Reference and Command Governors: A Tutorial on Their Theory and Automotive Applications

Ilya Kolmanovsky 1 Emanuele Garone 2 Stefano Di Cairano 3

Abstract—This paper provides a tutorial overview of reference governors and command governors, which are add-on control schemes for reference supervision and constraint enforcement in closed-loop feedback control systems. The main approaches to the development of such schemes for linear and nonlinear systems are described. The treatment of unmeasured disturbances and parametric uncertainties is addressed. Generalizations to extended command governors, feedforward reference governors, reduced order reference governors, parameter governors, networked reference governors, decentralized reference governors, and virtual state governors are summarized. Examples of applications of these techniques to automotive systems are given. A comprehensive list of references is included. Comments comparing reference and command governor approaches with Model Predictive Control and input shaping, and on future directions in reference and command governor research are included.

I. INTRODUCTION

With the advances in control theory, many effective techniques have become available for the design of feedback control systems with the desired stability, performance and disturbance rejection properties. The interest in treating the requirements that have the form of pointwise-in-time state and control constraints has also been growing, in particular, in the automotive domain. Examples of constraints in practical systems include actuator magnitude and rate limits, and bounds imposed on process variables to ensure safe and efficient system operation. Controllers that achieve high performance in systems with constraints are typically nonlinear and are often (but not always) based on prediction and optimization.

A control designer faced with the task of satisfying the constraints has several choices. One route is to re-design the controller within the Model Predictive Control (MPC) framework [102], [98], [24], [93], [54]. Another route is to augment a well-designed nominal controller, that already achieves high performance for small signals, with constraint handling capability for larger signals and transients that have the potential to induce constraint violation. The second route may be attractive to practitioners interested in preserving an existing/legacy controller or concerned with computational effort, tuning complexity, stability and robustness certification, etc. Anti-windup compensation and the augmentation of Lyapunov controllers with barrier functions are examples of the second approach and so are the reference and command governors.

As its name suggests, the reference governor is an add-on scheme for enforcing pointwise-in-time state and control constraints by modifying the reference command to a well-designed (for small signals) closed-loop system. See Figure 1. Numerous reference governor like schemes have been proposed. The range of potential options includes scalar and vector reference governors, command governors, extended command governors, incremental reference governors, feedforward reference governors, network reference governors, reduced order reference governors and others. The common intent of these governors is to preserve, whenever possible, the response of the closed loop system designed by conventional control techniques. Frequently (but not always), they achieve this by ensuring that the modified reference command is as close as possible to the original reference command subject to satisfying the constraints.

The scalar reference governor is attractive as it leads to computationally simple implementations for both linear and nonlinear systems with disturbances and parameter uncertainties. Other reference/command governor like schemes are more complex but provide better performance or address special problems, e.g., networked implementation for systems with communication delays and data drops, or decentralized and reduced order implementation for large scale systems.

The conventional scalar reference governors, vector reference governors, command governors and extended command governors generate the applied reference \( v(t) \) in Figure 1 based on the current value of the desired reference, \( r(t) \), the state estimate, \( \hat{x}(t) \), and, in the case of the extended command governor, based also on an \( \bar{n} \)-vector state \( \bar{x} \), of a supplementary and fictitious dynamic system.


*This work was supported in part by the National Science Foundation Award Number 1130160

1 Ilya Kolmanovsky (corresponding author) is with Department of Aerospace Engineering, The University of Michigan, Ann Arbor, MI. Email: ilya@umich.edu
2 Emanuele Garone is with Ecole Polytechnique de Bruxelles, Université Libre de Bruxelles Email: egarone@ulb.ac.be
3 Stefano Di Cairano is Mitsubishi Electric Research Laboratory, Cambridge, MA. Email: dicairano@ieee.org. This work has not been supported by Mitsubishi Electric or any of its subsidiaries.
[48], [51] has emerged that has several advantages from the implementation standpoint. The static reference governor [48] used \( v(t) = \tilde{\beta}(t)r(t), \) where the parameter \( \tilde{\beta}(t), 0 \leq \tilde{\beta}(t) \leq 1, \) was maximized subject to conditions that guaranteed the constraint enforcement. Specifically, \( x(t+1) \in O_{\infty}, \) where \( O_{\infty} \) is the maximum output admissible set [47] of all states that with zero reference command do not lead to subsequent constraint violation. Because of the possibility of oscillations [48], the static reference governor was replaced by a dynamic reference governor (see Section II) for which finite-time convergence for constant or nearly constant reference commands was established. Other formulations of reference and command governors have appeared in [10], [11], [49], [12], [25]. See also references therein. The developments included the treatment of linear systems with uncertainties and set-bounded disturbance inputs, and the implementation based on non-positively invariant sets. Extended command governors [52] represented a further generalization with a potential to provide a larger constrained domain of attraction and faster response at the price of increased computational complexity.

For nonlinear systems, approaches to reference governor design have been also developed, see e.g., [8], [50], [101], [53], [17] and references therein. Some of these approaches exploit on-line prediction through simulations or level sets of Lyapunov functions to guard against constraint violation. The robust reference governor that handles parametric uncertainty based on response approximations has been presented in [123]. The parameter governor has been proposed in [84] to adjust constant controller parameters or controller states based on prediction and optimization.

More recently, classical reference and command governor ideas have been extended in several additional directions. These include the treatment of networked systems and large scale systems. Related schemes such as parameter governors and virtual state governors have also emerged.

Automotive control systems are designed to satisfy stringent and numerous fuel economy, emissions, safety, performance and drivability requirements [58]. Their traditional implementation involves a family of hierarchically arranged real-time control algorithms, where reference commands from an upper layer controller are passed to a lower level controller, with the highest level commands corresponding to the driver and the road input. The on-board computing power in automotive systems (ROM, RAM and chronometrics) is very limited and successful control solutions must minimize the computational footprint. In addition, these solutions must be easy to understand, modify and calibrate. With the trend towards growing use of model-based control in the automotive industry and increasing importance of handling limits, reference and command governors represent an attractive option for several automotive applications where by supervision and minimum modification of reference commands applied to lower level controllers various constraints can be enforced.

In this paper we survey several basic and more recent reference governor results. We then discuss some of the research on applications of reference governors to automotive control problems. Comments on connections with MPC and on directions for future research are discussed at the end.

II. REFERENCE GOVERNORS FOR LINEAR SYSTEMS

The classical reference governor is designed for a discrete-time linear system model of the form,

\[
\begin{align*}
 x(t+1) &= Ax(t) + Bu(t) + B_ww(t), \\
 y(t) &= Cx(t) + Du(t) + D_ww(t),
\end{align*}
\]

where \( x \) is an \( n \)-vector state, \( v \) is an \( m \)-vector input representing the command (reference) applied to the system, \( w \) is an \( l \)-vector disturbance, and \( y \) is a \( p \)-vector system output. The disturbance \( w(t) \) is unmeasured but has known bounds specified as \( w(t) \in W \) for all \( t \in Z^+ \), where \( W \) is a given compact set and \( Z^+ \) denotes the set of non-negative integers.

Typically, the model (1) represents the closed-loop system thereby reflecting the combined closed-loop dynamics of the plant and of the controller in Figure 1. As normally the closed-loop system is designed to be asymptotically stable, the matrix \( A \) is assumed to be a Schur matrix (all eigenvalues are in the interior of the unit disk).

The constraints are imposed on the output variables, \( y(t) \), and have the form,

\[
y(t) \in Y \text{ for all } t \in Z^+, \tag{2}
\]

where \( Y \subset \mathbb{R}^p \) is a prescribed set, with \( 0 \in intY \). While not required, for computational reasons constraints are often defined so that \( Y \) is a polytope (compact set defined by a set of linear inequalities).

