Journal of Guidance, Control, and Dynamics
Vol. 40, No. 2, February 2017

Engineering Notes

Fast Reference Governor for Linear Systems

Marco M. Nicotra* and Emanuele Garone†
Free University of Brussels, 1050 Brussels, Belgium
and
Ilya V. Kolmanovsky‡
University of Michigan, Ann Arbor, Michigan 48109

DOI: 10.2514/1.G000337

I. Introduction

REFERENCE governors (RGs) are add-on schemes that provide constraint-handling capabilities to existing closed-loop systems. This is done by modifying the reference of the prestabilized system to a safe value that will guarantee constraint satisfaction if maintained constant. Several RG schemes exist for linear [1], nonlinear [2], and uncertain systems [3]. The current state of the art can be found in a recently published survey [4]. Due to the relatively small computational requirements, reference governors are becoming increasingly popular in the aerospace domain. Examples include the constrained control of aircrafts [5,6], missiles [7,8], hypersonic vehicles [9,10], spacecraft [11], tethered lighter-than-air wind turbines [12], helicopters [13], and quadrotors [14,15].

This Note proposes a simple implementation algorithm that reduces the computational footprint of the scalar RG for linear closed-loop systems subject to linear, quadratic, and/or convex constraints. The main idea of the proposed scheme, denoted hereafter as the fast reference governor (FARG), is to order the constraints based on growing complexity and to use closed-form rules to progressively update the solution only if necessary.

The performance of the proposed algorithm is compared to the well-established bisection-based algorithm [16,17], as well as the recently introduced explicit RG [18]. Numerical simulations illustrating the constrained control of an F-16 aircraft show that the proposed algorithm represents an attractive option for constraint enforcement in the aerospace domain.

II. Scalar Reference Governor

This Note introduces a novel computational method for the scalar RG presented in [1,3]. For the sake of completeness, this section provides a brief summary of the theoretical framework.

Consider a discrete-time linear system in the form

$$x_{t+1} = Ax_t + Bv_t \tag{1}$$

where $x \in \mathbb{R}^n$, $v \in \mathbb{R}^p$, and A is a Schur matrix. The system is subject to J convex constraints

$$\mathcal{C}_j(x_t, v_t) \leq 0 \qquad \begin{array}{l} \forall\, j = 1, \ldots, J \\ \forall\, t \geq 0 \end{array} \tag{2}$$

which reflect both state and input constraints, since system (1) represents a closed-loop system and v_t is the set point of the primary controller.

Given a desired reference r_t and a previously applied reference v_{t-1} that, if kept constant, ensures constraint satisfaction over the infinite horizon, the scalar reference governor assigns v_t using the linear interpolation

$$v_t = (1-\lambda)v_{t-1} + \lambda r_t \tag{3}$$

with $\lambda \in [0, 1]$. The idea behind Eq. (3) is that $\lambda = 1$ implies $v_t = r_t$, whereas $\lambda = 0$ implies $v_t = v_{t-1}$ and guarantees constraint satisfaction for all future instants. The value of λ can then be chosen by solving the scalar convex optimization problem

$$\begin{aligned} &\max\ \lambda \\ &\text{subject to} \\ &\begin{cases} 0 \leq \lambda \leq 1 \\ \mathcal{C}_{j\tau}(\lambda) \leq 0 & j = 1, \ldots, J, \quad \tau = 0, \ldots, T \\ \mathcal{C}_{j\infty}(\lambda) + \delta_j \leq 0 & j = 1, \ldots, J \end{cases} \end{aligned} \tag{4}$$

where

$$\mathcal{C}_{j\tau}(\lambda) = \mathcal{C}_j(\hat{x}_{t+\tau|t}, v_t) \tag{5}$$

with

$$\hat{x}_{t+\tau|t} = \mathcal{A}_\tau x_t + \mathcal{R}_\tau v_t \tag{6}$$

$$\mathcal{A}_\tau = A^\tau, \qquad \mathcal{R}_\tau = \sum_{i=0}^{\tau-1} A^i B$$

represents the predicted constraint values if $v_{t+\tau} = v_t$, $\forall\, \tau \geq 0$, whereas the term

$$\mathcal{C}_{j\infty}(\lambda) = \mathcal{C}_j(\hat{x}_{\infty|t}, v_t) \tag{7}$$

with

$$\hat{x}_{\infty|t} = (I_n - A)^{-1} B v_t, \tag{8}$$

represents the steady-state constraint value if $v_{t+\tau} = v_t$, $\forall\, \tau \geq 0$. Given a prediction horizon T and suitable steady-state safety margins δ_j, the optimization problem [Eq. (4)] guarantees that $\lambda = 0$ remains a feasible solution for all future time steps. See [1,3] for further details.

