactness of \( \mathcal{X} \) and continuity of the weighted distance between two points: \( \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}_+, \delta_p(a,b) = ||a - b||_p \), then \( \delta_p(\cdot,\cdot) \) reaches a maximum over \( \mathcal{X} \times \mathcal{X} \). We can bound the \( c_1 \) constants by \( c_1 = \frac{1}{2} \max \{\delta_p(a,b) \mid a,b \in \mathcal{X} \times \mathcal{X} \} \). In a similar way to the proof of Lemma 1, we define \( N_{\delta,v} = N_{\delta,v} + N \). Therefore \( N_{\delta,v} \leq N_{\delta,v} \), \( \forall x \in \mathcal{R} \).

Before presenting the next Lemma we introduce \( \mathcal{R}_k = \{v \in \mathcal{R} \mid \{v\} + B(0,1,\kappa) \subseteq \mathcal{R} \} \) where \( \kappa > 0 \).

**Lemma 4:** If Assumptions 1-3 hold, there exists \( N_{e/2} \in \mathbb{N} \) such that given \( v \in \mathcal{R}_k \) and \( x \in \mathcal{P}(v) \) then \( \forall x \in [0,\kappa], \forall j \geq N_{e/2}, x_{j}^{ext}(x,v) \in \mathcal{P}(v^{+}) \), where \( v^{+} = v + \kappa \delta_{\text{div}}, \delta_{\text{div}} = r - v_0 \) and \( \{v_0, r\} \subseteq \mathcal{R}_k \). Moreover, the constant \( \kappa \) is independent from \( x \).

**Proof:** Select \( \epsilon \) from Assumption 3. By Lemma 3, \( \exists N_{e/2} \in \mathbb{N} \) such that
\[
\forall j \geq N_{e/2}, x_{j}^{ext}(x,v) \in B(x_{ss}(v),\epsilon/2), \forall x \in \mathcal{P}(v).
\]
Now, let $x^{ext}_j = x^{ext}_j(x,v)$. Then, for any $j \geq N_\varepsilon/2$, it follows that

$$
||x^{ext}_j - x_{ss}(v^+)|| = ||x^{ext}_j - x_{ss}(v) - \kappa x_{ss}(v_{dir})|| \\
\leq ||x^{ext}_j - x_{ss}(v)|| + ||\kappa x_{ss}(v_{dir})||, \\
\leq \frac{\varepsilon}{2} + \kappa ||x_{ss}(v_{dir})||,
$$

where the second line follows from the triangle inequality and the third from (11). Defining: $\bar{\kappa} = \frac{\varepsilon}{2||x_{ss}(v_{dir})||}$

$$
||x^{ext}_j - x_{ss}(v^+)|| \leq \varepsilon, \forall k \in [0, \bar{\kappa}].
$$

Therefore, $x^{ext}_j(x,v) \in P(v^+)$, by Assumption 3.

**Remark 3:** Note that the maximum step size, $\bar{\kappa}||x_{ss}(v_{dir})||$ depends only on $\varepsilon$. The set $R_{\bar{\kappa}}$ can be made arbitrarily close to $R$ by decreasing the value of $\bar{\kappa}$. This is achieved by decreasing the value of $\varepsilon$. Validity of Assumption 3 is still ensured. Also, compactness of $R$ is inherited by $R_{\bar{\kappa}}$ [16, Theorem 2.1 (x)]. Furthermore, in Lemma 4 the convex hull of $\{v_0, r\}$ lies inside $R_{\bar{\kappa}}$. This is relevant, as in Algorithm 1, $v^+$ is inside the convex hull of $\{v_0, r\}$.

**Proposition 1:** Consider system (1) with IC $x_0$ and desired set-point $r \in R$. If $\exists v^0 \in R$ such that $(x_0, v^0) \in V$, initializing $v_0 = v^0$ in Algorithm 1 and defining $v_{dir}$ according to (8) ensures recursive feasibility of Algorithm 1: $x_k \in X \Rightarrow z_k \in Z, \forall k \geq 0$.