Since (1) is a model of the closed-loop system, (2) can represent constraints on either state or control variables. For instance, a control constraint \( |u_1(t)| \leq 1 \) where the control is generated by a state feedback law, \( u = Kx \), can be restated as \( y_1(t) \leq 1, y_2(t) \leq 1 \) with \( y_1 = \hat{e}_1Kx, y_2 = -\hat{e}_1Kx \), and \( \hat{e}_1 = [1 \ 0 \ \cdots 0] \).

The objective of the reference governor is to manage the applied reference, \( v(t) \), in such a way that \( v(t) \) is as close as possible to the desired reference, \( r(t) \), and the constraints are enforced. The desired reference, \( r(t) \), is assumed to be unknown in advance, representing an input from human operator or a higher level controller.

To do so, the Scalar Reference Governor (SRG) uses the following update [48], [51], [49]:

\[
v(t) = v(t-1) + \beta(t)(r(t) - v(t-1)), \tag{3}
\]

where \( \beta(t) \) is a scalar adjustable bandwidth parameter, \( 0 \leq \beta(t) \leq 1 \). If no danger of constraint violation exists, \( \beta(t) = 1 \), and \( v(t) = r(t) \) so that the reference governor does not interfere with the operation of the system. If a potential for constraint violation exists, the value of \( \beta(t) \) is decreased by the reference governor. In the extreme case, \( \beta(t) = 0, v(t) = v(t-1) \) so that the reference governor isolates the system from further application of reference command to ensure safety.

To assure safety, the following constraint is enforced,

\[
(v(t), x(t)) \in P, \tag{4}
\]
where $P \subset O_\infty \subset \mathbb{R}^m \times \mathbb{R}^n$. The set $O_\infty$ is the maximum output admissible set, i.e., the set of all states $x(t)$ and constant inputs $\bar{v}$ such that, if $\bar{v}$ is constantly applied to the system $(v(t+k) = \bar{v}, k \geq 0)$, the response will not violate constraints for any possible realization of the disturbance $w(t+k) \in W$,  
\begin{equation}
O_\infty = \{ (\bar{v}, x(t)) : \ y(t+k) \in Y \ \forall w(t+k) \in W, \ v(t+k) = \bar{v}, \ \forall k \in \mathbb{Z}^+ \} ,
\end{equation}
The choice of $P = \hat{O}_\infty$, where $\hat{O}_\infty$ is a slightly tightened version of $O_\infty$ is frequently made in (4), where
\begin{equation}
\hat{O}_\infty = O_\infty \cap \hat{O}_\infty^c
\end{equation}
and $\hat{O}_\infty^c$ is the set of commands such that the associated steady state constrained output $(D+C(I-A)^{-1}B)\bar{v}$ satisfies constraints with a margin $\epsilon > 0$ (typically small),  
\begin{equation}
\hat{O}_\infty^c = \{ (\bar{v}, x(t)) : \ (D+C(I-A)^{-1}B)\bar{v} \in (1-\epsilon)Y \} .
\end{equation}
Assuming that $Y$ is a polytope, the pair $(C,A)$ is observable and the minimum invariant set for (1) with $v = 0$ is strictly constraint admissible, i.e.,
\begin{equation}
CF + DW \subset intY, \ \ F = \bigoplus_{k=0}^{\infty} \mathbb{R}^k BW,
\end{equation}
where $\bigoplus$ denotes the Minkowski set sum [16], it can be shown that $O_\infty$ is non-empty, positively-invariant (with $v(t)$ maintained constant) and finitely-determined polytope, representable by a finite set of linear inequalities of the form,
\begin{equation}
\hat{O}_\infty = \{ (x(0), v) : \ H_k x(0) + H_v v \leq s \} .
\end{equation}
The set $\hat{O}_\infty$ is a finitely-determined inner approximation of $O_\infty$ which can be made arbitrary close to $O_\infty$ by decreasing $\epsilon$. Procedures for computing $\hat{O}_\infty$ are detailed in [47] and [83].

With $\hat{O}_\infty$ computed off-line as in (8) and using $P = \hat{O}_\infty$ in (4), the selection strategy for $\beta(t)$ at the time instant $t$ consists of maximizing $\beta(t)$, subject to the constraints
\begin{equation}
0 \leq \beta(t) \leq 1,
\end{equation}
and
\begin{equation}
(v(t), x(t)) \in \hat{O}_\infty.
\end{equation}
Due to the positive invariance of $\hat{O}_\infty$, recursive feasibility is maintained at each time step: the value of $\beta(t) = 0$ remains a feasible solution of the above optimization problem provided it is feasible at the initial time. Since only the scalar parameter $\beta(t)$ is optimized on-line, the computational complexity of this approach is very low and, in fact, the optimization problem is explicitly solvable, see [49] for details. If the system model or the constraints change, $H_k, H_v$ and $s$ can be computed on-line.

To further illustrate the conditions on $\beta(t)$, consider the case $W = \{ 0 \}$, $Y = \{ y \in \mathbb{R}^p : \ Sy \leq s \}$. The condition $(v(t), x(t)) \in \hat{O}_\infty$ reduces to
\begin{equation}
S(CA^k x(t) + C(I-A)^{-1}(I-A^k)Bv(t) + Dv(t)) \leq s,
\end{equation}
\begin{equation}
S(C(I-A)^{-1}Bv(t)) \leq s(1-\epsilon),
\end{equation}
\begin{equation}
k = 0, \ldots, k^*,
\end{equation}
where $k^*$ is any upper bound on the finite-determination index [51], [49] and $1 > \epsilon > 0$. Hence the conditions on the scalar $\beta(t)$ take the following form,
\begin{equation}
\beta(t)H_k(k)(v(t) - v(t-1)) \leq s - H_k(k)x(t) - H_v(k)v(t-1),
\end{equation}
\begin{equation}
\beta(t)H_k(\infty)(v(t) - v(t-1)) \leq s(1 - \epsilon) - H_k(\infty)v(t-1),
\end{equation}
\begin{equation}
k = 0, \ldots, k^*,
\end{equation}
for appropriately defined matrices $H_k(k), H_k(\infty)$ and $H_v(k)$ based on (9). These matrices can be either pre-computed off-line and stored (while eliminating redundant constraints to simplify the representation and the associated computations), or computed on-line in case the model or the constraints undergo on-line changes. Each of the above inequality conditions and $0 \leq \beta(t) \leq 1$ bound $\beta(t)$ in $[0, \beta_{\max}(t, k)]$, $k = 0, 1, \ldots, k^*$ and $k = \infty$. Therefore, $\beta(t)$ is set to the minimum of $k^* + 1$ numbers $\beta_{\max}(t, k)$. Similar conditions on $\beta(t)$ are obtained if $W \neq \{ 0 \}$, see [49] for details.

Further simplifications occur with $P \neq \hat{O}_\infty, P \subset \hat{O}_\infty$ in (4). While $P$ has to satisfy certain assumptions, it is not required to be positively-invariant and can be much simpler than $\hat{O}_\infty$. With the non-positively invariant $P$, a situation that no feasible $\beta(t)$ yielding (4) exists can occur. In such a case, $\beta(t) = 0$ is chosen. Following this procedure, the constraint enforcement and usual response properties of the reference governor to constant inputs are maintained. In fact, $P$ can be obtained from $\hat{O}_\infty$ by the systematic elimination of almost redundant constraint and applying a pull-in procedure, see [49]. This strategy can lead to a ten-fold reduction in the on-line computing effort with some loss in performance, see e.g., [134].