The scalar optimization problem [Eq. (4)] is typically solved using a bisection algorithm. This Note proposes an alternative approach that significantly reduces the computational cost by rearranging the constraints based on their complexity (linear, quadratic, general convex), and by sequentially handling the restrictions on λ to avoid unnecessary computations.

Received 6 December 2015; revision received 24 May 2016; accepted for publication 16 August 2016; published online 14 December 2016.

*Ph.D. Student, Department of Control Engineering and System Analysis; mnicotra@ulb.ac.be.

†Associate Professor, Department of Control Engineering and System Analysis; egaroene@ulb.ac.be.

‡Professor, Department of Aerospace Engineering; ilya@umich.edu.

III. Linear and Quadratic Constraints

Consider the case where the first J_q constraints are in the form

$$\begin{bmatrix} x_t \\ v_t \end{bmatrix}^T \begin{bmatrix} Q_{xj} & P_j \\ P_j^T & Q_{vj} \end{bmatrix} \begin{bmatrix} x_t \\ v_t \end{bmatrix} + \begin{bmatrix} L_{xj} & L_{vj} \end{bmatrix} \begin{bmatrix} x_t \\ v_t \end{bmatrix} + h_j \leq 0 \tag{9}$$

with

$$\begin{bmatrix} Q_{xj} & P_j \\ P_j^T & Q_{vj} \end{bmatrix} \geq 0 \tag{10}$$

Taking into account Eqs. (5) and (6), recursive constraint satisfaction is ensured if

$$\begin{bmatrix} x_t \\ v_t \end{bmatrix}^T \mathcal{Q}_{j\tau} \begin{bmatrix} x_t \\ v_t \end{bmatrix} + \mathcal{L}_{j\tau} \begin{bmatrix} x_t \\ v_t \end{bmatrix} + h_j \leq 0 \tag{11}$$

with

$$\mathcal{Q}_{j\tau} = \begin{bmatrix} \mathcal{A}_\tau^T Q_{xj} \mathcal{A}_\tau & \mathcal{A}_\tau^T (Q_{xj} \mathcal{R}_\tau + P_j) \\ (Q_{xj}\mathcal{R}_\tau + P_j)^T \mathcal{A}_\tau & \mathcal{R}_\tau^T Q_{xj} \mathcal{R}_\tau + \mathcal{R}_\tau^T P_j + P_j^T \mathcal{R}_\tau + Q_{vj} \end{bmatrix}$$

$$\mathcal{L}_{j\tau} = \begin{bmatrix} L_{xj}\mathcal{A}_\tau & L_{xj}\mathcal{R}_\tau + L_{vj} \end{bmatrix}$$

By substituting Eq. (3), the inequality [Eq. (11)] becomes

$$a_{j\tau}\lambda^2 + b_{j\tau}\lambda + c_{j\tau} \leq 0$$

with

$$a_{j\tau} = \begin{bmatrix} 0 \\ (r_t - v_{t-1}) \end{bmatrix}^T \mathcal{Q}_{j\tau} \begin{bmatrix} 0 \\ (r_t - v_{t-1}) \end{bmatrix}$$
$$b_{j\tau} = \left(2\begin{bmatrix} x_t \\ v_{t-1} \end{bmatrix}^T \mathcal{Q}_{j\tau} + \mathcal{L}_{j\tau}\right) \begin{bmatrix} 0 \\ (r_t - v_{t-1}) \end{bmatrix}$$
$$c_{j\tau} = \left(\begin{bmatrix} x_t \\ v_{t-1} \end{bmatrix}^T \mathcal{Q}_{j\tau} + \mathcal{L}_{j\tau}\right) \begin{bmatrix} x_t \\ v_{t-1} \end{bmatrix} + h_j \tag{12}$$

The solution to the optimization problem [Eq. (4)] can therefore be found by computing $\lambda_{j\tau}$, satisfying

$$a_{j\tau}\lambda_{j\tau}^2 + b_{j\tau}\lambda_{j\tau} + c_{j\tau} = 0$$

and then choosing $\lambda = \min(\lambda_{j\tau})$. The following proposition illustrates how to avoid the computation of $\lambda_{j\tau}$ whenever possible.

