2013 European Control Conference (ECC)
July 17-19, 2013, Zürich, Switzerland.

Prioritization Schemes for Reference and Command Governors

Uroš Kalabić Yash Chitalia Julia Buckland Ilya Kolmanovsky

***Abstract*— Reference and command governors are predictive add-on schemes that are applied to closed-loop systems and guarantee constraint enforcement while tracking desired reference inputs. This paper introduces two methods for using these predictive methods in the presence of prioritized constraints. The first method handles the case of "soft" constraints by using a slack variable to relax the constraints. The second method prioritizes the reference inputs, enforcing the constraints by modifying the lower priority reference inputs first. Two examples are reported consisting of a constrained spring-mass-damper system and an F-16 aircraft with actuator constraints.**

I. Introduction

Reference governors (RGs) [1]–[4] and command governors (CGs) [5]-[7] ensure constraint enforcement for discrete-time linear closed-loop systems of the form,

$$x(t+1) = Ax(t) + Bv(t), \tag{1}$$

$$y(t) = Cx(t) + Dv(t) \in Y,\ \forall t \in \mathbb{Z}_+, \tag{2}$$

where $x(t) \in \mathbb{R}^n$ is the state vector, $v(t) \in \mathbb{R}^m$ is the feasible reference input, $Y \subset \mathbb{R}^p$ is the constraint set containing zero, and $\mathbb{Z}_+$ is the set of non-negative integers.

Given a reference signal, $r(t) \in \mathbb{R}^m$, RGs and CGs modify $v(t)$ to ensure constraint-admissibility. Specifically, $v(t) = r(t)$ if no constraint violation is predicted in tracking the nominal reference; thus the tracking behavior is unaffected if the desired reference is constraint admissible. Otherwise, $v(t)$ is computed through the solution of an optimization problem. See Fig. 1 for a schematic of governor operation.

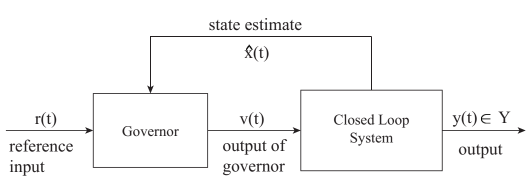

Fig. 1. Schematic of RG or CG placement within the control loop

In this paper, we consider the use of RGs and CGs as applied to prioritized constraint enforcement and prioritized reference tracking. In the first case, CGs are applied to a set of "soft" constraints that have been prioritized through a quadratic cost penalty on a slack variable. In the second case, we consider the application of the RG to inputs $v(t)$, which are modified in order of priority.

The following section presents the motivation for this problem and further outlines the contribution of the paper. The rest of the paper is organized as follows. Section III introduces RG and CG theory; the next two sections present the two methods considered in this paper along with theoretical results and a numerical example: Section IV introduces a method for prioritized constraint optimization using slack variables and Section V introduces the prioritized RG; Section VI has the concluding remarks.

II. Problem Motivation

Typical formulations of the reference governor are applied to systems with hard constraints, *i.e.* systems where $y(t) \in Y$ for all $t \in \mathbb{Z}_+$ is a required condition. However, soft constraints with $y(t) \notin Y$ being undesirable but permitted under certain conditions, are also of interest. For instance, in some appications especially in model predictive control, output constraints are often treated as soft to ensure the solution can be computed even if constraint violation cannot be avoided [8]. In this situation, a control action which mitigates and reduces constraint violation is generated. In other applications, it is allowed to trade-off constraint violation against the improvements in tracking performance [9].

In this paper, we firstly introduce an algorithmic method of relaxing constraints through the use of quadratic penalty functions on slack variables that weight constraint infringement of Y against desired reference set-points in order to achieve a balance between tracking performance and constraint enforcement. We apply this quadratic cost function to the CG and illustrate the operation of the CG in the presence of prioritized constraints using a mass spring damper example that is subject to both hard and soft constraints in mass position. The hard constraint corresponds to a hard barrier which the mass must not hit and the soft constraint is imposed to limit the amount of overshoot.

Secondly, we consider the problem of applying RG to prioritized reference inputs. In this approach, we aim to achieve $v_i(t) = r_i(t)$ for higher priority i before lower priority i. The example reported in this paper is that of an F-16 aircraft for which the pitch attitude and flight path angle reference reference inputs are prioritized. Two simulations are performed corresponding to two different prioritizations: higher priority of the pitch attitude increases the aerial maneuverability of the aircraft, while higher priority of the flight path angle directly affects how quickly it tracks a desired trajectory.

This work was supported in part by the Ford Motor Company and in part by the National Science Foundation, Award Number 1130160.