As an illustration, we consider a reference governor designed for a double integrator system. The system model in continuous-time has the following form,
\begin{equation}
x_1 = x_2,
\end{equation}
\begin{equation}
x_2 = u.
\end{equation}
The model is converted to discrete-time assuming a sampling period of $T_s = 0.1$ sec. A nominal controller is defined as
\begin{equation}
u = -0.917(x_1 - v) - 1.636x_2,
\end{equation}
where $v$ is a set-point for $x_1$. The state and control constraints are imposed as
\begin{equation}|x_1| \leq 1, \ |x_2| \leq 0.1, \ |u| \leq 0.1.
\end{equation}
The initial state is $x_1(0) = x_2(0) = 0$. For small $r$, the nominal response with $v = r$ satisfies the constraints and the reference governor remains inactive. For a larger command, $v = r = 0.5$, the nominal controller results in the maximum excursion of $x_2$ of about 0.2 and $u$ of about 0.46, both violating the imposed constraints by large amounts. To avoid de-tuning the nominal controller (and thus compromising the response for small commands that do not cause constraint violation), we apply SRG to govern $v$. The $P = \hat{O}_\infty$ is used in the reference governor implementation, which is a polytope defined by 36 linear inequalities. The closed-loop responses of the system...
to the command \( r = 0.5 \), after augmentation with SRG, are
given in Fig. 2 and compared to the unconstrained response. The
effect of the reference governor is to slow down the command and the subsequent system response. Note that the
modified reference command signal \( v(t) \) converges to the command \( r = 0.5 \) in a finite time.

The response properties of the reference governor, in-
cluding conditions for the finite-time convergence of \( v(t) \)
to \( r(t) \), are detailed in references [51], [49]. Essentially, if
\( r(t) \) for \( t \geq t_0 \) remains constant or varies in a sufficient small
neighborhood of a constant value, then \( v(t) \) converges to \( r(t) \)
in a finite-time if \( r(t) \) is steady-state constraint admissible. If \( r(t) \) is not steady-state constraint admissible then \( v(t) \)
will converge to the closest feasible value in a finite time. Similar finite-time convergence results can be developed for
sufficiently slowly varying \( r(t) \). The finite-time convergence is a desirable property indicating that, after transients caused
by large changes in \( r(t) \), the reference governor becomes inactive and nominal closed-loop system performance is
recovered. This or similar properties are retained by other
governors discussed next.

A. Vector Reference Governor (VRG)

The Vector Reference Governor (VRG) approach is a
modification of (3) for \( m \geq 1 \). The VRG uses a diagonal
matrix \( \mathbf{K}(t) \) in place of a scalar \( \beta(t) \) to decouple the
governing of different channels [51]:

\[
    v(t) = v(t - 1) + \mathbf{K}(t)(r(t) - v(t - 1)),
\]

where \( \mathbf{K}(t) = \text{diag}(\beta_i(t)) \). The values of \( \beta_i(t) \), \( i = 1, \cdots, n \), are chosen to minimize \( (v(t) - r(t))^T Q (v(t) - r(t)) \), where
\( Q = Q^T > 0 \) with \( v(t) \) given by (12), subject to the constraints
\( 0 \leq \beta_i(t) \leq 1, i = 1, \cdots, n \), and \( (v(t), x(t)) \in \mathcal{O}_\infty \). This optimization problem can be solved online using quadratic program-
ing techniques. Online use of conventional iterative techniques can be avoided by using explicit multi-parametric quadratic programming [14], [75], [76]. The VRG is superior to the SRG in that it offers more flexibility in the choice of
\( v(t) \). In applications, it can provide faster response than the
SRG, at the cost of increased computational effort.

B. Command Governor (CG) and Extended Command Gov-
ernor (ECG)

The command governor and extended command governor
approaches were proposed in [11], [25], [52], see also references therein. The simplest variant of the command
governor is similar to the VRG. In it, a cost function,
\[
    J = ||v(t) - r(t)||_2^2 = (v(t) - r(t))^T Q (v(t) - r(t)),
\]
where \( Q = Q^T > 0 \), is minimized with respect to \( v(t) \) subject
to the constraint
\[
    (v(t), x(t)) \in \mathcal{O}_\infty.
\]
In case \( \mathcal{O}_\infty \) is polyhedral, \( v(t) \) is computed by solving a
quadratic programming problem either online or offline using
multi-parametric programming techniques [14]. This explicit
solution gives \( v(t) \) as a piecewise affine function of \( x(t) \) and
\( r(t) \).

A variant of the CG, the so called Prioritized Reference
Governor (PRG) has been proposed in [64]. The PRG strictly
enforces hard constraints and it satisfies soft constraints in
the order of their priority. The soft constraints are relaxed
by slack variables and the penalty on the slack variables is
added to the cost with lower weights corresponding to lower
priority constraints.

The extended command governor approach of [52] is an
extension of the command governor approach that generates
\( v(t) \) according to,
\[
    v(t) = \hat{C} \hat{x}(t) + \rho(t), \tag{13}
\]
where, over the prediction horizon, the fictitious state, \( \hat{x}(t) \in \mathbb{R}^n \), and \( \rho(t) \) evolve as,
\[
    \hat{x}(t+k+1) = \tilde{A} \hat{x}(t+k), \quad k \geq 0,
    \rho(t+k) = \rho(t). \tag{14}
\]
The values of \( \rho(t) \) and \( \hat{x}_0(t) \) are optimized using a quadratic
cost function,
\[
    J = \frac{1}{2} ||\rho(t) - r(t)||_2^2 + \frac{1}{2} ||\hat{x}(t)||_2^2,
\]
with \( P = P^T > 0 \), satisfying \( \tilde{A}^T P \tilde{A} - P < 0 \), subject to the
constraint that \( (\rho(t), \hat{x}(t), x(t)) \in \mathcal{O}_\infty \). The set \( \mathcal{O}_\infty \) is a
finitely determined inner approximation to the set of all triplets
\( (\rho(t), \hat{x}(t), x(t)) \) that do not induce subsequent constraint
violation when the input sequence \( v(t+k) \) is determined by the
fictitious dynamics per (13) and (14). Without the fictitious
states, i.e., when \( \tilde{n} = 0 \), \( \mathcal{O}_\infty = \mathcal{O}_\infty \), the ECG becomes the sim-
ple command governor [52]. The optimization problem can
be solved online using conventional quadratic programming
techniques. Again iterative procedures can be avoided by
using explicit multi-parametric quadratic programming [14].

Various choices of \( \tilde{A} \) and \( \tilde{C} \) in (13), (14) can be made. The
shift sequences used in [52] are generated by the fictitious
dynamics with,
\[
\tilde{A} = \begin{bmatrix}
0 & I_m & 0 & 0 & \cdots \\
0 & 0 & I_m & 0 & \cdots \\
0 & 0 & 0 & I_m & \cdots \\
0 & 0 & 0 & 0 & \cdots \\
\vdots & \vdots & \vdots & \vdots & \ddots
\end{bmatrix},
\]
\[
\tilde{C} = \begin{bmatrix} I_m \\
0 \\
0 \\
0 \end{bmatrix},
\]
where \(I_m\) is an \(m \times m\) identity matrix. In this case, ECG can be re-formulated as a Model Predictive Controller (MPC).