Proposition 1: Let the current state x_t and the previous reference v_{t-1} of system (1) be such that $c_{jt} \leq 0$ for $j = 1, \ldots, J_l$, $\tau \geq 0$. Let $\hat{\lambda} \geq 0$ be the solution of the optimization problem

$$\max \lambda$$
subject to
$$\begin{cases} \lambda \leq 1; & \\ a_{j\tau}\lambda^2 + b_{j\tau}\lambda + c_{j\tau} \leq 0, & \begin{matrix} \tau = 0, \ldots, \hat{\tau}-1, & j = 1, \ldots, J_q \\ \tau = \hat{\tau}, & j = 1, \ldots, \hat{j}-1 \end{matrix} \end{cases} \tag{13}$$

then the following is true:

1) If Eq. (11) holds true for $\hat{j}$, $\hat{\tau}$, then $\hat{\lambda}$ is also the solution to

$$\max \lambda$$
subject to
$$\begin{cases} \lambda \leq 1; & \\ a_{j\tau}\lambda^2 + b_{j\tau}\lambda + c_{j\tau} \leq 0, & \begin{matrix} \tau = 0, \ldots, \hat{\tau}-1, & j = 1, \ldots, J_q \\ \tau = \hat{\tau}, & j = 1, \ldots, \hat{j} \end{matrix} \end{cases} \tag{14}$$

2) If Eq. (11) is false for $\hat{j}$, $\hat{\tau}$ and $a_{\hat{j}\hat{\tau}} = 0$, the solution of the optimization problem [Eq. (14)] is

$$\lambda_{\hat{j}\hat{\tau}} = -\frac{c_{\hat{j}\hat{\tau}}}{b_{\hat{j}\hat{\tau}}} \tag{15}$$

3) If Eq. (11) is false for $\hat{j}$, $\hat{\tau}$ and $a_{\hat{j}\hat{\tau}} \neq 0$, the solution of the optimization problem [Eq. (14)] is

$$\lambda_{\hat{j}\hat{\tau}} = \sqrt{\frac{b_{\hat{j}\hat{\tau}}^2}{a_{\hat{j}\hat{\tau}}^2} - \frac{c_{\hat{j}\hat{\tau}}}{a_{\hat{j}\hat{\tau}}}} - \left|\frac{b_{\hat{j}\hat{\tau}}}{a_{\hat{j}\hat{\tau}}}\right| \tag{16}$$

Proof: The three parts will be proven separately.

Part 1: If Eq. (11) holds true for $\hat{j}$, $\hat{\tau}$, the additional constraint in Eq. (14) is redundant with respect to the constraints that are already accounted for in Eq. (13). As such, $\hat{\lambda}$ remains the maximum admissible value.

Part 2: Since $c_{\hat{j}\hat{\tau}} \leq 0$ and $\hat{\lambda} \geq 0$ by hypothesis, Eq. (11) can only be violated for $b_{\hat{j}\hat{\tau}} > 0$. This ensures $0 \leq \lambda_{\hat{j}\hat{\tau}} < \hat{\lambda}$. If Eq. (11) is false, problem (14) can be simplified into

$$\max \lambda$$
subject to
$$b_{\hat{j}\hat{\tau}}\lambda + c_{\hat{j}\hat{\tau}} \leq 0$$

because all the other constraints are redundant. As such, the analytic solution is Eq. (15).