**Proof:** The claim follows directly from the assumptions and implementation of Algorithm 1. Using Algorithm 1, the trajectory of the system between two subsequent reference increments at times $k_1, k_2 \in \mathbb{N}$, $k_1 < k_2$ is given by $\{z^{ext}(x_{k_1}, v_{k_1})\}_{j=0}^{k_2-k_1}$. For a reference increment to be performed, at time $k_2$, $\{z^{ext}(x_{k_2}, v_{k_2})\}_{j=0}^{\infty} \subseteq Z$ is required. Finally, $(x_0, v_0) \in V$ implies that $\{z^{ext}(x_0, v_0)\}_{j=0}^{\infty} \subseteq Z$. Therefore $z_k \in Z, \forall k \geq 0$. As a result recursive feasibility is ensured.

**Proposition 2:** Consider the problem of bringing system (1), controlled using Algorithms 1-2, to the final set-point $r \in R_{\bar{\kappa}}$ which is constant in time, from the initial state $x_0$ subject to constraints (2). Suppose that Assumptions 1-3 hold, and that $\exists v^0$ such that $(x_0, v^0) \in V$. If $v_0 = v^0$ and $v_{dir}$ is defined according to (8), then finite-convergence of $v_k$ to $r$ and asymptotic convergence of the state, $x_k$, to $x_{ss}(r)$ is ensured.

**Proof:** Suppose $v_0 \neq r$. We then need to show there exists $k^* \in \mathbb{N}$ such that $\forall j \geq k^*$, $v_j = r$. Define

$$
\Delta v_k = v_k - v_0 = s_k v_{dir}, \quad s_k = \sum_{j=0}^{k} \kappa_j,
$$

Since $\kappa_k \geq 0$ $\forall k$, showing finite-time convergence to $r$ is equivalent to showing that $\exists k^* \in \mathbb{N}$ such that $\forall j \geq k^*$, $s_j = 1$. We do this by contradiction.

**Hypothesis (H):** $\exists k^* \in \mathbb{N}$ such that $\forall j \geq k^*$, $s_j = 1$. First, define $N_{\varepsilon/2}$ as in Lemma 4, $N_{\bar{\kappa}} = \min\{i \in \mathbb{N} \mid \frac{\varepsilon}{2||x_{ss}(v_{dir})||} \leq \bar{\kappa}\}$ and define $N = \max\{N_{\varepsilon/2}, N_{\bar{\kappa}}\}$. At any time instant, $k_1 \in \mathbb{N}$ consider the last instance such that there was a change in the reference: $k' = \max\{i \leq k_1 \mid v_{i-1} \neq v_i\}$. Now, assume that $v_{k'+N-1} = v_k$, then, following Algorithms 1-2, $x^{ext}_{k'+N} = x^{ext}_N(x_k, v_k) \in P(v_k + \bar{\kappa} v_{dir})$, where $v_{k'+N}$ is the tested reference at time $k' + N$. This is because, from Lemma 4, $\forall j \geq N_{\varepsilon/2}, x^{ext}_j(x_k, v_k) \in P(v_k + \bar{\kappa} v_{dir})$, and because at time $N$ the tested increment is smaller than $\bar{\kappa}$. This implies $v_{k'+N} \neq v_k$ and $\kappa_{k'+N} \geq \frac{\varepsilon}{2||x_{ss}(v_{dir})||}$. Now, if an advance of the reference takes place at any $j \in \mathbb{N} \cap [k', k'+N]$, then $\kappa_k \geq \frac{\varepsilon}{2||x_{ss}(v_{dir})||} \geq \frac{\varepsilon}{2||x_{ss}(v_{dir})||}$ by the implementation of Algorithm 2. Hence, the reference is increased at least every $N$ steps. Thus, there exists an infinite sequence of time instants $\{k_j\}_{j=0}^{\infty}$ such that $s_{k_j} \geq \frac{\varepsilon}{2||x_{ss}(v_{dir})||}$. In turn, the sequence $\{s_{k_j}\}_{j=0}^{\infty}$ diverges. Now, choose $k^* = \min k$ such that $s_k \geq 1$. At that time instant, Line 6 of Algorithm 2 is executed, and given that $\kappa_r \leq \bar{\kappa}$ an increment of $\kappa_r$ is performed, ensuring $s_k = 1$. Lines 5-6 of Algorithm 2 ensure that $\forall j \geq k^*$