Another approach [62], motivated by [113], uses Laguerre sequences. These sequences possess orthogonality properties and are generated by the fictitious dynamics with,
\[
\tilde{A} = \begin{bmatrix}
\alpha & 0 & \cdots & 0 \\
0 & \alpha & \cdots & 0 \\
\cdots & \cdots & \cdots & \cdots \\
0 & 0 & \cdots & \alpha
\end{bmatrix},
\]
\[
\tilde{C} = \sqrt{\mu} \begin{bmatrix} I_m \\
\alpha \cdot I_m \\
\cdots \\
\alpha^{N-1} \cdot I_m
\end{bmatrix},
\]
where \(\mu = 1 - \alpha^2\), and \(0 \leq \alpha \leq 1\) is a selectable parameter that corresponds to the time-constant of the fictitious dynamics. Note that with the choice of \(\alpha = 0\), (16) coincides with the shift register considered in [52]. The advantage of the ECG is that it can produce larger domains of attraction for \(x(0)\) than the ones obtainable using the SRG, VRG, or CG. Application examples, such as in [110], suggest that this is the case, in particular, for systems with rate limits.

### III. Reference Governors for Nonlinear Systems

In the case of nonlinear system models, the design of the reference governor can be based on model linearization at a selected operating point, or by applying the reference governor to the nonlinear model directly.

#### A. Reference governor design based on model linearization

In the case model linearization is employed, the differences between the nonlinear model and the linearized model can be compensated by modifying the output of (1) with a constant offset term, \(g(t)\), [134], [62],
\[
y(t) = Cx(t) + Dv(t) + Dw(t) + g(t).
\]
The offset term is then augmented to the linearized system model as an extra state, with the assumed dynamics,
\[
g(t+k) = g(t).
\]
During the on-line operation, \(g(t)\) is set to the difference between the currently measured or estimated actual output and its prediction based on the linear system model, i.e.,
\[
o(t) = y_{\text{nonlinear}}(t) - y_{\text{linear}}(t).
\]
This tightens the constraints and serves as a practical measure to protect against constraint violation. In the case constrained outputs are not directly measured, they can be estimated with state or input observers, see e.g., [65] for an example. While errors can be tolerated in prediction, accurate measurements or estimates of current values of constrained outputs are useful for non-conservative treatment of constraints.

#### B. Reference governor design based on nonlinear model

Approaches to reference governor design for constraint enforcement based on nonlinear closed-loop system models include [5], [8], [50], [53], [101]. Several of the reported approaches exploit predictive on-line simulations or sub-level sets of Lyapunov functions to guard against constraint violation. An incremental step reference governor strategy with one update of fixed magnitude per time step has been investigated in [133]. Landing reference governors have been proposed for systems with terminal constraints, e.g., for reaching a desired position with a small velocity in mechanical systems [80], [81], [71]. Strategies for the design of reference governors based on approximating nonlinear models by piecewise affine models have been reported in [17]. Reference [6] compares direct nonlinear versus linearization-based treatment.

In a typical setup, the nonlinear system model has the form,
\[
x(t+1) = f(x(t), v(t), w(t)),
\]
with the constraints
\[
y(t) = h(x(t)) \leq 0,
\]
where \(y \in \mathbb{R}^p\). The input \(w\) is assumed to be stationary, e.g., it may represent a parametric uncertainty, a set-bounded or a set-bounded and a rate-bounded unmeasured disturbance.

Let a function \(S\) be such that with \(v(t+k) = \bar{v}\) for all \(x(t)\) in the sub-level set of \(S\), i.e., satisfying
\[
S(x(t), \bar{v}) \leq 0,
\]
the subsequent trajectory is guaranteed to be safe and strongly returnable. The trajectory is safe if the constraints \(y(t+k) \in Y\) hold for all \(k \in \mathbb{Z}^+\). The trajectory is strongly returnable [50], if there exists \(k^*\) that does not depend on \(x(t)\) or \(\bar{v}\), and \(k > k^*\), that may depend on \(x(t)\) or \(\bar{v}\), such that \(S(x(t+k), \bar{v}) < 0\). Note that no requirement is made on the sub-level set (20) to be invariant. Technical conditions require \(S\) to be continuous but it can be (and often is) non-smooth.

With \(S\) defined, \(\beta(t)\) in the SRG (3) can be chosen based on solving a scalar optimization problem,
\[
\beta(t) = \max \{ \lambda \in [0, 1] : S(x(t), v(t-1) + \lambda (r(t) - v(t-1)) \leq 0 \}\},
\]
If no feasible solution to (21) exists, \(\beta(t) = 0\) is used.

Several methods to construct \(S\) exist. If Lyapunov or Input-to-State Stable (ISS) Lyapunov functions, \(V\), for the closed-loop system are available then typically
\[
S(x, v) = V(x, v) - c,
\]
with an appropriately chosen \(c > 0\). This approach exploits the positive-invariance properties of sub-level sets of Lyapunov and ISS-Lyapunov functions. In this case, the choice
\( \lambda = 0 \) is always feasible in (21).

Another approach to construct \( S \) in the disturbance-free case uses off-line simulations of the closed-loop system and generates \( S \) as a classifier so that \( S(x(t), \tilde{v}) \leq 0 \) distinguishes safe pairs of states and constant reference commands from the unsafe ones [50]. Machine learning techniques [108] are useful in computing such a classifier. Combining several such classifiers, \( S_j(x, v), \; j = 1, \cdots , J \), with the choice \( S(x, v) = \max_{j=1}^{J} S_j(x, v) \) is permitted under appropriate assumptions [50]. In the case with disturbances, the offline computations are more involved as the classification of pairs of states and constant reference commands into safe and unsafe pairs involves evaluating multiple disturbance scenarios or solving optimal control problems with respect to \( w(\cdot) \).

Finally, instead of developing an explicit functional representation for \( S \), \( S \) can be also defined implicitly [8] as
\[
S(x(t), \tilde{v}) = \max_{i=1,\cdots ,k^*} \max_{w(\cdot)} h_i(x(t+k|x(t), \tilde{v}, w(\cdot))),
\]
where \( x(t+k|x(t), \tilde{v}, w(\cdot)) \) is the predicted trajectory emanating from the state \( x(t) \) with \( v(t+k) = \tilde{v} \) and \( w(t+k) \in W \). The horizon \( k^* \) must be sufficiently long, see [8]. The determination of \( S \) based on (22) reduces to an offline simulation if \( W = \{0\} \); if \( W \neq \{0\} \), in evaluating (22) either multiple scenarios of \( w(t) \) are considered or a set of optimal control problems with respect to \( w(\cdot) \) (corresponding to \( 0 < k \leq k^* \) and to different components of the output vector) is solved.

The online optimization of a scalar parameter \( \lambda \) in (21) is generally simple. Procedures based on bisections or grid search (while checking \( \beta(t) = 1 \) first) can be employed. The incremental reference governor approach of [133] distributes the solution over time by checking the feasibility of a single value of \( v(t) \) that differs from \( v(t-1) \) by a fixed step size. In case \( S \) is quadratic, an explicit solution for \( \lambda \) in (21) is easily derived; in other cases, numerical solution strategies such as the one proposed in [90] based on predictor-corrector form of Newton’s method applied to parameteric root-finding can be used.

The response properties of the nonlinear reference governors are similar to the ones in the linear case: If an initial selection of \( v(0) \) exists such that \( S(x(0), v(0)) \leq 0 \) then constraints (19) will be satisfied for all \( t \in Z^+ \) and \( v(t) \) will converge to a constant steady-state constraint admissible \( r(t) \) in a finite-time. See [50] for the precise statement of the relevant conditions.