U. Kalabić and I. Kolmanovsky are with the Department of Aerospace Engineering and Y. Chitalia is with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI, 48109, USA. Email: {kalabic,ilya,yashc}@umich.edu

J. Buckland is with the Ford Motor Company, Dearborn, MI, 48124, USA. Email: jbucklan@ford.com

III. Preliminaries

The RG and CG ensure constraint adherence of the system (1)-(2) starting from the offline determination of the maximal constraint admissible set of feasible initial condition-constant reference pairs, which is the set of initial conditions and constant references under which the constraints in (2) are satisfied. This set [1], [3] is defined as,

$$O_\infty = \{(x_0, v) : x(0) = x_0, v(t) \equiv v, \text{(1)-(2) satisfied}\}. \tag{3}$$

Using O_∞ along with a given state measurement or estimate, $x(t)$, and desired set-point, $r(t)$, the RG and CG search for feasible set-points in the following set,

$$\Pi(x(t)) = \{v : (x(t), v) \in O_\infty\}. \tag{4}$$

Furthermore, the set of feasible steady-state set-points is,

$$\Omega = \{r : (C\Gamma + D)r \in Y\}, \tag{5}$$

where $\Gamma = (I - A)^{-1}B$ and $x(t) \to \Gamma r$ for $v(t) \equiv r$. If the constraint set is polyhedral, all of the sets above will be polyhedral, so if Y is given as a set of linear inequalities,

$$Y = \{y : Py \le p\}, \tag{6}$$

then we can find matrices [10], [3], H_x, H_r, and h such that,

$$O_\infty = \{(x, v) : H_x x + H_r v \le h\}, \tag{7}$$
$$\Pi(x) = \{v : H_r v \le h - H_x x\}, \tag{8}$$
$$\Omega = \{r : P(C\Gamma + D)r \le p\}. \tag{9}$$

If the set O_∞ is not finitely determined, it admits an arbitrarily close finitely determined inner approximation, which is positively invariant and can be used in place of O_∞ [3].

The RG and CG methods differ in their online search algorithm. The RG assumes an update for $v(t)$ of the form,

$$v(t) = v(t-1) + \kappa(t)(r(t) - v(t-1)), \tag{10}$$

where the largest $\kappa(t) \in [0, 1]$ is chosen that satisfies $v(t) \in \Pi(x(t))$,

$$\begin{aligned} \kappa(t) = &\max \kappa \in [0,1] \\ &\text{sub. t. } v(t-1) + \kappa(r(t) - v(t-1)) \in \Pi(x(t)). \end{aligned} \tag{11}$$

Since Y is polyhedral, the solution to (11) can be obtained by solving finitely many scalar division problems [1], [3].

The CG is updated according to the minimization of a quadratic cost function,

$$v(t) = \underset{v \in \Pi(x(t))}{\arg\min} ||r(t) - v||_Q \tag{12}$$

IV. CGs with Prioritized Constraint Sets

In this section we apply the CG to output constraints of the following form,

$$y(t) \in Y \cap Y_1 \cap \cdots \cap Y_q, \tag{13}$$

where Y is a hard constraint set and Y_i is a soft constraint set for $1 \le i \le q$. That is, Y_i is a set for which $y(t) \in Y_i$ is not a strict inclusion but can be violated, incurring a penalty for doing so.

The penalty is introduced by way of a modifying the set, Y_i. Specifically, we expand Y_i in all directions by a certain amount so that the output, $y(t)$, is contained in this modified set. Since Y_i is polyhedral, this proves especially simple and the scheme is described below. Let Y_i be described as a set of $n_c^{(i)}$ linear inequalities,

$$Y_i = \{y : P^{(i)} y \le p^{(i)}\}, \tag{14}$$

where $P^{(i)} \in \mathbb{R}^{n_c^{(i)} \times n}$ and $p^{(i)} \in \mathbb{R}^{n_c^{(i)}}$. $P^{(i)}$ is such that every row has unit norm, *i.e.*

$$||P_j^{(i)}||_2 = 1. \tag{15}$$

Note that if the constraints are not given in this form, then we can redefine, $p_j^{(i)} := p_j^{(i)}/||P_j^{(i)}||_2$ and $P_j^{(i)} := P_j^{(i)}/||P_j^{(i)}||_2$, without altering the structure of (14).

For each $1 \le i \le q$, we can define an expanded constraint set, Y_i^ε,

$$Y_i^\varepsilon = \{(y, \varepsilon) : P^{(i)} y \le p^{(i)} + \varepsilon_i \mathbf{1}_{n_c^{(i)}}\}, \tag{16}$$
$$= \{(y, \varepsilon) : \begin{bmatrix} P^{(i)} & -\mathbf{1}_{n_c^{(i)}} \end{bmatrix} \begin{bmatrix} y \\ \varepsilon \end{bmatrix} \le p^{(i)}\}, \tag{17}$$

where $\mathbf{1}_{n_c^{(i)}} \in \mathbb{R}^{n_c^{(i)}}$ is a vector with all elements equal to 1 and $\varepsilon \ge 0$ is a scalar. Y_i^ε is the "relaxed" version of Y_i because it expands Y_i by extra dimensions, where $Y_i \times \{0\} \subset Y_i^\varepsilon$ so that ε is the amount of relaxation. Because of condition (15), the polytope constraint Y_i^ε retains the same shape as that of Y_i.