Part 3: If Eq. (11) is false, problem (14) can be simplified into

$$\max \lambda$$
subject to
$$a_{\hat{j}\hat{t}}\lambda^2 + b_{\hat{j}\hat{t}}\lambda + c_{\hat{j}\hat{t}} \leq 0$$

because all the other constraints are redundant. The maximizer can therefore be found by solving the quadratic equation

$$a_{\hat{j}\hat{t}}\lambda^2 + b_{\hat{j}\hat{t}}\lambda + c_{\hat{j}\hat{t}} = 0 \tag{17}$$

Due to conditions (10) and $c_{\hat{j}\hat{\tau}} < 0$, Eq. (17) admits two real solutions: one positive and one negative. The positive solution is Eq. (16). □

Following from Proposition 1, the optimization problem can be solved sequentially by initializing $\hat{\lambda} = 1$ and updating its value if, and only if, the currently selected $\hat{\lambda}$ would violate constraint $\hat{j}$ at time $\hat{\tau}$. The following section extends this result by considering arbitrary convex constraints.

IV. Arbitrary Convex Constraints

The following proposition addresses the case where the first J_q constraints are in the form of Eq. (9) and the remaining constraints are convex.

Proposition 2: Given system (1) subject to constraints (2), let the current state x_t and the previous reference v_{t-1} be such that

$$\mathcal{C}_j(\mathcal{A}_\tau x_t + \mathcal{R}_\tau v_{t-1}, v_{t-1}) \leq 0 \quad \begin{matrix} \forall\, j = 1, \ldots, J \\ \forall\, \tau \geq 0 \end{matrix} \tag{18}$$

Given $\hat{\lambda} \geq 0$ equal to the solution of the optimization problem

$$\max \lambda \quad \text{such that}$$
$$\begin{cases} \lambda \le 1; \\ a_{j\tau}\lambda^2 + b_{j\tau}\lambda + c_{j\tau} \le 0, & \tau = 0,\dots,T \quad j = 1,\dots,J_q \\ \mathcal{C}_{j\tau}(\lambda) \le 0, & \tau = 0,\dots,\hat{\tau}-1, \quad j = J_q+1,\dots,J \\ \mathcal{C}_{j\hat{\tau}}(\lambda) \le 0, & j = J_q+1,\dots, \quad \hat{j}-1 \end{cases} \tag{19}$$

where $\mathcal{C}_{jt}(\lambda)$ is given by Eqs. (5) and (6), then the following is true:

1) If $\mathcal{C}_{\hat{j}\hat{\tau}}(\hat{\lambda}) \le 0$, $\hat{\lambda}$ is also the solution to

$$\max \lambda \quad \text{such that}$$
$$\begin{cases} \lambda \le 1; \\ a_{j\tau}\lambda^2 + b_{j\tau}\lambda + c_{j\tau} \le 0, & \tau = 0,\dots,T, \quad j = 1,\dots,J_q \\ \mathcal{C}_{j\tau}(\lambda) \le 0, & \tau = 0,\dots,\hat{\tau}-1, \quad j = J_q+1,\dots,J \\ \mathcal{C}_{j\hat{\tau}}(\lambda) \le 0, & j = J_q+1,\dots,\hat{j}, \end{cases} \tag{20}$$

2) If $\mathcal{C}_{\hat{j}\hat{\tau}}(\hat{\lambda}) > 0$, the solution of the optimization problem [Eq. (20)] is $\lambda_{\hat{j}\hat{\tau}} \ge 0$, satisfying

$$\mathcal{C}_{\hat{j}\hat{\tau}}(\lambda_{\hat{j}\hat{\tau}}) = 0 \tag{21}$$

Proof: The proof is analogous to the one provided for Proposition 1. It is worth noting that Eq. (21) always admits a solution

$$\lambda_{\hat{j}\hat{\tau}} \in [0, \hat{\lambda})$$

since $\mathcal{C}_{\hat{j},\hat{\tau}}(0) \le 0$ and $\mathcal{C}_{\hat{j},\hat{\tau}}(\hat{\lambda}) > 0$. □

In analogy to Proposition 1, Proposition 2 provides a systematic way to avoid unnecessary computations by checking if the currently selected $\hat{\lambda}$ satisfies constraint $\hat{j}$ at time $\hat{\tau}$. Because the value of $\hat{\lambda} \in [0,1]$ becomes increasingly restrictive, it is preferable to prioritize the constraints for which $\lambda_{\hat{j}\hat{t}}$ can be computed efficiently. This will reduce the probability of having to solve the more onerous constraints without changing the final result.