C. Robust reference governor

In the robust reference governor approach of [123], the model (18) has no time-dependent disturbance input but depends on uncertain parameters, \( \theta \in \Theta \subset \mathbb{R}^l \),
\[
x(t+1) = f(x(t), v(t), \theta),
\]
and constraints have the form,
\[
y(t) = h(x(t)) \leq 0.
\]
In verifying the feasibility of \( \beta \), the range of \( \theta \in \Theta \subset \mathbb{R}^l \) is covered by a grid of values, \( \Theta^j \), and sets \( \Theta^j \), \( j = 1, \cdots , n_\theta \). The admissibility of given \( x(t) \), \( v \) is tested based on whether the predicted response satisfies the constraints (24). In this prediction, (24) is replaced by a set of conditions
\[
y^j(t; \theta) = y(t; \theta^j) + \sum_{i=1}^l z_i^j(t)(\theta_i - \theta_i^j) + M||\theta - \theta^j||^2 \leq 0,
\]
where the \( j \)th condition must hold for all \( \theta \in \Theta^j \). In (25), \( z_i^j(t) = \frac{\partial f}{\partial \theta_i}(x^j(t), \theta^j) \) is computed using the parametric sensitivity equations,
\[
z_i^j(t+1) = \frac{\partial f}{\partial x}(x^j(t), v, \theta^j)z_i^j(t) + \frac{\partial f}{\partial \theta_i}(x^j(t), v, \theta^j),
\]
\[
z_i^j(0) = 0.
\]
The expression (25) is based on Taylor series expansion of the solution, with \( M > 0 \) chosen to protect against the inaccuracies of this approximation. Often the implementation with \( n_\theta = 1 \) suffices.

D. Reference governor for linear systems with nonlinear constraints

In [61] a special case of reference governor design based on a linear system model with nonlinear constraints has been studied. The nonlinear constraints are given by
\[
Y = \{y : h_i(y) \leq 0, \; i = 1, \cdots , p\},
\]
where \( h_i \) are nonlinear functions. Feedback linearization [74] can be applied to many systems to render the closed loop system linear, however, the resulting \( Y \) is typically non-polyhedral.

The existing reference governor results in [51], [49] for system (1) with constraints (28) hold for any compact, convex set \( Y \) with \( Y \subset int Y \). When \( Y \) is not polyhedral, it often can be approximated by a polyhedron, however, such approximations may lead to reference governor designs that are either conservative or have high computational complexity especially for systems with multi-dimensional constrained outputs. Instead of employing polyhedral approximations, in [61] the linear model (1) is used to predict \( y(t) \) and the predicted violations of \( h_i(y(t+k)) \leq 0 \) for \( k > 0 \) are used to constrain \( \beta(t) \) in (3). With this approach, several types of constraints can be treated, including convex constraints, convex quadratic constraints, Mixed-Logical-Dynamic (MLD) constraints of if-then type and concave constraints. For instance, in the case of quadratic constraints, \( \beta(t) \) is given by an explicit formula. In the case of concave constraints, constraint linearization is employed, leading to replacing the condition \( y(t+k|t) \in Y \) by
\[
y(t+k|t) \in Y, \quad Y = \{y : h_i(y) \leq 0, \; i = 1, \cdots , p\},
\]
and \( y_{i^*}(t) \) is appropriately chosen, \( i = 1, \cdots , p \). Finally, the
landing reference governor handles if-then terminal constraints in systems requiring the soft-landing of the components [80], [81], [71].

E. Parameter governor

Parameter governors [84], [85], [86] adjust parameters, \( \theta(t) \in \Theta \), in nominal control laws to optimize predicted system response over a finite, receding horizon subject to constraints. Parameters are assumed to remain constant over the prediction horizon, \( T \), and the cost function of the general form,

\[
J(t) = ||\theta||^2_{\Psi_0} + \sum_{k=0}^{T} \Omega(x(t+k|t), \theta(t), r(t)),
\]

is minimized that penalizes the predicted response, \( x(t+k|t) \), as well as parameter deviations. The assumption of constant parameters over the prediction horizon reduces computational and implementation effort, and simplifies the analysis. In fact, \( \Theta \) is permitted to be a finite set so the evaluation of (31) and of the constraints reduce to a finite number of online simulations.

Specific parameter governor schemes considered in [84] include the offset governor and the gain governor. For these schemes, terminal set conditions need not be imposed to assure stability provided the horizon, \( T \), is chosen sufficiently long.

In the offset governor approach of [84], a disturbance free system is considered with an integrator included as a part of the overall system,

\[
\begin{align*}
x(t+1) &= f(x(t), u(t)), \\
x_i(t+1) &= x_i(t) + z(t) - r, \\
z(t) &= h_e(x(t)),
\end{align*}
\]

where \( z \) is an auxiliary output which may differ from the constrained output. The control law includes integral action and an adjustable feed-forward offset \( \theta(t) \),

\[
u(t) = u_e(r) - \varepsilon x_i(t) + \bar{u}_{fb}(x(t), r) + \theta,
\]

where \( x_e(r), u_e(r) \) denote the equilibrium values of state and control variables corresponding to the given \( r \). Due to the use of integral action, if \( \theta \) is constant, then as \( t \to \infty \) it follows that \( z(t) \to r, x(t) \to x_e(r), u(t) \to u_e(r) \). The small gain integral control leads to dynamics decomposition into slow and fast modes. The fast dynamics can be made to avoid constraints by changing \( \theta(t) \); consequently, the constraints will be satisfied if slow manifold is well within the constraint admissible region. The offset governor of [84] can thus handle large reference changes and recover a large set of initial states. The cost (31) is modified to include the penalty on the integral states.

In the gain governor approach of [84], the adjustable parameters are the control gains

\[
\begin{align*}
x(t+1) &= f(x(t), u(t)), \\
u(t) &= u_e(r) + u_{fb}(x(t), r, \theta(t)),
\end{align*}
\]

where \( u_{fb}(x_e(r), r, \theta) = 0 \) for all \( \theta \in \Theta \). The gain governor is, in particular, effective for systems with control constraints.

F. Other developments

Other approaches to reference governor design for nonlinear systems are possible. We mention the developments in [41] (and early paper [40] for linear systems) and [57] as specific examples.

IV. RECENT REFERENCE GOVERNOR DEVELOPMENTS

In this section, we discuss several more recent reference governor developments; these include feedforward reference governor, reduced order reference governor, reference governor for decentralized systems, and reference governor for networked systems.

A. Feedforward reference governor

Most of the command/reference governor approaches presented in the literature make explicit use of state measurements or of suitable state estimates, see e.g. [7], to modify the reference in order to ensure constraints satisfaction. However, as has been noted in recent papers, sensorless command/reference governor schemes that do not make explicit use of the plant state can also be designed, at the price of some additional conservativeness. This is not surprising as feedback is not a necessary requirement in many classical schemes to manage the reference, e.g. when filtering the reference signal to avoid high frequency responses.

The main idea behind feedback command/reference governor approaches is that if the set-point modifications are sufficiently slow then one can have high confidence on the expected value of the state, even in the absence of an explicit measurement of it, because of the asymptotic stability of the pre-compensated system at hand. This feature is of interest in applications where obtaining state measurements/estimates is difficult or undesirable, and in multi-agent distributed supervisory schemes where to know the entire aggregate state (or part of it) at each time instant can be very costly or even unrealistic.

A feedforward command governor scheme was first introduced in [44] within a decentralized command governor scheme. In that paper, the main idea is to choose at each time instant an input \( v(t) \) such that the corresponding steady-state equilibrium satisfies the constraints (i.e., \( v(t) \in \hat{\Theta}_e \) where \( \hat{\Theta}_e \) is defined in (7)), and to change \( v(t) \) 'slow enough' so as to ensure that the constrained output is always 'close' to the steady-state equilibrium. To do so, two ingredients are exploited:

- The input \( v(t) \) is changed every \( \tau \) steps and kept constant in between, where \( \tau \) is a generalized settling time for the system (see [45] for details);
- The variation \( v(t+\tau) - v(t) \) is constrained to belong to a pre-computed set \( \Delta \) which, in combination with the generalized settling time, ensures that the output trajectory is always inside a ball of radius \( \varepsilon \) from the steady state.