The projection of Y_i^ε onto the y-axis is an expanded set that contains Y_i. In fact, $\text{Proj}_y Y_i^\varepsilon \sim \mathcal{B}_\varepsilon = Y_i$, where we define $\mathcal{B}_\varepsilon = \{y \in \mathbb{R}^p : ||y||_2 \le \varepsilon\}$ as the unit ball and $U \sim V = \{z : z + v \in U, \ \forall v \in V\}$ as the Minkowski (Pontryagin) difference. The proof of this stems from the requirement that the rows of $P^{(i)}$ in (14) are unit norm, *i.e.* $\text{Proj}_y Y_i^\varepsilon \sim \mathcal{B}_\varepsilon = \{z : P_j^{(i)} z \le p_j^{(i)} + \varepsilon - h_{P_j^{(i)\intercal}}(\mathcal{B}_\varepsilon), \ 1 \le j \le n_c\} = \{z : P_j^{(i)} z \le p_j^{(i)}, \ 1 \le j \le n_c\} = Y_i$.

Thus for each soft constraint set, Y_i and slack variable, $\varepsilon_i \ge 0$, there exists a companion relaxed constraint set, $Y_i^{\varepsilon_i}$. We apply the regular CG theory to the set, $Y_i^{\varepsilon_i}$ where each ε_i is treated as a control input constrained to be non-negative. This introduces a new input vector of slack variables,

$$\varepsilon(t) = (\varepsilon_1(t), \ldots, \varepsilon_q(t)). \tag{18}$$

The system (1)-(2) can then be redefined as,

$$x(t+1) = Ax(t) + \begin{bmatrix} B & 0 \end{bmatrix} \begin{bmatrix} v(t) \\ \varepsilon(t) \end{bmatrix}, \tag{19}$$

$$y'(t) = \begin{bmatrix} y(t) \\ \varepsilon(t) \end{bmatrix} = \begin{bmatrix} C \\ 0 \end{bmatrix} x(t) + \begin{bmatrix} D & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} v(t) \\ \varepsilon(t) \end{bmatrix} \in Y', \tag{20}$$

where,

$$\begin{aligned} Y' = \{(y, \varepsilon) \in \mathbb{R}^p \times \mathbb{R}^q : y \in Y, \ y \in Y_i^{\varepsilon_i}, \ \varepsilon_i \ge 0, \\ 1 \le i \le q\}, \end{aligned} \tag{21}$$

is a hard constraint on $y'(t)$. The rest of the development of the CG as applied to prioritized constraints follows by noting that different sets, Y_i, may have different degrees of "softness", *i.e.* $y(t) \in Y_1$ may only admit a small violation, while $y(t) \in Y_q$ being of secondary significance, may allow much larger constraint infringement. We treat this by weighting the elements of the input (both the reference and slack variables) in the same way as the regular CG, expanding the number of set-point commands under consideration by the number of soft constraints.

This amounts to applying the CG to (19)-(20) and updating $(v(t), \varepsilon(t))$ through a modified version of the algorithm in (12). Namely, we define modified constraint admissible sets, O'_∞ and $\Pi'(x)$, for the expanded control vector, $(v(t), \varepsilon(t))$, and develop an update function by introducing a matrix, $Q_\varepsilon = Q_\varepsilon^{\mathrm{T}} > 0$, and solving the following optimization problem at each time step,

$$(v(t), \varepsilon(t)) = \underset{(v,\varepsilon)\in\Pi'(x(t))}{\arg\min} ||r(t) - v||_Q^2 + ||\varepsilon||_{Q_\varepsilon}^2. \quad (22)$$

In this way, we relate the penalty for the constraint infraction on Y_i through the weighting matrix, Q_ε. This matrix is also the prioritization mechanism, *i.e.* a higher weight on ε_i amounts to a higher priority of the associated constraint.

Note that since Y' is the intersection of sets of linear inequality constraints, then Y', and hence O'_∞ and $\Pi'(x)$, can also be expressed as a set of linear inequalities.

The prioritization of constraints guarantees satisfaction of the hard constraint, Y, so as a design consideration, any constraints whose enforcement is required should be included in Y. Because the approach to satisfying the constraints in a soft way is essentially a redefinition of the known CG scheme, the prioritized method shares many of the same properties with the regular CG; though all the properties apply to the output of expanded dimension, $y'(t)$, some properties, such as finite time convergence, are dimension-independent. We now discuss this in further detail.

A. Theoretical results

The use of CGs with prioritized constraints is an extension of CG theory. All of the results of the CG theory apply when considering the dimensionally-expanded system (19)-(20). However, not all results apply to (1)-(2).