$$
s_j = 1, \forall j \geq k^*.
$$

Thus, convergence of $v_k$ to $r$ in finite-time is proved.

Convergence of the state to $x_{ss}(r)$ is directly implied using Lemma 3.

**Remark 4:** By examining the proof of Proposition 2, it can be shown that the results hold for other $\kappa_k$ selection strategies as long as such strategies ensure that whenever $k - k'$ becomes large, $\kappa_k \leq \bar{\kappa}$.

In Proposition 2, finite-time convergence to the desired reference is proven. This ensures that after a given time the reference governor becomes inactive and the system recovers nominal closed-loop performance.

Future work will focus on robustness of the proposed scheme. In the case of bounded external disturbances, one direction is to verify constraints under all possible disturbances and ensuring that the terminal set $I^{LQR}(v)$ is a robustly forward invariant set in presence of the disturbances.

**IV. APPLICATION TO CONSTRAINED SPACECRAFT RENDEZVOUS**

We consider a problem of spacecraft rendezvous to a target on a circular orbit. The relative motion dynamics are represented by the CWH equations [17] given by

\begin{align*}
\dot{x} &= A_c x + B_c u, \\
\dot{y} &= C_c x,
\end{align*}

with $x \in \mathbb{R}^6, u \in \mathbb{R}^3$, describing the relative motion of the spacecraft in the Hill’s frame centered at the target. The first three and last three states represent radial, along track and cross track positions and velocities of the spacecraft, respectively. The inputs are relative accelerations (thrust normalized by mass) along the three axes. In (12),

$$
A_c = \begin{bmatrix}
0_{3 \times 3} & I_{3 \times 3}
\end{bmatrix},
$$

$$
B_c = \begin{bmatrix}
0_{3 \times 3} \\
I_{3 \times 3}
\end{bmatrix}, \quad C_c = \begin{bmatrix}
I_{3 \times 3} & 0_{3 \times 3}
\end{bmatrix},
$$

1205
where $n = \sqrt{\mu/r_0}$, $\mu$ is the gravitational parameter and $r_0$ is the orbital radius of the nominal orbit.

The system has the following state and control constraints:

- **Input saturation:** $\|u\|_\infty \leq 0.1$.
- **Maximum velocity:** $|x_i| \leq 3$, $i = 4, 5, 6$.
- **The spacecraft must remain in front of the target in the in-track direction, $x_2 \geq 0$.**
- **Line of sight cone (nonlinear, convex constraint):** The spacecraft should remain in the 15 degree cone defined by $x_1^2 + x_3^2 - \tan^2(15^\circ)(x_2 + 1)^2 \leq 0$.
- **Final velocity (if-then constraint):** When close to the target, the relative velocity should be small enough to avoid damage: If $x_2 \leq 2$ then $x_1^2 + x_3^2 + x_6^2 \leq 0.1^2$