V. Handling Strict Steady-State Feasibility Constraints

The steady-state constraints based on Eq. (7) can be treated as in the previous two sections by sequentially checking if the currently selected $\hat{\lambda}$ asymptotically satisfies each constraint. For quadratic constraints, the update $\hat{\lambda}$ is the same as Sec. III, with

$$\mathcal{Q}_{j\infty} = \left(B^T(I_n - A)^{-T}Q_{xj} + 2P_j^T\right)(I_n - A)^{-1}B + Q_{vj}$$
$$\mathcal{L}_{j\infty} = L_{xj}(I_n - A)^{-1}B + L_{vj}$$

As for the remaining convex constraints, the update of $\hat{\lambda}$ must be obtained by solving Eqs. (7) and (8). As a general rule, it is usually preferable to check the conditions on the steady-state outputs [Eq. (7)] before addressing the constraints on the predicted trajectory.

VI. Implementation

The main idea behind the proposed algorithm is to solve the optimization problem [Eq. (4)] by performing as little computations as possible. Based on all the previous considerations, the following algorithm is proposed:

1) Initialize the algorithm with $\hat{\lambda} = 1$;

2) Update $\hat{\lambda}$ by checking the steady-state constraints [Eq. (7)] for $j = 1, \dots, J_q$;

3) Update $\hat{\lambda}$ by checking the constraints on the trajectory predictions [Eq. (11)] for $j = 1, \dots, J_q$, $\tau = 1, \dots, T$. For computational simplicity (i.e., to compute $\mathcal{A}_\tau$, $\mathcal{R}_\tau$ only once), it is preferable to check all constraints j before moving on to the next time instant τ. Moreover, for a given time instant, it is preferable to check the linear constraints before the quadratic ones;

4) Update $\hat{\lambda}$ by checking the steady-state constraints [Eq. (7)] for $j = J_q + 1, \dots, J$. The convex constraints should be ordered by growing complexity;

5) Update $\hat{\lambda}$ by checking the constraints on the trajectory predictions [Eq. (5)] for $j = J_q + 1, \dots, J$, $\tau = 1, \dots, T$. Careful consideration should be given to whether it is more efficient to check all constraints j at the same time instant τ or if it is better to delay more difficult constraints to the very end.

In the best-case scenario (i.e., when the desired reference $\hat{\lambda} = 1$ is feasible), the fast reference governor will only perform $J \times (T+1)$ checks to verify constraint satisfaction. In the worst-case scenario (i.e., constraints are violated at every check), the fast reference governor will perform $J \times (T+1)$ checks, solve $J_q \times (T+1)$ first- or second-degree equations, and solve $(J - J_q) \times (T+1)$ convex root searches. By comparison, the bisection approach used by classic reference governors has a computational cost equivalent to $J \times (T+1)$ convex root searches.

As such, the proposed algorithm will outperform the standard method even in the worst-case scenario by providing a closed-form solution for all the linear and quadratic constraints. Moreover, because the number of updates will typically be much smaller than $J \times (T+1)$, the performances will typically be closer to the best-case scenario.

The following numerical example compares the behavior of the FARG to the bisection-based RG [16] and the explicit RG [18].

VII. Numerical Example

The proposed method is applied to the flight-path control of an F-16 aircraft. The aircraft is open-loop unstable and can become closed-loop unstable in response to large command changes due to actuator saturation. The linearized system dynamics are described by the following state-space model [19]:

$$\dot{x} = Ax + Bv$$
$$y = Cx + Dv$$

where

$$A = \begin{bmatrix} 0 & 0.0067 & 1.341 & 0.169 & 0.252 \\ 0 & -0.869 & 43.2 & -17.25 & -1.577 \\ 0 & 0.993 & -1.341 & -0.169 & -0.252 \\ 65 & 17.82 & 142.3 & -30.5 & -1.68 \\ -122 & -17.95 & -200.6 & 8.412 & -17.89 \end{bmatrix}$$
$$B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ -57.6 & -7.34 \\ 40.4 & 81.6 \end{bmatrix}$$
$$c = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 65 & 17.82 & 142.3 & -30.5 & -1.68 \\ -122 & -17.95 & -200.6 & 8.412 & -17.89 \end{bmatrix}$$
$$D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ -57.6 & -7.34 \\ 40.4 & 81.6 \end{bmatrix}$$