For example, a result from the regular CG theory is the convergence to the nearest feasible reference. The relaxed form of the CG exhibits another property, namely that if the steady-state response to the command is not inside the intersection of the soft constraints then, although the CG converges, it may not converge to the desired equilibrium. In order to demonstrate this, we first state the following. According to the conventional CG theory, when there exists an r_s such that for all $t \geq t_s$, $r(t) = r_s$, then there exists a t_f such that $v(t) = r_s^*$ and $\varepsilon(t) = \varepsilon^*$ for all $t \geq t_f$, where,

$$(r_s^*, \varepsilon^*) = \underset{(r,\varepsilon)\in\Omega'}{\arg\min} ||(r - r_s)||_Q^2 + ||\varepsilon||_{Q_\varepsilon}^2, \quad (23)$$

$$\Omega' = \{(r, \varepsilon) : \begin{bmatrix} (C\Gamma + D)r \\ \varepsilon \end{bmatrix} \in Y'\}. \quad (24)$$

Similarly, in the case of prioritized constraints, if the steady state response to r_s is not within Y_i for all $1 \leq i \leq q$, *i.e.* $(C\Gamma + D)r_s \notin Y_i$, then r_s^* may not coincide with r_s. We state the following proposition.

Proposition 1: $r_s^* = r_s$ if and only if $(C\Gamma + D)r_s \in Y_i$ for all $1 \leq i \leq q$.

Proof: We prove sufficiency first. The hypothesis directly implies that $(r_s, 0) \in \Omega'$ by the definition in (24). Since $(r_s, 0)$ is the unique and unconstrained minimum of (23) and $(r_s, 0)$ is contained within constraints, then the solution to (23) is $(r_s^*, \varepsilon^*) = (r_s, 0)$.

For necessity, $r_s^* = r_s$ implies that the solution to (23) is $(0, \varepsilon^*)$. For the case that $\varepsilon^* > 0$, the necessary gradient condition for (23) is not satisfied because we can always tighten the slack variables by letting $r_s^* \neq r_s$ so that $(C\Gamma + D) \in Y_i^{\varepsilon_i}$ for all i where ε_i are elements of ε^{**} and $||\varepsilon^{**}||_{Q_\varepsilon} < ||\varepsilon^*||_{Q_\varepsilon}$. Therefore $\varepsilon^* = 0$ and the proof is complete by the definition in (24). ■

B. Numerical example

This example considers a mass-spring damper with equations of motion,

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -k/m & -c/m \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1/m \end{bmatrix} u(t), \quad (25)$$

where the parameters are taken from [11] to be, $c = 0.6590$, $k = 38.94$, $m = 1.54$, and,

$$u(t) = \begin{bmatrix} 0 & c_d \end{bmatrix} x(t) + kv(t), \quad (26)$$

and where $c_d = 4.0$ is a stabilizing feed-back gain and $v(t)$ is the steady-state set point for $x_1(t)$.

The constraint is that of a hard barrier imposed as $x_1(t) \leq 0.009$. Additionally, a soft constraint of $x_1(t) \leq 0.0075$ is imposed to limit the overshoot. Furthermore, the control is constrained to $0 \leq u(t) \leq 0.3$. To proceed with the controller design, we perform a zero-order hold on the system with the time step, $T = 0.01$.

In the form of the modified output (20),

$$y'(t) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_d & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ \varepsilon(t) \end{bmatrix} + \begin{bmatrix} 0 \\ k \\ 0 \end{bmatrix} v(t) \in Y', \quad (27)$$

where Y' is the expanded constraint (21) set and $\varepsilon(t)$ is the new auxiliary control input measuring soft-constraint adherence.

The simulations consist of a constant set reference set-point, $v(t) \equiv 0.0075$ with initial condition set at the origin, $x(0) = 0$. The three different runs consist of three different weights for the optimization (22), $Q = 10$ and $Q_\varepsilon = 10^1, 10^2, 10^3$. The results, along with a comparison with the ordinary CG, which was developed under the assumption that all constraints are hard, are presented in Figs. 2-5 and show that prioritizing constraint adherence against set-point tracking can be used to manage overshoot of the output.