This very early feedforward scheme was proven to have the same theoretical properties as the classical command governor, but also turned out to be quite conservative. In [45], the conservatism of the approach has been substantially...
reduced by constraining \(v(t + \tau) - v(t)\) to a new set \(\Delta V(v(t - \tau), \rho(t))\) which depends on the previously applied command \(v(t - \tau)\) and on the scalar \(\rho(t)\), representing an estimate of the maximal possible distance between the current value of the output and the steady-state value associated with the input \(v(t - \tau)\). A further reduction of conservatism has been achieved in [34], where a scheme that allows the command governor to modify the command at each time instant has been presented. This scheme has been proved to be asymptotically equivalent to a command governor with feedback in the case of non-noisy systems. However, as shown in the same paper, feedforward command/reference governors may behave poorly with plants affected by large disturbances.

B. Reduced order reference governors

The reduced order reference/command governor [63, 68] for systems with states decomposable into “slow” and “fast” states can be based on the reduced order model for “slow” states, provided constraints are tightened to ensure that the contributions of fast states do not cause constraint violation. The governor implementation based on the reduced order model has lower computational complexity and can use a reduced order observer for state estimation. Also, if the reduced order model is a second order model (plus a possible time delay), fast reference governor designs that exploit second order system response properties become possible [96].

Consider system (1), where we assume that \(W = \{0\}\) for the ease of exposition. After appropriate state transformations, the decomposed system has the following state space realization,

\[
\begin{bmatrix}
x_i(t + 1)
\end{bmatrix} = \begin{bmatrix}
A_x & 0 \\
0 & A_f
\end{bmatrix} \begin{bmatrix}
x_i(t)
\end{bmatrix} + \begin{bmatrix}
B_x \\
B_f
\end{bmatrix} v(t),
\]

\[
y(t) = \begin{bmatrix}
C_x & C_f
\end{bmatrix} \begin{bmatrix}
x_i(t)
\end{bmatrix} + Dv(t),
\]

where \(x_i(t) \in \mathbb{R}^{n_i}\) and \(x_f(t) \in \mathbb{R}^{n_f}\) are the slow and fast vector states, respectively, \(n_i + n_f = n\), and the matrices are appropriately sized.

The reference governor design is based on the reduced order system model, representing the dynamics of the slow states and steady-state values of fast states,

\[
x_i(t + 1) = A_x x_i(t) + B_x v(t),
\]

\[
y_i(t) = C_x x_i(t) + (C_f \Gamma_f + D) v(t),
\]

where \(\Gamma_f = (I_{n_f} - A_f)^{-1} B_f\).

In order for the reference governor based on the reduced order model, (37)-(38), to enforce the constraints for the full order model, we tighten the constraints. Specifically, we introduce two “error sets" \(E_x \subset \mathbb{R}^{n_i}\) and \(E_y \subset \mathbb{R}^p\). We require that \(E_x\) be \(A_f\)-invariant, i.e.,

\[
A_f E_x \subseteq E_x,
\]

and also satisfy the following inclusion,

\[
C_f E_x \subseteq E_y.
\]

The the reduced order reference governor enforces tightened constraints on the reduced order output,

\[
y_i(t + k) \in Y \sim E_y, \forall k \geq 0.
\]

To ensure that the contributions of the fast states do not lead to constraint violation, the reduced order reference governor enforces an extra constraint of the form,

\[
-A_f \Gamma_f \Delta v(t) \in E_x \sim A_f E_x.
\]

Where \(\Delta v(t) = v(t) - v(t - 1)\). Here, \(\sim\) denotes the \(P\)-difference of two sets\(^1\). The constraint (42) can be re-written as a constraint on \(\dot{\beta}(t)\),

\[
-A_f \Gamma_f \beta(t)(t) - v(t - 1)) \in E_x \sim A_f E_x.
\]

If \(v(-1)\) is initialized so that \(e_f(0) = x_f(0) - \Gamma_f v(-1) \in E_x\), the reference governor response properties [49] hold, specifically, the recursive feasibility of \(\dot{\beta}(t) = 0\), the guaranteed constraint enforcement and the finite-time convergence for constant set-points. As shown in [63, 68], the approach of imposing ancillary constraints on the evolution of \(v(t)\) is also applicable to ensure that observer errors do not cause constraint violation. In [68], the results are extended to handle the case of set-bounded unmeasured disturbances inputs and to the design of the extended command governors based on the reduced order models.

C. Network reference governor handling variable delays

Reference governor-based approaches for network control systems have been proposed in [9], [27], [28] and more recently in [18], [19]. In these approaches, the controller and the plant are connected via a, usually non-ideal, communication network. In [18], [19], the set-point commands, \(v(t)\), are transmitted through a communication channel that has variable continuous time delay, \(\delta(t) \in [0, \delta]\). When the delay \(\delta(t)\) is a smaller than the reference governor update period, \(T_s\), the effect of the delay on the state is shown in [18] to satisfy the following relation:

\[
x(t + 1) = A x(t) + B v(t) + R(\delta(t)) \Delta v(t)
\]

\[
\Delta v(t) = v(t + 1) - v(t),
\]

where

\[
R(\delta) = -\int_0^{\delta} e^{A_x(T_s - \tau)} B_x d\tau,
\]

and \((A_x, B_x)\) is the continuous-time realization of the system controlled through the communication network. Consequently, the effective disturbance introduced by the delay in (43) is modulated by \(\Delta v(t)\), i.e., change in \(\Delta v\).

To overbound \(R(\delta)\Delta v\), suppose a matrix \(P\) and a set of vertices \(\{w_i, i = 1, \cdots, n_i\}\) is given such that \(\|P\Delta v\| \leq 1\) implies \(R(\tau) \Delta v \in convh\{w_i, i = 1, \cdots, n_v\}\) for all \(0 \leq \tau \leq T_s\), where \(convh\) denotes the convex hull. Then the reference governor algorithm that ensures constraint enforcement and finite time convergence properties has the following form

\[^1\text{A} \sim \text{B} := \{a \in A : a + b \in A, \forall b \in B\}. \text{See [83].}\]
The VSG modulates the effect of the controllers by modifying the state from which the feedback is computed. In this way the closed-loop system dynamics become

$$x(t+1) = Ax(t) + B_1K_1 B_2K_2 \Omega(x).$$ (48)

The case where only disjoint constraints on the actuators $u_i \in U_i, U_i$ being a polytope, for $i = 1, 2$, need to be enforced is considered in [23]. By defining

$$\Omega(x(t)) = \arg \min_{x_1, x_2} x_1^2 P_2 x_2,$$ (49)

$$x_1 \in O^{(i)}_{\omega}, \quad i = 1, 2,$$

$$x_1 + x_2 = x(t),$$

where $O^{(i)}_{\omega}$ is the maximum output admissible set for $(A + B_iK_i)$ and $u_i = K_i x_i \in U_i, \quad i = 1, 2, P_2$ is a solution of a Lyapunov equation for $(A + B_2K_2)$, and $u_2$ is the actuator input whose use must be minimized, it is shown that the closed-loop system origin: $(i)$ is asymptotically stable, $(ii)$ has a domain of attraction $O^{(1)}_{\omega} \cup O^{(2)}_{\omega}$, and $(iii)$ for every initial state in $O^{(1)}_{\omega} \cup O^{(2)}_{\omega}$, there exists a finite time $\bar{t} \in Z^+$ such that $u_2(t) = 0$ for all $t \geq \bar{t}$.