The state vector $x = [\gamma, q, \alpha, \delta_e, \delta_f]^T$ contains the flight-path angle, the pitch rate, the angle of attack, the elevator deflection, and the flaperon deflection. The input vector $v = [\theta_c, \gamma_c]^T$ consists of the desired pitch angle $\theta_C = \gamma_C + \alpha_C$ and the desired flight-path angle γ_C. The output vector on which constraints are imposed, $y = [\alpha, \delta_e, \delta_f, \dot{\delta}_e, \dot{\delta}_f]^T$, reflects the angle of attack, the elevator deflection, the flaperon deflection, the elevator deflection rate, and the flaperon deflection rate. The system is subject to the set of linear constraints

$$|y_1| \le 4\ \text{deg}; \qquad |y_2| \le 25\ \text{deg}; \qquad |y_3| \le 20\ \text{deg};$$
$$|y_4| \le 42\ \text{deg/s}; \qquad |y_5| \le 56\ \text{deg/s}$$

The FARG is implemented with a sampling period of $\Delta T = 0.02$ s and a prediction horizon of $T = 50$ steps. The results are compared with a standard (i.e., bisection-based) RG (STDRG) with a precision of 2^{-7} (i.e. <1%) and an explicit RG (ERG) [18] with a gain of $\kappa = 10^3$. Figure 1 provides the output step response obtained using the three different RG schemes, all of which successfully enforce the system constraints. All three simulations are performed on MATLAB on a Notebook running an Intel core i7-3610QM processor with a clock rate of 2.30 GHz.

As shown in Fig. 1, the closed-loop behaviors of the STDRG and the FARG are practically indistinguishable. This is because both schemes compute v_t by solving the optimization problem [Eq. (4)]. However, the FARG provides the exact analytic solution to Eq. (4), whereas the STDRG provides a solution within the selected tolerance margin. Although this numerical error is typically negligible in terms of the output response, its presence leads to small spikes on the control input, as illustrated in Fig. 2.

For what concerns the computational footprint, in this example, the fast RG runtime is approximately three times faster than the standard RG (≈0.12 ms versus ≈0.35 ms). As discussed in the previous section, this disparity will scale with the number of constraints and the simulation horizon. In other terms, the FARG represents an overall improvement with respect to classic bisection-based schemes, both in terms of smoother control inputs and lower computational requirements.

For what regards the ERG, Fig. 1 clearly shows that the FARG achieves better closed-loop performances. This is due to the fact that

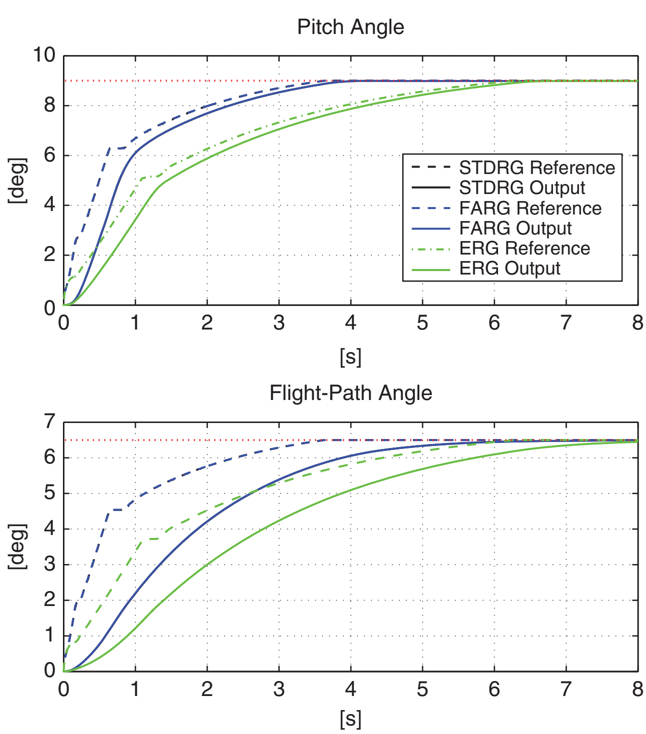

Fig. 1 Desired reference, applied reference, and actual value of pitch angle θ and flight-path angle γ.