Fig. 2 shows the three responses of $y(t)$ with the soft constraint plotted. None of the trajectories violate the hard constraint and we see that, as the weight on the slack variable

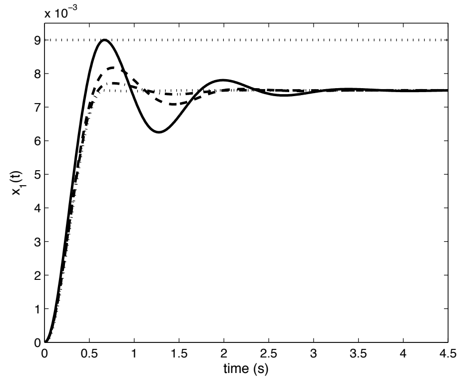

Fig. 2. $x_1(t)$ for three different cases of $Q_\varepsilon = 10^1$ (solid), $Q_\varepsilon = 10^2$ (dashed), $Q_\varepsilon = 10^3$ (dot-dashed) and ordinary CG (dotted); the soft constraint is also dotted

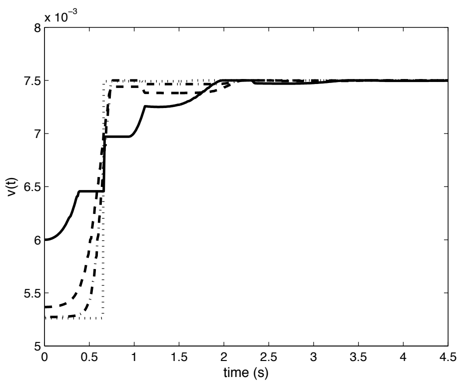

Fig. 3. $v(t)$ for three different cases of $Q_\varepsilon = 10^1$ (solid), $Q_\varepsilon = 10^2$ (dashed), $Q_\varepsilon = 10^3$ (dot-dashed) and ordinary CG (dotted)

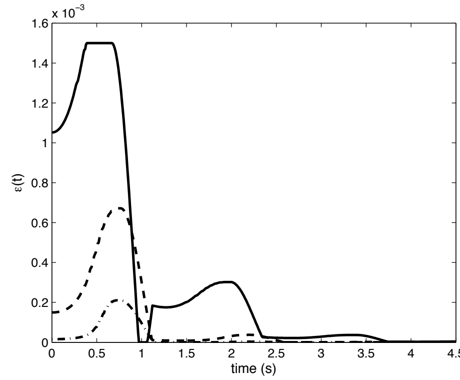

Fig. 4. $\varepsilon(t)$ for three different cases of $Q_\varepsilon = 10^1$ (solid), $Q_\varepsilon = 10^2$ (dashed), $Q_\varepsilon = 10^3$ (dot-dashed)

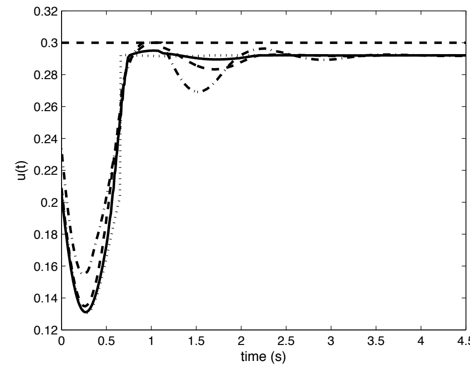

Fig. 5. $u(t)$ for three different cases of $Q_\varepsilon = 10^1$ (solid), $Q_\varepsilon = 10^2$ (dashed), $Q_\varepsilon = 10^3$ (dot-dashed) and ordinary CG (dotted)

increases, the amount of overshoot in the corresponding response is lessened and the trajectory becomes more similar to the ordinary CG case. Conversely, the higher the weight, the longer it takes for the reference, $v(t)$, to converge, as shown in Fig. 3. Fig. 4 shows the slack variable response and that lower weights correspond to higher amounts of slack. Note that in all cases, the constraint on $u(t)$ is satisfied as shown in Fig. 5.

V. Prioritized Reference Governors

Prioritized RGs (PRGs) operate by modifying a set of desired set-points in order of priority. With this approach, the higher priority commands are modified as little as possible and the closest feasible set-points are achieved for the highest priority commands.

In order to accomplish this, we order the elements of the vector, $v(t)$, in order of priority, such that $v_i(t)$ has higher priority than $v_j(t)$ when $i < j$. We then define *prioritized constraint sets*, which are slices of $\Pi(x(t))$ that depend on the desired reference, $r(t)$,

$$\Pi_i(x,r) = \{v \in \Pi(x) : v_{i-1} = r_{i-1}, \dots, v_1 = r_1\}. \quad (28)$$

In this way $\Pi_i(x,r)$ is the set of feasible set-points for x when we set the first $i-1$ set-points to their desired values. From the above definition, it follows that $\Pi_1(x,r) = \Pi(x)$ and $\Pi_i(x,r) \supset \Pi_{i+1}(x,r)$. Note that the recoverable domain of initial states consists of all $x(0)$ for which $\Pi(x(0))$ is non-empty.