In [23] it is also discussed how the proposed approach can be extended to handle binding constraints by computing bound-parametric output admissible sets and by introducing additional constraints in (49). This approach has similarities with handling soft constraints in the prioritized reference governor introduced in Section II-B.

E. Decentralized Command Governor

The development of decentralized command governor schemes has been an active research topic in recent years. The considered system setting is the following. Let $\mathcal{N}$ be a set consisting of $N$ interconnected subsystems, each one being an LTI closed-loop dynamical system regulated by a local controller which ensures the stability of the overall interconnected system with good closed-loop properties in linear regimes (i.e., when the constraints are not active). Let the $i$th closed-loop subsystem be described by the following discrete-time model,

$$\begin{cases}
  x_i(t+1) = A_i x_i(t) + B_i v_i(t) + \sum_{j \in \mathcal{N}-\{i\}} A_{ij} x_j(t), \\
  y_i(t) = C_i x_i(t) + D_i v_i(t),
\end{cases}$$ (50)

where $x_i$ is the local state, $y_i$ is the local constrained output, and $v_i$ is the local manipulable reference vector which, if no constraints (and no command governor) were present, would coincide with the desired reference $r_i$. Each subsystem has its own reference signal and is governed by a local reference management unit. The management units are organized in a communication network described by an undirected graph $\Gamma = (\mathcal{N}, \mathcal{E})$, where the set of edges $\mathcal{E} \subseteq \mathcal{N} \times \mathcal{N}$ describes the existing communication links among the units. Common assumptions for the schemes developed thus far are that each unit may exchange information only with its neighbors (i.e. the agent $i$ may communicate with the node $j$ only if $(i, j) \in \mathcal{E}$).

The considered system setting is the following. Let $\mathcal{N} = \{1, 2, \ldots, N\}$ be a set of controllers. The obtained control architecture must be capable of effectively exploiting all the actuators at the same time if needed, but it also needs to minimize the use of those actuators that are “expensive” to operate. This problem is of interest in automotive applications, for instance in cornering control [21], engine control [20], and energy management in hybrid powertrains [22], and in aerospace applications such as attitude control [23].
and that each couple of nodes may communicate only once between two time steps.

The goal of decentralized command governor is to determine a distributed reference management strategy able to locally modify the commands \(v_i, i = 1, \ldots, N\), in such a way that:

- The application of all the \(v_i, i = 1, \ldots, N\), is such that the aggregated output \(\mathbf{y} = [y_1^T, \ldots, y_N^T]^T\) does not violate global (coupling) constraints \(y(t) \in Y, \forall t\);
- Each \(v_i(t)\) approximates as close as possible \(r_i(t)\).

The first solutions proposed to solve this problem made use of the feedforward command governor approach [44]. The use of the feedforward approach in its simpler formulation, allows one to reformulate the decentralized reference management problem as the static problem of locally determining every \(\tau\) steps the commands \(v_i, i = 1, \ldots, N\) such that the aggregated vector \(\mathbf{v} = [v_1^T, \ldots, v_N^T]^T\) belongs to the static set \(\hat{\mathcal{O}}_v\) and such that the variation of \(v\) between two update times is constrained as \(v(t + \tau) - v(t) \in \Delta \mathcal{V}\).

To solve this problem the general idea is that at each time instant:

- On the basis of the information common to all agents, each agent computes locally the same family of local artificial regions \(\Delta V_i(t), i = 1, \ldots, N\). These regions are convex and compact sets containing the origin as an internal point and are such that
  \[
  \Delta V_1(t) \times \ldots \times \Delta V_N(t) \subseteq \Delta V \cap \{\hat{\mathcal{O}}_v \sim \{v(t - \tau)\}\},
  \]
  which means that whenever \(v_i(t) \in \Delta V_i(t), i = 1, \ldots, N\), constraints are satisfied.
- On the basis of the locally available information, each agent computes the command to be applied minimizing its local cost function \(\|g_i(t) - r_i(t)\|_2\), with matrix \(\Psi_i = \Psi_i^T > 0\), subject to the constraint that \(v_i(t) \in \Delta V_i(t)\).

Following this general philosophy, sequential and parallel approaches have been introduced. Sequential approaches [35] are schemes where only one agent at a time is permitted to modify its command. Parallel approaches [36] are schemes where all agents are permitted to modify their commands at the same time, making worst case assumptions on the choices of the others. This second approach has been shown to work quite well when the aggregated command \(v\) is far from the borders of \(\hat{\mathcal{O}}_v\), but quite poorly close to them. For this reason, hybrid approaches based on switching between parallel and sequential modes have been proposed in [46] and [128]. We note that while all the above mentioned schemes ensure constraint satisfaction, they may experience problems of convergence to “good approximations” of the desired reference signals \(r_i(t), i = 1, \ldots, N\). In fact, as shown in [35], these schemes may experience convergence of the commands to Nash equilibria that are not Pareto optimal. In [37] and [127] this phenomenon has been carefully investigated and algorithms to check the existence of these anomalies and to eliminate them have been provided. Decentralized schemes making use of the same ideas developed for feedforward distributed command governors have been recently presented in [129] and [38].

### F. Other developments

We mention also approaches in [125] that combine reference governing and controller switching to improve performance. Related strategy is described in [87]. Reference governor schemes for systems with delays and faults are considered in [120], [121].

### V. AUTOMOTIVE APPLICATIONS

In this section we discuss several applications of reference and command governors to practical systems arising in automotive applications.

#### A. Turbocharged automotive engines

As gasoline engines are downsized and turbocharged to improve fuel consumption, protecting the engine from violating constraints without compromising engine response is becoming harder and requires systematic treatment. The constraints include the compressor surge limit, actuator limits on throttle and wastegate, turbine speed and temperature limits, intake pressure overshoot limit, combustion limits, etc.

The surge constraint handling in turbocharged gasoline engines (see the schematic in Fig. 3) using reference governor techniques is addressed in [62], [63]. The surge constraint is based on 2 constrained outputs \((y(t))\): boost pressure (kPa), and compressor flow \((g/sec)\) that to avoid surge are constrained by a single affine inequality as \(y(t) \in Y = \{y : Sy \leq s\}\).

![Fig. 3. Schematics of a turbocharged gasoline engine with an electronic throttle and a wastegate actuators.](image)

In [62], it is shown that the wastegate has little if any authority to prevent surge constraint violation, and that the scalar reference governor (SRG) for the throttle is as effective as the vector reference governor (VRG) and the extended command governor (ECG) schemes. Hence, the SRG design for the throttle is adopted to prevent surge. This design is later extended to limit the intake manifold pressure overshoot and prevent torque overshoot and drivability concerns.

The application of the reduced order reference governor in [63] is motivated by different time scales of the engine model.
variables. The model is order five with the following state variables: intake manifold pressure (kPa), boost pressure (kPa), exhaust manifold pressure (kPa), turbocharger speed (rpm), and wastegate flow (g/sec). The eigenvalues of the linearized continuous-time model are,

\[-2.39, -3.16, -24.3, -161, -259\],

suggesting that the dynamics can be decomposed into a second or a third order slow subsystem and, respectively, a third or a second order fast subsystem. The validation results for the reduced order reference governor design based on the nonlinear model and the observer for unmeasured states are presented in [63]. See Figures 4-5. This reduced order reference governor design was based on the linear model and utilized the procedure for mismatch compensation with the nonlinear model described in in Section III-A.