The spacecraft relative motion dynamics have forced equilibria of the form:

$$r = [a, b, c]^\top, \quad a, b, c \in \mathbb{R}$$

$$\ddot{u}_{ss}(r) = -3n^2 a, \quad 0, n^2 c \big]^\top,$$

where the elements of $r$ correspond to the output states in $y$ and the relative velocity is zero at all forced equilibria. Simulations are performed considering a nominal orbit at 500 km altitude above the earth. When discretizing the linearized system, a sampling period $T_s = 0.5$ sec is used. When we apply Algorithm 1, the MPC has a prediction horizon $N_{sp} = 20$ and the constraint satisfaction is assessed over a horizon of $N_{cg} = 120$ steps. The terminal set is defined as $T^{\text{LQR}}(v) = T^s(v)$, $c = 0.05$, $T^s(v)$ is a sublevel set of a Lyapunov function as defined in Remark 2. The value $c = 0.05$ was selected such that $T^{\text{LQR}}(v)$ is constraint admissible for references selected through (13) for any initial condition $x$ such that $x_2 \geq 50$[m] and $x_1^2 + x_3^2 - \tan^2(14.5^\circ)(x_2 + 1)^2 \leq 0$.

The MPC weight matrices were chosen as $R = I_3$, $Q = \text{diag}([100, 1, 100, 10, 1, 10])$. By relying on the flexibility in the choice of $k_s$, as described in Remark 4, we utilized the following scheme for computing $v^+$ which is better adapted to the problem at hand:

$$v^+ = v_{k-1} + \kappa \Delta v,$$

$$\Delta v = \begin{cases} \text{sign}(r - v_0) \cdot \Delta v_{fix} & |r_2 - v_{k-1, 2}| \geq 20 \text{[m]}, \\ r - v_{k-1} & |r_2 - v_{k-1, 2}| < 20 \text{[m]}, \end{cases}$$

$$\Delta v_{fix} = r - [3.67, 20, 3.67]^\top,$$

$$\kappa = \begin{cases} 1, & |r_2 - v_{k-1, 2}| \leq 0.005|\Delta v_{fix, 2}| \\ 0.1 & \text{else}, \end{cases}$$

where $r = 0_{3 \times 1}$ is the final set-point, and $v_{k-1}$ denotes the previous set-point. The reference is incremented by a fixed amount when far from the target ($\geq 20$[m]), proportionally to the difference between $v_{k-1}$ and $r$ when close to the target, and (13c) ensures $v^+ = r$ when sufficiently close. Initialization of the reference is done by setting $v_0 = Cx_0$ where $x_0$ is the IC.

Figure 2 shows the time histories of states, reference commands and inputs for the spacecraft starting at $x(0) = [10, 100, 20, 0, 0, 0]^\top$ and controlled by the proposed RGMPc scheme with the Fast MPC solver [2]. The simulation shows convergence to the target spacecraft in 100 sec while respecting constraints on both states and inputs. Figure 3 (bottom, left and center) shows two dimensional projections of the trajectory as well as of the line of sight constraint which are respected at all times. Finally, Figure 3 (bottom right) depicts the velocity norm for times from around 75 sec and onward as well as the terminal velocity constraint when it is active. The velocity norm rides the constraint boundary before going to 0 as the spacecraft converges to the final set-point.

Figure 3 (top) shows the instants at which the reference is changed during the maneuver. After 95 sec the final set-point is reached. In most cases when the reference is held constant, it remains only for 1 or 2 time instants. Only in one occasion does this occur for a significantly longer period: for 14 time steps (approx. 80 sec after the start). Hence, with the proposed reference switching logic, the saturated LQR is not used.
A. Comparison to the Fast-MPC without add on scheme

To confirm the necessity of a state constraint handling mechanism we perform simulations over a grid of IC with either a uMPC or the proposed RGMPC. We consider, at a distance $x_2 = 50$ [m], 200 points from concentric circles in the $x_1$-$x_3$ plane. The radii go up to $r^2 = \tan^2(14.5^\circ)(50^2 + 1)$. This set of values combined with zero initial velocity is used as the set of ICs. Simulations resulted in the RGMPC not violating constraints a single time while the uMPC violated constraints for each IC. In particular, for each IC the spacecraft passed behind the target spacecraft and the terminal velocity constraint was violated. Figure 4 highlights which ICs lead to violation of the cone constraint by the uMPC controller. As expected this is often when starting away from the center line of the cone.