The prioritization scheme is developed by defining a diagonal matrix,

$$\mathbf{K}(t) = \text{diag}(\kappa_1(t), \dots, \kappa_m(t)), \quad (29)$$

with the requirement that $\kappa_i(t) \in [0,1]$ for all $1 \le i \le m$. Subsequently, we implement the Vector RG update policy from [12] that is similar to the regular reference governor,

$$v(t) = v(t-1) + \mathbf{K}(t)(r(t) - v(t-1)). \quad (30)$$

The PRG online solution algorithm is different from that of the Vector RG. It proceeds by finding the first set, $\Pi_i(x(t))$, such that there exists a feasible solution for $\kappa(t)$. Thus we choose the largest i such that $\Pi_i(x(t), r(t)) \neq \emptyset$ and set $\kappa_1(t) = \cdots = \kappa_{i-1}(t) = 1$ in order to perform the following optimization,

$$\kappa_i(t) = \underset{\kappa_i}{\arg\max}\ \kappa_i \in [0,1] \text{ sub. to } v(t) \in \Pi'(i), \quad (31)$$

where $\Pi'(i) = \Pi_i(x(t), r(t))$. If $\Pi'(i)$ is a polyhedral set, then (31) is a linear programming problem.

Note that the recoverable domain of initial states consists of all $x(0)$ for which $\Pi(x(0))$ is non-empty. Furthermore, the solution to (31) may not be unique. In order to ensure the uniqueness of $\kappa_i(t)$ (and therefore $v_i(t)$), we can set $r_i(t) = v_i(t)$ and re-run the optimization by maximizing $\kappa_{i+1}(t)$ with the new set, $\Pi'(i+1)$. We repeat this until we obtain a unique solution to (31). The need for this sequential optimization of references can arise when direct constraints on the reference are present; *e.g.* suppose $y(t) = v(t) \in Y$, where $Y = [-0.5, 0.5]^m$, $v(t-1) = 0$, and $r(t) = (1, 1)$. Then any $v(t) = (0.5, v_{cet})$, where $v_{cet} \in [0, 0.5]^{m-1}$ solves (31). In such a case, sequentially optimizing in the manner described above obtains a unique solution.

To summarize, we present the PRG in algorithmic form.

PRG Algorithm

1) Obtain the current state and desired set-point pair, $(x(t), r(t))$. Set $i := m + 1$.
2) Decrement $i := i - 1$.
3) If $\Pi_i(x(t)) = \emptyset$, go to Step 2.
4) Set $\Pi'(i) := \Pi_i(x(t), r(t))$. Perform the optimization in (31).
5) If the solution is not unique, set $r_i(t) = v_i(t-1) + \kappa_i(t)(r_i(t) - v_i(t-1))$. Increment $i := i + 1$, and go to Step 4.
6) Set $v(t) = v(t-1) + \mathbf{K}(t)(r(t) - v(t-1))$.

In this way, $\kappa_{1,\ldots,i}(t) = 1$ for the largest possible $1 \le i \le m$, while the PRG satisfies imposed constraints with the rest of the available $m - i$ command inputs.

1) Computational procedure: The PRG algorithm follows by first computing a half-space representation for O_∞ of the form (7). In this way,

$$\Pi_i(x) = \{v : H_{r,i}v_i + \cdots + H_{r,m}v_m \le h - H_x x - H_{r,1}r_1 - \cdots - H_{r,i-1}r_{i-1}, v_1 = r_1, \ldots, v_{i-1} = r_{i-1}\}. \quad (32)$$

Furthermore, as per (30), if we set $v_j(t) = v_j(t-1) + \kappa_j(t)(r_j(t) - v_j(t-1))$ for $j > i$, then we can define a set of feasible κ_i,

$$\boldsymbol{\kappa}(t) = \{\kappa : \alpha_i(t)\kappa_i + \cdots + \alpha_m(t)\kappa_m \le \beta(t), \kappa_1 = \cdots = \kappa_{i-1} = 1\}, \quad (33)$$

where,

$$\alpha_j(t) = H_{r,i}(r_i(t) - v_i(t-1)), \; j = i, \ldots, m, \quad (34)$$

$$\beta(t) = h - H_x x - H_{r,1}r_1(t) - \cdots - H_{r,i-1}r_{i-1}(t) - H_{r,i}v_i(t) - \cdots - H_{r,m}v_m(t). \quad (35)$$

The solution proceeds by finding the first set $\Pi_i(x(t))$ for which a solution exists, *i.e.* for which $\beta(t) \ge 0$, and then searching over all the vertices of $\boldsymbol{\kappa}(t)$ for the vertices with the highest value for $\kappa_i(t)$. If there exists more than one solution, then we recursively choose the vertex with the highest value for $\kappa_{i+j}(t)$ for $j = 2, 3, \ldots$ until there is only one solution for $\mathbf{K}(t)$.

A. Theoretical results

Under the condition that there exists a solution for $v(0)$ given the initial condition, $x(0)$, *i.e.* $\Pi(x(0)) \ne \emptyset$, the PRG, like the RG, guarantees constraint admissibility for all future time. Furthermore, it guarantees finite-time convergence to an admissible reference. We summarize this in the following proposition.