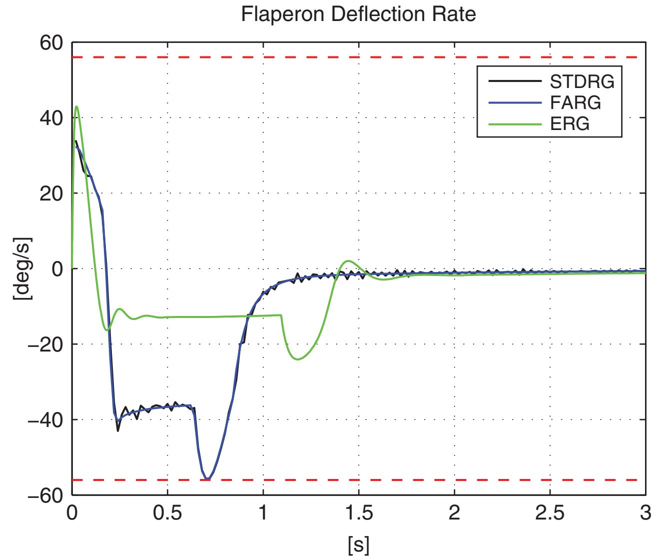

Fig. 2 Control signal generated by the bisection-based RG (STDRG) the Explicit RG (ERG), and the fast RG (FARG).

the ERG is an invariance-based scheme instead of an optimization-based scheme. This loss of performance is compensated by the fact that the ERG does not need to compute the system trajectories. As such, it is not surprising that the FARG is approximately five times slower with respect to the ERG (≈0.12 ms versus ≈0.025 ms).

This is consistent with the pre-established tradeoff [18] between ERG and optimization-based RG schemes (which include the FARG). The choice between the two methods will therefore depend on whether the application prioritizes high performances, in which case the FARG would be preferable, or low computational requirements, in which case the ERG remains the better option.

VIII. Conclusions

A fast reference governor algorithm was derived for constraint enforcement based on linear discrete-time system models with linear, quadratic, or arbitrary convex constraints. Comparison with other state-of-the-art RG schemes showed that the proposed method was an attractive option for constraint enforcement in aerospace systems.

References

[1] Gilbert, E., Kolmanovsky, I. V., and Tan, K., "Discrete-Time Reference Governors and the Nonlinear Control of Systems with State and Control Constraints," *International Journal of Robust and Nonlinear Control*, Vol. 5, No. 5, 1995, pp. 487–504. doi:10.1002/(ISSN)1099-1239

[2] Bemporad, A., "Reference Governor for Constrained Nonlinear Systems," *IEEE Transactions on Automatic Control*, Vol. 43, No. 3, 1998, pp. 415–419. doi:10.1109/9.661611

[3] Gilbert, E., and Kolmanovsky, I. V., "Fast Reference Governors for Systems with State and Control Constraints and Disturbance Inputs," *International Journal of Robust and Nonlinear Control*, Vol. 9, No. 15, 1999, pp. 1117–1141. doi:10.1002/(ISSN)1099-1239

[4] Kolmanovsky, I. V., Garone, E., and Di Cairano, S., "Reference and Command Governors: A Tutorial on Their Theory and Automotive Applications," *Proceedings of American Control Conference (ACC)*, IEEE Xplore, 2014, pp. 226–241. doi:10.1109/ACC.2014.6859176

[5] Franze, G., Mattei, M., Ollio, L., Scordamaglia, V., and Tedesco, F., "A Reconfigurable Aircraft Control Scheme Based on an Hybrid Command Governor Supervisory Approach," *Proceedings of American Control Conference*, IEEE Xplore, 2014, pp. 1273–1278. doi:10.1109/ACC.2014.6859230