![Fig. 4: Values of the IC that lead to subsequent violation of the cone constraint when considering uMPC. All ICs have $x_2 = 50$ [m] and zero velocity. For all points shown, the RGMPC satisfies the constraints at all times.](image)

B. Comparison to a saturated LQR-IRG scheme

We assess the advantages of the presented scheme with respect to more conventional RG implementation. To do so, we compare the performance of the RGMPC scheme with that of a saturated LQR extended with an IRG, referred to as sLQR-RG. The choice of a saturated LQR is motivated by its closeness to the uMPC. In the comparison, we consider the following metrics:

- Successful initialization of the RG and no constraint violation, denoted as “succ. sim” type Boolean.
- Time required to reach the target spacecraft within a specified tolerance, denoted as $t_{conv}$ [s].
- An input cost that relates to fuel consumption [18, Section 14.3], computed as $u_{cost} = \int_0^\infty ||u(t)||^2 dt$ [N^2 kg^{-2} s].

Taking the same uniform grid of ICs as in the previous section and for the same $Q$ and $R$ matrices, we performed simulations with both the RGMPC and sLQR-RG controllers. Table I summarizes results for the different metrics for the two controllers. The differences in input cost and time of convergence are also shown. Successful simulations are achieved for every IC and both controllers. It is notable that the RGMPC outperforms the sLQR-RG in every single simulation both in maneuver time and in fuel cost. In particular, the RGMPC provides a mean reduction of 21% in maneuver time and of 70% in fuel consumption, both substantial values.

|                | # succ. sim. | mean $u_{cost}$ | mean $t_{conv}$ |
|----------------|--------------|-----------------|-----------------|
| RGMPC          | 200          | 0.9             | 75.91           |
| sLQR-RG        | 200          | 2.98            | 95.74           |

Table I: Number of successful simulations and mean values of $u_{cost}$ and $t_{cost}$ for the RGMPC and sLQR-RG schemes.

To explain the difference in performance between the sLQR-RG and RGMPC we consider the state and input trajectories. Figure 5 shows the radial component of the state (top) and input (bottom) trajectories for a single IC for the sLQR-RG (red) and RGMPC (blue). One can observe that unlike the RGMPC, the sLQR-RG generated input is prone to oscillations between the saturation values. This oscillation is directly translated into the position evolution as depicted in Figure 5.

![Fig. 5: Radial component of position (top) and input thrust (bottom) evolution when considering the RGMPC (blue) and sLQR-RG (red).](image)

C. Comparison to state and input constrained MPC

We next assess the viability of the proposed RGMPC scheme in comparison to a state and input constrained MPC, referred to as cMPC. In this section, all OCPs, both for the RGMPC and the cMPC, are solved with a dual active set solver [19]. Once again, we compare the set of trajectories obtained starting at the 200 initial conditions described in Section IV-A. Additionally, we also use a second, similar set of trajectories starting from $x_2 = 100$ [m]. To estimate computational power requirement we collect, at each time step, the time required to compute the control input: $t_{comp}$. For one simulation, the average time required to compute the input commands is referred to as $t_{comp,av}$. Simulations were performed using Matlab on a machine with a 2.3 GHz 8-Core Intel Core i9.

The RGMPC has $N_{RGMPC} = 20$ and $N_R = 120$, while three cMPC formulations were used for comparison. The cMPC differed in their horizon lengths: $N_1 = 20, N_2 = 60$ and $N_3 = 120$. Additionally, to implement cMPC, constraints were made polyhedral by the following modifications:

- A polyhedral approximation of the line of sight cone using 15 linear inequalities.
- The if-then terminal constraint on velocity is enforced by setting the terminal reference to $r = [0 \ 4 \ 0]^T$ and imposing $x_2 \geq 3$. 