Proposition 2: The following holds for the PRG. (i) If $\Pi(x(0)) \ne \emptyset$ then there exists a sequence of admissible references, $v(t)$, such that $y(t) \in Y$ for all $t \in \mathbb{Z}_+$. (ii) Assume $r(t) \equiv r \in \Omega$, then there exists a t_f such that $v(t) = r(t)$, for all $t \ge t_f$.

Proof: Assume that for $t \ge 0$, $y(t+1) \in Y$ and $v(t) \in \Pi(x(t+1))$. Then for some i, $v(t+1) \in \Pi_i(x(t+1)) \subset \Pi(x(t+1))$, which implies that $(x(t+1), v(t+1)) \in O_\infty$ and that $y(t+1) \in Y$.

By assumption, there exists a $v(0) \in \Pi(x(0))$. This implies that $(x(0), v(0)) \in O_\infty$ and that $(x(1), v(0)) \in O_\infty$, implying that $y(1) \in Y$ and $v(0) \in \Pi(x(1))$.

Part (i) is proved by induction. By the definition of $\Pi(x(t))$, since there always exists an i such that $v(t) \in \Pi_i(x(t)) \subset \Pi(x(t))$, then $(x(t), v(t)) \in O_\infty$. This proves part (i).

Part (ii) is left to the journal version of this paper. ■

B. Numerical example

We apply the PRG to a linear model of an F-16 aircraft. The linear equations of motion are taken from [13], where the state, $x(t) = (\gamma(t), q(t), \alpha(t), \delta_e(t), \delta_f(t))$, consists of the flight path angle, pitch rate, angle of attack, elevator deflection, and flaperon deflection, respectively; the controls in $u(t) = (\delta_{ec}(t), \delta_{fc}(t))$ are commanded elevator and flaperon deflections; the output is $y(t) = (\theta(t), \gamma(t))$, where $\theta(t) = \gamma(t) + \alpha(t)$ is the pitch attitude; the reference input is $v(t) = (\theta_c(t), \gamma_c(t))$, where $\theta_c(t)$ is the commanded pitch attitude and $\gamma_c(t)$ is the commanded flight path angle. The system matrices can be found in [13].

The system is subject to constraints on the elevator and flaperon deflections where, in degrees,

$$-25 \le \delta_e \le 25, \quad (36)$$

$$-20 \le \delta_f \le 20. \quad (37)$$

If the controller signals violate these limits, the closed-loop system can easily become unstable as the open-loop system is unstable and the inputs will saturate.

We perform a zero-order hold on the system with time-step, $T = 0.01$, and perform two simulations starting from zero initial condition, $x(0) = 0$, with the desired reference values set constant at $(\theta_c(t), \gamma_c(t)) \equiv (11, 13.65)$, where the reference commands are given in degrees. The simulations are performed corresponding to two different prioritizations.

The first simulation is done by ordering the vector $r(t) = (\theta_c(t), \gamma_c(t))$ so that the pitch attitude is given higher priority than flight path angle. The second simulation is done *vice versa*, with $r(t) = (\gamma_c(t), \theta_c(t))$. The results are presented in Figs. 6-8 along with a plot of the ordinary RG response.

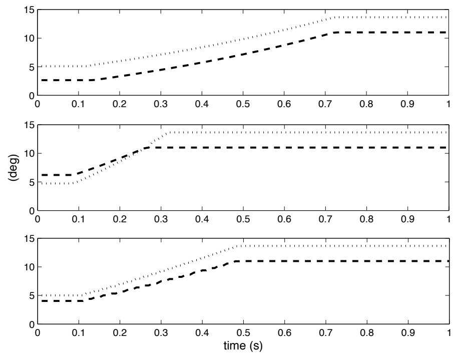

Fig. 6. Top subplot corresponds to the first prioritization ordering with $\theta_c(t)$ (dashed) and $\gamma_c(t)$ (dotted); the middle subplot corresponds to the second prioritization ordering; the bottom subplot corresponds to the ordinary RG

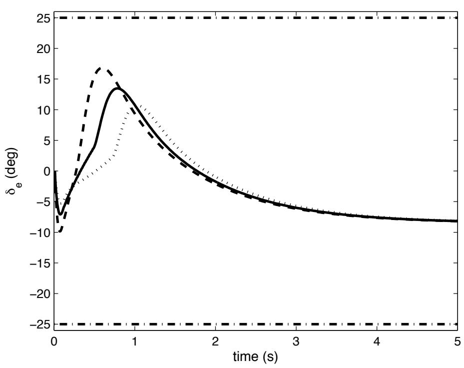

Fig. 7. Response of $\delta_e(t)$ corresponding to the first (dashed) and second (dotted) prioritization orderings compared against the ordinary RG (solid) with constraints marked as a dot-dashed line

The effect of prioritization is best shown with the use of Fig. 6. The figure first shows that, when $\gamma_c(t)$ is prioritized, the initial value for $\gamma_c(t)$ is slightly higher than for the second prioritization. The same is true for $\theta_c(t)$ with the second prioritization. Fig. 6 also suggests that both references converge faster when $\theta_c(t)$ is prioritized, with only a modest reduction in the initial value of $\gamma_c(t)$. The response of the ordinary RG is between that of the other two subplots, due to the fact that the ordinary RG gives equal priority to both inputs.