[6] Weber, P., Boussaid, B., Khelassi, A., Theilliol, D., and Aubrun, C., "Reconfigurable Control Design with Integration of Reference Governor and Reliability Indicators," *International Journal of Applied*

Mathematics and Computer Science, Vol. 22, No. 1, 2012, pp. 139–148. doi:10.2478/v10006-012-0010-0

[7] Rodriguez, A. A., and Wang, Y., "Performance Enhancement Methods for Unstable Bank-to-Turn (BTT) Missiles with Saturating Actuators," *International Journal of Control*, Vol. 63, No. 4, 1996, pp. 641–678. doi:10.1080/00207179608921862

[8] Casavola, A., and Mosca, E., "Bank-to-Turn Missile Autpilot Design via Observer-Based Command Governor," *Journal of Guidance, Control, and Dynamics*, Vol. 27, No. 4, 2004, pp. 705–710. doi:10.2514/1.11163

[9] Zinnecker, A., Serrani, A., and Bolender, M., "Combined Reference Governor and Anti-Windup Design for Constrained Hypersonic Vehicles Models," *AIAA Guidance, Navigation, and Control Conference*, AIAA Paper 2009-6283, 2009.

[10] Petersen, C., Baldwin, M., and Kolmanovsky, I. V., "Model Predictive Control Guidance with Extended Command Governor Inner-Loop Flight Control for Hypersonic Vehicles," *AIAA Guidance, Navigation, and Control Conference*, AIAA Paper 2013-5028, 2013.

[11] Bacconi, F., Mosca, E., and Casavola, A., "Hybrid Constrained Formation of Flying Control of Micro-Satellites," *IET Control Theory and Applications*, Vol. 1, No. 2, 2007, pp. 513–521. doi:10.1049/iet-cta:20050397

[12] Kalabic, U., Vermillion, C., and Kolmanovsky, I. V., "Reference Governor Design for Computationally Efficient Attitude and Tether Tension Constraint Enforcement on a Lighter than Air Wind Energy System," *Proceedings of European Control Conference*, IEEE Xplore, 2013, pp. 1004–1010.

[13] Hirata, K., and Minemura, H., "Experimental Evaluations of Reference Governor Control Schemes with Applications to the Constrained Control of RC Helicopters," *Proceedings of IEEE International Conference on Control Applications*, IEEE Xplore, 2004, pp. 855–859. doi:10.1109/CCA.2004.1387475

[14] Lucia, W., Sznaier, M., and Franze, G., "An Obstacle Avoidance and Motion Planning Command Governor Based Scheme: The Qball-X4 Quadrotor Case of Study," *Proceedings of IEEE Conference on Decision and Control (CDC)*, IEEE Xplore, 2014, pp. 6135–6140. doi:10.1109/CDC.2014.7040350

[15] Nicotra, M. M., Naldi, R., and Garone, E., "Taut Cable Control of a Tethered UAV," *19th IFAC World Congress*, Vol. 47, No. 3, Cape Town, South Africa, 2014, pp. 3190–3195.

[16] Kolmanovsky, I., Kalabic, U., and Gilbert, E. G., "Developments in Constrained Control Using Reference Governors," *Proceedings of IFAC Conference on Nonlinear Model Predictive Control*, Vol. 45, No. 17, 2012, pp. 282–290.

[17] Kalabic, U., Gupta, R., Di Cairano, S., Bloch, A., and Kolmanovsky, I. V., "Constrained Spacecraft Attitude Control on SO(3) Using Reference Governors and Nonlinear Model Predictive Control," *Proceedings of the American Control Conference*, IEEE Xplore, 2014, pp. 5586–5593. doi:10.1109/ACC.2014.6858865

[18] Nicotra, M. M., and Garone, E., "Explicit Reference Governor for Continuous Time Nonlinear Systems Subject to Convex Constraints," *Proceedings of the American Control Conference*, IEEE Xplore, 2015, pp. 4561–4566. doi:10.1109/ACC.2015.7172047

[19] Sobel, K. M., and Shapiro, E. Y., "A Design Methodology for Pitch Pointing Flight Control Systems," *Journal of Guidance, Control, and Dynamics*, Vol. 8, No. 2, 2013, pp. 181–187. doi:10.2514/3.19957