1207
The rest of the constraints as well as the rest of the simulation parameters were kept identical to those of previous sections. For all ICs an infeasible OCP for the $N_1$ cMPC was encountered. In all cases, this was due to overstepping the lower saturation bound on $x_2$. The spacecraft reached high velocities and was not able to decelerate in time to avoid constraint violation due to the short horizon of the cMPC. All other controllers successfully performed the docking maneuver for all ICs. It should be noted that, apart from $t_{\text{comp}}$, results for $N_2$ and $N_3$ were almost identical, with only slight differences in $u_{\text{cost}}$. The difference in time required to reach the final reference for the three controllers: RGMPC, cMPC $N_2$, $N_3$ was never longer than 1 sec.

Figure 6 shows statistics of $u_{\text{cost}}$ and $t_{\text{comp,av}}$ for RGMPC and for cMPC with horizons $N_2$, $N_3$. With respect to the input cost, both cMPC controllers perform similarly. On average, the RGMPC slightly outperforms both cMPCs when considering ICs with $x_2 = 50$ [m] and underperforms either cMPC when starting at $x_2 = 100$ [m]. The average computational time, $t_{\text{comp,av}}$, (lower figure) differs substantially among the controllers. The RGMPC has $t_{\text{comp,av}}$ that are one order of magnitude smaller than the cMPC with $N_2$ and almost two orders smaller than the cMPC with $N_3$. For this example, by looking at the $t_{\text{comp}}$ of the cMPC with $N_1$ (not presented here), it was assessed that the difference in $t_{\text{comp,av}}$ came from the difference in the MPC prediction horizon lengths and not so much from the additional constraints in the OCPs of the cMPC.

VI. CONCLUSION

An input constrained Linear Quadratic MPC can be augmented by a variant of an incremental reference governor (IRG) to avoid violations of (possibly nonlinear) state constraints and nonlinear control constraints. The proposed scheme is designed to avoid MPC optimization at every time instant over the IRG prediction horizon by relying on the previously computed MPC input sequence padded with the saturated LQR. Finite-time convergence properties of the modified IRG reference command to a strictly steady-state constraint admissible reference command have been established. Simulation results demonstrate computational advantages of the proposed scheme over both input and state constrained MPC and performance advantages over a saturated LQR controller augmented with the IRG.

References

[1] J. B. Rawlings, D. Q. Mayne, and M. Diehl, *Model predictive control: Theory, computation, and design*. Nob Hill Publishing Madison, WI, 2017, vol. 2.
[2] M. Kögel and R. Findeisen, “A fast gradient method for embedded linear predictive control,” *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 1362–1367, 2011.
[3] Y. E. Nesterov, “A method for solving the convex programming problem with convergence rate $0(1/k^2)$,” in *Dokl. akad. nauk Sssr*, vol. 269, 1983, pp. 543–547.
[4] 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)*. IEEE, 2021, pp. 3384–3391.
[5] A. Bemporad, “Reference governor for constrained nonlinear systems,” *IEEE Transactions on Automatic Control*, vol. 45, no. 3, pp. 415–419, 1998.
[6] E. G. Gilbert and I. Kolmanovsky, “Nonlinear tracking in the presence of state and control constraints: a generalization of the reference governor,” *Automatica*, vol. 38, no. 12, pp. 2063–2073, 2002.
[7] E. Gilbert and I. Kolmanovsky, “Theory and computation of disturbance invariant sets for discrete-time linear systems,” *Mathematical problems in engineering*, vol. 4, no. 4, pp. 317–367, 1998.
[8] W. Chohessy and R. Wiltshire, “Terminal guidance system for satellite rendezvous,” *Journal of the Aerospace Sciences*, vol. 27, no. 9, pp. 653–658, 1960.
[9] P. Gurfil and P. K. Seidelmann, *Celestial mechanics and astrodynamics: theory and practice*. Springer, 2016, vol. 436.