Figs. 7-8 show the responses of $\delta_e(t)$ and $\delta_f(t)$ with respect to the two prioritizations. It is shown that the RG keeps the elevator and flaperon responses within system constraints; the results also show that only the flaperon constraint becomes active for both orders of priority, with slightly faster convergence rate when $\theta_c(t)$ is prioritized.

VI. Conclusion

This paper presents two strategies in applying the theories of the reference and command governors to constrained systems subject to prioritization. The first method is applied to soft constraints which are satisfied in order of priority

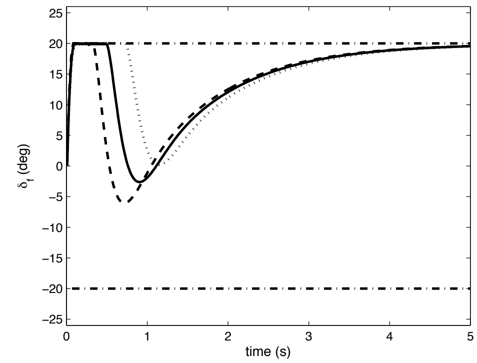

Fig. 8. Response of $\delta_f(t)$ corresponding to the first (dashed) and second (dotted) prioritization orderings compared against the ordinary RG (solid) with constraints marked as a dot-dashed line

by applying the CG to slack variables that are augmented to the constraints. The second method applies the RG to a prioritized list of inputs in order to maintain the set-point with highest priority as close as possible to its desired value. Theoretical results and simulations were also presented.

References

[1] E. G. Gilbert, I. V. Kolmanovsky, and K. T. Tan, "Discrete-time reference governors and the nonlinear control of systems with state and control constraints," *Int. J. Robust Nonlin.*, vol. 5, no. 5, pp. 487–504, 1995.

[2] A. Bemporad, "Reference governor for constrained nonlinear system," *IEEE Trans. Autom. Control*, vol. 43, no. 3, 1998.

[3] E. G. Gilbert and I. V. Kolmanovsky. "Fast reference governors for systems with state and control constraints and disturbance inputs," *Int. J. Robust Nonlin.*, vol. 9, no. 15, pp. 1117–1141, 1999.

[4] E. Gilbert and I. Kolmanovsky. "Nonlinear tracking control in the presence of state and control constraints: a generalized reference governor," *Automatica*, vol. 38, no. 12, pp. 2063–2073, 2002.

[5] A. Bemporad, A. Casavola, and E. Mosca, "Nonlinear control of constrained linear systems vie predictive reference management," *IEEE Trans. Autom. Control*, vol. 42, no. 3, pp. 340–349, 1997.

[6] A. Casavola, E. Mosca, and D. Angeli, "Robust command governors for constrained linear systems," *IEEE Trans. Autom. Control*, vol. 45, no. 11, pp. 2071–2077, 2000.

[7] E. G. Gilbert and C.-J. Ong, "Constrained linear systems with hard constraints and disturbances: An extended command governor with large domain of attraction," *Automatica*, vol. 47, no. 1, pp. 334–340, 2011.

[8] S. Di Cairano, D. Yanakiev, A. Bemporad, I. V. Kolmanovsky, and D. Hrovat, "Model predictive idle speed control: Design, analysis, and experimental validation," *IEEE Trans. Control Syst. Technol.*, vol. 20, no. 1, pp. 84–97, 2011.

[9] A. Weiss, I. V. Kolmanovsky, and W. Merill, "Incorporating risk into control design for emergency operation of turbo-fan engines," in *Proc. Infotech@Aerospace*, St. Louis, MO, 2011, AIAA-2011-1591.

[10] I. Kolmanovsky and E. G. Gilbert, "Theory and computation of disturbance invariant set of discrete-time linear systems," *Math. Probl. Eng.*, vol. 4, no. 7, pp. 317–367, 1998.

[11] R. Miller, I. V. Kolmanovsky, E. G. Gilbert, and P. Washabaugh, "Control of constrained nonlinear systems: A case study," *IEEE Control Syst. Mag.*, vol. 20, no. 1, pp. 23–32, 2000.

[12] U. Kalabić, I. Kolmanovsky, J. Buckland, and E. Gilbert, "Reference and extended command governors for control of turbocharged gasoline engines based on linear models," in *Proc. Multi-Conf. Syst. Control*, Denver, CO, 2011, pp. 319–325.

[13] K. M. Sobel and E. Y. Shapiro, "A design methodology for pitch pointing flight control systems," *J. Guid. Control Dynam.*, vol. 8, no. 2, pp. 181–187, 1985.