B. Vehicle dynamics and rollover protection

The applications of the reference and extended command governors to vehicle rollover protection are considered in [88]. The schemes modify steering angle and operation of the brakes so that vehicle constraints are satisfied. The dynamics of vehicle yaw and roll motion are captured by a discrete-time linear model with four states: lateral velocity of Center of Gravity of the vehicle, yaw rate of the unsprung mass, roll rate of the sprung mass and roll angle of the sprung mass. The constraints on the load transfer ratio, which is the difference between the load on right tires minus the load on left tires divided by total weight of the vehicle, between \(-1\) and \(1\), are enforced. As shown in [88], both scalar reference governor (SRG) and extended command governor (ECG) prevent rollover, while modifications of the vehicle trajectory are slight. The on-line computing effort associated with the SRG (explicitly solvable scalar optimization problem) is shown to be less than of the extended command governor (simple and explicitly solvable parametric quadratic program), while the domain of recoverable states of the extended command governor can be much larger. Further as Figures 6-7 illustrate, ECG is robust to a mismatch between the model and actual system dynamics.

C. Electromagnetic actuators

Reference governors were applied to enforce a variety of constraints in electromagnetic actuators. See [71], [61], [101], [80], [81]. In particular, the constraint induced by the limited coil current leads to a constraint expressed by a concave nonlinear function, while the soft landing constraint is of MLD type. These types of constraints can be handled effectively using techniques in [61].

D. Fuel cells

The publications [134], [123], [94] consider handling constraints in fuel cell applications using reference governor techniques. In these systems constraints are imposed to
maintain the oxygen over hydrogen ratio sufficiently high, thereby preventing oxygen starvation, to avoid compressor surge and choke regions, and to avoid compressor voltage saturation.

The developments in [134] are based on the linearized model of order 10, and exploit the procedure in Section III-A for compensating the mismatch between the linearized model and the actual nonlinear system. The reference governor is applied to control fuel cell load (current) requests. Since in [134] (and also in [123], [94]) the reference governor controls the load, it is referred to as the load governor. The design exploits the use of non-positively invariant $P \subset \hat{O}_w$ in (4) that are obtained from $\hat{O}_w$ by eliminating almost redundant inequalities and the pull-in procedure. The number of inequalities in the representation of $\hat{O}_w \subset \mathbb{R}^{12}$ was 325 and reduced to 62 in the representation for $P$. This led to the reduction of the number of flops from 10100 to 1650. The implementation in the production micro-controller has shown that the reference governor computations require 1.3 msec at 10 msec update rate and 4kB of ROM.

The nonlinear reference governor is applied to fuel cell constraint handling in [123]. Parameter uncertainties in temperature and humidity are handled using the robust reference governor approach of Section III-C. Robust constraint enforcement capability with minimum impact on system response time has been demonstrated.

E. Other automotive applications

Applications to constraint handling in turbocharged diesel engine aftertreatment control and air path control are reported in [104], [105], [106]. Applications of the reference governor to constrained control of HCCI engines are considered in [60]. Reference [137] presents reference governor design to prevent piston and cylinder head collisions in free piston engines [137] based on a control-oriented engine model in an implicit form. References [89], [3] have considered the applications of reference and command governors to engine speed control. Reference governors were applied to handling constraints in electric batteries in [103] and [119]. Applications to vehicle dynamic control were considered in [17]. Finally, reference [135] has addressed reference governor application to belt restraint systems.

F. Non-automotive applications

This tutorial paper would be incomplete without mentioning several non-automotive applications of reference and command governors. Applications to aerospace systems are treated in [109], [31], [42] [82], [112], [72], [138], [110], [111], [39]. Applications to electric power systems are considered in [32], [29], [56], [30], [130], [115]. Reference [67] uses the reference governor to enforce constraints in a flying wind turbine system affected by wind disturbances. Other applications include chemical processes [77]; cable robots [107]; disk drives [55]; rotary cranes [59]; open water channel networks [95]; gas turbine engines [82], [66], [126]; inverted pendulum [26]; cooperative vehicle control [131]; four tank laboratory system [33]; electrostatically actuated membrane mirrors [79]; and tokamak reactors used in thermonuclear fusion [136], [99]. The growing breadth of these applications suggests widening interest in the reference governors for engineering applications.

VI. CONNECTIONS WITH OTHER DESIGN TECHNIQUES

In this Section we briefly comment on connections between reference/command governors and related control techniques.

A. Connections with Model Predictive Control

As predictive control schemes for constrained reference tracking, governors have many common features with Model Predictive Control (MPC) and, in fact, can be designed within MPC framework [114], [1]. At the same time, they are special schemes with unique motivation and several unique properties, results, and simplifications (such as finite-time convergence for constant reference commands or the design based on reduced constraint set) that are not easily available to more general MPC controllers. Furthermore, reference handling in MPC [43], [97], tube MPC [4], [100] and reduced complexity MPC approaches in [91] (arguably) incorporate some elements similar to reference/command governors. Parameter governors proposed in [84] have similarities with parameterized nonlinear MPC of [2]. Other related techniques for reference tracking in constrained systems include [15].

B. Connections with input shaping

The input shaping techniques have been proposed to minimize residual vibrations in flexible structures, see e.g. [116], [117], [118]. Similar to reference/command governors, input shapers modify the input to the system, however, they are typically not designed to explicitly enforce state and control constraints. The feedforward and reduced order reference governors may be suitable for problems where input shaping has traditionally been used and there are constraints; however, their properties in such applications remain to be further studied.
VII. RESEARCH TOPICS

While the subject of reference governors has been researched for over twenty years and connects naturally with Modern Predictive Control, a variety of new research directions can be identified.

The nonlinear reference governor results (see e.g., [50], [123] and references therein) extend to the case when the system has uncertain set-bounded constant parameters. At the same time, a non-conservative application of reference governors to systems with uncertain parameters being estimated online remains an area to be explored further. Special assumptions appear to be necessary in this case if one wishes to guarantee recursive feasibility and other reference governor properties.

The treatment of the case when constraints are time-varying or reconfigured dynamically is of interest for many applications. While we have had success in treating specific examples, see e.g., [89], the theory remains largely to be developed.

Traditionally, reference and extended command governors modify the set-points to closed-loop systems. While the theory assumes that the set-points are given, in real systems the set-points may be adjusted by a human operator (or a higher level automatic controller) in response to external conditions. This dependence of set-points on external conditions can create a feedback loop encompassing the reference governor and nominal closed-loop system. A closely related situation occurs when the governor augments control signals at the actuator command level. This implementation of reference governors is appealing as the design and calibration of the nominal controller can be changed without the need to redesign the governor. The properties of the reference governor in the loop remain to be studied.

Error governors are schemes related to reference governors, however, they act on the tracking error at the controller input and not on the reference command. They are also primarily intended for handling control input constraints and not output constraints. See [124], [47], [70], [132]. As compared to reference governors, error governors have received relatively little attention; obtaining convergence guarantees for them that are similar to reference/command governor has been elusive. It is interesting that the error governor can be applied with relative ease to handle constraints in direct adaptive control [73].

Finally, we mention the topic of handling constraints in systems affected by large disturbances. Such problems arise in control of aerospace systems, such as a very flexible aircraft [39] of a flying wind turbine [67], where a large disturbance is due to wind. The assumption that the disturbance is set-bounded can lead to conservative governor designs or may not be even feasible. Less conservative approaches that incorporate disturbance models, dynamic disturbance constraints, and stochastic constraint enforcement guarantees can be beneficial in these applications.


